

**SISTEMA DE GESTÃO DE TRABALHOS
DE FINAL DE CURSO**

LUCAS BRITO ARRUDA

**SISTEMA DE GESTÃO DE TRABALHOS
DE FINAL DE CURSO**

Monografia apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Especialista em Ciência da Computação.

ORIENTADOR: ANGELO DE MOURA GUIMARÃES

Belo Horizonte
Fevereiro de 2010

© 2010, Lucas Brito Arruda.
Todos os direitos reservados.

Arruda, Lucas Brito

Sistema de Gestão de Trabalhos de Final de Curso / Lucas
Brito Arruda. — Belo Horizonte, 2010

xxiv, 26 f. : il. ; 29cm

Universidade Federal de Minas Gerais

Orientador: Angelo de Moura Guimarães

1. Software. 2. Metodologia ágil. 3. Monografia. I. Título.

[Folha de Aprovação]

Quando a secretaria do Curso fornecer esta folha,
ela deve ser digitalizada e armazenada no disco em formato gráfico.

Se você estiver usando o `pdflatex`,
armazene o arquivo preferencialmente em formato PNG
(o formato JPEG é pior neste caso).

Se você estiver usando o `latex` (não o `pdflatex`),
terá que converter o arquivo gráfico para o formato EPS.

Em seguida, acrescente a opção `approval={nome do arquivo}`
ao comando `\ppgccufmg`.

*Dedico esse trabalho à Deus, minha noiva Michelly, minha família e meus amigos,
que sempre me apoiaram incondicionalmente.*

Agradecimentos

Agradeço ao Prof. Angelo, pois seus conselhos, sugestões e correções foram cruciais para a qualidade e conteúdo do trabalho. Agradeço também a Prof. Maria de Lourdes pelas sugestões e pela ajuda na pesquisa.

*“Uma mente que se abre a uma nova idéia
jamais voltará ao seu tamanho original.”*
(Einstein)

Resumo

O objetivo deste trabalho é propor a criação de uma ferramenta para padronização e gerenciamento de trabalhos de conclusão de curso. Presente em diversas instituições, desde a graduação até o doutorado, este trabalho envolve alunos, professores, bibliotecários. Durante o processo de escrita, são elaboradas e impressas diversas versões do texto, comentários e revisões são tecidas e mais versões são geradas. Após “pronto”, o texto ainda irá para uma revisão final do Orientador, outra versão será enviada para a Banca, e esta, mediante aprovação do trabalho, ainda irá propor correções e melhorias e, finalmente, o produto final do texto ainda irá para revisão do bibliotecário. É possível perceber, que além de todo o desperdício de papel, ainda é desperdiçado tempo com entregas pessoais, diversas impressões do texto e anotações e comentários que podem vir a se perder. Isso também dificulta rastrear a evolução do desenvolvimento do trabalho e outras métricas acerca do mesmo. Este software proposto, irá permitir o armazenamento de revisões do texto, o envio do texto para Professores, Bibliotecários e Revisores, o registro de comentários e resposta aos comentários de uma revisão, além de mostrar prazos de entrega e mostrar evolução do trabalho de acordo com as etapas do processo.

Palavras-chave: Software, Metodologia ágil, Monografia.

Abstract

This objective of this work is to propose the development of a tool to manage and to standardize end of course works. Present in many institutions and Universities, since the Graduation to higher levels as PhDs, this kind of work involves Studentes, Teachers and also Librarians. During the writing process, many versions of the work are printed, delivered to revision and with that more versions are generated and printed again. Also, we have comments made about the text. Many paper is lot and also time. As the work is “done”, you have to deliver it to your Teacher/Revisor which will make corrections, send you again another version and you will have to generate and print another version to the judging committee. If they approve it, they will ask for some chances in the text and after that, also the Librarian which will revise all the References for the text. So, it’s clear the is a lot of effort, money and time being wasted in this process. All that could be traced, logged and stored digitally and one could see if the work is evolving properly and another metrics could also be used too. The proposed software, is going to allow a more efficient way of doing this, without the need to print, deliver, to comment on paper and to have to fisically meet with the people involved in the work.

Keywords: Software, Agile Development, Paper.

Lista de Figuras

4.1	Mapa mental do problema.	12
4.2	Tela do SCG.	14
4.3	Tela do JEMS.	15
4.4	Modelo do banco de dados.	16
5.1	Tela inicial do sistema: Autenticação.	18
5.2	Tela de autenticação preenchida.	19
5.3	Trabalhos do aluno.	20
5.4	Detalhes do trabalho.	21
5.5	Detalhes do trabalho.	22
5.6	Visão geral do administrador.	23

Lista de Tabelas

4.1	Funcionalidades do software.	13
5.1	Tarefas do software.	17

Lista de Símbolos

<i>XP</i>	eXtreme Programming
<i>CMMI</i>	Capability Maturity Model Integration
<i>RUP</i>	Rational Unified Process
<i>DCC</i>	Departamento de Ciência da Computação
<i>UFMG</i>	Universidade Federal de Minas Gerais
<i>SCG</i>	Sistema de Gestão de Conferências
<i>JEMS</i>	Journal and Event Management System

Sumário

Agradecimentos	ix
Resumo	xiii
Abstract	xv
Lista de Figuras	xvii
Lista de Tabelas	xix
Lista de Símbolos	xxi
1 Introdução	1
1.1 Objetivo Geral	2
1.2 Objetivos Específicos	2
2 A Monografia	3
2.1 Busca do Conhecimento	3
2.2 Tema	3
2.3 Revisão Bibliográfica	4
2.4 Objetivos	4
2.5 Planejamento da Pesquisa	4
2.6 Coleta e Análise de Dados	4
2.7 Relatório final	5
2.8 O Processo	5
3 Desenvolvimento Ágil	7
3.1 O Que é	7
3.2 Origem	8
3.3 Atualmente	8

3.4	Conceitos	8
3.4.1	Objetivos	8
3.4.2	Atividades	9
3.4.3	Valores	9
4	Elaboração	11
4.1	Modelo	11
4.2	Atividades	12
4.3	Revisão sobre produtos correlatos	13
4.4	Concepção	15
5	Desevolvimento	17
5.1	Tarefas	17
5.2	Funcionalidades	18
5.2.1	Autenticação	18
5.2.2	Upload e Submissão	19
5.2.3	Lista de trabalhos	21
5.2.4	Gerência de usuários	22
5.3	Trabalhos futuros	23
	Referências Bibliográficas	25

Capítulo 1

Introdução

O desenvolvimento de uma monografia é um processo comum à alunos de graduação e pós-graduação. Esse processo de escrita, embora pessoal, envolve geralmente os mesmos elementos: orientador, aluno, trabalho proposto e concretizado e uma banca julgadora. Embora possamos descrever melhor este processo, e posteriormente isto será feito neste trabalho, podemos dizer que ele tem basicamente uma constância e semelhança no mundo acadêmico.

O que este trabalho propõe é criar um software de gestão do processo de escrita de um trabalho de conclusão de curso, focando principalmente nas atividades do mesmo e da suas interações, como, por exemplo, o planejamento, entregas e devoluções de revisões, recomendações e pareceres. Isso poderá facilitar o processo como um todo, para o gesto do processo, para o aluno, para o bibliotecário, para o professor, para o comitê de avaliação e para quem mais possa fazer parte deste processo, como professores e outros colaboradores.

Existem manuais para confecção como o manual para elaboração de monografias da pós-graduação da UFMG (COSTA [2005] e Forte [2004]), um manual para desenvolvimento de monografias, dissertações e teses. Existe também a ABNT NBR 14724 [2005]. Porém, os mesmos se limitam a falar sobre a formatação e normas do texto, não abrangendo o processo de confecção e as etapas do processo de escrita mencionadas anteriormente.

A área abrangida por este trabalho é a área de desenvolvimento de software ou análise de sistemas, aonde será focado um desenvolvimento ágil com várias pequenas entregas funcionais do produto. Outras possíveis áreas são a literária e de gestão de documentos.

1.1 Objetivo Geral

O trabalho visa a auxiliar um processo, em muitos casos manual, praticado por Universidades que é o gerenciamento do trabalho de final de curso. Este trabalho é muito comum e conhecido de alunos do ensino superior e muitas vezes pode ter dificuldades se pouco organizado ou uma com organização pouco efetiva, levando a problemas de comunicação e interação entre alunos e os demais participantes.

Mediante a assunção de tais problemas, um sistema de gestão de trabalhos de final de curso será desenvolvido, de modo a amenizar tais problemas e padronizar o desenvolvimento deste tipo de trabalho.

O desenvolvimento deste trabalho se baseia nas técnicas ágeis de software, para que seja produzido um software fácil de se manter e que tenha uma boa utilidade desde o início, dado que segundo Beck & Andres [2004], o software desenvolvido com processos ágeis é feito em pequenas iterações com entrega de código funcional, no qual pode-se ainda obter o feedback.

1.2 Objetivos Específicos

São objetivos específicos:

- Estudar o processo básico de elaboração, desenvolvimento e apresentação de monografias;
- Procurar ferramentas com propostas similares e comparar com a solução proposta;
- Estudar metodologias de desenvolvimento de software;
- Elaborar do modelo básico e sucessivas interações até o produto final;
- Apresentar de telas do produto, relatórios e gráficos comparativos da primeira iteração;
- Desenvolver um protótipo do produto da primeira iteração;

Capítulo 2

A Monografia

2.1 Busca do Conhecimento

Na conclusão de curso, Universidades tem, geralmente, um “trabalho de formatura” ou “trabalho de final de curso” Martins [2010]. Este, resumidamente, começa com uma pesquisa, seguida da escrita e normalmente tem-se um produção com resultados e conclusões, sejam estes positivos ou negativos. Mas qual é o porque disto?

Segundo Martins [2010], o objetivo, apesar de parece óbvio, é “conhecer a verdade”/“conhecer a realidade” e para isso o homem faz uso de várias técnicas como a leitura, escrita e pesquisa para obter isto. E, as duas últimas tem origem na primeira, pois é com leitura que se possibilita conhecer mais sobre tudo. Portanto, *“a leitura é fundamental para a busca do conhecimento”* Martins [2010].

Nesta busca, o método científico entra com um grande papel para se obter e verificar conhecimento, apresentando caminhos, métodos, técnicas e procedimentos. Com isso, é possível levantar hipóteses, testar e comprovar ou não estas hipóteses.

2.2 Tema

A escolha do tema não possui um regra pronta. Há muitas variáveis como tendências, especialidade no assunto, preferência ou curiosidade pelo assunto, entre outras. Porém, segundo Martins [2010] tem caráter lógico, acrítico, pouco idealista - para não ser tendencioso - e não se enquadrar no senso-comum.

Novamente, neste passo a leitura se faz muito importante, pois pode ser o ponto de partida para um questionamento, uma idéia ou inquietação. Porém, é preciso que o mesmo seja original e viável, tendo importância e trazendo relevância para alguma

área do conhecimento.

2.3 Revisão Bibliográfica

Após a definição do tema, é importante pesquisar diversas referências sobre o mesmo. É preciso saber o que já foi feito até o momento e se o trabalho proposto poderá ser inovador ou não. Com isso é possível levantar material para dar base ao trabalho.

Com a leitura crítica do material, documenta-se o que interessa, pois isto servirá de apoio para as idéias levantadas e para o que for discutido e feito no trabalho. Sem as fontes, o trabalho poderá estar repetindo algo que já foi feito, ter pouca fundamentação de que aquilo é válido, além de comprometer a qualidade do mesmo pois não haverá contrapontos com os quais comparar o que foi feito.

2.4 Objetivos

O objetivo descreve as intenções que se tem com aquele trabalho. Ele deve ser claro, pois “objetivos vagamente anunciados não serão atingidos” Martins [2010]. Fixa-se, então, um objetivo geral e alguns objetivos específicos, o que servirá de guia para as atividades do projeto.

2.5 Planejamento da Pesquisa

De acordo com Martins [2010], nesta seção, há que se definir metodologias, procedimentos e planejar como será feita a pesquisa. É um plano de operação que fará perguntas como “como fazer?”, “com o que fazer?”, “onde fazer?” e “quando fazer?”. Esse planejamento servirá de guia para orientar a pesquisa. Esse planejamento deve considerar também o orçamento para o projeto e o universo de pesquisa, pois estes podem ser fatores limitantes.

2.6 Coleta e Análise de Dados

Dados podem ser coletados em publicações são chamados de dados secundários e portanto exigem identificação da fonte. Enquanto dados obtido através de questionários ou entrevistas, são dados primários. Estes devem ter cópia anexa no relatório final da pesquisa.

Com os dados em mãos, segundo Martins [2010], é preciso analisar visando atacar o problema de pesquisa que foi proposto, alcançar os objetivos propostos e para testar hipóteses feitas. Para isso pode-se aplicar técnicas estatísticas e classificar em gráficos, curvas, médias e porcentagens, e assim verificar índices, como por exemplo a aceitação de um produto ou de uma teoria. É preciso fazer uso de senso crítico e também pode haver dados subjetivos que necessitam de análise crítica para que possa ser inferido significado dos mesmos.

2.7 Relatório final

O relatório basicamente “deve apoiar-se numa redação clara, precisa, objetiva e consistente, (...) a linguagem científica não deve deixar margem a interpretações diversas (...), não deve ser redundante (...), deve ser modesta e desprovida de adjetivações” Martins [2010]. Com isso, é possível alcançar objetividade no mesmo.

“A finalidade do relatório é transmitir com exatidão o desenrolar da pesquisa, suas limitações, descobertas, análise de dados, interpretações, conclusões e recomendações.” Martins [2010]

2.8 O Processo

O processo de escrita pode variar de uma Universidade para outra. Embora, o processo de elaboração seja similar, dependendo mais do tipo de trabalho do que do local em que o mesmo é realizado, não há uma documentação e padronização do processo como um todo. Pois, o mesmo envolve aluno, professor, revisor, bibliotecário, banca, entre outros. Porém, este acompanhamento pode ser tornar trabalhoso, envolvendo diversas impressões do texto e entregas pessoais e conversas, o que poderia ser feito de maneira mais prática com o auxílio de um software ou sistema.

O que este trabalho propõe é justamente entender, padronizar e facilitar este trabalho por meio de um sistemas para gerir estas entregas, revisões, comentários acerca do trabalho, que serão feitas durante o curso do mesmo.

Capítulo 3

Desenvolvimento Ágil

3.1 O Que é

O desenvolvimento ágil, é “*um estilo de desenvolvimento de software focado na aplicação de excelentes técnicas de desenvolvimento de software, comunicação clara e trabalho em equipe (...)*” [Beck & Andres, 2004, tradução livre]. Ele basea-se no fato de um bom programa é “*leve, conciso e inclui apenas as funcionalidades que agregam valor ao cliente*” [Beck & Andres, 2004, tradução livre].

O movimento do desenvolvimento ágil surge com a proposta do eXtreme Programming (XP), elaborado por Kent Beck e conhecido por movimento ágil [Beck & Andres, 2004]. Sua intenção é que se melhore a qualidade do softwares produzidos atualmente e a responsividade à constante mudança dos requerimentos de um software.

Embora ele seja um movimento novo Beck [1999], algumas das suas técnicas como planejar e escrever testes antes de cada pequena interação já eram usadas pela Nasa em 1960 no Projeto *Mercury* (Larman 2003), sendo que a maior diferença é que tais práticas são levadas ao extremo e o software deve ser o mais conciso e limpo possível.

Outras abordagens ágeis também foram propostas como, por exemplo, o Scrum, em que a parte de planejamento é muito mais focada. Projetos Scrum progredem através de uma série de iterações chamadas “sprints”. Cada “sprint” é tipicamente de 2 a 4 semanas. Assim como XP, Scrum é usado para projetos que sofrem rápida mudança ou que tenham requisitos altamente emergentes, típicos em desenvolvimento de software. [<http://www.mountangoatsoftware.com/topics/scrum>].

3.2 Origem

Antes do desenvolvimento ágil, técnicas e metodologias para o desenvolvimento de software já se encontravam amplamente usadas e divulgadas, como o Capability Maturity Model Integration (CMMI) e o Rational Unified Process (RUP). Em 1990, ganha força a programação por orientação à objetos e com isso há uma mudança no paradigma do desenvolvimento de software, fazendo com que os requerimentos de software mudassem com maior velocidade e assim também o seu ciclo de vida.

Em 1999, nascia, então, o XP Beck [1999]. Esta nova metodologia de desenvolvimento prezava por ciclo de vida menores, e por um desenvolvimento cíclico, aonde se mudava o foco do software para as pessoas (*“indivíduos e interações, em detrimento a processos e ferramentas”* Paula Filho [2009]) e o mesmo era entregue em partes, de tempos em tempos, sempre com entregas funcionais e que proviam utilidade para o cliente.

3.3 Atualmente

Embora esta nova metodologia tenha atraído quando veio a público, sua adoção foi difícil pela grande mudança de paradigma, aonde os testes tem tanta importância, porém gastam boa parte do tempo de desenvolvimento, complicando os cronogramas. Sem contar outras características rígidas, como o pequeno ciclo de desenvolvimento com entrega de software funcional e a programação em pares.

Nos dias atuais, sua adoção vem crescendo bastante. Isso se deve à conscientização do mercado de que testes são importantes, à crescente demanda de softwares que tem pequeno ciclo de vida e que precisam ser lançados ao mercado com grande rapidez, à evolução da metodologia e ao maior entendimento das práticas do XP. Além disso, todo o mundo do software se volta à web, que é um meio bem mais dinâmico do que o “velho” desktop, aonde todos os dias surge um novo concorrente com um software melhor e mais inovador.

3.4 Conceitos

3.4.1 Objetivos

O objetivo primário do XP é ser uma disciplina que organiza pessoas para que elas produzam software de melhor qualidade e para que sejam mais produtivas. Ao contrário do RUP e do CMMI, aonde tudo é padronizado, cada indivíduo tem a sua própria tarefa

e indivíduos podem ser substituídos, no XP cada indivíduo tem grande valor e deve focar no que faz de melhor, produzindo somente o necessário. Isso possibilita uma maior qualidade no que é feito, além de aumentar a produtividade.

Além disso, o XP força um ciclo curto de desenvolvimento, o que mais tarde se mostra como grande redutor de custos se comparado às outras duas metodologias supracitadas, pois faz com que seja mais possível e viável haver mudanças e atualizações dos requisitos. Mudanças, essas, que serão incorporadas no próximos ciclos de acordo com a sua importância.

3.4.2 Atividades

São basicamente quatro as atividades principais no XP. Testar o software, de preferência antes que as funcionalidades sejam desenvolvidas. Codificar, pois sem o código não temos o programa funcional. Habilidade de ouvir, pois é preciso escutar e entender o que o cliente realmente precisa. Design, de modo que seja o mais simples possível e que seja evolucionário, adaptável e adaptado com o tempo.

3.4.3 Valores

Os valores do desenvolvimento ágil são mais relativos ao aspecto humano do que ao aspecto máquina. Segundo Beck & Andres [2004], é importante que a equipe se entenda bem e que tudo seja comunicado de forma clara e de modo que ninguém da equipe fique preso numa tarefa por não saber como fazer ou por estar tendo dificuldades. Portanto, o primeiro e o principal valor de uma equipe é a comunicação.

O segundo valor é simplicidade. Tudo deve ser pensado e feito da forma mais simples possível. E isso não significa de “qualquer forma”, mas indica que o desenvolvimento deve ser focado naquilo que é essencial e que deve ser feito com cuidado. Isso evita *bugs*, a superestimação e gasto de tempo do desenvolvedor em design e implementação e cortando também funcionalidades que não serão usadas e agregaram pouco valor e alto custo ao cliente. Futuramente isto cortará custos de manutenção. Nada impede também que posteriormente seja feito uma reestruturação para que o programa tenha mais funcionalidades.

Feedback é o terceiro valor. Duas cabeças pensam melhor que uma, portanto mostrar o que foi feito para equipe é ter uma oportunidade de ver se foi implementado da melhor maneira e é uma oportunidade de aprendizado e de correção de problemas antes que os mesmos surjam. Além disso, o XP propõe o desenvolvimento em pares, onde um programador revê imediatamente o trabalho do outro, não deixando um erro

propagar para ser excluído mais tarde, quando será muito mais difícil e custoso. Apresentar para o cliente também valida o que foi feito e garante que cliente e contratada estão de acordo, novamente cortando futuros custos de manutenção e aumentando a satisfação do mesmo.

O quarto valor é coragem. As vezes é preciso mudar parte do software e o XP diz que o desenvolvedor deve se sentir seguro para refatorar quando precisar, de acordo com as necessidades. Isso inclui sempre remover código obsoleto e não utilizado, a fim de sempre manter o código limpo e conciso.

Finalmente, o quinto valor é respeito. O programador tem que ter respeito por seus colegas. Nunca quebrar o código-fonte, tornando o não funcional. Nunca quebrar testes, tornando-os inúteis. Respeitar as diferenças e qualidades de cada um, sem desvalorizar o trabalho do outro. Além disso, a documentação bem feita do código-fonte é essencial (e quase sempre a única documentação gerada).

É, então, uma metodologia muito mais humana, que valoriza o indivíduo, sempre tentando estimular e desafiar o mesmo. Isso faz com que o programador trabalhe mais motivado e satisfeito, aumentando a produtividade e a qualidade do produto.

Capítulo 4

Elaboração

4.1 Modelo

Para se entender melhor o problema e o trânsito que ocorre com uma monografia durante o processo de escrita, correções, revisões, até a aprovação e correção final, foi elaborado um modelo mental.

Histórias de usuários são um dos artefatos de desenvolvimento primário para equipes em um projeto Scrum e XP. Uma história de usuário é uma definição de alto nível de requisitos, contendo informações suficientes para que os desenvolvedores possam produzir uma estimativa razoável do esforço para implementá-lo. O diagrama abaixo corresponde à história de usuários e foi revisado a partir de entrevistas com a Professora Maria de Lourdes Coelho, responsável pela logística de apoio a alunos e professores dos cursos de especialização do DCC/UFMG (Engenharia de Software e Análise de Sistemas) no desenvolvimento de trabalhos finais de curso.

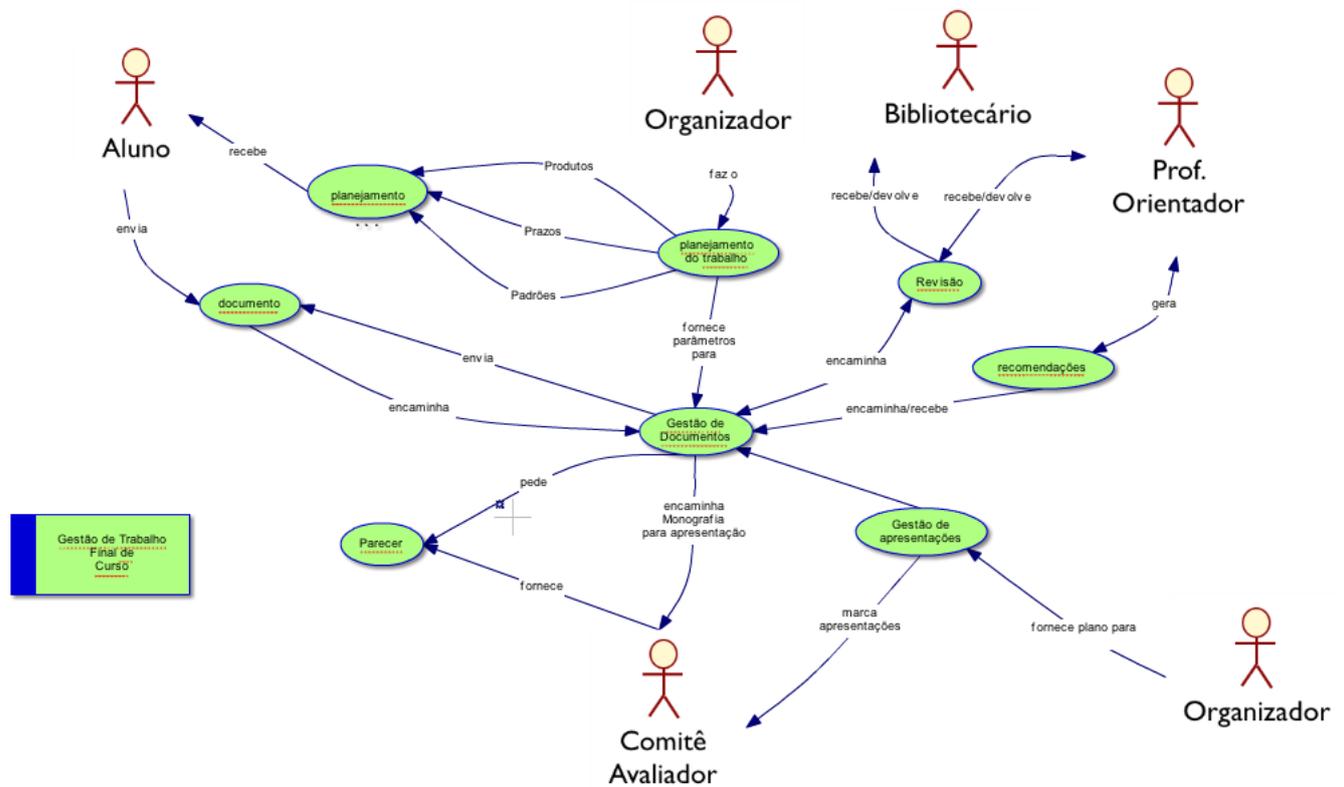


Figura 4.1. Mapa mental do problema.

4.2 Atividades

Para elaborar mais o sistema, foram eleitas as principais funcionalidades do mesmo, que serão, mais tarde, detalhadas em atividades para a sua elaboração.

À partir destas quatro principais funcionalidades da ferramenta, herdaram todas as outras tarefas que a mesma deverá ser capaz de executar. Como, por exemplo, prover um meio de autenticar usuários e outras tarefas que estão dentro das funcionalidades sugeridas, porém podem ter sua função resumida com as próprias.

Tabela 4.1. Funcionalidades do software.

<i>No.</i>	<i>Funcionalidade</i>	<i>Descrição</i>
1	Salvar revisões do texto	Salvar todas as versões enviadas do texto
2	Disponibilizar o texto para o Orientador	Disponibiliza uma maneira do Orientador baixar o texto e tecer comentários sobre o mesmo
3	Disponibilizar o texto para a Banca	Disponibiliza texto para a Banca baixar e uma caixa para comentários
4	Disponibilizar o texto para o Bibliotecário	Disponibiliza o texto para o Bibliotecário e um caixa para comentários

4.3 Revisão sobre produtos correlatos

Foram investigados alguns produtos de software correlatos ao proposto neste trabalho. Não foi encontrado nenhum produto com as mesmas especificidades, mas foram localizados alguns com certa similaridade. Dois deles se aproximaram mais dos objetivos desse trabalho e seram descritos sucintamente a seguir.

Coelho D et ali. [2008] apresenta uma proposta de um sistema de conferências (SCG - Sistema de Gestão de Conferências) que tem uma boa analogia com o sistema de gestão de trabalhos de fim de curso. A Figura, abaixo, mostra a interface de captação de um trabalho de um autor.



Figura 4.2. Tela do SCG.

O JEMS [2010] tem por objetivo dar suporte ao gerenciamento de submissões e revisão de artigos para eventos científicos. O acrônimo JEMS vem da descrição do sistema na língua inglesa "Journal and Event Management System". A adoção do JEMS é recomendado pela SBC na organização de seminários, congressos e simpósios.

O JEMS dá suporte a diversos tipos de usuários: Autor de artigo, com instruções para submissão (upload) de arquivos. Revisor de artigo, com informações sobre tópicos de interesse, e acesso aos artigos atribuídos ao revisor, bem como informar sobre as revisões publicadas. Coordenador de evento, com auxílio para gerência do evento.

Coordenador de publicação, com ajudas para impressão e organização dos anais do evento e acesso a informações sobre os artigos aceitos e o download dos arquivos. Coordenador de sessão, com auxílio sobre uma sessão no evento e acesso aos arquivos, e informações sobre os autores, e revisões dos artigos de uma dada sessão.

A Figura, mostra um exemplo da interface de captação de um artigo de um dado autor.

Accepting paper registration and upload

Paper submission	Conference official site	Conference chair email	Conference administrator	Paper registration deadline	Paper upload deadline	Subconferences
SBRC 2008				Feb 06, 2008	Feb 07, 2008	-
NOMS 2008				Dec 10, 2007	Dec 10, 2007	-

2. Na página do evento no sistema JEMS o autor deve clicar no botão laranja "Submit paper":



3. Preencher as informações do artigo no formulário de registro:

- **Author:** o sobrenome ou e-mail do autor registrado no JEMS. Todos os autores do artigo devem ser registrado no JEMS antes do registro do artigo. O JEMS permite um número máximo de 8 autores por artigo;
- **Paper title:** o título do artigo;
- **Paper abstract:** o resumo do artigo;
- **Paper topics:** selecione os tópicos de interesse do artigo;
- **Track:** seleciona a trilha de submissão do artigo. Se o evento tiver apenas uma trilha ela já estará selecionado para você por padrão.

4. Clique no botão "Submit" e você verá uma tela de confirmação do registro do seu artigo;

5. Para submeter o arquivo associado ao artigo clique no link "upload manuscript" que aparece na tela de confirmação do registro do artigo. Os formatos aceitos para o arquivo são definidos pelo coordenador do evento, por exemplo, PDF, PS, ou DOC. O "manuscript" do artigo também pode ser enviado após feito o registro, veja abaixo.

Figura 4.3. Tela do JEMS.

4.4 Concepção

Primeiramente, foi construído um modelo de banco de dados para dar apoio ao sistema. À partir desse modelo, é possível se ter uma idéia de como funcionará o software e o relacionamento entre as entidades do mesmo.

Com este modelo em mãos, podemos partir para o desenvolvimento do software.

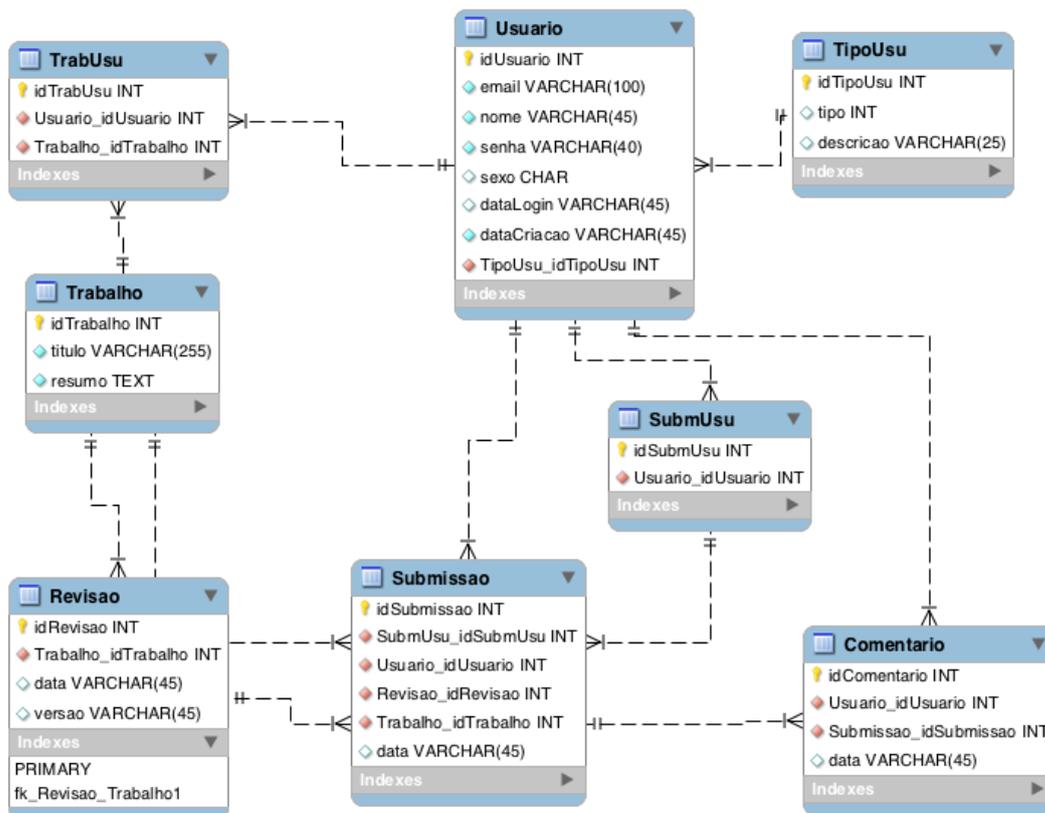


Figura 4.4. Modelo do banco de dados.

Capítulo 5

Desevolvimento

5.1 Tarefas

Usando as funcionalidades definidas anteriormente, iremos determinar as tarefas que devem ser feitas. Para sabermos qual será a ordem de implementação de acordo com sua importância, será feita atribuição de pesos. Para que estes pesos possam ser interpretados de forma correta, os mesmos terão um certa distância entre seu valor.

Tabela 5.1. Tarefas do software.

<i>No.</i>	<i>Descrição</i>	<i>Peso</i>
1	Autenticação de usuários	10
2	Upload de texto	10
3	Envio de texto para revisão	10
5	Controle de versão	7
4	Comentários nos textos	7
6	Enviar e-mails de notificação	4
7	Trocar senha	4
8	Mostrar prazos	4
9	Mostrar evolução do trabalho	4
10	Atualizar dados cadastrais	1
11	Chavear a visibilidade de trabalhos concluídos	1

Então, o esforço será focado para que todos as tarefas de nível 10 estejam prontas na primeira interação do software. Outras tarefas podem ser feitas na mesma interação, caso seja possível concluir as anteriores. Porém não são tão primordiais como as primeiras.

Na interação seguinte, escolhe-se mais tarefas, de acordo com a prioridade e assim se segue até que o trabalho esteja concluído. Mas, isso não significa que um software

esteja concluído, pois segundo Paula Filho [2009] um software nunca está terminado e o que irá determinar isso são fatores como tempo e dinheiro, porém há sempre possibilidade de melhoria e adição de funcionalidades.

5.2 Funcionalidades

5.2.1 Autenticação

Toda a base do sistema se dá na autenticação e envio do texto. Abaixo é possível ver a tela principal de autenticação:

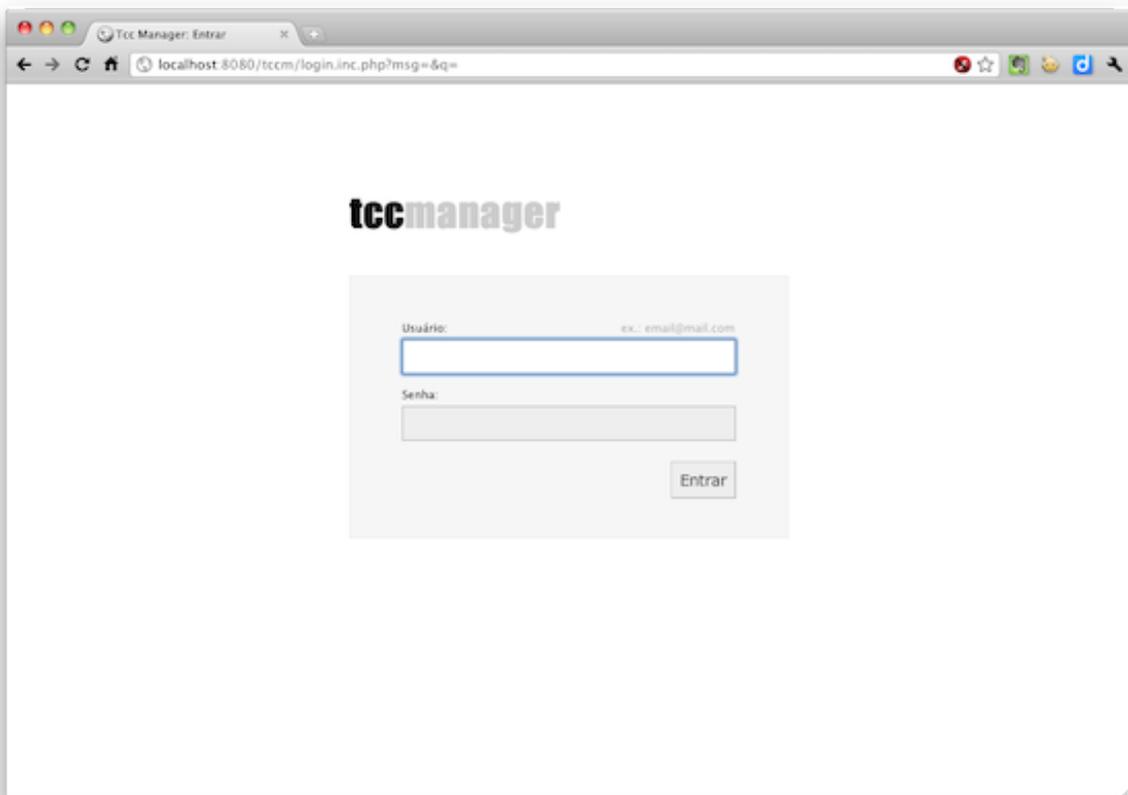


Figura 5.1. Tela inicial do sistema: Autenticação.

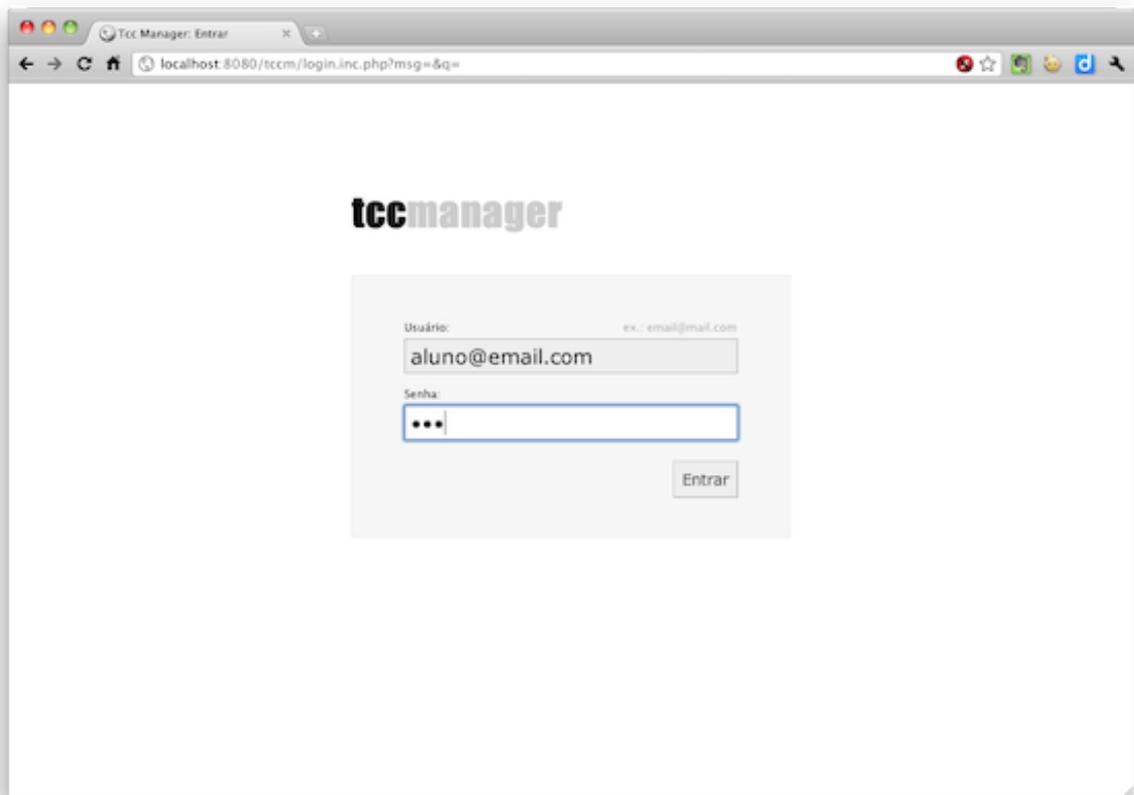


Figura 5.2. Tela de autenticação preenchida.

Para cada usuário o sistema mostrará um contexto. Sendo que se ele for um administrador, irá poder apenas gerenciar o sistemas, criando novas contas de usuário no mesmo. Se for um professor, poderá ver trabalhos, submissões de alunos, cadastrar trabalhos, cadastrar alunos em trabalhos e ver alunos.

5.2.2 Upload e Submissão

Na tela principal do aluno, podemos ver uma lista dos trabalhos que ele participa:

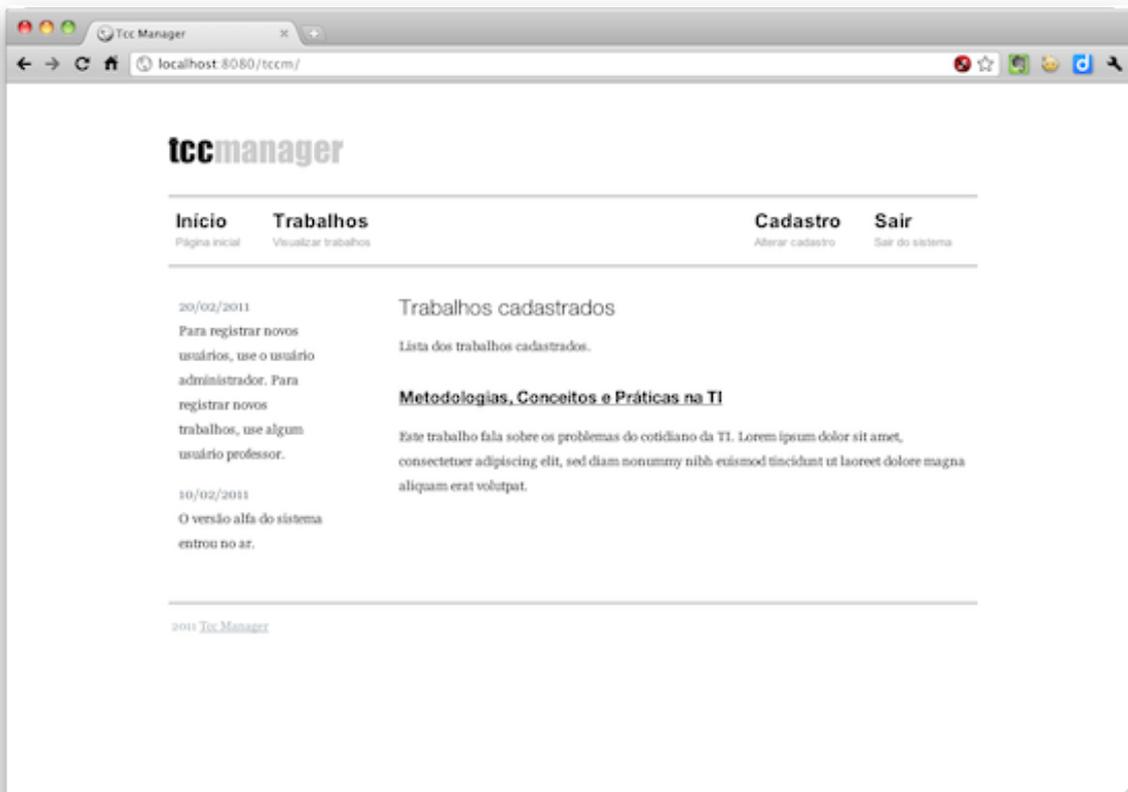


Figura 5.3. Trabalhos do aluno.

Clicando-se no trabalho, podemos ver o detalhes do mesmo. Onde é possível enviar nova versão, ver alguma das versões enviadas e enviar uma versão ao Orientador, Banca ou Bibliotecário.

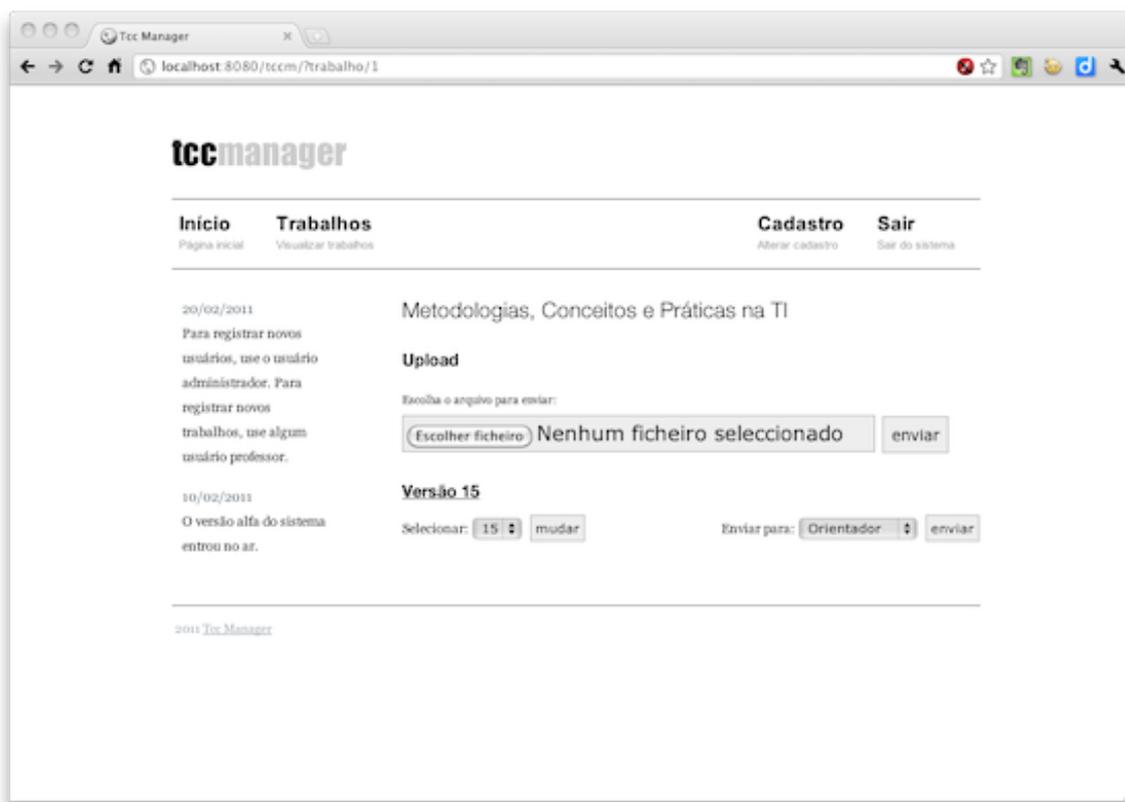


Figura 5.4. Detalhes do trabalho.

5.2.3 Lista de trabalhos

Na tela do professor, podemos ver a lista de trabalho de que ele faz parte, seja como Orientador ou como banca. Clicando-se no trabalho é possível ver as submissões daquele mesmo.

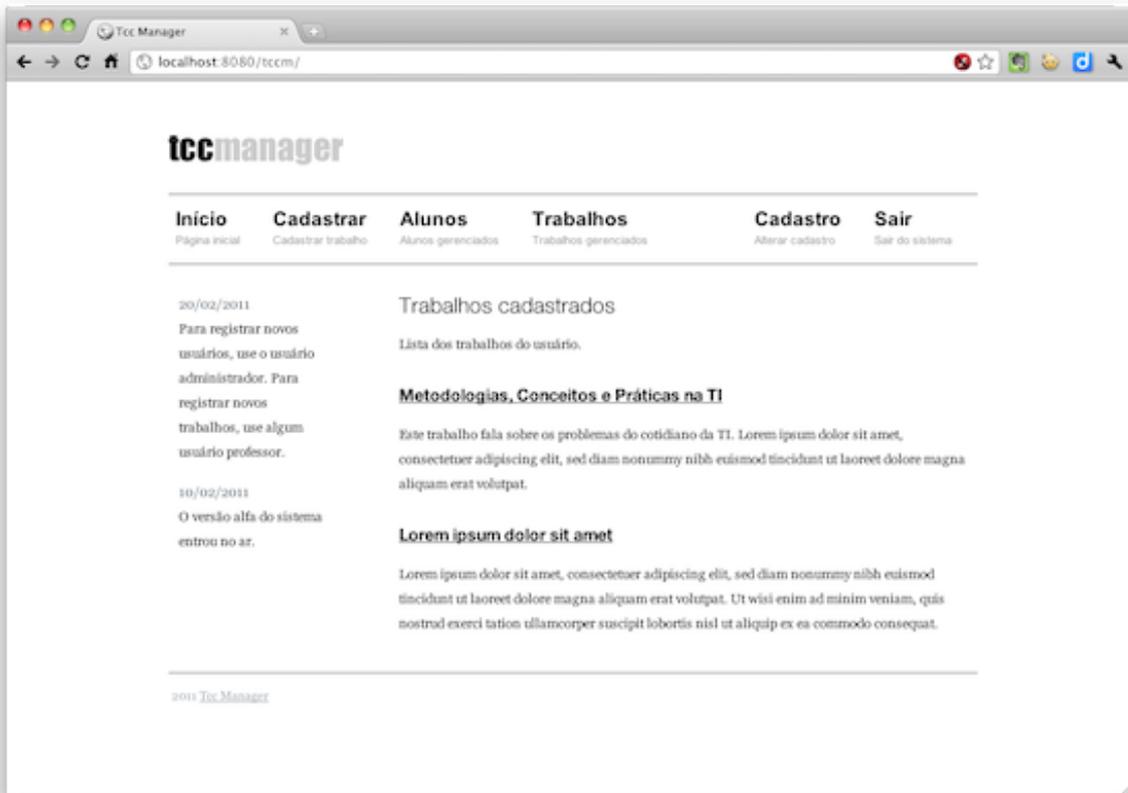


Figura 5.5. Detalhes do trabalho.

5.2.4 Gerência de usuários

Na tela do administrador, é possível ver os trabalhos existentes e é possível cadastrar novos usuários no sistema e gerenciá-los. Este trabalho não é pertinente ao professor, que apenas fará a ligação entre um novo trabalho do qual irá fazer parte com o aluno que irá fazer o trabalho.

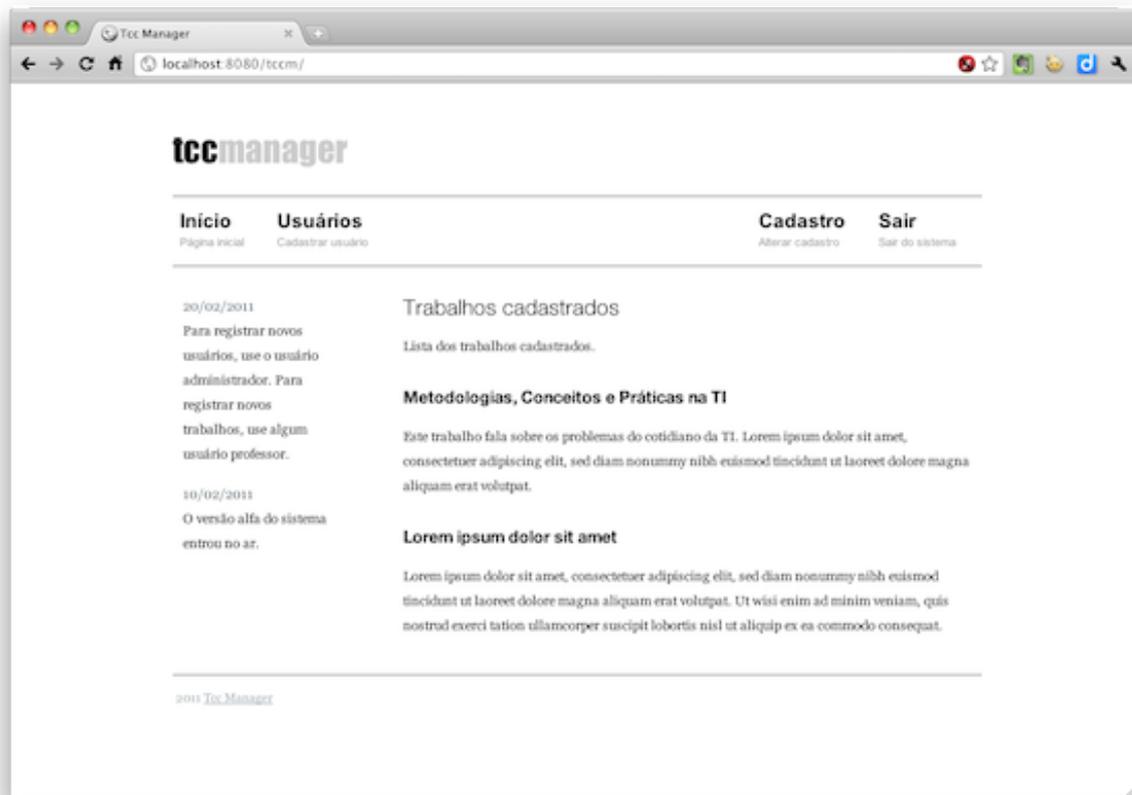


Figura 5.6. Visão geral do administrador.

5.3 Trabalhos futuros

Para trabalhos futuros, seria interessante testar e validar o software em um ambiente de Universidade propício ao desenvolvimento de trabalhos de conclusão de curso. Também, adicionar funcionalidades de menor prioridade mas que agregam valor e trazem além de um apelo de uso, um diferencial sobre outros produtos similares.

Poderia ser feito um estudo no ambiente a fim de levantar funcionalidades interessantes para serem adicionadas no software, bem como funcionalidade que não são muito usadas e portanto podem ser retiradas do mesmo.

Referências Bibliográficas

- Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, us ed edição.
- Beck, K. & Andres, C. (2004). *Extreme Programming Explained: Embrace Change (2nd Edition)*. Addison-Wesley Professional.
- Coelho, D. A. P. & Vilaça, D. (2010). *SCG - Sistema de Gestão de Conferências. ERSS-FEUP. 208*. Manual disponível na Internet em <https://submissoes.sbc.org.br/manual/index.php>. Acesso em 30/08/2010.
- COSTA, B. I. R. & FERNANDES, W. R. (2005). *NORMAS PARA APRESENTAÇÃO DE MONOGRAFIAS*. Belo Horizonte.
- Forte, S. H. A. C. (2004). *Manual de elaboração de Tese, Dissertação e Monografia*. UNIVERSIDADE DE FORTALEZA, Fortaleza.
- Martins, G. d. A. (2010). *Manual para elaboração de monografias e dissertações*. Atlas, 3. ed. edição.
- Paula Filho, W. d. P. (2009). *Engenharia de Software: Fundamentos, Métodos e Padrões*. LTC, Rio de Janeiro, 3a. edição edição.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS E TÉCNICAS (2005). *informação e documentação: trabalhos acadêmicos: apresentação*. Rio de Janeiro, NBR 14724.
- JEMS (2010). *Jornal and Event Management System*. Manual disponível na Internet em <https://submissoes.sbc.org.br/manual/index.php>. Acesso em 30/08/2010.

