



ESCOLA POLITÉCNICA
DE PERNAMBUCO



Departamento de
Sistemas
Computacionais

Sistema de Edição Colaborativa de Artigos Científicos - WebLatex

Trabalho de Conclusão de Curso

Engenharia da Computação

Assis Clemente

Orientador: Prof. Tiago Massoni

Recife, 21 dezembro de 2007





ESCOLA POLITÉCNICA
DE PERNAMBUCO

POLI
ESCOLA POLITÉCNICA
DE PERNAMBUCO



Departamento de
Sistemas
Computacionais

Sistema de Edição Colaborativa de Artigos Científicos - WebLatex

Trabalho de Conclusão de Curso

Engenharia da Computação

Este Projeto é apresentado como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Assis Clemente

Orientador: Prof. Tiago Massoni

Recife, 21 dezembro de 2007



Assis Clemente da Silva Neto

Sistema de Edição Colaborativa de Artigos Científicos - WebLatex

Resumo

Atualmente, devido ao elevado custo para estabelecer reuniões presenciais e a dificuldade de reunir um grupo em horários regulares que satisfaçam a todos, aliado às novas formas de comunicação remota, como correio eletrônico, conferência por telefone, vídeo conferências e listas de discussão, as pessoas procuram reduzir o número de reuniões reais. Para este objetivo, a colaboração remota auxiliada por aplicações de *groupware* apresenta-se como uma alternativa à realização de tarefas compartilhadas. Ambientes de colaboração auxiliam com sua estrutura para o desenvolvimento grupal de um determinado projeto, proporcionando uma interação mais eficiente entre os membros que formam o grupo.

O propósito deste trabalho é o desenvolvimento de uma ferramenta de edição colaborativa de projetos, sendo que esses projetos são artigos científicos. Será mostrada a importância do trabalho colaborativo para o desenvolvimento de artigos científicos, assim como a contribuição da ferramenta proposta.

Abstract

Currently, due to the high cost to establish meetings presence and the difficulty of bringing together a group on regular schedules that meet to all, combined with new forms of remote communication, such as electronic mail, conferencing by telephone, video conference and mailing lists, people seeking reduce the number of actual meetings. For this goal, assisted by the collaboration remote applications of groupware are presented as an alternative to the realization of shared tasks. Environments of collaboration help with its structure to the development group for a particular project, providing a more efficient interaction between members that make up the group.

The purpose of this work is to develop a tool for collaborative editing of projects, and that those projects are scientific articles. It will be shown the importance of collaborative work for the development of scientific articles, as well as the contribution of the tool motion.

Sumário

Índice de Figuras	vii
Índice de Tabelas	viii
Tabela de Símbolos e Siglas	ix
1 Introdução	11
1.1 Objetivos e Metas	12
1.2 Visão Geral do Trabalho	12
2 Produção Colaborativa	14
2.1 Introdução	14
2.2 Edição Colaborativa Suportada por Computador	15
2.3 Aprendizagem colaborativa	18
2.4 Características das Ferramentas colaborativas	21
2.5 Ferramentas Relacionadas	23
3 Latex - edição de textos científicos	25
3.1 Introdução	25
3.2 O que é Latex?	25
3.3 Texto, comandos e Ambientes	28
3.4 Distribuições Latex	32
4 Desenvolvimento	34
4.1 Requisitos	34
4.2 Tecnologias	38
4.3 Arquitetura	39
5 O Sistema	42
5.1 O WebLatex – A Ferramenta de execução	42
5.2 A Ferramenta de execução	43
6 Considerações Finais	48
6.1 Conclusões	48
6.2 Contribuições	48
6.3 Trabalhos Futuros	50

Índice de Figuras

Figura 1. Formas básicas de distribuições de papéis	17
Figura 2. Arquitetura de editor de texto colaborativo Centralizado	20
Figura 3. Esquema de apresentação de um arquivo Latex	26
Figura 4. Símbolos Latex	30
Figura 5. Diagrama de Casos de Uso do WebLatex	34
Figura 6. Modelo PAC*	38
Figura 7. Diagrama de classes simplificado da camada de persistência do WebLatex	39
Figura 8. Tela Inicial do WebLatex	41
Figura 9. Tela de Gerenciamento do WebLatex	42
Figura 10. Fragmento da Tela de Cadastro do WebLatex	43
Figura 11. Convite de Colaboradores para um Projeto do WebLatex.	43
Figura 12. Tela de edição de seção	44
Figura 13. Visualização de uma seção que esta sendo editada por outro membro	45

Índice de Tabelas

Tabela 1. Matriz Tempo x Espaço proposta por Ellis [9]	14
Tabela 2. Ambiente itemize	27
Tabela 3. Ambiente Enumerate	28

Tabela de Símbolos e Siglas

J2EE - *Java 2 Enterprise Edition*

CSCL - *Computer Supported Collaborative Learning*

CSCW - *Computer-Supported Cooperative Work*

WYSWIS - *What You See is What I See*

WYGIWIG - *What You Get Is What I Get*

WYSIAWIS - *What You See Is Almost What I See*

WYSIWYG - *WHAT-YOU-SEE-IS-WHAT-YOU-GET*

HTML - *HyperText Markup Language*

ASCII - *American Standard Code for Information Interchange*

SQL - *Structured Query Language*

PAC - *Presentation, Abstraction, Controller*

MVC - *Model – View – Controller*

JVM - *Java Virtual Machine*

GNU - *GNU Não é Unix*

GPL – *General Public Licence*

Agradecimentos

Agradeço primeiramente a Deus por toda minha vida e por todas as coisas que ele tem me proporcionado.

A todos meus avôs que me deram apoio desde o meu nascimento.

À minha mãe (Marlécia Guerra Clemente da Silva) pelo carinho, amor, dedicação e compreensão.

Ao meu pai (Assis Clemente da Silva Filho) por todos os ensinamentos, proteção e apoio em todos os momentos de minha vida.

Aos meus irmãos Anderson e Antonioni, por todo o apoio, ajuda e incentivo.

Ao meu professor, orientador e AMIGO Tiago Massoni pela ajuda, paciência e dedicação para que este trabalho fosse concluído com sucesso e a todos os professores da UPE que participaram da minha formação.

Aos meus amigos da empresa mWare, por todos os ensinamentos e incentivos para que este trabalho fosse concluído. A todos aqueles que contribuíram para a pesquisa realizada neste trabalho.

Deus, obrigado por tudo que eu tenho nesta vida e por todas essas pessoas que constituem a pequena história da minha vida.

Capítulo 1

Introdução

A produção de conhecimento é um fator importante para todo centro de pesquisa, ou seja, gerar conhecimento de qualidade e de fácil assimilação é almejado por todo profissional, que queira o reconhecimento de seu trabalho publicado. As publicações científicas eletrônicas na Internet constituem atualmente um dos temas de maior repercussão dentro da comunicação científica [1], pois apresentam uma maior flexibilidade e variedade de aplicações quando comparadas com as baseadas em papel. A preparação dos textos deve seguir formalidades, para que qualquer pessoa da comunidade científica tenha condições de ler e absorver o conteúdo proposto. Para isso um dos pacotes mais aceitos de produção é o Latex.

O pacote Latex foi desenvolvido em 1985 por Leslie Lamport [6], a partir do programa TEX criado por Donald Knuth [6]. Ele é um sistema de formatação e configuração de layout, que permite aos autores criar e imprimir os seus documentos com a mais alta qualidade tipográfica, onde através de comandos pode-se até mesmo definir fórmulas matemáticas e citações em seu texto. Esse pacote é de domínio público o que faz existirem diversas implementações.

Ainda no contexto de pesquisas científicas, grande parte delas é composta por grupos [2] e gerenciar a produção de conhecimento nesse ambiente colaborativo não é uma tarefa trivial. Para um cientista, documentar os seus trabalhos é tarefas que exigem muito cuidados e precisão, todavia para um conjunto de pessoas realizarem essa documentação, surgem problemas de compartilhamento, onde vários usuários devem utilizar o mesmo trabalho científico o qual deve ser coordenado e sincronizado entre outros. Nesse caso a comunicação é de extrema importância, seja ela para trocar idéias, negociar, discutir, aprender e até mesmo tomar decisões [2]. Ao mesmo tempo os computadores oferecem recursos que podem ser usados nestas tarefas que envolvem concorrência e comunicação,

dando suporte ao processo colaborativo. Como exemplo, pode-se desenvolver uma ferramenta computacional, onde se aplica recursos de controle de concorrência, o qual oferece áreas exclusivas de edição (no caso de um texto compartilhado), sem que o trabalho seja interferido por outro usuário e, além disso, ofereça a funcionalidade para outros usuários de acompanhar o trabalho do colega (melhorando a comunicação) e opinar sobre seu progresso.

A estratégia de ferramentas de edição que suportam a colaboração via computador é incentivar os participantes a trabalharem em grupo num documento comum, respeitando as características individuais. Usando esses argumentos, o desenvolvimento de uma ferramenta web colaborativa para elaboração de artigos científicos baseada em modelo Latex é de grande valia para a comunidade científica, já que incentiva e facilita o desenvolvimento dos artigos científicos em um modelo amplamente aceito e com a colaboração remota de todos os integrantes.

1.1 Objetivos e Metas

O objetivo deste trabalho é elaborar um sistema colaborativo que gerencie, dinamize e torne mais eficiente o desenvolvimento de artigos, utilizando o compilador Latex para estruturar e desenvolver o texto científico na internet, o que aumenta a flexibilidade de acesso e propicia uma maior interação por conta dos usuários devido às suas funcionalidades. Além disso, o sistema terá mecanismos de segurança para que pessoas devidamente autenticadas tenham acesso ao artigo em produção de forma que o texto possa ser editado e gerenciado remotamente.

O trabalho enfatiza desenvolver esse ambiente colaborativo que apresente controle de concorrência para que mais de um usuário possa editar o artigo em construção, oferecendo o status de desenvolvimento, de forma que se uma determinada pessoa estiver trabalhando nele os demais terão acesso de apenas leitura daquela região do texto e observarão em tempo real a produção do usuário, o qual tem permissão de escrita no momento.

1.2 Visão Geral do Trabalho

O trabalho está organizado da seguinte maneira. No Capítulo 2 abordamos os conceitos de edição colaborativa, apresentando algumas vantagens do seu uso em projetos e suas características.

O Capítulo 3 define a ferramenta Latex usada para a produção dos textos científicos no projeto, apresentando suas vantagens, características e impactos para o projeto.

O Capítulo 4 apresenta os elementos utilizados para a construção da ferramenta proposta, as tecnologias usadas, os requisitos do sistema assim como sua arquitetura e a forma que foi implementado.

No Capítulo 5 é demonstrado o sistema e uma avaliação do mesmo.

Por fim, no Capítulo 6, temos as conclusões, as principais contribuições deste trabalho e as propostas para trabalhos futuros.

Capítulo 2

Produção Colaborativa

Este capítulo apresenta o conhecimento necessário em edição colaborativa para o desenvolvimento da ferramenta, conceitos de *groupware* (software em grupo) e aprendizagem colaborativa.

2.1 Introdução

A área computacional, junto com a internet vem tendo grandes avanços, associado ao mundo globalizado onde a comunicação deve ser bastante eficiente, surgem diferentes métodos e ferramentas para tornar a educação e a produção de conhecimentos cada vez mais interativos e objetivos. Ferramentas de *groupware*, ou seja, os softwares e hardwares que suportam e ampliam a aprendizagem e o trabalho em grupo [9], fazem da educação à distância um ponto forte na produção de conhecimento, através do uso de métodos colaborativos. Ambientes como estes tendem a ficar cada vez mais comuns, com os recursos disponíveis oferecidos pela internet, e muitos trabalhos científicos comprovam que o uso destas ferramentas em escolas, facilita o trabalho em conjunto e a construção de conhecimento [16].

Muitas ferramentas no mercado podem ser usadas como ambiente de colaboração, como exemplo tem listas de discussão, onde os usuários inserem em páginas da internet suas opiniões a respeito de um assunto qualquer de seu interesse, chats que propiciam a comunicação por texto em tempo real, e-mails, videoconferências, skype, whiteboard e até mesmo editores de texto colaborativo, esses também são vistos como importantes canais de comunicação. No entanto, o fato de se personalizar a aplicação para resolver o problema específico é o que a torna mais eficiente e maximiza a produtividade e o aprendizado, pois garante uma solução específica para aquela realidade.

Esses ambientes devem possuir características como criatividade, autonomia, criticidade e cooperação [16]. Uma das áreas de pesquisa de ambientes cooperativo é conhecida como aprendizagem colaborativa suportada por Computador (Computer Supported Collaborative Learning - CSCL) que pode ser vista como uma subdivisão do Trabalho Cooperativo Suportado por Computador (Computer-Supported Cooperative Work - CSCW) que constituem disciplinas do *groupware*.

2.2 Edição Colaborativa Suportada por Computador

A área de pesquisa do CSCW, aplicada nas atividades em grupo tecnologias de comunicação associadas a técnicas de computação, pode ser encontrada na literatura brasileira [10] pelo nome de SATC (Sistema de Apoio ao Trabalho Colaborativo), sendo que quando o objetivo está relacionado à aprendizagem estamos falando de aprendizagem colaborativa suportada por computador (CSCL), que pode ser definida como uma estratégia educativa em que dois ou mais sujeitos constroem o seu conhecimento através da discussão, da reflexão e tomada de decisão, onde os recursos informáticos atuam como mediadores do processo de ensino-aprendizagem [27].

A CSCL cresceu sobre a CSCW, a qual surgiu na década de 80 e seu estudo apresenta diversos aspectos que incluem conceitos das áreas como a psicologia, sociologia e educação, para apoiar o estudo de componentes organizacionais, individuais e de dinâmica das interações do grupo [10]. Os resultados de pesquisas do CSCW dão origem a novo conceito importante para esse trabalho que são os sistemas computacionais que apóiam grupos de pessoas engajadas em uma tarefa ou objetivo comum e que provêm uma interface para um ambiente compartilhado [9], conhecido como *groupware*.

Os conceitos do CSCW e *groupware* apresentam laços estreitos, tanto que é comum haver confusão quanto ao uso deles no contexto cooperativo. Mas o objetivo do *groupware* é simplesmente utilizar elementos como colaboração, comunicação e coordenação de atividades, aliados às tecnologias computacionais existentes para facilitar e tornar mais eficiente a realização de tarefas cooperativas, enquanto que o CSCW é a área que representa a produção cooperativa e como o computador apóia a execuções dessas atividades.

As Pesquisas em cima do CSCW resultam em características que classificam os sistemas *groupware* existentes, o que ajuda a entender como funcionam. Um critério bastante usado para classificar esses sistemas é a interação do usuário com a aplicação. Por esse critério divide em dois tipos: síncronos e assíncronos.

- Síncronos: Se destaca pelo fato que a ação realizada por um dos membros da aplicação é instantaneamente identificada pelos outros do grupo, através de algum meio de percepção apresentado pela aplicação, que pode ser uma letra de cor diferente, por exemplo. Assim os usuários realizam suas atividades simultaneamente e toda modificação é imediatamente identificadas por todos os membros do projeto.
- Assíncrono: Cada membro realiza suas atividades, com uma cópia do projeto em que estão trabalhando, dessa forma, quando os membros reportam as atividades, é que os outros tomam conhecimento de suas modificações, ou seja, eles trabalham com os mesmos objetos só que em tempos diferentes e existe um repositório que é responsável por reunir as suas informações e reportar para todas as modificações realizadas.

Outros critérios utilizados para classificação são: tempo, espaço, tamanho do grupo, funcionalidade da aplicação e tipos de interface, onde se pode resultar nas chamadas matrizes de classificações de *groupware*. Um exemplo é a matriz classificatória de Ellis [10], que utiliza os critérios de tempo e espaço, os quais representam dimensões distintas na representação. O tempo define em que momento os membros participam, através das premissas *mesmo tempo* e *tempo diferente* e o espaço envolve a localização dos membros, e pode ser dividido em *mesmo local* e *local diferente*. Pode se observar a matriz na Tabela 1.

Tabela 1 – Matriz Tempo x Espaço proposta por Ellis [9]

	Mesmo Tempo	Tempo Diferente
Mesmo Local	Interação Síncrona (face-a-face)	Interação Assíncrona
Local Diferente	Interação Síncrona Distribuída	Interação Assíncrona Distribuída

De acordo com a tabela os sistemas de groupware são divididos em quatro grupos distintos, que são gerados pelos valores das dimensões na tabela. Como exemplo uma aplicação de vídeo conferência seria classificada como uma interação Síncrona Distribuída.

Os critérios acima usados servem para determinar as propriedades dos sistemas groupware, mas nada dizem sobre as funcionalidades dos mesmos, então é necessária uma classificação que indique os sistemas de acordo com suas funcionalidades.

Com base nas funcionalidades indicadas por Ellis [9] temos as seguintes classes de aplicações:

- Sistemas de mensagens: como exemplo tem o correio eletrônico;
- Salas de Reunião Eletrônicas: são sistemas que oferecem ambientes especiais dotados de um ótimo suporte de software e hardware para gerenciar e realizar reuniões face-a-face, entre as funcionalidades temos assistência à preparação da pauta da reunião, à geração de idéias, à organização das idéias propostas e à tomada de decisão, onde os hosts estão interligados em rede e fazem uso de telões computadorizados e equipamentos de áudio e vídeo;
- Sistema de Apoio a Decisão do Grupo: São sistemas que suportam problemas não estruturados ligados a mecânica de grupos com a utilização de elementos computacionais;
- Conferências: Permite que os usuários desse sistema, troquem informações com outros usuários em tempo real, um bom exemplo é o chat cuja informação em questão é texto, outros sistemas utilizam vídeo ou áudio, são os chamados sistemas de teleconferência;
- Sistemas de coordenação: Estes sistemas são responsáveis pela integração e o ajuste harmonioso dos esforços individuais dos participantes em rumo à obtenção de um objetivo geral [10].
- Agentes inteligentes: Faz uso de agentes computacionais que realizam tarefas específicas como se fossem pessoas reais;

- Editores multiusuários: Permitem a edição colaborativa de um texto compartilhado, é a proposta desse trabalho.

2.3 Aprendizagem colaborativa

Um editor multiusuário é uma ferramenta de edição, usada para trabalhar em cooperação para desenvolvimento de um texto, no caso o espaço compartilhado é o texto em construção, onde todos os membros irão interagir de acordo com as funcionalidades propostas pelo sistema, que deve ser de preferência estruturado, ou seja, uma ferramenta modelada para suportar e facilitar um tipo de tarefa específica ou um trabalho de um grupo específico [28], e que detenha a fundo os conceitos de colaboração aplicados à aprendizagem. Em contrapartida, os sistemas não-estruturados, podem deixar de modelar certos elementos associados à realidade da aplicação, e para suprir essa necessidade do projeto, os usuários são obrigados a ter um maior conhecimento das possibilidades do sistema e muitas vezes limitam sua produção aos recursos disponíveis.

Então, é importante uma ferramenta que ofereça uma modelagem estruturada, a fim de se ganhar a característica de maior interação, que diminua as barreiras de acesso de tempo, espaço e idade, que possa atender a um grande número de pessoas, em um único momento, a um custo bastante reduzido, que motive o usuário a permanecer no ambiente colaborativo conectado ao trabalho, sem ter que sair de seu ambiente familiar, profissional, etc., possibilitando a flexibilidade, de cooperar onde quiser, quando quiser e no ritmo que preferir. Mas, com tarefas e prazos bem definidos, para que a produção de conhecimento evolua num ritmo semelhante entre os membros do grupo.

Para que características como estas tenham efeito, além do sistema é fundamental se ter um grupo comprometido na elaboração do projeto, pois problemas como os de se ter poucas reuniões reais com os outros membros, menor socialização entre os mesmos, descaso das tarefas apresentadas, por não ter uma pessoa física cobrando como em um ambiente real, podem fazer um projeto fracassar, além de outros problemas, como o empobrecimento da experiência de troca entre os participantes.

Alguns elementos são recomendados para escolha de um bom grupo. Laborde defende que, ao se reunir pessoas para realizar uma tarefa não há garantia da ocorrência de processos de troca cooperativa [7]. Sendo assim ela defende um conjunto de condições que devem ser aplicadas para favorecer uma interação social que gere ‘frutos’ na resolução de um trabalho em grupo. Entre estas condições, está a escolha dos parceiros (colaboradores), onde o responsável pelo projeto deve escolher membros que tenham opiniões e soluções contrárias, para favorecer o debate e que cada um defenda seu ponto de vista, a fim de se ter

uma nova solução, mas que não seja um grupo muito heterogêneo, pois corre o risco dos parceiros não compreenderem as proposições um dos outros.

Outra condição é a escolha da tarefa, onde os membros aplicarão todos seus conhecimentos. Mas estes não devem ser suficientes para resolver todo o problema, pois isso impede a interação cooperativa entre os colaboradores. É necessário estimular o planejamento conjunto, promover diferentes perspectivas, que gerem múltiplas soluções permitindo a aquisição de habilidades. É fato, que algumas tarefas são mais compartilháveis que outras, por isso em um processo devem-se enfatizar o trabalho colaborativo, quando a situação assim o permitir, como, por exemplo, o planejamento de um projeto deve ser feito em grupo, para permitir a interação e não por apenas um indivíduo.

As tarefas geralmente são limitadas por prazos, e durante esse período é necessário levar em consideração a latência de interação entre os indivíduos, pois alguém só aplica a idéia de outro efetivamente, quando tem a necessidade de usá-la e não quando simplesmente toma conhecimento dela.

O tamanho do grupo também influencia, mas depende bastante do tipo de tarefa e o meio de comunicação que é utilizado e isso também tornam a aplicação um fator importante, pois a tecnologia nela pode favorecer o sucesso da tarefa.

As atividades no grupo podem ser genéricas, que se aplica a qualquer problema, ou específica para uma solução. No entanto a distribuição dessas tarefas pode ser dividida em quatro formas básicas [7]:

- Cada membro possui uma única responsabilidade: Só uma pessoa realiza a execução de uma determinada responsabilidade. (figura 1 (a))
- Mais de uma pessoa possui a mesma responsabilidade: neste caso mais de uma pessoa compartilha a responsabilidade de realizar determinadas atividades; (figura 1 (b))
- Sem responsabilidades definidas: todos os participantes realizam a execução de todas as atividades e; (figura 1 (c))
- Solução individual compartilhada: cada pessoa resolve o problema sozinho e então as diferentes soluções são apresentadas e comparadas. (figura 1 (d))

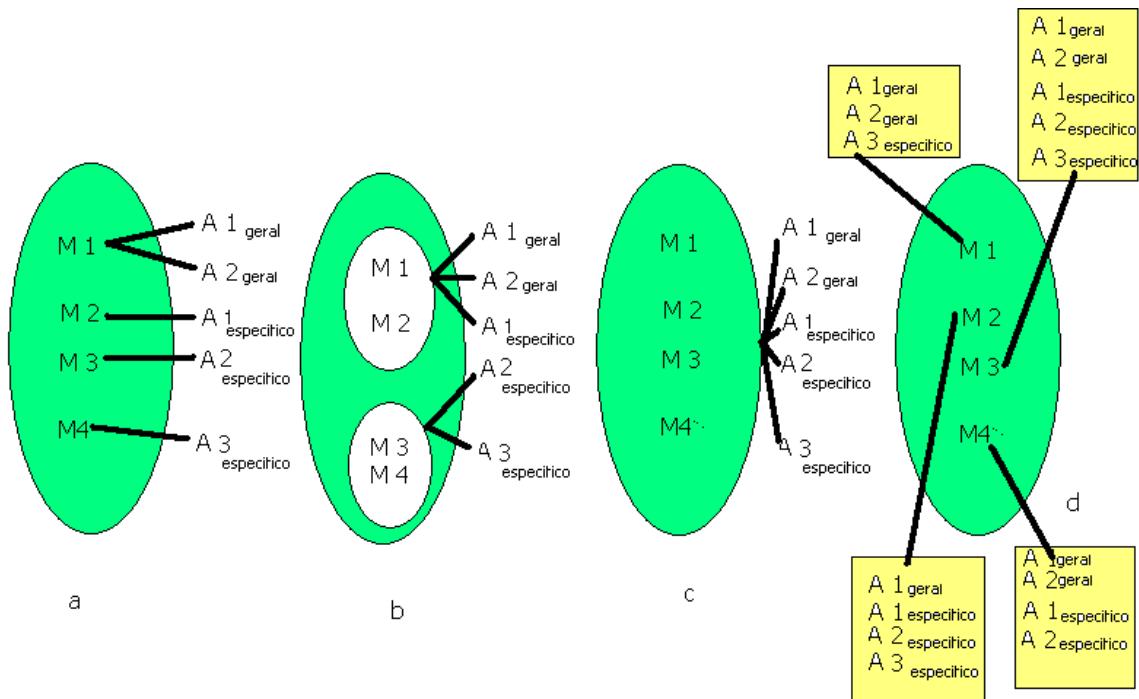


Figura 1. Formas básicas de distribuições de papéis

Então, cada pessoa do grupo, deve ter uma responsabilidade definida, para realizar bem suas atividades, de forma que se realize, uma boa interação e participação, definindo bem as tarefas do grupo e a gerencia dos elementos compartilhados. Cada responsabilidade dos participantes distingue bem as atribuições e responsabilidades de cada colaborador no sistema e garantem uma boa coordenação e controle aos elementos compartilhados. O ponto negativo está quando se utiliza de maneira rígida essas definições, o que pode restringir a capacidade de um individuo no projeto, pois um individuo fica limitado às suas atividades, mas cabe ao responsável do projeto definir bem as tarefas aproveitando ao máximo as habilidades de cada componente. Uma boa coordenação das atividades exige um complexo controle de concorrência. Mecanismos de controle de concorrência, como travas ou transformação operacional, são utilizados para garantir a consistência dos elementos, que estão sendo manipulados pelos participantes da colaboração, no documento compartilhado [16].

As tarefas cooperativas ainda podem classificas em:

- Divergentes: que procuram produzir os conceitos em conjunto estabelecendo metas e;
- Convergentes: onde se busca a solução situação-problema [7], ou seja, os membros estão empenhados em um objetivo comum.

Então não basta apenas ter um editor multiusuário que ofereça um artigo compartilhado. É necessário nesse ambiente estabelecer responsabilidades para os membros, e para cada participante, estabelecer quais atividades (tarefas) deve-se seguir para que o projeto tenha prazo e comprometimento bem definidos.

2.4 Características das Ferramentas colaborativas

Para se desenvolver uma ferramenta colaborativa é necessário conhecer as principais características que são encontradas nas ferramentas existentes. Abaixo temos alguns dos elementos que devem formar uma ferramenta colaborativa:

- Compartilhamento: Caso em que vários usuários têm acesso a um determinado recurso, sendo que esse recurso é gerenciado de forma que nenhum usuário pode alterar algo que outro usuário já esteja alterando (controle de concorrência), pois é necessário que todos estejam utilizando a mesma informação sem inconsistências, o que deve garantir trabalhos não-repetitivos.
- Comunicação e Coordenação: A comunicação tem grande impacto, pois ela fornece a troca de idéias, discussões, negociações, tomada de decisões, aprendizado, entre outras e pode ser aplicada como síncrona assíncrona e multisíncrona, o que depende de como e quando cada participante recebe a mensagem no processo. Mas outras formas de comunicações podem ser usadas no que diz respeito à ligação entre os participantes, que vai da comunicação direta, como uma reunião, até a indireta como a memória de grupo, onde a produção de conhecimento em conjunto pode caracterizar uma forma de comunicação. Na coordenação os usuários se organizavam para utilizarem e realizarem suas atividades de forma eficiente. São necessários mecanismos que garantam o controle de concorrência, por exemplo, em ambientes síncronos, mas em sistemas onde a capacidade de conflito for baixa pode se utilizar o chamado protocolo social onde os membros do grupo se organizam de forma a distribuir as iterações pelo meio de comunicação.
- Percepção (*awareness*): A percepção é mostrada através de um sinal que é passado para os membros da tarefa colaborativa e dependendo da quantidade e qualidade da informação de percepção (*awareness information*) contida no sistema eles têm uma percepção bem detalhada dos membros em relação aos outros. Como exemplo pode ser apresentado uma cor diferente ou representações gráficas que permitem transferir informações a respeito das atividades dos demais participantes para cada participante [29]. Em ferramentas assíncronas o fator que determina a percepção é o tempo, neste caso um colaborador faz um trabalho e posteriormente envia aos demais do grupo. Essa informação complementa a comunicação explícita, ou seja, aquela feita por chats ou voz.

- Designação de papéis: É a responsabilidade atribuída a cada membro que usa a ferramenta colaborativa, indicando seu modo de participação e forma como irá interagir com os demais, ou seja, define as suas atividades, gerencia o acesso aos recursos compartilhados e estrutura as interações entre os membros, mas deve-se ter cuidado para que não limite a criatividade e produtividade de alguns integrantes dependendo da coordenação de como as tarefas dos membros devam ser realizadas.
- Visualização: Uma das formas de visualização do espaço compartilhado é o estilo *What You See is What I See (WYSWIS)*. Existem dois tipos desse estilo o *strict* que é um pouco rebuscado pelo fato de que a ação de um membro tem influência sobre o trabalho de todos os outros o que o torna problemática para um grande número de participantes no projeto. Um exemplo de *WYSWIS strict* (pesquisar) é o NetMeeting da Microsoft. O *WYSWIS relaxed* é um estilo mais aceitável pelos membros, pois a sua visualização é individual, ou seja, cada membro trabalha do seu modo sem interferir os outros. Outro modelo é o *What You Get Is What I Get (WYGIWIG)* onde as interações dão-se de forma assíncrona onde cada membro se torna defasado em relação ao conteúdo real, pois só quando determinadas condições forem verdadeiras é que as alterações de um membro serão propagadas. Já o modelo *What You See Is Almost What I See (WYSIAWIS)* cada alteração é propagada com pequenas variações com a condição de que não comprometa a interpretação por partes dos outros membros, existem outros modelos e até mesmo pequenas variações desses demonstrados.

- Arquitetura: Pode ser distribuída, onde o armazenamento dos documentos está em diversos hosts. Tipicamente, o servidor armazena o modelo do documento e os clientes possuem uma cópia (do documento). Desta forma, apenas os eventos dos modelos, situados no servidor, são atualizados nos hosts clientes. Uma vantagem desta arquitetura é que os clientes podem trabalhar independente do funcionamento da rede. Já a arquitetura centralizada (figura 2) mantém apenas uma aplicação e envia mensagens do evento a todos os clientes. Quando um arquivo é atualizado, todas as alterações são enviadas para o servidor, que por sua vez, enviará para os demais usuários, caso estejam conectados. Assim, o servidor mantém arquivos compartilhados de forma consistente. Um problema encontrado nessa arquitetura é que se o servidor não estiver funcionando, todos os usuários não podem trabalhar, mas em contrapartida todos os usuários estarão trabalhando com a versão mais recente da produção, evitando retrabalho, e que os membros realizem a mesma tarefa que outro já esteja trabalhando.

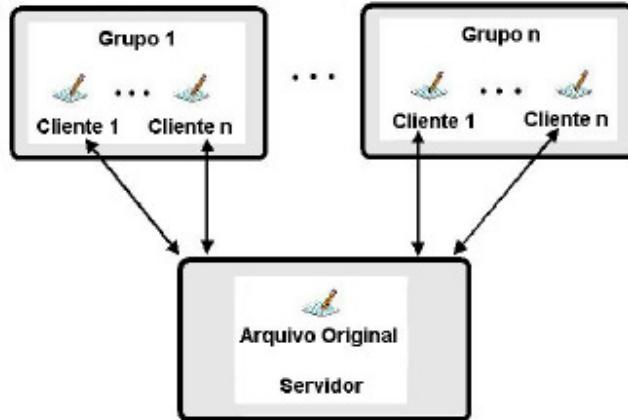


Figura 2. Arquitetura de editor de texto colaborativo Centralizado[4]

2.5 Ferramentas Relacionadas

Algumas das ferramentas de escrita colaborativa, que seguem como casos semelhantes no mundo groupware ao WebLatex. A seguir uma breve descrição dessas ferramentas:

- **Google Docs** em <http://docs.google.com>
 - Editor semi-síncrono, porque fica constantemente atualizando a tela, em intervalos regulares, aceita *uploads* de documentos do MS Word para edição e salva como o formato doc. Usa o navegador como processador de texto.
- **Gobby** em <http://gobby.0x539.de/>
 - Editor síncrono mais focado para criação de código em grupo funciona em Windows, Mac, Linux e Unix.
- **EquiText** em <http://equitext.pgie.ufrgs.br/>
 - Usa a web como editor, foi feito por um grupo brasileiro.
- **LiveDrive and Collaboration Manager for Word** em <http://www.chasseral.com/products/index.shtml>
 - Plug-in comercial para o *Microsoft Word*, pois permite que grupos trabalhem no Word compartilhando uma cópia de um único documento.
- **Bloki** em www.bloki.com
 - Usado para criação compartilhada de páginas web, pode ser usado também para escrita colaborativa. Usa a web como processador de texto.

- **DocReview** em <http://purl.oclc.org/DocReview/get>
 - Unix apenas permite escrita de comentários em algum documento, usa a web para armazenar comentários, não modifica o arquivo original.
- **SubEthaEdit** em <http://www.codingmonkeys.de/subethaedit/>
 - Para o Mac OS X. SubEthaEdit é focado na criação de código colaborativo, mas pode funcionar como editor colaborativo em tempo real. Ferramenta peer-to-peer.
- **SynchroEdit** em <http://www.synchroedit.com/>
 - Editor síncrono multiplataforma, que usa a web para processar o texto.
- **SparrowWeb** em <http://www.alphaavenue.com/details.php?tech=Sparrow%20Web>
 - Usado na web, com muitas características de comunidade virtual.
- **MoonEdit** em <http://me.sphere.pl/indexen.htm>
 - Editor de texto multiplataforma, permite que várias pessoas trabalhem ao mesmo tempo no mesmo documento pela web.
- **Web Collaborator** em <http://webcollaborator.com/>
 - Gratuito se usado no servidor padrão, mas é pago para ser instalado em um servidor particular. Oferece um fórum para o grupo discutir o texto e histórico de versões além de RSS feed. Converte documento HTML para pdf.
- **TellTable** em <http://telltable-s.sourceforge.net/>
 - Precisa ser instalado em um servidor Unix ou Linux, usa o OpenOffice como processador de texto, mas abre uma janela do OpenOffice no navegador.
- **REDUCE** em <http://reduce.qpsf.edu.au/>
 - Esse sistema permite edição síncrona. Usa a web como processador.
- **CoWord** em <http://reduce.qpsf.edu.au/coword/>
 - Mesmos criadores do REDUCE. Permite a edição síncrona de documentos no Microsoft Word.

Capítulo 3

Latex - edição de textos científicos

Neste capítulo será apresentado os conceitos básicos do sistema tipográfico digital Latex seu surgimento e suas principais características.

3.1 Introdução

O Latex é um sistema tipográfico digital que utiliza o Tex [6] como base de processamento, é muito popular na comunidade acadêmica e científica, e extensivamente usado na indústria [6]. Antes de conhecê-lo a fundo é necessário ter uma visão sobre o Tex. Tex é uma aplicação criada por volta dos anos 70 por Donald Knuth, cujo principal objetivo era aumentar a qualidade da impressão para as impressoras da época. Idéia que surgiu ao receber da editora uma revisão de seu livro *The Art of Computer Programming* com baixa qualidade tipográfica. O seu motor tipográfico começou a ser escrito em 1977, para explorar o potencial dos equipamentos digitais de impressão que estavam a infiltrar na indústria de publicação naquele tempo [15]. De sua implementação original pouca coisa foi melhorada, até o ano de 1989, onde acrescentou um melhor desempenho para caracteres de 8-bits, múltiplas linguagens e uma estrutura estável que até hoje é apresentada. Usado em diferentes arquiteturas de computadores, ele também é utilizado para o processamento de fórmulas matemáticas e texto.

3.2 O que é Latex?

O Latex pode ser visto como um conjunto de macros do Tex, onde utiliza seu conjunto de comandos com diferentes funções para preparação de documentos. Foi desenvolvido por Leslie Lamport na década de 80, com a finalidade de facilitar o uso do Tex. Ele é adequado para a confecção de longas monografias, artigos. Pode ser considerado um editor voltado para área matemática, pois contém uma variedade de comandos para construir diversas fórmulas

matemáticas, além de poder ser utilizado para outros tipos de documentos, que pode ser de uma simples carta até um livro.

Para entender o uso do Latex, faz-se necessário conhecer o processo de publicação de um trabalho, na época em que suas funcionalidades foram desenvolvidas.

Para publicação de trabalhos os autores constroem seu texto e o enviam para uma editora, onde esta possui empregados especializados (Designer), que estudam a melhor forma de criar o formato do documento, que constituirá o texto final pronto para o público, ou seja, o designer decide sobre o layout do manuscrito (fonte, tamanho da letra, largura da coluna, margens, etc.), baseado na percepção do que o autor deseja e em seu conhecimento a respeito das regras de tipografia, a fim de que o conjunto demonstre o sentimento do autor e facilite a leitura por parte do público. Logo após a finalização desses requisitos o designer entrega o resultado de seu trabalho ao tipógrafo, que o interpreta e gera a matriz para impressão do documento. Voltando ao contexto original, o Latex representa o designer no processo de desenvolvimento e o tex representa o tipógrafo.

No entanto, o Latex é um software e por isso necessita de ajuda humana para realizar suas tarefas, no qual é passado como entrada comandos que resultam na estrutura lógica do trabalho. Assim durante o desenvolvimento de um texto com o Latex, não é possível visualizar interativamente o documento enquanto se escreve, como muitos processadores de texto, mas pode-se realizar uma pré-visualização do documento e realizar possíveis correções caso necessário. Para facilitar o entendimento do Latex no processamento do texto, é necessário saber como se classificam os tipos processadores de textos.

Os processadores de texto podem ser divididos em duas classes:

- Estilo Visual: São utilizados em processadores de texto, onde são apresentados menus com diversos recursos, podendo ser usado o mouse. O texto digitado aparece para o usuário, no momento da entrada do texto na tela, da mesma forma que será impresso. Este é o chamado *WHAT-YOU-SEE-IS-WHAT-YOU-GET*. Como exemplo, tem-se a *Microsoft Word*.

- Estilo lógico: Nesta classe o processamento é feito em duas etapas. Na primeira etapa o texto e os comandos de formatação são escritos em um arquivo fonte, com o uso de um editor qualquer, e logo depois na segunda etapa este arquivo é compilado, onde gera um arquivo de saída que pode ser visualizado. Como exemplo no tem-se o *HTML*.

Ao usar estilo lógico, que apresenta duas etapas, embora pareça mais complexo, tem algumas vantagens de flexibilidade, pois pode mudar toda formatação do texto editando apenas alguns comandos. O Latex segue o estilo lógico, com isso o usuário apenas se preocupa com a geração do conteúdo, sendo a estética e o estilo de boa qualidade é garantido pelos diversos formatos profissionais disponíveis, e sua inserção é de forma transparente, possui hifenização superior a outros processadores de textos e melhor espaçamento entre as palavras [12]. Outra característica é que o Latex oferece suporte a fórmulas matemáticas de alta qualidade, onde o usuário deve apenas decorar alguns comandos facilmente comprehensíveis que especificam a estrutura lógica de um documento [16], outros pacotes tipográficos que suportam gráficos, podem ser facilmente incorporados flexibilizando a aplicação, que possui o código aberto

(disponibilidade), o que faz com que apareçam inúmeras implementações como um exemplo, pode citar o Miktex.

O Latex é portável, pois pode ser executado em qualquer máquina, sobre qualquer sistema operacional, onde o resultado de um projeto feito em Latex em uma máquina, tem resultado idêntico em outra, fato que não é suportado por outros processadores de texto como o Word, por exemplo. A implementação do Latex é estável, a possibilidade de encontrar um bug sério no Latex é bastante remota [12], o formato de seus documentos são em ASCII, o que permite que eles possam ser criados e manipulados por editores de textos convencionais, além de seus arquivos resultantes serem relativamente menores em relação a muitos sistemas WYSIWIG, além de demandar uma menor necessidade de hardware para realizar suas funcionalidades. Uma desvantagem visível em relação a uma ferramenta WYSIWIG é na construção de tabelas, que são mais difíceis de serem feitas com o Latex e para usá-lo é necessário o conhecimento de um editor de texto.

Para gerar um documento em Latex, de início escreve-se um texto com comandos Latex e um conteúdo de interesse em um processador de texto de sua escolha como nas linhas seguintes:

```
\documentclass [a4paper, 12pt] {article}
\usepackage[brazil]{babel}
\usepackage[latin1]{inputenc}
\begin{document}
\section{Texto, Comandos e Ambientes}
\section*{Introdução}
\end{document}
```

Todo documento deve começar com o comando `\documentclass [opções]{classe}` (figura 3), onde em opções temos configurações de tamanho de letra, tipo de papel, número de colunas, etc. Em classe temos o tipo de documento, que pode ser livro, artigo, etc. Logo após temos o comando `\usepackage` (figura 3), usado para adicionar pacotes no texto. O conteúdo do documento é colocado entre os comandos `\begin{document}` (figura 3) e `\end{document}` (figura 3) e a partir daí o documento é estruturado em seções, no caso de artigo.

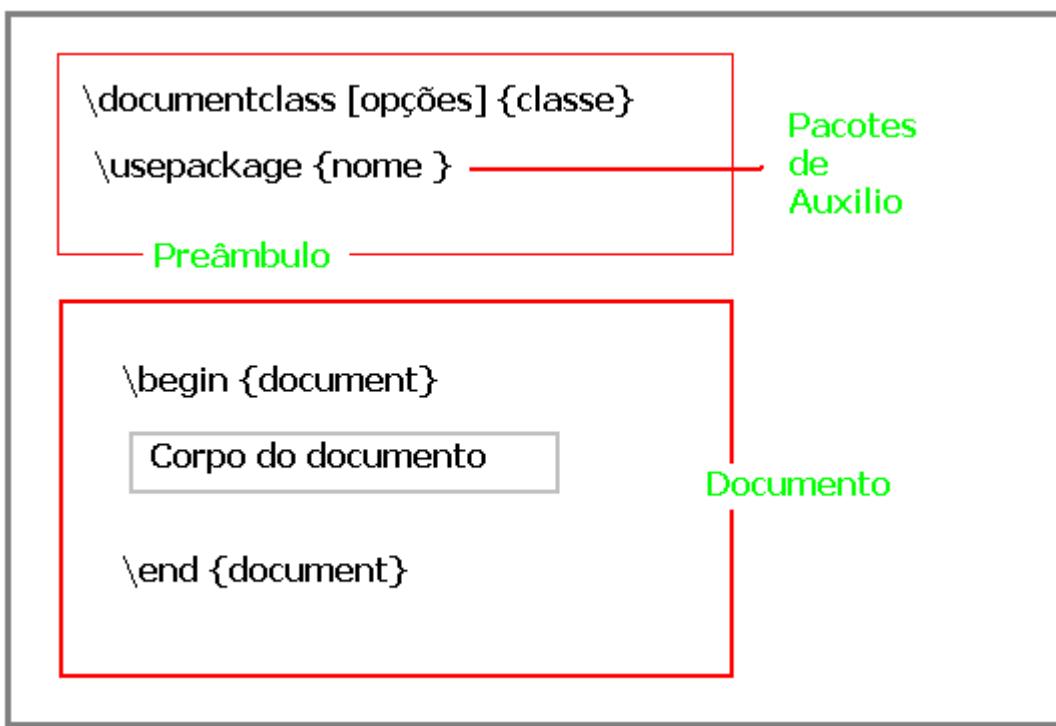


Figura 3. Esquema de apresentação de um arquivo Latex

O Latex não respeita quebras de linha e interpreta qualquer bloco de texto como um parágrafo. Parágrafos são separados pelo salto de uma linha em branco.

Ao finalizar o documento, este é salvo como um arquivo *.tex. Logo após pode ser compilado em uma linguagem pelo tex, o que resulta em um arquivo *.dvi (*device independent*) ou *.pdf, que já é um arquivo pronto para ser visualizado.

3.3 Texto, comandos e Ambientes

Em textos Latex as palavras são delimitadas por um ou mais espaços, já os parágrafos são separados por uma ou mais linhas em branco. A maioria dos comandos é iniciada com o caractere “\”. Uma “\” sozinha produz um espaço. Um ambiente é uma região do texto que tem um tratamento especial. Um ambiente é iniciado com `\begin{ }{ }` e finalizado com `\end{ }{ }`. É importante ter cuidado para não digitar nenhum dos dez caracteres especiais (apresentam um significado reservado para o Latex):

& \$ # % _ { } ^ ~ \

Eles podem ser adicionados no texto através dos seguintes comandos:

`\&` `\$` `\#` `\%` `_` `\{` `\}` `\^{}{ }` `\~{}{ }` `\backslash`

Os comandos Latex são case-sensitive, ou seja, existe distinção entre maiúsculas e minúsculas e apresentam-se de duas formas, onde uma se inicia com a barra invertida “/” seguido de um conjunto de letras e finaliza-se com um espaço em branco ou qualquer símbolo que não seja uma letra e o outro formato é caracterizado por uma barra invertida seguida por um caractere que não seja letra. Espaços em branco após os comandos são ignorados. Os comentários são representados pelo caractere “%”, onde cada elemento até o fim da linha é ignorada, outra forma é usando o ambiente `comment`.

Um arquivo é formado de duas partes: o preâmbulo e corpo. No preâmbulo temos todos os comandos que especificam os parâmetros globais do texto, ou seja, toda regra de formatação como tipo do papel, largura e altura do texto entre outros, ele deve conter pelo menos o comando `\documentclass [opções] {classe}` onde em opções temos configurações de tamanho de letra, tipo de papel, número de colunas, e especifica o tipo de documento que pode ser um report que é a classe de relatórios, book para a de livros, entre vários outras como `article` para artigos que é a classe utilizada na proposta de sistema desse trabalho e termina com o comando `\begin{ }` , onde se inicia o corpo. Os comandos que não forem colocados no preâmbulo são atribuídos pelo Latex com valores standard. Para algumas funcionalidades o pacote básico do Latex não será suficiente, então é necessário utilizar pacotes, alguns exemplos são mostrados na tabela em seguida. No corpo fica todo o conteúdo do documento além de comandos locais. O corpo termina como o comando `\end{ }`. A seguir temos alguns dos comandos globais mais utilizados :

```
\usepackage[brazil]{babel}
\usepackage{graphicx, color}
\usepackage[latin1]{inputenc}
```

O primeiro indica para usar o pacote `babel`, com a língua do Brasil, faz com que ele gere datas e nomes no estilo brasileiro, no segundo temos um exemplo do uso os pacotes `graphicx` e `color`, que permitem incluir figuras e colorir o texto e na terceira linha o uso do pacote `inputenc` com a opção `latin1`, que define uma codificação para os caracteres em que os acentos são digitados.

```
\setlength{\textwidth}{16 cm}
\evensidemargin 0 cm
```

O comando `\setlength`, diz que a largura do texto deve ser de 16 cm, enquanto que o outro comando, indica que a margem esquerda das páginas pares é zero. O tamanho real da margem esquerda das páginas pares é a soma da variável `\hoffset` com a variável `\evensidemargin`. Para margem direita o comando é `\oddsidemargin`.

Outros comandos, como o `\topmargin 0 cm`, que indica que a margem superior zero. O tamanho real da margem superior é a soma da variável `\voffset` com a variável `\topmargin` e o `\sloppy` reduz o número de divisões nas palavras que são impressas nos finais das linhas.

Além destes, um comando que pode ser útil é o `\newcommand`. Com ele você pode definir nomes mais simples para comandos que têm nomes grandes.

Por exemplo:

```
\newcommand{\binv}{\$\\backslash\$}
\newcommand{\pot}{\\^{}{}}
\newcommand{\til}{\\~{}{}}
```

Onde o primeiro `\newcommand`, dá um novo nome ao comando que imprime “\”, o segundo dá para o que imprime “^” e o terceiro para o que imprime “~”.

No que se diz a respeito de paginação são suportados três combinações pré-definidas de cabeçalho / rodapé denominados estilos de paginas. Para a exibição de texto pode se mudar o tipo e o tamanho da letra além de outros :

A seguir temos mudança de estilo, texto, tamanho de letra:

```
\pagestyle{estilo}; : para mudança de estilo
\textit{itálico} itálico; : para mudança de texto para itálico
\textbf{negrito} negrito; : texto em negrito
\textrm{romano} romano; : texto em romano
{\tiny o menor} o menor;

{\scriptsize muito pequeno} muito pequeno;
{\footnotesize menor} menor;
{\small pequeno} pequeno;
{\large grande} grande;
{\huge ainda maior} ainda maior;
{\Huge o maior de todos} o maior de todos.
```

Para construção de listas temos os seguintes ambientes :

- Ambiente itemize (tabela 2)

Tabela 2. Ambiente itemize

Comando	Resultado
<code>\begin{itemize}</code> <code>\item itens \\$\bullet\\$;</code> <code>\item item adicional.</code> <code>\end{itemize}</code>	Itens são precedidos por •; Itens são separados por um espaço adicional

- Ambiente Enumerate (tabela 3)

Tabela 3. Ambiente Enumerate

Comando	Resultado
<pre>\begin{enumerate} \item Os itens são numerados com algarismos arábicos, no primeiro nível, \begin{enumerate} \item são numerados com letras no segundo nível e \begin{enumerate} \item são numerados com algarismos romanos no terceiro nível. \end{enumerate} \end{enumerate} \end{enumerate}</pre>	<p>1. Os itens são numerados com algarismos arábicos, no primeiro nível,</p> <p>(a) são numerados com letras no segundo nível e</p> <p>i. são numerados com algarismos romanos no terceiro nível.</p>

Pode-se uma variedade de outras construções como teorema, preposições, fórmula matemáticas com expoentes, índices, frações, raízes, somatórios, integrais, funções, matrizes, equações e símbolos (figura 4) como mostrados adiante :

Operadores Binários					
Símbolo	Comando	Símbolo	Comando	Símbolo	Comando
\pm	\pm	\mp	\mp	\times	\times
\div	\div	\cdot	\cdot	$*$	\ast
\star	\star	\dagger	\dagger	\ddagger	\ddagger
\cap	\cap	\cup	\cup	\setminus	\setminus
\vee	\vee	\wedge	\wedge	\otimes	\otimes
\triangleup	\bigtriangleup	\triangledown	\bigtriangledown	\oplus	\oplus
\triangleleft	\triangleleft	\triangleright	\triangleright	\odot	\odot
\circ	\circ	\bigcirc	\bigcirc	\diamond	\diamond

Relações					
Símbolo	Comando	Símbolo	Comando	Símbolo	Comando
\leq	\le	\geq	\ge	\sim	\sim
$\not<$	\not<	$\not>$	\not>	\neq	\neq
\subset	\subset	\supset	\supset	\approx	\approx
\subseteq	\subseteq	\supseteq	\supseteq	\simeq	\simeq
\in	\in	\notin	\notin	\equiv	\equiv
\perp	\perp	\propto	\propto	\cong	\cong

Outros Símbolos					
Símbolo	Comando	Símbolo	Comando	Símbolo	Comando
\forall	\forall	\exists	\exists	∞	\infty
∇	\nabla	∂	\partial	\emptyset	\emptyset
\Re	\Re	\Im	\Im	\neg	\neg

Figura 4. Símbolos Latex

Ainda possui diversos outros recursos como referência cruzada, espaçamento horizontal, vertical, tabelas, etc. Dessa variedade, tem-se a importância dessa ferramenta e aceitação dela como uma boa ferramenta para formatação e construção da maior parte dos textos acadêmicos e científicos.

3.4 Distribuições Latex

A seguir tem-se algumas das principais distribuições usadas pela comunidade, que podem facilmente confeccionar um documento com Latex.

- **LEd-LaTeXEditor** **0.50** em <http://baixaki.ig.com.br/download/LEd-LaTeXEditor.htm>
 - LEd ou LaTeXEditor é um ambiente livre para processamento rápido de documentos TeX/LaTeX. Ele oferece recursos de gerenciamento de projetos, edição, checador de ortografia e enciclopédia; visualizador DVI integrado,

dicas descritivas para comandos LaTeX e mecanismos de complementação de códigos.

- **Miktex** em <http://Miktex.org/>
 - Atualização da implementação Tex para Windows. Contém um conjunto completo de pacotes e fontes (ConTeXt, Latex,...)
- **LaTeX Plugin For Eclipse** em <http://sourceforge.net/projects/eclipse-latex/>
 - Gerenciamento de documentos LaTeX com Eclipse. Fácil compilação de documentos LaTeX, oferece acesso a todas as informações do compilador LaTeX. Existem muitas outras implementações de plugins para eclipse Latex.
- **Latex Mng 2.0** em <http://newfreedownloads.com/download-LatexMng.html>
 - Aplicação que integra todas as ferramentas necessarias para criação de documentos Latex. Fácil de usar e integrado para o ambiente Windows.
- **Latex-Suite** <http://vim-latex.sourceforge.net/index.php?subject=download&title=Download>
 - Contém um conjunto de elementos Latex e utilize modo gráfico.

Capítulo 4

Desenvolvimento

Neste capítulo são mostrados as tecnologias utilizadas na construção do software proposto, seus principais requisitos, a arquitetura adotada e uma descrição da implementação.

4.1 Requisitos

Para desenvolver uma ferramenta colaborativa de artigos científicos utilizando Latex é necessário analisar às características das ferramentas groupware, como foi mostrado no capítulo dois, e associar essas abordagem às funcionalidades de construção e estruturação de textos oferecidos pelas ferramentas Latex já existentes. No capítulo três mostrou-se as funcionalidades do Latex e suas diversas vantagens na elaboração de texto, assim como algumas das distribuições mais usadas pela comunidade científica. Foi dito que a implementação Latex é gratuita, a maioria dos sistemas apenas não apresenta uma interface gráfica amigável, para utilização de suas bibliotecas ricas em funcionalidades e para construção de textos.

Com base nesses fatos, não se faz necessário uma reimplementação das funcionalidades Latex, já que existem diversas implementações, mas com interfaces que muitas vezes fazem uso do prompt de comando (caso do Windows, Shell para o Linux) para usá-las. Além desses argumentos, para o desenvolvimento de um ambiente colaborativo utilizando Latex é necessário apenas seguir suas regras de estruturação, usadas para a construção de um texto.

No capítulo três foi mostrado o processo de criação de um projeto com o Latex, no qual o usuário cria um arquivo *.tex (que contém todos os comandos para formatação do texto) e após a compilação desses, é formado um arquivo *.dvi, onde já pode ser visualizado ou transformado para qualquer outro formato desejado pelo usuário. A fase colaborativa na geração de um texto, no caso em questão, é a construção do arquivo inicial *.tex e para construção desse arquivo, foi visto que é apenas necessário criá-lo com um editor comum e salvá-lo com a extensão para depois compilá-lo, onde se analisará os erros e em seguida visualizá-lo. Com isso, o texto

compartilhado para o ambiente colaborativo é a entrada inicial *.tex e podem-se utilizar compiladores Latex já existentes para compilar essa entrada e criar a saída em formato pdf, que será usada pelo WebLatex para mostrar ao usuário o resultado.

Para construção do arquivo de entrada, que contém os comandos Latex, faz-se necessário o desenvolvimento de um editor multiusuário, no que diz respeito à funcionalidade, pois o arquivo de entrada será dividido em pequenos blocos, com um controle de concorrência, a fim de garantir o desenvolvimento colaborativo dos membros. O WebLatex é uma ferramenta de Interação Síncrona Distribuída, pois cada modificação, por parte daqueles que compõem o grupo colaborativo, será facilmente identificada pelos demais em tempo real, e distribuído, porque seus usuários estão em lugares remotos, o que justifica a decisão do sistema estar disponível na internet.

Mas é preciso garantir as características de groupware, apresentada no capítulo dois, para fazer do WebLatex uma ferramenta colaborativa. Para garantir o compartilhamento, definiu-se um bloco mínimo de compartilhamento, que para o WebLatex são as seções, pois é a menor unidade para edição de um único usuário, sem que comprometa o contexto e a concisão do artigo. As seções têm um controle de concorrência de acesso, garantindo a coordenação. Para comunicação, o sistema não tem implementado qualquer funcionalidade de comunicação em tempo real, mas ele pode ser usado em conjunto com outras ferramentas colaborativas do gênero, que são bastante comuns, como chats, videoconferências, sistemas de voip e até mesmo telefone. A comunicação pelo programa se dará de forma indireta, com os usuários percebendo a ação dos outros e seguindo suas atividades. Todo usuário ao ter acesso ao sistema pela internet, tem uma grande probabilidade de estar numa estação, que contenha pelo menos uma dessas ferramentas de comunicação citadas.

Mas em contrapartida, o WebLatex tem um bom sistema de percepção associado à funcionalidade de designação de papéis. O usuário ao criar um projeto automaticamente já recebe a responsabilidade de ser o administrador do artigo, com isso cabe a ele escolher os colaboradores de seu projeto, além de definir o preâmbulo do texto e atribuir as atividades. Essa distribuição constitui a designação de papéis, onde cada membro tem atividades proposta pelo criador para realizar, sem que estas limitem sua criatividade, como fosse visto no capítulo dois. A característica de percepção está implementada nas seções, pois cada membro pode ver em tempo real o trabalho do colega. Quanto à visualização o sistema segue a mesma estrutura de visualização do Latex, e a construção do texto de entrada é síncrona.

Para visualização do artigo, o sistema faz todo processo para os usuários, onde eles só têm de saber alguns comandos básicos de Latex para criar seus textos científicos, ou seja, a geração completa para visualizar e obter o arquivo final pronto é abstraído pelo sistema.

Por questão de usabilidade, tomou-se a decisão de abstrair o processo de compilação e geração do arquivo final em pdf para o usuário, com isso a pessoa clica num botão e terá o resultado, que pode ser uma pré-visualização do projeto ou um *log* de erros caso o texto contenha erros.

A proposta do software que realiza o trabalho colaborativo implementa os seguintes casos de uso, os quais seguem os conceitos apresentados no capítulo dois para criação de uma ferramenta groupware respeitando a estrutura de criação de texto Latex:

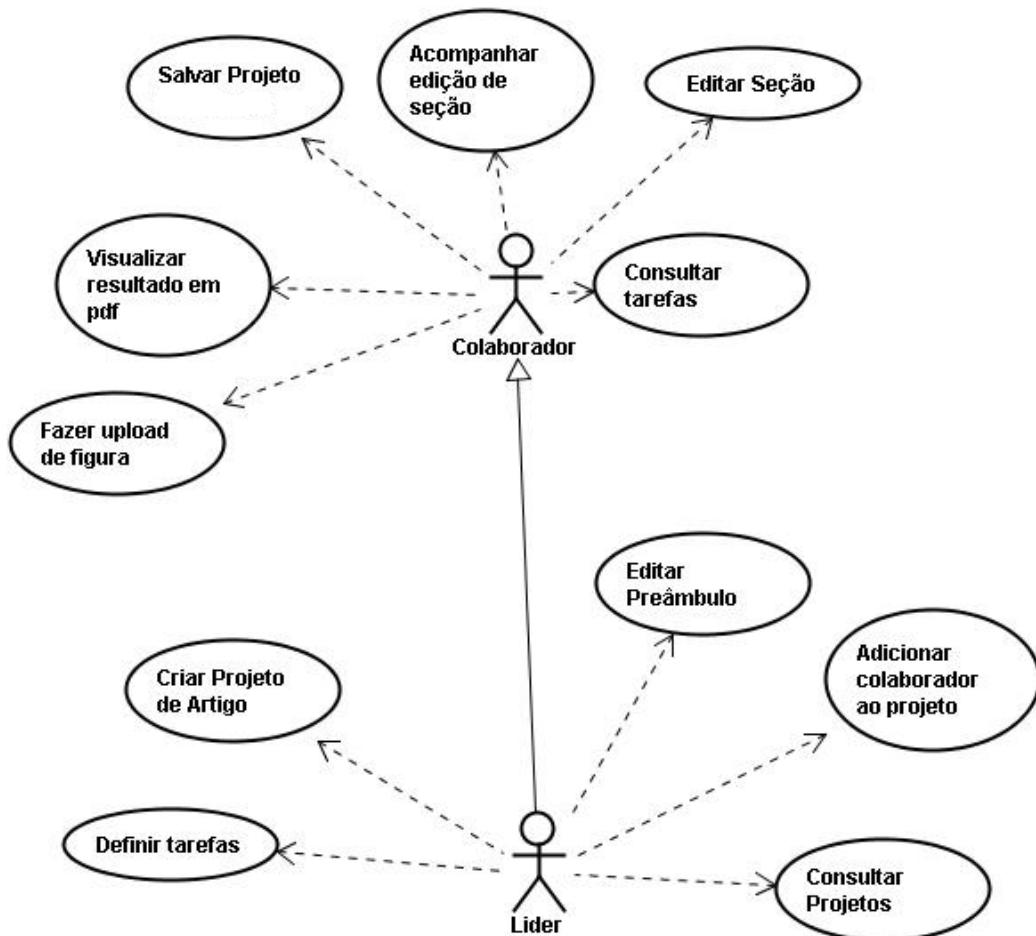


Figura 5. Diagrama de Casos de Uso do WebLatex

Os casos de uso (figura 5) representam as funcionalidades do sistema. Como é um projeto colaborativo de artigos científicos utilizando Latex, todos os comandos Latex são aceitos dentro do conteúdo das seções, sendo que alguns não necessitam ser digitados, pois a própria ferramenta já faz a construção desses comandos automaticamente, por questão de usabilidade.

Dentre os comandos que não são necessários para o desenvolvimento do artigo estão:

- Comandos de compilação Latex.
- Os comandos de visualização, sendo apenas mostrado o resultado em formato pdf, já que é um formato suportado pela maioria dos sistemas operacionais.
- O Latex pode criar diversas classes de textos, como foi visto no capítulo 3, mas o sistema abstrai o comando das classes, sendo apenas possível criar a classe artigos na ferramenta, que é a proposta do trabalho.

Os principais casos de uso que merecem destaque são: A edição de seções , o visualizador do artigo e visualização das tarefas, por apresentar uma complexidade maior em relação aos demais.

Na edição de seção, o usuário captura a permissão de edição da seção para realizar suas devidas modificações, enquanto isso, a seção em destaque estará bloqueada para os demais colaboradores e estes só terão acesso de leitura à seção, podendo ver o trabalho do colega em tempo real. Para realização dessa tarefa, foram aplicadas técnicas mostradas nas ferramentas de groupware, ou seja, o controle de concorrência, que será mais bem explicado a seguir. O caso de uso funciona da seguinte maneira: O usuário escolhe a seção que queira editar, nesse momento o sistema captura a requisição do usuário e bloqueia a seção para todos os outros, de forma que o usuário recebe a resposta como uma nova tela, onde poderá editar o conteúdo. À medida que o usuário insere texto, este é salvo pelo sistema automaticamente, ao mesmo tempo em que atualiza seu progresso para os demais colaboradores.

Na visualização do artigo o colaborador requisita visualização do projeto em pdf, em um determinado momento, ou seja, todas as seções serão agrupadas e o sistema irá gerar o artigo utilizando o compilador Latex, onde em seguida mostra na tela um link para o documento em pdf, cujo conteúdo será o artigo como ele deve ser. Importante ressaltar, que essa opção só pode ser feita quando o projeto já estiver sido compilado e não houver erros por parte do Latex no momento, pois se houver erros é mostrado o link para o arquivo *log* de erros, a fim de que o usuário faça a devida modificação para que o artigo seja mostrado de forma correta.

Após se criar um projeto, o administrador deve criar as tarefas para os colaboradores de seu projeto, a fim de se manter formalmente o papel deles no trabalho de colaboração, fortalecendo a organização do trabalho. Cada colaborador ao se autenticar no sistema terá uma funcionalidade que permite ver quais tarefas foram atribuídas para ele, o que lhe dará um rumo concreto nas suas atividades, como vimos anteriormente a gerencia em trabalhos colaborativos é uma tarefa complexa e se faz necessário a atribuição de papéis. Cada atividade terá sua descrição e sua data de conclusão para que o administrador saiba realmente como está o andamento do seu projeto e possa cobrar aos seus colaboradores os resultados.

Para a confecção do WebLatex, adotou-se como ciclo de vida de desenvolvimento de software, o modelo Iterativo, pois é útil para pequenos projetos, e um ciclo ou iteração é desenvolvida dentro de um tempo estimado[26].

No desenvolvimento se usou três iterações:

- Iteração 1: onde se determina os objetivos e restrições do sistema;
- Iteração 2: analisam-se os riscos das decisões a respeito dos requisitos que serão escolhidos na iteração, para alguns requisitos considerados mais complexos determinam-se soluções alternativas;
- Iteração 3: realizava-se a implementação (codificação) dos requisitos de maior importância para o sucesso do projeto;

4.2 Tecnologias

Para determinar o controle de concorrência do software proposto, é preciso determinar a granularidade, pois quanto menor for o nível de granularidade que o sistema suporta, mais complexo é o controle, no entanto maior será flexibilidade do usuário. A política de controle de um nível muito baixo exige uma complexidade de gerenciamento enorme, pois à medida que aumenta o número de colaboradores, o sistema tende a demandar um maior processamento.

Como já foi mencionado, como nível de granularidade de bloqueio do sistema foi escolhido às seções, ou seja, um colaborador terá acesso apenas ao corpo de uma seção por vez, sendo que os outros nesse momento terão apenas permissões de leitura do trabalho do colega. Para ser mais específico, ao criar uma seção e editar o seu corpo, essa seção estará bloqueada para os demais usuários. Esse nível de granularidade foi escolhido para que haja uma maior flexibilidade de interação na produção do artigo, ou seja, os colaboradores poderão criar pequenos blocos de texto, concomitantemente com outros.

Após definida a granularidade para controle de concorrência, escolhe-se a linguagem na qual o sistema deve ser implementado, para garantir as características de um sistema colaborativo, como já se foi visto. A linguagem escolhida foi *Java* [20], por apresentar vantagens como multiplataforma, o que flexibiliza a implantação do servidor, independente do sistema operacional, além de todas as outras vantagens que vem com a linguagem *Java* como: encapsulamento, tipagem forte entre outros. Para implementação do suporte web, que o sistema deve oferecer, a fim de garantir a característica de estar disponível em qualquer lugar a qualquer momento, foi utilizado um conjunto de frameworks de desenvolvimento para aplicações web em *Java*, conhecido como *Appfuse*, juntamente com o container *Apache Tomcat 5.0*[21] e o banco de dados *Mysql* [23].

Os *Frameworks* são conjuntos de softwares que provêm soluções para uma família de problemas semelhantes [10], tendo uma implementação quase completa de um determinado problema, mas com específicos pedaços faltando, pois caracterizam a particularidade da resolução. Um framework difere de uma biblioteca por algumas características, onde numa biblioteca cada classe é independente e única, enquanto que nele as dependências e colaborações estão embutidas (*wired-in interconnections*), em bibliotecas as aplicações criam colaborações, enquanto que o framework impõe um modelo de colaboração e por fim tem-se a inversão de dependência. Onde o código do framework é criado e mantido acessível pelo próprio, ou seja, ele realiza a chamada para esse código e após sua utilização o fluxo retorna para ele, diferente de uma biblioteca que após utilizar um método o controle volta para o usuário.

O *Appfuse* faz uso de um conjunto de *frameworks* utilizando o *J2EE* de *Java*, sendo que nem todos os frameworks do *appfuse* são utilizados nesse projeto. Dentre os frameworks utilizados os mais importantes são:

- *Ant* [22] – Ferramenta de automação de *builds*, ou seja, atualiza o *classpath*, compila o código nos específicos diretórios, podendo até gerar o javadoc do projeto, além de configurar e executar a aplicação [22].

- *Hibernate* [31] – Ferramenta que faz o mapeamento objeto/relacional no ambiente *Java*, ou seja, mapeia uma representação de dados de um modelo orientado a objetos para um modelo de dados relacional, onde o esquema é baseado no *SQL*, além de facilitar as consultas e recuperações de dados o que impacta na redução do tempo gasto de desenvolvimento.

- *Spring* [30] – Framework que utiliza o conceito de injeção de dependências (*Dependency Injection – DI*), o qual é parte do conceito de Inversão de controle (IoC), onde um componente não instancia as suas dependências, mas o ambiente (*container*) automaticamente as fornece [30], ou seja, o container se encarrega de toda parte de instanciar um objeto e injetar essa instância sendo que as dependências são especificadas em um arquivo de configuração.

- *JSF* [32] – Framework de componentes de interface gráfica de usuário (UI), criado para facilitar à escrita e a manutenção de aplicações web *java2EE*.

- *Acegi* [33] – Sistema de segurança desenvolvido para o controle de métodos de cada objeto, ou para todos os objetos de uma classe. Faz uso de autenticação e a instancia é baseada no acesso e canais de segurança.

- *Ajax4JSF* [17] – adiciona elementos *Ajax* para as páginas *Jsf*, sem muita mudança em seus componentes.

Todo o sistema colaborativo foi implementado sobre da plataforma do *Appfuse*, que contém os frameworks e segue o padrão MVC, sendo que, para a utilização dos componentes Latex foi usado o pacote do Miktex, que faz uso de seus recursos, para compilar e gerar os textos de alta qualidade para composição final do artigo. Sendo assim, a máquina que deve hospedar o sistema deve ter o Miktex instalado e o administrador deve configurar no sistema o path dos seus executáveis.

4.3 Arquitetura

Existem inúmeras arquiteturas que são usadas em sistemas colaborativos, muitos deles se tornaram referência, com a popularização dos sistemas groupware e algumas foram estendidas das já conhecidas arquiteturas não-colaborativas, como o modelo arquitetural PAC (*Presentation, Abstraction, Controller*), que divide hierarquicamente um sistema em agentes, dos quais cada um possui três facetas:

- A faceta de apresentação, ou seja, a interface do usuário;
- A de abstração, que cuida dos dados do sistema e;
- A de controle responsável pela comunicação das duas facetas e de todos agentes presentes

Essa arquitetura foi estendida para o PAC* de acordo com um estudo posposto por Dewan [10], que demonstrou uma estrutura para sistemas de groupware dividido números de camadas variáveis, que compreendem desde o nível de domínio do problema até o nível de hardware, onde cada camada é um componente que corresponde a um nível de abstração [10]. No entanto o PAC* possui a particularidade de permitir divisões e junções distintas diferente da Arquitetura Colaborativa genérica de Dewan [10], muito comum na literatura. Com isso seus controladores

podem, além de se comunicar com outros agentes do sistema, com outros sistemas colaborativos PAC*. Na figura 6 temos os agentes e como eles se distribuem no modelo PAC*.

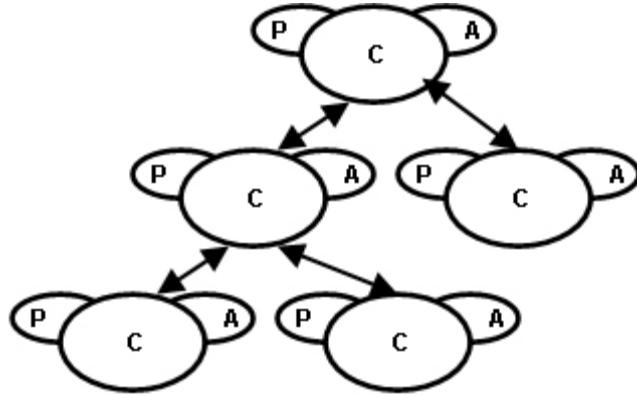


Figura 6. Modelo PAC*

Existem outras arquiteturas que seguem construções semelhantes, mas a estrutura utilizada para este projeto foi uma que originalmente é mais utilizada em sistemas não-colaborativos: A arquitetura MVC. No entanto é a estrutura mais utilizada em sistemas desenvolvidos para internet.

O padrão de projeto MVC (*Model – View – Controller*) utiliza camadas, onde fica distintos a lógica da aplicação (*Model*), que representa a estrutura baixo nível, sendo nesse projeto a caracterização do modelo objeto-relacional, ou seja, a camada de dados. O fluxo da aplicação (*Controller*), que tem o papel de administrar os eventos da aplicação, que são as repostas ao cliques dos usuários através de requisições a camada model, ou seja, controla e mapeia as seções e a interface (*View*), que representa a apresentação e visualização dos dados e eventos requisitados pelo usuário ao controller. Uma das vantagens em se usar esse padrão é a fácil manutenção e atualização da aplicação que o implementa, o que será de grande ajuda na implementação dos trabalhos futuros, que podem ser integrados a essa aplicação, além de tornar a aplicação escalável.

Já que possui uma organização bem robusta para esse propósito, e como já foi dito na explicação de padrão de projeto MVC, possui um baixo acoplamento entre as camadas de visão, controle e modelo, o que garante várias implementações na plataforma. Com pequenas modificações nos componentes organizados de acordo com o estilo arquitetural MVC, pode se desenvolver um software, para gerar aplicações colaborativas derivadas de aplicações não colaborativas [10]. Assim como é feito no WebLatex. Parte-se de uma aplicação não colaborativa, que é o pacote Latex de desenvolvimento de texto, e desenvolve-se uma estrutura com os conceitos colaborativos, utilizando a vantagens do Latex. A seguir temos o diagrama de classes que compõe as classes básicas (figura 7) principais que compõem o WebLatex.

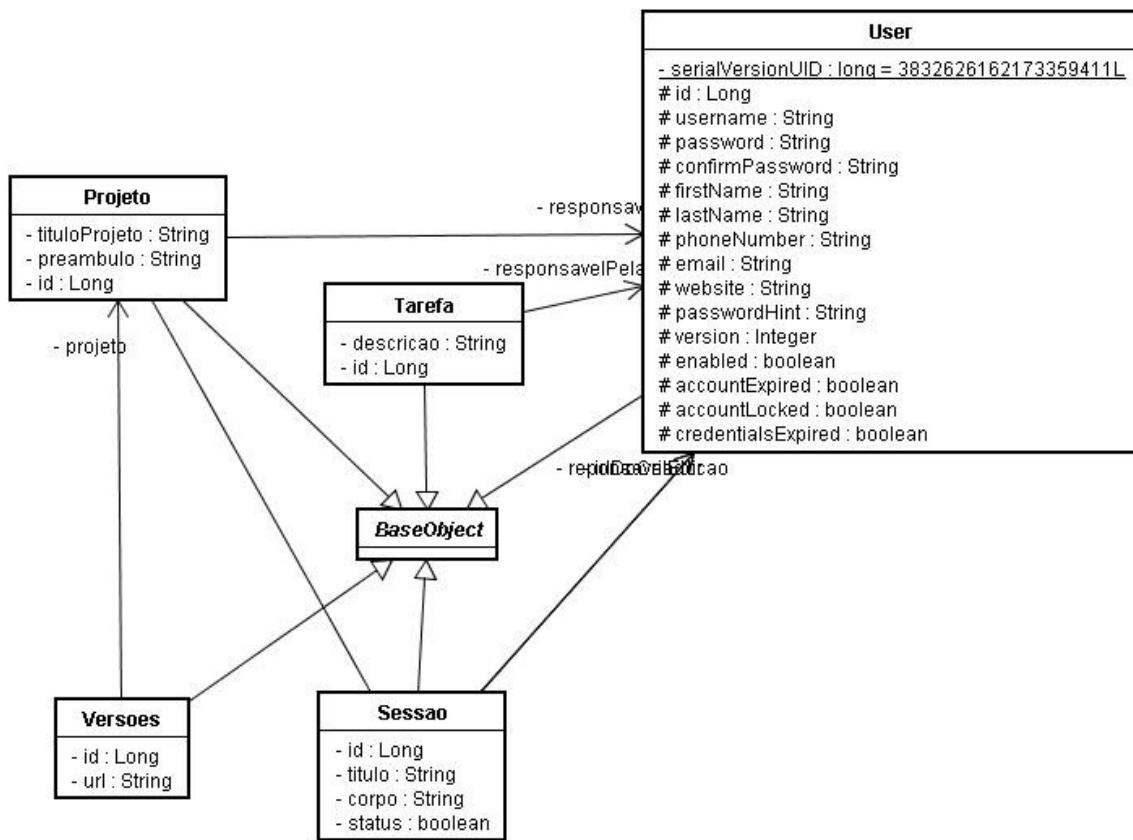


Figura 7. Diagrama de classes simplificado da camada *model* do WebLatex

Capítulo 5

O Sistema

Como resultado deste trabalho é desenvolvido a ferramenta WebLatex – Sistema colaborativo para geração de artigos Científicos. Nas seções a seguir são apresentadas as principais características do WebLatex, tais como desenvolvimento de um projeto, escolha de colaboradores e desenvolvimento de um artigo na ferramenta.

5.1 O WebLatex – A Ferramenta de execução

WebLatex é um sistema groupware para desenvolvimento de artigos científicos, que permite a construção de um artigo científico de maneira colaborativa na internet utilizando o Latex e apresenta as seguintes características:

- Ambiente multiplataforma: por ter sido desenvolvido na linguagem de programação *Java* [*Sun Microsystem*], o funcionamento do WebLatex torna-se praticamente independente do sistema operacional utilizado. O sistema pode ser operacionalizado em qualquer servidor, sobre os sistemas operacionais Linux, Windows 9x, Windows XP, mas é necessário ter uma versão do Miktex (para Windows) ou semelhante instalado para esses sistemas operacionais, e alguns parâmetros devem ser atualizados conforme o apêndice B. Para utilização, o servidor deve ter a máquina virtual Java (*JVM – Java Virtual Machine*) 5.0 ou superior, O container *Apache Tomcat* 5.0 ou versão superior, Banda Larga de no mínimo 512 kbps para uma boa utilização.
- Ambiente bilíngüe: Visando alcançar um público grande, o WebLatex foi desenvolvido em dois idiomas. O idioma português que funciona em modo nativo, e o idioma inglês que funciona como opcional, na tela Principal o usuário pode escolher o idioma preferido além que o sistema identifica automaticamente o browser que estiver sobre esses dois idiomas e se configura automaticamente;
- Código aberto: o WebLatex é disponibilizado sob a licença de uso GPL (*GNU General Public Licence*) ou licença pública geral GNU [24]. De acordo com esta licença, o

sistema pode ser modificado, ampliado e manuseado livremente desde que se mantenham as referências aos autores originais que formam todo projeto;

- Robustez: visando ter uma maior segurança, o sistema possui autenticação de forma que em cada projeto só usuários devidamente convidados poderão fazer parte. Qualquer usuário pode criar e começar um projeto, mas sós usuários convidados poderão fazer parte dos projetos criada por outros.

5.2 A Ferramenta de execução

Ao entrar com o endereço do WebLatex no browser o usuário é encaminha para tela de login do sistema (figura 8):

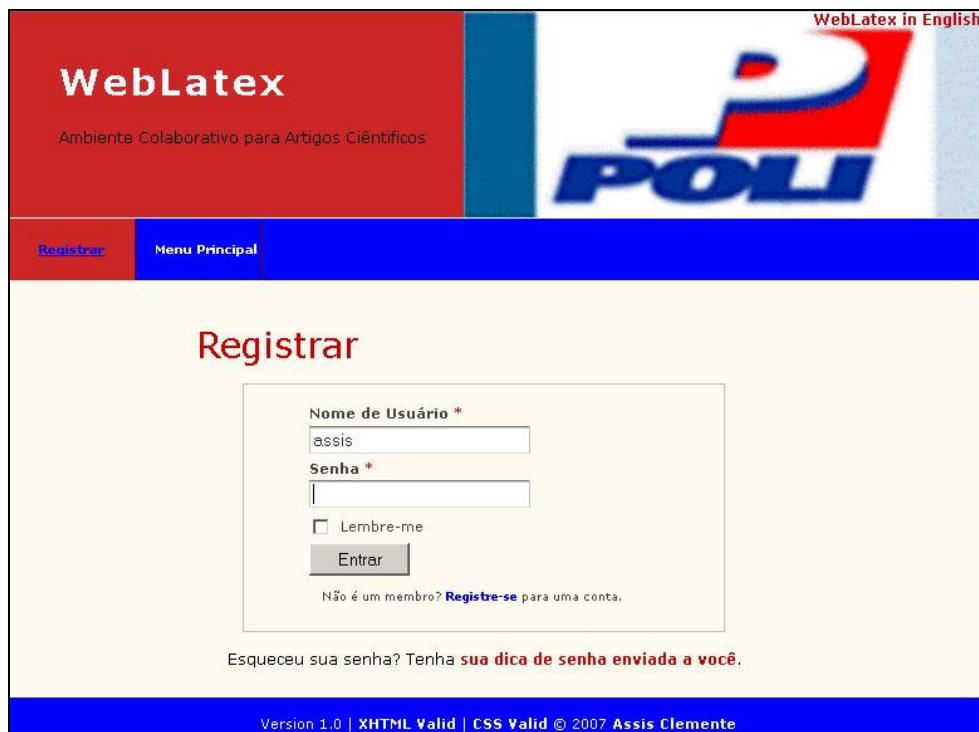


Figura 8. Tela Inicial do WebLatex

Após se registrar corretamente (caso já seja cadastrado no sistema), temos a área de gerenciamento do sistema, onde se pode criar projetos, visualizar os projetos atuantes, entre outros (figura 9).



Figura 9. Tela de Gerenciamento do WebLatex

O usuário poderá utilizar o sistema, no momento em que for convidado por email, por algum usuário já cadastrado ou simplesmente criando uma conta na tela de login. Ao ser convidado, o pré-colaborador receberá um link por email, que o link para o cadastro do WebLatex (figura 10), sendo que este deve se cadastrar com o mesmo email, com o qual foi convidado, pois é pelo email que o WebLatex reconhece o convidado, e o adiciona aos projetos aos quais ele recebeu o convite. Caso o novo membro se cadastre com um email diferente, do que foi convidado, ele não irá ser automaticamente adicionado ao projeto, sendo que o criador do projeto terá que adicioná-lo novamente para que esse tenha acesso ao projeto do artigo.

Como foi mencionado, ao criar um projeto esse usuário tem permissões especiais sobre o mesmo, cabendo a ele as tarefas de configurar o preâmbulo do projeto, que atualiza as características estruturais do artigo científico, convidar os colaboradores (figura 11), que irão compor os membros do projeto colaborativo e definir as tarefas de cada membro. Os membros poderão criar seções no projeto, assim como editá-las poderão compilá-las e visualizá-las, o criador também pode fazer essas funcionalidades.

Registrar **Menu Principal**

Registro de Novo Usuário

Por favor entre com sua informação no formulário abaixo.

Nome de Usuário *

Senha * **Confirmar Senha ***

Dica de Senha *

Primeiro Nome * **Ultimo Nome ***

E-Mail * **Numero Telefone**

Website *

Endereço

 Endereço

 Cidade * Estado *

 Código Postal * País *

Registre-se **Cancelar**

Version 1.0 | XHTML Valid | CSS Valid © 2007 Assis Clemente

Figura 10. Fragmento da Tela de Cadastro do WebLatex

WebLatex
 Ambiente Colaborativo para Artigos Científicos

WebLatex in English

Menu Principal **Editar Perfil** **Sair**

Convidar Colaboradores

Email *
 assis@dsc.upe.br

+

Email

tiago@dsc.upe.br
 jose@hotmail.com

Enviar **Cancelar**

Version 1.0 | XHTML Valid | CSS Valid | Registrado como: Tomcat User © 2007 Assis Clemente

Figura 11. Convite de Colaboradores para um Projeto do WebLatex.

A criação de tarefas é muito importante, pois dão aos membros uma direção sobre seus papéis no desenvolvimento do artigo científico, por exemplo: Ao se criar um artigo para esse trabalho, definir-se-ia o nome, a estrutura do artigo no preâmbulo, adicionaria João e José, por exemplo, para serem os colaboradores no projeto e definiria as tarefas deles como:

- João – responsável pela seção 1, que fala de produção colaborativa
- José – responsável pela seção 2, que fala da ferramenta Latex

Quando João se autenticasse no sistema, teria acesso no projeto às suas atividades. Também é possível acrescentar mais colaboradores e adicionar mais tarefas no meio do projeto, sendo cada um só poderá ver as tarefas designadas para ele. Ao realizar suas atividades, os colaboradores irão editar, criar e observar as seções que compõem o artigo científico. Ao editar uma seção, o usuário tem plenos poderes de acrescentar ou retirar, qualquer trecho que forma a seção (figura 12), pode-se colocar texto, comandos Latex e adicionar figuras. Durante esse momento os demais membros que participam do projeto poderão acompanhar o trabalho do colega em tempo real, mas não poderão editar aquela seção (figura 13), enquanto o usuário inicial não desbloqueá-la, ou seja, finalizar seu trabalho de edição.

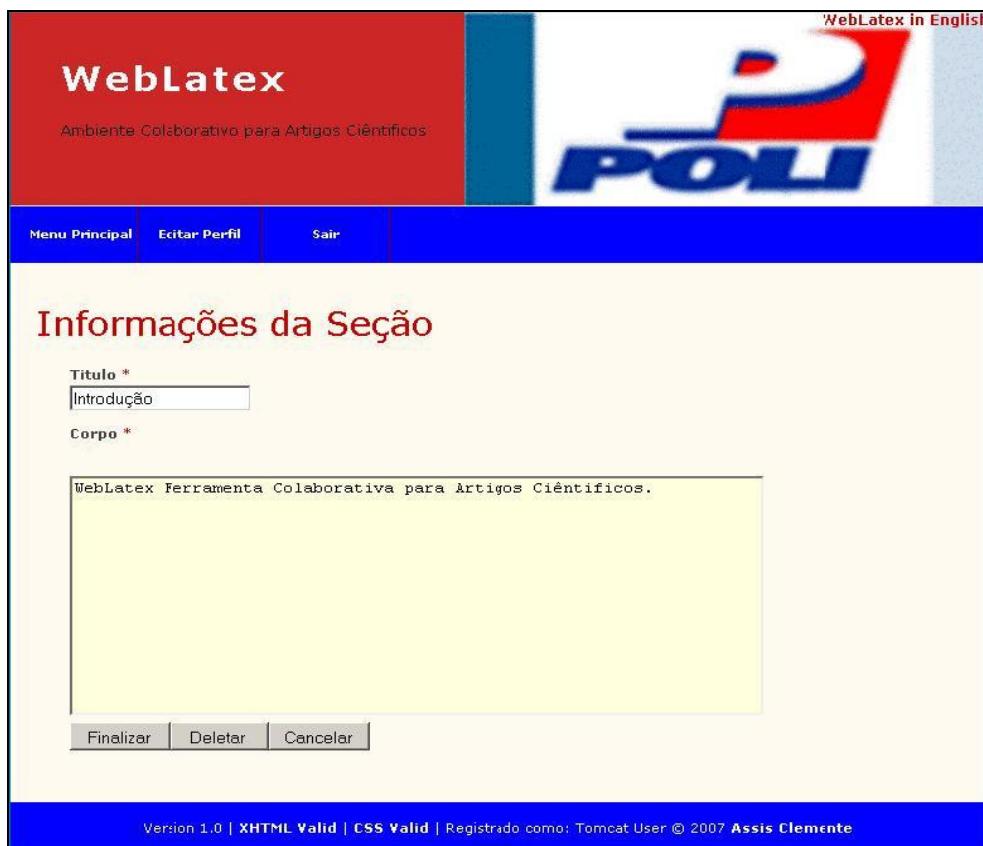


Figura 12. Tela de edição de seção



Figura 13. Visualização de uma seção que esta sendo editada por outro membro

É importante que cada colaborador compile o projeto após suas modificações, e caso verifiquem erros, faça as devidas alterações para que o projeto não fique com dependências para os demais membros. No momento de pré-visualizar o sistema indica qual seção contém o erro, então o administrador do projeto saberá quem está responsável por tal seção. Outro fator importante é que as tarefas são como indicadores, podendo os membros alterar qualquer seção, mesmo que não seja a que foi designado. Para ele é necessário bom senso e responsabilidade, para que os usuários não alterem seções, as quais não lhe cabem, deixando-as com erros ou apagando trechos sem permissão.

O sistema possui um pequeno controle de versão, na qual os membros ao terminarem suas tarefas, compilarem e visualizarem o projeto possa guardar essa versão de forma, que se algo acontecer no futuro o criador possa voltar a uma versão anterior, que esteja robusta. Uma versão só pode ser salva se não contiver erros.

Depois de finalizado o projeto, pode-se baixar a versão em pdf do mesmo já formatado e pronto para ser submetido ou impresso.

Capítulo 6

Considerações Finais

Neste capítulo, são apresentadas as principais conclusões deste trabalho de pesquisa, as contribuições para a sociedade e algumas propostas de trabalhos futuros.

6.1 Conclusões

O objetivo principal desse trabalho foi a elaboração de um ambiente colaborativo para a produção de conhecimento através de artigos científicos. Para desenvolvimento dessa ferramenta foi fundamental a utilização do padrão MVC, pois como foi visto deixou bem definido os dados da aplicação, o código de tratamento desses dados e a interface de visualização.

Para elaboração de uma ferramenta colaborativa foram estudados os conceitos de ambientes colaborativos de forma a selecionar a classificação e as funcionalidades que o WebLatex tem que seguir, além de como as atividades cooperativas devem ser organizadas para melhor suportar o trabalho colaborativo por computador. Foi identificado que o WebLatex é um sistema de groupware síncrono, pois no momento em que um membro realiza uma ação todos os outros devem perceber. Estudando mais detalhadamente esses conceitos chegou-se a conclusão que o sistema deveria estar classificado como um sistema de interação síncrona distribuída, pois deve ser utilizado em lugares remotos e em tempos distintos. Logo após foi classificado quanto à funcionalidade dentro do contexto do CSCW, e facilmente foi identificado como um editor multiusuário, porque a principal característica deste é permitir a edição colaborativa de um texto. O nosso objetivo também é a edição colaborativa de um artigo. Verificou-se no nosso caso que é necessária uma organização do grupo, devido a variáveis como: tamanho, comprometimento dos participantes; conclui-se que prazos e responsabilidades distintas ajudariam no processo de confecção do artigo, visto que o processo como um todo é uma tarefa convergente, ou seja, visa um objetivo comum. Então, logo em seguida, procurou-se catalogar as características das ferramentas colaborativas utilizando as premissas de classificações mencionadas, para estabelecer o conjunto de funcionalidades que o WebLatex poderia apresentar como percepção, comunicação, compartilhamento, designação de papéis, visualização e arquitetura.

O WebLatex tem grande parte dessas características selecionadas, resultantes do estudo de outras ferramentas colaborativas. Ele possui percepção, pois todos os membros vêm em tempo real a atividades dos demais, ele possui compartilhamento devido ao rigoroso controle de concorrência, que separa o artigo em seções editáveis unicamente por uma pessoa em um dado momento, possui visualização que se concluiu que pode ser dividida em duas partes no projeto: Para efeitos de edição colaborativa temos a visualização WYSIWIG, pois todos vêm aquilo que estão digitando no momento da edição, mas para visualização do artigo pronto, se usa o estilo de visual do Látex. Temos designação de papéis, pois foi anexado além dos papéis de sistema como criador do artigo e colaborador uma funcionalidade onde se podem designar tarefas aos membros. Quanto à comunicação o sistema não apresenta uma forma de comunicação direta, apenas indireta, pois foi avaliado que não prejudica o desempenho da confecção do artigo, mas pode ser anexado como algo extra em trabalhos futuros.

Como editor tipográfico foi anexado o latex devido à robustez e à qualidade dos seus resultados, pois ele apresenta características indispensáveis para um projeto open source, como o fato de ser um software livre e por sua simplicidade que foi facilmente acoplável ao WebLatex. Essa decisão faz com que quaisquer comandos latex possam ser usados na confecção do artigo.

Pode-se concluir também a eficiência da linguagem Java, e dos diversos frameworks que usados para o projeto como um conjunto bem definido de ferramentas próprias para o desenvolvimento web, com facilidades que aumentaram e muito a eficiência no desenvolvimento, a velocidade, padronização e aplicação dos conceitos colaborativos.

Além do alto grau de recursos oferecidos, a linguagem de programação Java é multiplataforma, permitindo que um mesmo programa possa ser utilizado em máquinas com sistemas operacionais diferentes, garantindo a portabilidade de qualquer aplicação implementada nessa linguagem. Além disso, o Java disponibiliza primitivas de concorrência. Estas foram relevâncias que nos levaram a escolher a linguagem Java entre outras.

Com todos esses recursos, conclui-se que com a aplicação pode-se facilmente desenvolver um artigo com um grupo de pessoas de maneira eficiente, inclusive com pessoas de países distintos já que a ferramenta é bilíngüe, além disso, ela é flexível devido ao MVC bem estruturado, o que deixa a aplicação robusta a manutenção e ampliações, pode ser ampliada para outros propósitos mais abrangentes como é mostrada nas propostas de trabalhos futuros. A prioridade principal foi criar essa ferramenta de acordo com os conceitos apresentados, outras ferramentas já demonstram sucesso na comunidade científica e esse projeto por ser open source e com código aberto mostra que é possível avançar nessa área de forma que todos participem e tenham acesso, tornando o dia-a-dia mais eficiente com as ferramentas de grupo.

Desta forma, podemos dizer que este trabalho apresentou as características de ferramentas groupware suas vantagens, conceitos e as definições de aprendizagem colaborativa. Foi mostrado o pacote de edição Latex suas vantagens, seu processo na criação de texto, assim como seu uso na elaboração da ferramenta. Também foi mostrada a ferramenta WebLatex: um software para edição colaborativa de artigos científicos utilizando o Latex, desenvolvido durante o trabalho,

assim como as ferramentas utilizadas para confeccioná-lo e uma breve descrição de como utilizá-lo.

Mas, contudo o principal objetivo do trabalho foi a elaboração de um ambiente colaborativo para produção de conhecimento através de artigos científicos e obtendo um sistema de groupware síncrono na internet onde se pode criar e editar artigos com a participação de mais de uma pessoa de maneira remota e com designação de papéis.

6.2 Contribuições

A ferramenta WebLatex, ao ser disponibilizado em código aberto, adquire um imenso potencial didático, dando a qualquer interessado a possibilidade de aprofundar-se um pouco mais em ferramentas colaborativas. Oferece um ambiente que pode ser usado por qualquer pessoa interessado em aumentar a produtividade e eficiência dos seus projetos científicos, pois é uma ferramenta que também proporciona o aprendizado, porque os membros ao acompanharem os trabalhos dos colegas também enriquecem seu conhecimento com a experiência mutua.

6.3 Trabalhos Futuros

O campo de pesquisas em ferramentas colaborativas (*groupware*) está em constante evolução. Novas ferramentas são inovadas a cada dia. Neste sentido, apresentam-se a seguir algumas melhorias que poderiam ser desenvolvidas e adequadas ao WebLatex.

- Acoplar um Chat no WebLatex;
 - Para melhorar a comunicação, é acoplar na ferramenta um sistema de bate-papo, de voip ou até mesmo videoconferência;
- Melhorar a percepção sistema;
 - Um estudo mais detalhado das técnicas de percepção que vão desde uma simples mudanças de cores na interface até técnicas dinâmicas mais complexas utilizando Ajax, pode tornar mais simples o uso do sistema;
- Melhorar a usabilidade do sistema;
 - Através de um estudo detalhado nas abordagens de usabilidade, ciência que vem tendo grande importância na elaboração do sistema. Um estudo de usabilidade iria garantir um desenvolvimento de artigos mais intuitivo para os usuários.
- Ampliar o Controle de Versões;
 - Acrescentar mais funcionalidade no controle de versões, já que o atual apenas guarda o estado atual.
- Ampliar o WebLatex para outras classes Suportadas pelo Latex;

- Ampliar o WebLatex, para que se possam usar outras classes já suportadas pelo Latex como book, para construção de livro.
- Realizar um trabalho de campo
 - Fazer um trabalho de campo para comprovar cientificamente os resultados do uso da ferramenta WebLatex, na produção de conhecimento, e na melhora da eficiência e produtividade na confecção de artigos.
- Acrescentar uma Ajuda no Sistema
 - Colocar uma ajuda de como elaborar textos bem escritos, a definição e funcionalidade dos comandos Latex.
- Acrescentar um Corretor de Latex em Tempo Real
 - Para que mesmo usuários que não conhecem Latex, sejam alertados por erros comuns no momento em que está confeccionando o texto.
- Acrescentar notas de colaboradores
 - Para que um colaborador ao ler o texto do outro possa acrescentar uma nota que só o colaborador alvo enxergue, de forma a contribuir com conhecimento e possíveis conselhos para enriquecer o grupo
- Colocar um mural de lições aprendidas
 - Para que todos estejam em dia com os conhecimentos adquiridos durante a elaboração do artigo, fortalecendo a aprendizagem.

Bibliografia

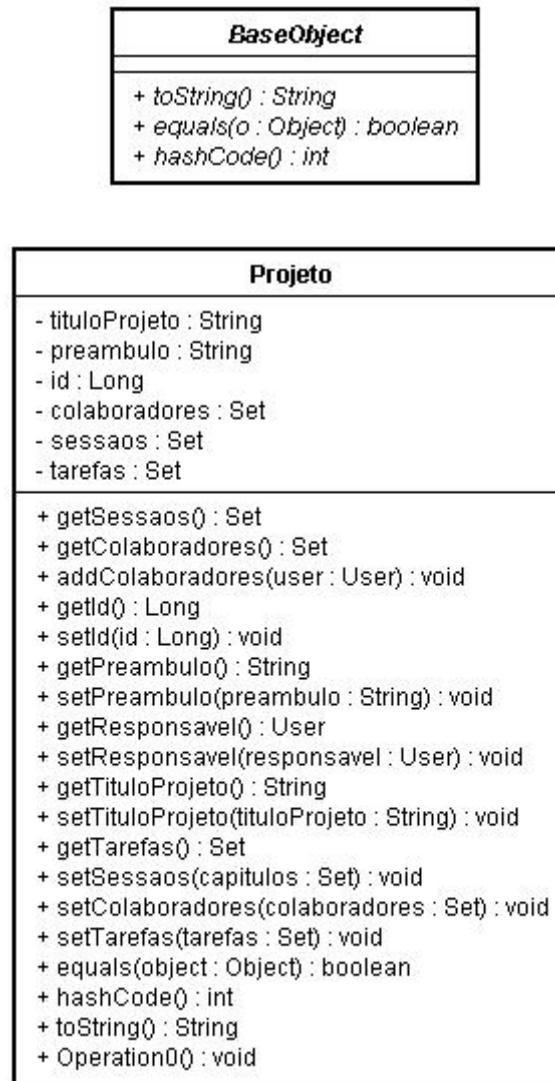
- [1] Moraes da Silva, S. R. Q. “Controle de versões – Um Apoio à Edição Colaborativa na Web”, Universidade de São Paulo, São Carlos, 2005.
- [2] C. Pichiliani, Mauro, M. Hirata, Celso, “Usando a Modelagem Colaborativa no Aprendizado da UML”, Divisão de Ciência da Computação - ITA, São José dos Campos, 2006.
- [3] http://www.mat.ufmg.br/~regi/topicos/intlat.html#tth_sEc0, acessado em 18/09/06
- [4] R. Gonçalves, Paulo e P. P. Padilha, Thereza “Implementação de uma Ferramenta de Edição de Texto Colaborativa”, Sistema de Informação, Centro Universitário Lutero de Palmas, Palmas, Brasil, 2003.
- [5] Gamma, Erich e Helm, Richard e Johnson, Ralph e Vlissides, John, “Padrões de Projeto – Soluções Reutilizáveis de Software Orientado a Objetos”, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.
- [6] Lamport, Leslie, “A- Document Preparation System - User’s Guide and Reference Manual 2nd”, 1985.
- [7] Arriada, Mônica Carapeços, Ramos, EdlaFaust. “Como Promover Condições Favoráveis à Aprendizagem Cooperativa Suportada por Computador ”, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Santa Catarina.
- [8] Santos de Souza, Maria Carolina, Fróes Burnham, Teresina, “Compondo: Uma metodologia para Produção Colaborativa do conhecimento em educação a distância”, Universidade Federal da Bahia.
- [9] Nitze, Júlio, Carneiro, Mára, Geller, Marlize, et. al. “Avaliando Aplicações para Criação de Ambientes de Aprendizagem Colaborativa”. Anais do X Simpósio Brasileiro de Informática na Educação. Curitiba, 1999.
- [10] Pichiliani, Mauro Carlos. “Mapeamento de Software para Permitir a Colaboração Síncrona”, Tese de Pós Graduação, Instituto Tecnológico de Aeronáutica, São Jose dos Campos, SP, Brasil, 2006.
- [11] Quaresma, Pedro. “Curso de Latex”. Parte II, Departamento de Matemática, Faculdade de Ciências e Tecnologia Universidade de Coimbra, 2007.
- [12] Garcia de Carvalho, Marco Antônio. “O Uso de Softwares Livres no Aprendizado de Computação - Latex”, Curso de Ciência da Computação, Campinas, SP, Brasil.
- [13] Campani, Carlos A. P.. “Introdução ao Uso do Preparador de Documentos Latex”, Universidade Federal de Pelotas, Pelotas, RS, 2006.
- [14] Santos, Reginaldo J.. “Introdução ao Latex”, Departamento de Matemática, Universidade Federal de Minas Gerais, Belo Horizonte, 2002.
- [15] Oetiker, Tobias, Partl, Hubert, Hyna, Irene e Schlegl, Elisabeth. “Uma não tão pequena introdução ao Latex2e”, Cambridge, 2007.
- [16] Pichiliani, Mauro Carlos e Hirata, Celso M. “Usando a modelagem colaborativa no aprendizado da UML”, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, 2006.

- [17] <http://ajax4jsf.dev.java.net>, acessado em 15/11/2007.
- [18] <https://appfuse.dev.java.net/>, acessado em 18/09/2006.
- [19] [http://pt.wikipedia.org/wiki/AJAX_\(programa%C3%A7%C3%A3o\)](http://pt.wikipedia.org/wiki/AJAX_(programa%C3%A7%C3%A3o)), acessado em 02/03/2007.
- [20] <http://jva.sun.com> , acessado em 12/11/2007.
- [21] <http://tomcat.apache.org>, acessado em 12/11/2007.
- [22] <http://ant.apache.org>, acessado em 12/11/2007.
- [23] <http://dev.mysql.com/doc/>, acessado em 12/11/2007.
- [24] <http://www.gnu.org>, acessado em 12/11/2007.
- [25] <HTTP://wikipedia.br/fgttytr>, acessado em 25/11/2007.
- [26] Oliveira, K., Zlot, F., Rocha, AR., Travassos, G., “Domain Oriented Software Development Environment”, Jornal of Systems and Software,2004.
- [27] <http://www.minerva.uevora.pt/cscl/>, acessado em 12/11/2007.
- [28] McConnell, David. “ Implementing Computer Supported Cooperative Learning”. Reino Unido: Biddles Ltd.,1994.
- [29] Werner, Cláudia e colaboradores. “OdysseyShare: Um ambiente para o Desenvolvimento Cooperativo de Componentes”, Universidade Federal de Juiz de Fora, Rio de Janeiro, RJ.
- [30] <http://www.springframework.org>, acessado em 11/11/2007.
- [31] <http://www.hibernate.org>, acessado em 09/11/2007.
- [32] <http://java.sun.com/javaee/javaserverfaces>, acessado em 08/11/2007.
- [33] <http://www.acegisecurity.org>, acessado em 15/11/2007.

Apêndice A

Classes Básicas do Sistema

Apresentaremos aqui as todas as classes que compõem o pacote model do WebLatex



Address
<pre> - serialVersionUID : long = 3617859655330969141L # address : String # city : String # province : String # country : String # postalCode : String + getAddress() : String + getCity() : String + getProvince() : String + getCountry() : String + getPostalCode() : String + setAddress(address : String) : void + setCity(city : String) : void + setCountry(country : String) : void + setPostalCode(postalCode : String) : void + setProvince(province : String) : void + equals(o : Object) : boolean + hashCode() : int + toString() : String </pre>

Sessao
<pre> - id : Long - titulo : String - corpo : String - status : boolean - responsavelEdicao : User - idDoCriador : User - projeto : Projeto + getIdDoCriador() : User + getReponsavelEdicao() : User + isStatus() : boolean + setIdDoCriador(idDoCriador : User) : void + setReponsavelEdicao(responsavelEdicao : User) : void + setStatus(status : boolean) : void + getCorpo() : String + setCorpo(corpo : String) : void + getId() : Long + setId(id : Long) : void + getTitulo() : String + setTitulo(titulo : String) : void + equals(object : Object) : boolean + hashCode() : int + toString() : String + getProjeto() : Projeto + setProjeto(projeto : Projeto) : void </pre>

User
<pre> - serialVersionUID : long = 3832626162173359411L # id : Long # username : String # password : String # confirmPassword : String # firstName : String # lastName : String # phoneNumber : String # email : String # website : String # passwordHint : String # version : Integer # enabled : boolean # accountExpired : boolean # accountLocked : boolean # credentialsExpired : boolean - projetos : Set - roles : Set + User() + User(username : String) + getId() : Long + getUsername() : String + getPassword() : String + getConfirmPassword() : String + getFirstName() : String + getLastName() : String + getFullName() : String + getAddress() : Address + getEmail() : String + getPhoneNumber() : String + getWebsite() : String + getPasswordHint() : String + getRoles() : Set + addRole(role : Role) : void + getAuthorities() : GrantedAuthority[] + getVersion() : Integer + isEnabled() : boolean + isAccountExpired() : boolean + isAccountNonExpired() : boolean + isAccountLocked() : boolean + isAccountNonLocked() : boolean + isCredentialsExpired() : boolean + getProjetos() : Set + setProjetos(projetos : Set) : void + isCredentialsNonExpired() : boolean + setId(id : Long) : void + setUsername(username : String) : void + setPassword(password : String) : void + setConfirmPassword(confirmPassword : String) : void + setFirstName(firstName : String) : void + setLastName(lastName : String) : void + setAddress(address : Address) : void + setEmail(email : String) : void + setPhoneNumber(phoneNumber : String) : void + setWebsite(website : String) : void + setPasswordHint(passwordHint : String) : void + setRoles(roles : Set) : void + setVersion(version : Integer) : void + setEnabled(enabled : boolean) : void + getRoleList() : List + setAccountExpired(accountExpired : boolean) : void + setAccountLocked(accountLocked : boolean) : void + setCredentialsExpired(credentialsExpired : boolean) : void + equals(o : Object) : boolean + hashCode() : int + toString() : String </pre>

Tarefa
- descricao : String - id : Long - prazo : Date - ResponsavelPelaTarefa : User - projeto : Projeto + getId() : Long + setId(id : Long) : void + getPrazo() : Date + setPrazo(prazo : Date) : void + getDescricao() : String + setDescricao(descricao : String) : void + getResponsavelPelaTarefa() : User + setResponsavelPelaTarefa(responsavelPelaTarefa : User) : void + equals(object : Object) : boolean + hashCode() : int + toString() : String + getProjeto() : Projeto + setProjeto(projeto : Projeto) : void

Role
- serialVersionUID : long = 3690197650654049848L - id : Long - name : String - description : String + Role() + Role(name : String) + getId() : Long + getAuthority() : String + getName() : String + getDescription() : String + setId(id : Long) : void + setName(name : String) : void + setDescription(description : String) : void + equals(o : Object) : boolean + hashCode() : int + toString() : String

Apêndice B

Parâmetros de configuração para o WebLatex:

O WebLatex utiliza o Programa Miktex para versões que estão instalados em servidores Windows. Mas outras versões de Latex podem ser usadas no caso de estar em outra plataforma como o Linux.

Para isso são necessários alguns parâmetros de configurações :

- Deve-se deixar os comandos apresentados a seguir configurados como variáveis de ambiente, pois o WebLatex utiliza o prompt (ou Shell no caso do Linux), para fazer as chamadas ao Latex.
- Deve-se configurar no arquivo comandos.txt o comando de compilar o arquivo *.tex e o comando de gerar o pdf a partir do *.tex, pois esses comandos podem sofrer alterações dependendo da plataforma.

Exemplo de como o comando deve ser inscrito no arquivo para geração de pdf (Miktex) :

Comando [diretorio] [nomeDoArquivo.tex]

pdflatex c:// teste.tex ;

O arquivo comandos.txt está no pacote util da camada *viewer* do WebLatex