



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

ESCOLA DE INFORMÁTICA APLICADA

Gerindo Dados de Docentes de Programas de Pós-Graduação

Leonardo dos Santos Silva

Orientador

Leonardo Guerreiro Azevedo

RIO DE JANEIRO, RJ – BRASIL

FEVEREIRO DE 2018

Catálogo informatizada pelo(a) autor(a)

S586 Silva, Leonardo dos Santos
Gerindo dados de docentes de programas de pós-
graduação / Leonardo dos Santos Silva. -- Rio de
Janeiro, 2018.
54 f.

Orientador: Leonardo guerreiro azevedo.
Trabalho de Conclusão de Curso (Graduação) -
Universidade Federal do Estado do Rio de Janeiro,
Graduação em Sistemas de Informação, 2018.

1. Automatização de Sistemas. 2. Avaliação
Docente. 3. Sistemas de Informação. 4. Java. 5.
Banco de Dados Relacionais. I. azevedo, Leonardo
guerreiro, orient. II. Título.

Gerindo Dados de Docentes de Programas de Pós-Graduação

Leonardo dos Santos Silva

Projeto de Graduação apresentado à Escola de
Informática Aplicada da Universidade Federal do
Estado do Rio de Janeiro (UNIRIO) para obtenção do
título de Bacharel em Sistemas de Informação.

Aprovada por:

Leonardo Guerreiro Azevedo, D. Sc. (UNIRIO; IBM Research)

Flávia Maria Santoro, D. Sc. (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

FEVEREIRO DE 2018

Agradecimentos

Agradeço, primeiramente, a Deus por ter me dado saúde e motivação para que eu pudesse alcançar este objetivo.

Agradeço também aos meus pais, Rosângela e Carlos, por sempre acreditarem em mim e pelo apoio incondicional, com total suporte para ultrapassar os obstáculos, além de todos os esforços para me educar e formar o homem que sou hoje.

Agradeço ao meu orientador, Leonardo, por toda a paciência, dedicação e apoio neste trabalho.

Agradeço também a todo o corpo docente da Unirio que compartilhou o conhecimento necessário para minha formação, especialmente aos professores Leonardo Azevedo, Geiza Hamazaki e Marcio Barros, sem dúvida são professores diferenciados com excelentes aulas e por serem ótimas pessoas.

Por fim, agradeço a todos os meus amigos da faculdade, mas também fora dela que me ajudaram no decorrer do curso, seja estudando juntos, nos momentos de desespero das provas, até mesmo com conselhos sobre a vida. Todos foram muito importantes, tanto no meu crescimento acadêmico, como pessoa.

RESUMO

Em nossa sociedade altamente dependente da tecnologia, encontramos uma variedade de aplicações web suportando tarefas a serem realizadas e sempre sendo mencionadas como uma possível solução para a resolução de um novo problema. Este trabalho apresenta o desenvolvimento de uma aplicação para gerenciamento dos dados de docentes do Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal do Estado do Rio de Janeiro (UNIRIO). A aplicação foi desenvolvida utilizando a linguagem JAVA e frameworks como Struts2 e Bootstrap e o uso de banco de dados relacional para a persistência dos dados. O objetivo foi desenvolver uma aplicação que facilitasse o processo de importação dos dados e geração automática da avaliação dos docentes do programa, agilizando o processos e tratando inconsistências, que, caso contrário, deveriam ser tratadas de forma manual. Além disso, buscou-se o uso de ferramentas e técnicas de mercado amplamente difundidas no mercado a fim de ilustrar como o curso de graduação em informática da UNIRIO está alinhado com a realidade.

Palavras-chave: Automatização de Sistemas, Avaliação Docente, Sistemas de Informação, Java, Banco de Dados Relacionais.

ABSTRACT

In our society highly dependent on technology, we find a variety of web applications supporting tasks to be performed and always being mentioned as a possible solution to solve new problems. This work presents the development of an application for data management of professors of the Graduate Program in Informatics (PPGI) within Federal University of the State of Rio de Janeiro (UNIRIO). The applications was developed using the JAVA language and frameworks such as Struts2 and Bootstrap, and the use of relational database for data persistence. The goal was to develop an application that improves the data load and automatic generation of the graduate program's professors evaluation, speeding-up the process and handling inconsistencies, which, on the other hand, should be managed manually. Moreover, we aimed at the use of tools and technologies widely employed in the industry in order to illustrate how the undergraduate course in informatics of UNIRIO is aligned with the reality.

Keywords: Systems Automation, Professor Evaluation, Information Systems, Java, Relational Database.

Índice

1 Introdução	9
2 Principais Conceitos.....	11
2.1 Modelo Entidade Relacionamento.....	11
2.2 Modelo Relacional.....	13
2.3 Transformando modelo conceitual em lógico	15
2.4 Java e MVC	18
3 Desenvolvimento do projeto	20
3.1 Visão geral.....	20
3.2 Estrutura do projeto	25
3.3 Tecnologias Utilizadas.....	26
3.4 Inconsistências solucionadas	27
3.5 Casos de Uso.....	29
4 Funcionalidades do SGPP (Sistema de Gerenciamento de Produtividade de Professores)	32
4.1 Visão Geral	32
4.2 Baixar XML.....	32
4.3 Solicitar Criação de Usuário	34
4.4 Logar no Sistema	35
4.5 Criar e deletar banco de dados.....	37
4.6 Carregar informações do arquivo XML	38
4.7 Listar Publicações.....	39
4.8 Calcular Critério	40
5 Conclusão.....	43
6 Referências Bibliográficas	48
Apêndice I – Saída de dados (Orientações) do SGPP.....	50
Apêndice II – Saída de dados (Participações) do SGPP.....	51

Apêndice III – Saída de dados (Produções) do SGPP.....	52
Apêndice IV – Codificação do projeto.....	54

Índice de Tabelas

Tabela 1 - Regras para implementação de relacionamentos [3].....	18
Tabela 2 – Exemplo de saída de dados das orientações	50
Tabela 3 – Exemplo de saída de dados das participações	51
Tabela 4 – Exemplo de saída de dados das produções	52

Índice de Figuras

Figura 1 - Exemplo de entidades	11
Figura 2 - Exemplo de relacionamento entre departamento e pessoa	11
Figura 3 - Representação dos atributos da entidade Projeto.....	12
Figura 4 - Modelo entidade relacionamento de uma universidade.....	12
Figura 5 - Exemplo de chave primária composta mínima.....	14
Figura 6 - Chave estrangeira referenciando a própria tabela.....	14
Figura 7 - Coluna CIC atuando como chave alternativa	15
Figura 8 - Exemplo de modelo ER transformado em modelo relacional	16
Figura 9 - Representação do padrão de arquitetura de software MVC	19
Figura 10 – Planilha de Cálculo da Auto Avaliação	21
Figura 11 – Passo-a-passo de uso do sistema	22
Figura 12 – Modelo conceitual versão final	24
Figura 13- Estrutura MVC dos arquivos do projeto JAVA.....	25
Figura 14 - Fórmula adotada para identificar semelhança de nomes de fóruns	28
Figura 15 – Casos de uso do sistema.....	31
Figura 16 – Portal Lattes	33
Figura 17 – Pesquisa de Currículo.....	33
Figura 18 - Seleção do currículo	34
Figura 19 - Download em XML do currículo.....	34
Figura 20 – Cadastro de Usuário	35
Figura 21 – Mensagem de sucesso da aplicação	35
Figura 22 – Tela de acesso ao sistema através de usuário e senha.....	36
Figura 23 – Tela inicial para usuário comum.....	36
Figura 24 – Tela inicial para administrador	37
Figura 25 – Mensagem de sucesso para exclusão de base de dados	37
Figura 26 – Mensagem de sucesso para criação das bases de dados do sistema.....	37
Figura 27 - Upload do arquivo XML	38
Figura 28 – Validação e inserção de novos sinônimos de autores	38
Figura 29 – Tela de download dos arquivos textos	39
Figura 30 – Busca por nome.....	39
Figura 31 – Tratamento de ambiguidade na pesquisa	40
Figura 32 – Listagem das publicações carregadas na base de dados	40

Figura 33 – Tabela que provê o cálculo do primeiro critério	41
Figura 35 – Exemplo de DAO para busca do qualis.....	54
Figura 36 – Exemplo de Controller disponibilizando arquivos de saída para download	55
Figura 37 – Exemplo de trecho de view responsável pelo retorno do calculo.....	56

1 Introdução

Automação pode ser definida como a técnica de tomar um processo ou sistema automático e refere-se tanto a serviços executados como a produtos fabricados automaticamente e às tarefas de intercâmbio de informações [1].

Com as mudanças que a sociedade vem presenciando, o tempo torna-se mais precioso e os esforços para resoluções de problemas complexos demandam uma quantidade significativa de tempo. A automatização se insere neste contexto, atuando como um agente facilitador de trabalhos manuais, que seguem um processo definido e lógico, otimizando o tempo e recursos empregados na execução da tarefa.

A informática e a tecnologia contribuem de forma clara nesta tendência, com a popularização dos computadores, e, em um segundo momento, as aplicações e sistemas dependentes dos mesmos também se popularizaram. É comum se deparar com negócios que em um momento anterior utilizavam-se de papel para armazenar informações, evoluindo para as planilhas eletrônicas que dominaram o mercado como conceito de informatização do negócio. Atualmente, aplicações que substituem planilhas ou até mesmo o próprio papel são o foco da automatização das tarefas, diminuindo a probabilidade de erro, interação humana, diminuindo custos e trazendo eficiência.

A motivação para este trabalho se deu devido ao interesse em desenvolver uma aplicação que fosse útil para o ambiente acadêmico bem como ampliar a experiência em banco de dados e desenvolvimento de sistemas. Dessa forma, após entrevistas com professores do Programa de Pós-Graduação em Informática da UNIRIO, decidiu-se pela criação de um gerenciador dos dados da avaliação do programa de Pós-Graduação, visando automatizar o processo de preenchimento das informações na planilha de cálculo de avaliação de professores. Logo, a aplicação desenvolvida neste projeto facilita o processo de importação dos dados, que caso contrário seria manual, agilizando e tratando inconsistências, buscando alcançar o máximo de efetividade da importação das informações no contexto correto. Tratar essas possibilidades foi desafiador, visto que a cada cenário tratado era identificado uma nova questão que geravam inconsistências de informações.

Além disso, a possibilidade de utilizar neste projeto ferramentas e técnicas de mercado amplamente difundidas incentivou na forma de exemplificar como o curso de graduação está alinhado com a realidade, tornando o discente com capaz de evoluir no aprendizado, tratando diferentes problemas e utilizando abordagens específicas sempre que necessário, desde que atendam de forma adequada o problema.

Portanto, o principal objetivo deste trabalho é aplicar técnicas e utilizar tecnologias comuns ao mercado, apoiado pelo conhecimento obtido no decorrer da graduação. Para isso será desenvolvido uma aplicação com interface Web, que será responsável pelo gerenciamento das informações de avaliação docente do programa de Pós-Graduação em Informática (PPGI) da UNIRIO. Dessa forma, levando à automatização de parte da tarefa que hoje é realizada através de uma planilha Excel.

O presente trabalho está estruturado em capítulos e, além desta introdução, será desenvolvido da seguinte forma:

- Capítulo II: Apresenta os conceitos inerentes às etapas de desenvolvimento e as tecnologias usadas.
- Capítulo III: Apresenta uma descrição detalhada de cada etapa do projeto, desde a estruturação do banco de dados até a codificação da aplicação e sua implantação.
- Capítulo IV: Descreve as funcionalidades e as dificuldades encontradas, relatando as soluções implementadas.
- Capítulo V: Apresenta as considerações finais e sugestões de melhorias e aprimoramentos

2 Principais Conceitos

Este capítulo apresenta os conceitos relacionados a este trabalho.

2.1 Modelo Entidade Relacionamento

O modelo ER (entidade relacionamento) é empregado para modelagem de banco de dados. Ele é constituído de alguns elementos, tais como, entidades, relacionamentos e suas cardinalidades e atributos que posteriormente serão para estruturas de banco de dados relacionais [3]. Alguns exemplos de elementos modelo entidade relacionamento são apresentados na Figura 1, Figura 2 e

Figura 3.

Entidades são um conjunto de objetos que se desejam guardar informações.



Figura 1 - Exemplo de entidades

Relacionamentos são a forma como os objetos (entidades) se relacionam.



Figura 2 - Exemplo de relacionamento entre departamento e pessoa

Por fim, tem-se os atributos que analogamente pode-se considerar uma propriedade ou característica do objeto a que pertence.

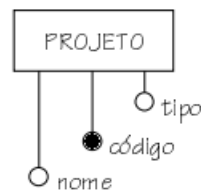


Figura 3 - Representação dos atributos da entidade Projeto

Um DER (diagrama entidade relacionamento) é o diagrama utilizado para representar graficamente um Modelo Entidade Relacionamento e é definido por Heuser da seguinte forma [3]: “Um DER é um modelo formal, preciso, não ambíguo. Isto significa que diferentes leitores de um mesmo DER devem sempre entender exatamente o mesmo”. Além da definição dada ser de extrema importância para qualquer projeto de banco de dados, diagramas de entidade relacionamento atuam de forma exemplar na tarefa de abstrair a tecnologia empregada no armazenamento de dados. De forma secundária, eles tornam o processo de entendimento do contexto das informações mais intuitivo ao utilizar uma representação gráfica.

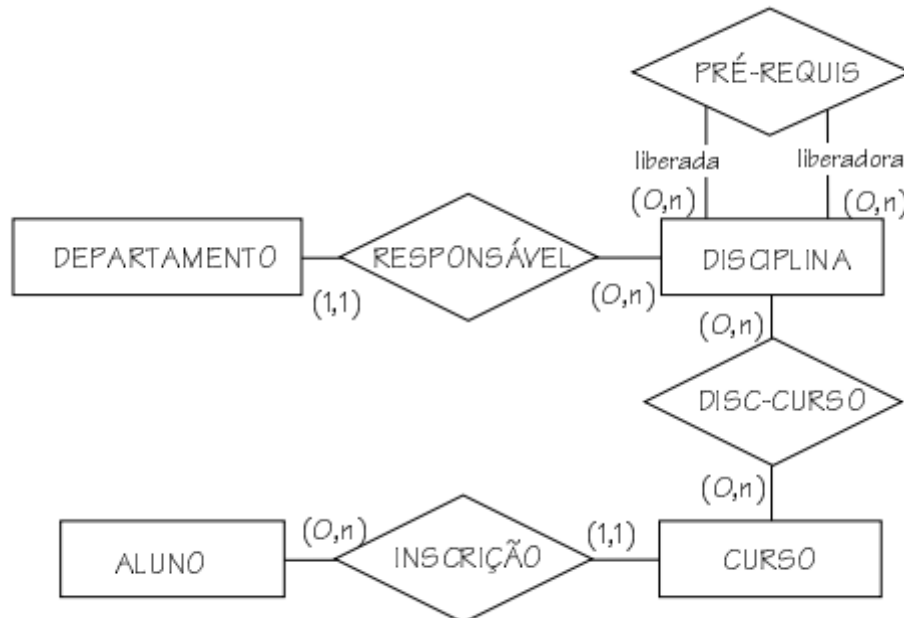


Figura 4 - Modelo entidade relacionamento de uma universidade

A Figura 4, apresenta um exemplo de um Diagrama Entidade-Relacionamento elaborado utilizando a notação de Peter Chen [4]. Neste modelo temos as entidades e as cardinalidades associadas a cada relacionamento entre as entidades. No modelo, temos que disciplina tem um e no máximo um departamento responsável, bem como um departamento pode ter nenhuma ou várias disciplinas sob sua responsabilidade. Disciplinas podem ser pré-requisito de nenhuma ou várias outras disciplinas, sendo o inverso também verdadeiro, onde uma disciplina pode depender de nenhuma ou mais de uma para ser possível cursá-la. Temos também o conceito onde curso e disciplina estão igualmente relacionados, podendo ter nenhuma ou vários associados a nenhuma ou várias instancias. Finalmente, um aluno deve estar inscrito em apenas um único curso e um curso pode ter vários ou até mesmo nenhum aluno.

2.2 Modelo Relacional

A partir do modelo conceitual, aplica-se um conjunto de regras de transformação para elaboração do modelo relacional. Este modelo é totalmente dependente do SGBD (Sistema de Gerenciamento de Banco de Dados) escolhido para construção do sistema de informação e inclui os seguintes conceitos.

- Tabela: Conjunto não ordenado de linhas. Cada linha é composta por um conjunto de campos ou colunas. Cada coluna é identificada por um nome.
- Chaves: Há ao menos três tipos de chaves a considerar:
 - Chave Primária: Coluna ou combinação de colunas, onde seu conteúdo é distinto em toda e qualquer ocorrência nas demais linhas da mesma tabela, permitindo assim a identificação de uma linha. Uma chave primária, por definição, deve ser mínima, ou seja, todas as colunas que compõe a chave primária devem obrigatoriamente ser necessárias para garantir a unicidade de valores da chave, tal restrição é chamada de restrição de chave. A Figura 5 apresenta um exemplo de modelo lógico, com dados, da tabela Dependente, onde *CodigoEmp* e *NoDepen* são as colunas correspondentes à chave primária.

Dependente

CódigoEmp	NoDepen	Nome	Tipo	DataNasc
E1	01	João	Filho	12/12/91
E1	02	Maria	Esposa	01/01/50
E2	01	Ana	Esposa	05/11/55
E5	01	Paula	Esposa	04/07/60
E5	02	José	Filho	03/02/85

Figura 5 - Exemplo de chave primária composta mínima

- Chave Estrangeira: Utilizada para relacionar duas tabelas. Neste caso, uma ou mais colunas em uma tabela *T* são utilizadas para referenciar outra tabela *W*, cujos valores pertencem necessariamente a uma chave primária de *W*. A Figura 6 apresenta um exemplo onde a coluna *codigoEmpGerente* é utilizada para referenciar a coluna *CodigoEmp* da tabela *Emp*, representando o relacionamento de quem é o gerente do empregado.

Emp

CodigoEmp	Nome	CodigoDepto	CodigoEmpGerente
E5	Souza	D1	—
E3	Santos	D2	E5
E2	Silva	D1	E5
E1	Soares	D1	E1

chave estrangeira:
referencia a chave primária
da própria tabela

Figura 6 - Chave estrangeira referenciando a própria tabela

Além disso, toda chave estrangeira implica em restrições de integridade, assunto que será abordado no decorrer deste capítulo.

- Chave Alternativa (ou chave candidata): É uma coluna ou um conjunto de colunas que também identificam unicamente uma linha da tabela. Tem as mesmas características da chave primária, sendo que, como uma tabela só pode ter uma chave primária, esta chave é definida como chave alternativa. A Figura 7 ilustra tal cenário, onde o campo CIC, que por conceito é único para cada pessoa, é a chave candidata visto que *CodigoEmp* já atua como chave primária.

Emp				
CodigoEmp	Nome	CodigoDepto	CategFuncional	CIC
E5	Souza	D1	C5	132.121.331-20
E3	Santos	D2	C5	891.221.111-11
E2	Silva	D1	C2	341.511.773-45
E1	Soares	D1	—	631.692.754-88

chave estrangeira:
referencia a chave primária
da própria tabela

Figura 7 - Coluna CIC atuando como chave alternativa

- Outras restrições relevantes são:
 - Restrição de domínio: Define o conjunto de valores possíveis ou permitidos que um campo possa ter.
 - Restrição de nulo: Define se um campo pode ou não receber valor nulo (NULL).

2.3 Transformando modelo conceitual em lógico

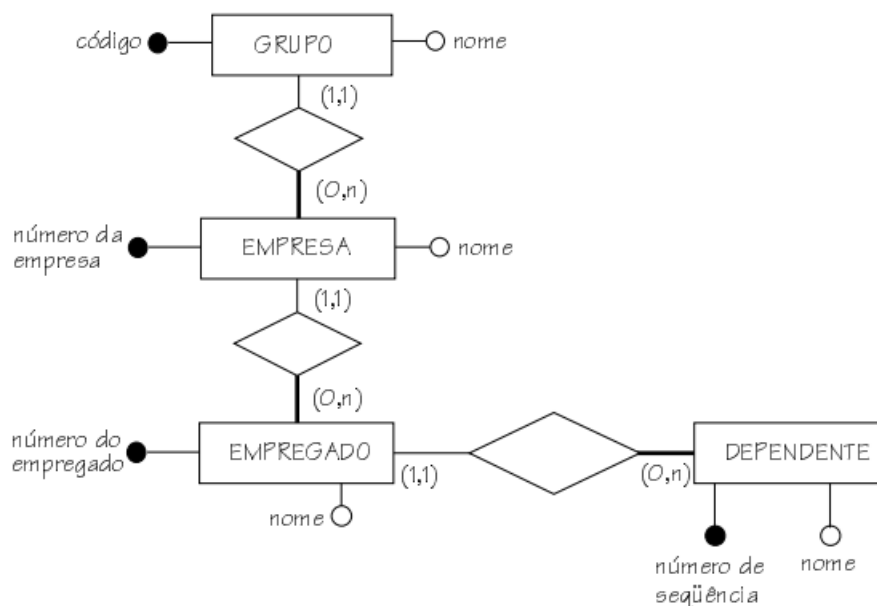
A partir de um modelo ER pode-se obter inúmeros modelos relacionais, e todos podem estar retratando de forma assertiva o modelo conceitual em questão, porém cada versão pode resultar em diferentes resultados de desempenho, além de facilitar ou dificultar o desenvolvimento ou manutenção do sistema construído.

Em função disto, utilizamos a metodologia que tem como objetivos [3]:

Obter um banco de dados que permita bom desempenho de instruções de consulta e alteração do banco de dados. Obter bom desempenho significa basicamente diminuir o número de acessos a disco, já que estes consomem o maior tempo na execução de uma instrução de banco de dados. Além disso, obter um banco de dados que simplifique o desenvolvimento e a manutenção de aplicações.

Para atingir tais objetivos, adota-se os seguintes princípios em ordem de prioridade. Primeiramente evita-se junções, ou seja, procura-se ter todos os dados necessários em uma única linha, diminuindo assim a quantidade de acessos a disco. Em seguida, diminuição de chaves primárias. Em geral é criado um índice para toda chave para acelerar o processamento de consultas, o que faz com que o SGBD seja obrigado a manter esta estrutura sempre atualizada, toda vez que ocorrer alguma alteração em uma tabela que envolva uma coluna que faça parte de um índice, estes índices terão que refletir a alteração. Por último, evitar campos opcionais, pois apesar dos SGBDs relacionais não desperdiçarem espaço devido a campos vazios, campos opcionais dificultam o

entendimento do banco de dados. Por exemplo, não se sabe se o campo não foi preenchido para uma linha da tabela por não permitir o valor para aquela linha ou por não saber o valor no momento da inserção. Além disso, em muitos dos casos, o preenchimento ou não do campo terá que ser tratado pela camada de aplicação e não pelo próprio SGBD. Dependendo da forma de implementação, isto pode resultar em erros difíceis de serem identificados.



Esquema relacional correspondente:

GRUPO(COD_GRUPO,NOME)

EMPRESA(COD_GRUPO,NO_EMPRESA,NOME)

COD_GRUPO REFERENCIA GRUPO(COD_GRUPO)

EMPREGADO(COD_GRUPO,NO_EMPRESA,NO_EMPREGADO,NOME)

(COD_GRUPO,NO_EMPRESA) REFERENCIAM EMPRESA(COD_GRUPO,NO_EMPRESA)

DEPENDENTE(COD_GRUPO,NO_EMPRESA,NO_EMPREGADO,NO_SEQ,NOME)

(COD_GRUPO,NO_EMPRESA,NO_EMPREGADO) REFERENCIAM EMPREGADO

(COD_GRUPO,NO_EMPRESA,NO_EMPREGADO)

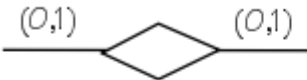
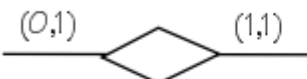
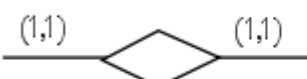
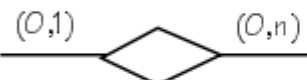
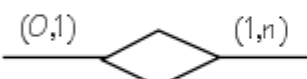
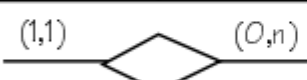
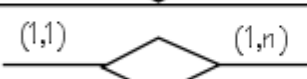
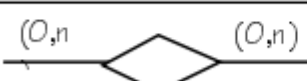
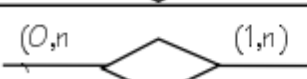
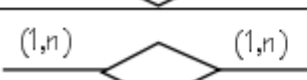
Figura 8 - Exemplo de modelo ER transformado em modelo relacional

A Figura 8 ilustra a transformação dos modelos aplicando as regras de transformação apresentadas por Heuser [3]. Em linhas gerais, as regras de transformação são as seguintes. Cada entidade forte do modelo conceitual (por exemplo, Grupo) é transformada em uma tabela. Cada atributo identificador da entidade (por exemplo, código) é transformado em uma coluna da chave primária da tabela. Entidades fracas (por exemplo, Empresa) são transformadas em tabelas cuja chave primária é composta pela

transformação dos atributos identificadores da entidade fraca combinados com as colunas da chave primária da tabela correspondente à entidade forte a qual a entidade fraca se relaciona.

Ao implementar os relacionamentos no modelo lógico, temos algumas possibilidades de transformações de acordo com a cardinalidade do relacionamento envolvido. São eles, tabela própria, adição de coluna ou fusão de tabelas. Na tabela própria, o relacionamento é transformado em uma tabela composta pelos atributos do relacionamento e pelas chaves primárias das tabelas que ele relaciona. As chaves primárias são transformadas em chaves estrangeiras para as tabelas relacionadas. A chave primária da tabela resultante do relacionamento dependerá da cardinalidade do relacionamento. A adição de coluna corresponde a adicionar a uma tabela os atributos do relacionamento e a chave primária correspondente a outra entidade do relacionamento. A fusão de tabelas corresponde a fundir em uma única tabela os atributos do relacionamento e todos os atributos das tabelas relacionadas. A Tabela 1 apresenta cada opção associada à um tipo de relacionamento e sua melhor utilização indicando se a solução é a mais apropriada, se a solução é mais ou menos ou se a solução não pode ser aplicada. Por exemplo, no caso de relacionamentos $(0, 1) \times (0,1)$, a adição de colunas é a mais apropriada, a fusão de tabelas não pode ser aplicada e a tabela própria não é uma solução muito boa (sendo indicada como mais ou menos).

Tabela 1 - Regras para implementação de relacionamentos [3]

Tipo de relacionamento	Regra de implementação		
	Tabela própria	Adição coluna	Fusão tabelas
Relacionamentos 1:1			
	±	✓	×
	×	±	✓
	×	±	✓
Relacionamentos 1:n			
	±	✓	×
	±	✓	×
	×	✓	×
	×	✓	×
Relacionamentos n:n			
	✓	×	×
	✓	×	×
	✓	×	×

✓ Alternativa preferida ± Pode ser usada × Não usar

2.4 Java e MVC

A linguagem de programação Java é baseada no conceito de programação orientada ao objeto, sendo derivada da linguagem C, o que possibilita que um desenvolvedor familiarizado com linhas de código em C necessite de uma curva de aprendizado menor em relação a outras linguagens. Ao se executar um programa na plataforma Java, todo o código escrito em .java é compilado, e após as verificações de sintaxe da linguagem é gerado arquivos em bytecode, os denominados .class. Tais arquivos são criados para que a Java Virtual Machine (JVM) possa executar as instruções.

Uma importante vantagem da linguagem é sua portabilidade, já que a JVM se trata de um software desenvolvido para diversas plataformas e por ser responsável pela etapa de traduzir as instruções e executar para o *chipset* que sua versão foi disponibilizada [5].

O padrão de arquitetura de software MVC, em português modelo-visão-controlador, tem como objetivo separar a forma como a informação é representada da interação do usuário. Analisando cada camada, temos, a camada de modelo que consiste nos dados da aplicação, regras de negócio, lógica e funções. A visão é definida por qualquer representação de saída de dados. Já o controlador atua como uma camada de mediação entre ambas como demonstrado na Figura 1, tal padrão é largamente utilizado, pois a possibilidade de reutilização de código e separação de conceitos traz maior dinamismo no que se trata do desenvolvimento e manutenção das aplicações.

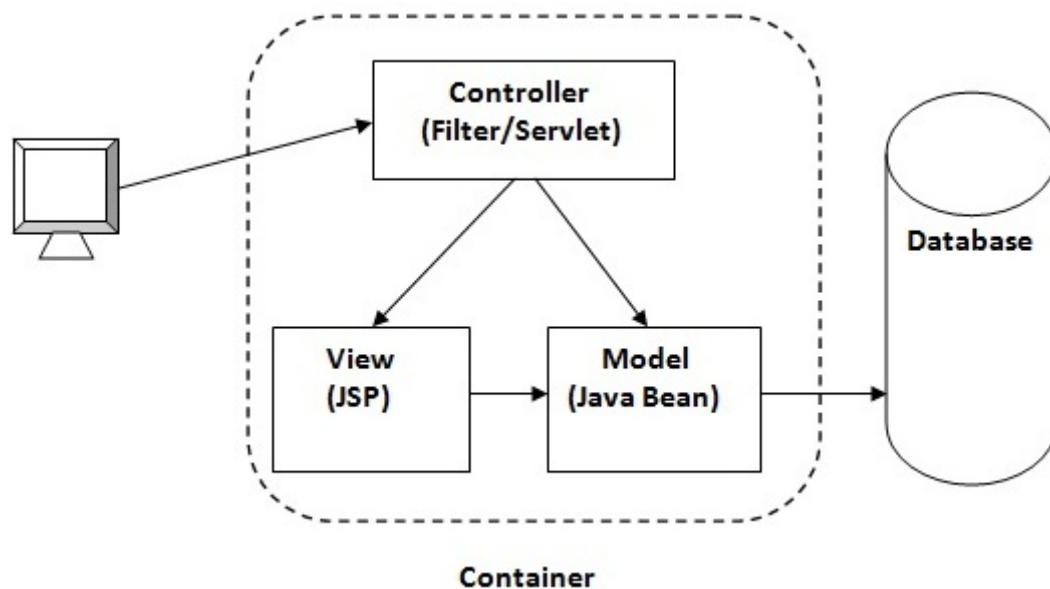


Figura 9 - Representação do padrão de arquitetura de software MVC

3 Desenvolvimento do projeto

3.1 Visão geral

O sistema desenvolvido neste trabalho tem como função, selecionar e estruturar o armazenamento dos dados de arquivos XML que contém informações do Currículo Lattes de cada professor, informações estas que são geridas pela plataforma Lattes do CNPQ¹. No entanto, tal plataforma não disponibiliza mecanismos simples para uso destas informações. O objetivo do sistema é calcular, de acordo com critérios já definidos, a pontuação de cada professor na avaliação do programa de Pós-Graduação em Informática (PPGI)² da Unirio. Esta avaliação é utilizada para avaliar a produção dos professores. Esta pontuação é um dos componentes para avaliar a permanência do professor no programa.

A pontuação contemplada no escopo deste trabalho engloba quatro macros-critérios principais:

- Publicações de artigos, que inclui a publicação em periódicos, conferências, escrita de livros e capítulo de livro.
- Orientações, desde que sua situação esteja como concluída ou em andamento e seja relacionada à graduação, ao mestrado, ao doutorado ou à iniciação científica.
- Projetos de pesquisa, com atuação na coordenação do projeto ou somente participando do mesmo, além de considerar sua abrangência sendo nacional ou internacional.
- Organização de Eventos, Comitês de programas científicos ou funções em instituições relacionadas.

De acordo com os macros-critérios acima e seus critérios internos ocorre um cálculo de peso que atribui a cada item um valor de acordo com sua relevância, conforme ilustrado na Figura 10.

¹ <http://lattes.cnpq.br/>

² <http://www2.uniriotec.br/ppgi>

Planilha de Avaliação Docente - PPGI/UNIRIO							
versão: Maio/2016							
Docente: <PREENCHER NOME DO DOCENTE AQUI>							
Data: <PREENCHER A DATA DA AVALIAÇÃO AQUI>							
Período de Referência: Quadrênio - 2013 a 2016							
Pontuação 100							
Índice da Coluna do ano na aba Sumário 6 7 8 9 13							
Disciplinas							
Nome da disciplina							ano.semestre
Última disciplina ministrada no PPGI							
Publicações							
Peso: 60,00%		Limite de pontos:		100			
		Limite de pontos por fora do Qualis restrito:		70			
		Limite de pontos sem Qualis e Qualis C:		20			
	Peso	2016	2015	2014	2013	Total	
Artigo em revista Qualis A1 em Ciência da Computação	30	0	0	0	0	30	
Artigo em revista Qualis A2 em Ciência da Computação	26	0	0	0	0	0	
Artigo em revista Qualis B1 em Ciência da Computação	20	0	0	0	0	0	
Artigo em revista Qualis B2 em Ciência da Computação	16	0	0	0	0	0	
Artigo em revista Qualis B3 em Ciência da Computação	14	0	0	2	1	42	
Artigo em revista Qualis B4 em Ciência da Computação	12	0	0	0	0	0	
Artigo em revista Qualis B5 em Ciência da Computação	10	0	0	1	0	10	
Artigo em revista Qualis C em Ciência da Computação	4	0	0	0	0	0	
Artigo em revista (sem Qualis) em Ciência da Computação	4	0	0	0	0	0	
Artigo em conferência Qualis A1 em Ciência da Computação	20	0	1	0	0	20	
Artigo em conferência Qualis A2 em Ciência da Computação	18	0	0	0	0	0	
Artigo em conferência Qualis B1 em Ciência da Computação	16	0	0	0	0	0	
Artigo em conferência Qualis B2 em Ciência da Computação	12	0	1	0	0	12	
Artigo em conferência Qualis B3 em Ciência da Computação	10	0	0	0	0	0	
Artigo em conferência Qualis B4 em Ciência da Computação	8	0	2	2	0	32	
Artigo em conferência Qualis B5 em Ciência da Computação	6	0	0	0	0	0	
Artigo em conferência Qualis C em Ciência da Computação	4	0	0	0	0	0	
Artigo em conferência (sem Qualis) em Ciência da Computação	4	0	2	7	0	36	
Capítulo de Livro	4	0	0	0	0	0	
Livro	4	0	0	0	0	0	
Edição de Livro	4	0	0	0	0	0	
		Pontuação sem Qualis e Qualis C:				20	
		Pontuação fora do Qualis restrito:				70	
		Pontuação total:				100	
		Pontuação normalizada:				60	
Orientações							
{Número total de orientações ou co-orientações em andamento}		2016	2015	2014	2013	Total	
Doutorado	0	0	0	0	0	0	
Mestrado	0	0	1	1	0	2	
Projeto Final	0	0	0	0	0	0	
Iniciação Científica	0	0	0	0	0	0	
Orientações concluídas							
Peso: 20,00%		Limite de pontos:		20			
	Peso	2016	2015	2014	2013	Total	
Doutorado	8	0	0	0	0	0	
Mestrado	6	0	1	2	1	24	
Projeto Final	4	0	0	0	0	0	
Iniciação Científica	2	0	0	0	0	0	
		Total no período:				20	
Participação em projetos de pesquisa nos últimos 5 anos							
Peso: 15,00%		Limite de pontos:		5			
	Peso	2016	2015	2014	2013	Total	
Participação em projeto de pesquisa nacional	1	0	1	0	2	4	
Coordenação de projeto de pesquisa nacional sem financiamento (máx. 1 por	2	0	0	0	1	2	
Coordenação de projeto de pesquisa nacional com financiamento (máx. 1 por	3	0	0	0	0	0	
Publicações com co-autores internacionais (máx. 1 por ano)	1	0	0	1	1	2	
Participação em projetos de pesquisa internacionais	2	0	0	0	0	0	
Coordenação de projeto de pesquisa internacional sem financiamento (máx. 1	3	0	0	0	0	0	
Coordenação de projeto de pesquisa internacional com financiamento (máx. 1	5	0	0	0	0	0	
		Total no período:				15	
Organização de eventos científicos, Participação em Comitês de Programas de Eventos Científicos e Funções em instituições relacionadas							
Peso: 5,00%		Limite de pontos:		5			
	Peso	2016	2015	2014	2013	Total	
Função em instituição relacionada (CAPES, SBC, CNPq) (por local)	1	0	0	0	0	0	
Organização de evento nacional	2	0	0	0	0	0	
Organização de evento internacional	1	0	0	0	0	0	
Participação em Comitês de Programas de eventos científicos	1	0	0	0	0	0	
Revisor de periódico	1	1	1	1	3	6	
Membro de comitê editorial de periódico	1	0	0	0	0	0	
Bolsista de produtividade (CNPq, FAPERJ)	3	0	0	0	0	0	
		Total no período:				5	

Figura 10 – Planilha de Cálculo da Auto Avaliação

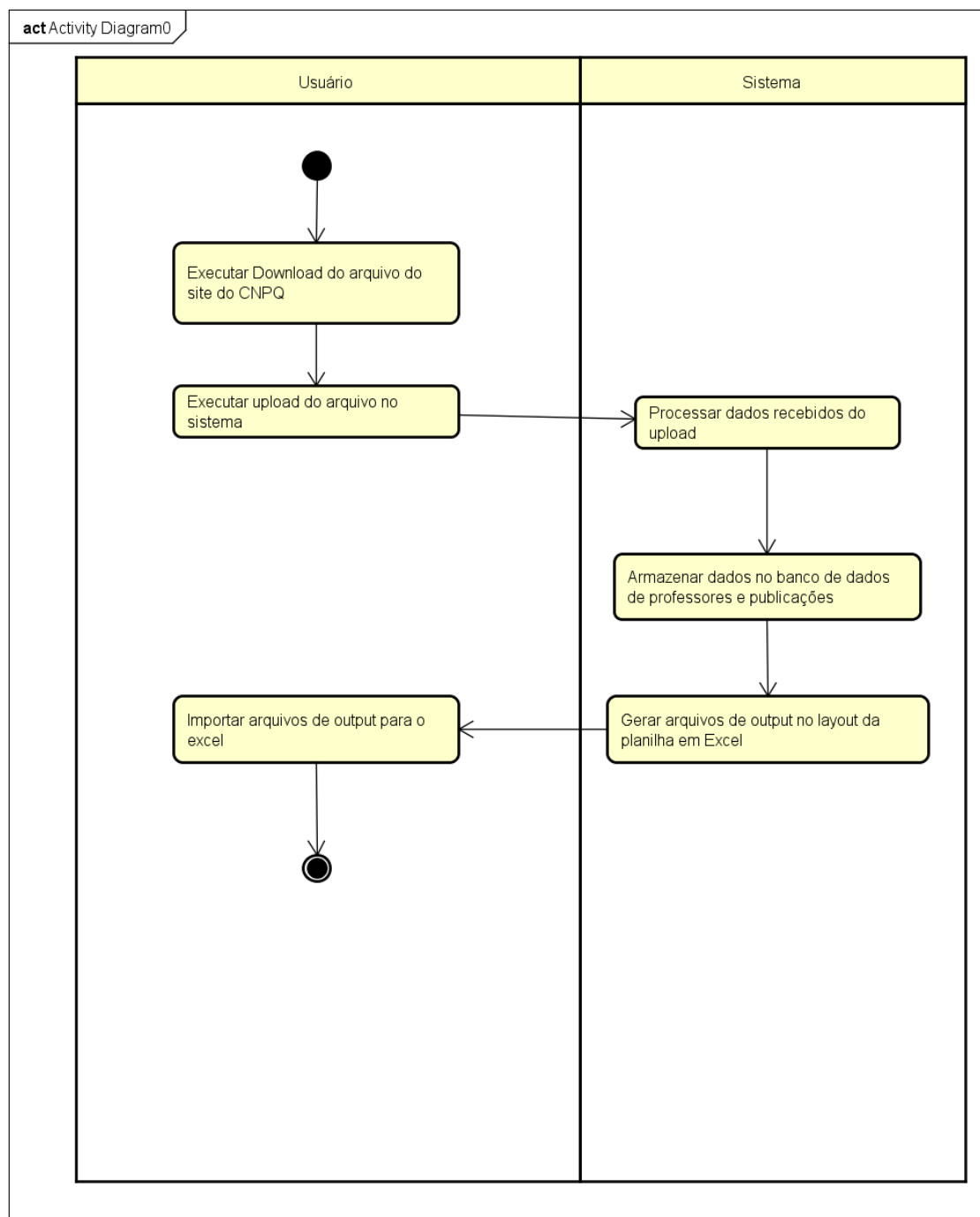


Figura 11 – Passo-a-passo de uso do sistema

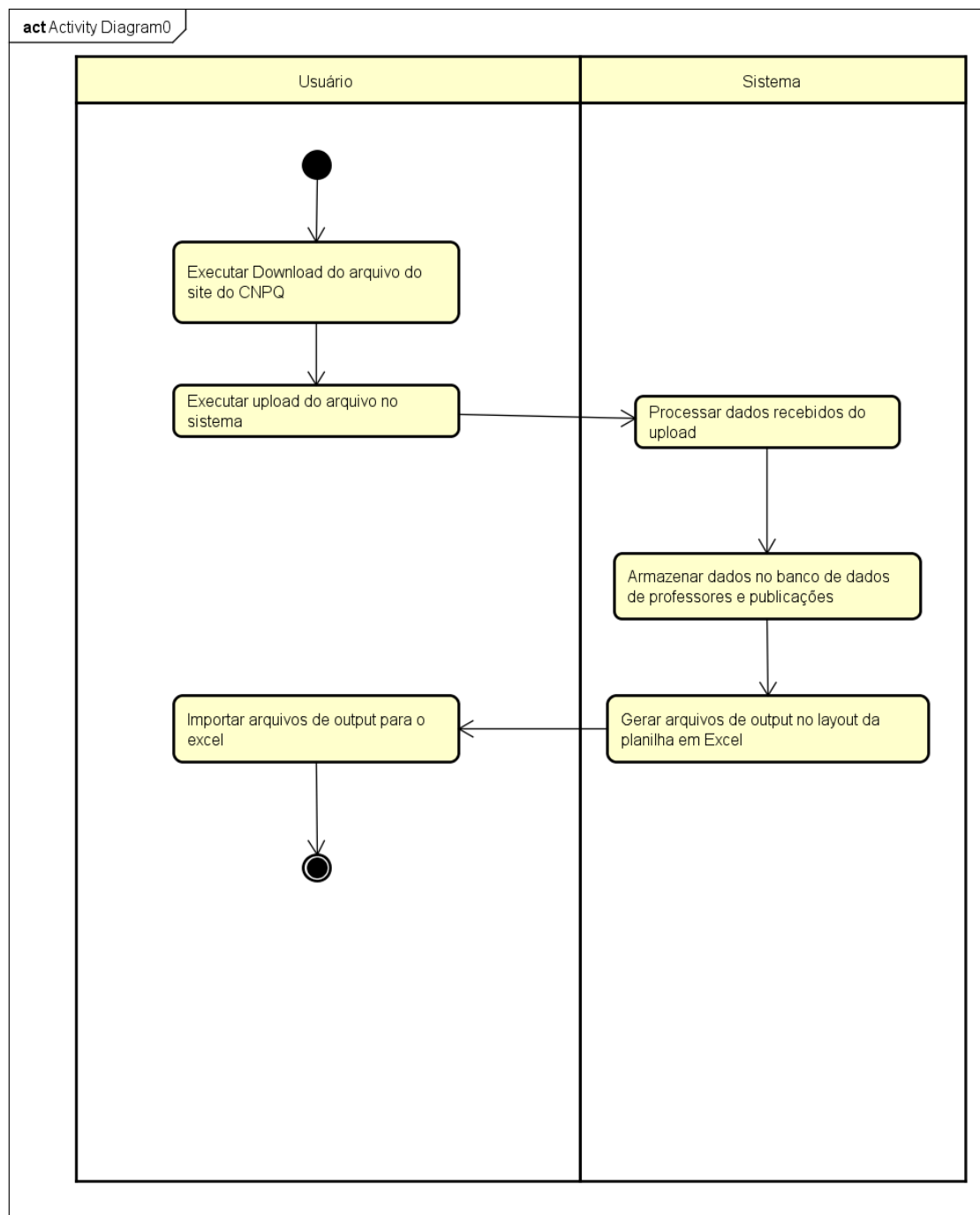


Figura 11, temos uma visão dos passos executado do sistema. Ele visa ser capaz de extrair os dados do arquivo baixado do site do CNPq e armazená-los de forma estruturada em banco de dados para que os cálculos necessários para a geração da pontuação possam ser efetuados. Dessa forma, o sistema minimiza o trabalho manual inerente a todo o processo de inserção de informações. Devido à complexidade da aplicação, apenas o cálculo relacionado às produções bibliográficas foi desenvolvido e

disponibilizado via aplicação web. A solução encontrada para os demais critérios, após todos os tratamentos necessários, foi o sistema exportar os arquivos com as informações no layout da planilha para que possam ser importados manualmente na planilha em Excel. Esta planilha possui toda a inteligência de pontuação e peso dos critérios.

Conforme a Figura 12 **Erro! Fonte de referência não encontrada.**, temos a modelagem das relações e tabelas que possibilitam a estruturação dos critérios de orientação, participações e publicações de artigos ou livro que cada professor se envolveu no período avaliado.

O modelo de dados da aplicação tem como seu principal conceito a *Publicação Bibliográfica*, que por sua vez possui uma *Área* relacionada (por exemplo, Ciência da Computação) e um auto relacionamento chamado *Inconsistência* que armazena novas entradas de dados que resultam em possíveis inconsistências ainda não tratadas pelo sistema. Além disso, publicações são publicadas em *Fóruns* e são produzidas por um ou mais *Autores*. *Autores* podem estar envolvidos em alguma orientação, seja de projeto final de graduação, dissertação de mestrado ou tese de doutorado.

Autor possui um auto relacionamento chamado *Citação*, que procura mapear diferentes formas de referência a uma mesma pessoa através das formas de citações conhecidas para um determinado autor.

Fóruns estão relacionados a *Qualis*³, que possuem seus índices apresentados em (A1, A2, B1, B2, B3, B4, B5, C ou SEM QUALIS), e este a *Tipo_Publicação*, que apresenta os tipos de publicação associado ao Qualis (“Artigo em Periódico”, “Artigo em Conferência”, “Livro”, “Capítulo de Livro”, “Edição de Livro”), para que a identificação do qualis do fórum seja indicada de forma correta, no Apêndice IV na Figura 34 temos o exemplo do DAO que executa a busca do qualis de uma determinada publicação de acordo com os parâmetros recebidos, neste projeto utilizamos uma extração pontual do Qualis, sem nenhum tipo de processo de atualização e manutenção de histórico do Qualis. Além disso, professores podem participar de fóruns como revisores de periódicos, membro do corpo editorial, etc.

Fórum também possui *Sinônimo*, que identifica nomes divergentes que se referem ao mesmo fórum. Além das diversas *Edições* de um fórum que é referenciada pelo seu título sem a descrição da edição.

³ <http://www.capes.gov.br/component/content/article?id=2550:capex-aprova-a-nova-classificacao-do-qualis>

A entidade *Cadastro* se refere a relação dos logins do sistema e a entidade *Upload* é necessária para salvar informações relacionadas ao arquivo que é feito o upload na ferramenta, tais como, data e hora do upload do arquivo, nome do arquivo XML, caminho físico onde o arquivo se encontra salvo e nome do autor referente ao arquivo, estas informações são necessárias para que seja possível recuperar informações em diferentes etapas da aplicação sem necessidade de transportar grandes quantidades de dados via requisições GET ou POST, por serem entidades mais relacionadas ao sistema não estão contempladas na modelagem.

Algumas tabelas foram adicionadas ao modelo, não por necessidade de modelar o minimundo, mas como um recurso para que possíveis ambiguidades de nomes fossem resolvidas automaticamente.

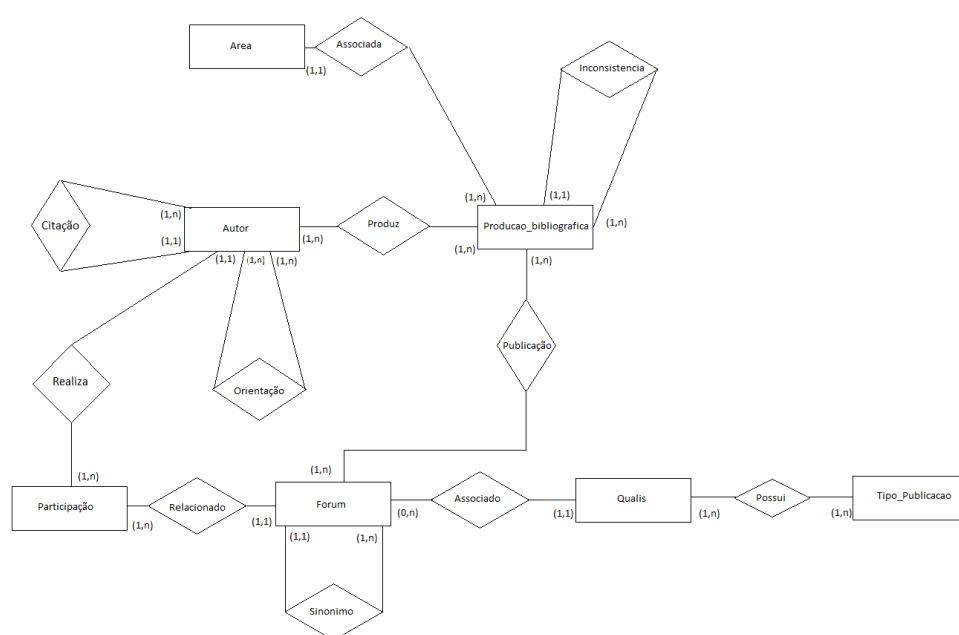


Figura 12 – Modelo conceitual versão final

3.2 Estrutura do projeto

O projeto foi construído empregando o modelo MVC, utilizando a linguagem Java e alguns frameworks que apoiaram no desenvolvimento do sistema, tais como Struts2⁴,

⁴ <https://getbootstrap.com/>

Bootstrap⁵, JXL⁶ e StaX⁷, cada uma delas teve papel fundamental no sistema, seja viabilizando processos através de funcionalidades novas, ou agilizando a implementação de código que sem o auxílio das mesmas tornaria o projeto mais complexo, dificultando também sua manutenção e evolução posterior.

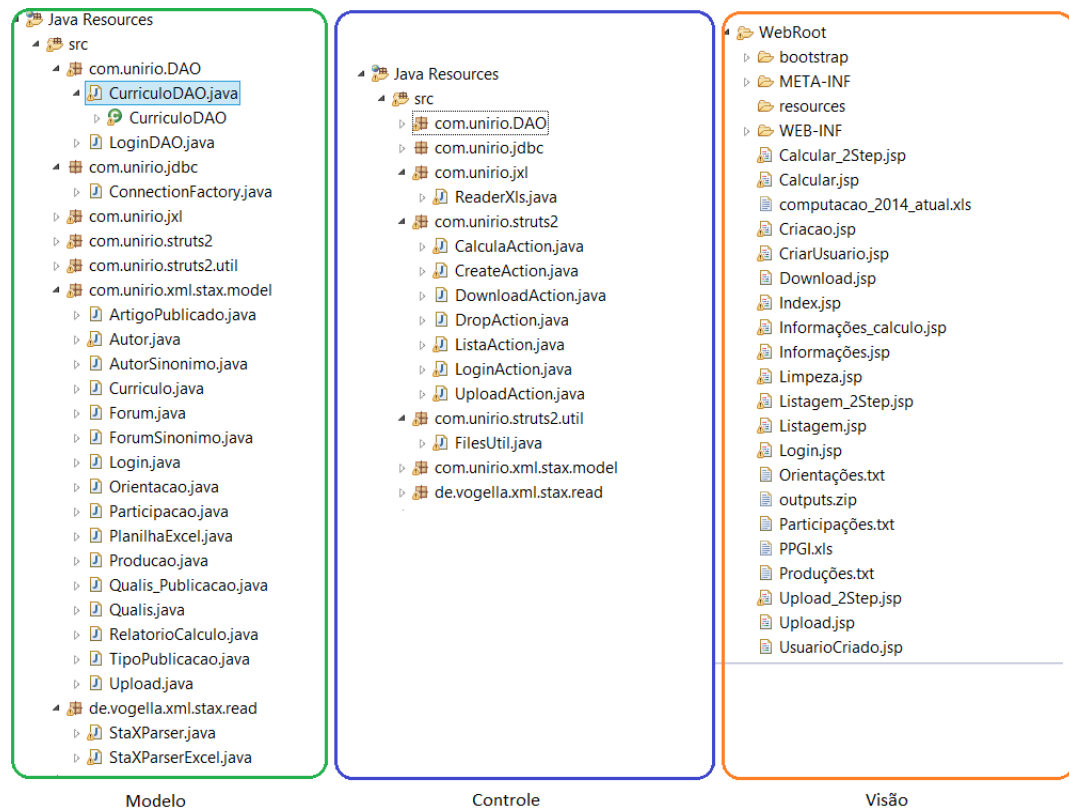


Figura 13- Estrutura MVC dos arquivos do projeto JAVA

A Figura 13 ilustra a estrutura do projeto em JAVA, no diversos pacotes relacionados ao modelo, temos as classes de dados com os atributos necessários à persistência dos dados no banco de dados, além de classes genéricas buscando uma arquitetura dentro do projeto coesa, de fácil manutenção e desenvolvimento como, por exemplo, as bibliotecas StaX de leitura e escrita do xml, bibliotecas da framework struts2. Já na parte do Controle, estão as classes responsáveis por gerenciar todo o fluxo lógico e regras pertinentes a cada conceito estruturado no projeto de acordo com os casos de uso que serão abordados posteriormente. Por fim na visão, temos todos os arquivos

⁵ <https://struts.apache.org/>

⁶ <http://jexcelapi.sourceforge.net/>

⁷ <https://en.wikipedia.org/wiki/StAX>

responsáveis pelas páginas web que o usuário final irá navegar, a fim do usuário cadastrar dados e visualizar os retornos de cada funcionalidade que o sistema possui.

3.3 Tecnologias Utilizadas

- **Struts2**

Framework que oferece padrão MVC para desenvolvimento web, facilitando o gerenciamento, reutilização e a manutenção do código. Possui algumas características que o popularizaram como framework de desenvolvimento, tais como, convenções ao invés de configurações, oferecendo mapeamento de ações e valores de resultados retornados para que as páginas JSP possam ser apresentadas, anotações ao invés de configuração XML, injeção de dependências, plug-ins, facilidade nos testes de software, simplificando o teste das ações, interceptadores e outras classes do Struts2 [6].

- **BootStrap**

Criado para resolver inconsistências no padrão de desenvolvimento dos projetos da equipe de engenharia do Twitter. Após notar-se o potencial do que havia sido criado, o bootstrap se tornou um projeto de desenvolvimento de software livre, voltado para a camada front-end. Composto por elementos customizáveis, obtidos por uma combinação de HTML, CSS e JavaScript. Ao utilizar este framework, possibilitamos agilidade no desenvolvimento, sem riscos de conflitos e design responsivo provido pelo próprio framework de forma transparente [7], cuja técnica se baseia na estruturação do html e css para adaptação da resolução do browser sem necessidade de se definir todas as variantes de estilos necessárias

- **JXL**

API desenvolvida para possibilitar a leitura e escrita de planilhas Excel. Seu desenvolvimento se deu a partir de do entendimento das funcionalidades do Excel por se tratar de um software proprietário [8]. Portanto, nem todas as funcionalidades disponíveis

no Excel estão presentes na API. Porém a versão utilizada neste projeto, possibilitou de forma funcional a manipulação de planilhas em operações de escrita e leitura.

- StaXParser

API que possibilita a escrita e leitura de arquivos XML criado de acordo com as seguintes regras:

1. Todas as *tags* abertas são devidamente fechadas;
2. Não há sobreposições de *tags*;
3. Existe um e somente um elemento raiz.

Através desta API, foi possível extrair as informações pelo arquivo XML que o portal Lattes disponibiliza, realizando leitura a cada nó e consultando seu DTD (Document Type Definition), além de estudar os próprios dados provenientes dos currículos tratados [9].

3.4 Inconsistências solucionadas

Várias inconsistências foram encontradas durante o processamento dos arquivos, sendo necessário desenvolver soluções específicas para as mesmas a fim de ter como resultado uma base de dados consistente. A seguir, estas inconsistências são descritas.

Uma inconsistência foi o fato de várias informações que contextualmente eram iguais, mas tinham grafias diferenciadas. Exemplificando, um evento como *Simpósio Brasileiro de Sistemas de Informação* era tratado como um fórum diferente devido às suas edições estarem escritas com algarismo romano ou árabe. Logo, casos onde a grafia se dava por “IX Simpósio Brasileiro de Sistemas de Informação” e “9º Simpósio Brasileiro de Sistemas de Informação” eram tratados como entradas diferentes, apesar de conceitualmente estarem se referindo ao mesmo fórum. Em outras oportunidades, o mesmo fórum acima, ao possuir edições diferentes, era tratado como fóruns essencialmente diferentes, onde-se o correto seria afirmar que o fórum é o mesmo, exceto sua edição.

Para tratar tal inconsistência, foi criado um passo verificador onde todos os nomes de fóruns eram lidos a procura de expressões que indicassem uma edição. O padrão

considerado para busca se deu pela expressão em números ordinais ou algarismos romanos relativos à edição até a sua quinquagésima edição, como descrito abaixo.

- [1^a, 2^a, 3^a, 4^a, 5^a, ..., 50^a]
- [1º, 2º, 3º, 4º, 5º, ..., 50º]
- [I, II, III, IV, V, ..., L]
- [1st, 2nd, 3rd, 4th, 5th, ..., 50th]

Após a identificação, tratamos o nome do fórum juntamente com a edição como um sinônimo e efetuamos o processo de inserção na tabela de sinônimos de fóruns. Além disso, o nome sem a edição foi tratado como o fórum de fato, sendo o sinônimo associado ao mesmo para futuras pesquisas e aumento da assertividade de busca de um possível fórum já cadastrado no banco de dados.

Além disso, foi criada uma heurística onde se 80% do nome do fórum for semelhante com algum dado já cadastrado no sistema, esta nova inserção será considerada um sinônimo do padrão de semelhança identificado.

A fórmula é apresentada na Figura 14 e é independente de idioma.

$$\text{Expressão se dá por,} \quad \frac{\text{Quantidade de palavras correspondentes}}{\text{Quantidade de palavras analisadas}} = \text{Se } 0,8 \leq \text{resultado} < 1, \text{ afirma-se que existe um padrão}$$

Figura 14 - Fórmula adotada para identificar semelhança de nomes de fóruns

Como exemplo para a fórmula acima, temos:

- Entrada 1: “SDMS - IV Simpósio de Desenvolvimento e Manutenção de Software da Marinha“
- Entrada 2: “Simpósio de Desenvolvimento e Manutenção de Software da Marinha”

Considerando que o caractere “-“ é retirado antes do início da comparação temos o seguinte resultado, entrada 1 composta de 11 palavras e entrada 2 composta de 9 palavras, realizando o cálculo temos um coeficiente de 0,81 e, portanto, será considerado um sinônimo.

Ao manipular as informações de autores, foi levantada a necessidade de mapear possíveis sinônimos ou abreviações referentes a um autor específico. Esta etapa é muito importante, e somente foi possível pois cada currículo inserido possui sua listagem de sinônimos ou abreviações, ao se carregar um novo currículo o sistema varre todas as entradas anteriores de nomes de autores, e caso haja algum padrão encontrado este nome de autor que deveria ser na realidade um sinônimo ou abreviação de autor é deletado da base de dados e todas as suas referências são apontadas para o nome principal que consta no currículo carregado. Conforme a carga de currículos na base de dados vai sendo realizada, a tendência é obter uma tabela de informações de autores coesa, e que consiga identificar uma variedade de sinônimos ou abreviações de autores.

Exemplificando, ao se carregar um currículo que não seja de Leonardo Guerreiro Azevedo, é possível termos publicações associadas a abreviação AZEVEDO, L. G. Se for identificado que Leonardo Guerreiro Azevedo é sinônimo de AZEVEDO, L. G. ou sua forma de citação, a publicação proveniente do currículo anterior terá sua referência automaticamente atrelada também ao Leonardo. A entrada AZEVEDO, L. G. deixará de ser considerada um nome de autor e será tratada como um sinônimo. Com esta técnica é garantida a não duplicação de publicações na base de dados proveniente de currículos diferentes.

A verificação de sinônimos de autores e fóruns se mostrou obrigatória no desenvolvimento do sistema, a partir do momento que se foi necessário carregar mais de um currículo nas bases de dados, sem estes tratamentos seria impossível manter um banco de dados coerente, sem repetição, seja literalmente ou conceitualmente.

Como trabalho futuro, é sugerida a implementação de técnicas de linguagem natural de processamento para que possibilite o sistema interpretar e decidir como resolver as inconsistências, através da análise de informações morfológicas (léxicas), sintáticas (regras gramaticais) e semânticas (significados), armazenadas em um dicionário, juntamente com as palavras que o sistema compreende [2].

3.5 Casos de Uso

Com o intuito de documentar as funcionalidades do projeto, foi criado um diagrama composto por alguns casos de uso apresentado na (Figura 15):

- **Logar no Sistema:** Refere-se a etapa de acessar o sistema com um usuário previamente cadastrado.

- Solicitar criação de usuário: Etapa onde fornece-se o nome de usuário, senha e indica-se o perfil do usuário (comum ou administrador).
- Listar produções: Lista todas as produções cadastradas, somente com informações essenciais, tais como nome e ano da publicação.
- Calcular critérios: Cálculo inicial do primeiro macro critério relativo a publicação de produções em periódicos, conferências ou publicações de livros.
- Baixar XML: Usuário realiza busca no portal Lattes, pelo nome da pessoa ou assunto de interesse, após a escolha do currículo, executa o download do arquivo XML que será carregado posteriormente no aplicativo web criado.
- Carregar informações do XML: Carrega o arquivo XML no sistema, extrai e estrutura todas as informações necessárias no modelo relacional proposto.
- Apagar e criar banco de dados: Estes dois casos, estão relacionados à manutenção do sistema e só podem ser executados pelo administrador do sistema. Toda a estrutura de tabelas do banco de dados descrita na **Erro! Fonte de referência não encontrada.**¹ que suporta a aplicação será deletada ou criada através destas funcionalidades, com exceção da tabela de cadastro que fica responsável pelo gerenciamento de senhas da aplicação.

A Figura 15 5 ilustra todo o cenário de casos de uso do sistema exposto.

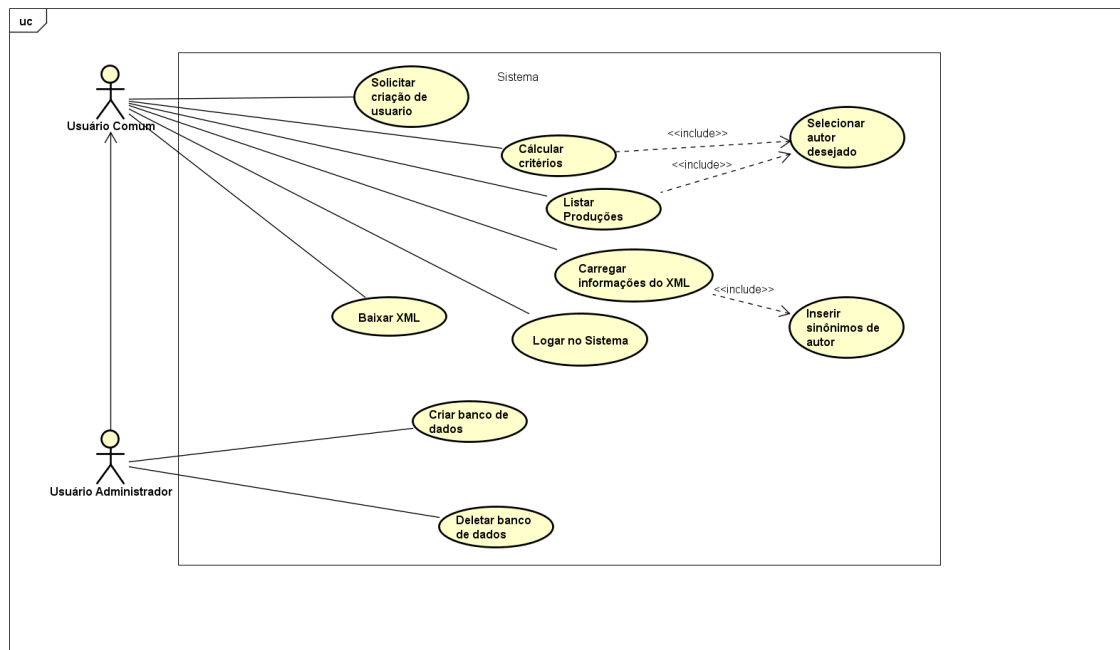


Figura 15 – Casos de uso do sistema

4 Funcionalidades do SGPP (Sistema de Gerenciamento de Produtividade de Professores)

4.1 Visão Geral

Nesta seção ilustraremos as funcionalidades da aplicação que foram desenvolvidas com o intuito de atender aos casos de uso mapeados anteriormente. Além de descrevermos de forma mais aprofundada o que cada parte do projeto executa. A intenção é dar mais clareza e abstração da parte técnica, da mesma forma que relaciona toda a descrição técnica, diagramas e modelos abordados anteriormente.

4.2 Baixar XML

O fluxo se inicia através da necessidade de obtermos a informação que será trabalhada posteriormente, portanto, deve-se acessar o portal do lattes conforme Figura 16. Devido a modificações na estrutura do portal no decorrer do projeto, tornou burocrática a possibilidade de extrações automatizadas impedindo de serem realizadas sem solicitação formal da instituição de ensino. Anteriormente esta etapa era realizada de forma semiautomática via script Python, onde era apenas necessário a execução do script, definindo como parâmetro os currículos desejados.



Figura 16 – Portal Lattes

Após solicitar a busca de um determinado currículo é necessário parametrizar a busca, seja por nome ou assunto. É possível também refinar a busca através de filtros ou opções avançadas que a ferramenta de busca do site disponibiliza conforme Figura 17.

Figura 17 – Pesquisa de Currículo

Devido à nossa busca ter sido direcionada especificamente a uma pessoa, tivemos como retorno da pesquisa apenas um currículo, porém nesta etapa seria possível escolher um dos possíveis resultados retornados caso houvesse mais de um conforme Figura 18.

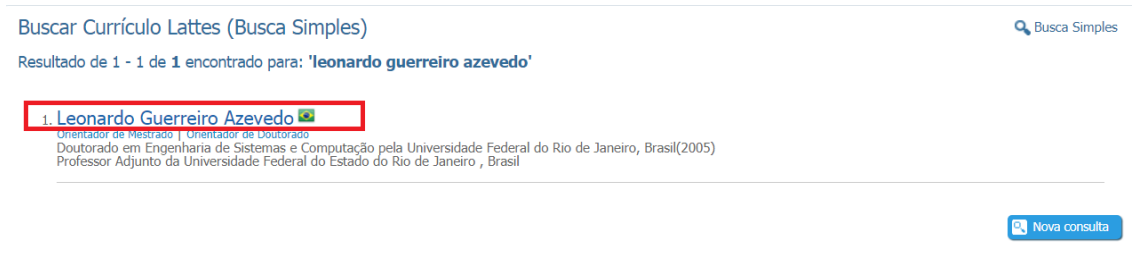


Figura 18 - Seleção do currículo

Ao acessar a página Web do currículo escolhido, no canto superior esquerdo, conforme Figura 19 é possível solicitar o download das informações retratadas na página via formato XML, que posteriormente trataremos em nossa aplicação.



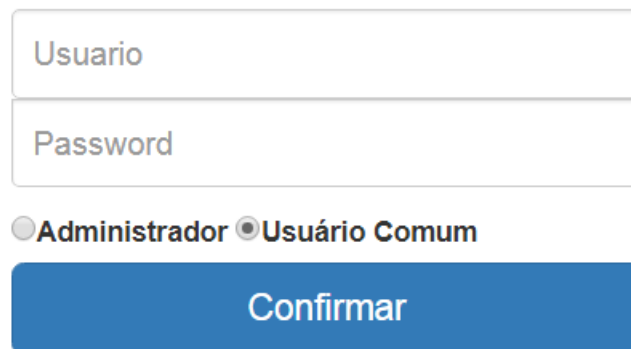
Figura 19 - Download em XML do currículo

4.3 Solicitar Criação de Usuário

Para acessar a aplicação web criada temos dois perfis de usuários, administrador e usuário comum, solicitando apenas o identificador do usuário e sua senha (

Figura 20). O cadastro de usuário foi implementado, porém seria necessário criar uma funcionalidade de validação de acessos administrativos futuramente pelos administradores já cadastrados no sistema, visto que não existe restrição prévia do perfil de login passível de criação.

Cadastro de Usuario



A user registration form with two input fields: 'Usuario' and 'Password'. Below the fields are two radio buttons: 'Administrador' (unselected) and 'Usuário Comum' (selected). At the bottom is a blue 'Confirmar' button.

Usuario
Password

☐ Administrador ☒ Usuário Comum

Confirmar

Figura 20 – Cadastro de Usuário

Para fins de experiência do usuário, caso tudo ocorra corretamente uma mensagem de sucesso na operação é retornada após a criação do usuário, conforme Figura 21.

Usuário criado com sucesso



A blue button with the text 'Back'.

Back

Figura 21 – Mensagem de sucesso da aplicação

4.4 Logar no Sistema

Na Figura 22, temos a tela inicial que será apresentada no momento que o endereço da aplicação for inserido no navegador, sendo uma simples tela de login, com a opção de criação de usuário, cuja funcionalidade foi abordada anteriormente.

Painel de Login



Usuario

Password

Entrar

Criar Usuário

Figura 22 – Tela de acesso ao sistema através de usuário e senha

De acordo com o perfil do usuário existem dois fluxos que a aplicação poderá direcionar, caso seja um usuário comum a tela inicial será como descrita na Figura 23, contendo apenas a opção inicial de upload do arquivo XML.

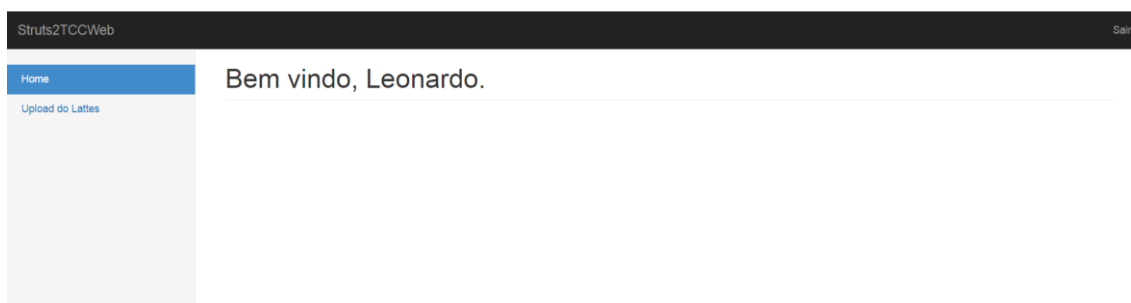


Figura 23 – Tela inicial para usuário comum

No perfil administrativo as opções de funcionalidades são maiores, visto que determinadas opções da aplicação não devem ser acessadas pelo o usuário comum conforme Figura 24. Todo esse fluxo é configurável através do framework Struts2, que possibilita o redirecionamento das páginas conforme parâmetros passados via método GET ou POST do HTTP.

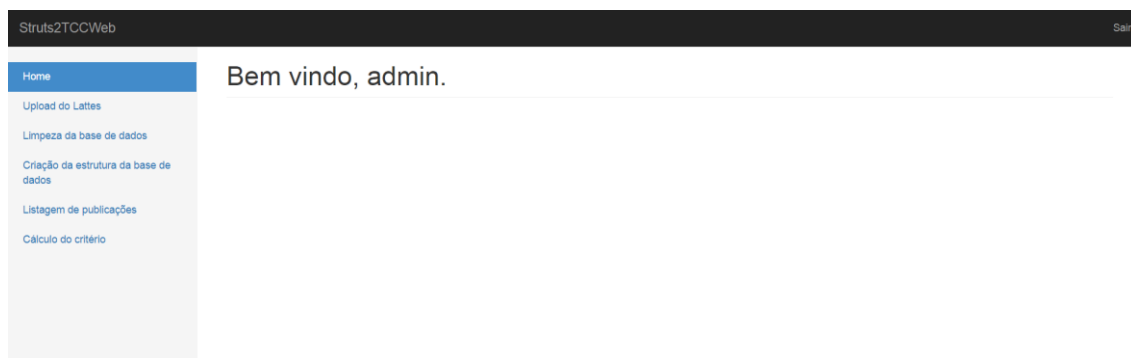


Figura 24 – Tela inicial para administrador

4.5 Criar e deletar banco de dados

Para atender os casos de uso, de exclusão e criação das estruturas de banco de dados do sistema, existem duas opções na barra de navegação, que se encontram disponíveis apenas para administradores do sistema. Ao término da execução de cada procedimento o sistema irá retornar as mensagens descritas nas Figura 25 e Figura 26.

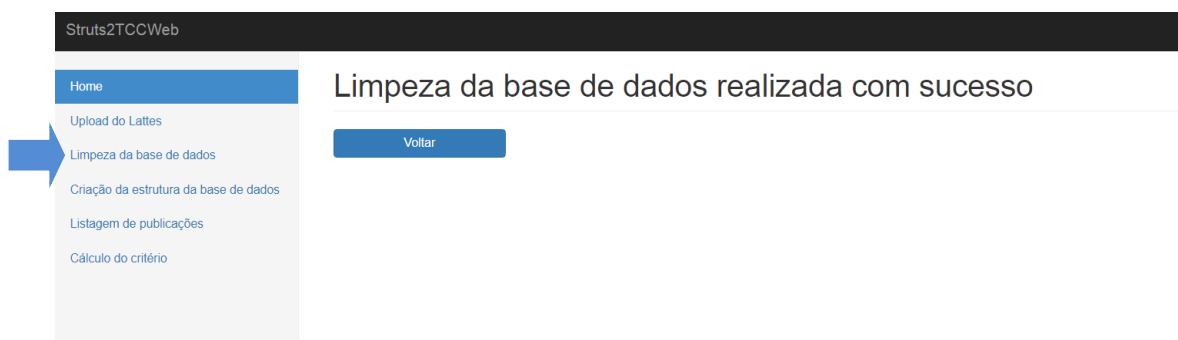


Figura 25 – Mensagem de sucesso para exclusão de base de dados

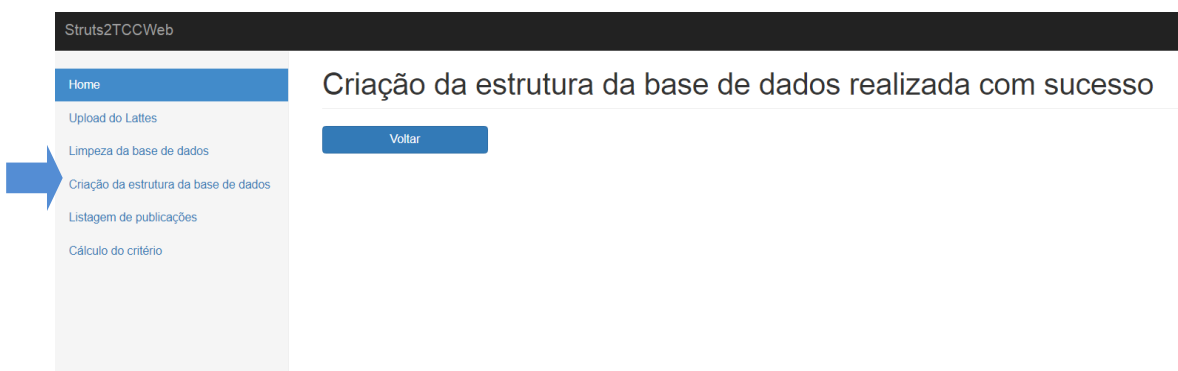


Figura 26 – Mensagem de sucesso para criação das bases de dados do sistema

4.6 Carregar informações do arquivo XML

Com o arquivo XML referente a um currículo escolhido salvo localmente, a Figura 27 retrata a etapa de seleção e upload das informações contidas no XML.



Figura 27 - Upload do arquivo XML

Na Figura 28, é exemplificada a etapa de validação dos possíveis sinônimos ou abreviações encontradas no arquivo carregado ou proveniente de informações anteriormente carregadas.



Figura 28 – Validação e inserção de novos sinônimos de autores

Por fim, após a carga das informações nas bases de dados da aplicação, é possível baixar arquivos de saída em texto, cujo exemplos estão contidos nos Apêndices I, II, III, organizados em Orientações, Participações e Produções, que serão importados posteriormente na planilha ilustrada na Figura 10. No Apêndice IV, na Figura 35 temos o exemplo do Controller responsável pela funcionalidade apresentada na Figura 29.

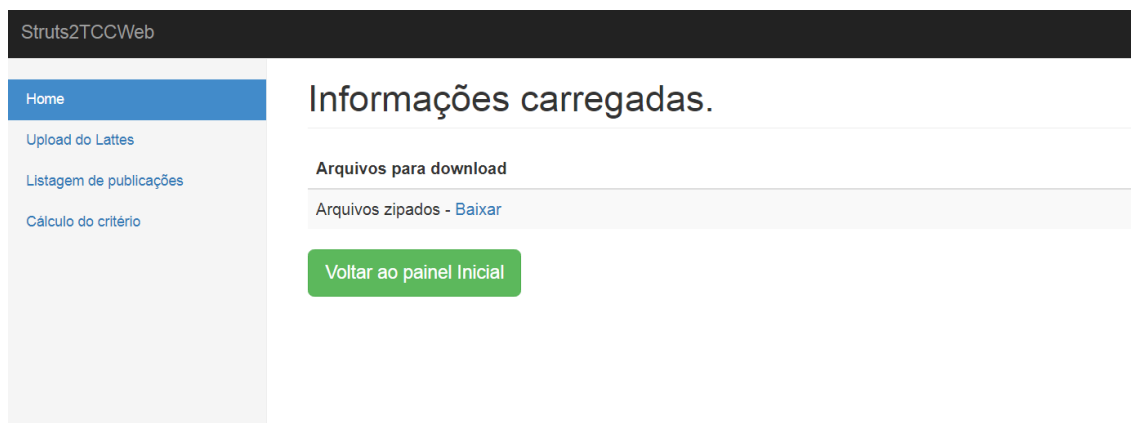


Figura 29 – Tela de download dos arquivos textos

4.7 Listar Publicações

Para listagem das publicações é necessário primeiramente, que se insira o nome de um autor ou parte do nome que se queira pesquisa como indica a **Erro! Fonte de referência não encontrada..**



Figura 30 – Busca por nome

Caso haja mais de um nome que atenda ao parâmetro buscado anteriormente, entraremos em um fluxo de exclusão de ambiguidade, sendo solicitado ao usuário que escolha um dos nomes encontrados na base de dados, conforma a Figura 31.

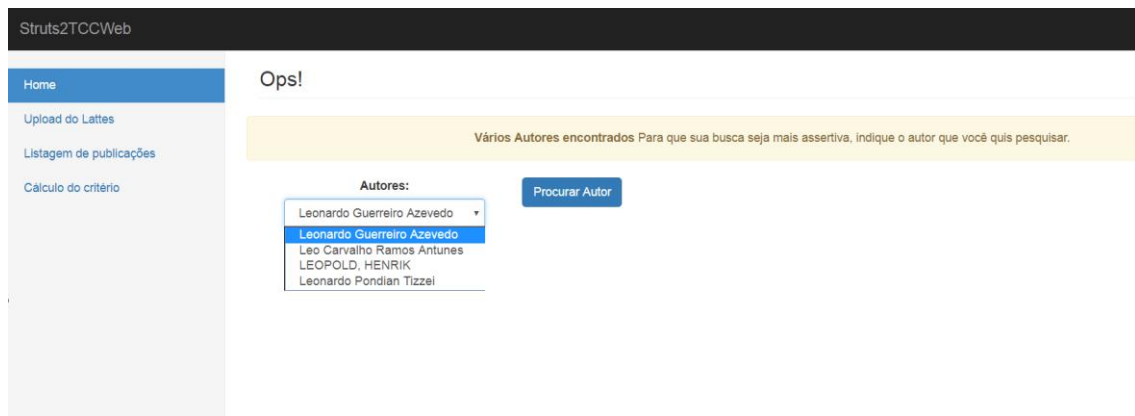


Figura 31 – Tratamento de ambiguidade na pesquisa

Na Figura 32, é demonstrado a aplicação retornando a listagem das publicações.

Informações carregadas.

#	Título da Publicação	Ano da Publicação
1	Comparacao de Funcoes de Avaliacao em Revisao de Teorias Probabilisticas de Primeira-ordem	2005
2	Further Experimental Results of Probabilistic First-order Theory	2005
3	Probabilistic First-order Theory Revision from Examples	2005
4	Revisao de Teorias Probabilisticas de Primeira-ordem	2005
5	Search-based Class Fiscretization for Hidden Markov Model for Regression	2004
6	Hidden Markov Model for Regression in Eletic Load Forecasting	2003
7	Revision of First-order Bayesian Classifiers	2002
8	Theory Refinement of Bayesian Logic Program	2001
9	Theory Refinement of Bayesian Logic Programs	2001
10	PFORTE: Revising Probabilistic FOL Theories	2006
11	Combinando Invencao de predicados e revisao de teorias probabilisticas de primeira-ordem	2007
12	Combining Predicate Invention and Revision of Probabilistic FOL Theories	2006
13	Identificacao de Servicos a partir da Modelagem de Processos de Negocio	2009
14	Revising Probabilistic Prolog Programs	2006
15	Revisando Redes Bayesianas atraves da Introducao de Variaveis nao-observadas	2009
16	A Method for Service Identification from Business Process Models in a SOA Approach	2009

Figura 32 – Listagem das publicações carregadas na base de dados

4.8 Calcular Critério

Para a funcionalidade de cálculo do critério, se faz necessário as mesmas etapas descritas nas Figura 30 e Figura 31, para que seja possível identificar o Autor cujo cálculo deverá ser realizado.

Cálculo efetuado com sucesso.

Publicações	Peso	2015	2014	2013	2012	Total
Artigo Em Artigo em Periódico A1 Em Ciência da Computação	30	0	0	1	0	30
Artigo Em Artigo em Periódico A2 Em Ciência da Computação	26	0	0	0	0	0
Artigo Em Artigo em Periódico B1 Em Ciência da Computação	20	0	0	0	0	0
Artigo Em Artigo em Periódico B2 Em Ciência da Computação	16	0	0	0	0	0
Artigo Em Artigo em Periódico B3 Em Ciência da Computação	14	0	2	1	1	56
Artigo Em Artigo em Periódico B4 Em Ciência da Computação	12	0	0	0	2	24
Artigo Em Artigo em Periódico B5 Em Ciência da Computação	10	0	1	0	0	10
Artigo Em Artigo em Periódico C Em Ciência da Computação	4	0	0	0	1	4
Artigo Em Artigo em Periódico SEM QUALIS Em Ciência da Computação	4	0	0	0	0	0
Artigo Em Artigo em Conferência A1 Em Ciência da Computação	20	0	0	0	0	0
Artigo Em Artigo em Conferência A2 Em Ciência da Computação	18	0	0	0	0	0
Artigo Em Artigo em Conferência B1 Em Ciência da Computação	16	0	0	0	0	0
Artigo Em Artigo em Conferência B2 Em Ciência da Computação	12	0	0	0	0	0
Artigo Em Artigo em Conferência B3 Em Ciência da Computação	10	0	0	0	0	0
Artigo Em Artigo em Conferência B4 Em Ciência da Computação	8	0	0	2	1	24
Artigo Em Artigo em Conferência B5 Em Ciência da Computação	6	0	0	0	0	0
Artigo Em Artigo em Conferência C Em Ciência da Computação	4	0	0	0	0	0
Artigo Em Artigo em Conferência SEM QUALIS Em Ciência da Computação	4	11	12	3	2	112
Artigo Curto em Conferência QUALIS A1 Em Ciência da Computação	20	0	0	0	0	0
Artigo Curto em Conferência QUALIS A2 Em Ciência da Computação	18	0	0	0	0	0
Artigo Curto em Conferência QUALIS B1 Em Ciência da Computação	16	0	0	0	0	0
Artigo Curto em Conferência QUALIS B2 Em Ciência da Computação	12	0	0	0	0	0
Artigo Curto em Conferência QUALIS B3 Em Ciência da Computação	10	0	0	0	0	0
Artigo Curto em Conferência QUALIS B4 Em Ciência da Computação	8	0	0	0	0	0
Artigo Curto em Conferência QUALIS B5 Em Ciência da Computação	6	0	0	0	0	0
Artigo Curto em Conferência QUALIS C Em Ciência da Computação	4	0	0	0	0	0
Artigo Curto em Conferência SEM QUALIS Em Ciência da Computação	4	0	0	0	0	0
Capítulo de Livro SEM QUALIS Em Ciência da Computação	4	0	0	0	0	0
Livro SEM QUALIS Em Ciência da Computação	4	0	0	0	0	0
Patente SEM QUALIS Em Ciência da Computação	0	0	0	0	0	0
Produção Técnica SEM QUALIS Em Ciência da Computação	0	0	0	0	0	0
Edição de Livro SEM QUALIS Em Ciência da Computação	4	0	0	0	0	0
Total Consolidado						260

Figura 33 – Tabela que provê o cálculo do primeiro critério

Como demonstrado na Figura 33, o relatório do cálculo do critério, especifica cada tipo de publicação, além de diferencia-lo pelo Qualis associado, com isso temos uma matriz que expressa a quantidade por ano de tipo de publicações realizadas e diferenciadas pelo Qualis. Essa diferenciação se dá, devido ao peso imposto a cada publicação de

acordo com seu Qualis identificado. A codificação desta View é realizada e demonstrada no código presente no Apêndice IV, na Figura 36.

5 Conclusão

Neste trabalho foi desenvolvido o sistema SGPP com a finalidade de automatizar parte do fluxo de cadastramento de informações e mensuração da avaliação de docentes do Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal do Estado do Rio de Janeiro (UNIRIO). Para o objetivo ser atingido foi-se utilizado diversos conhecimentos utilizados ao longo de toda a graduação, como:

- Modelagem de dados, sendo aplicado o modelo entidade-relacionamento.
- Bancos de dados relacionais e SQL com o intuito de manter a persistência das informações que o SGPP deve tratar.
- A linguagem JAVA foi escolhida para o desenvolvimento WEB da aplicação devido a familiarização inicial obtida ao cursar determinadas disciplinas, que embora não fosse suficiente para um desenvolvimento de uma aplicação por completo é fundamental como base para que o graduando possa entender suas necessidades e buscar formas de atendê-las.
- Diagramas de caso de uso e diagramas de atividade, para melhor entendimento dos requisitos do sistema e de seus fluxos de uso.

O desenvolvimento da aplicação foi complexo, seja pelas inúmeras dificuldades que o mundo apresentava devido à falta de conhecimento do graduando, o que implicava na necessidade de remodelar algum conceito, ou seja pela necessidade de aprofundamento na linguagem Java e utilização de frameworks como o Struts2 e o Bootstrap, além das API's StaXParser e JXL, que demandaram tempo para se obter domínio suficiente das mesmas e poder dar andamento no projeto.

Ao final do projeto foi gratificante visualizar todo o esforço, horas de estudos e dedicação em uma aplicação funcional e que atende ao que se foi proposto.

Com a criação do SGPP podemos elencar algumas de suas vantagens, em relação ao método anterior que era completamente manual e sem tratamentos, alguns deles são:

- Possibilidade de manter todas as informações cadastradas por diferentes pessoas em uma base de dados, que reduzirá a quantidade de informações duplicadas.

- Centralização das regras de inconsistências e tratamento das informações, garantindo que não haverá formas diferentes de tratar uma inconsistência
- Inserção das informações apenas uma única vez no site do CNPQ, sendo apenas necessário reenviar o arquivo do currículo lattes novamente ao SGPP para que as informações estejam atualizadas

A proposta inicial do SGPP seria a automatização de todo o fluxo de inserção de informações e disponibilização de todo o cálculo da avaliação via interface WEB. No entanto, devido às dificuldades apresentadas ao longo do projeto e sua complexidade por si só não foi possível concluir o projeto nos moldes iniciais, sendo somente carregadas as informações, tratadas e estruturadas de forma a possibilitar sua importação para a planilha de produção (de artigos, capítulos de livros etc.). Porém como trabalho futuros propõe-se que:

- Seja solicitado ao CNPQ a possibilidade da extração automatizada dos currículos de sua base de dados, com isso seria possível criar um fluxo automatizado fim a fim, já que hoje a existência de um captcha no site do CNPQ impossibilita este fluxo.
- Utilização de técnicas de processamento de linguagem natural, para a resolução das inconsistências mencionadas no projeto de forma mais assertiva.
- Implementação e disponibilização dos cálculos restantes pertinentes a avaliação dos docentes.
- Disponibilização do SGPP via aplicação em Cloud com a criação de um tutorial, o que implicaria em um diferencial visto que tal material não é encontrado de forma consistente atualmente, nem mesmo via web.
- Utilização da plataforma sucupira, como origem das informações de Qualis, por ser uma fonte em constante atualização
- Rever a utilização das entidades cadastro e upload, proporcionando valor contextual, por exemplo, relatórios de quantas vezes um currículo de determinado autor foi carregado na ferramenta e quem realizou o procedimento, quais usuários são heavy users da aplicação através da quantidade de logins registrados.

6 Referências Bibliográficas

- [1] Luz, G.B., Kuiawinski, D.L. (2006) “Mecanização, Automação e Automação – Uma Revisão Conceitual e Crítica”. XIII SIMPEP, Bauru, São Paulo. Disponível em < http://www.simpep.feb.unesp.br/anais/a.ais_13/artigos/1210.pdf >. Acessado em 16 de janeiro de 2018.
- [2] Silva, R. R., Lima, S. M. B. (2013) “Consultas em Bancos de Dados Utilizando Linguagem Natural”. Disponível em < <http://re.granbery.edu.br/artigos/MjQ0.pdf> >. Acessado em 17 de fevereiro de 2018.
- [3] Heuser, C. A. (2009). Projeto de Banco de Dados. 4 ed. Porto Alegre: Bookman, 1998.
- [4] Chen, P. P. S. (1976) The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems (TODS), 1(1), 9-36.
- [5] Deitel, H. M., Deitel, P. J. (2010) “Java como Programar”. São Paulo Prentice Hall, 6ª Edição.
- [6] Medeiros, H. (2013) “Introdução ao Struts2”. Disponível em <<http://www.devmedia.com.br/introducao-ao-struts-2/28127>> Acesso em: 17 de fevereiro de 2018
- [7] Tomazini, M, Lopes, L.F.B. (2015) “Web design responsivo – Bootstrap”. Disponível em <http://fcv.edu.br/admin/assets/repositorio_arquivo/014529452c44398122c6a634ad4e1755.pdf>. Acessado em 17 de fevereiro de 2018
- [8] “Java Excel API - A Java API to read, write, and modify Excel spreadsheets”. Disponível em< <http://jexcelapi.sourceforge.net>> Acesso em: 29 de outubro de 2017

[9] "Java and XML – Tutorial". Disponível em
<<http://www.vogella.com/tutorials/JavaXML/article.html>> Acesso em: 29 de outubro de
2017

Apêndice I – Saída de dados (Orientações) do SGPP

Tabela 2 – Exemplo de saída de dados das orientações

Aluno	Tipo	Em Andamento	Situação	Situação (2)	Ano Considerado	Início	Fim	Bolsa	Orientador Principal	Outro orientador PPGI	Outro orientador externo	Título do Trabalho	Projeto de Pesquisa	Observações
Rafael Brand Rodrigues	Mestrado		Concluído			2007		Sem bolsa	Nao			Filtro Raster de Tres Cores para Juncao de Objetos Espaciais		
Marcos Souza Veloso	Mestrado		Concluído			2011		Sem bolsa	Sim			CG-OLAP: Uma Estrategia para Otimizacao de Consultas Espaciais sobre Clusters de Banco de Dados		

Apêndice II – Saída de dados (Participações) do SGPP

Tabela 3 – Exemplo de saída de dados das participações

Ano	Tipo de Participação	Função em Instituição	Organização de Evento	Tipo de Local	Evento / Veículo / Instituição	Qualis	Título / Descrição
2009	Revisor de periodico				Journal of the Brazilian Computer Society	SEM QUALIS	
2009	Revisor de periodico				iSys - Revista Brasileira de Sistemas de Informacao	SEM QUALIS	

Apêndice III – Saída de dados (Produções) do SGPP

Tabela 4 – Exemplo de saída de dados das produções

Situação	Aceitação	Ano	Artigo	Tipo	Local	Qualis Computação	Projeto de Pesquisa	Veículo de publicação	Título	Autor 1	Autor 2	Autor 3	Autor 4	Autor 5	Outros Autores	Número Professores PPGI	Número Discentes PPGI	Número Participantes fora do PPGI	Possui Autor Internacional?	Uma das 3 Produções Relevantes do Ano?	Pontuação CAPE S	Pontuação PPGI	Pontuação Individual Relativa
Publicado		2000		Artigo em Conferência	Nacional	SEM QUALIS		XV Simposio Brasileiro de Banco de Dados	A Raster Approximation for Processing of Polyline Joins	Leonardo Guerreiro Azevedo	Geraldo Zimbrão	Janaína de Souza	Rodrigo Salvador Monteiro										
Publicado		2002		Artigo em Conferência	Nacional	B2		SIMPOSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE	Gerenciamento de Documentos XML no GOA	Leonardo Guerreiro Azevedo	Rodrigo Salvador Monteiro	Marta Mattoso	Maria Claudiana Cavalcanti	Humberto Vieira	Robson Pinheiro, Carlete Marques, Fatima Cristina Goncavels, Claudia Werner								

Apêndice IV – Codificação do projeto

Nesta seção serão apresentadas algumas seções de códigos da aplicação desenvolvida para acesso a toda aplicação e scripts exemplificados nesta etapa foi disponibilizado no bitbucket toda a aplicação e scripts de criação e carga do banco no repositório que pode ser acessado através do link <https://bitbucket.org/leoss21/tcc-leonardo.silva>.

```
private int buscaClassificacaoQualis(long id_qualis, String tipo) {
    try {
        String sql= "SELECT ID_QUALIS_PUBLICACAO FROM QUALIS_PUBLICACAO
        AS QP,QUALIS AS Q, TIPO_PUBLICACAO AS TP WHERE Q.ID_QUALIS=? AND
        Q.ID_QUALIS=QP.ID_QUALIS AND TP.DSC_TIPO_PUBLICACAO=? AND
        TP.ID_TIPO_PUBLICACAO=QP.ID_TIPO_PUBLICACAO";
        PreparedStatement stmt = connection.prepareStatement(sql);
        stmt.setLong(1, id_qualis);
        stmt.setString(2, tipo);
        ResultSet rs=stmt.executeQuery();
        while(rs.next()){
            return (int)
                rs.getLong("ID_QUALIS_PUBLICACAO");
        }
        stmt.close();
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
    return 0;
}
```

Figura 34 – Exemplo de DAO para busca do qualis

```

public String execute() throws Exception {

    String zipFile =
"C:/Users/Leo/workspace/Struts2TCCWeb/WebRoot/outputs.zip";
    String[] srcFiles = {
"C:/Users/Leo/workspace/Struts2TCCWeb/WebRoot/Orientações.txt",
"C:/Users/Leo/workspace/Struts2TCCWeb/WebRoot/Participações.txt",
"C:/Users/Leo/workspace/Struts2TCCWeb/WebRoot/Produções.txt"};

    try {
        byte[] buffer = new byte[1024];
        FileOutputStream fos = new FileOutputStream(zipFile);
        ZipOutputStream zos = new ZipOutputStream(fos);
        for (int i=0; i < srcFiles.length; i++) {
            File srcFile = new File(srcFiles[i]);
            FileInputStream fis = new FileInputStream(srcFile);
            zos.putNextEntry(new ZipEntry(srcFile.getName()));
            int length;
            while ((length = fis.read(buffer)) > 0) {
                zos.write(buffer, 0, length);
            }
            zos.closeEntry();
            fis.close();
        }
        zos.close();
        fileInputStream=new FileInputStream(new
File("C:/Users/Leo/workspace/Struts2TCCWeb/WebRoot/outputs.zip"));
    }catch (IOException ioe) {
        System.out.println("Error creating zip file: " + ioe);
    }

    return SUCCESS;
}

```

Figura 35 – Exemplo de Controller disponibilizando arquivos de saída para download


```

<div class="col-sm-9 col-sm-offset-3 col-md-10 col-md-offset-2 main">
  <h1 class="page-header">Cálculo efetuado com sucesso.</h1>
  <div class="table-responsive">
    <table class="table table-striped">
      <thead>
        <tr>
          <th>Publicações</th>
          <th>Peso</th>
          <th>2015</th>
          <th>2014</th>
          <th>2013</th>
          <th>2012</th>
          <th>Total</th>
        </tr>
      </thead>
      <tbody>
        <s:set var="totalconsolidado" value="0"></s:set>
        <s:iterator status="i" value="linhas">
          <tr>
            <td><s:property value="publicacoes"/></td>
            <td><s:property value="peso"/></td>
            <td><s:property value="primeiroano"/></td>
            <td><s:property value="segundoano"/></td>
            <td><s:property value="terceiroano"/></td>
            <td><s:property value="quartoano"/></td>
            <td><s:property value="total"/></td>
            <s:set var="totalconsolidado"
value="%{#totalconsolidado+total}"> </s:set>
          </tr>
        </s:iterator>
        <tr>
          <td><h3><strong>Total Consolidado</strong></h3></td>
          <td><strong><h3><s:label name="totalconsolidado"
value="%{#totalconsolidado}"></s:label></strong></h3></td>
        </tr>
      </tbody>
    </table>
  </div>
  <form class="form-inline" action="retornaInicio" method="post" >
    <div class="form-group ">
      <button class="btn btn-lg btn-success btn-
block"type="submit">Voltar ao painel Inicial</button>
    </div>
  </form>
</div>

```

Figura 36 – Exemplo de trecho de view responsável pelo retorno do calculo