

SOFTWARE WEB PARA GERENCIAMENTO E AVALIAÇÃO DE PROPOSTAS DE TCC, PROJETOS DE PESQUISA E ARTIGOS CIENTÍFICOS

Jonis Nogueira dos Santos

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
jonis@faccat.br

Everton Luís Berz

Professor Orientador

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil
everton@faccat.br

Resumo

Este artigo apresenta resultados da pesquisa que teve como objetivo o desenvolvimento de um *software* web que permite controlar o fluxo do processo de revisão e avaliação dos documentos acadêmicos que são gerados por alunos nas disciplinas de conclusão de curso. Com o software é possível eliminar o tráfego dos documentos através de e-mails, permitir ao professor da disciplina selecionar avaliadores para o documento, acompanhar o processo identificando quais avaliadores concluíram ou faltam concluir o processo de revisão e avaliação. Este *software* também traz como inovação a revisão do documento no próprio ambiente da aplicação, permitindo ao avaliador marcar textos e incluir comentários no documento.

Palavras-chave: Gerenciamento de documentos acadêmicos, software web, revisão de artigos

WEB SOFTWARE FOR MANAGEMENT AND REVISION OF PAPERS

Abstract

This article shows the results of a research with the aim to develop a *software* that allows a flow control for papers written by graduating students. Using the software it is possible to eliminate the traffic of these documents through e-mails and allow the course teacher to manage the revision and evaluation progress, choose the referees and identify which evaluation has concluded or will conclude the progress. This *software* also brings an innovation that allows to review the documents in the same environment, enabling the referee to mark texts and include comments in the document.

Key-words: Academic papers management, web software, papers review

1. Introdução

Com a crescente evolução das tecnologias da informação e sua utilização nos meios acadêmicos, torna-se imprescindível o uso de ferramentas de apoio as atividades, tais como, o gerenciamento do tráfego de documentos (Artigos científicos, monografias e projetos de TCCs) desenvolvidos por alunos em fase de conclusão de curso e que são enviados para serem revisados e avaliados.

Há uma grande dificuldade de controlar o tráfego dos documentos enviados por alunos e avaliadores por e-mail, também é comum os alunos enviarem os documentos mais de uma vez, o que acaba dificultando ainda mais este controle.

Quanto ao retorno dos documentos revisados pelos avaliadores, cabe lembrar que estes podem estar revisando documentos de um ou mais alunos, o que também dificulta ao professor controlar de quem é o documento que está retornando

Outra dificuldade encontrada neste processo de controle manual é ter que verificar no documento que está sendo recebido quem é o aluno e identificar quem é o orientador deste aluno, pois na hora de direcionar o documento para avaliação o orientador não poderá ser uma opção de envio por conta do *blindreview*.

O metodo *blindreview* permite que os nomes dos autores e avaliadores sejam mantidos em sigilo. O documento é submetido para avaliação podendo ser aceito, aceitável com revisões, ou rejeitado.

Mesmo depois de enviado existe a necessidade de ter controle de qual documento foi enviado para qual ou quais avaliadores. No momento do retorno destes documentos existem outros problemas, tais como: (i) ter o controle de quais avaliadores já concluíram a avaliação e já enviaram o arquivo de avaliação; (ii) quais ainda estão pendentes de envio. Quando isto é feito de forma não automatizada, requer maior atenção e tempo gasto no processo.

Outro problema identificado é a falta de uma padronização dos aplicativos usados para avaliação destes documentos, o que acaba sendo ruim pois os editores que são utilizados atualmente permitem a edição do arquivo, podendo causar alterações indesejáveis no conteúdo, ou na formatação do documento, que também faz parte da revisão e avaliação.

Para resolver estes problemas, se propôs a criação de um software web, que visa facilitar o gerenciamento dos arquivos de trabalhos acadêmicos produzidos por alunos em fase de conclusão de curso, bem como revisá-lo e avaliá-lo no mesmo ambiente, o que é um diferencial do trabalho. O gerenciamento em um único ambiente, permite ter um fluxo confiável, eficiente e organizado das informações.

Este artigo apresenta os resultados da pesquisa e desenvolvimento de uma ferramenta que visa facilitar o controle de documentos acadêmicos enviados por alunos em fase de conclusão de curso, eliminando a necessidade de enviar os documentos por e-mail, e permitindo que os mesmos sejam revisados e avaliados no mesmo ambiente na web.

O trabalho possui a seguinte estrutura: a seção 2 o referencial teórico, a seção 3 apresenta a metodologia, a seção 4 os resultados obtidos e a seção 5 traz as conclusões da pesquisa e desenvolvimento.

2. Referencial Teórico

Segundo Ferreira *et al.* (2007), desde 1999, várias ferramentas (tanto de provedores de dados como de serviços) começaram a ser desenvolvidas em universidades americanas, canadenses e europeias, todas elas voltadas à organização e gestão da informação interoperáveis, aberta e aplicáveis a negócios que vão desde gerenciamento de revistas científicas, de eventos acadêmicos e de repositórios institucionais e/ou temáticos.

No Brasil, o IBICT desde 2001 vem trabalhando na criação e customização de sistemas de informação para o uso pelas instituições brasileiras, tais como: Biblioteca Digital de Teses e Dissertações, Diálogo Científico (DICI/Eprints), Sistema Eletrônico de Editoração de Revistas (Seer/OJS), Sistema de Publicação Eletrônica de Teses e Dissertações e Sistema On-Line de Apoio a Congressos (Soac/OCS). (FERREIRA *et al.*, 2007).

2.1 Sistema Eletrônico de Editoração de Revistas (SEER/OJS)

O *Open Journal Systems/OJS*, desenvolvido pelo *PublicKnowledge Project* da Universidade *British Columbia*, Canadá, foi traduzido para o português com o nome de Sistema Eletrônico de Editoração de Revistas – SEER, pelo Instituto Brasileiro de Informação em Ciência e Tecnologia. Tem como objetivo principal a construção e gestão de publicações periódicas eletrônicas. Atualmente, existem mais de 210 revistas brasileiras utilizando-o. (FERREIRA *et al.*, 2007)

O SEER, é resultado da prospecção tecnológica realizada pelo IBICT¹ para identificar aplicativos que possibilitassem o tratamento e a disseminação da produção científica brasileira na web, O sistema SEER surgiu em 2003 a partir da customização do *Open Journal Systems* (OJS). (SEER, 2011).

¹ IBICT: Instituto Brasileiro de Informação em Ciência e Tecnologia

O SEER/OJS, é um sistema voltado a gestão, submissão e editoração de revistas, as submissões seguem um fluxo onde temos claramente as figuras dos atores, (Autor, Editores e Avaliadores), cada ator com funções diferentes dentro do sistema, como pode ser visualizadas na Figura 1.

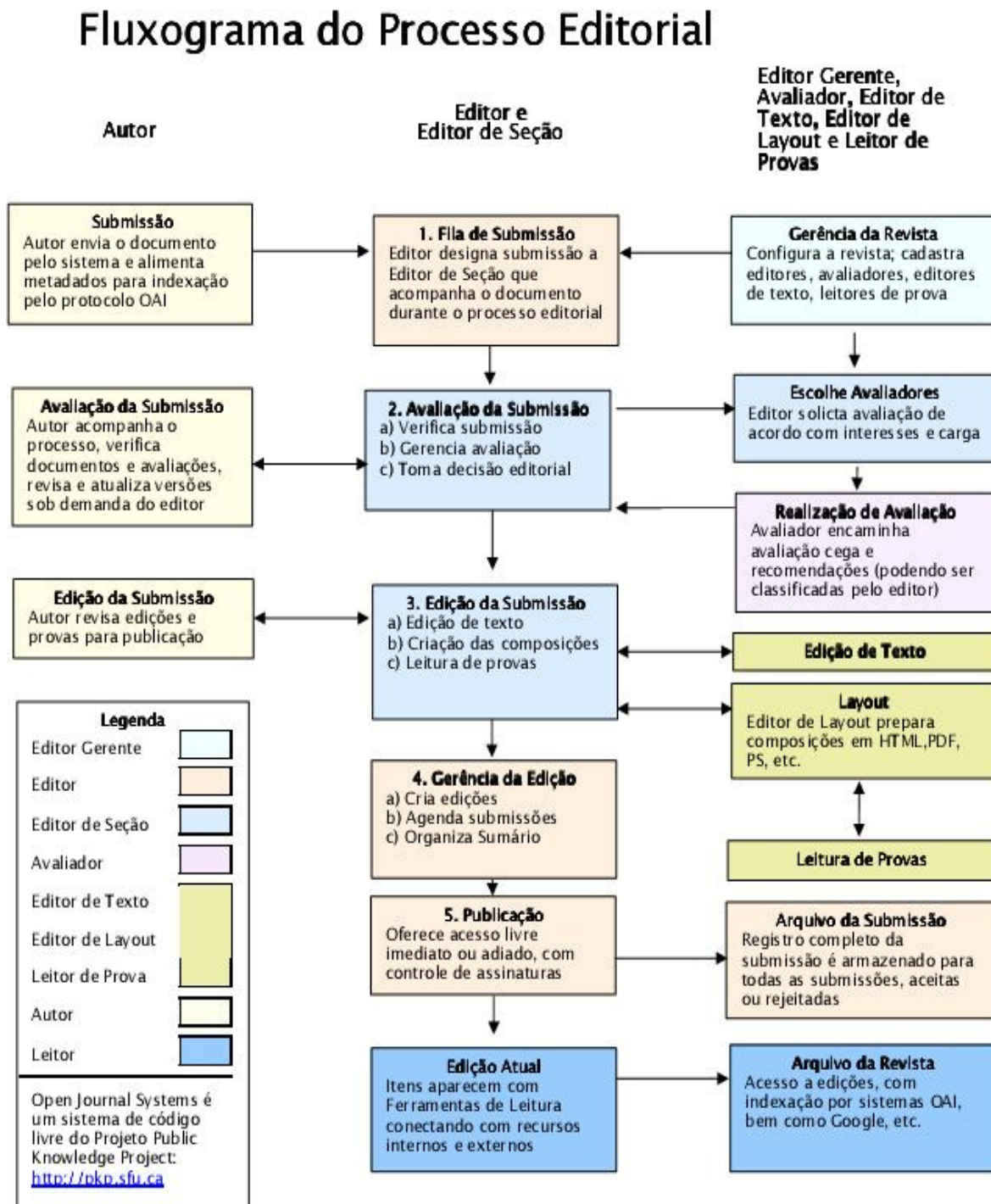


FIGURA 1: Fluxo do processo editorial
 Fonte: Fonseca (2006, p.11)

2.2 Eprints/DICI

O *EPrints/DICI* (Diálogo Científico) é um software livre que favorece a comunicação entre professores e pesquisadores, foi traduzido e customizado pelo Instituto Brasileiro de Informação em Ciência e Tecnologia. Trata-se de um espaço virtual, disponível na internet, para registro e discussão de textos completos produzidos por pesquisadores que atuam nas áreas de ciência e tecnologia.

O *EPrints/DICI* serve para dinamizar o processo de comunicação entre pesquisadores que atuam nas áreas de ciência e tecnologia, sendo destinado à criação de repositórios temáticos. Ele funciona por meio de mecanismos de publicação web gerenciados pelas próprias comunidades, utiliza tecnologia Internet com alto grau de padronização, o que permite a interoperacionalidade com outros sistemas nacionais e internacionais da mesma natureza. Além de um maior acesso à produção científica, proporciona a sua preservação digital, tornando o processo de produção do conhecimento científico mais transparente, dinâmico e democrático.

2.3 Soac/OCS

O *Open Conference System/OCS* foi desenvolvido pela Universidade de British Columbia, no Canadá, e traduzido para o português com o nome de Sistema *On-line* de Apoio a Congressos – SOAC, pelo Centro de Informações Nucleares da Comissão Nacional de Energia Nuclear – CIN/CNEN com o apoio do Instituto Brasileiro de Informação em Ciência e Tecnologia (Ibict). Tem como objetivo auxiliar, via web, os processos de planejamento, organização e gestão de conferências, assim como na publicação dos respectivos anais. Inúmeros congressos e seminários brasileiros vêm utilizando esse *software*, em especial aqueles da área de ciência da computação. (FERREIRA *et al.*, 2007)

2.4 GNP Sistema de gestão de publicações

O GNP é uma ferramenta de gestão de publicações mantida pela Genesis Network, esta ferramenta permite que autores criem artigos *on-line* para serem avaliados, os principais atores neste cenário são os autores que submetem os artigos, editores que são responsáveis por analisar os artigos, devolver ao autor caso não esteja nos padrões científicos, ou direcionar para os avaliadores que por sua vez, revisão e comentam as publicações. (GN1 GENESIS NETWORK, 2012).

Esta ferramenta não necessita uploads ou download de arquivos, uma vez que as publicações são criadas no próprio ambiente, o autor tem a possibilidade de copiar e colar textos e planilhas do word no momento que esta criando a publicação no *site*, outra informação importante é que esta

ferramenta tem um custo que é atribuído a sua manutenção e upgrades. (GN1 GENESIS NETWORK, 2012).

3. Metodologia

A partir da problematização, análise dos requisitos e ferramentas disponíveis, foi desenvolvido um *software* web para auxiliar no controle do fluxo dos documentos, bem como permitir que as revisões e avaliações possam ser executadas dentro do mesmo ambiente do sistema. Tal *software* foi denominado SGDA (Sistema Gerenciador de Documentos Acadêmicos).

Para o desenvolvimento do *software* foi utilizado o modelo incremental, segundo Pressman (2005) este modelo combina elementos do modelo sequencial linear com a filosofia interativa da prototipagem. Este modelo assume que o *software* desenvolvido poder sempre crescer e agregar novas funcionalidades, sendo que cada uma dessas funcionalidades, ou o conjunto delas, será desenvolvido por um incremento e esse segue todas as etapas descritas no modelo cascata. Desta forma o desenvolvimento foi dividido em análise de requisitos, projeto, implementação e teste.

O primeiro incremento deste modelo é denominado de núcleo do produto, pois nele estarão os requisitos básicos do sistema. Este modelo é um melhoramento do modelo em cascata pois permite a alteração dos requisitos durante o desenvolvimento do *software*. A Figura 2 apresenta o modelo incremental na versão que foi utilizado neste trabalho.

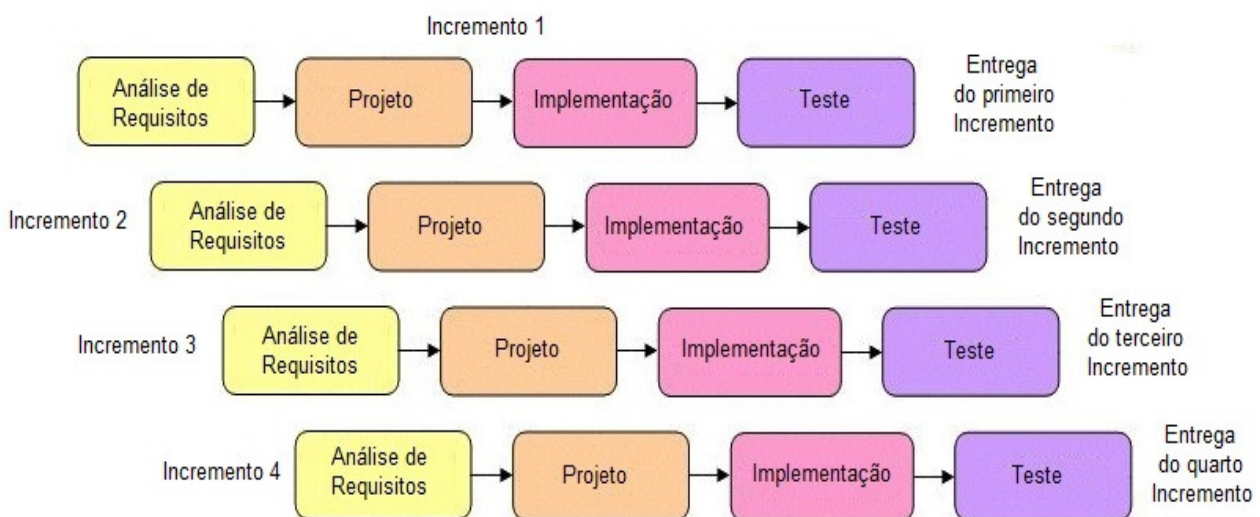


FIGURA 2: Modelo incremental
Fonte: Pressman (2002, p.33)

3.1 Análise de requisitos

Nesta fase de análise foram coletados os requisitos necessários para o desenvolvimento do *software*, que segundo Pressman (2002) permite obter informações que resultam na especificação das características operacionais da aplicação.

A análise de requisitos iniciou com entrevistas realizadas com o Prof. Dr. Carlos Fernando Jung, coordenador do curso de Engenharia de Produção e professor do curso de Sistemas de Informação das Faculdades Integradas de Taquara (FACCAT).

A partir das entrevistas, identificou-se o formato de como é feito o tráfego dos documentos no modelo atual e observou-se a dificuldade que o professor tem para controlar o tráfego dos documentos de alunos e avaliadores, controle este que é feito através de planilhas eletrônicas. Com isto, verificou-se a necessidade de um *software* que auxiliasse no controle do tráfego de documentos, entre alunos e professor, professor e avaliadores, e vice-versa. A Figura 3 exibe o fluxograma de como é o processo atual de envio dos documentos.

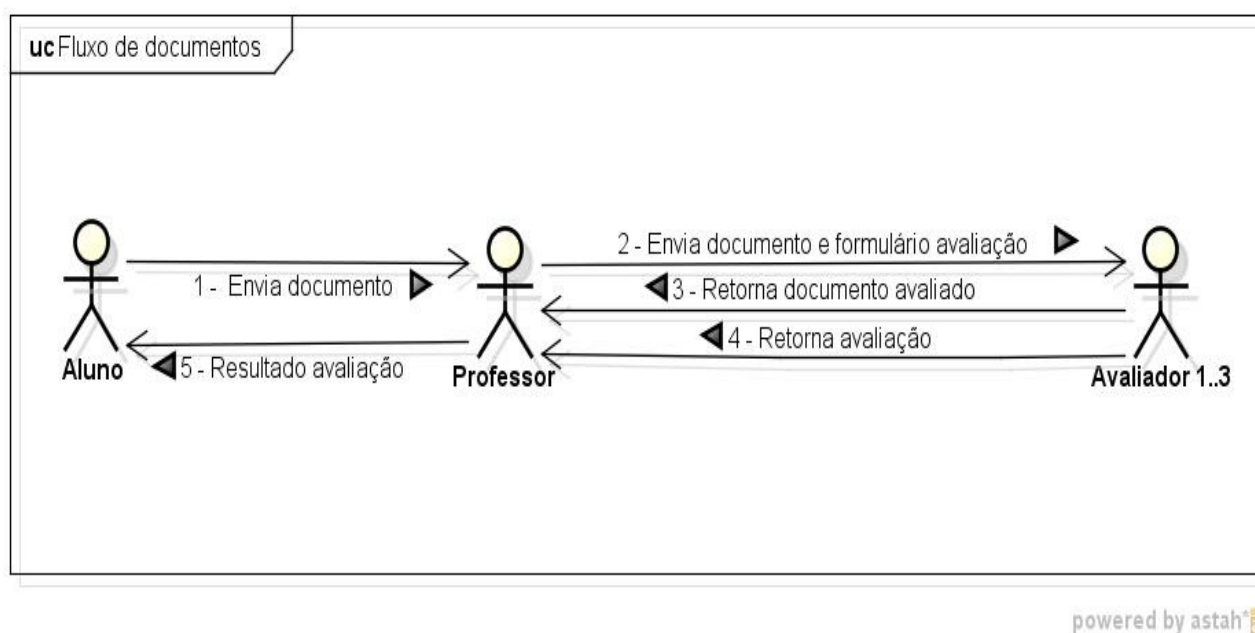


FIGURA 3: Fluxograma – Tráfego de documentos no modelo atual.

Durante a fase de coleta de dados e análise foram identificados os requisitos relevantes para que o *software* a ser desenvolvido pudesse suprir os todas as funções necessárias para solução do problema verificado durante as entrevistas. Desta forma os requisitos foram categorizados em requisitos funcionais, mostrados no quadro 1 e requisitos não funcionais, conforme mostrados no quadro 2.

Quadro 1 - Requisitos funcionais

Requisito	Descrição
Autenticação	Disponibilizar interface de autenticação para que o sistema possa identificar o usuário que esta acessando
Upload documentos	Disponibilizar interface para <i>upload</i> de documentos somente em formato PDF (Portable document format)
Controle de versões	Gerenciar versões de documentos postados, sequenciando-o quando houver postagem extra
	Gerar uma cópia do documento para cada avaliador selecionado e mantendo a cópia do aluno íntegra
Mensagens de aviso (envio de e-mails)	Para o administrador quando houver nova postagem de documentos
	Para o avaliador quando um documento for direcionado para ele avaliar
	Para o administrador quando um avaliador concluir a revisão do documento
	Para o administrador quando um avaliador concluir a avaliação do documento
	Para o aluno quando o administrador disponibilizar a avaliação do documento
Revisão de documento	Disponibilizar interface para revisão dos documentos que permita inclusão de anotações no texto
Avaliar documento	Disponibilizar interface para que o avaliador possa avaliar o documento através de nota que pode variar de 0 a 10
Pesquisa de documento	Disponibilizar interface de pesquisa de documentos para o administrador

Fonte: Própria

Quadro 2 - Requisitos não funcionais

Requisito	Descrição
Web	O <i>Software</i> deve ser desenvolvido para ser utilizado via internet, possibilitando ser acessado de qualquer lugar através de um navegador web
Acessos	Permitir que somente de usuários cadastrados e com status “ativo” possam acessar o sistema
	Restringir acesso as funcionalidades de acordo com o tipo de usuário que esta acessando o sistema
Documento	Permitir que somente documentos em formato PDF possam ser postados
	Não permitir que o conteúdo original do documento seja alterado

Fonte: Própria

3.2 Projeto

Para criar os diagramas da aplicação na linguagem UML², foi utilizada a ferramenta Astah Community, (CHANGE VISION, 2011). Foram criados os diagramas de casos de uso, diagramas de sequencias, diagrama ER³ e diagramas de classes, que demonstram a estrutura da aplicação e as funcionalidades do sistema. Um exemplo destes diagramas pode ser visto na Figura 4, que traz o diagrama de caso de uso geral mostrando como cada ator pode interagir com a aplicação.

²UML: Unified modeling language

³ ER: Entidade Relacionamento

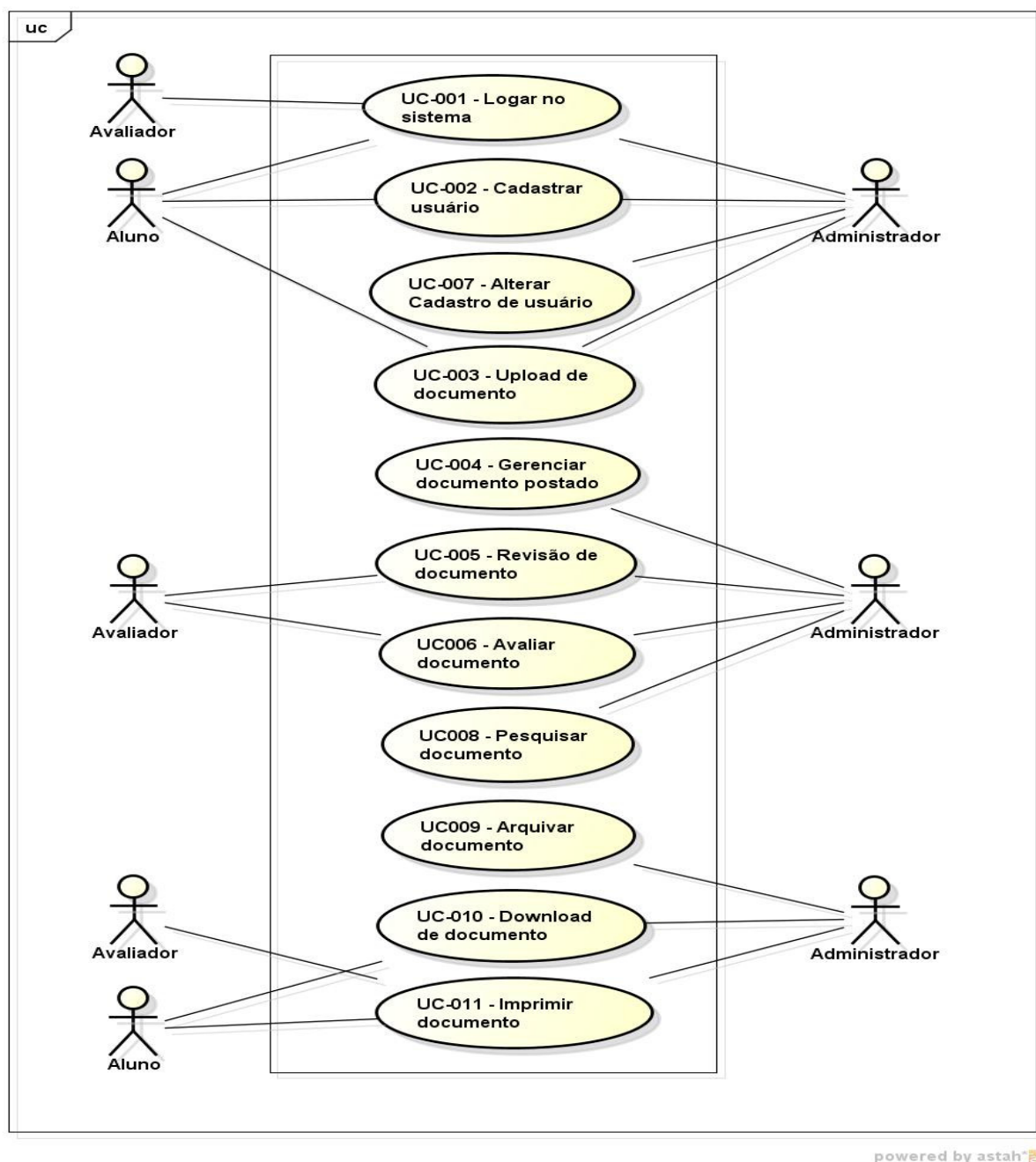
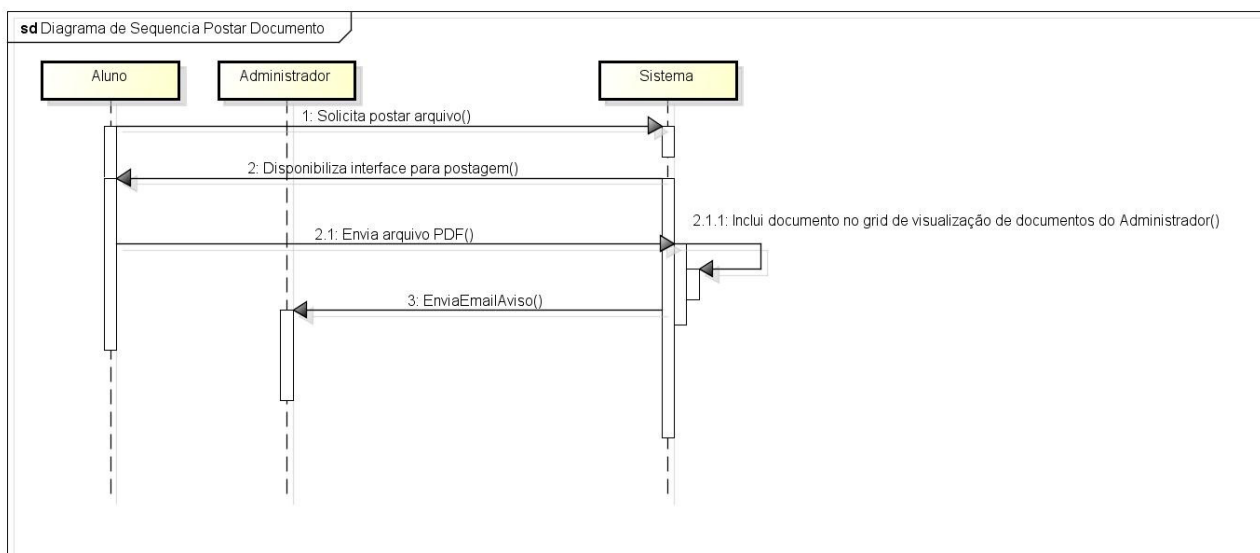


FIGURA 4: Caso de uso geral

Entre os requisitos do projeto inclui-se a funcionalidade que permite aos alunos postarem seus documentos no ambiente do aplicativo (UC-003), com o objetivo de eliminar o tráfego destes documentos através de e-mails. Após a postagem o sistema envia um e-mail de aviso ao administrador informando que um arquivo foi postado. A Figura 5 mostra o diagrama de sequência da postagem de documentos dos alunos.



powered by astah®

FIGURA 5: Diagrama de sequência postar documento

Fonte: própria

Uma vez que o documento tenha sido postado pelo aluno, este documento fica visível na interface de visualização de documentos do administrador, deixando-o apto a selecionar o(s) avaliador(es) para o documento. A sequência deste processo é exibido na Figura 6.

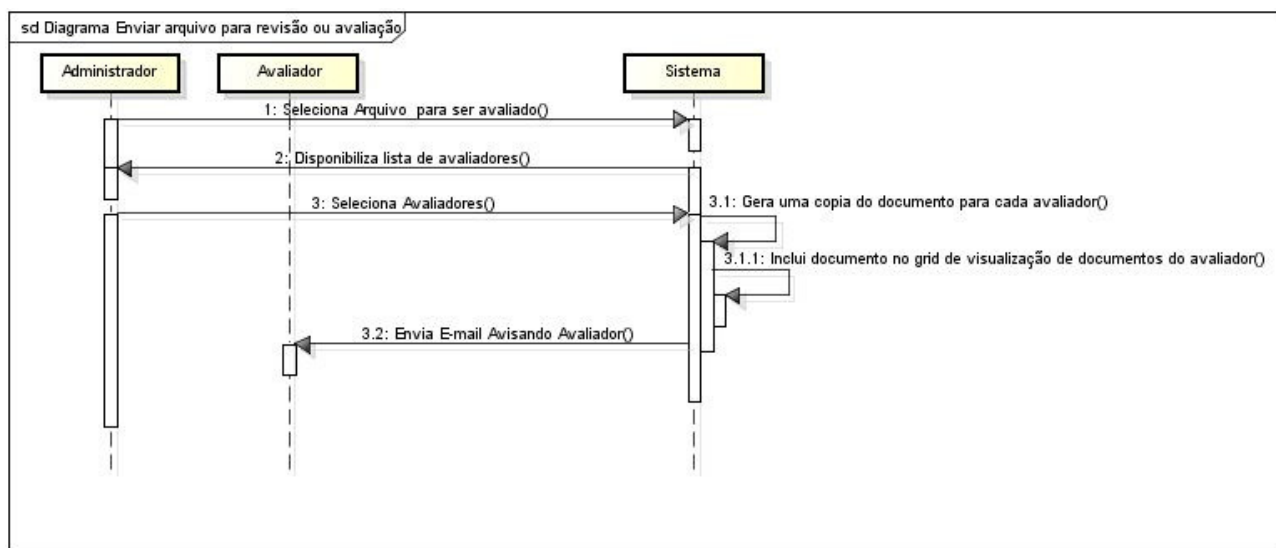


FIGURA 6: Diagrama de sequência enviar documento para revisão e avaliação

Fonte: própria

O Quadro 3 mostra o Caso de uso expandido da operação mostrada acima, onde o administrador executa a operação de selecionar avaliadores para um determinado documento, o sistema gera uma copia para cada avaliador e envia e-mail informando o avaliador de que há um documento para ser avaliado..

Quadro 3 - Diagrama de Caso de Uso – Selecionar avaliadores para documento

Nome do Caso de Uso	Selecionar avaliadores para documento
Caso de Uso	UC008 – Selecionar avaliadores para documento
Atores	Administrador, Avaliador, Sistema
Descrição	O administrador do sistema verifica que há um documento para ser enviado para revisão e avaliação e clica no documento, sistema disponibiliza lista de avaliadores disponíveis. Administrador seleciona os avaliadores e confirma a operação.
Tipo	Primário
Referência	RF008
Fluxo principal	
Ações do ator	Ações do sistema
1. Administrador acesso interface de visualização de documentos e verifica que há documento a ser enviado para revisão e avaliação.	
2. Administrador clica no título do documento para selecionar os avaliadores.	3. Sistema disponibiliza lista de avaliadores disponíveis para revisão e avaliação do documento.
4. Administrador seleciona os avaliadores para o documento e confirma a operação.	5. Sistema grava no banco de dados a informação de quem são os avaliadores do documento. 6. Gera cópia do arquivo original para cada avaliador. 7. Envia e-mail de aviso informando o avaliador que há um documento para ser revisado e avaliado. 8. Retorna para interface de visualização de documentos e aguarda próxima requisição do administrador.

Fonte: Própria

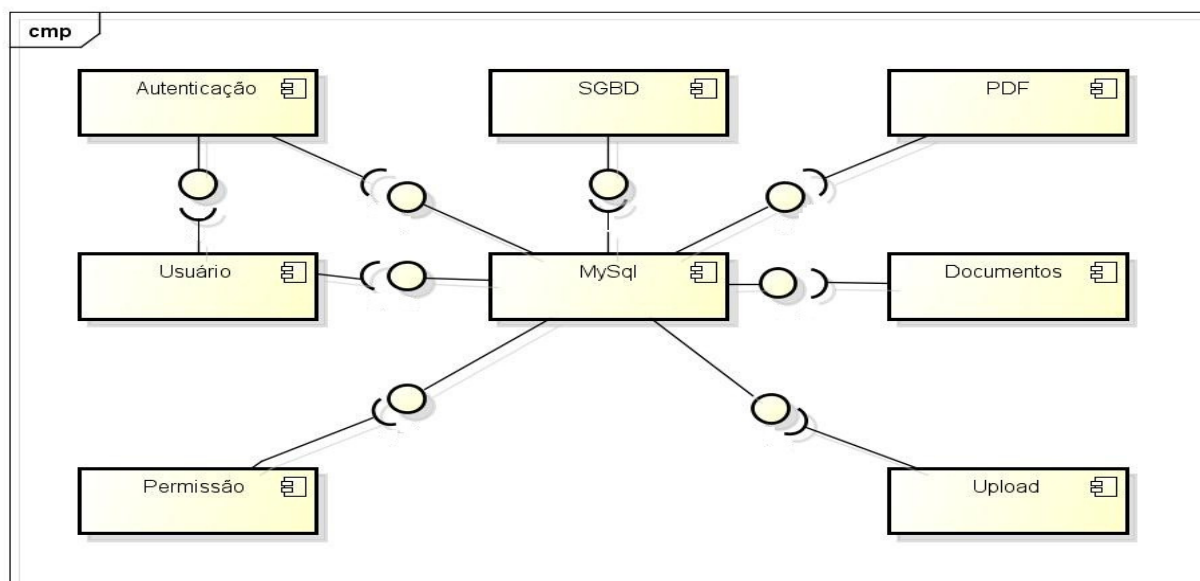
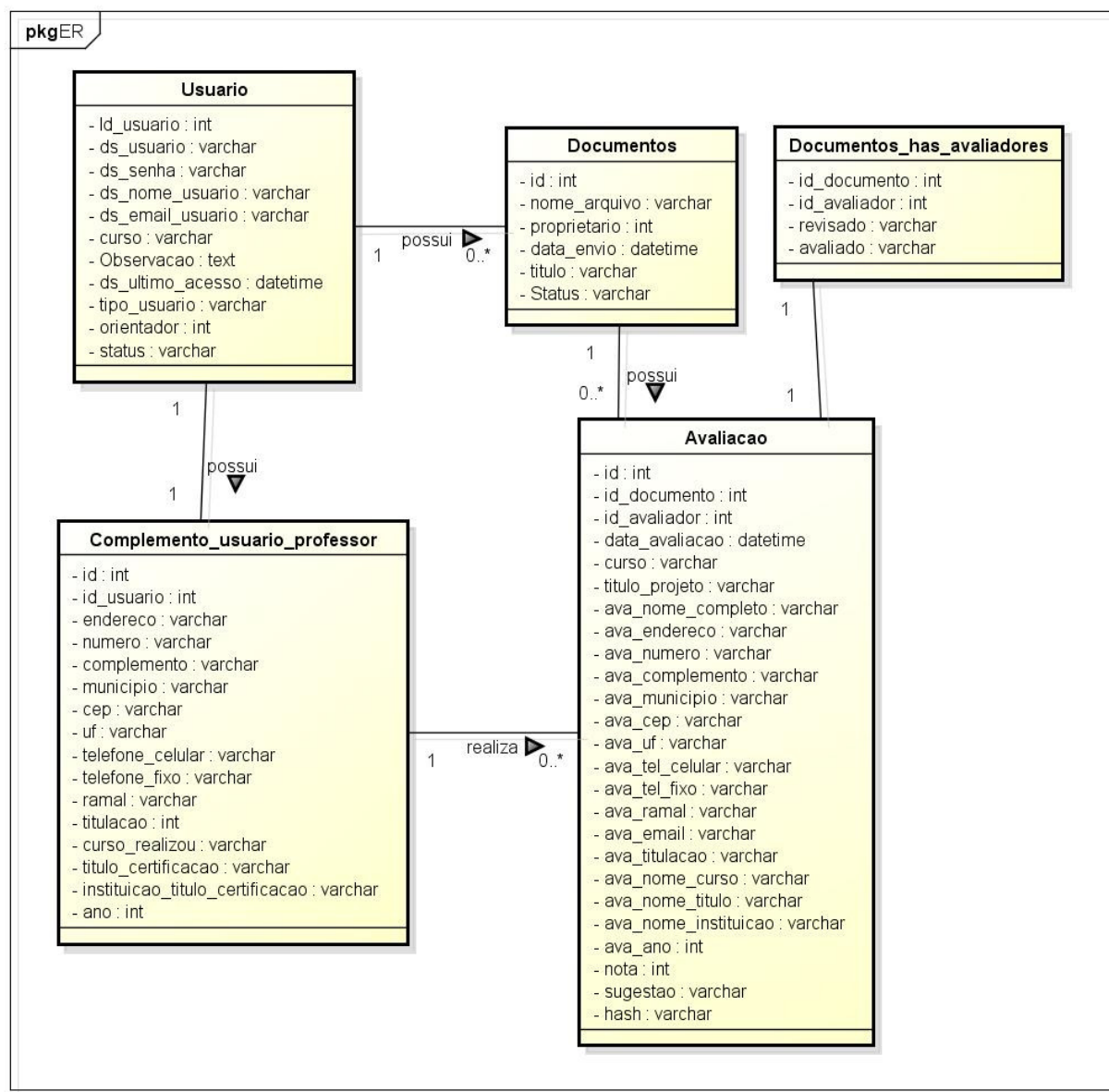


FIGURA 7: Diagrama de componentes internos – Sistema SGDA

Fonte: própria

Outro artefato construído nesta fase foi o diagrama de componentes internos. Cabe observar que o componente MySQL tem uma interface requerida com o componente SGBD⁴, e este, uma interface fornecida com o componente MySQL, e assim ocorre entre o componente MySQL e os outros componentes da aplicação, um componente requerendo uma informação e outro fornecendo. Como pode ser visto na Figura 7.

A Figura 8 exibe o diagrama ER do banco de dados da aplicação, suas tabelas e como elas se relacionam entre si.



powered by astah

FIGURA 8: Diagrama ER
Fonte: própria

3.2.1 Arquitetura

Durante esta etapa foram verificados e testados algumas APIs de editores *on-line* que possibilitassem a revisão dos documentos em formato PDF (*Portable document format*,) e que fosse possível agregar esta ferramenta ao *software* que estava sendo projetado.

⁴ SGBD: Sistema gerenciador de banco de dados

Alguns plugins como por exemplo o TinyMCE e o CKEditor, que possuem a funcionalidade de permitir anotações em documentos foram analisados, mas não tiveram o resultado esperado, além de difícil integração, também permitem que o texto seja alterado, o que inflinge um dos requisitos do sistema que é não permitir alteração no texto original, e sim, apenas efetuar anotações.

Nos testes realizados para encontrar o editor que melhor atendesse aos requisitos propostos para o projeto, foi escolhido utilizar a API⁵ CrocoDoc, esta API permite que um documento em formato PDF⁶ possa ser visualizado em sua interface e possibilita a inserção de marcas de textos e anotações. (CROCODOC, 2012).

A Figura 9 mostra a estrutura de pastas da aplicação conforme ela foi arquitetada, sendo que as principais são: na pasta raiz denominada SGDA estão os arquivos de inicialização e configuração de ambiente, tais como index.php, common.php e .htaccess, nas subpastas temos a pasta de projeto que é onde esta toda documentação criada durante as fases de análise e projeto, a pasta arquivos é a pasta utilizada para armazenar os arquivos postados pelos alunos, bem como as cópias geradas para avaliação, a pasta classes onde estão todas classes criadas durante o processo de desenvolvimento e a pasta *pages* onde estão os formulários (*views*) da aplicação.

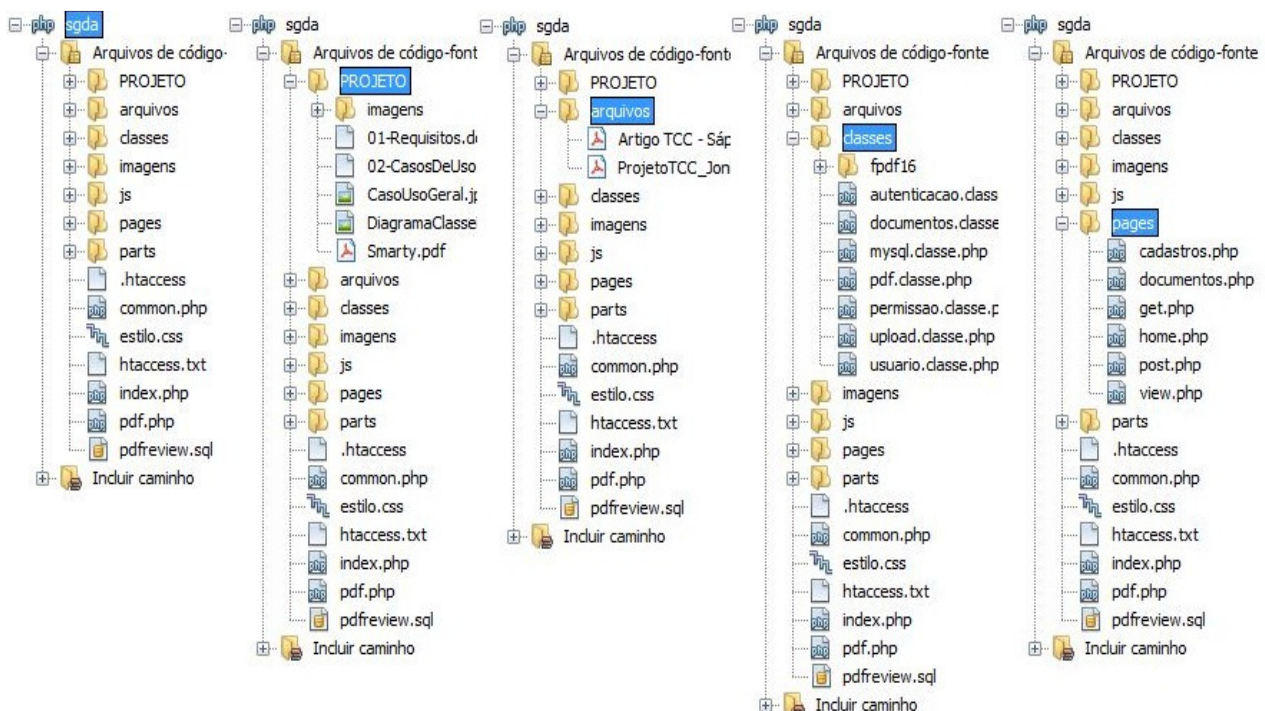


FIGURA 9: Estrutura de pastas da aplicação
Fonte: própria

⁵API: Application program interface, ou interface de programação de aplicativos

⁶PDF: Portable document format, ou formato de documento portátil

3.3 Desenvolvimento

Para o desenvolvimento do *software* foi utilizado tecnologias *open sources*, tais como: HTML, PHP 5 (W3C, 2011), Javascript, Servidor Apache (APACHE, 2011), banco de dados MySQL, NetBeans IDE. Também foi utilizado a API CrocoDoc que possibilita agregar ao *software* a funcionalidade de edição de documentos em formato PDF para inserção de comentários.

Por se tratar de uma aplicação web, este *software* é multiplataforma, com isto, o usuário não precisa preocupar-se com o sistema operacional que está rodando em seu equipamento, basta apenas possuir um navegador web para conseguir acessá-lo, não sendo necessário instalar *plugins*⁷ ou outros aplicativos.

3.3.1. PHP 5

O PHP⁸ é uma das linguagens de programação que mais se difundiu para aplicações *web* em todo mundo. Milhões de sites utilizam PHP, sendo que a principal diferença em relação as outras linguagens é a capacidade de interagir com o mundo *web*. (NIEDERAUER, 2011)

Niederauer (2011) lembra que além do PHP ser gratuito, é um *software* de código fonte aberto. Outra vantagem é o fato de ser executado no servidor. Ou seja, quando uma página PHP é acessada por meio do navegador, todo código PHP é executado no servidor e os resultados enviados para o navegador que exibe a página já processada, não consumindo recursos de processamento local.

Minetto (2007) comenta que o PHP possui uma grande vantagem com relação a facilidade de aprendizado. Um programador é capaz de aprender com a leitura de poucas páginas de tutoriais ou de algum livro. Com isto, favoreceu-se o rápido aumento no número de programadores e o surgimento de grandes *softwares*.

O PHP foi escolhido para ser utilizado no desenvolvimento deste *software* por ser uma das linguagens mais utilizadas para o desenvolvimento na *web*, por ser de fácil aprendizado, e ter a sintaxe semelhante a linguagens já conhecidas pelo autor.

3.3.2. Javascript

Segundo Flanagan (2002), *Javascript* é uma linguagem de programação leve, e com recursos de orientação a objetos. O núcleo de uso geral da linguagem foi incorporado no Internet Explorer e em outros navegadores *web* e aprimorado para programação de *websites*. Isto significa que uma

⁷Plugin: Programa de computador usado para adicionar funções a outros programas maiores.

⁸ PHP: Hypertext Preprocessor

página *web* não precisa mais de HTML estático, mas pode incluir programas que interagem com o usuário, controlam o navegador e criam conteúdo HTML dinamicamente.

No software desenvolvido foi utilizado *Javascript* para dar dinamismo em algumas funções do sistema que necessitam de interação com o usuário.

3.3.3. Servidor Apache

O Servidor Apache é o mais bem sucedido servidor *web* livre, foi criado em 1995 pela NCSA (*National Center for Supercomputing Applications*), mais tarde teve seu desenvolvimento continuado pela *Apache Software Foundation*.

Em pesquisa realizada em 2010, foi constatado que o Apache serviu mais de 54,68% de todos os sites e mais de 66% dos milhões de sites mais movimentados. (APACHE, 2011)

Segundo Magrin (2006), o servidor web Apache é o mais utilizado do mundo. Isso se deve ao fato de tratar-se de um servidor de código-fonte aberto, além de ser totalmente configurável, adaptando-se a praticamente todos os tipos de necessidades.

3.3.4. MySQL

Um dos mais populares SGBDs (Sistema Gerenciador de Banco de Dados) Open Source, criada na década de 1980 pela MySQL AB, foi adquirida pela Sun Microsystems em 2008. Amplamente utilizado em aplicativos web, seu sucesso deve-se em grande parte a fácil integração com o PHP, além disto, é de fácil usabilidade e possui uma ampla biblioteca de livros e manuais o que facilita o aprendizado da ferramenta.

Justamente por ser amplamente utilizado na web e ser de fácil usabilidade é que o MySQL foi utilizado para criação do banco de dados, bem como das tabelas necessárias para o armazenamento das informações incluídas no SGDA.

3.3.5. Crocodoc

O Crocodoc é uma aplicação *online* que permite editar documentos em formato PDF para incluir anotações e marcações no texto. Esta funcionalidade pode ser agregado a um site pessoal. Assim, a API do Crocodoc foi agregada ao SGDA, permitindo que os documentos acadêmicos possam ser visualizados e que os avaliadores possam incluir comentários ou marcações de texto. A Figura 10 mostra um exemplo de anotações e marcações em documentos PDF.

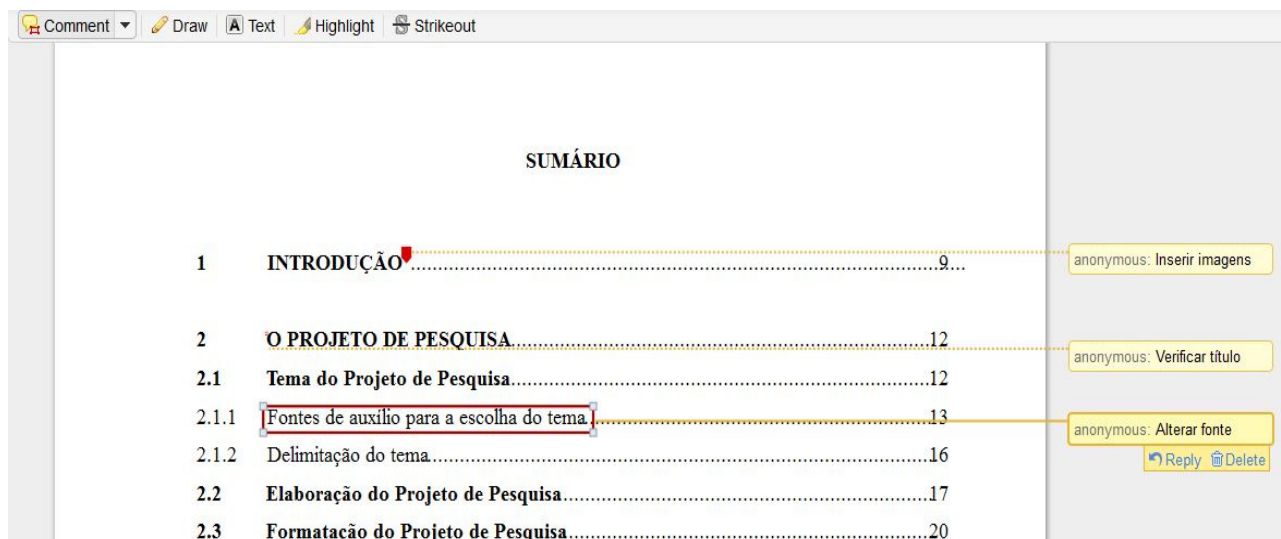


FIGURA 10: Exemplo de inclusão de comentário e marcações em documentos PDF.
Fonte: Própria.

A integração entre o SGDA e o Crocodoc é feita através de *tags* onde o SGDA faz uma chamada a API do Crocodoc, passando o endereço e o nome do documento a ser editado, nesta mesma chamada também é informado o Token que é a chave de registro do SGDA no Crocodoc. Esta tag pode ser verificada no exemplo mostrado no Quadro 4.

Quadro 4 – Exemplo de comunicação entre o SGDA e o *plugin* do Crocodoc

```
echo "<a target='_blank'
href='https://personal.crocodoc.com/api/v1/document/upload?url=https://fit.faccat.br/~jonis/sgda/arquivos/Smarty.pdf&token="cwYMMxVXFqglBish20Ub"'>"
```

Fonte: Própria

3.3.6. NetBeans IDE

Para o desenvolvimento do software foi escolhido utilizar o NetBeans IDE (NetBeans.org, 2012), pela facilidade de aprendizagem do uso da ferramenta, que possui vasta documentação, tutoriais e exemplos na web. O NetBeans é um projeto open source de sucesso fundado pela Sun Microsystems, com uma grande base de utilizadores e uma crescente comunidade mundial.

A utilização desta ferramenta auxiliou no desenvolvimento do *software*, com recursos de autocompletar, indentação do código, organização de pastas, recursos de pesquisa, e recuperação de códigos, identificação de erros de escrita e sintaxe, ou seja, facilitou o desenvolvimento com sua interface e funcionalidades amigáveis

3.4 Testes

Segundo Pressman (2002), o teste caixa-preta tenta encontrar erros das seguintes categorias:

(1) funções incorretas ou omitidas, (2) erros de interface, (3) erros de estrutura de dados ou de acesso a base de dados externa, (4) erros de comportamento ou de desempenho e (5) erros de inicialização e término. Na fase final de cada etapa do processo de desenvolvimento, foram realizados testes utilizando o método “caixa-preta”, com simulações executadas na aplicação a fim de verificar se o aplicativo estava efetivamente comportando-se de acordo com o que havia sido programado. Os seguintes testes foram realizados no decorrer do projeto:

- a) Teste de inclusão de usuário: foram feitos testes de inclusão de usuários do tipo “aluno”, “professor” e “administrador”, cada tipo tem permissões de visualização e de acessos diferentes, de acordo com a necessidade de cada ator verificada durante a fase de análise.
- b) Teste de postagem de documentos: foram feitos testes de inclusão de documentos com usuários do tipo “aluno”, durante os testes foi validado a inclusão do documento no sistema, disponibilização da visualização do documento por parte do aluno e do professor.
- c) Teste de direcionamento para revisão e avaliação: durante estes testes foram verificados a criação de um arquivo unico para cada avaliador selecionado, bem como a disponibilização da deste documento na interface de visualização de documentos do avaliador.
- d) Tese de revisão de documento: neste processo foi testado a inclusão de comentários e marca de textos nos documentos PDF, e a finalização da revisão que é feita através de um link na página de visualização do documento.
- e) Tese de avaliação de documento: foi testado a geração do formulário de avaliação, que já traz carregado a identificação do avaliador, tais como: Nome, Titulação, Nome do título ou certificação, Instiuição que conferiu o título ou certificação, e disponibiliza o campo para que o avaliador possa informar a nota dada ao documento.

4. Resultados

A pesquisa e desenvolvimento teve como resultado um *software* web denominado SGDA. Este *software* permite atender os objetivos propostos no projeto de desenvolvimento, que são: (i) Eliminar o tráfego de documentos acadêmicos através de e-mails, disponibilizando uma interface para que os alunos possam submeter estes documentos, (ii) disponibilizar interfaces para o gerenciamento destes documentos, (iii) disponibilizar interface onde os avaliadores possam revisar os documentos acadêmicos, inclusive com a possibilidade de incluir comentários e marcas de texto na revisão, (iiii) disponibilizar formulário de avaliação dos documentos acadêmicos.

O SGDA possui um *design* de interface que é comum a todos usuários, no entanto para garantir a integridade e confidencialidade das informações algumas visualizações e permissões de

acessos são restritas conforme o tipo de usuário que está acessando o *software*. Existem 3 tipos de usuário, são eles: Administrador, Aluno e Professor. O Quadro 5 mostra as funcionalidades que cada tipo de usuário terá acesso.

Quadro 5 – Acessos por tipo de usuário

Tipo de usuário	Acessos
Administrador	Cadastrar usuário
	Postar documento
	Visualização de documento
	Visualização da identificação do dono do documento
	Selecionar avaliadores para documento
	Revisar documento
	Avaliar documento
	Arquivar documento
	Pesquisar documento
Professor	Dados complementares do cadastro de usuários
	Alterar senha
	Visualizar documento (nesta visualização a identidade do aluno é ocultada para atender o requisito de <i>blindreview</i>)
	Revisar documento
	Avaliar documento
Aluno	Auto cadastro do usuário
	Alterar senha
	Postar documento
	Visualizar documento (somente documentos postados pelo usuário)

Fonte: Própria

Os usuários do tipo Professor podem ser selecionados como orientadores de alunos e também serem selecionados como avaliadores. No caso do professor ser orientador de um determinado aluno, ele não será visível na lista de avaliadores disponível para avaliar o documento daquele aluno ao qual ele é orientador, esta restrição é um dos requisitos levantados durante a fase de análise.

Para facilitar o andamento dos processos, o SGDA conta com um sistema de avisos através de e-mails referente a algumas ações realizadas no sistema, conforme mostrado no Quadro 6.

Quadro 6 - Mensagem de e-mail por processo.

Processo	Ação
Postagem de documento	Envia e-mail para o administrador informando que um novo documento foi postado.
Seleção de avaliadores para revisão e avaliação do documento	Envia e-mail para o(s) avaliador(es) informando que um documento foi enviado para ele revisar e avaliar.
Revisão de documento concluída	Envia e-mail para o administrador informando que a revisão do documento tal foi concluída.
Avaliação de documento	Envia e-mail para o administrador informando que a avaliação do documento tal foi concluída.
Liberação da avaliação para aluno	Envia e-mail para o aluno informando que o documento já foi avaliado e a avaliação esta disponível para visualização.

Fonte: Própria

A Figura 11 mostra o menu principal do *software* na visão do professor que administra o sistema, nele temos o link que fornece acesso ao cadastro de usuários do sistema, link para visualização e controle dos documentos postados, link para postagem de documentos caso se faça necessário e link para pesquisa de documentos ativos ou arquivados de anos anteriores.



FIGURA 11: Interface principal do *software* SGDA
Fonte: Própria.

A Figura 12 mostra a interface de cadastramento de usuários, permite incluir novo usuário, mostrar todos usuários cadastrados ou filtrá-los pelo seu tipo que pode ser: Aluno, Professor ou Administrador. O tipo de usuário define qual o tipo de visualização e permissões que o usuário terá acesso.

FIGURA 12: Cadastro de usuários
Fonte: Própria.

Atendendo o requisito de eliminar o tráfego de documentos por e-mail, os documentos são postados no sistema e ficam disponíveis para serem revisados e avaliados, a Figura 13 mostra a interface de postagem dos documentos.

FIGURA 13: Postar documentos
Fonte: Própria.

A Figura 14, mostra a interface de visualização dos documentos postados na visão do administrador, exibindo o título do documento, nome do arquivo postado, proprietário e também consegue-se visualizar facilmente através de cores quais avaliadores já concluíram o processo de revisão e avaliação. A cor vermelha indica que o processo ainda não foi concluído e a cor verde indica que esta concluída. Nesta interface clicando no link do título, o administrador pode selecionar os avaliadores para o documento.

Título	Arquivo	Proprietário
Projeto teste de postagem de arquivo	apostila_1_0.pdf...pdf	Jonis dos Santos
Projeto teste 2	apostila_2_0.pdf...pdf	Jonis dos Santos
Avaliadores	Revisado Avaliado Liberar	
Eurico Antunes		
Francisco Assis Nascimento	ABRIR ABRIR Libera avaliação para aluno	
Projeto RFID	apostila_5_0.pdf...pdf	Jonis dos Santos
Avaliadores	Revisado Avaliado Liberar	
Eurico Antunes		
Marcelo Azambuja	ABRIR ABRIR Libera avaliação para aluno	
Projeto Horário 2009	horario2009_1_v1_0.p...pdf	Maria da Silva
Avaliadores	Revisado Avaliado Liberar	
Francisco Assis Nascimento	ABRIR	
Projeto Horário Disciplinas 2006-1	horsiinfo2006_1_v3...pdf	Juliano da Silva
Avaliadores	Revisado Avaliado Liberar	
Everton Luis Berz		
Marcelo Azambuja		

FIGURA 14: Interface de visualização de documentos (visão administrador)
Fonte: Própria.

A Figura 15, mostra os documentos a serem revisados e avaliados na interface do professor, aqui pode-se identificar o título do documento e o nome do arquivo, o proprietário do documento não é visível justamente para atender o requisito de *blindreview*, nesta interface clicando no nome do arquivo, tem-se acesso a interface de revisão do documento. Também estão disponíveis os links para confirmar a conclusão da revisão e para preencher a avaliação do documento.

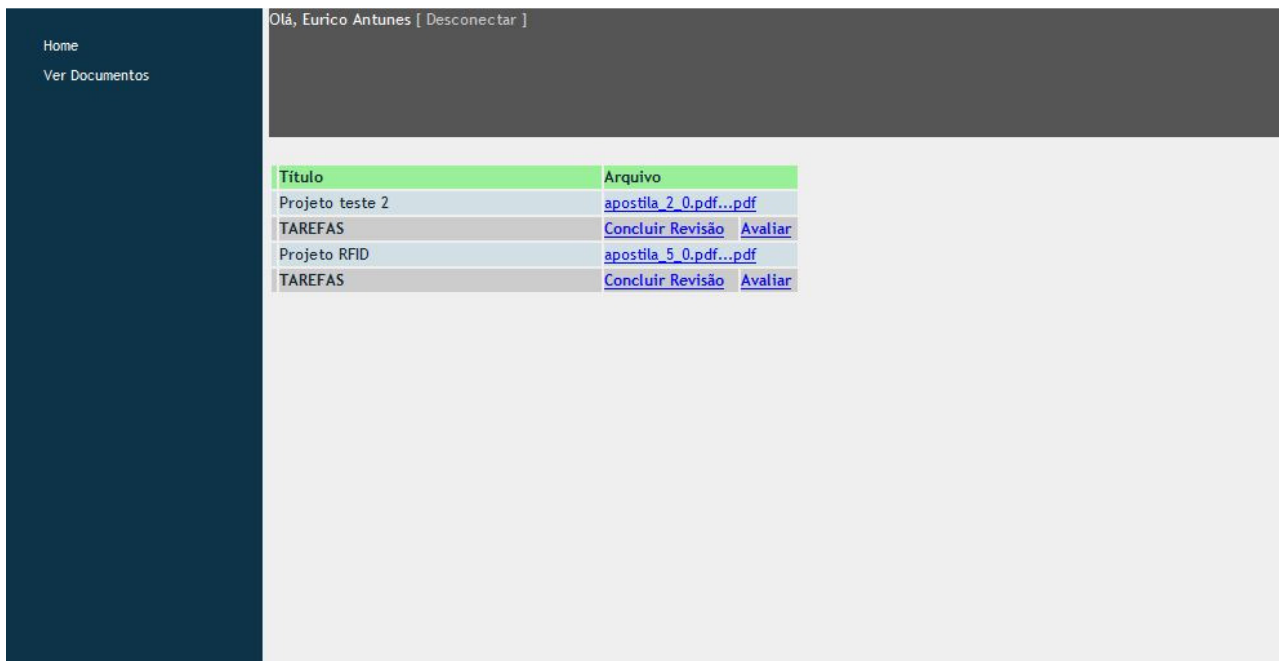


FIGURA 15: Interface de visualização de documentos (visão professor)
Fonte: Própria.

A Figura 16 mostra um exemplo de documento sendo revisado através do plugin do crocodoc que foi agregado ao SGDA.

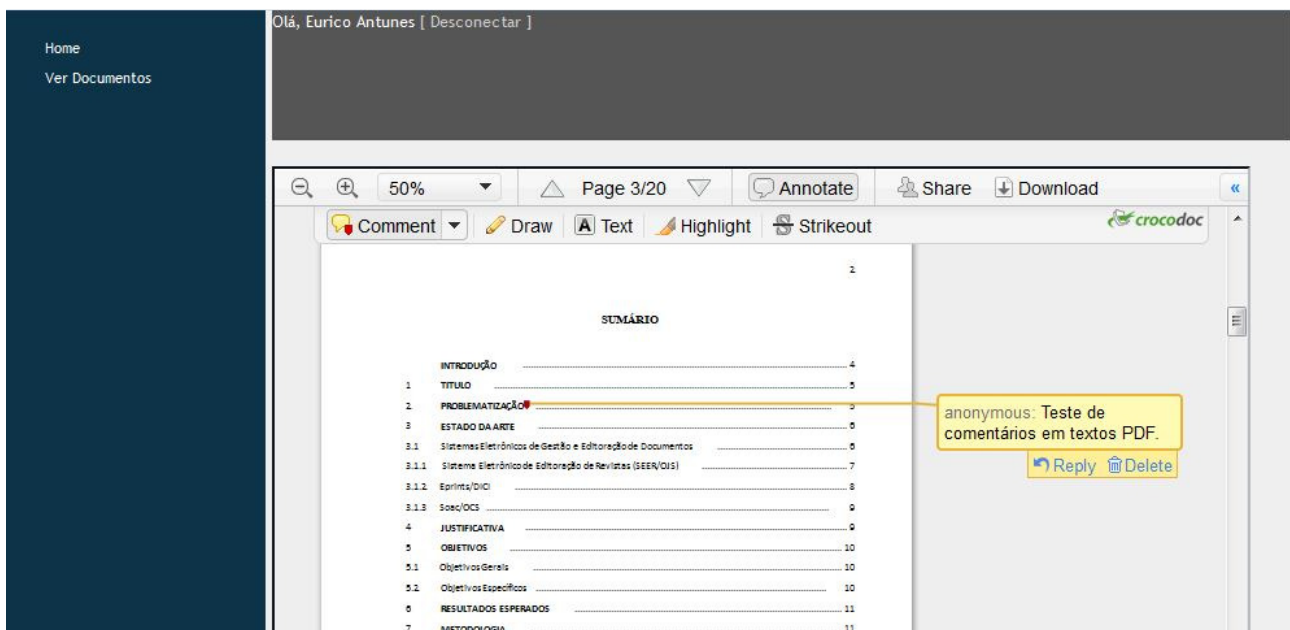


FIGURA 16: Interface de revisão de documento PDF
Fonte: Própria

A Figura 17 mostra a interface de avaliação de documentos de projeto de pesquisa. Os campos desta interface são carregados automaticamente com todas as informações referentes ao curso do aluno, título do projeto e identificação do avaliador, sendo necessário apenas ao avaliador preencher o resultado final da avaliação, atribuindo uma nota que poderá variar entre 0 e 10. Também é possível inserir sugestões para melhorias a serem feitas no projeto, caso ache necessário.

AVALIAÇÃO DO PROJETO DE PESQUISA (ESTÁGIO)

Curso:
Sistemas de Informação

01. TÍTULO DO PROJETO DE PESQUISA (ESTÁGIO)
Projeto teste 2

02. IDENTIFICAÇÃO DO AVALIADOR

Nome Completo:
Eurico Antunes

Endereço:
Rua dos Andradas

Número: 55 Complemento: Apartamento Município: Porto Alegre

CEP: 90000-000 UF: RS Telefone Celular: 99887766 Telefone Fixo: 35335050 Ramal:

E-mail:
Eurico@faccat.br

Titulação ou Certificação: Mestre

Nome do Curso que realizou:
CIENCIA DA COMPUTAÇÃO

Nome do Título ou Certificação (que consta no diploma):
CERTIFICACAO TESTE

Instituição que conferiu o título ou certificação:
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Ano: 2000

03. RESULTADO FINAL DA AVALIAÇÃO DO PROJETO DE PESQUISA (ESTÁGIO)
Atribuir uma Nota da 0 a 10

Nota Final da Avaliação:

04. SUGESTÕES PARA MELHORIAS
Envie na sequência seus comentários para melhorias a serem feitas no desenvolvimento do projeto durante a disciplina de TCC:

ENVIA AVALIAÇÃO

FIGURA 17: Interface de avaliação de projeto de pesquisa
Fonte: Própria.

5. Conclusão

O trabalho de pesquisa e desenvolvimento consistiu na criação de um *software* web para auxiliar no processo de revisão e avaliação dos documentos acadêmicos, facilitando o fluxo dos documentos, auxiliando na seleção dos avaliadores, no gerenciamento dos documentos enviados para revisão e na visualização das informações de processos pendentes de ação de algum ator e ações já concluídas. O software fornece como diferencial inovador uma interface para revisão dos documentos no próprio ambiente da aplicação.

Conclui-se que o *software* SGDA atende aos requisitos captados para concepção do sistema, possibilitando eliminar o tráfego de documentos através de e-mails, gerenciar o envio e o retorno dos documentos enviados para revisão e avaliação, bem como prover um ambiente na web para que os documentos possam ser revisados e avaliados no ambiente da aplicação.

Pretende-se para as versões futuras do *software*, incluir outros formulários de avaliação que atendam a cada tipo de documento, melhorar o *design* das interfaces utilizando recursos gráficos para facilitar acessos e visualizações das informações, com foco voltado para melhoria da usabilidade da aplicação. Pretende-se também desenvolver uma interface de revisão de documentos própria do SGDA para que o sistema seja totalmente independente de aplicações externas.

Este projeto de pesquisa e desenvolvimento foi elaborado para auxiliar o controle e gerenciamento dos documentos acadêmicos gerados nas fases finais de conclusão de curso das disciplinas de Sistemas de Informação, Tecnólogo Sistemas para Internet e Engenharia de Produção das Faculdades Integradas de Taquara (FACCAT). O professor responsável pelas disciplinas de conclusão, Prof. Dr. Carlos Fernando Jung, terá total acesso ao código fonte e documentação do *software*, inclusive das melhorias mencionadas anteriormente.

Almeja-se que futuramente outros cursos desta instituição ou até mesmo outras instituições utilizem-se desta ferramenta para facilitar o processo de avaliação dos documentos acadêmicos, propostas de TCCs, projetos de pesquisa e artigos científicos. Para o caso de expansão do uso da ferramenta para outras instituições, ainda será analisado de que forma será feita a distribuição.

Concluindo, resta mencionar que o desenvolvimento deste *software* foi extremamente gratificante a todos os envolvidos neste trabalho, dado o desafio pessoal que contribuiu na ampliação do conhecimento do autor e colaboradores.

Referências

- APACHE, **Servidor web**. Disponível em: <http://pt.wikipedia.org/wiki/Servidor_Apache>. Acesso em: 15 Ago. 2011.
- CHANGE VISION. **Astah Community**. 2011. Disponível em: <<http://astah.net>>. Acesso em: 02mar. 2012.

CKEDITOR.**What is CKEditor**. Disponível em: <<https://ckeditor.com/what-is-ckeditor/>>. Acesso em: 14abr. 2012.

CROCODOC.**API Documentation**. Disponível em: <<https://crocodoc.com/docs/api>>. Acesso em: 12abr. 2012.

FLANAGAN, David, **JavaScript– O GUIA DEFINITIVO**. Disponível em:
<<http://books.google.com.br/books?hl=pt-BR&lr=&id=ZuTxAKCczMkC&oi=fnd&pg=PA19&dq=javascript&ots=F7YLQaTJfS&sig=AGUemw78hFW5K7JBYahiKcvDd3w#v=onepage&q=javascript&f=false>>. Acesso em: 25 Nov. 2011.

FERREIRA, Sueli *et al.* **Em busca de um repositório ideal: Análise de software baseado em arquivos abertos**. Disponível em: <<http://www.febab.org.br/rbbd/index.php/rbbd/article/viewFile/66/57>>, Acesso em: 01 out. 2011.

FONSECA, Ramón Martins Sodoma: **OJS em uma hora**. Disponível em:<www.uff.br/RVQ/ManualOJS.pdf>. Acesso em: 12 nov. 2011.

GN1 GENESIS NETWORK. **Visão geral do SGP. 2012**. Disponível em: <[http://www.gn1.com.br/tutorial/Visão geral do SGP.pdf](http://www.gn1.com.br/tutorial/Visão%20geral%20do%20SGP.pdf)>. Acesso em 30 jun. 2012.

MAGRIN, Maria Heloiza. **Guia do profissional Linux**. Disponível em:
<http://books.google.com.br/books?id=pYosPwNV7UUC&pg=PT105&dq=servidores+web+apache&hl=pt-BR&ei=jxfUTqnLAejf0QHG5qlH&sa=X&oi=book_result&ct=result&resnum=&ved=0CG8Q6AEwBw#v=onepage&q=servidores%20web%20apache&f=false>. Acesso em: 27 Nov. 2011.

MINETTO, Elton L. **Frameworks para desenvolvimento em PHP**. São Paulo: Novatec Editora, 2007.

MySQL,**Open Source Database**. Disponível em: <<http://www.mysql.com/>>. Acesso em: 14 Ago. 2011.

NIEDERAUER, Juliano. **Desenvolvimento Websites com PHP**. São Paulo: Novatec Editora, 2011.

PESSMAN, R.S. **Engenharia de Software**. 5.ed. Rio de Janeiro: McGraw-Hill, 2002.

SEER. **Portal do Sistema Eletrônico de Editoração de Revistas**. Disponível em: <<http://seer.ibict.br/>>. Acesso em: 12 nov. 2011.

TINYMCE.**Javascript WYSIWYG Editor**. Disponível em: <<https://www.tinymce.com/wiki.php/installation>>. Acesso em: 12 abr. 2012.

W3C, **Hipertext Markup Language**. Disponível em:
<<http://www.w3.org/TR/html>>. Acesso em: 20 Ago. 2011.