

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

SISTEMA DE INFORMAÇÕES DE APOIO À ELABORAÇÃO DO
RELATÓRIO INDIVIDUAL DE ATIVIDADES

FREDERICO JOSÉ DE SOUZA FILHO

FREDERICO JOSÉ DE SOUZA FILHO

**SISTEMA DE INFORMAÇÕES DE APOIO À ELABORAÇÃO DO RELATÓRIO
INDIVIDUAL DE ATIVIDADES**

**Monografia apresentada a Universidade
Federal de Santa Catarina como requisito
para obtenção do grau de Bacharel em
Ciências da Computação.**

ORIENTADOR: MARCIO CLEMES

CO-ORIENTADOR: RENATO CISLAGHI

BANCA

JOSE MARCOS DA SILVA

FLORIANÓPOLIS

2005

RESUMO

Neste trabalho propôs-se um sistema para automatizar o preenchimento do Relatório Individual de Atividades (RIA). Para tanto, desenvolveu-se um estudo sobre as fontes de informação existentes na instituição de ensino que pudessem contribuir para o preenchimento e eliminar o fornecimento de informações inconsistentes. Fez-se, ainda, uma análise dos processos envolvidos para a aprovação do formulário na Universidade Federal de Santa Catarina-UFSC e a partir deste estudo, propôs-se um sistema para acompanhamento do formulário. Por fim, iniciou-se a implementação de um protótipo capaz de suportar o preenchimento automático do formulário e acompanhar a tramitação do mesmo através da instituição.

Palavras-Chave: GED, Relatório Individual de Atividades, Aplicações WEB, Java, Jsp

ABSTRACT

In this project it was considered a system to automatize the fulfilling of Individual Report of Activities. For this, it was developed a study about the sources of information existing in the education institution that they could contribute for the fulfilling and eliminate the supply of inconsistentes information. One became, still, an analysis of the processes involved for the approval of the report in the Universidade Federal de Santa Catarina-UFSC and from this study, it was considered a system for the accompaniment of the form. Finally, it was initiated the implementation of the prototype capable to support the automatic fulfilling of the form and accompaniment the transaction through the institution.

Key words: GED, Individual Report of Activities ,Web Application, Java, JSP.

Agradecimentos

A meus pais, pelo amor, carinho, confiança e pela educação que me fez crescer.

A minha noiva Grazzi, pelo amor, pelos momentos felizes e por sua ajuda na revisão deste trabalho.

Agradeço a minha vó Guta, pelos exemplos e ensinamentos sempre presentes na minha vida.

Agradeço a Marcio Cledes, Renato Cislighi e José Marcos por terem me acompanhado na elaboração deste trabalho e contribuído com suas orientações.

A toda equipe do NPD da UFSC pelo conhecimento repassado durante meu estágio e elaboração deste trabalho e pela infraestrutura proporcionada para sua realização.

SUMÁRIO

LISTA DE FIGURAS.....	viii
LISTA DE SIGLAS.....	ix
1 INTRODUÇÃO.....	10
2 PROPOSTA DO TRABALHO.....	11
2.1 OBJETIVOS.....	11
2.1.1 OBJETIVOS GERAIS.....	11
2.1.2 OBJETIVOS ESPECÍFICOS.....	11
2.2 MOTIVAÇÃO.....	12
3 INFORMAÇÕES RELATIVAS ÀS ATIVIDADES DOCENTES.....	14
4 RELATÓRIO INDIVIDUAL DE ATIVIDADES.....	15
5 GRATIFICAÇÃO DE ESTÍMULO À DOCÊNCIA.....	16
5.1 CONCEITO.....	16
5.2 VISÃO GERAL.....	16
5.3 OBJETOS DE AVALIAÇÃO DO DESEMPENHO DOCENTE.....	17
5.3.1 ATIVIDADES DE ENSINO.....	18
5.3.2 PRODUÇÃO INTELECTUAL.....	19
5.3.3 ATIVIDADES DE PESQUISA E DE EXTENSÃO.....	19
5.3.4 ATIVIDADES DE QUALIFICAÇÃO.....	20
5.3.5 ATIVIDADES ADMINISTRATIVAS E DE REPRESENTAÇÃO.....	20
5.3.6 AVALIAÇÃO QUALITATIVA DAS ATIVIDADES DE ENSINO.....	21
5.3.7 OUTRAS ATIVIDADES.....	21
6 PROGRESSÃO FUNCIONAL.....	22
7 ANÁLISE DAS POSSÍVEIS SOLUÇÕES.....	25
7.1 O PROFESSOR PREENCHE O FORMULÁRIO.....	25
7.2 AS INFORMAÇÕES SÃO OBTIDAS A PARTIR DO CURRÍCULO LATTES.....	26
7.3 AS INFORMAÇÕES SÃO OBTIDAS A PARTIR DOS SISTEMAS EXISTENTES NA INSTITUIÇÃO	28

8	TECNOLOGIAS.....	29
8.1	<i>APLICATIVOS WEB</i>	30
8.2	<i>JAVA</i>	32
8.3	<i>JSP</i>	33
8.4	<i>BANCO DE DADOS</i>	34
9	DESENVOLVIMENTO	36
9.1	<i>FONTES DE INFORMAÇÃO</i>	36
9.2	<i>DEFINIÇÃO DOS ATORES</i>	38
9.3	<i>CASO DE USO</i>	38
9.3.1	<i>BANCO DE DADOS</i>	45
10	CONSIDERAÇÕES FINAIS	46
10.1	<i>TRABALHOS FUTUROS</i>	46
10.2	<i>CONCLUSÃO</i>	47
11	REFERÊNCIAS BIBLIOGRÁFICAS	48
	ANEXO 1 - MAPEAMENTO DAS ATIVIDADES UTILIZADAS PARA PREENCHIMENTO DO	
	RIA	49
	ANEXO 2 – CODIGO FONTE	59

LISTA DE FIGURAS

Figura 1: Tramitação do formulário de acordo com o Departamento de Engenharia Química - UFSC	13
Figura 02 – Arquitetura 2 camadas.....	31
Figura 03 – Arquitetura 3 camadas.....	31
Figura 04 - Contribuição dos sistemas para o preenchimento do formulário GED.	37
Figura 05 – Diagrama de Casos de Uso	39

LISTA DE SIGLAS

CAGR	Sistema de Controle Acadêmico de Graduação
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CAPG	Sistema de Controle Acadêmico de Pós-Graduação
CIAG	Comissão Institucional de Atribuição da GED
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
CPPD	Comissão Permanente de Pessoal Docente
GED	Gratificação de Estímulo a Docência
IES	Instituição de Ensino Superior
IFES	Instituição Federal de Ensino Superior
J2EE	Java 2 SDK, Enterprise Edition
JDBC	<i>Java Database Connectivity</i>
JSP	JavaServer Pages
LATTES	Plataforma LATTES CNPq
MVC	Model-View-Controller
NPD	Núcleo de Processamento de Dados
PREG	Pró-Reitoria de Ensino e Graduação
RIA	Relatório individual de atividades
SIAPE	Sistema Integrado de Administração de Recursos Humanos
UFSC	Universidade Federal de Santa Catarina

1 INTRODUÇÃO

Os professores de ensino superiores, lotados e em exercício nas instituições federais vinculadas ao Ministério da Educação preenchem anualmente um formulário para o recebimento da Gratificação de Estímulo à Docência (GED), sendo que, alguns deles, ainda preenchem o Currículo Lattes, os Formulários da CAPES e outros formulários próprios de cada instituição como o Relatório Individual de Atividades (RIA).

Neste projeto busca-se apresentar uma proposta para automatizar o preenchimento do formulário RIA, eliminar o fornecimento de informações inconsistentes, desenvolver um repositório para integração dessas informações.

2 PROPOSTA DO TRABALHO

Este projeto visa integrar as informações disponíveis nos diversos sistemas institucionais em um ambiente projetado para automatizar o preenchimento do formulário Relatório Individual de Atividades (RIA). Para tanto será desenvolvido um repositório para que os dados sejam unificados.

2.1 OBJETIVOS

2.1.1 OBJETIVOS GERAIS

Desenvolver um ambiente que proporcione o preenchimento automático do formulário RIA.

2.1.2 OBJETIVOS ESPECÍFICOS

- a) Integrar as informações relativas às atividades realizadas pelos docentes em um repositório de dados;
- b) Agilizar e otimizar os procedimentos necessários para o preenchimento do formulário RIA;
- c) Desenvolver de uma ferramenta baseada na WEB para capturar, visualizar e acompanhar as informações geradas.

2.2 MOTIVAÇÃO

A partir da observação do procedimento utilizado para o preenchimento do Relatório Individual de Atividades (RIA) na Universidade Federal de Santa Catarina – UFSC, constatou-se que muitas das informações necessárias para o preenchimento do referido formulário poderiam ser retiradas dos sistemas existentes na própria instituição de ensino. Entretanto, até o presente, isso não acontece.

Assim, o formulário que poderia ser preenchido automaticamente pela própria instituição, cabendo ao docente apenas validar as informações constantes no relatório, hoje tramita através de diversos setores dentro dela, conforme se observa no diagrama abaixo, formulado através de entrevista realizada com servidores do departamento do curso de engenharia química da UFSC.

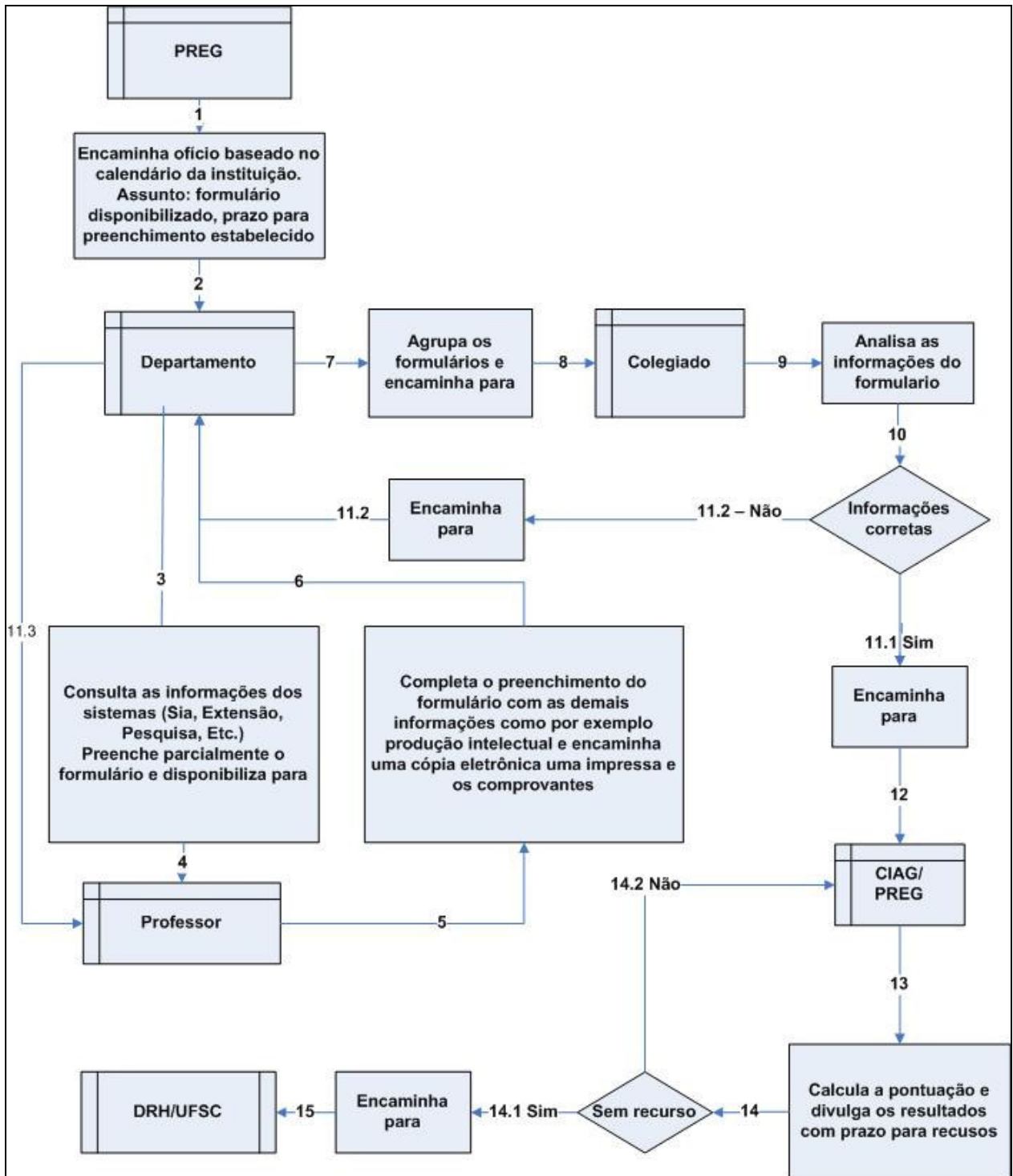


Figura 1: Tramitação do formulário de acordo com o Departamento de Engenharia Química - UFSC

3 INFORMAÇÕES RELATIVAS ÀS ATIVIDADES DOCENTES

O docente, durante sua vida acadêmica, desenvolve atividades que podem ser utilizadas em diversos processos de avaliação. Esses processos podem desencadear, por exemplo, a concessão de gratificações ou a progressão funcional.

Todavia, para que se possam implementar esses processos de avaliação, torna-se necessário o preenchimento de formulários. Nesses, os docentes fornecem uma série de dados relativos às atividades que vem desenvolvendo, tais como quantidade de horas-aula ministradas, participação em bancas de avaliação, orientação de trabalhos, participação em cursos, produção intelectual, etc.

Os formulários acima referidos apresentam formatos diferentes de acordo com o objetivo com o qual são criados. Entretanto, os dados constantes em cada um deles não variam substancialmente, tendo em vista serem todos relativos às atividades desempenhadas pelo docente.

Dentre as principais finalidades dos formulários pode-se ressaltar a elaboração do Relatório Individual de Atividades (RIA), a concessão da Gratificação de Estímulo a Docência (GED) e a progressão funcional, abordados a seguir.

4 RELATÓRIO INDIVIDUAL DE ATIVIDADES

As normas para elaboração do Relatório Individual de Atividades (RIA) foram estabelecidas a partir da Resolução nº 053/CEPE/95, de 31 de agosto de 1995, da Universidade Federal de Santa Catarina. Dentre outras funcionalidades, o relatório é utilizado para subsidiar a elaboração do Relatório de Atividades do Departamento (RAD). Para tanto, agrupa as atividades do magistério superior desenvolvidas pelo docente.

A partir da criação da Gratificação de Estímulo à Docência (GED) o RIA passou a ser utilizado também para dar suporte ao cálculo da pontuação da gratificação.

5 GRATIFICAÇÃO DE ESTÍMULO À DOCÊNCIA

5.1 CONCEITO

Gratificação é a vantagem de natureza financeira que o servidor percebe em razão do exercício de uma atividade comum em condições especiais.

Essa vantagem pecuniária está prevista na Lei n. 8.112/90, em seu artigo 49, inciso II, que dispõe:

“Art. 49. Além do vencimento, poderão ser pagas ao servidor as seguintes vantagens:
I - indenizações;
II - gratificações;
III – [...]”

A Gratificação de Estímulo à Docência - GED foi instituída em 3 de julho de 1998 pela Lei n. 9.678 e é percebida anualmente pelos professores do 3º grau, lotados e em exercício nas instituições federais de ensino superior vinculadas ao MEC. É calculada com base nas atividades realizadas pelos professores obedecendo a uma série de critérios e limites de pontuação.

5.2 VISÃO GERAL

Com base no texto da Comissão de Acompanhamento e Orientação da Gratificação de Estímulo à Docência “o processo de avaliação da atividade docente nas Instituições Federais de Ensino Superior – IFES, com vistas à concessão da Gratificação de Estímulo à Docência, segue

as diretrizes estabelecidas na Lei n. 9.678, de 03/07/1998, no Decreto n. 2.668, de 13/07/1998 e no Relatório da Comissão Nacional, de 03/08/1998 [...].¹”

Tal processo de avaliação foi encaminhado para diversas instituições ligadas ao Ministério da Educação – MEC com o objetivo de ser apreciado e discutido em um prazo estabelecido. Após a conclusão deste prazo, as diretrizes do processo de avaliação foram traçadas com os seguintes objetivos:

- contribuir para o aperfeiçoamento dos mecanismos de implementação e concessão da GED;
- sistematizar as recomendações discutidas nas reuniões regionais, de modo a propiciar maior convergência na interpretação das diretrizes normativas da GED;
- estabelecer parâmetros de avaliação da atividade docente, de forma a promover certa coerência do processo para o conjunto das IFES, sem comprometer a autonomia de cada instituição para o desenvolvimento de políticas próprias.²

Após a criação das diretrizes e dos parâmetros para concessão da gratificação, percebeu-se a necessidade de um relatório a partir do qual fossem colhidos os dados necessários para o cálculo da GED.

5.3 OBJETOS DE AVALIAÇÃO DO DESEMPENHO DOCENTE

Visando criar um critério para o processo de avaliação do desempenho docente a Comissão Institucional de Atribuição da GED (CIAG) adotou o Relatório Individual de Atividades (RIA).

¹ Ministério da Educação, 2004.

² Ministério da Educação, 2004.

Nesse relatório constam as informações que são objetos da avaliação do desempenho docente, ou seja, aquelas utilizadas para pontuação e cálculo da gratificação. As atividades estão organizadas da seguinte forma:

- atividades de ensino;
- produção intelectual;
- atividades de pesquisa e de extensão;
- atividades de qualificação;
- atividades administrativas e de representação;
- avaliação qualitativa das atividades de ensino; e
- outras atividades.

Cabe, neste momento, analisar cada uma dessas atividades.

5.3.1 ATIVIDADES DE ENSINO

Conforme o artigo 44 da Lei de Diretrizes e Bases da Educação Nacional – LDB, as atividades de ensino superior, tais como cursos sequenciais, de graduação e pós-graduação “stricto” e “lato sensu”, são entendidas como aquelas formalmente incluídas nos planos de integralização curricular dos cursos das IFES. Assim, as atividades de orientação e de supervisão não previstas nos referidos planos só podem ser consideradas no item ‘outras atividades’.

Assim, não devem ser consideradas como atividades de ensino:

- as atividades de ensino pelas quais o docente receba remuneração adicional específica, exceto em caso de Pós-Graduação “stricto sensu” interinstitucional

instituída no âmbito do Programa, por ela avaliada, e que exija o deslocamento do docente da instituição-sede;

- participação em banca examinadora.³

5.3.2 PRODUÇÃO INTELECTUAL

A produção intelectual compreende as produções científica, artística, tecnológica e cultural, representadas através de publicações ou de outras formas de expressão usuais e pertinentes aos ambientes acadêmicos específicos, avaliadas de acordo com a sistemática utilizada pela CAPES e pelo CNPq para as diferentes áreas do conhecimento.

5.3.3 ATIVIDADES DE PESQUISA E DE EXTENSÃO

Neste item devem ser levados em consideração os seguintes aspectos:

- projetos de pesquisa, de desenvolvimento tecnológico e de extensão aprovados pela instância competente da IFES no período de avaliação considerado;
- relatórios parciais de pesquisa em andamento cujo projeto não tenha sido pontuado no mesmo período de avaliação ou gerado resultados incluídos no item Produção Intelectual;
- atividades artísticas, culturais e assistenciais, assim como de disseminação e transferência de conhecimento científico, tecnológico e cultural, e atividades de

³ Ministério da Educação, 2004.

prestação de serviços, não considerados nos grupos II. 1 – Produção Bibliográfica e II. 2 – Produção Artística, aprovados pela instância competente da IFES.

Cumpra salientar que não devem ser consideradas neste item as atividades de prestação de serviços pelas quais o docente receba remuneração específica.

5.3.4 ATIVIDADES DE QUALIFICAÇÃO

Segundo o texto da Comissão de Acompanhamento e Orientação da Gratificação de Estímulo à Docência, são atividades de qualificação que somam pontos para GED

“atividades desenvolvidas pelo docente como aluno de curso de pós-graduação stricto sensu, com dispensa parcial de atividades de ensino, ou como participante de estágio de pós-doutoramento. A consideração dessas atividades é necessariamente condicionada à aprovação dos respectivos relatórios, segundo os procedimentos e pelas instâncias competentes da IFES”.⁴

5.3.5 ATIVIDADES ADMINISTRATIVAS E DE REPRESENTAÇÃO

São consideradas aqui: a representação acadêmica e participação em órgãos colegiados; as atividades de coordenação, chefia ou direção na IFES, exceto as situações previstas no art. 4º, § 1º da Lei 9.678/98 e no item 4.4 das normas estabelecidas no Relatório da Comissão Nacional. Representação Sindical Docente; a participação não remunerada em conselhos ou comissões de

⁴ Ministério da Educação, 2004.

órgãos governamentais e de entidades científicas, culturais e profissionais; bem como outras atividades assemelhadas.

5.3.6 AVALIAÇÃO QUALITATIVA DAS ATIVIDADES DE ENSINO

Avaliação da qualidade da atividade de ensino é realizada nos termos de um projeto institucional de avaliação, aprovado e acompanhado pelas instâncias competentes da IFES.

5.3.7 OUTRAS ATIVIDADES

Consistem nas atividades de orientação e supervisão não incluídas nos planos de integralização curricular dos cursos da IFES, na participação em bancas examinadoras, em cursos de qualificação não incluídos no item II. 4, além de outras atividades assemelhadas.

6 PROGRESSÃO FUNCIONAL

Regulamentada pelo Decreto nº 96.664, de 23 de julho de 1987, e pela Portaria nº 475/87, de 26 de agosto do mesmo ano, a progressão funcional nas carreiras do magistério ocorre, segundo a legislação acima citada, exclusivamente por titulação e desempenho acadêmico.

A avaliação para a progressão funcional por desempenho acadêmico obedece às normas e aos critérios estabelecidos pelo Conselho Superior competente da Instituição Federal de Ensino (IFE) e incide sobre as atividades relacionadas ao exercício do cargo ou emprego de Magistério. De acordo com a Portaria nº 475/87, eis alguns critérios:

- a) “desempenho didático, avaliado com a participação do corpo discente;
- b) orientação de dissertações e teses de Mestrado e Doutorado, de monitores e de estagiários ou bolsistas de iniciação científica;
- c) participação em bancas examinadoras de dissertações, de teses e de concurso público para o magistério;
- d) cursos ou estágios de aperfeiçoamento, especialização e atualização, bem como créditos e títulos de pós-graduação “stricto sensu”;
- e) produção científica, técnica ou artística;
- f) atividade de extensão à comunidade dos resultados da pesquisa, de cursos e de serviços;
- g) participação em órgãos colegiados na própria IFE ou vinculados aos Ministérios da Educação, da Cultura e da Ciência e Tecnologia;

- h) exercício de funções de direção, coordenação, assessoramento e assistência na própria IFE, ou em órgãos dos Ministérios da Educação, da Cultura e da Ciência e Tecnologia, bem como em outros previstos na legislação vigente.”⁵

Com base nesses critérios a Comissão Permanente de Pessoal Docente (CPPD) analisa e avalia as atividades realizadas pelo docente e seu desempenho. O período para a avaliação do docente é determinado no Decreto nº 96.664 que estabelece:

“A progressão prevista no item II far-se-á sem interstício, por titulação ou mediante avaliação de desempenho acadêmico do docente que não obtiver a titulação necessária mas que esteja, no mínimo, há dois anos no nível 4 da respectiva classe ou com interstício de quatro anos de atividade em órgão público.”⁶

O processo para concessão de progressão funcional por desempenho acadêmico, dentro da Universidade Federal de Santa Catarina (UFSC), foi analisado a partir de reunião realizada com servidores da Comissão Permanente de Pessoal Docente (CPPD).

Segundo eles, é utilizado neste processo um formulário que contém o resumo das atividades realizadas pelo docente, devendo as informações serem convalidadas através de comprovantes enviados pelo docente.⁷

A partir disso, verificou-se que essas informações referentes às atividades desempenhadas pelos também constam no formulário do Relatório Individual de Atividades (RIA).

Fica evidenciado, portanto, que a utilização do sistema proposto neste trabalho poderá eliminar a preenchimento manual do formulário necessário para a concessão da progressão

⁵ Portaria n. 475, de 26 de agosto de 1987.

⁶ Decreto n. 96.664 de 23 de julho de 1987.

⁷ O formulário utilizado para a progressão funcional por desempenho acadêmico pode ser analisado com maiores detalhes no Anexo 2.

funcional e a elaboração dos comprovantes se o departamento de ensino aprovar através do sistema proposto as atividades desenvolvidas pelo docente.

7 ANÁLISE DAS POSSÍVEIS SOLUÇÕES

Neste capítulo será apresentada uma análise das possíveis soluções para agilizar o preenchimento do formulário RIA. Serão abordadas as vantagens e desvantagens de cada uma das soluções cogitadas visando selecionar a que melhor se adequa às necessidades das instituições de ensino.

7.1 O PROFESSOR PREENCHE O FORMULÁRIO

A proposta inicial abordada nesse capítulo é disponibilizar para o docente um formulário capaz de capturar as informações necessárias para o preenchimento do RIA. Este possuiria uma série de regras visando validar as informações fornecidas pelo docente.

Vantagens

1. Baixa complexidade para desenvolvimento da solução;
2. Flexível, pois o docente poderia digitar as informações observando apenas as regras de validação;
3. Possui baixo acoplamento, pois não é integrada a nenhuma outra aplicação.

Desvantagens

1. Durante o processo de digitação erros poderiam ocorrer;
2. Pouca integração com as informações já mantidas pelos sistemas da instituição, o que poderia gerar inconsistências. Por exemplo, o docente poderia informar que ministra uma disciplina e essa informação não estar presente nos sistemas da instituição;
3. Docentes com número elevado de atividades desenvolvidas gastariam muito tempo para o preenchimento do formulário.

7.2 AS INFORMAÇÕES SÃO OBTIDAS A PARTIR DO CURRÍCULO LATTES

O Currículo LATTES é um componente da Plataforma Lattes que foi desenvolvido para o CNPq e é utilizado pelo MCT, FINEP, CAPES/MEC, por todos os atores institucionais, bem como pela comunidade científica brasileira como sistema de informação curricular. Tem por objetivo facilitar a avaliação da competência de candidatos para obtenção de bolsas e auxílios, auxiliar na seleção dos consultores de membros de comitês e de grupos assessores, além de dar subsídio à avaliação da pesquisa e da pós-graduação brasileiras. (LATTES, 2005)

Uma das facilidades da Plataforma Lattes é sua disponibilidade de instalação nas instituições cadastradas pelo CNPq. Dessa forma, os currículos enviados para o CNPq pelo docente podem ser enviados para a instituição automaticamente e as informações armazenadas pela plataforma da instituição acessada para consulta.

A partir da análise detalhada das funcionalidades do Currículo Lattes, constatou-se que todas as atividades que devem ser preenchidas no RIA podem ser retiradas dele. Concluída esta, a

utilização do mesmo foi cogitada para o preenchimento do RIA e as seguintes conclusões foram obtidas.

Vantagens

1. Média complexidade para desenvolvimento da solução;
2. Flexível, pois o docente poderia digitar as informações observando apenas as regras de validação;
3. Os docentes que já utilizam o Lattes não necessitariam digitar todas as informações novamente.

Desvantagens

1. Necessidade de a instituição possuir a Plataforma Lattes Institucional;
2. Possui alto acoplamento, pois qualquer atualização realizada no Currículo Lattes poderia refletir no sistema desenvolvido;
3. Conforme descrito anteriormente, possui pouca integração com as informações já mantidas pelos sistemas da instituição, o que poderia gerar inconsistências;
4. Usuários que não utilizam o Lattes teriam que passar a utiliza-lo;
5. A versão *on-line* para preenchimento do formulário não realiza envio das informações para a plataforma institucional e os docentes necessitariam instalar a versão *off-line* e posteriormente enviar o currículo.

7.3 AS INFORMAÇÕES SÃO OBTIDAS A PARTIR DOS SISTEMAS EXISTENTES NA INSTITUIÇÃO

Grande parte das informações necessárias para o preenchimento do RIA estão contidas nos sistemas da instituição e sua captura para utilização no relatório traria uma série de benefícios descritos a seguir:

Vantagens

1. Média complexidade para desenvolvimento da solução;
2. A utilização das informações contidas nos sistemas traria benefícios para instituição uma vez que possíveis erros e informações desatualizadas seriam constatadas pelos docentes e conseqüentemente corrigidas;
3. Maior confiabilidade nas informações pois já passaram pelas regras de validação do sistema de origem;
4. Os docentes não necessitariam digitar todas as informações novamente.

Desvantagens

1. Os sistemas acadêmicos da UFSC não possuem um cadastro integrado de docentes. Cada sistema possui cadastro próprio, o que dificulta a integração entre os mesmos;
2. Os sistemas da instituição não possuem todas as informações para o preenchimento do RIA. Seria necessário, portanto, altera-los ou extrair as informações de outras fontes.

8 TECNOLOGIAS

Esta seção detalha as principais arquiteturas e tecnologias utilizadas no desenvolvimento da proposta. Tais tecnologias envolvem recursos para o processamento no servidor e bancos de dados.

As tecnologias utilizadas para o desenvolvimento deste trabalho foram escolhidas de acordo com as propostas do governo para incentivo do uso de *software* livre.

8.1 APLICATIVOS WEB

Um aplicativo Web é uma aplicação servidora que atende a solicitações de clientes e é acessada através de *browsers* da Web. Ele pode ser simples como uma pesquisa de palavra-chave ou complexa com uma loja eletrônica virtual.

O desenvolvimento de aplicativos Web vem se popularizando a cada dia, tanto em *Intranet* como na Internet. A maior flexibilidade trazida pelas novas arquiteturas para o desenvolvimento de software como servidores de aplicação que separam a lógica de programação dos servidores de banco de dados impulsionam ainda mais o seu desenvolvimento.

Os paradigmas mais usados para o desenvolvimento de aplicativos Web são cliente-servidor tradicional em duas camadas (Figura 02) e o baseado em três camadas (Figura 03).

O paradigma em duas camadas consiste em delegar toda a carga de trabalho para o computador servidor que atua como um controlador de tráfego entre o cliente e os dados.

O baseado em três camadas divide a carga de trabalho entre as camadas existentes: a primeira fica responsável pela apresentação, a segunda pela lógica da aplicação e a terceira pelo armazenamento dos dados. A razão para se colocar uma camada intermediária entre o cliente e os dados é abstrair as formas de armazenamento da lógica da aplicação, sendo assim, a forma de armazenamento pode ser alterada mais facilmente.

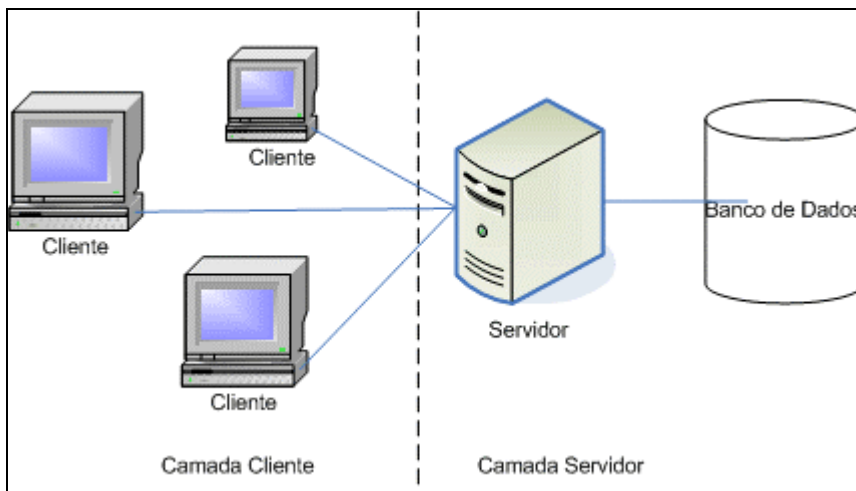


Figura 02 – Arquitetura 2 camadas

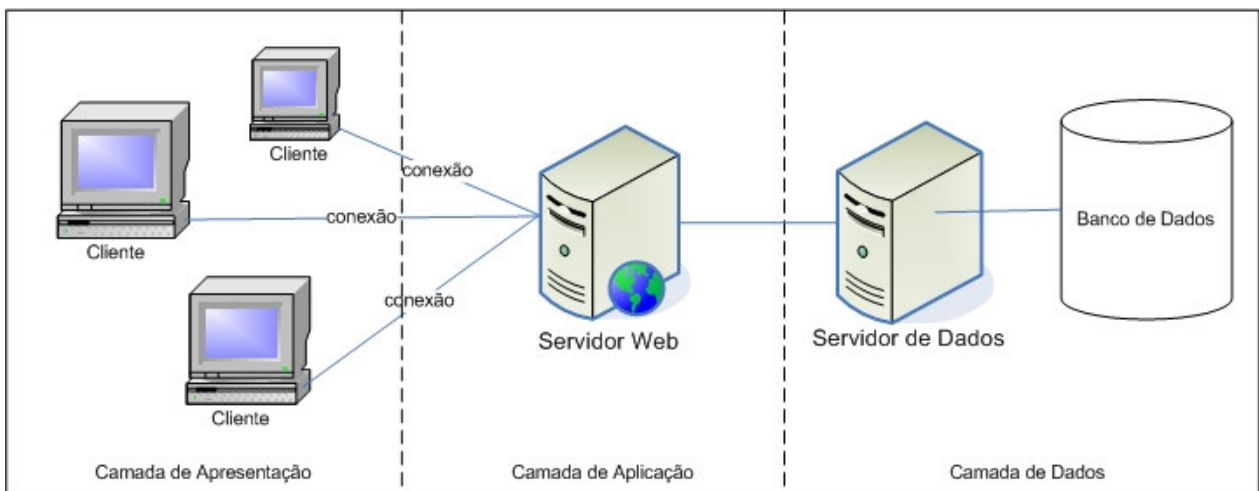


Figura 03 – Arquitetura 3 camadas

8.2 JAVA

Java é uma linguagem de programação muito conveniente para o desenvolvimento de *software* que funcione em conjunto com a Internet. É uma linguagem de programação orientada a objetos que utiliza uma metodologia que está se tornando cada vez mais útil no mundo do design de *software*. Além disso, é uma linguagem multiplataforma, o que significa que seus programas podem ser criados para executar do mesmo modo em diferentes sistemas operacionais.

Inicialmente, o entusiasmo pela Java girou em torno de suas capacidades para a montagem de aplicações incorporadas para a Web, chamadas *applets*. As *applets* poderiam ser programas independentes por si ou, então, *front-ends* sofisticados para programas executados em servidor.

Com Java 2, o interesse passou para outras áreas. A inclusão das classes *Swing* facilitou o desenvolvimento de interfaces gráficas com o usuário e permitiu que a Java se tornasse uma plataforma popular para o desenvolvimento de software aplicativo tradicional.

A Java também se tornou uma plataforma importante para as aplicações no lado do servidor, usando as tecnologias *servlet* e *jsp*, e para aplicações corporativas, usando tecnologias como *Enterprise Java Beans*. Java é a interface preferida para as modernas aplicações distribuídas.

Um dos desenvolvimentos mais importantes na programação para a plataforma Java é o JDBC (*Java Database Connectivity*). O JDBC se apresenta como uma solução elegante e eficiente para programadores que precisam conectar seus aplicativos Web a banco de dados de *back-end*.

8.3 JSP

JavaServer Pages (JSP) é uma tecnologia Java para desenvolver aplicativos Web. Com JSP, os *designers* da Web e programadores podem rapidamente incorporar elementos dinâmicos em páginas da Web usando Java embutido e algumas *tags* de marcação simples. A plataforma JSP permite ao desenvolvedor acessar dados e lógica comercial sem ter que dominar as complexidades do desenvolvimento de aplicações.

JSP foi lançada quando a tecnologia de servlet tinha atingido popularidade como uma das melhores tecnologias disponíveis. JSP, porém, não se destina a substituir os servlets. Na verdade, JSP é uma extensão da tecnologia de servlet, e é prática comum usar ambas, servlet e páginas JSP nos mesmos aplicativos Web.

JSP adapta-se as necessidades de desenvolvedores e organizações. Para uns, JSP é uma maneira simples de misturar código Java e texto HTML para produzir sites da Web dinâmicos. Para outros, JSP ajuda separar código Java do texto da apresentação, dando a não programadores uma maneira para produzir sites Web funcionais e dinâmicos. Por ser uma tecnologia Java, JSP é portátil, isto é, pode ser utilizado em diferentes sistemas operacionais.

8.4 BANCO DE DADOS

Banco de dados é uma estrutura em que os dados são armazenados para futura consulta. Essas estruturas organizam os dados e oferecem recursos de busca que facilitam a obtenção e a manutenção desses dados.

O método ou mecanismo organizado usado para manter os dados é referido como um sistema de gerenciamento de banco de dados (*Database Management System - DBMS*). O sistema gerenciador de banco é uma ferramenta computacional que, entre outras tarefas, automatiza a construção e o gerenciamento de um banco de dados a partir de seu modelo. Tais sistemas têm como características principais a criação, alimentação, alteração e manutenção de banco de dados.

A maioria dos sistemas de gerenciamento de banco de dados desenvolvidos recentemente utiliza o modelo relacional, devido ao fato de que, nestes sistemas, o acesso aos dados é mais rápido e eficiente, além de ser baseado em noções simples e intuitivas. No nível mais básico, um banco de dados relacional organiza os dados em tabelas. Essas tabelas são organizadas em linhas e colunas muito parecidas com uma planilha.

Atualmente são muitas as boas opções de sistemas de gerenciamento de banco de dados relacionais. Existem bancos de dados livres como o PostgreSQL e o MySQL, e comerciais, como Oracle, Informix, DB2 e Sybase, sendo estes mais voltados ao mercado corporativo. A escolha de cada tipo de banco de dados vai depender da necessidade de segurança, desempenho, escalabilidade da aplicação e das limitações financeiras.

Visando possibilitar maior integração com os sistemas utilizados na Universidade Federal de Santa Catarina o banco de dados Sybase foi utilizado para o desenvolvimento proposto neste

trabalho, pois ele é o responsável pelo armazenamento da maioria das informações dos sistemas na instituição.

9 DESENVOLVIMENTO

Iniciada a fase inicial do desenvolvimento do projeto, buscou-se verificar quais informações são necessárias para o preenchimento do RIA e a possibilidade de extração das mesmas dos sistemas na instituição visando que o processo se tornasse eficiente e conseqüentemente mais confiável, se comparado com uma nova digitação.

Durante essa etapa do trabalho foi realizado um levantamento das fontes de dados existentes na instituição que poderiam contribuir para o preenchimento do formulário.

9.1 FONTES DE INFORMAÇÃO

O levantamento e o mapeamento das informações necessárias para o preenchimento do formulário foi realizado através de reuniões com analistas de sistemas do Núcleo de Processamento de Dados (NPD) da Universidade Federal de Santa Catarina (UFSC).

Durante esse processo constatou-se que os sistemas existentes na instituição possuíam somente parte das informações necessárias para o preenchimento do formulário e seria necessário utilizar outras fontes de informação.

A partir desse momento, o Currículo LATTES passou a ser analisado, constatando-se que as informações não contempladas pela instituição poderiam ser retiradas do mesmo.

O resultado da análise e mapeamento das informações necessárias e das fontes de informação utilizadas pode ser observado no Anexo 1 e no gráfico a abaixo:

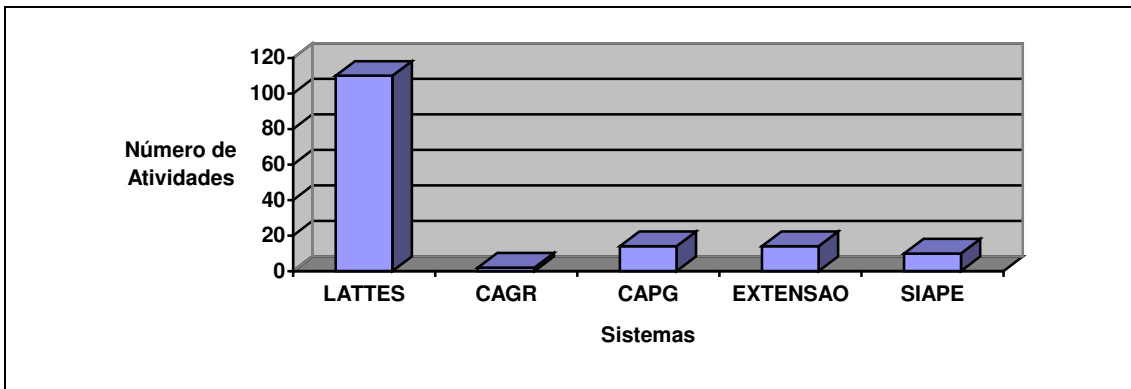


Figura 04 - Contribuição dos sistemas para o preenchimento do formulário GED.

Após o término do levantamento e mapeamento das informações evidenciou-se que a grande parte das informações necessárias para o preenchimento do formulário não estão presentes nos sistemas da instituição e seria necessário utilizar a plataforma LATTES em conjunto.

A decisão de retirar as informações necessárias para o preenchimento do formulário dos sistemas existentes na instituição trouxe diversos benefícios, tal como a maior confiabilidade nas informações, mas aumentou a complexidade do sistema proposto.

Logo abaixo são listadas as dificuldades encontradas e as respectivas soluções:

1. A UFSC não possui um cadastro integrado de pessoas e não existe padronização para identificação dos docentes nos sistemas, o que dificultou a identificação dos docentes e a relação entre os sistemas. Sendo assim foram descartados os docentes que não estavam cadastrados no sistema Sistema Integrado de Administração de Recursos Humanos (SIAPE) do governo federal e o cadastro de pessoa física (CPF) que consta no SIAPE foi utilizado para identificação dos docentes.
2. Os dados existentes nos sistemas são voláteis, isto é, uma vez coletada a informação de um sistema ela pode ser atualizada no mesmo e os dados extraídos

estariam desatualizados. Verificado esse problema, foi criada uma base de dados que chamamos de intermediária, onde os dados coletados seriam armazenados.

3. Após a coleta de dados, se as informações coletadas estivessem erradas como o professor realizaria a alteração? Neste caso, o professor deve atualizar as informações no sistema onde de onde foi extraída e uma nova coleta de dados seria realizada para todos os professores que apresentaram o mesmo problema.

9.2 DEFINIÇÃO DOS ATORES

A partir da análise do funcionamento da aplicação, foram identificados quatro atores para o sistema. O administrador do sistema, o professor, a pessoa designada pelo departamento de ensino (Departamento) para análise das informações e a pessoa designada pela instituição (Instituição) para análise final.

9.3 CASO DE USO

1. Extração das informações
2. Autenticar professor
3. Autenticar departamento
4. Autenticar instituição
5. Aprovar formulário (Professor)
6. Aprovar formulário (Departamento)
7. Aprovar formulário (Instituição)

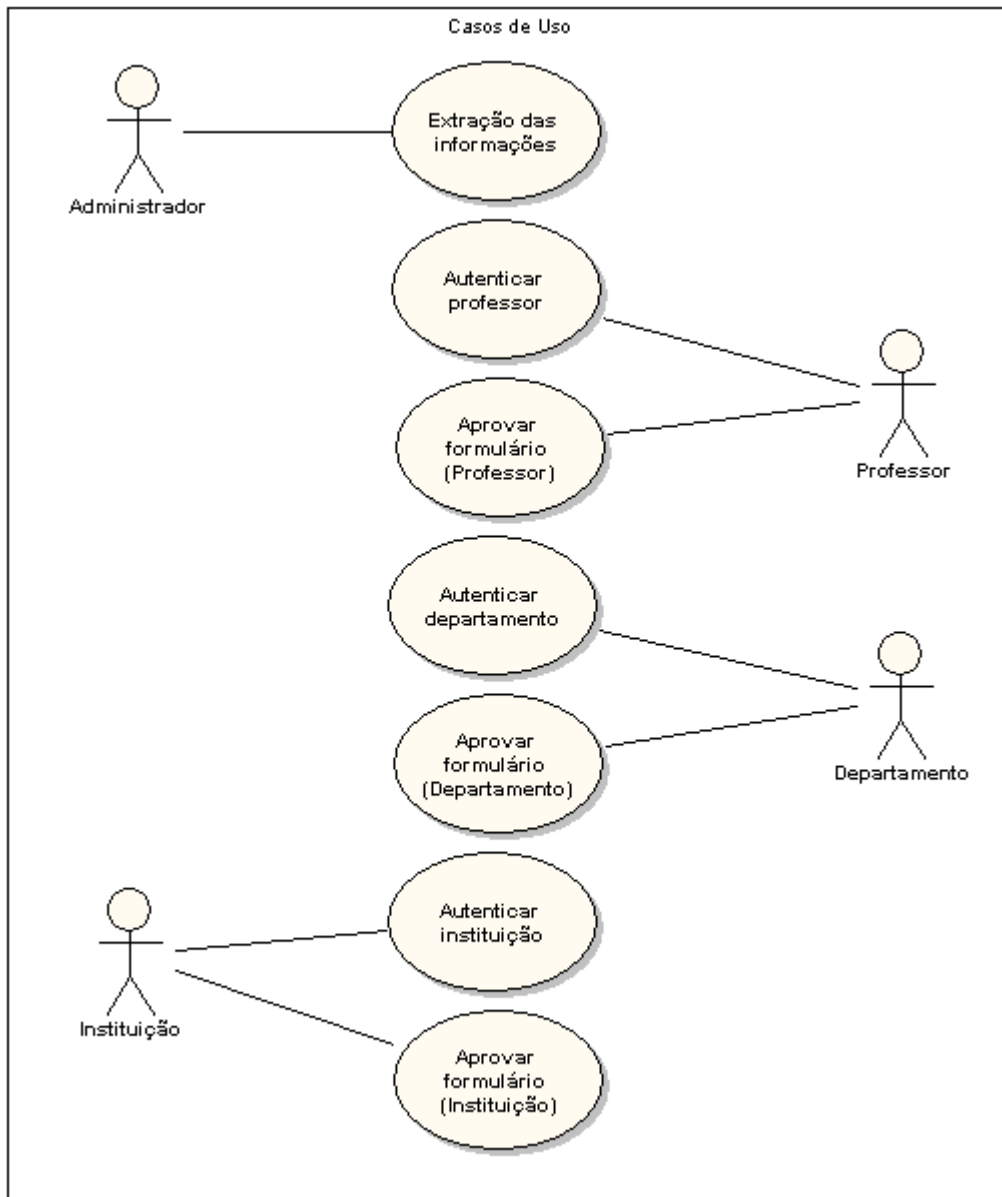


Figura 05 – Diagrama de Casos de Uso

Caso de uso:	Extração das informações
Atores:	Administrador.
Finalidade:	Alimentar a fonte de dados intermediária com as atividades realizadas pelo professor
Visão Geral:	O Administrador acessa o sistema, informa o período que deseja extrair os dados e pressiona o botão executar.

Tipo:	Principal e essencial.
-------	------------------------

Tabela 1 – Caso de Uso: Extração das informações

Seqüência Típica de Eventos:

1. O Administrador informa o período que deseja e pressiona executar.
2. O sistema consulta as fontes de informação e alimenta a fonte de dados temporária.

Caso de uso:	Autenticar Professor
Atores:	Professor
Finalidade:	Autenticar um professor para obter seu formulário
Visão Geral:	O professor digita seu nome de usuário e senha e pressiona o botão “Autenticar” para fazer a autenticação.
Tipo:	Principal e essencial.

Tabela 2 – Caso de uso: Autenticar Professor

Seqüência Típica de Eventos:

1. O professor abre o formulário para fazer autenticação, digita usuário e senha e clica no botão “Autenticar”.
2. O sistema verifica se o usuário e senha conferem no banco de dados, e se esse usuário está ativo para acessar o sistema.
3. Em caso positivo de autenticação, o sistema redireciona para a página inicial da área restrita do usuário.

Seqüências alternativas:

1. Caso o professor não esteja ativo no sistema, ou usuário e senha estejam inválidos, o sistema retorna para a tela de autenticação informando a mensagem de erro.

Caso de uso:	Autenticar Departamento
Atores:	Departamento
Finalidade:	Autenticar o departamento para obter a lista de professores.
Visão Geral:	O usuário digita seu nome de usuário e senha, e pressiona o botão “Autenticar” para fazer a autenticação.
Tipo:	Principal e essencial.

Tabela 3 – Caso de uso: Autenticar Departamento

Seqüência Típica de Eventos:

2. O usuário abre o formulário para fazer autenticação, digita usuário e senha e clica no botão “Autenticar”.
3. O sistema verifica se o usuário e senha conferem no banco de dados, e se esse usuário está ativo para acessar o sistema.
4. Em caso positivo de autenticação, o sistema redireciona para a página inicial da área restrita do usuário.

Seqüências alternativas:

1. Caso o usuário não esteja ativo no sistema, ou usuário e senha estejam inválidos, o sistema retorna para a tela de autenticação informando a mensagem de erro.

Caso de uso:	Autenticar Instituição
Atores:	Instituição
Finalidade:	Autenticar a instituição para obter a lista de departamento.
Visão Geral:	O usuário digita seu nome de usuário e senha e

	pressiona o botão “Autenticar” para fazer a autenticação.
Tipo:	Principal e essencial.

Tabela 4 – Caso de uso: Autenticar Instituição

Seqüência Típica de Eventos:

2. O usuário abre o formulário para fazer autenticação, digita usuário e senha e clica no botão “Autenticar”.
3. O sistema verifica se o usuário e senha conferem no banco de dados, e se esse usuário está ativo para acessar o sistema.
4. Em caso positivo de autenticação, o sistema redireciona para a página inicial da área restrita do usuário.

Seqüências alternativas:

1. Caso o usuário não esteja ativo no sistema, ou usuário e senha estejam inválidos, o sistema retorna para a tela de autenticação informando a mensagem de erro.

Caso de uso:	Aprovar Formulário (Professor)
Atores:	Professor
Finalidade:	Aprovar as informações contidas no formulário
Visão Geral:	O usuário visualiza o formulário e pressiona o botão “Aprovado”
Tipo:	Principal e essencial.

Tabela 5 – Caso de uso: Aprovar Formulário (Professor)

Seqüência Típica de Eventos:

2. O usuário abre o formulário visualiza as informações e pressiona o botão “Aprovado”.

3. O sistema altera a situação do formulário para aprovado e altera a etapa do formulário para “Aguardando Departamento”.

Seqüências alternativas:

1. Caso as informações não sejam aprovadas pelo usuário, o mesmo pressiona o botão “Rejeitado”.
2. O sistema altera a situação do formulário para rejeitado e exibe o formulário para preenchimento do motivo.
3. O usuário preenche o motivo da rejeição e pressiona o botão “Salvar”.

Caso de uso:	Aprovar Formulário (Departamento)
Atores:	Departamento
Finalidade:	Aprovar as informações contidas no formulário
Visão Geral:	O usuário visualiza o formulário e pressiona o botar “Aprovado”
Tipo:	Principal e essencial.

Tabela 6 – Caso de uso: Aprovar Formulário (Departamento)

Seqüência Típica de Eventos:

1. O usuário abre o formulário visualiza as informações e pressiona o botão “Aprovado”.
2. O sistema altera a situação do formulário para aprovado e altera a etapa do formulário para “Aguardando Instituição”.

Seqüências alternativas:

1. Caso as informações não sejam aprovadas pelo usuário, o mesmo pressiona o botão “Rejeitado”.
2. O sistema altera a situação do formulário para rejeitado e altera a etapa para “Aguardando Professor” e exibe o formulário para preenchimento do motivo.

3. O usuário preenche o motivo da rejeição e pressiona o botão “Salvar”.

Caso de uso:	Aprovar Formulário (Instituição)
Atores:	Instituição
Finalidade:	Aprovar as informações contidas no formulário
Visão Geral:	O usuário visualiza o formulário e pressiona o botar “Aprovado”
Tipo:	Principal e essencial.

Tabela 7 – Caso de uso: Aprovar Formulário (Instituição)

Seqüência Típica de Eventos:

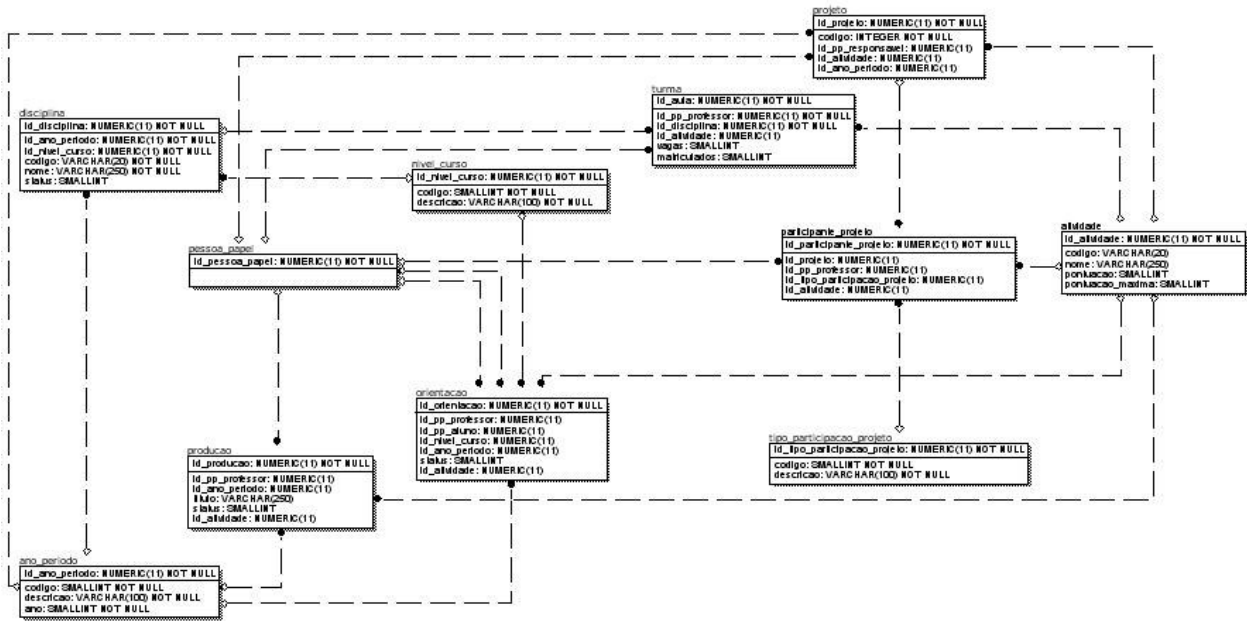
1. O usuário abre o formulário visualiza as informações e pressiona o botão “Aprovado”.
2. O sistema altera a situação do formulário para aprovado.

Seqüências alternativas:

1. Caso as informações não sejam aprovadas pelo usuário o mesmo pressiona o botão “Rejeitado”.
2. O sistema altera a situação do formulário para rejeitado, altera a etapa para “Aguardando Departamento” e exhibe o formulário para preenchimento do motivo.
3. O usuário preenche o motivo da rejeição e pressiona o botão “Salvar”.

9.3.1 BANCO DE DADOS

A figura 06 mostra o diagrama do banco de dados proposto.



10 CONSIDERAÇÕES FINAIS

10.1 TRABALHOS FUTUROS

A proposta desenvolvida neste trabalho é baseada nas informações disponibilizadas pelos servidores e professores da UFSC e uma possível implementação futura é o levantamento dos requisitos de outras instituições e o desenvolvimento de um modelo capaz de suporta-los, assim, o protótipo poderia dar origem a um sistema capaz de atender a necessidade de outras instituições.

A elaboração de um cadastro único com as informações dos docentes é um trabalho futuro que poderá auxiliar na identificação das atividades realizadas pelos docentes da instituição e diminuir o esforço para integração das informações.

10.2 CONCLUSÃO

Neste projeto foram apresentados aspectos teóricos e práticos que tiveram como objetivo automatizar o preenchimento de formulários pelos docentes, especialmente, o Relatório Individual de Atividades.

Durante o desenvolvimento deste trabalho observou-se diversas dificuldades para automatizar o preenchimento do formulário RIA na UFSC, a seguir algumas delas:

1. Os sistemas possuem diferente padronização para identificação dos professores na instituição, o que dificulta a integração entre eles;
2. Grande parte das informações necessárias para o preenchimento dos formulários, tais como as atividades de produção intelectual, não é informada para a instituição pelos docentes;
3. Algumas informações constam nos sistemas mas só são atualizadas quando convém ao docente ou à administração da universidade, o que exige atualização das informações e conscientização de todos antes da implantação de sistemas que utilizam tais informações.

Com a elaboração do protótipo concluiu-se que é possível preencher o RIA através da captura das informações dos diversos sistemas já citados, que atividades realizadas hoje pelos servidores e docentes da UFSC podem ser automatizadas visando diminuir o tempo necessário para sua conclusão e que a captura de tais informações poderá trazer benefício para os sistemas da instituição.

11 REFERÊNCIAS BIBLIOGRÁFICAS

BRASIL. Decreto n. 94.664, de 23 de julho de 1987. Classificação e Retribuição de Cargos e Empregos. **Presidência da República**, Brasília. Disponível em: <<http://www.planalto.gov.br>>. Acesso em: 8 outubro de 2005.

_____. Portaria n. 475, de 26 de agosto de 1987. Normas Complementares para a execução do Decreto nº 94.664, de 23 de julho de 1987. **Presidência da República**, Brasília. Disponível em: <<http://www.planalto.gov.br>>. Acesso em: 8 outubro de 2005.

FIELDS, D. K.; KOLB, M. A.; BAYERN, S. **Web development with JavaServer Pages. 2.** ed. Greenwich, CT: Manning Publications Co., 2002.

HUNTER, J.; CRAWFORD, W. **Java servlet programming.** Sebastopol, CA: O'Reilly & Associates, 2001.

KURNIAWAN, B. **Java for the Web with Servlets, JSP, and EJB: a developer's guide to J2EE Solutions.** 1. ed. Indianapolis: New Riders, 2002.

LEMAY, L.; CADENHEAD, R. **Sams teach yourself Java 2 in 21 days.** Indianapolis, IN: Sams Publishing, 1999.

MINISTÉRIO DA EDUCAÇÃO. **Orientações Gerais – GED.** Brasília. Disponível em: <<http://www.mec.gov.br/sesu/gedorienta.shtm>>. Acesso em: 5 junho 2004.

LATTES. Conheça o Lattes. Disponível em: <<http://lattes.cnpq.br>>. Acesso em: 11 novembro de 2005.

NIEMEYER, P; KNUDSEN, J. **Learning Java.** Sebastopol, CA: O'Reilly & Associates, 2000.

ANEXO 1 - MAPEAMENTO DAS ATIVIDADES UTILIZADAS PARA PREENCHIMENTO
DO RIA

1. Atividades de Ensino

1.1 Aulas do primeiro semestre

Atividade	Fonte
Graduação	CAGR
Especialização	CAPG
Mestrado / Doutorado	CAPG

1.2 Aulas do segundo semestre

Atividade	Fonte
Graduação	CAGR
Especialização	CAPG
Mestrado / Doutorado	CAPG

1.3 Outras Atividades Curriculares

Atividade	Natureza	Fonte
Em Andamento		
Supervisão de Estágio Curricular	-	LATTES
Orientação de Trabalho Final de Curso de Graduação	-	LATTES
Orientação de Trabalho Final de Curso de Especialização	-	CAPG
Orientação de Dissertação de Mestrado	orientador	CAPG
	co-orientador	CAPG
Orientação de Tese de Doutorado	orientador	CAPG
	co-orientador	CAPG
Orientação Acadêmica	-	LATTES
Coordenação de Matéria / Disciplina	-	LATTES
Supervisão ambulatorial	-	LATTES
Internato	-	LATTES
Comissão de Avaliação de Trabalho de Ateliê	-	LATTES
Concluídas		
Orientação de Trabalho Final de Curso de Graduação	-	LATTES
Orientação de Trabalho Final de Curso de Especialização	-	CAPG
Orientação de Dissertação de Mestrado	orientador	CAPG
	co-orientador	CAPG
Orientação de Tese de Doutorado	orientador	CAPG
	co-orientador	CAPG

2. Produção Intelectual

2.1 Produção Bibliográfica

Tipo	Natureza	Fonte
Artigo Publicado em Periódico	Completo	LATTES
	Resumo	LATTES
Livro ou Capítulo de Livro	Capítulo	LATTES
	Organização	LATTES
	Livro	LATTES
Trabalho Publicado em Anais de Evento Técnico-Científico	Completo	LATTES
	Resumo	LATTES
Textos em Jornal ou Revista (magazines)	Jornal de Notícias	LATTES
	Revista	LATTES
Demais tipos	Partitura Musical	LATTES
	Tradução Livro	LATTES
	Tradução Artigo	LATTES
	Outro	LATTES

2.2 Produção Artística

Tipo	Natureza	Fonte
Apresentação de obra artística	Coreográfica	LATTES
	Literária	LATTES
	Musical	LATTES
	Teatral	LATTES
	Outra	LATTES
Apresentação em rádio ou TV	Dança	LATTES
	Teatro	LATTES
	Música	LATTES
	Outra	LATTES
Arranjo musical	Canto	LATTES
	Coral	LATTES
	Orquestra	LATTES
	Outra	LATTES
Composição musical	Canto	LATTES
	Coral	LATTES
	Orquestra	LATTES
	Outra	LATTES
Obra de artes visuais	Cinema	LATTES
	Desenho	LATTES

	Escultura	LATTES
	Fotografia	LATTES
	Gravura	LATTES
	Instalação	LATTES
	Pintura	LATTES
	Televisão	LATTES
	Vídeo	LATTES
	Outra	LATTES
Sonoplastia	Cinema	LATTES
	Música	LATTES
	Rádio	LATTES
	Televisão	LATTES
	Teatro	LATTES
	Outra	LATTES
Outra produção artística/ cultural		LATTES
Premiação		LATTES

2.3 Produção Técnica

Tipo	Natureza	Fonte
Software	Computacional	LATTES
	Multimídia	LATTES
	Outra	LATTES
Produto tecnológico <i>(aparelho, equipamento, fármacos e similares, instrumentos e outros)</i>	Piloto	LATTES
	Projeto	LATTES
	Protótipo	LATTES
	Outro	LATTES
Processo ou Técnica	Analítica	LATTES
	Instrumental	LATTES
	Pedagógica	LATTES
	Processual	LATTES
	Terapêutica	LATTES
	Outra	LATTES
Propriedade intelectual <i>(processo ou técnica, produto tecnológico e software)</i>	Patente/ registro	LATTES
Demais produções técnicas	Desenvolvimento de material didático e instrucional publicado	LATTES
	Editoração	LATTES
	Maquete	LATTES
	Manutenção de obra artística <i>(arquitetura, desenho, fotografia, escultura, gravura, pintura, acervos bibliográficos e arquivísticos históricos)</i>	LATTES
	Apresentação em reuniões científicas <i>(conferência, palestra ou comunicação)</i>	LATTES
	Carta, mapa ou similar	LATTES
	Outra	LATTES

3. Atividades de Pesquisa e de Extensão

3.1 Pesquisa

Tipo	Natureza	Fonte
Projeto aprovado	Coordenação	LATTES
	Participação	LATTES
Relatório aprovado	Parcial	LATTES
	Final	LATTES

3.2 Extensão

Natureza	Função	Fonte
Projeto aprovado		
Programa permanente	Coordenação / autoria	EXTENSÃO
	Participação	EXTENSÃO
Projeto em comunidade	Coordenação	EXTENSÃO
	Participação	EXTENSÃO
Relatório aprovado		
Programa permanente	Coordenação	EXTENSÃO
	Participação	EXTENSÃO
Cursos (mínimo de 20 horas)	Coordenação	EXTENSÃO
	Docência	EXTENSÃO
Eventos	Coordenação	EXTENSÃO
	Participação em equipe organizadora	EXTENSÃO
Serviços	Coordenação	EXTENSÃO
	Participação	EXTENSÃO
Projetos em comunidades	Coordenação	EXTENSÃO
	Participação	EXTENSÃO

4. Atividades de qualificação

4.1 Em andamento

Tipo	Liberação das Atividades de Ensino	Fonte
Mestrado	Total	LATTES
	Parcial	LATTES
	Sem liberação	LATTES
Doutorado	Total	LATTES
	Parcial	LATTES
	Sem liberação	LATTES
Pós-Doutorado	Total	LATTES

4.2 Concluída

Tipo	Natureza	Fonte
Especialização	Monografia	SIAPE
Mestrado	Dissertação	SIAPE
Doutorado	Tese	SIAPE
Pós-Doutorado	Relatório	SIAPE

5. Atividades Administrativas e de Representação

5.1 Atividades Administrativas e de Representação

Atividade	Fonte
Membro dos Conselhos Superiores e da CPPD	LATTES
Membros de outros órgãos colegiados	LATTES
Membro de diretoria de associação científica, agência de fomento ou órgão de classe	LATTES
Presidente de Câmara	LATTES
Membro de Conselho Editorial e Consultivo	LATTES
Representação Sindical Docente	LATTES
Participação em conselhos ou comissões de órgãos governamentais e de entidades científicas e culturais	LATTES
Coordenação de Curso de Especialização	CAPG
Comissões e Coordenações Especiais por Portaria de Âmbito Institucional	LATTES
Comissão de Avaliação Interna e/ou de Reformulação curricular	LATTES

5.2 Atividades Administrativas - Cargos de Direção ou Funções Gratificadas

Atividade	Fonte
FG 1 e 2	SIAPE
CD 1 a 4	SIAPE
Vice-Diretor de unidade de ensino	SIAPE
Presidente da CPPD	LATTES
Substituto de cargo FG 1 e 2 em exercício	SIAPE
Substituto de CD 1 a 4 em exercício	SIAPE

6. Outras Atividades

Tipo	Natureza	Fonte
Orientação	Bolsista de iniciação científica (em andamento)	LATTES
	Bolsista de iniciação científica (concluída)	LATTES
	Outras (Bolsista PET; Monitor; Bolsista trabalho; Estudante estrangeiro de instituição conveniada; Supervisão de estágio extra curricular)	LATTES
Participação em bancas	Defesa de dissertação	LATTES
	Defesa de tese	LATTES
	Exame de qualificação de mestrado ou doutorado	LATTES
	Defesa de trabalho final de curso de graduação	LATTES
	Defesa de trabalho final de curso de especialização	LATTES
	Concursos para docentes do quadro permanente	LATTES
	Outras (Seleção de professores Substitutos; Seleção de candidatos a residência médica, especialização, mestrado ou doutorado; Seleção de bolsistas; progressão funcional; estágio probatório)	LATTES

7. Afastamentos

7.1 Afastamentos com ônus (exceto para qualificação)

Natureza	Fonte
Licença prêmio	SIAPÉ
Outros	SIAPÉ

7.2 Outros Afastamentos (do Estado ou País)

Natureza	Fonte
Curso de especialização	SIAPÉ
Estágio/ intercâmbio	SIAPÉ

ANEXO 2 – CODIGO FONTE

Arquivo: ged/ged/index.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>index</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="lib/stylesheet/ged.css" rel="stylesheet" type="text/css">
</head>

<body onLoad="">
<br><br>

<table width="200" border="0" align="center" cellspacing="5">
<tr>
<td class="link"><a href="faces/loginDocente.jsp">Docente</a></td>
</tr>
<tr>
<td class="link"><a href="faces/loginDepartamento.jsp">Departamento</a></td>
</tr>
<tr>
<td class="link"><a href="faces/loginInstituicao.jsp">Instituição</a></td>
</tr>
</table>

</body>
</html>
```

Arquivo: ged/ged/lib/javascript/arvore.js

```
//Atribuição de eventos
function inittree(){
    var uls = document.getElementsByTagName("ul")
    for(i = 0; i < uls.length; i++)
        if(uls[i].className == "treelist"){
            var lis = uls[i].childNodes
            for(k = 0; k < lis.length; k++)
                if(lis[k].nodeType == 1) {
```

```

        if(lis[k].getElementsByTagName("ul").length > 0){
            if (lis[k].className == "atv") {
                lis[k].className = "atvclose"

            }
            if (lis[k].className == "noatv"){
                lis[k].className = "noatvclose"
            }
            chi = lis[k].childNodes
            addEvent(lis[k].childNodes[0], "click", clicado)
        }
    }
}

//Abre/fecha quando clicado
function clicado(e){
    var source = getSource(e)
    var li = source.parentNode
    if (li.className == "atvclose" || li.className == "atvopen"){
        li.className = (li.className == "atvclose") ? "atvopen" : "atvclose"
    } else if (li.className == "noatvclose" || li.className == "noatvopen"){
        li.className = (li.className == "noatvclose") ? "noatvopen" : "noatvclose"
    }
    return false
}

```

Arquivo: ged/ged/lib/stylesheet/arvore_minimal.css

```

ul.treelist,ul.treelist ul{
    margin:0;
    padding:0;
}

ul.treelist li{
    margin:0;
    padding:0 0 0 30px;
    list-style-type:none;
}

ul.treelist li.fechado,ul.treelist li.aberto{
    margin:0 0 0 30px;
}

```

```
padding:0;
list-style-type:circle;
}
```

```
ul.treelist li{
font-family:arial;
line-height:20px;
}
```

```
ul.treelist li a{
color:#00F;
font-weight:normal;
}
```

```
ul.treelist li a:hover{
color:#F00;
}
```

```
ul.treelist li.aberto ul{
display:block;
}
```

```
ul.treelist li.fechado ul{
display:none;
}
```

Arquivo: ged/ged/lib/stylesheet/arvore_windows.css

```
ul.treelist,ul.treelist ul{
margin:0;
padding:0;
}
```

```
ul.treelist li{
margin:0;
padding:0 0 0 30px;
list-style-type:none;
}
```

```
ul.treelist li{
background-position:10px 1px;
background-repeat:no-repeat;
```

```
        font-family:arial;
        line-height:20px;
    }

    ul.treelist li.atv {
    }

    ul.treelist li.noatv {
    }

    ul.treelist li.atvclose{
        background-image:url(../images/atvclose.gif);
    }

    ul.treelist li.atvopen{
        background-image:url(../images/atvopen.gif);
    }

    ul.treelist li.noatvopen{
        background-image:url(../images/noatvopen.gif);
    }
    ul.treelist li.noatvclose{
        background-image:url(../images/noatvclose.gif);
    }

    ul.treelist li.atividade{
        background-image:url(../images/atividade.gif);
    }

    ul.treelist li.nochildren{
        background-image:url(../images/nochildren.gif);
    }

    ul.treelist li a{
        color:#009;
    }

    ul.treelist li a:hover{
        color:#F00;
    }
```

```
ul.treelist li.atvopen ul{
    display:block;
}
```

```
ul.treelist li.noatvopen ul{
    display:block;
}
```

```
ul.treelist li.atvclose ul{
    display:none;
}
```

```
ul.treelist li.noatvclose ul{
    display:none;
}
```

Arquivo: ged/ged/listaDepartamentos.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<html>
    <head>
        <title>GED - Lista de Departamentos</title>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <link href="lib/stylesheet/ged.css" rel="stylesheet" type="text/css">
    </head>
    <body>

        <f:view>
            <h:form>

                <div align="right"><h:commandLink action="#{ListaDepartamentosBean.sairSessao}"
styleClass="link"><h:outputText value="sair" styleClass="link" /></h:commandLink></div><br>
                <br><br>

                <h:outputText id="inst" value="#{ListaDepartamentosBean.instituicao.nome}" styleClass="titulo"
/><br>

                <span class="titulo2">Departamentos</span>

                <br><br><br>
```

```

        <h:dataTable                                id="departamentos"
binding="#{ListaDepartamentosBean.DTDepartamentos}"    value="#{ListaDepartamentosBean.departamentos}"
var="departamento" style="lib/ged.css" rowClasses="tpAtv" width="60%" border="1" cellpadding="0" cellspacing="0">

        <h:column>
            <f:facet name="header">
                <h:outputText value="Código" styleClass="cabecalho" />
            </f:facet>
            <h:outputText        value=" #{departamento.idDepartamento}"
styleClass="dado" />
        </h:column>

        <h:column>
            <f:facet name="header">
                <h:outputText value="Nome" styleClass="cabecalho" />
            </f:facet>
            <h:outputText        value=" #{departamento.nome}"
styleClass="dado" />
        </h:column>
        <h:column>
            <f:facet name="header">
                <h:outputText value="Visualização" styleClass="cabecalho" />
            </f:facet>
            <h:commandLink
action="#{ListaDepartamentosBean.selectDepartamento}"    styleClass="link"><h:outputText    value=" Visualizar"
styleClass="link" /></h:commandLink>
        </h:column>
    </h:dataTable>
</h:form>
</f:view>
</body>
</html>

Arquivo: ged/ged/listaDocentes.jsp

<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
    <f:view>
        <h:form>
<html>
    <head>

```



```

<title>GED - <h:outputText value="#{ListaDocentesBean.departamento.nome}" /> - Lista
de Docentes</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="lib/stylesheet/ged.css" rel="stylesheet" type="text/css">
</head>
<body>
<div align="right"><h:commandLink action="#{ListaDocentesBean.sairSessao}"
styleClass="link"><h:outputText value="sair" styleClass="link" /></h:commandLink></div><br>
<br><br>

<h:outputText id="dpto" value="#{ListaDocentesBean.departamento.nome}" styleClass="titulo"
/><br>
<span class="titulo2">Docentes Lotados</span>

<br><br><br>

<h:dataTable id="docentes"
binding="#{ListaDocentesBean.DTDocentes}" value="#{ListaDocentesBean.docentes}" var="docente"
style="lib/ged.css" rowClasses="tpAtv" width="80%" border="1" cellpadding="0" cellspacing="0">

<h:column>
<f:facet name="header">
<h:outputText value="Código GED" styleClass="cabecalho" />
</f:facet>
<h:outputText value=" #{docente.id_docente}"
styleClass="dado" />
</h:column>

<h:column>
<f:facet name="header">
<h:outputText value="Código Lattes" styleClass="cabecalho" />
</f:facet>
<h:outputText value=" #{docente.idLattes}" styleClass="dado" />
</h:column>

<h:column>
<f:facet name="header">
<h:outputText value="Nome" styleClass="cabecalho" />
</f:facet>
<h:outputText value=" #{docente.nome}" styleClass="dado" />

```

```

        </h:column>

        <h:column>
        <f:facet name="header">
            <h:outputText value="Status" styleClass="cabecalho" />
        </f:facet>
            <h:graphicImage alt="status" width="15" height="15"
url="#{docente.statusIcon}" />
        </h:column>
        <h:column>
            <f:facet name="header">
                <h:outputText value="Visualização" styleClass="cabecalho" />
            </f:facet>
                <h:commandLink action="#{ListaDocentesBean.selectDocente}"
styleClass="link"><h:outputText value=" Visualizar" styleClass="link" /></h:commandLink>
        </h:column>
    </h:dataTable>
</body></html></h:form></f:view>

```

Arquivo: ged/ged/loginDepartamento.jsp

```

<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<html>
    <head>
        <title>GED - Login Departamento</title>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <link href="lib/stylesheet/ged.css" rel="stylesheet" type="text/css">
    </head>
    <body onLoad="_id0.reset()">
        <br><br>
        <f:view>
            <h:form>
                <table border="0" cellspacing="2" cellpadding="0" align="center">
                    <tr>
                        <td align="center" class="titulo2" colspan="2">Login
Departamento</td>
                    </tr>

```

```

        <tr>
            <td height="15" colspan="2"></td>&nbsp;</td>
        </tr>
        <tr>
            <td class="campo_form">Usuário&nbsp;</td>
            <td align="center"><h:inputText id="loginDpto"
value="#{LoginDepartamentoBean.username}" tabindex="1" styleClass="input_text"/> </td>
        </tr>
        <tr>
            <td class="campo_form">Senha&nbsp;</td>
            <td align="center"><h:inputSecret id="senhaDpto"
value="#{LoginDepartamentoBean.senha}" tabindex="2" styleClass="input_text"/></td>
        </tr>
        <tr><td height="4"></td></tr>
        <tr>
            <td colspan="2" align="right">
                <h:commandButton id="botaoDpto" value="OK"
action="#{LoginDepartamentoBean.login}" tabindex="3" type="submit" styleClass="input_botao"/>
            </td>
        </tr>
    </table>
</h:form>
</f:view>
</body>
</html>

```

Arquivo: ged/ged/loginDocente.jsp

```

<%@ taglib uri="http://java.sun.com/jsp/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsp/core" prefix="f" %>
<html>
    <head>
        <title>GED - Login Docente</title>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <link href="lib/stylessheet/ged.css" rel="stylesheet" type="text/css">
    </head>
    <body onLoad="_id0.reset()">
        <br><br>
        <f:view>
            <h:form>
                <table border="0" cellspacing="2" cellpadding="0" align="center">
                    <tr>

```

```

Docente</td>
        <td align="center" class="titulo2" colspan="2">Login
Docente</td>
    </tr>
    <tr>
        <td height="15" colspan="2"></td>&nbsp;</td>
    </tr>
    <tr>
        <td class="campo_form">Usuário&nbsp;</td>
        <td align="center"><h:inputText id="loginDocente"
value="#{LoginDocenteBean.username}" tabindex="1" styleClass="input_text"/> </td>
    </tr>
    <tr>
        <td class="campo_form">Senha&nbsp;</td>
        <td align="center"><h:inputSecret id="senhaDocente"
value="#{LoginDocenteBean.senha}" tabindex="2" styleClass="input_text"/> </td>
    </tr>
    <tr><td height="4"></td></tr>
    <tr>
        <td colspan="2" align="right">
            <h:commandButton id="botaoDocente" value="OK"
action="#{LoginDocenteBean.login}" tabindex="3" type="submit" styleClass="input_botao" />
        </td>
    </tr>
</table>
</h:form>
</f:view>
</body>
</html>

```

Arquivo: ged/ged/loginInstituicao.jsp

```

<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<html>
    <head>
        <title>GED - Login Instituição</title>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <link href="lib/stylesheet/ged.css" rel="stylesheet" type="text/css">
    </head>
    <body onLoad="_id0.reset()">

```

```

<br><br>
    <f:view>
        <h:form>
            <table border="0" cellspacing="2" cellpadding="0" align="center">
                <tr>
                    <td align="center" class="titulo2" colspan="2">Login
Instituição</td>
                </tr>
                <tr>
                    <td height="15" colspan="2"></td>&nbsp;</td>
                </tr>
                <tr>
                    <td class="campo_form">Usuário&nbsp;</td>
                    <td align="center"><h:inputText id="loginDpto"
value="#{LoginInstituicaoBean.username}" tabindex="1" styleClass="input_text"/> </td>
                </tr>
                <tr>
                    <td class="campo_form">Senha&nbsp;</td>
                    <td align="center"><h:inputSecret id="senhaDpto"
value="#{LoginInstituicaoBean.senha}" tabindex="2" styleClass="input_text" /></td>
                </tr>
                <tr><td height="4"></td></tr>
                <tr>
                    <td colspan="2" align="right">
                        <h:commandButton id="botaoDpto" value="OK"
action="#{LoginInstituicaoBean.login}" tabindex="3" type="submit" styleClass="input_botao"/>
                    </td>
                </tr>
            </table>
        </h:form>
    </f:view>
</body>
</html>

Arquivo: ged/ged/save_ok.jsp

<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<html>
    <head>
        <title>GED - Dados Processados</title>

```

```

        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
        <link href="lib/stylesheet/ged.css" rel="stylesheet" type="text/css">
    </head>
    <body>
    <f:view><h:form>
    <div align="center">
        <br><br>
        <span class="titulo2">Dados salvos com sucesso.</span><br><br>
        <h:commandLink
                                action="#{SaveCompleteBean.sairSessao}"
styleClass="link"><h:outputText value="sair" styleClass="link" /></h:commandLink>
        <br><br>
    </div>
    </h:form></f:view>
    </body>
</html>

```

Arquivo: ged/src/br/ufsc/ine5328/Constantes.java

```
package br.ufsc.ine5328;
```

```
public class Constantes {
```

```

    public static final String DOCENTE_CORRENTE = "docenteCorrente";
    public static final String DEPARTAMENTO_CORRENTE = "departamentoCorrente";
    public static final String INSTITUICAO_CORRENTE = "instituicaoCorrente";

```

```

    public static final String LOGIN_DOCENTE_OK = "docente_success";
    public static final String LOGIN_DOCENTE_ERRO = "docente_failure";

```

```

    public static final String LOGIN_DEPARTAMENTO_OK = "dpto_success";
    public static final String LOGIN_DEPARTAMENTO_ERRO = "dpto_failure";

```

```

    public static final String LOGIN_INSTITUICAO_OK = "inst_success";
    public static final String LOGIN_INSTITUICAO_ERRO = "inst_failure";

```

```

        public static final String SAVE_DOCENTE_OK = "save_success";
        public static final String SAVE_DOCENTE_ERRO = "save_failure";

```

```

    public static final int USER_DOCENTE = 0;
    public static final int USER_DEPARTAMENTO = 1;
    public static final int USER_INSTITUICAO = 2;

```

```
}

```

Arquivo: ged/src/br/ufsc/ine5328/data/GedDaoImpl.java

```
package br.ufsc.ine5328.data;

```

```
import java.util.ArrayList;

```

```
import java.util.HashMap;

```

```
import br.ufsc.ine5328.util.SqlMapDelegate;

```

```
/**

```

```
 * @author fred

```

```
 */

```

```
public class GedDaoImpl implements IGedDao {

```

```
    public HashMap autentiqueDocente(String username, String senha) {

```

```
        HashMap n = new HashMap();

```

```
        n.put("username", username);

```

```
        n.put("senha", senha);

```

```
        ArrayList l = (ArrayList) SqlMapDelegate.executeQueryForList("getDocenteForList", n);

```

```
        n.clear();

```

```
        if (!l.isEmpty()){

```

```
            n = (HashMap) l.get(0);

```

```
        }

```

```
        return n;

```

```
    }

```

```
    public HashMap autentiqueDepartamento(String username, String senha) {

```

```
        HashMap n = new HashMap();

```

```
        n.put("username", username);

```

```
        n.put("senha", senha);

```

```
        ArrayList l = (ArrayList) SqlMapDelegate.executeQueryForList("getDepartamentoForList",

```

```
n);

```

```
        n.clear();

```

```
        if (!l.isEmpty()){

```

```
            n = (HashMap) l.get(0);

```

```
        }

```

```
        return n;

```

```

    }

    public HashMap autentiqueInstituicao(String username, String senha) {
        HashMap n = new HashMap();
        n.put("username", username);
        n.put("senha", senha);
        ArrayList l = (ArrayList) SqlMapDelegate.executeQueryForList("getInstituicaoForList", n);
        n.clear();
        if (!l.isEmpty()){
            n = (HashMap) l.get(0);
        }
        return n;
    }
}

```

Arquivo: ged/src/br/ufsc/ine5328/data/IGedDao.java

```

package br.ufsc.ine5328.data;

import java.util.HashMap;

public interface IGedDao {

    HashMap autentiqueDocente(String username, String senha);
    HashMap autentiqueDepartamento(String username, String senha);
    HashMap autentiqueInstituicao(String username, String senha);

}

```

Arquivo: ged/src/br/ufsc/ine5328/Departamento.java

```

package br.ufsc.ine5328;

import java.util.HashMap;

public class Departamento implements IDepartamento {

    private Long idDepartamento;
    private String nome;

```



```

public Departamento(HashMap d) {
    this.idDepartamento = new Long((String) d.get("idDepartamento"));
    this.nome = (String) d.get("nome");
}

public Long getIdDepartamento() {
    return this.idDepartamento;
}

public String getNome() {
    return this.nome;
}

public void setIdDepartamento(Long idDepartamento) {
    this.idDepartamento = idDepartamento;
}

public void setNome(String nome) {
    this.nome = nome;
}
}

```

Arquivo: ged/src/br/ufsc/ine5328/Docente.java

```

package br.ufsc.ine5328;

import java.util.HashMap;

public class Docente implements IDocente {

    private Long id;
    private int status;
    //private int ria;
    private int ano;
    private int periodo;
    private int etapa;
    private String nome;
    private String idLattes;
    private String cpf;
    private Long idInstituicao;
    private Long idDepartamento;
}

```

```
public Docente(){

}

public Docente(HashMap d) {
    this.id = new Long((String) d.get("id_docente"));
    this.status = (Integer) d.get("status");
    this.ano = (Integer) d.get("ano");
    this.periodo = (Integer) d.get("periodo");
    this.etapa = (Integer) d.get("etapa");
    this.nome = (String) d.get("nome");
    this.idLattes = (String) d.get("idLattes");
    this.cpf = (String) d.get("cpf");
    this.idInstituicao = new Long((String) d.get("idInstituicao"));
    this.idDepartamento = new Long((String) d.get("idDepartamento"));
}

public int getAno() {
    return ano;
}

public void setAno(int ano) {
    this.ano = ano;
}

public String getCpf() {
    return cpf;
}

public void setCpf(String cpf) {
    this.cpf = cpf;
}

public int getEtapa() {
    return etapa;
}

public void setEtapa(int etapa) {
    this.etapa = etapa;
}
```

```
public Long getId() {
    return id;
}

public void setId(Long id) {
    this.id = id;
}

public Long getIdDepartamento() {
    return idDepartamento;
}

public void setIdDepartamento(Long idDepartamento) {
    this.idDepartamento = idDepartamento;
}

public Long getIdInstituicao() {
    return idInstituicao;
}

public void setIdInstituicao(Long idInstituicao) {
    this.idInstituicao = idInstituicao;
}

public String getIdLattes() {
    return idLattes;
}

public void setIdLattes(String idLattes) {
    this.idLattes = idLattes;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public int getPeriodo() {
    return periodo;
}
```

```

    }

    public void setPeriodo(int periodo) {
        this.periodo = periodo;
    }

    public int getStatus() {
        return status;
    }

    public void setStatus(int status) {
        this.status = status;
    }
}

```

Arquivo: ged/src/br/ufsc/ine5328/GedFacade.java

```

package br.ufsc.ine5328;

import java.util.ArrayList;
import java.util.GregorianCalendar;
import java.util.HashMap;
import java.util.List;

import br.ufsc.ine5328.collection.IVisitor;
import br.ufsc.ine5328.collection.PopulatorVisitor;
import br.ufsc.ine5328.data.DaoFactory;
import br.ufsc.ine5328.util.ArvoreAtividade;
import br.ufsc.ine5328.util.AtividadeNode;
import br.ufsc.ine5328.util.GedHelper;

public class GedFacade {

    public static IDocente autentiqueDocente(String username, String senha) {
        HashMap m = DaoFactory.getGedDao().autentiqueDocente(username, senha);
        if (m.isEmpty()) {
            return null;
        } else {
            return new Docente(m);
        }
    }
}

```

```
public static IDepartamento autentiqueDepartamento(String username, String senha) {
    HashMap m = DaoFactory.getGedDao().autentiqueDepartamento(username, senha);
    if (m.isEmpty()) {
        return null;
    } else {
        return new Departamento(m);
    }
}
```

```
public static IInstituicao autentiqueInstituicao(String username, String senha) {
    HashMap m = DaoFactory.getGedDao().autentiqueInstituicao(username, senha);
    if (m.isEmpty()) {
        return null;
    } else {
        return new Instituicao(m);
    }
}
```

```
public static IDocente getDocente(Long idGedDocente) {
    return DaoFactory.getDocenteDao().getDocente(idGedDocente);
}
```

```
public static IDocente getDocente(String idGedDocente) {
    return DaoFactory.getDocenteDao().getDocente(idGedDocente);
}
```

```
public static List getDocentes(Long idDepartamento) {
    return DaoFactory.getDepartamentoDao().getDocentes(idDepartamento);
}
```

```
public static IDepartamento getDepartamento(Long idDepartamento) {
    return DaoFactory.getDepartamentoDao().getDepartamento(idDepartamento);
}
```

```
public static List getDepartamentos(Long idInstituicao) {
    return DaoFactory.getInstituicaoDao().getDepartamentos(idInstituicao);
}
```

```
public static List getAtividadeEsino(IDocente docente) {
    return DaoFactory.getDocenteDao().getAtividadeEsino(docente);
}
```

```

public static List getProducao(IDocente docente) {
    return DaoFactory.getDocenteDao().getProducao(docente);
}

public static List getAtividadePesquisaExtensao(IDocente docente) {
    return DaoFactory.getDocenteDao().getAtividadePesquisaExtensao(docente);
}

public static List getAtividadeQualificacao(IDocente docente) {
    return DaoFactory.getDocenteDao().getAtividadeQualificacao(docente);
}

public static List getAtividadeAdministrativa(IDocente docente) {
    return DaoFactory.getDocenteDao().getAtividadeAdministrativa(docente);
}

public static List getOutraAtividade(IDocente docente) {
    return DaoFactory.getDocenteDao().getOutraAtividade(docente);
}

public static void updateGedDocente(Long idGedDocente, int status) {
    DaoFactory.getDocenteDao().updateGedDocente(idGedDocente, status);
}

public static void updateStatusAtividadeEsino(Long idGedDocente, Long idAtividadeEsino, int
status) {
    DaoFactory.getDocenteDao().updateStatusAtividadeEsino(idGedDocente,
idAtividadeEsino, status);
}

public static String updateStatusProducao(Long idGedDocente, Long idProducao, int status) {
    DaoFactory.getDocenteDao().updateStatusProducao(idGedDocente, idProducao, status);
    return null;
}

public static void updateStatusAtividadePesquisaExtensao(Long idGedDocente, Long
idAtividadePesquisaExtensao, int status) {
    DaoFactory.getDocenteDao().updateStatusAtividadePesquisaExtensao(idGedDocente,
idAtividadePesquisaExtensao, status);
}

```

```
        public static void updateStatusAtividadeQualificacao(Long idGedDocente, Long
idAtividadeQualificacao, int status) {
            DaoFactory.getDocenteDao().updateStatusAtividadeQualificacao(idGedDocente,
idAtividadeQualificacao, status);
        }

        public static void updateStatusAtividadeAdministrativa(Long idGedDocente, Long
idAtividadeAdministrativa, int status) {
            DaoFactory.getDocenteDao().updateStatusAtividadeAdministrativa(idGedDocente,
idAtividadeAdministrativa, status);
        }

        public static void updateStatusOutraAtividade(Long idGedDocente, Long idOutraAtividade, int
status) {
            DaoFactory.getDocenteDao().updateStatusOutraAtividade(idGedDocente,
idOutraAtividade, status);
        }

        public static void insertAtividadeEnsino(HashMap atv) {
            DaoFactory.getDocenteDao().insertAtividadeEnsino(atv);
        }

        public static void insertProducao(HashMap atv) {
            DaoFactory.getDocenteDao().insertProducao(atv);
        }

        public static void insertAtividadePesquisaExtensao(HashMap atv) {
            DaoFactory.getDocenteDao().insertAtividadePesquisaExtensao(atv);
        }

        public static void insertAtividadeQualificacao(HashMap atv) {
            DaoFactory.getDocenteDao().insertAtividadeQualificacao(atv);
        }

        public static void insertAtividadeAdministrativa(HashMap atv) {
            DaoFactory.getDocenteDao().insertAtividadeAdministrativa(atv);
        }

        public static void insertOutraAtividade(HashMap atv) {
            DaoFactory.getDocenteDao().insertOutraAtividade(atv);
        }
    }
```

```

/**
 * Consulta as atividades do docente e monta a arvore de atividades
 *
 * @param docente
 * @param user
 * @return
 */
public static AtividadeNode getRootArvoreDeAtividadeDocente(IDocente docente, int user) {
    GedHelper helper = new GedHelper();
    HashMap auxMap = new HashMap();

    /**
     * Seleciona dados de outros banco de o usuário for um docente.
     */
    if (user == Constantes.USER_DOCENTE) {
        // apenas teste
        HashMap aux = new HashMap();
        ArrayList tmp = new ArrayList();
        aux.put("nome", "teste dado externo @ " + new GregorianCalendar().getTime());
        aux.put("horas_aula", "108");
        aux.put("status", 0);
        aux.put("codGed", "121");
        aux.put("id", -1);
        tmp.add(aux);

        auxMap.put("121", tmp);
    }

    /**
     * Obtém dados da banco GED (dados já aprovados pelo docente)
     */

    helper.buildAtividadesMap(getAtividadeEsino(docente), auxMap);
    helper.buildAtividadesMap(getProducao(docente), auxMap);
    helper.buildAtividadesMap(getAtividadePesquisaExtensao(docente), auxMap);
    helper.buildAtividadesMap(getAtividadeQualificacao(docente), auxMap);
    helper.buildAtividadesMap(getAtividadeAdministrativa(docente), auxMap);
    helper.buildAtividadesMap(getOutraAtividade(docente), auxMap);

    IVisitor visitor = new PopulatorVisitor(auxMap);
    ArvoreAtividade arv = new ArvoreAtividade();
    AtividadeNode result = arv.getRaiz();

```



```

        atvMap.put("periodo", docente.getPeriodo());
        atvMap.put("id_docente", id_docente.toString());
        insertAtividadePesquisaExtensao(atvMap);
    } else
        updateStatusAtividadePesquisaExtensao(id_docente, new
Long((String) atvMap.get("id")), (Integer) atvMap.get("status"));
    }
}

aux = atividades.get("4");
if (aux != null) {
    for (HashMap atvMap : aux) {

        if (((String) atvMap.get("id")).equals("-1")) {
            atvMap.put("ano", docente.getAno());
            atvMap.put("periodo", docente.getPeriodo());
            atvMap.put("id_docente", id_docente.toString());
            insertAtividadeQualificacao(atvMap);
        } else
            updateStatusAtividadeQualificacao(id_docente, new
Long((String) atvMap.get("id")), (Integer) atvMap.get("status"));
        }
    }

aux = atividades.get("5");
if (aux != null) {
    for (HashMap atvMap : aux) {
        if (((String) atvMap.get("id")).equals("-1")) {
            atvMap.put("ano", docente.getAno());
            atvMap.put("periodo", docente.getPeriodo());
            atvMap.put("id_docente", id_docente.toString());
            insertAtividadeAdministrativa(atvMap);
        } else
            updateStatusAtividadeAdministrativa(id_docente, new
Long((String) atvMap.get("id")), (Integer) atvMap.get("status"));
        }
    }

aux = atividades.get("6");
if (aux != null) {
    for (HashMap atvMap : aux) {
        if (((String) atvMap.get("id")).equals("-1")) {

```

```

        atvMap.put("ano", docente.getAno());
        atvMap.put("periodo", docente.getPeriodo());
        atvMap.put("id_docente", id_docente.toString());
        insertOutraAtividade(atvMap);
    } else
        updateStatusOutraAtividade(id_docente, new Long((String)
atvMap.get("id")), (Integer) atvMap.get("status"));
    }
}
}
}
}
}

```

Arquivo: ged/src/br/ufsc/ine5328/IDepartamento.java

```

package br.ufsc.ine5328;

public interface IDepartamento {

    Long getIdDepartamento();
    String getNome();

}

```

Arquivo: ged/src/br/ufsc/ine5328/IDocente.java

```

package br.ufsc.ine5328;

public interface IDocente {

    Long getId();

    int getStatus();

    int getAno();

    int getPeriodo();

    String getNome();

    String getIdLattes();
}

```

```
        String getCpf();

        Long getIdInstituicao();

        Long getIdDepartamento();

        void setAno(int ano);

        void setIdDepartamento(Long idDepartamento);

        void setId(Long idGed);

        void setIdInstituicao(Long idInstituicao);

        void setIdLattes(String idLattes);

        void setNome(String nome);

        void setCpf(String cpf);

        void setPeriodo(int periodo);

        void setStatus(int status);

    }
```

Arquivo: ged/src/br/ufsc/ine5328/Instituicao.java

```
package br.ufsc.ine5328;
```

```
public interface IInstituicao {
```

```
    Long getIdInstituicao();
    String getNome();
```

```
}
```

Arquivo: ged/src/br/ufsc/ine5328/Instituicao.java

```
package br.ufsc.ine5328;
```

```
import java.util.HashMap;

public class Instituicao implements IInstituicao {

    private Long idInstituicao;
    private String nome;

    public Instituicao(HashMap d) {
        this.idInstituicao = new Long((String) d.get("idInstituicao"));
        this.nome = "Instituicao 1";
    }

    public Long getIdInstituicao() {
        return this.idInstituicao;
    }

    public String getNome() {
        return this.nome;
    }
}
```

Arquivo: ged/src/br/ufsc/ine5328/Mensagens.java

```
package br.ufsc.ine5328;

public class Mensagens {

    public static final String ERROR_USUARIO_INVALIDO = "user_failure";
    public static final String LOGIN_DOCENTE_ERRO = "docente_failure";

    public static final String LOGIN_DEPARTAMENTO_OK = "dpto_success";
    public static final String LOGIN_DEPARTAMENTO_ERRO = "dpto_failure";

    public static final String LOGIN_INSTITUICAO_OK = "inst_success";
    public static final String LOGIN_INSTITUICAO_ERRO = "inst_failure";

}
```

Arquivo: ged/src/br/ufsc/ine5328/ui/AbstractJsfBean.java

```
package br.ufsc.ine5328.ui;

import java.text.MessageFormat;
import java.util.Map;

import javax.faces.application.Application;
import javax.faces.application.FacesMessage;
import javax.faces.context.ExternalContext;
import javax.faces.context.FacesContext;
import javax.faces.el.ValueBinding;

import br.ufsc.ine5328.Constantes;
import br.ufsc.ine5328.IDepartamento;
import br.ufsc.ine5328.IInstituicao;

public abstract class AbstractJsfBean{

    protected Application getApplication()
    {
        return FacesContext.getCurrentInstance().getApplication();
    }

    protected Map getApplicationMap()
    {
        return getExternalContext().getApplicationMap();
    }

    protected FacesContext getContext()
    {
        return FacesContext.getCurrentInstance();
    }

    protected ExternalContext getExternalContext()
    {
        return FacesContext.getCurrentInstance().getExternalContext();
    }

    protected Map getRequestMap()
    {
        return getExternalContext().getRequestMap();
    }
}
```

```
protected Map getSessionMap()
{
    return getExternalContext().getSessionMap();
}

protected Object getBean(String name)
{
    return getApplication().getVariableResolver().resolveVariable(getContext(), name);
}

protected void setBean(String name, Object value)
{
    setValue("#{ " + name + "}", value);
}

protected Object getValue(String expr)
{
    ValueBinding vb = getApplication().createValueBinding(expr);
    return vb.getValue(getContext());
}

protected void setValue(String expr, Object value)
{
    ValueBinding vb = getApplication().createValueBinding(expr);
    vb.setValue(getContext(), value);
}

public void addMessage(String msg, Object[] params) {
    FacesContext c = this.getContext();
    FacesMessage m = new FacesMessage(MessageFormat.format(msg, params));
    m.setSeverity(FacesMessage.SEVERITY_INFO);
    c.addMessage(c.getViewRoot().createUniqueId(), m);
}

public void addMessage(String msg) {
    FacesContext c = this.getContext();
    FacesMessage m = new FacesMessage(msg);
    m.setSeverity(FacesMessage.SEVERITY_INFO);
    c.addMessage(c.getViewRoot().createUniqueId(), m);
}
```