



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO  
DEPARTAMENTO DE EXATAS E NATURAIS  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ISRAEL DOUGLAS DE SOUSA LIMA

**SISTEMA DE POSICIONAMENTO DE AFINIDADE DO BACHARELADO EM  
CIÊNCIA E TECNOLOGIA – CAMPUS MOSSORÓ**

MOSSORÓ-RN  
2014

ISRAEL DOUGLAS DE SOUSA LIMA

**SISTEMA DE POSICIONAMENTO DE AFINIDADE DO BACHARELADO EM  
CIÊNCIA E TECNOLOGIA – CAMPUS MOSSORÓ**

Monografia apresentada à Universidade Federal Rural do Semi-Árido – UFERSA, Departamento de Ciências Exatas e Naturais para a obtenção do título de Bacharel em Ciências da Computação.

Orientadora: Prof. M. Sc. Flávia Estélio Silva Coelho – UFERSA.

Co-orientador: Prof. M. Sc. Lázaro Luis de Lima Sousa – UFERSA.

MOSSORÓ-RN

2014

**O conteúdo desta obra é de inteira responsabilidade de seus autores**

**Dados Internacionais de Catalogação na Publicação (CIP)**

**Biblioteca Central Orlando Teixeira (BCOT)**

**Setor de Informação e Referência**

L732s Lima, Israel Douglas de Sousa.

Sistema de posicionamento de afinidades do bacharelado em ciência e tecnologia – Campus Mossoró / Israel Douglas de Sousa Lima -- Mossoró, 2014.

67f.: il.

Orientadora: Profª. M. Sc. Flávia Estélia Silva Coelho.

Co-orientador: Lazaro Luis de Lima Sousa

Monografia (Graduação em Ciência da Computação) – Universidade Federal Rural do Semi-Árido. Pró-Reitoria de Graduação.

1. Bacharelado em ciência e tecnologia.
  2. Índice de medida de afinidade.
  3. Ranking.
  4. SIPAF.
  5. Seleção.
- I. Título.

RN/UFERSA/BCOT/671-14

CDD: 378

Bibliotecária: Vanessa Christiane Alves de Souza Borba

CRB-15/452

ISRAEL DOUGLAS DE SOUSA LIMA

**SISTEMA DE POSICIONAMENTO DE AFINIDADE DO BACHARELADO EM  
CIÊNCIA E TECNOLOGIA– CAMPUSMOSSORÓ**

Monografia apresentada à Universidade Federal  
Rural do Semi-Árido – UFRSA, Departamento  
de Ciências Exatas e Naturais para a obtenção do  
título de Bacharel em Ciências da Computação.

APROVADA EM: 04/08/2014

**BANCA EXAMINADORA**

\_\_\_\_\_  
Prof. MSc. Flávia Estévia Silva Coelho – UFRSA  
Presidente

\_\_\_\_\_  
Prof. MSc. Lázaro Luis de Lima Sousa – UFRSA  
Co-orientador

\_\_\_\_\_  
Prof. DSc. Angélica Félix de Castro – UFRSA  
Primeiro Membro

\_\_\_\_\_  
Prof. DSc. Daniellé Simone da Silva Casillo – UFRSA  
Segundo Membro

## **AGRADECIMENTOS**

A Deus por todas as bênçãos que me proporcionou na vida, por ter me colocado em uma família maravilhosa, por aproximar pessoas boas que levarei para sempre, por me apontar o caminho quando estava perdido e por todas as conquistas de minha vida.

Aos meus pais José Washington de Lima e Hulga Francisca de Sousa Vidal que sempre acreditaram nas minhas ideias, que me apoiaram nos momentos mais difíceis nessa jornada, que sempre se esforçaram ao máximo para garantir tudo do bom e do melhor, que me proporcionaram conforto e alegrias, e que nunca, mas nunca me deixaram se sentir sozinho.

A minha família, que sempre me acolheu e por toda a ajuda que necessitei em todo o período que passei longe de casa.

Aos grandes amigos Rosenildo Lopes, Alisson Maurício, Hugo Wendel, Klênio Pinheiro e Antonio Queiroz do curso que me motivaram quando pensei em desistir e me ajudaram em todas as atividades disciplinares durante esses cinco anos.

A Fernanda Sachia Chaves Maia, por mais complicado tenha sido nossa história, ela foi uma pessoa fundamental nesse trabalho e é quem me trouxe felicidades nesse período. Motivou-me a continuar sempre de cabeça erguida em todos os momentos de desesperos e estresse durante o desenvolvimento do trabalho.

A minha amiga Bárbara Maria Nogueira Maciel que por mais distante que estivesse fisicamente, durante anos sempre foi uma pessoa muito próxima, amiga e leal.

Às minhas amigas Jarina Maia, Nara Santos e Elka Paulino que me ajudaram muito a enfrentar as dificuldades de morar longe da família estando presente quando mais precisei.

Aos amigos da secretaria do curso Bacharelado em Ciência e Tecnologia: Luciana Angélica, Lázaro Luis, Vanessa Elionara e Barbara Paula que me proporcionaram grandes alegrias e conforto durante todo esse tempo de serviços.

A minha orientadora Flávia Estélia por toda a dedicação, ensino e paciência em desenvolver esse trabalho. Sem a cautela e a organização dela, esse trabalho não teria sido finalizado.

Ao meu co-orientador Lázaro Luis que sempre me ajudou no desenvolvimento do trabalho em qualquer hora que precisei.

Às professoras Angélica Félix e Danielle Casillo por aceitaram o convite em participar da banca avaliadora.

Aos professores do Bacharelado em Ciência da Computação que me proporcionaram grande conhecimento e uma ótima formação profissional.

## LISTA DE FIGURAS

|                                                                                               |    |
|-----------------------------------------------------------------------------------------------|----|
| Figura 1: Formulário de acesso administrativo .....                                           | 16 |
| Figura 2: Área de trabalho do Administrador.....                                              | 17 |
| Figura 3: Tela de inserção do arquivo CSV .....                                               | 17 |
| Figura 4: Exemplo de arquivo CSV .....                                                        | 18 |
| Figura 5: Tela de atualização das disciplinas equivalentes .....                              | 19 |
| Figura 6: Tela de menu para cálculo do IMA por engenharia.....                                | 20 |
| Figura 7: Tela de visualização do <i>ranking</i> por engenharia.....                          | 21 |
| Figura 8: Tabela do <i>Ranking</i> .....                                                      | 21 |
| Figura 9: Tela de cadastro de disciplinas.....                                                | 22 |
| Figura 10: Tela de exclusão de dados do sistema.....                                          | 23 |
| Figura 11: Tela inicial do aluno.....                                                         | 23 |
| Figura 12: Formulário de cadastro do aluno.....                                               | 24 |
| Figura 13: Área do Aluno.....                                                                 | 24 |
| Figura 14: Diagrama de caso de uso.....                                                       | 26 |
| Figura 15: Diagrama de classes .....                                                          | 27 |
| Figura 16: Diagrama de atividade .....                                                        | 30 |
| Figura 17: Diagrama de sequência .....                                                        | 31 |
| Figura 18: Diagrama de Comunicação .....                                                      | 32 |
| Figura 19: Modelo Relacional do Banco de Dados .....                                          | 33 |
| Figura 20: Arquitetura MVC .....                                                              | 34 |
| Figura 21: Namespace do PrimeFaces .....                                                      | 36 |
| Figura 22: Formulário de cadastro desenvolvido com <i>PrimeFaces</i> .....                    | 37 |
| Figura 23: Código do formulário com <i>PrimeFaces</i> .....                                   | 38 |
| Figura 24: Divisão da página em cabeçalho e centro com <i>Primefaces</i> .....                | 39 |
| Figura 25: Resultado do código da Figura 23 .....                                             | 39 |
| Figura 26: Popularidade do <i>PrimeFaces</i> versus concorrentes.....                         | 40 |
| Figura 27: Estrutura do Maven.....                                                            | 41 |
| Figura 28: Arquitetura do POM .....                                                           | 42 |
| Figura 29: Acessos a Repositórios <i>Maven</i> .....                                          | 43 |
| Figura 30: Ciclo de vida e fases do <i>Maven</i> .....                                        | 44 |
| Figura 31: Exemplo de coordenada de artefatos .....                                           | 45 |
| Figura 32: Ciclo de vida do JSF .....                                                         | 46 |
| Figura 33: Arquitetura do JSF .....                                                           | 48 |
| Figura 34: Estrutura de tags do JSF .....                                                     | 49 |
| Figura 35: Visualização de sistema sem modelo ORM (acima) e com modelo ORM (embaixo)<br>..... | 51 |

## LISTA DE SIGLAS

AJAX - *Asynchronous Javascript and XML*  
API - *Application Programming Interface*  
BC&T - Bacharelado em Ciências e Tecnologia  
CHE - Carga Horária Eletiva  
CHO - Carga Horária Obrigatória  
CRUD - *Create, Read, Update, Delete*  
CSV - *Comma Separated Values*  
DCEN - Departamento de Ciências Exatas e Naturais  
DDL - *Data Definition Language*  
DML - *Data Manipulation Language*  
DQL - *Data Query Language*  
EJB - *Enterprise Java Beans*  
HQL - *Hibernate Query Language*  
HTTP - *Hypertext Transfer Protocol*  
IMA - Índice de Medida de Afinidade  
JavaEE - *Java Platform, Enterprise Edition*  
JPA - *Java Persistence API*  
JPC - *Java Community Process*  
JPQL - *Java Persistence Query Language*  
JSF - *Java Server Faces*  
JSTL - *Java Server Pages Standard Template Library*  
MA- Média Aprovação  
MR - Média Aritmética Notas Reprovadas  
MVC - Modelo – Visão – Controle  
OO - Orientado a Objetos  
ORM - *Object Relation Mapping*  
POJOS - *Plain Old Java Objects*  
POM - *Project Object Model*  
SGBD - Sistema Gerenciador de Banco de Dados  
SIPAF - Sistema de Posicionamento de Afinidade  
SQL - *Structured Query Language*  
SUTIC - Superintendência de Tecnologia da Informação e Comunicação

UFERSA - Universidade Federal Rural do Semi-Árido

UI - *UserInterface*

W3C - *WorldWideWebConsortium*

XHTML - *eXtensibleHypertextMarkupLanguage*

XML - *eXtensibleMarkupLanguage*

## SUMÁRIO

|             |                                       |           |
|-------------|---------------------------------------|-----------|
| <b>1.1</b>  | <b>Problema.....</b>                  | <b>11</b> |
| 1.1.1       | Equação do IMA.....                   | 11        |
| 1.1.2       | Processos de Seleção .....            | 12        |
| <b>1.2</b>  | <b>Objetivos.....</b>                 | <b>14</b> |
| 1.2.1       | Geral .....                           | 14        |
| 1.2.2       | Específicos.....                      | 14        |
| <b>1.3</b>  | <b>Justificativa.....</b>             | <b>15</b> |
| <b>2.</b>   | <b>Desenvolvimento .....</b>          | <b>16</b> |
| <b>2.1.</b> | <b>Desenvolvimento do SIPAF .....</b> | <b>16</b> |
| <b>2.2</b>  | <b>Arquitetura do SIPAF.....</b>      | <b>25</b> |
| 2.2.1       | Diagrama de Caso de Uso .....         | 25        |
| <b>2.3</b>  | <b>Referencial Teórico.....</b>       | <b>34</b> |
| 2.3.1       | Modelo – Visão – Controle .....       | 34        |
| 2.3.2       | Ferramentas de Desenvolvimento .....  | 35        |
| 2.3.2.1     | Primefaces .....                      | 36        |
| 2.3.2.2     | Maven .....                           | 40        |
| 2.3.2.3     | Java Server Faces.....                | 45        |
| 2.3.2.4     | Hibernate .....                       | 50        |
| <b>3.</b>   | <b>Conclusões .....</b>               | <b>54</b> |
| <b>4.</b>   | <b>Trabalhos Futuros .....</b>        | <b>55</b> |
|             | <b>Referências .....</b>              | <b>56</b> |
|             | <b>ANEXO A.....</b>                   | <b>59</b> |
|             | <b>ANEXO A.1.....</b>                 | <b>65</b> |
|             | <b>ANEXO A.2.....</b>                 | <b>66</b> |

## 1. Introdução

O Bacharelado em Ciências e Tecnologia (BC&T) surgiu a partir de um modelo que visava transformar a maneira de graduar estudantes nos cursos de engenharia e licenciatura. Esse modelo dividiu o curso em dois ciclos. No primeiro ciclo, o aluno aprende os conceitos de ciências exatas, humanas e ambientais e, no segundo ciclo, os conceitos específicos das engenharias. O curso BC&T da Universidade Federal Rural do Semi-Árido (UFERSA), criado em 2008, é um curso do primeiro ciclo que prepara os alunos para as áreas de engenharias do segundo ciclo desta instituição.

Desde a primeira turma de BC&T formada na UFERSA, as solicitações por uma vaga no curso de segundo ciclo, a engenharia, começaram a exceder o número de vagas, o que acaba conduzindo ao processo seletivo os alunos.

As eletivas de BC&T são disciplinas optativas que direcionam o aluno para as engenharias. Estas começaram a passar pelo problema de excesso de solicitações de matrículas, o que acarretou em um processo de seleção por uma vaga. Para realizar a seleção dos alunos para os cursos de engenharias e para as disciplinas eletivas, o curso de BC&T adotou um índice baseado no desempenho e na afinidade do aluno para as engenharias específicas, obtido através da equação Índice de Medida de Afinidade (IMA).

Sendo assim, a proposta desse trabalho é desenvolver o Sistema de Posicionamento de Afinidade (SIPAF) para efetuar o cálculo do IMA, de modo automatizado, para os alunos que solicitarem vagas nas disciplinas eletivas e engenharias, e retornar uma tabela classificatória, em ordem decrescente, em função dos valores calculados do IMA.

O trabalho foi organizado da seguinte maneira. No capítulo 1, é apresentada uma introdução sobre a importância do assunto, a descrição do problema, a definição dos objetivos geral e específicos e uma justificativa para a sua realização. No capítulo 2, será especificado o modelo de programação utilizado para organizar as classes e entidades que compõem o sistema, as ferramentas, as tecnologias de desenvolvimento e gerenciamento do ambiente de trabalho, a arquitetura desenvolvida para moldar os requisitos levantados para o SIPAF, de acordo com os conceitos de Engenharia de Software, e as páginas de interação entre o sistema e o usuário. O capítulo 3 apresenta as considerações finais, debatendo sobre os testes e resultados obtidos. No capítulo 4, sugerem-se implementações futuras para o sistema.

## 1.1 Problema

Diante da necessidade de selecionar alunos do primeiro ciclo de BC&T para os cursos de engenharia do segundo ciclo, com base nas etapas estabelecidas em edital próprio (Anexo A), surgiu uma equação matemática, denominada IMA.

Quanto às disciplinas eletivas, conseguir uma vaga se tornou um desafio, pois o número de solicitações para algumas disciplinas aumentaram além do número de vagas ofertadas pelo curso. Portanto, se fez necessário estabelecer um processo de seleção para os alunos que ainda estão cursando BC&T e desejam cursar uma disciplina eletiva utilizando o IMA.

Devido ao grande número de alunos no BC&T, manipular informações para realizar o cálculo do IMA se torna um processo exaustivo e demorado (em média um arquivo, com dados dos alunos para as seleções, possui 112 (cento e doze) mil linhas em uma planilha com 14 (quatorze) informações em cada linha), o que gera mais um problema, pois a UFERSA segue, rigorosamente, um calendário acadêmico.

Calcular o IMA e gerar o *ranking* dos alunos de BC&T seguindo os editais de seleção das engenharias e de disciplinas eletivas, no período premeditado no calendário acadêmico da instituição, é um problema encontrado pela coordenação do curso de BC&T.

Para entender o esquema lógico da equação do IMA e a estrutura dos processos de seleções, as respectivas explanações são apresentadas a seguir.

### 1.1.1 Equação do IMA

O IMA é um valor resultante do desempenho que o aluno obteve no decorrer do curso para uma determinada engenharia, a partir dos resultados das  $i$ -ésimas disciplinas cursadas em BC&T, conforme a equação 1 :

$$IMA = \frac{0.7}{0.8} \left( \frac{\sum P_i * A_i * C_i}{\sum A_i * C_i} \right) + 0,3 \left( \frac{CHO}{183} + \frac{CHE}{48} \right) \quad (1)$$

A variável  $P_i$  é a nota obtida pela aprovação na  $i$ -ésima disciplina. Se o aluno cursou pela primeira vez uma disciplina e foi aprovado, o valor de  $P_i$  é a média final obtida na aprovação, se o aluno foi aprovado na disciplina tendo cursado ela mais de uma vez, o valor

de  $P_i$  é determinado pela média de aprovação (MA) e a média aritmética das notas (MR) quando o aluno não obteve êxito na disciplina, de acordo com a equação 2.

$$P_i = \frac{8*MA+2*MR}{10} \quad (2)$$

A variável  $A_i$  é a nota relacionada ao nível de afinidade da disciplina com relação à engenharia para qual será calculada o IMA. Tal valor é obtido a partir da equação 3.

$$A_i = \frac{10+3*NA}{10} \quad (3)$$

O valor NA é o índice de afinidade que poderá ser igual a 1 (um) para disciplinas obrigatórias de BC&T, igual a 2 (dois) para disciplinas eletivas que não fazem parte da lista de disciplinas da engenharia desejada para o cálculo do IMA, e igual a 3 (três) se a disciplina eletiva compõe a lista de disciplinas da engenharia escolhida para o cálculo.

A variável  $C_i$  se refere ao valor calculado, a partir do número de crédito (NC) que a disciplina possui, segundo a equação 4.

$$C_i = \frac{10+3*NC}{10} \quad (4)$$

A carga horária total das disciplinas obrigatórias (CHO) e a carga horária das disciplinas eletivas (CHE) dos alunos no curso de BC&T também definem o valor do IMA.

### 1.1.2 Processos de Seleção

O BC&T é um curso que viabiliza a inserção de um ensino interdisciplinar na educação brasileira. Usualmente, o curso de graduação para a formação de engenheiros é formado por dois ciclos: o primeiro ciclo é o curso de BC&T, no qual o aluno estuda os conceitos da ciência exata e humana e, o segundo ciclo são as engenharias, em que o aluno estuda os assuntos específicos de cada área. Até o final do primeiro ciclo, o aluno passa por

processos de seleção para garantir uma vaga nas disciplinas eletivas e para as engenharias do segundo ciclo. Esses processos são necessários para a organização justa dos alunos devido à grande concorrência em uma determinada área do curso, a qual possui um limite de vagas por período.

Os processos de seleção para as engenharias são realizados seguindo quatro etapas descritas em edital específico disponibilizado pela coordenação do curso de BC&T. Essa seleção ocorre devido à necessidade de preencher as vagas dos cursos para as engenharias, nas quais o número de inscritos é maior que o número de vagas disponíveis. Atualmente, as engenharias disponíveis na instituição para o campus Mossoró são: Engenharia Civil, Engenharia de Energia, Engenharia Mecânica, Engenharia Química, Engenharia de Petróleo e Engenharia de Produção.

Na primeira etapa do processo, é efetuado um *ranking* de todos os alunos que se inscreveram de acordo com as regras estabelecidas no edital. O *ranking* é uma listagem, em ordem decrescente, dos alunos do Bacharelado em Ciência e Tecnologia, baseando-se no desempenho obtido durante as 2400 horas mínimas necessárias para a conclusão do curso, calculado com base na equação do IMA.

Na segunda etapa, é aplicada uma prova objetiva aos alunos que não conquistaram uma vaga por meio do *ranking*. A prova é composta por assuntos estudados nas disciplinas do curso de BC&T e possui um total de 35 (trinta e cinco) questões.

Na terceira etapa, os alunos que não ingressaram na engenharia solicitada por meio do *ranking* ou via prova objetiva, poderão optar por se inscreverem em alguma das outras engenharias que, por sua vez, não obtiveram o número máximo de alunos por turma.

O aluno formado em BC&T possui um amplo conhecimento na área de ciências exatas, o que lhe proporciona um significativo domínio quanto aos princípios fundamentais das engenharias. Sendo assim, se o aluno optar em cursar qualquer outra engenharia que esteja vinculada ao curso de BC&T, o mesmo estará apto a realizar as atividades curriculares do curso.

Na quarta etapa, as vagas remanescentes dos cursos de engenharia serão ofertadas aos alunos que concluíram o curso de BC&T em períodos anteriores e para os alunos de outros campi da UFERSA. O objetivo é preencher todas as turmas por período, dando prioridade aos alunos que estão concorrendo às vagas pela primeira vez.

Um dos requisitos para a formação do aluno no curso de BC&T é ter, em seu histórico disciplinar, 420 horas de disciplinas integralizadas.

O processo de deferimento para as disciplinas eletivas do curso de BC&T também

segue os critérios estabelecidos em edital e consiste em três etapas. Na primeira etapa, o aluno solicita as disciplinas, por meio de um formulário *online* (UFERSA, 2014) que disponibiliza as eletivas do curso, juntamente com seus horários e pré-requisitos. Ao fazer essa solicitação, o aluno estará de acordo com as condições dispostas no edital e em consenso com o processo de seleção.

A segunda etapa, que ocorre após o encerramento do período de solicitação das eletivas, via formulário *online*, realiza o *ranking* a partir do cálculo do IMA para os alunos que possuem carga horária mínima integralizada de 1230 horas no curso de BC&T. As disciplinas eletivas garantem um peso de afinidade a uma ou mais engenharias. O IMA de cada aluno é calculado devido ao excesso de solicitações enviadas via formulário para uma determinada disciplina eletiva. As vagas são preenchidas de acordo com as notas, em ordem decrescente, do resultado obtido. Da mesma forma, a utilização do SIPAF é indispensável nessa etapa, pois devido à quantidade de solicitações por disciplina eletiva, o número a ser encontrado alcança, em média, a 2600 resultados segundo os dados da coordenação do curso.

Na terceira etapa, é verificado se o aluno está possibilitado de cursar a disciplina eletiva, baseando-se nas especificações estabelecidas no edital. Os alunos que não possuem cargas horárias suficientes ou não possuem os possíveis pré-requisitos de uma disciplina, terão suas solicitações indeferidas mesmo que garantam uma posição entre os mais bem colocados no *ranking*.

## 1.2 Objetivos

### 1.2.1 Geral

Desenvolver o SIPAF para calcular o valor do IMA e estruturar uma tabela classificatória dos alunos do curso de Bacharelado em Ciências e Tecnologia da UFERSA – Campus Mossoró.

### 1.2.2 Específicos

- Calcular o IMA dos alunos do curso de BC&T;

- Realizar a primeira e a terceira etapa do processo classificatório dos alunos para os cursos de engenharia e a segunda etapa do processo classificatória das disciplinas eletivas;
- Otimizar o tempo de processamento dos resultados;
- Estudar novas tecnologias aplicáveis ao desenvolvimento de aplicações Web.

### 1.3 Justificativa

Empregar sistemas computacionais em instituições, empresas e organizações vem sendo uma estratégia muito utilizada no mercado para estruturar e solucionar problemas de gerenciamento logísticos e matemáticos. O SIPAF é um *software* desenvolvido para realizar o processo matemático envolvido no IMA originado para a seleção dos alunos de BC&T, gerenciando todos os dados avaliativos do aluno no decorrer de seu desenvolvimento no curso.

Trabalhar com manipulação de dados é uma tarefa que acarreta certa dificuldade pois qualquer erro, por mínimo que seja, pode mudar todo o contexto do serviço. O IMA utiliza uma grade de pesos e o desempenho do aluno (aprovações, reprovações e carga horária de cada disciplina) em sua equação. Essas informações tornam o seu cálculo uma tarefa desgastante e duradoura diante de todos dados, assim chegando a favorecer falhas humanas, o que acabaria prejudicando um ou mais estudantes na seleção.

O SIPAF é desenvolvido em Java, uma linguagem de programação robusta que garante a otimização do processo matemático, com o auxílio de outras ferramentas específicas para cada particularidade do IMA, resultando assim, em uma resposta totalmente confiável.

## 2. Desenvolvimento

Nesse capítulo é descrito o desenvolvimento de telas do sistema, os diagramas utilizados para formalizar todos os processos realizados no SIPAF e a definição das ferramentas e tecnologias estudadas para a implementação.

### 2.1. Desenvolvimento do SIPAF

O SIPAF disponibiliza dois tipos de ambiente, um destinado ao administrador e o outro para o acesso dos alunos. Na página inicial, como mostra a figura 1, há duas abas que separam o local de acesso aos ambientes. Para ter acesso à área do administrador, será necessário o uso de um *login* e uma senha, previamente cadastrados, pois essa área é de uso restrito à Secretaria do curso BC&T.

Figura 1: Formulário de acesso administrativo

Aluno Administrador

Login:

Senha:

Enviar

[Primeiro acesso? Clique aqui](#)

Secretaria de BCT de Mossoró, Av. Francisco Mota, 572, Bairro Costa e Silva. Telefone: (84) 3317 8253. Email: secretariabct@ufersa.edu.br

Fonte: Autoria própria.

Após a comprovação dos dados informados via tela de *login*, o sistema é redirecionado para uma página contendo um menu de eventos, representado na Figura 2. No menu, estão disponíveis os eventos necessários para a execução do cálculo do IMA.

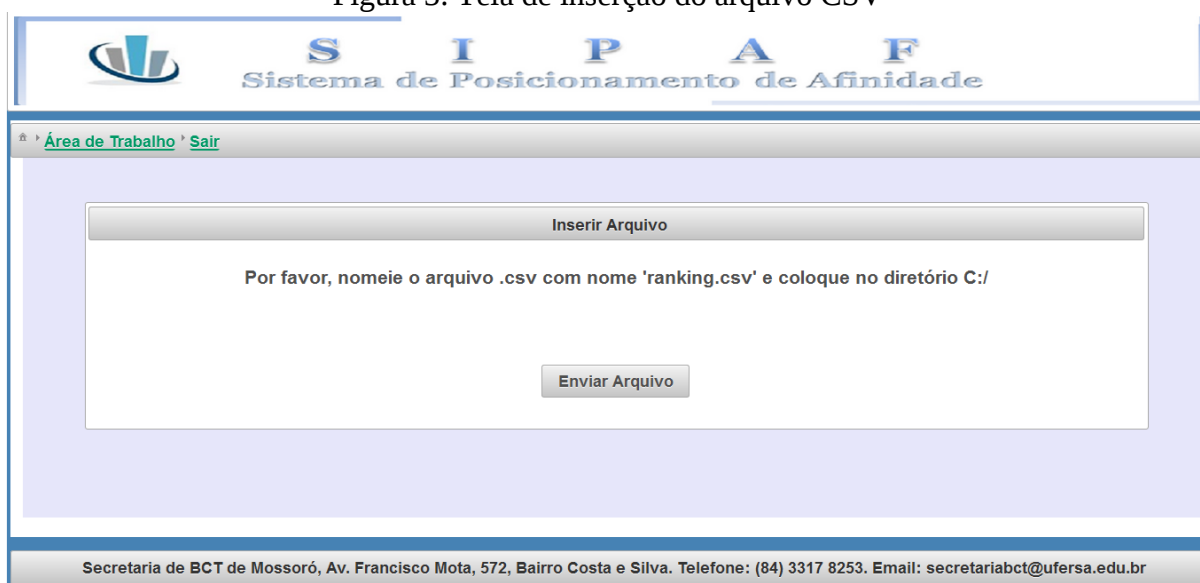
Figura 2: Área de trabalho do Administrador



Fonte: Autoria própria.

Como o IMA é uma equação matemática baseada nos dados de desempenho dos alunos, o SIPAF precisa ter acesso a esses dados. Portanto, existe uma página destinada à ação, acessível através do botão Inserir Arquivo. Na figura 3, visualiza-se a página onde informa como deve ser feito para acionar de forma correta a função de inserir o arquivo, com extensão *CommaSeparatedValues* (CSV), informando os dados do aluno, necessários ao cálculo do IMA.

Figura 3: Tela de inserção do arquivo CSV



Fonte: Autoria própria.

O documento com extensão CSV é um tipo de arquivo-texto utilizado para exportar dados de planilha eletrônica, separados por meio de vírgula, ponto-e-vírgula ou uma barra. Esse arquivo é gerado e enviado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC), uma unidade administrativa responsável por planejar, implantar e manter as atividades relacionadas à Tecnologia da Informação e Comunicação da UFERSA (SUTIC, 2009). A Figura 4, exibimos um exemplo de como é esse arquivo (o nome do aluno e a matrícula foram alterados para preservar a identificação do aluno).

Figura 4: Exemplo de arquivo CSV

|    | A                   | B      | C            | D            | E       | F          | G           | H       | I                                                | J       | K       | L          | M    | N       |
|----|---------------------|--------|--------------|--------------|---------|------------|-------------|---------|--------------------------------------------------|---------|---------|------------|------|---------|
|    | curso               | cidade | ano_ingresso | periodo_ingr | nome    | matricula  | media_final | codigo  | nome                                             | ch_aula | cr_aula | descricao  | ano  | periodo |
| 1  | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 1 | 2012000001 | 6.00        | AMB0076 | AMBIENTE ENERGIA E SOCIEDADE (1200340)           | 60      | 4       | APROVADO   | 2012 | 1       |
| 2  | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 1 | 2012000001 | 9.00        | ACS0050 | ANALISE E EXPRESSAO TEXTUAL (1200536)            | 60      | 4       | APROVADO   | 2012 | 1       |
| 3  | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 1 | 2012000001 | 8.40        | EXA0101 | CALCULO I (1200003)                              | 60      | 4       | APROVADO   | 2013 | 2       |
| 4  | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 1 | 2012000001 | 3.80        | EXA0101 | CALCULO I (1200003)                              | 60      | 4       | REPROVADO  | 2012 | 2       |
| 5  | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 1 | 2012000001 | 3.10        | EXA0101 | CALCULO I (1200003)                              | 60      | 4       | REPROVADO  | 2012 | 1       |
| 6  | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 1 | 2012000001 | 4.00        | EXA0101 | CALCULO I (1200003)                              | 60      | 4       | REPROVADO  | 2013 | 1       |
| 7  | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 1 | 2012000001 | 5.20        | ACS0701 | ECONOMIA PARA ENGENHARIA (1200374)               | 60      | 4       | APROVADO   | 2013 | 2       |
| 8  | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 1 | 2012000001 |             | AMB0099 | EXPRESSAO GRAFICA (1200557)                      | 60      | 4       | INDEFERIDO | 2013 | 1       |
| 9  | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 1 | 2012000001 | 8.40        | AMB0099 | EXPRESSAO GRAFICA (1200557)                      | 60      | 4       | APROVADO   | 2013 | 1       |
| 10 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 1 | 2012000001 | 8.30        | ACS0012 | FILOSOFIA DA CIENCIA E MET. CIENTIFICA (1200114) | 60      | 4       | APROVADO   | 2012 | 2       |
| 11 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 1 | 2012000001 | 4.10        | EXA0114 | GEOMETRIA ANALITICA (1200255)                    | 60      | 4       | REPROVADO  | 2013 | 1       |
| 12 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 1 | 2012000001 | 2.90        | EXA0114 | GEOMETRIA ANALITICA (1200255)                    | 60      | 4       | REPROVADO  | 2012 | 2       |
| 13 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 1 | 2012000001 | 2.50        | EXA0114 | GEOMETRIA ANALITICA (1200255)                    | 60      | 4       | REPROVADO  | 2013 | 2       |
| 14 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 2 | 2012000002 | 2.60        | AMB0076 | AMBIENTE ENERGIA E SOCIEDADE (1200340)           | 60      | 4       | APROVADO   | 2012 | 1       |
| 15 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 2 | 2012000002 | 6.10        | ACS0050 | ANALISE E EXPRESSAO TEXTUAL (1200536)            | 60      | 4       | APROVADO   | 2012 | 1       |
| 16 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 2 | 2012000002 | 7.10        | EXA0101 | CALCULO I (1200003)                              | 30      | 2       | APROVADO   | 2013 | 1       |
| 17 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 2 | 2012000002 | 7.80        | AMB0099 | EXPRESSAO GRAFICA (1200557)                      | 30      | 2       | APROVADO   | 2013 | 1       |
| 18 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 2 | 2012000002 | 5.40        | ACS0012 | FILOSOFIA DA CIENCIA E MET. CIENTIFICA (1200114) | 30      | 2       | APROVADO   | 2013 | 2       |
| 19 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 2 | 2012000002 | 5.00        | EXA0114 | GEOMETRIA ANALITICA (1200255)                    | 30      | 2       | APROVADO   | 2013 | 1       |
| 20 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 2 | 2012000002 | 8.90        | ACS0701 | ECONOMIA PARA ENGENHARIA (1200374)               | 60      | 4       | APROVADO   | 2013 | 2       |
| 21 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 3 | 2012000003 | 8.10        | AMB0076 | AMBIENTE ENERGIA E SOCIEDADE (1200340)           | 60      | 4       | APROVADO   | 2012 | 2       |
| 22 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 3 | 2012000003 | 9.50        | ACS0050 | ANALISE E EXPRESSAO TEXTUAL (1200536)            | 60      | 4       | APROVADO   | 2013 | 1       |
| 23 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 3 | 2012000003 | 7.70        | EXA0101 | CALCULO I (1200003)                              | 60      | 4       | APROVADO   | 2013 | 2       |
| 24 | BACHARELADI MOSSORÓ |        | 2012         | 1            | Aluno 3 | 2012000003 | 8.80        | AMB0099 | EXPRESSAO GRAFICA (1200557)                      | 60      | 4       | APROVADO   | 2013 | 2       |

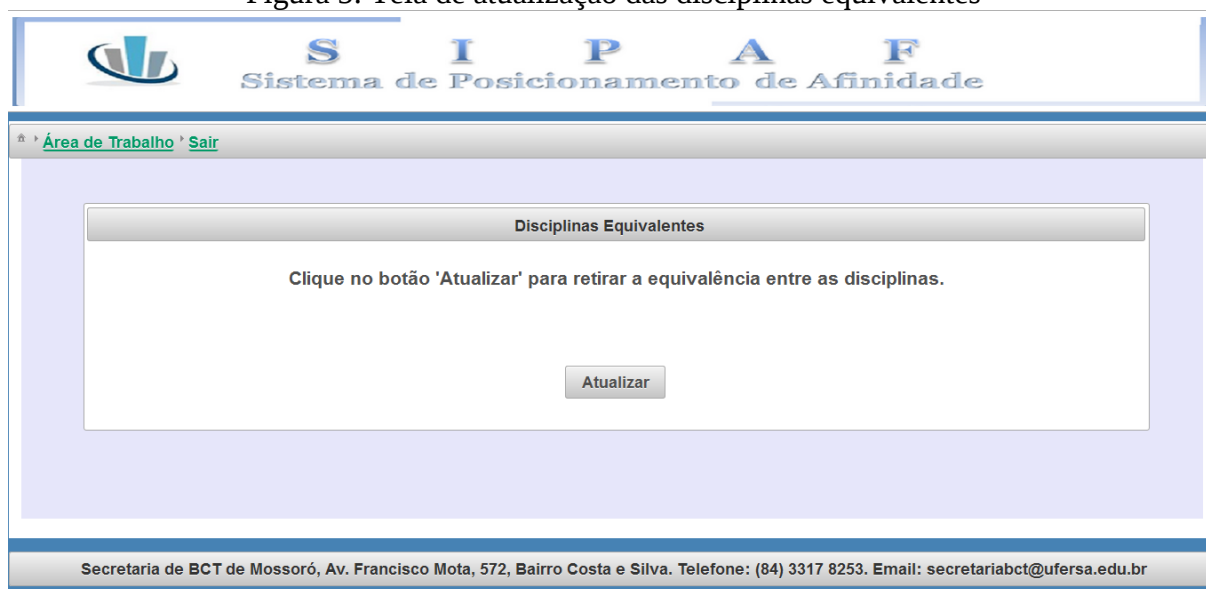
Fonte: Autoria própria.

Na coluna A, tem-se o nome do curso que o aluno está cursando, no caso para todas as linhas, Bacharelado em Ciências e Tecnologia. A coluna B identifica o campus em que o aluno cursou a disciplina. As colunas C e D indicam, respectivamente, o ano e o período que o aluno ingressou na universidade. Os campos E e F informam o nome do aluno e a matrícula na instituição. A coluna G indica a média final do aluno para a respectiva disciplina. O código e o nome da disciplina são identificados nas colunas H e I, nessa ordem. As colunas J e K definem a carga horária e o número de créditos daquela disciplina, respectivamente. A coluna L informa a situação do aluno naquela disciplina, ou seja, indica se o aluno foi aprovado (ou cumpriu, como aparece às vezes), reprovado, reprovado por falta, reprovou por falta e média, desistiu ou trancou a disciplina. Se o aluno não for aprovado na disciplina e decidir tentar cursá-la novamente, uma nova linha no arquivo é criada com as mesmas informações anteriores, apenas diferenciando a nota, a situação, o ano e o período que ele concluiu a disciplina nas colunas M e N, respectivamente.

Uma dificuldade encontrada para realizar o cálculo do IMA é a diversidade de turmas

de uma mesma disciplina, oferecidas pelos departamentos da instituição. Devido à diversidade, uma disciplina pode ter mais de um código identificador. O SIPAF trabalha diretamente com os códigos das disciplinas e uma solução para esse problema, foi a implementação do evento Disciplinas Equivalentes, exibida na figura 5. Nessa fase, o sistema faz uma busca dos códigos equivalentes, para cada disciplina, no banco de dados no qual foram inseridas as informações do arquivo .csv e converte para o código ofertado pelo DCEN (Departamento de Ciências Exatas e Naturais), o departamento responsável pelo curso de BC&T.

Figura 5: Tela de atualização das disciplinas equivalentes



Fonte: Autoria própria

O botão Calcular IMA redireciona para uma página contendo um menu com os botões Engenharia Civil, Engenharia de Energia, Engenharia Mecânica, Engenharia de Petróleo, Engenharia de Produção e Engenharia Química, conforme exibe a figura 6.

Figura 6: Tela de menu para cálculo do IMA por engenharia



Fonte: Autoria própria.

Um componente do IMA é o valor de peso que uma disciplina possui para uma ou mais engenharias definidas. Por esse motivo, o SIPAF calcula o IMA dos alunos separadamente para cada engenharia. Toda disciplina obrigatória no curso de BC&T possui peso 1 (um) na equação do IMA. Disciplinas eletivas, que não fazem parte da grade da engenharia selecionada para o cálculo, possuem peso 2 (dois) e disciplinas eletivas que fazem parte da grade curricular da engenharia selecionada possuem peso 3 (três).

O botão Ver Ranking redireciona a uma página contendo um menu com os mesmos botões visualizados na Figura 6, porém essa página (figura 7) separa, por engenharia, as tabelas de classificação dos alunos.



(figura 9) com um formulário para o cadastro das informações das disciplinas. No formulário, é apresentado um campo para nome, código da disciplina, códigos de disciplinas equivalentes, caso existam, e para qual(is) curso(s) a disciplina tem mais afinidade. Na mesma página, existe uma tabela informando quais disciplinas já foram cadastradas no sistema.

Figura 9: Tela de cadastro de disciplinas

**SIPAF**  
Sistema de Posicionamento de Afinidade

Área de Trabalho Sair

**Cadastro de Disciplinas**

Disciplina:  Código:

Curso 1:  Curso 2:  Curso 3:

Equivalente 1:  Equivalente 2:  Equivalente 3:  Equivalente 4:

Cadastrar

| Código | Disciplina | Curso 1 | Curso 2 | Curso 3 |
|--------|------------|---------|---------|---------|
|        |            |         |         |         |

Remover

Secretaria de BCT de Mossoró, Av. Francisco Mota, 572, Bairro Costa e Silva. Telefone: (84) 3317 8253. Email: secretariabct@ufersa.edu.br

Fonte: Autoria própria.

Com o botão Deletar Dados (figura 2), o administrador é redirecionado a uma página que dará a liberdade de apagar todas as informações das tabelas do *ranking*, as informações importadas do arquivo CSV do banco de dados e os cadastros dos alunos, como mostra figura 10. Essa função oportuniza, ao administrador, realizar o cálculo do IMA, todos os semestres.

Figura 10: Tela de exclusão de dados do sistema

Deletar Dados

Atenção, área destinada a apagar todos os dados registrados no sistema

Deletar dados do arquivo CSV

Deletar dados dos alunos cadastrados

Deletar dados do ranking

Secretaria de BCT de Mossoró, Av. Francisco Mota, 572, Bairro Costa e Silva. Telefone: (84) 3317 8253. Email: secretariabct@ufersa.edu.br

Fonte: Autoria própria.

Nas páginas do ambiente de acesso do administrador existe uma aba de comando que disponibiliza acesso ao menu do administrador e à saída do ambiente.

Como o SIPAF é uma aplicação Web, ao aluno também é fornecido um ambiente de acesso. Para isso ser possível, o aluno deverá preencher um cadastro de usuário com sua matrícula e senha. Esse ambiente está disponível via tela inicial do sistema, de acordo com a figura 11.

Figura 11: Tela inicial do aluno

Aluno Administrador

Matricula:

Senha:

Enviar

[Primeiro acesso? Clique aqui](#)

Secretaria de BCT de Mossoró, Av. Francisco Mota, 572, Bairro Costa e Silva. Telefone: (84) 3317 8253. Email: secretariabct@ufersa.edu.br

Fonte: Autoria própria.

O *link* direciona o aluno a uma página de cadastro (figura 12) do próprio sistema. Preenchendo e confirmando o cadastro, o aluno poderá acessar a sua página particular no SIPAF.

Figura 12: Formulário de cadastro do aluno

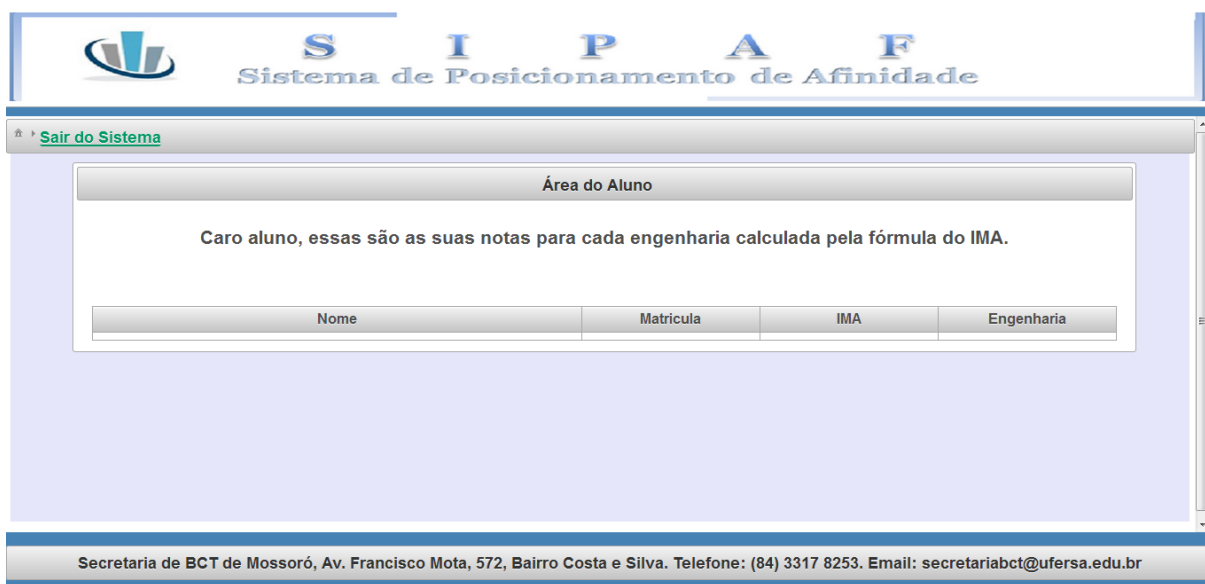


The screenshot shows the SIPAF (Sistema de Posicionamento de Afinidade) user registration page. At the top, there is a logo and the text 'SIPAF Sistema de Posicionamento de Afinidade'. Below this, a navigation bar contains links for 'Área de Trabalho' and 'Sair'. The main content area is titled 'Cadastro de Usuário' and contains a form with the following fields: 'Nome Completo', 'Matricula' (with a value of '0'), 'Email', 'Senha', and 'Telefone'. At the bottom of the form are two buttons: 'Limpar' and 'Enviar'. The footer of the page provides contact information for the Secretaria de BCT de Mossoró, including the address, phone number, and email.

Fonte: Autoria própria.

Após autenticação, o aluno é redirecionado a uma página contendo os resultados (figura 13) do IMA referentes às engenharias. Para sair da conta, basta clicar na aba Sair do Sistema.

Figura 13: Área do Aluno



The screenshot shows the SIPAF student area. At the top, there is a logo and the text 'SIPAF Sistema de Posicionamento de Afinidade'. Below this, a navigation bar contains a link for 'Sair do Sistema'. The main content area is titled 'Área do Aluno' and contains a message: 'Caro aluno, essas são as suas notas para cada engenharia calculada pela fórmula do IMA.' Below the message is a table with four columns: 'Nome', 'Matricula', 'IMA', and 'Engenharia'. The footer of the page provides contact information for the Secretaria de BCT de Mossoró, including the address, phone number, and email.

Fonte: Autoria própria

## 2.2 Arquitetura do SIPAF

Para realizar a implementação do sistema, é necessário moldar a estrutura utilizada para atender aos requisitos levantados, seguindo uma ordem lógica de atividades e de cronograma de execução para cada evento, a fim de prevenir erros e evitar que um requisito seja implementado com dependência de outro que não foi implementado. A seguir, serão definidos os diagramas formulados para conduzir a implementação do SIPAF.

### 2.2.1 Diagrama de Caso de Uso

Analisando o problema, foi montado o diagrama de caso de uso que especifica os requisitos funcionais a serem desenvolvidos para que o SIPAF atinja o objetivo geral proposto neste trabalho. O diagrama ainda especifica os autores (ou usuários) que terão acesso a cada funcionalidade.

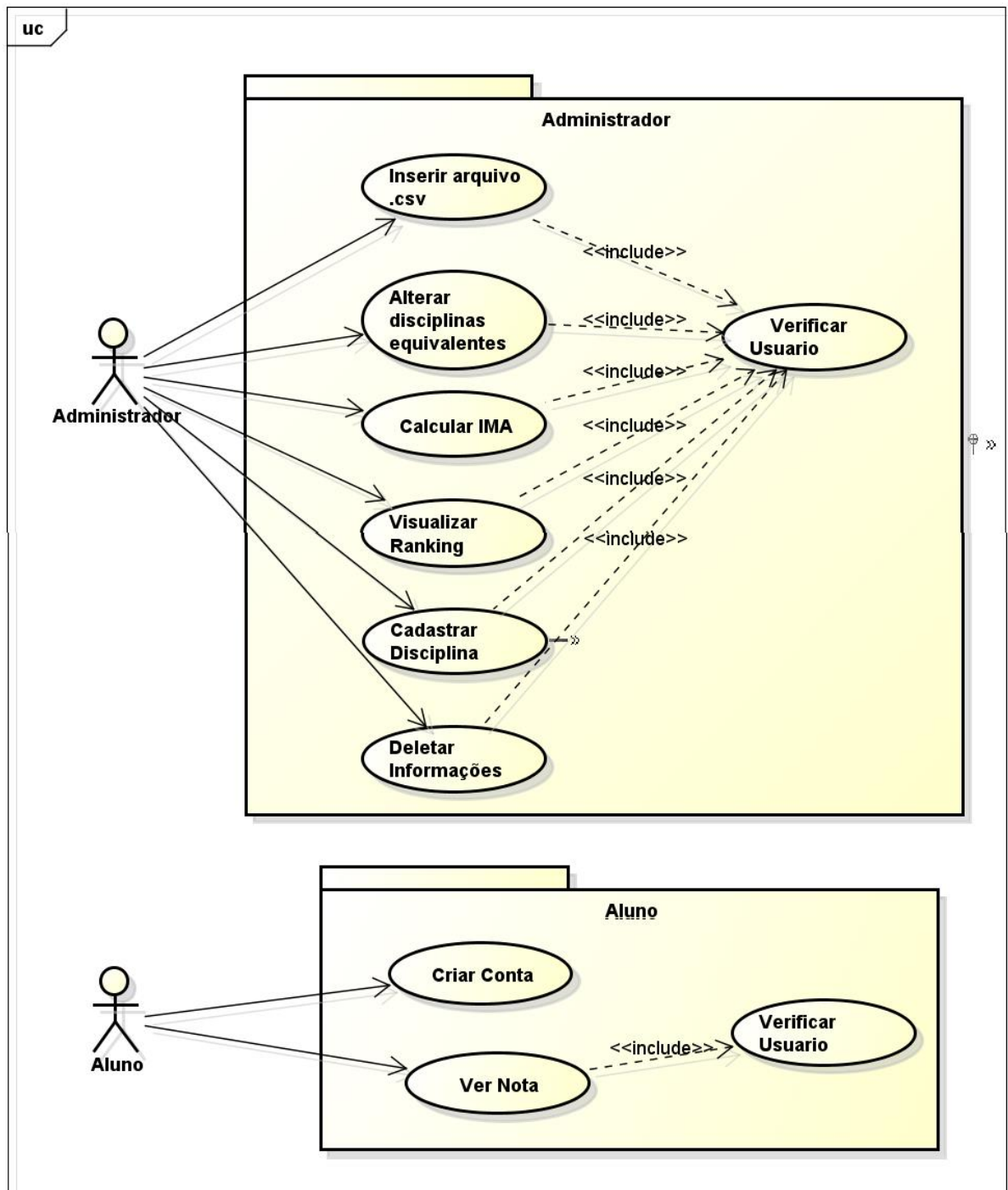
Na figura 14, cada balão representa um requisito funcional levantado para o sistema. O ator administrador tem acesso aos seguintes requisitos no ambiente administrativo:

- Inserir arquivo .csv, contendo as informações dos alunos;
- Alterar disciplinas equivalentes que desencadeia uma busca no banco e realiza uma alteração no código da disciplina, caso ela possua código equivalente predefinido pelo próprio administrador;
- Calcular IMA é o evento que efetua o cálculo, seguindo à equação do IMA;
- Visualizar *Ranking* exibirá as tabelas classificatórias baseadas no IMA dos alunos;
- Cadastrar Disciplinas permite o administrador cadastrar as disciplinas especificando o peso delas de acordo com o curso e os códigos de equivalências;
- Apagar Informações fornece ao administrador o acesso para excluir as informações do sistema.

As linhas tracejadas no balão Verificar Usuário indicam que para qualquer acesso a um dos requisitos, será necessária uma verificação de acesso por meio de *login* e senha. O autor aluno tem permissão para criar uma conta de usuário, onde será necessário informar dados pessoais como nome, matrícula e email, para ter acesso ao sistema e ver sua nota do

IMA, quando calculada. Para visualizar sua nota, o aluno passa por uma verificação de autenticidade do usuário, similar ao administrador.

Figura 14: Diagrama de caso de uso

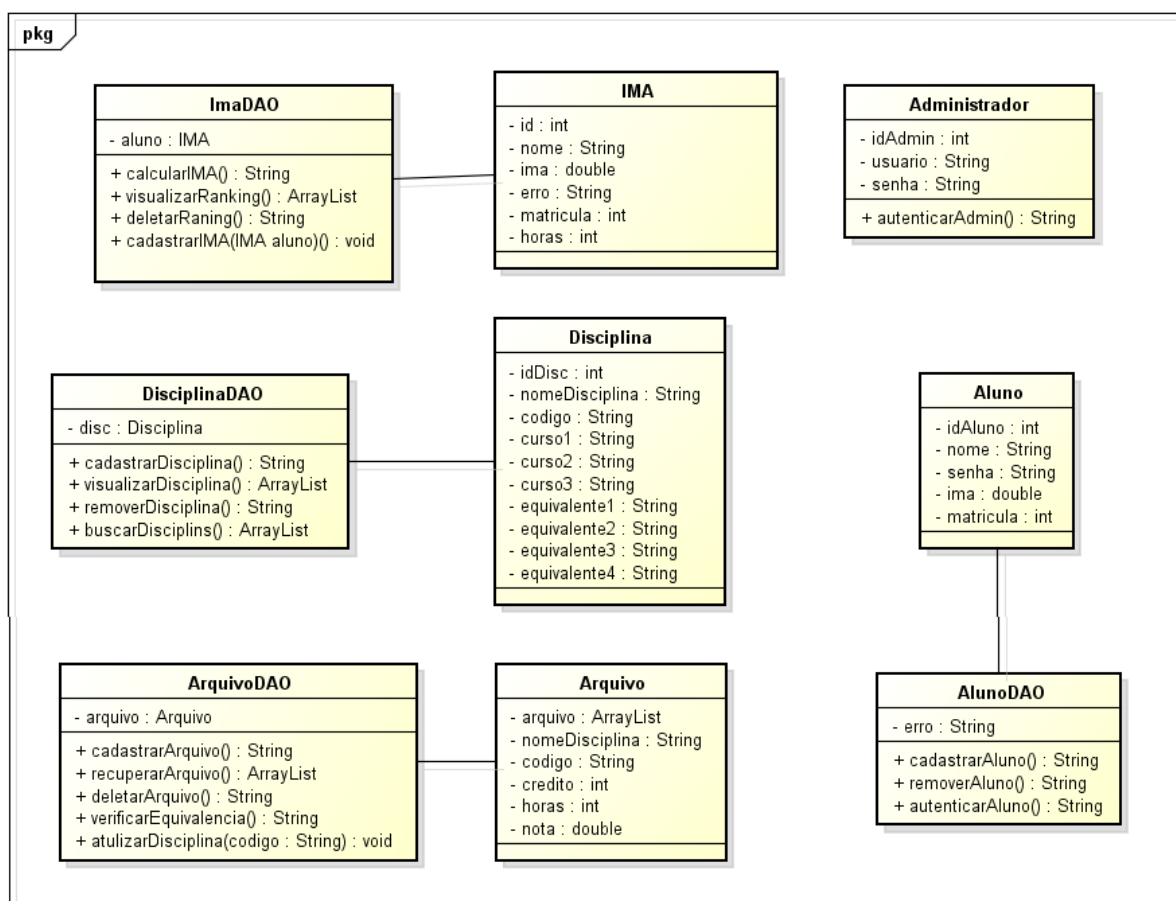


Fonte: Autoria própria

## 2.2.2 Diagrama de Classe

Como o SIPAF é um sistema orientado a objetos (OO), o próximo passo é dispor as classes de programação, especificando os atributos e métodos. Atributos são variáveis que definem a classe e os métodos são sub-rotinas que permitem a comunicação entre objetos. Para melhorar apresentar a estrutura do projeto, os atributos e métodos são dispostos em classes diferentes, de modo que as classes com métodos (referenciado com a *tag* DAO no nome) possuem uma instância da classe dos atributos, como mostra a figura 15.

Figura 15: Diagrama de classes



Fonte: Autoria própria

A classe IMA é a responsável por declarar as variáveis utilizadas na formalização das tabelas do *ranking*. Sua classe de métodos, ImaDAO, possui os seguintes eventos: *calcularIMA()*, que concentra a lógica da equação do IMA; *cadastrarIMA(IMA aluno)* que efetua o armazenamento do resultado do IMA de cada aluno no banco de dados;

*visualizarRanking()* possui a ação de retornar os valores do IMA salvos no banco; e *removerRanking()* destinado a apagar todas as informações do IMA.

A classe *Disciplina* possui as variáveis que incorporam informações essenciais utilizadas na verificação de disciplinas equivalentes e nos pesos de cada disciplina. *DisciplinaDAO* contém as funções *cadastrarDisciplina()*, *visualizarDisciplina()*, *buscarDisciplina()* e *removerDisciplina()*, responsáveis por salvar, visualizar, buscar e remover informações das disciplinas, respectivamente.

A classe *Arquivo* possui um atributo destinado a armazenar o arquivo inserido pelo administrador e a classe *ArquivoDAO* é responsável por salvar, buscar e excluir as informações dos arquivos por meio dos métodos *cadastrarArquivo()*, *recuperarArquivo()* e *deletarArquivo()*, respectivamente.

A classe *Aluno* e *AlunoDAO* foram definidas para tornar possível uma interação entre os alunos e o sistema, conforme exibido na figura 14. Assim, a classe *Aluno* abrange as informações dos alunos que utilizaram o sistema e a classe *AlunoDAO* compreende os eventos de cadastro, remoção e validação com os respectivos métodos *cadastrarAluno()*, *removerAluno()* e *autenticarAluno()*.

Por enquanto, o SIPAF será utilizado pela coordenação de BC&T campus Mossoró. Portanto, a classe *Administrador* será a única classe com métodos e atributos que validarão os dados do administrador com base no método *autenticarAdmin()*.

### 2.2.3 Diagrama de Atividade

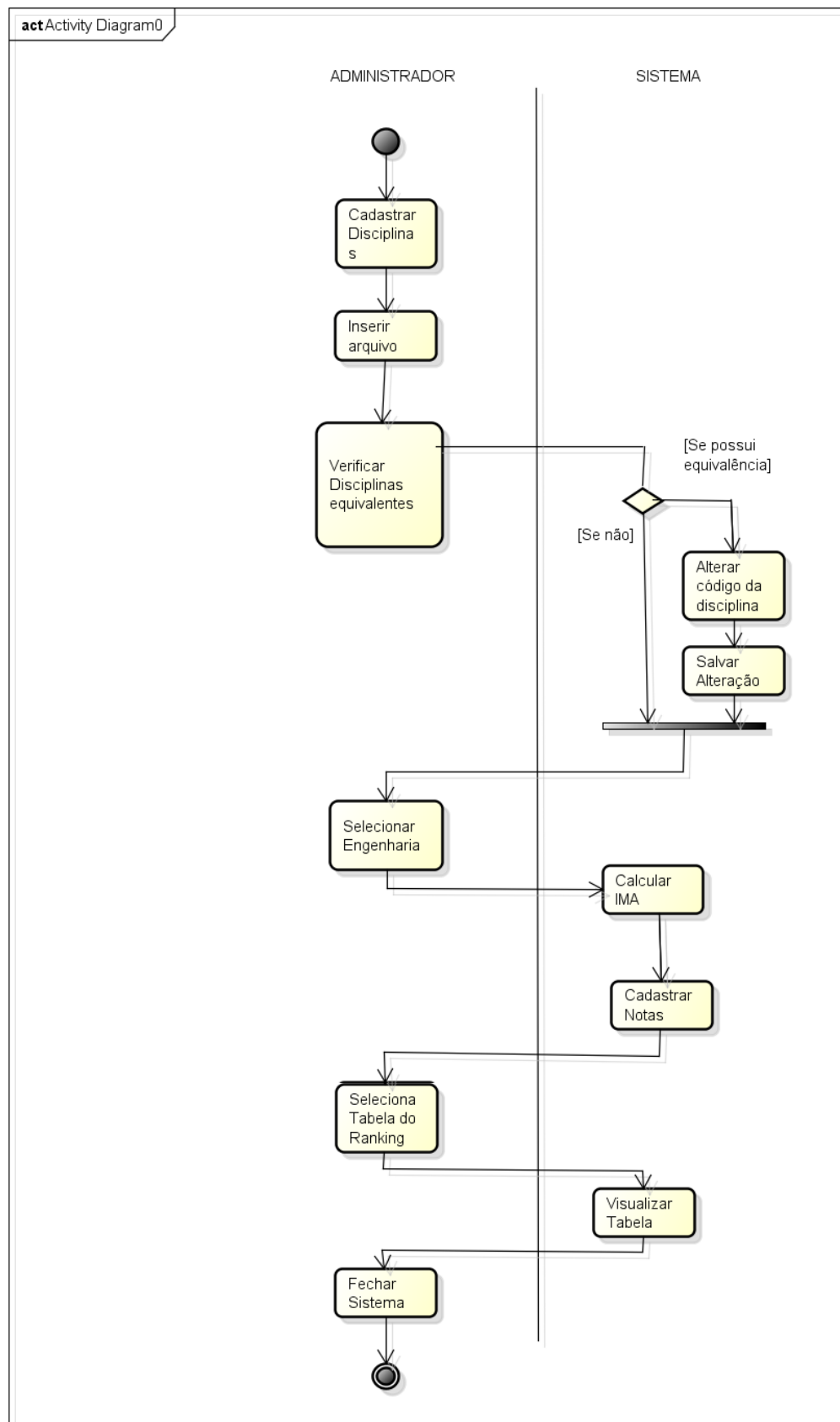
Para modelar o fluxo de controle de atividades, utiliza-se o diagrama de atividades. Com ele, o desenvolvedor tem conhecimento de como um processo funcionará, destacando a importância de cada atividade. A figura 16 apresenta a sequência entre as atividades para a obtenção do *ranking* classificatório do IMA, destacando a responsabilidade de cada atividade entre o administrador e o sistema.

O processo inicia a partir da decisão do administrador. Inicialmente, este deverá inserir o arquivo contendo informações dos alunos. Em seguida, deve ativar a busca por disciplinas eletivas no banco, com os dados do arquivo. Essa busca é realizada pelo próprio sistema, através de uma varredura em todas as linhas da tabela do banco, analisando código por código. Caso seja detectado um código equivalente a uma disciplina do BC&T, o SIPAF automaticamente atualiza o código para um estabelecido pelo administrador referente àquela

disciplina equivalente.

Após a verificação, o administrador seleciona uma das engenharias disponíveis para iniciar o cálculo do IMA. O cálculo é feito pelo sistema, que busca as informações no banco e efetua comparações e operações para cada aluno encontrado no arquivo de entrada. Nem todos os alunos são cadastrados, o resultado do IMA só é cadastrado se o aluno obtiver, no mínimo, 1230 horas do curso de BC&T. Após encerrar o cálculo, o administrador requisita a formação das tabelas de posição de cada aluno. O sistema é encarregado de ordenar os alunos em função do IMA, em ordem crescente, em uma tabela. Após a visualização das tabelas, o administrador encerra o sistema.

Figura 16: Diagrama de atividade

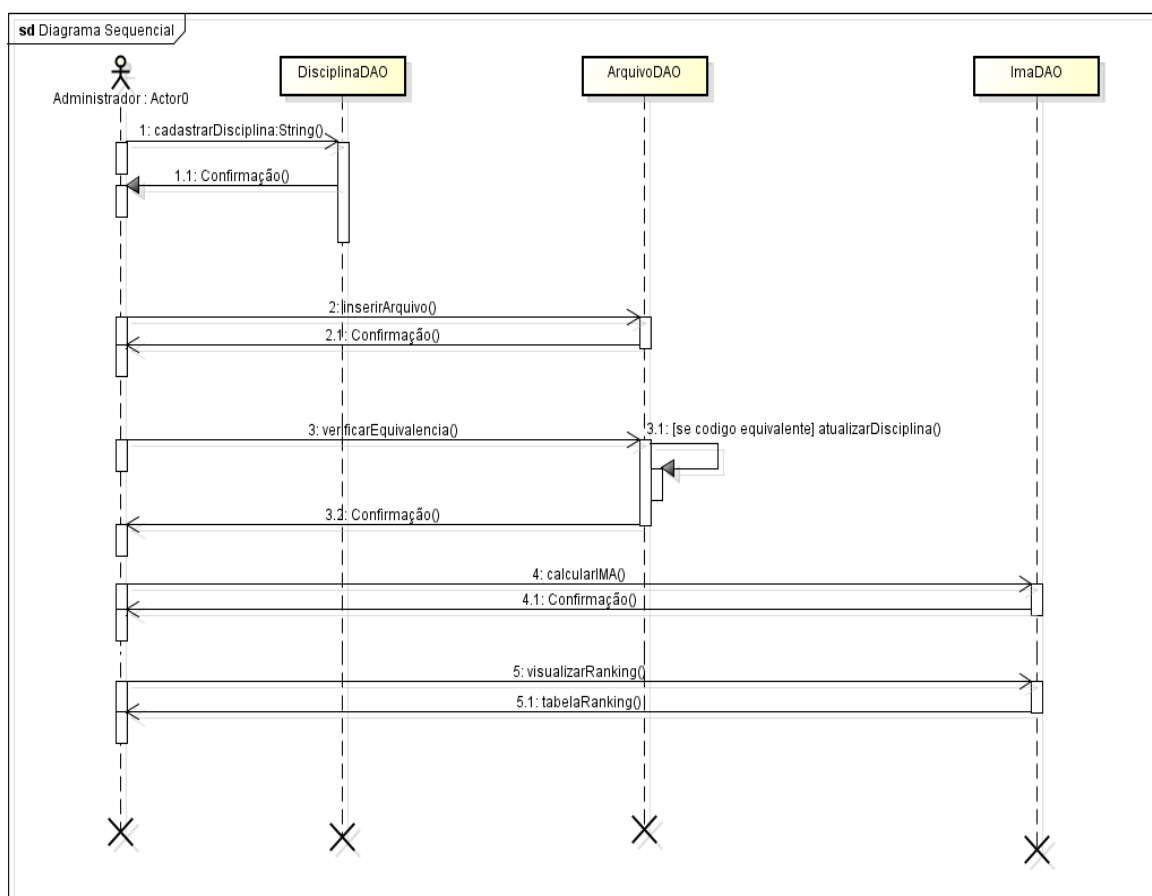


Fonte: Autoria própria

### 2.2.4 Diagrama de Sequência

Utilizando o diagrama de sequência, é possível estabelecer o tempo de resposta para cada objeto criado para um processo, assimilando interações por meio de métodos ou procedimentos. O diagrama de sequência também é útil para identificar como as mensagens são trocadas entre os objetos no decorrer de uma operação. Na figura 17, é exibido o diagrama de sequência referente ao processo classificatório dos alunos.

Figura 17: Diagrama de sequência



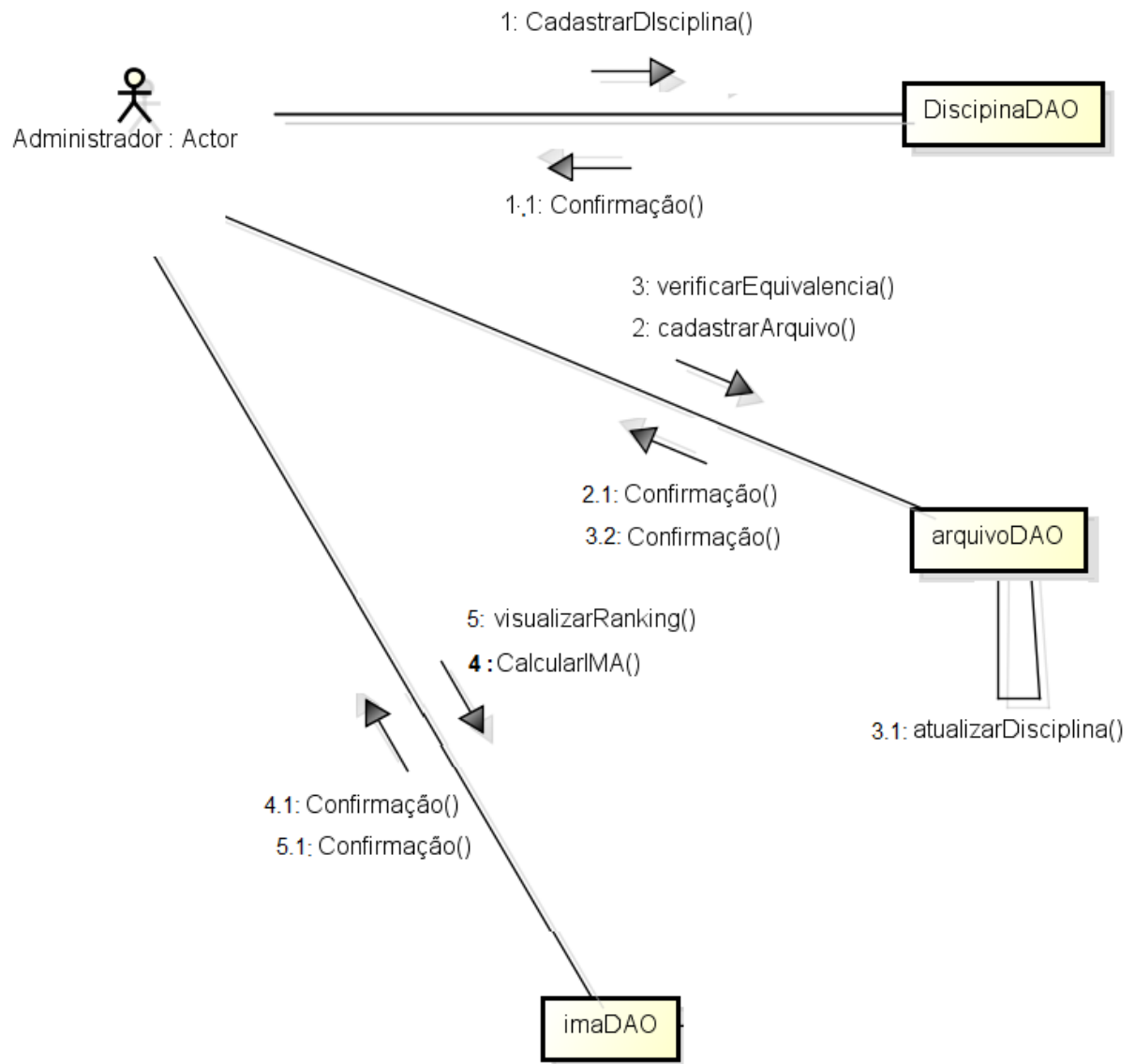
Fonte: Autoria própria

### 2.2.5 Diagrama de Comunicação

Outra forma de estruturar a ordem de um processo é utilizando o diagrama de comunicação. A diferença é que este define a estrutura vinculando os objetos criados, indicando as mensagens trocadas entre eles, estabelecendo critérios de prioridade na

realização do evento por meio de uma numeração. A Figura 18 exibe como está esquematizado o processo de classificação entre os objetos.

Figura 18: Diagrama de Comunicação

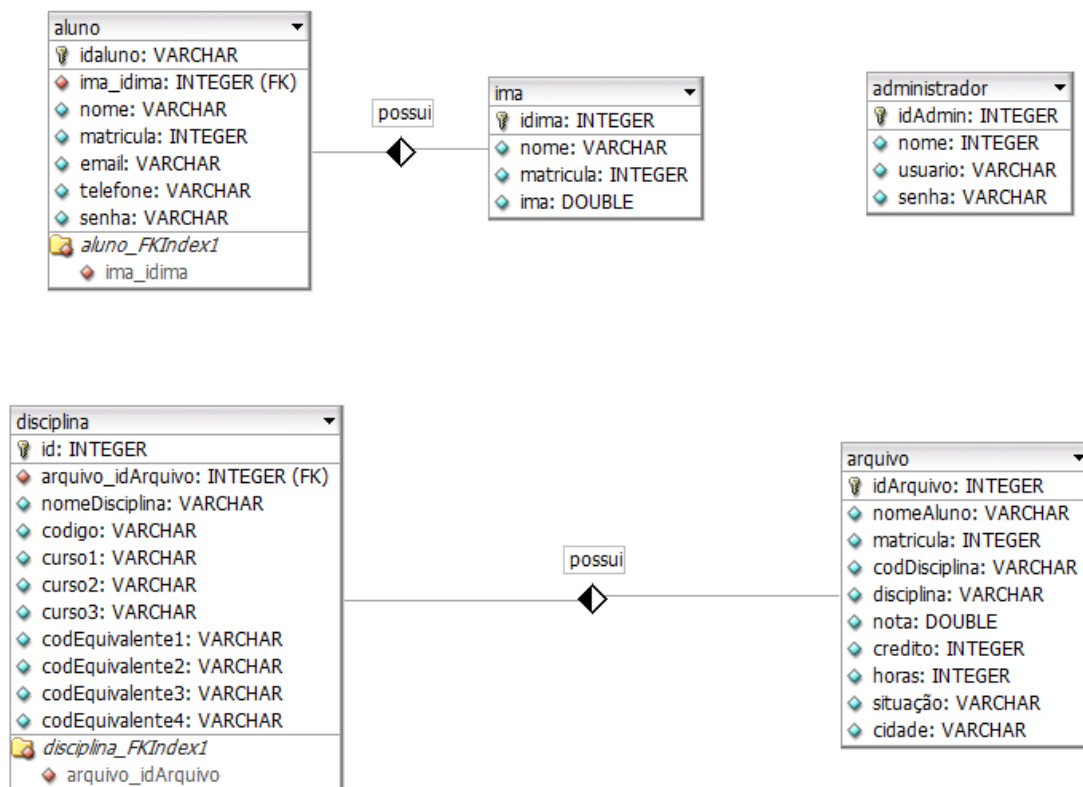


Fonte: Autoria própria

## 2.2.6 Modelo Relacional do Banco de Dados

O modelo relacional organiza as tabelas utilizadas no SIPAF, definindo seus relacionamentos no banco de dados, está estruturado conforme a figura 19.

Figura 19: Modelo Relacional do Banco de Dados



Fonte: Autoria própria

A tabela do banco de dados é formada por colunas que especificam o identificador (ou atributo) do valor que será armazenado, e por linhas que armazenam os dados registrados. Nesse caso, em alguns campos não é necessário armazenar dados, se estes forem especificados como vazios. A chave, que aparece em todas as tabelas, evidencia qual é a variável utilizada como identificador do registro armazenado na tabela, sendo esse um valor único na tabela, ou seja, que não se repete. A tabela ainda mostra o nome utilizado para identificar um atributo e o seu tipo (se é um número inteiro ou fracionário, uma palavra, uma data, etc).

Na tabela administrador, temos os atributos essenciais para a verificação do usuário administrador, como *login*, senha e nome. Na tabela ima, são definidos os atributos necessários para armazenar os valores finais recolhidos no cálculo do IMA e montar o *ranking*. A tabela ima possui um relacionamento com a tabela aluno de um para muitos, isto é, uma tabela ima armazenará dados de vários alunos.

A tabela aluno é composta por atributos utilizados na identificação do aluno, alguns desses utilizados na verificação de usuário, como matrícula e senha. A tabela arquivo

especifica os dados que serão armazenados, para o cálculo do IMA, do arquivo CSV inserido pelo administrador. Nota-se que a tabela arquivo possui um relacionamento com a tabela de disciplinas de um para muitos (um arquivo possui várias disciplinas).

A tabela disciplina armazena o nome da disciplina e o seu código, o nome dos cursos aos quais a disciplina possui mais afinidade, ajudando na declaração de pesos na equação do IMA, e campos para receber os códigos de disciplinas equivalentes para buscar no arquivo e trocar pela disciplina cadastrada.

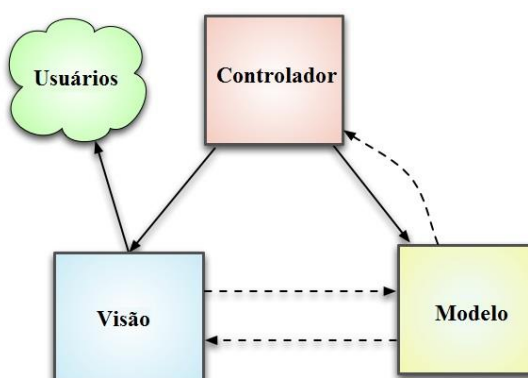
## 2.3 Referencial Teórico

Nesse capítulo, serão abordados o modelo estrutural empregado e as ferramentas utilizadas para o desenvolvimento do SIPAF, evidenciando conceitos, características e justificativas para a sua aplicação.

### 2.3.1 Modelo – Visão – Controle

O MVC (Modelo – Visão – Controle, em inglês *Model – View – Controller*) é um paradigma utilizado na modelagem de sistemas, conhecido por segmentar a estrutura de um projeto em três tipos de entidades: o controle para os valores de entrada de usuário, o modelo para a modelagem do mundo externo e a visão responsável pela atualização visual do usuário – cada entidade, portanto, é especializada em sua tarefa, como se observa na figura 20 (Burbeck, 1997).

Figura 20: Arquitetura MVC



Fonte: (Berbert, 2013)

O controle define o comportamento da aplicação a partir dos valores de entrada do usuário inseridos por meio de cliques de mouses ou teclado, por exemplo, para em seguida, analisar e mapear as instruções, seguindo um modelo predefinido pelo desenvolvedor ou atualizando uma resposta à página de interação da máquina com o usuário (em inglês *User Interface*, ou *UI*).

O modelo define a lógica e as regras de negócio, ou seja, é responsável pelo gerenciamento dos dados; responde a pedidos sobre o estado atual do processamento de dados (geralmente a partir do campo de visão do usuário) e atende a instruções para alterar o estado (geralmente, a partir do controle) (Burbeck, 1997).

A visão tem o objetivo de apresentar as informações ao modelo e capturar eventos do usuário, ou seja, os dados recebidos serão compilados por toda a lógica do modelo definido e atualizado na área do usuário (Souza, Pires, & Barros, 2003).

Por suas entidades apresentarem independência, o modelo MVC garante uma melhoria na estrutura do sistema a ser desenvolvido, pois favorece uma visão específica da arquitetura da aplicação, o que possibilita a melhor visualização de possíveis problemas. O MVC também facilita a programação de múltiplos clientes utilizando o mesmo modelo, ao favorecer a inserção de novos campos de visão (ou páginas de usuários), além de possibilitar a realização de atualizações em uma entidade sem prejudicar às outras.

O MVC foi escolhido para conceber o SIPAF por oferecer uma divisão clara de toda a estrutura do sistema, o que permite um gerenciamento eficaz do desenvolvimento. Dividir um projeto em partes específicas garante mais estabilidade, pois o desenvolver tem total domínio e visão de o que está sendo construído. Buscar alternativas oportunas para a lógica, garantir total controle de manipulação de dados e comunicar-se com o usuário de forma objetiva, são metas que o SIPAF necessita para realização dos requisitos estabelecidos e isso se torna alcançável ao aplicarmos o modelo MVC.

### 2.3.2 Ferramentas de Desenvolvimento

O desenvolvimento do SIPAF compreende uma ferramenta utilizada na montagem de páginas Web (*PrimeFaces*), um gerenciador de projetos e dependências (*Maven*), comandos de ligação entre as funcionalidades das entidades via JSF (*Java Server Faces*) e conexão entre as classes do sistema com o banco de dados através do *Hibernate* para atender aos requisitos levantados.

### 2.3.2.1 Primefaces

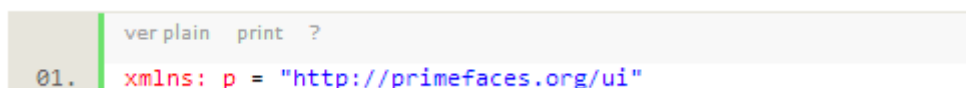
*Primefaces* é uma biblioteca de código aberto de componentes JSF 2.0 utilizados para a construção de interfaces de usuário, com um suporte nativo a AJAX (*Asynchronous Javascript and XML*), *JavaScript* e gráficos. Sua principal característica é ocultar a complexidade do *design* da interface sem perda de desempenho, qualidade e flexibilidade (PrimeTek, 2014). A sua arquitetura é organizada em três módulos: *UI Components*, *Optimus* e *FacesTrace*.

No módulo *UI Components* são encontrados os componentes AJAX, *JavaScript* e gráficos que garantem a facilidade no manuseio de objetos multimídia e de animação que podem ou não interagir com o usuário. O módulo *Optimus* está encarregado de facilitar o desenvolvimento junto com o JSF, ou seja, garante a manutenção da integridade no código e a garantia de envio de dados, evitando o acúmulo de linhas com declarações excessivas (Carmisini&Vahldick, 2012). O *FacesTrace* garante o desempenho nas aplicações baseadas em JSF, calculando o tempo de execução e rastreando os componentes (Lohn&Vahldick, 2013) utilizando *trace component*, componentes que coletam informações dos dados e problemas entre os eventos.

O *PrimeFaces* ainda possui um módulo com ferramentas voltadas para o desenvolvimento de interface de aplicações para dispositivos móveis, denominada *TouchFaces*. Como o SIPAF não é um aplicativo para dispositivos móveis, o *TouchFaces* não será aprofundado nesse trabalho. Para mais informações, consultar o site do *PrimeFacesMobile* (PrimeTek, Primefaces, 2014).

Por ser uma biblioteca leve, não há necessidade de realizar configurações ou buscar dependências para a sua utilização, basta incluir a biblioteca no projeto e importar o *namespace* como mostra a figura 21 na página XHTML (*eXtensible Hypertext Markup Language*). A declaração do *namespace* garante que os componentes *PrimeFaces* serão referenciados em toda página sempre que o desenvolvedor utilizando a *tag* `<p:>` para todos os componentes (PrimeTek, PrimeFaces, 2014).

Figura 21: Namespace do PrimeFaces



```

01.  xmlns:p = "http://primefaces.org/ui"
  
```

Fonte: (PrimeTek, PrimeFaces, 2009)

O *PrimeFaces* foi escolhido para o desenvolvimento do SIPAF por ser uma ferramenta simples, completa e intuitiva, capaz de formalizar páginas Web ricas em recursos e com um número de linhas reduzido, além de criar temas que podem ser personalizados pelo desenvolvedor, sem a necessidade de utilizar componentes de outras tecnologias. Na figura 22, podemos visualizar um exemplo de formulário desenvolvido no SIPAF com *PrimeFaces*.

Figura 22: Formulário de cadastro desenvolvido com *PrimeFaces*

O formulário, intitulado "Cadastro de Disciplinas", apresenta os seguintes elementos:

- Disciplina:** Campo de texto.
- Código:** Campo de texto.
- Curso 1, 2, 3:** Seletores de lista suspensa.
- Equivalente 1, 2, 3:** Campos de texto.
- Botão:** "Cadastrar".
- Tabela:**

| Código | Disciplina | Curso 1 | Curso 2 | Curso 3 |
|--------|------------|---------|---------|---------|
|        |            |         |         |         |
- Menu de Seleção:** Aberto no campo "Equivalente 3", mostrando opções:
  - Ciência e Tecnologia
  - Eng. Civil
  - Eng. Energia
  - Eng. Mecânica
  - Eng. Petróleo
  - Eng. Produção
  - Eng. Química

Na barra de rodapé, há informações de contato: "Secretaria de BCT de Mossoró, Av. Francisco Mota, 572, Bairro Costa e Silva. Telefone: (84) 3317 8253. Email: secretariabct@ufersa.edu.br".

Fonte: Autoria própria

No formulário, foi utilizado um componente `<p:panel>` para gerar o painel com cabeçalho onde estão os textos, campos e botão que indicam ao usuário o que é esperado como valor, onde inserir e iniciar evento, respectivamente. Para formatar esses recursos de interação com o usuário, foi utilizado o componente `<p:painelGrid>` que os organiza por meio de colunas. Os textos são desenhados através do componente `<p:outputLabel>`, os campos de inserção de valores, pelo componente `<p:keyboard>` que auxilia a digitação com um teclado virtual se necessário. O menu de seleção é moldado pelo componente `<p:selectOneMenu>` que utiliza o componente `<f:selectItem>` do JSF para atribuir valores ao menu. O código do botão que realiza uma ação, definida pelo desenvolvedor, é atribuído pelo componente `<p:commandButton>` como pode ser visto na figura 23 (PrimeTek, PrimeFaces, 2014).

Figura 23: Código do formulário com *PrimeFaces*

```

35 <p:panel header="Cadastro de Disciplinas" >
36   <h:panelGrid columns="6" cellpadding="10" >
37
38     <h:outputText />
39
40     <p:outputLabel value="Disciplina: " styleClass="alinhar" />
41     <p:keyboard value="" id="disciplina" />
42
43     <p:outputLabel value="Código: " styleClass="alinhar" />
44     <p:keyboard value="" id="codigo" />
45     <h:outputText />
46
47
48     <p:outputLabel for="eletival" value="Curso 1:" />
49     <p:selectOneMenu id="eletival" value="" >
50       <f:selectItem itemLabel="Nenhum" itemValue="" />
51       <f:selectItem itemLabel="Ciencia e Tecnologia" itemValue="Ciencia e Tecnologia" />
52       <f:selectItem itemLabel="Eng. Civil" itemValue="Eng. Civil" />
53       <f:selectItem itemLabel="Eng. Energia" itemValue="Eng. Energia" />
54       <f:selectItem itemLabel="Eng. Mecânica" itemValue="Eng. Mecanica" />
55       <f:selectItem itemLabel="Eng. Petróleo" itemValue="Eng. Petroleo" />
56       <f:selectItem itemLabel="Eng. Produção" itemValue="Eng. Producao" />
57       <f:selectItem itemLabel="Eng. Quimica" itemValue="Eng. Quimica" />
58     </p:selectOneMenu>
59
60     •
61     •
62     •
63
64   <p:commandButton value="Cadastrar" action="" type="Submit" />

```

Fonte: Autoria própria

Outro exemplo de utilização do *Primefaces* no SIPAF diz respeito à estrutura das páginas. Cada página é dividida em topo, onde é anexada a logomarca do sistema, em meio, onde se atribui todos os componentes de interação com o usuário, e rodapé. Com o *PrimeFaces*, essa divisão foi realizada a partir do componente `<p:layout>` que indica o tamanho da página e do componente `<p:layoutUnit>` que divide esse tamanho na posição que o desenvolvedor desejar (PrimeTek, PrimeFaces, 2014), como mostram o código da figura 24 e o resultado na figura 25.

Figura 24: Divisão da página em cabeçalho e centro com *PrimeFaces*

```

11 <h:body styleClass="ui-layout-hd" >
12
13 <p:layout fullPage="true" styleClass="ui-layout-hd" >
14
15 <p:layoutUnit position="north" resizable="true" maxSize="600" >
16
17 <center>
18 </img>
19 </center>
20
21 </p:layoutUnit>
22
23
24
25 <p:layoutUnit position="center" resizable="true" maxSize="600" styleClass="ui-layout-hd" >
26 <p:breadcrumb>
27
28 <p:menuitem value="Sports" url="#" />
29 <p:menuitem value="Área de Trabalho" url="areaAdministrativa.jsf" />
30 <p:menuitem value="Sair" url="index.jsf" />
31
32 </p:breadcrumb>

```

Fonte: Autoria própria

Figura 25: Resultado do código da Figura 23

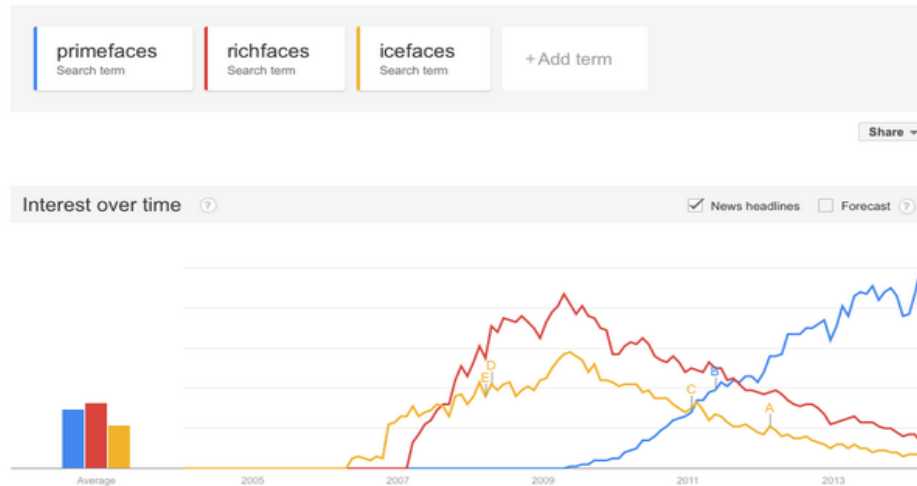


Fonte: Autoria própria

Outros componentes podem ser visualizados e testados no mostruário disponível no site do *PrimeFaces* (PrimeTek, PrimeFaces, 2014).

O *PrimeFaces* também conta com uma comunidade de desenvolvedores ativa e uma excelente documentação – o que resultou em sua popularidade diante de ferramentas concorrentes como mostra uma pesquisa elaborada pelo *Google Trends*, apresentada na figura 26 (PrimeTek, 2014).

Figura 26: Popularidade do *PrimeFaces* versus concorrentes



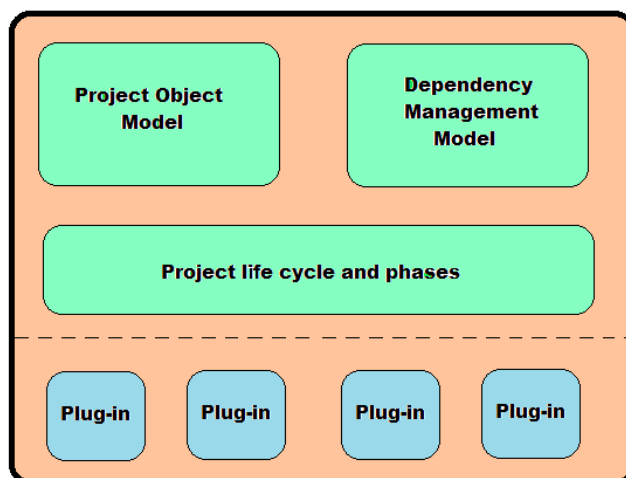
Fonte: (PrimeTek, 2014)

### 2.3.2.2 Maven

O *Apache Maven*, ou *Maven*, é uma ferramenta específica para o gerenciamento de projetos Java, hospedado pela *Apache Software Foundation*, que automatiza o desenvolvimento de *softwares*. Com esta ferramenta, é possível reduzir a complexidade do sistema, elaborando relatórios e documentação específica de eventos, e gerenciar dependências utilizadas no *software*, facilitando assim, a compreensão e o desenvolvimento do projeto ao desenvolvedor (Foundation, 1999).

Os componentes que formam o *Maven* são o POM (*Project Object Model*), o modelo de gestão de dependências, o ciclo de vida e fases e *plugins* (Figura 27).

Figura 27: Estrutura do Maven

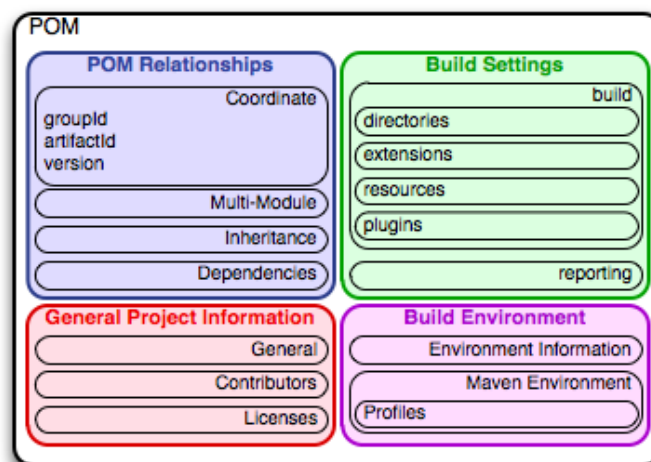


Fonte: Modificado de (Santos, 2008)

O POM destina-se a descrever todas as dependências, diretórios, ordem de compilação, *plugins*, especificações do sistema, configuração do ambiente e outros componentes externos utilizados na elaboração do projeto. O POM possui um arquivo XML (*eXtensibleMarkupLanguage*) denominado `pom.xml` no qual, de forma hierárquica, são definidas as dependências necessárias do sistema. Essa definição garante que todo artefato, como é dito na linguagem *Apache Maven*, seja incluso no projeto criado a partir de uma busca em um repositório local ou repositório externo da própria ferramenta (Bergsten, 2004).

O POM consiste em quatro categorias, conforme exibido na figura 28.

Figura 28: Arquitetura do POM



Fonte: (Sonatype, 2008)

No *POM Relationships* são armazenadas as definições de outros projetos *Maven*, a fim de herdar dependências e funcionalidades; e as coordenadas de arquivos dos repositórios.

Em *Build Settings*, pode-se personalizar o comportamento de compilação do *Maven* como, por exemplo, adicionar novos *plugins* ou fixar metas no ciclo de vida, modificar a localização de testes e personalizar parâmetros de localização do sistema.

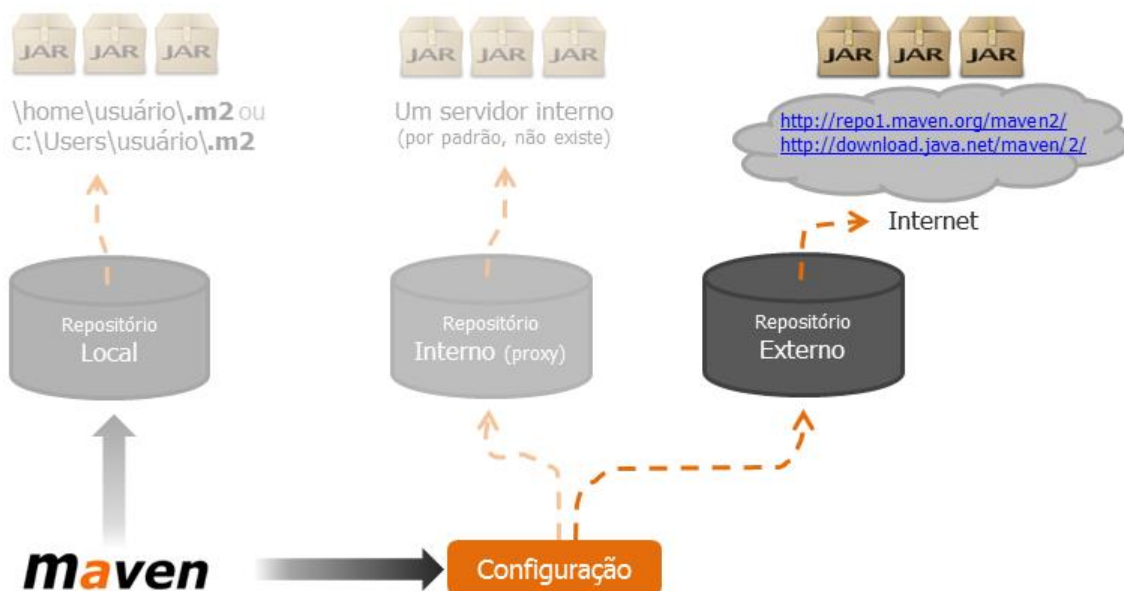
Através do *General Project Information* é especificado o nome do projeto, o diretório de armazenamento, nomes de contribuintes e desenvolvedores do sistema, e a licença do projeto.

No *Build Environment* é referenciado o ambiente de desenvolvimento do projeto. Um exemplo consiste em utilizar dois ambientes de construção: um de produção e outro de desenvolvimento. No ambiente de desenvolvimento, o sistema está configurado para ler a partir de uma instância de um banco de desenvolvimento, em execução na máquina local, enquanto na instância de produção é a partir de um banco de dados de produção. Essa categoria auxilia na portabilidade de execução em máquinas diferentes (Sonatype, 2008).

O repositório local armazena alguns artefatos localmente, facilitando uma busca rápida quando é necessário resolver alguma dependência ou executar um *plugin*. Para isso, é criado um repositório *.m2* no diretório de usuário. O repositório externo é formado por vários repositórios *Maven* encontrados na Internet (Bergsten, 2004). A necessidade do repositório global é visível quando o *Maven* não encontra uma dependência requisitada no repositório local (Figura 29). O *Maven* ainda conta com um repositório utilizado nas empresas, conhecido

como repositório interno para reduzir o consumo de link de Internet para compartilhar as bibliotecas .jar e outros artefatos utilizados pelos desenvolvedores da empresa, utilizando uma ferramenta designada *Maven Proxy* (Foundation, Apache Software, 2014).

Figura 29: Acessos a Repositórios *Maven*



Fonte: (Camargo, 2011)

O modelo de gestão de dependências é responsável por administrar todas as dependências solicitadas no desenvolvimento de um *software*, ou seja, ele é quem faz as buscas nos repositórios e armazena o resultado no repositório local, caso a máquina do desenvolvedor ainda não tenha o artefato.

O ciclo de vida e fases especifica todas as etapas de construção de um projeto *Maven*, desde a validação de requisição de dependências até a implementação do projeto, conforme exibido na figura 30.

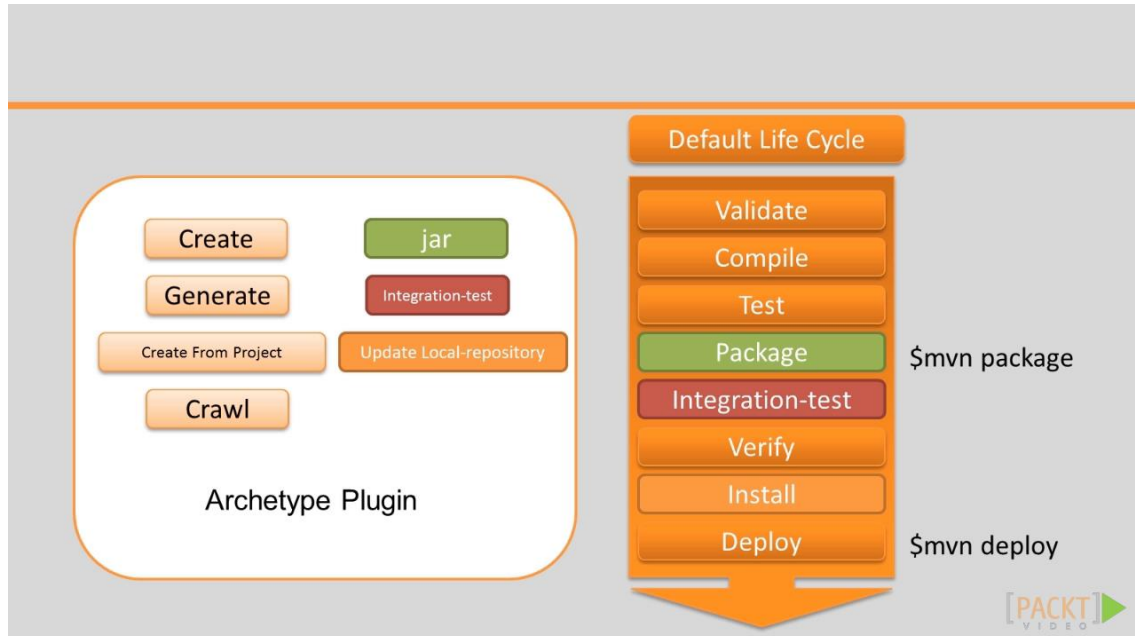
No processo de validação, o *Maven* busca os recursos da dependência solicitada no POM. Esses recursos são compilados (unidos) formando um diretório de dados que, em seguida, passam para a fase de teste, na qual se verifica a existência de erro de compilação.

Após a fase de teste, esse ficheiro é empacotado em arquivo do tipo .jar para encapsulamento de uma ou mais classes Java, .war para encapsulamento de arquivos *Web* desenvolvidas por *Servlets* ou .ear para encapsulamento de componentes *Java Platform, Enterprise Edition* (JavaEE). Em seguida, o arquivo passa por novo teste de integração para verificação de erros antes de ser instalado no repositório local. Após instalado, o arquivo é

implantado no projeto.

O ciclo de vida também é responsável por realizar uma ligação entre os modelos conceituais (o que é realizado pelo *Maven*, por exemplo, o POM) e os modelos lógicos (como é feito pelo *Maven*, por exemplo, repositórios) do projeto.

Figura 30: Ciclo de vida e fases do *Maven*



Fonte: (Bogahapitiya, 2013)

Os *plugins* são módulos de *software* utilizados pelo *Maven* para incorporar mais funcionalidades ao sistema, obtendo quaisquer dependências via repositório interno ou externo sem a necessidade de configuração (Santos, 2008).

Os artefatos são arquivos constituídos por bibliotecas binárias ou um executável. O *Maven* armazena uma única cópia do artefato em seu diretório, mesmo que vários projetos utilizem o mesmo artefato, o que garante o seu controle e evita possíveis conflitos entre os projetos. Cada artefato é identificado por meio de um conjunto de valores, chamado de coordenada, como mostra a figura 31. Os valores de coordenadas são: *GroupID* que se refere ao órgão ou entidade responsável pela criação do artefato, o *ArtifactID* que indica o identificador ou nome do arquivo e a *Version* que constitui a versão do artefato (Bergsten, 2004).

Figura 31: Exemplo de coordenada de artefatos

```
<dependency>
  <groupId>org.primefaces</groupId>
  <artifactId>primefaces</artifactId>
  <version>4.0</version>
</dependency>
```

Fonte: (Rodrigues, 2014)

O *Maven* é adotado no SIPAF pela simplicidade oferecida para organizar todo o ambiente de desenvolvimento. Buscar, instalar, gerenciar e configurar um arquivo são tarefas que chegam a ser complexas na hora de desenvolver uma aplicação e o *Maven* as trata de forma rápida e eficaz, especificando erros, caso ocorram, até a última fase do ciclo de vida e realizando o mesmo processo nas dependências de uma dependência requisitada pelo desenvolvedor.

### 2.3.2.3 Java Server Faces

JSF é um *framework* MVC para o desenvolvimento Web fundamentado no modelo de interfaces gráficas baseadas em eventos. Concebido em Java, é uma tecnologia formalizada pelo *Java Community Process* (JPC) – “entidade cujo objetivo é especificar a evolução da tecnologia Java, de acordo com o interesse do mercado” (Luckow& Melo, 2010).

O JSF possui um rico modelo de componentes prontos de interface de usuário para serem utilizados na construção de aplicativos Web. Tais componentes fazem parte da transição de dados entre o cliente e o servidor.

Os componentes JSF de uma página serão manipulados de forma hierárquica, conforme exemplificado a seguir, tendo como base o desenvolvimento do SIPAF. Por exemplo, para realizar um cadastro na página XHTML, primeiro é especificado um formulário, dentro do formulário é criado um painel que irá compor textos, caixas de texto e botões utilizados para enviar os dados. A ordem formulário – painel – texto, caixa de texto e botões formam uma hierarquia chamada de árvore de componente, onde o formulário no exemplo é a raiz e o texto, caixas de texto e botões são as folhas.

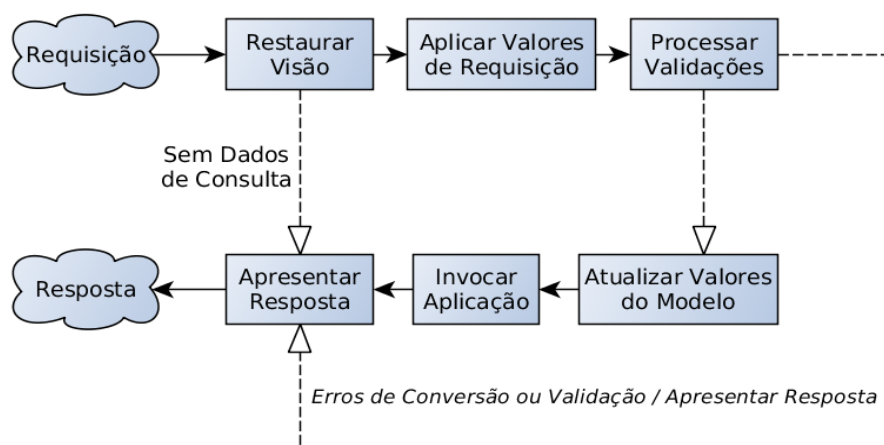
Uma preocupação relacionada à manipulação de componentes é a forma na qual estes serão armazenados entre os eventos realizados no sistema. Afinal, armazená-los em um servidor ocupa significativa memória. É necessário então que o JSF seja capaz de salvar informações, em algum lugar, com o objetivo de possibilitar a reconstrução da árvore de

componentes e restaurar o estado do componente quando recebe o próximo pedido.

Um agravante é a possibilidade de atualizar os componentes na área do cliente. O servidor detecta mudanças após uma nova requisição de evento, em outras palavras, a detecção de mudança só ocorre com uma chamada de método ou página. Diante disso, o JSF lida com esses problemas definindo um ciclo de vida que garante uma visão clara e objetiva para o acompanhamento de cada evento (Bergsten, 2004).

A figura 32 representa as seis fases do ciclo de vida do JSF: Restauração da Visão, Aplicação de Requisição, Processo de Validação, Atualização de Modelos, Invocação da Aplicação e Renderização de Resposta.

Figura 32: Ciclo de vida do JSF



Fonte: (Neves, 2014)

Na fase de restauração de visão, é montada uma árvore de componentes a partir de uma requisição à página JSF. Existem dois eventos distintos nessa etapa. O primeiro evento ocorre quando é realizada a primeira requisição a uma página – caso em que apenas é montada a árvore de componentes, em seguida, transita-se para a última fase, de renderização de resposta, para ser visualizada pelo cliente. O segundo evento acontece quando a página já está montada e ocorre alguma ação como, por exemplo, o clique em um botão (evento denominado *postback*). O armazenamento dessas informações ocorre no objeto *FacesContext* que contém dados do evento em andamento (Oracle, 2014).

Na aplicação de valores de requisição, cada componente da árvore gerencia seus respectivos valores, ou seja, cada componente terá a oportunidade de buscar o valor submetido na página, em seus respectivos campos. Um exemplo é o componente *inputText* com identificador nome, que buscará uma *String* com o valor de nome. No caso do

componente que espera um valor inteiro, nessa fase, ainda recupera o valor no formato de *String*, uma vez que o HTTP (*Hypertext Transfer Protocol*) opera exclusivamente com valores desse tipo.

Na fase de validação, os valores recuperados por todos os componentes serão verificados e, se necessário, convertidos para o valor real e, em seguida, serão validados. O valor nessa fase é um valor local. A conversão existe caso um dos componentes esteja esperando um valor diferente do que foi recuperado, por exemplo, se o valor esperado for um tipo *float*, o valor que antes era uma *String* agora passa a ser *float*. A validação ocorre devido a condições lógicas especificadas pelo desenvolvedor como, por exemplo, quando um componente que unicamente aceita 6 (seis) caracteres de uma *String*, na validação terá o seu valor limitado ao número especificado, caso fosse maior que o estipulado para o componente. Caso ocorra algum erro nessa fase, de conversão ou de validação, uma mensagem de erro é gravada e enviada para a última fase, onde será renderizada e exibida ao cliente.

Na atualização de modelos, os valores serão referenciados em uma classe de objeto, ou seja, um *ManagedBean*. No SIPAF, existe uma anotação *ManagedBean* de uma classe *Aluno* que possui um atributo nome do tipo *String*, cujo valor local do componente passará a ser do atributo ligado a ele.

Durante a fase de invocação da aplicação, a implementação JSF manipula os eventos do aplicativo. Um exemplo acontece no cadastro de alunos no SIPAF, quando o usuário preenche todos os dados e clica no botão enviar do formulário. Essa fase invoca o método da classe referenciado no botão, assim os valores dos atributos da classe passarão por eventuais mudanças daquele método. Estes eventos são ações que retornam geralmente uma *String* que está associada a uma navegação ou resultado (Gonçalves, 2010).

Na renderização de resposta, o JSF retorna uma página XHTML como resposta à solicitação do evento requisitado pelo cliente.

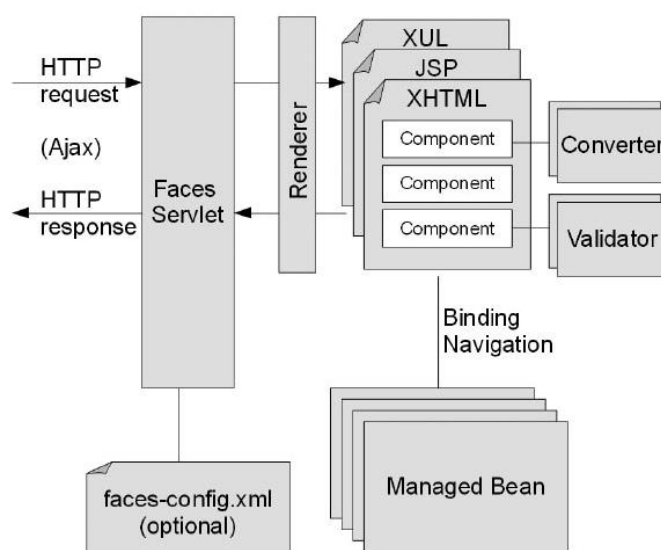
Essas fases garantem, ao desenvolvedor, a criação de uma infraestrutura de aplicativos sem a preocupação com detalhes do HTTP e sua integração com a interface do usuário, por meio de um modelo orientado a eventos, além de remover toda a logística de interação do sistema das páginas JSF deixando a cargo dos métodos de uma classe (Bergsten, 2004).

A arquitetura do JSF, como expõe a figura 33, é composta pelo *FacesServlets* que gerencia o ciclo de vida no processamento de construção de interfaces de usuário, o *Renderer* responsável por renderizar os componentes para o fluxo de saída, os Componentes que são as classes bases de toda a UI, o *Converter* utilizado na conversão de valores dos componentes no modelo *Object-to-String* ou *String-to-Object*, o *Validator* responsável por realizar uma

validação nos componentes com aspecto de verificação de erros, o *ManagedBean* utilizado no gerenciamento de navegação entre as páginas e classes do sistema e o AJAX que pode operar no envio e no recebimento de dados (Oracle, 2011).

O *faces-config* descreve as regras de navegação, os *beans* gerenciáveis, a configuração de localização e outros subelementos que compõem o projeto, no entanto, a sua aplicação passou a ser opcional a partir do JSF 2.0.

Figura 33: Arquitetura do JSF



Fonte: (Gonçalves, 2010)

Para utilizar JSF em um sistema, é necessário atribuir dois arquivos, um deles é o *jsf-api.jar*, em que se encontram as *Application Programming Interface* (API) públicas que controlam cada função descrita na arquitetura JSF e no ciclo de vida. O outro arquivo é o *jsf-impl.jar* que contém o tempo de execução por especificação (ORACLE, 2014).

O JSF também conta com uma biblioteca *JavaServerPages Standard Template Library* (JSTL) utilizada para a realização de atividades comuns como iterações, recuperação de dados de um banco e utilização de *taglibrary* (ou *taglib*) de internacionalização e para a manipulação de documentos XML, conforme demonstra a figura 34, sendo essa última recomendada pela *World Wide Web Consortium* (W3C) (Oracle, 2011).

Figura 34: Estrutura de tags do JSF

```
01. <html xmlns="http://www.w3.org/1999/xhtml"
02.     xmlns:h="http://java.sun.com/jsf/html"
03.     xmlns:f="http://java.sun.com/jsf/core"
04. >
05.
06. <h:head>
07.
08. </h:head>
09.
10. <h:body>
11.
12. </h:body>
13. </html>
```

Fonte: modificado de (PrimeTek, 2014)

O JSF 2.0, diferente de sua versão anterior JSF 1.2, dispõe de uma estrutura mais fácil e simples de manipulação de eventos, diminuindo a complexidade de sua utilização. Com uma melhoria de desempenho, passou a ter mais componentes e a utilizar *facelets* como tecnologia de *templates* padrão. Sua navegação pode ser feita de forma implícita, de modo que não é mais necessário registrar sua regra de navegação no *faces-config*, basta indicar o caminho da *view* sem a extensão que o próprio JSF vai buscar entre os arquivos do diretório. Pode ser declarada uma navegação condicional, caso o desenvolvedor ainda prefira utilizar navegações declarativas. Seu suporte a AJAX foi melhorado, sendo utilizável por meio de *tags*, e passou a ser utilizado o *ManagedBean* por anotação, ao invés de XML, o que garante maior praticidade e comunicação entre as classes e as páginas XHTML.

O SIPAF necessita de uma ferramenta que possa lidar com a complexidade de eventos (interação, envio de dados, manipulação de arquivos, conexões HTTP, navegação e conexão com o Sistema Gerenciador de Banco de Dados (SGBD)) que ocorrem em um aplicativo WEB proporcionando um desempenho satisfatório. O JSF mostrou-se como a melhor alternativa para a manipulação de eventos por existir um diversificado número de extensões próprias para uma mesma necessidade, possibilitando o desenvolvedor escolher a forma adequada da sequência de execução de sua aplicação. Essa característica vem garantindo, cada vez mais, espaço para o JSF no mercado e abrindo mais portas para melhorias em sua estrutura.

#### 2.3.2.4 Hibernate

O *Hibernate* é um *framework* de código aberto, desenvolvido em Java, utilizado no mapeamento objeto-relacional entre uma base de dados relacionais e o modelo objeto de uma aplicação, ou seja, é uma ferramenta que diminui a complexidade existente na conexão de uma programação orientada a objeto com um SGBD relacional.

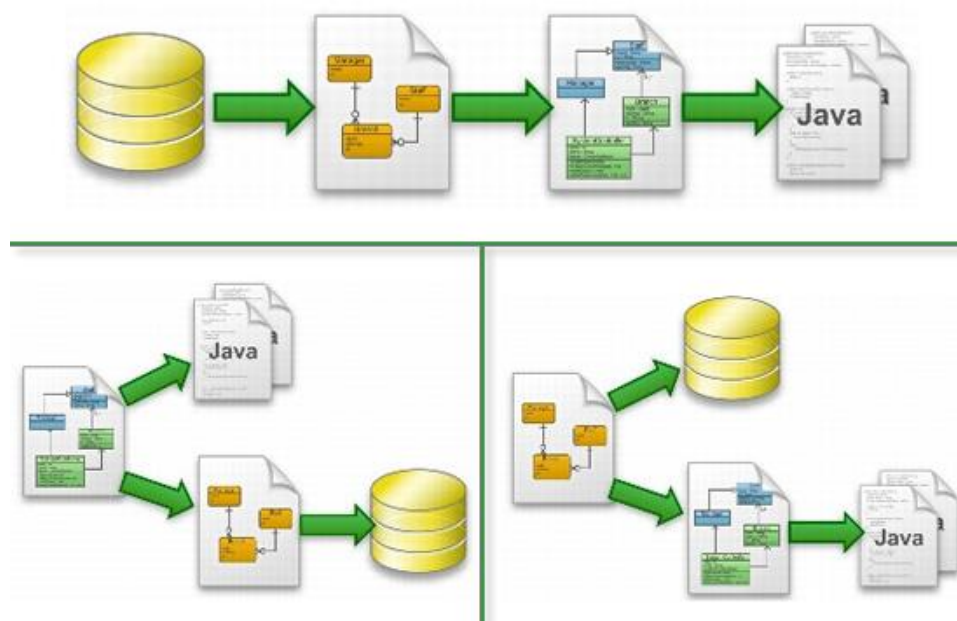
Para entender melhor como o *Hibernate* funciona na prática, o desenvolvedor deve, no mínimo, compreender os comandos da linguagem *Structured Query Language* (SQL) mesmo sem a necessidade de acesso direto a eles, afinal o *Hibernate* faz todo o gerenciamento. Com essa ideia, o desenvolvedor deve conhecer os comandos de definição *DataDefinitionLanguage* (DDL) como o *ALTER* e o *CREATE*, os de manipulação de dados *DataManipulationLanguage* (DML) como *INSERT*, *UPDATE* e *DELETE* e o comando *SELECT* de consulta *DataQueryLanguage* (DQL).

A *ObjectRelationMapping*, ou Mapeamento Objeto-Relacional (ORM) é um modelo de transformação de dados automatizado e reversível que, Segundo Bauer, “persiste, de maneira automática e transparente, os objetos de um aplicativo para tabelas em um banco de dados relacional. Em essência, transforma dados de uma representação para a outra” (Bauer & King, 2007). Tal modelo favorece as seguintes vantagens:

- Facilidade de especificação de mapeamento de dados;
- Melhoramento na padronização do sistema;
- Aumento na produtividade, pois sem os comandos SQL no código fonte, as classes passam a ser mais simples e desenvolvidas em menos tempo;
- Facilidade de manutenção, em função da redução no número de linhas do código fonte de um sistema.

Na figura 35, podemos visualizar o modelo de um sistema com e sem a utilização do ORM. Sem o modelo ORM o desenvolvedor tem o trabalho de escrever os códigos SQL, tornando o projeto improdutivo, pois será necessário ser bem específico nos comandos, o que deixa o sistema preso a um SGDB, além de aumentar o número de classes para gerenciar e dar manutenção, ou seja, o desenvolvedor tem que se preocupar com todo o processo de armazenamento ou inserção no banco.

Figura 35: Visualização de sistema sem modelo ORM (acima) e com modelo ORM (embaixo)



Fonte: (Eduardo, 2014)

O *Hibernate* segue as especificações do *Java Persistence API* (JPA) – um *framework* leve baseado em *PlainOld Java Objects* (POJOs). No JPA, os objetos possuem *design* simples e não dependem de herança de interfaces ou de classes de *frameworks* externos, além de mapeamentos por meio de anotações e/ou arquivo XML e consulta por linguagem específica que, no caso, é a *Java Persistence Query Language* (JPQL) (Medeiros, 2012).

JPA é um componente da especificação *Enterprise Java Beans* (EJB) – integrante da plataforma JavaEE criada para o desenvolvimento de aplicações Java multicamada, distribuídas, escaláveis e orientadas a objetos. EJB tem o objetivo de facilitar o trabalho do desenvolvedor em relação à infraestrutura do sistema (Luckow& Melo, 2010).

Para atender essas especificações, o *Hibernate* é formado por três softwares:

- *Hibernate Core*: Conhecido como *Hibernate3*, ou *Hibernate*, é à base de todo o processo de persistência da tecnologia. É formado por uma API nativa e metadados de mapeamento no arquivo XML. O *Hibernate Core* possui uma linguagem própria no estilo SQL, designada *Hibernate Query Language* (HQL) e possui uma interface para a realização de consultas chamada *criteria* (Luckow& Melo, 2010);

- *Hibernate Annotations*: Realiza o mapeamento objeto-relacional por meio de *annotations*, um tipo de *tag* especial com estilo *JavaDoc* que especifica classes e componentes que serão manipulados. A vantagem em relação ao mapeamento pelo arquivo XML é o número reduzido de linhas para as referências (Luckow & Melo, 2010);
- *Hibernate EntityManager*: Especializada em nível de interface, realiza consultas sobre todas as entidades, cria e remove instâncias de entidades persistentes e operações de persistência (Bernard, Ebersole, & Rei, 2012).

O HQL é uma linguagem muito semelhante ao SQL utilizada na obtenção de objetos em um banco de dados. A diferença para o SQL é que é uma linguagem orientada a objetos e não realiza manipulação de dados, ou seja, o HQL não atualiza, insere ou deleta dados de um SGBD (Bauer & King, 2005).

Com uma gramática escrita por meio de *String*, o HQL abrange noções de polimorfismo, associações e heranças que lhe garantem, segundo Bauer e King, as seguintes vantagens:

- A habilidade para ordenar os resultados da consulta;
- A habilidade para paginar os resultados;
- A habilidade para aplicar restrições a propriedades de objetos associados, relacionados por referência, ou contidos nas coleções;
- A habilidade para chamar funções SQL, definidas pelo usuário;
- *Subqueries* (consultas aninhadas);
- Agregação com o *groupID*, *having* e funções de agregação como soma, mínimo e máximo.

O MySQL é um SGBD de código aberto que utiliza linguagem SQL para realização dos comandos básicos do CRUD (criar, consultar, atualizar e remover, em inglês, *Create*, *Read*, *Update* e *Delete*). Popularizou-se por proporcionar processamento rápido dos dados e tempo curto de resposta sem exigir muito do hardware (Alecrim, 2008), se aplica tanto em aplicações Web como em sistemas corporativos ou *desktop*, se conecta facilmente com várias linguagens de programação e facilidade de realização de backup. O *Hibernate* fará conexão com o MySQL para armazenar todos os dados do SIPAF, para essa conexão ser

estabelecida é necessário utilizar um arquivo denominado `mysql-connector-java.jar` que possui todos os comandos de acesso para o Java.

O SIPAF recebe um arquivo de entrada, cujo conteúdo é uma tabela de dados distribuídos em linhas. O *Hibernate* foi escolhido por trabalhar de forma persistente no mapeamento dessas informações do banco relacional com a aplicação OO, o que simplifica o desenvolvimento da conexão, uma vez que torna possível a definição das colunas e das tabelas por meio de anotações nas classes de objeto.

### 3. Conclusões

Para garantir as funcionalidades corretas do SIPAF e atingir os objetivos propostos nesse trabalho, foram realizados alguns testes para cada requisito do sistema.

Na inserção dos dados do arquivo CSV foram realizados testes com mais de 200 (duzentos) mil linhas de informações, observando em algumas linhas aleatórias, se as informações estavam de acordo com o arquivo original. Testes na alteração dos códigos de disciplinas equivalentes foram realizados averiguando as informações das disciplinas cadastradas no sistema.

Para o cálculo do IMA, foi necessário utilizar uma planilha eletrônica para auxiliar no cálculo manualmente e, no final, comparar com o resultado encontrado pelo SIPAF. Esse teste contou com informações de 15 (quinze) alunos aleatórios e os resultados foram os mesmos. No entanto, em média, o cálculo realizado na planilha levou 6 (seis) minutos para buscar o resultados de um único aluno para uma engenharia enquanto o SIPAF levou 42 (quarenta e dois) segundos para efetuar o cálculos para os quinze alunos.

Em relação a cadastrar disciplinas e cadastrar aluno, foram realizadas diversas inserções, com êxito em todos os cadastros. O teste para verificar se o *ranking* estava sendo construído, de forma classificatória, foi realizado observando se a tabela montada com os valores do IMA calculadas estava em ordem decrescente. Todos os testes, incluindo os quinze alunos utilizados no teste para calcular o IMA ficaram com o índice em ordem decrescente.

A exclusão de informações do sistema foi analisada acessando diretamente o banco de dados pelo SGBD *MySQL* e confirmando a remoção dos dados das tabelas.

A autenticação tanto da conta administrativa como as dos alunos por meio de login e senha foram realizados diversas vezes, o que mostrou perfeita funcionalidade de controle de acesso.

Um problema encontrado no desenvolvimento do sistema foi a implementação de uma função que insere os valores do arquivo .csv no banco de dados. O método encontrado para prover essa funcionalidade baseou-se em um processo com demora significativa, de acordo com o número de linhas que o arquivo possui. Por outro lado, como já foi citado, obteve sucesso na função independentemente desse número.

Podemos concluir que o SIPAF trará um significativo benefício para a seleção das engenharias e das disciplinas eletivas, pois além de manter total controle na manipulação de inúmeros dados, realizar todos os cálculos do IMA com exata precisão em um curto período de tempo, além de possuir uma interface moderna que facilita a interação com o usuário.

#### **4. Trabalhos Futuros**

Sugerem-se funcionalidades adicionais ao SIPAF, tais como permitir alterações na equação do IMA - o que facilitaria sua expansão para outras instituições; disponibilizar um relatório para o aluno, mostrando os valores encontrados para cada disciplina integrante do seu respectivo histórico no processo do cálculo; gerar um gráfico comparativo entre a média geral de desempenho para cada semestre, efetuar a associação com o SIGAA da UFERSA; aceitar outras extensões de arquivos de dados de entrada; criar conta de acesso para mais de um administrador, no caso do curso BC&T da UFERSA que possui dois coordenadores (um para o curso diurno e outro para o curso noturno), identificar quem está utilizando o sistema por meio de uma conta própria, garantindo o controle de acesso; alterar a forma de inserir os valores do arquivo no banco de dados buscando mais velocidade no tempo de resposta.

Recomendamos, ainda, a implantação de recursos de segurança, a fim de garantir primordialmente a confidencialidade e a integridade dos dados.

## Referências

- Alecrim, E. (02 de abril de 2008). *Banco de dados MySQL e PostegreeSQL*. Acesso em 21 de julho de 2014, disponível em Infowester: <http://www.infowester.com/postgremysql.php>
- Bauer, C., & King, G. (2005). *Hibernate in Action*. Greenwich: Manning.
- Bauer, C., & King, G. (2007). *Java Persistence with Hibernate*. New York: Manning.
- Berbert, F. P. (2013). *Padrões de diretórios para aplicações do Zend Framework*. Acesso em 28 de junho de 2014, disponível em Viva o Linux: <http://www.vivaolinux.com.br/artigo/Implementando-servidor-de-aplicacoes-PHP-utilizando-Zend-Framework/?pagina=10>
- Bergsten, H. (2004). *Java Server Faces*. Sebastopol: O'REILLY.
- Bernard, E., Ebersole, S., & Rei, G. (10 de fevereiro de 2012). *Hibernate EntityManager*. Acesso em 2014 de junho de 02, disponível em Hibernate Commuity Documentation: [http://docs.jboss.org/hibernate/entitymanager/3.6/reference/en/html\\_single/](http://docs.jboss.org/hibernate/entitymanager/3.6/reference/en/html_single/)
- Bogahapitiya, K. (15 de novembro de 2013). Learning Apache Maven 3. *Learning Apache Maven 3*.
- Burbeck, S. (11 de março de 1997). *Applications Programming in Smalltalk-80(TM):How to use Model-View-Controller (MVC)*. Acesso em 20 de junho de 2014, disponível em CS Illinois: <http://st-www.cs.illinois.edu/users/smarch/st-docs/mvc.html>
- Camargo, R. (19 de setembro de 2011). *Billcode*. Acesso em 7 de julho de 2014, disponível em <http://blog.billcode.com.br/2011/09/repositorios-maven-em-5-minutos.html>
- Carmisini, A., & Vahldick, A. (2012). COMPARATIVO ENTRE FRAMEWORKS DE JAVASERVER FACES: APACHE TOBAGO, PRIMEFACES E RICHFACES. *REAVI: Revista Eletronica do Alto Vale do Itajaí*, 10-18.
- Eduardo, C. (2014). *ORM : Object Relational Mapper*. Acesso em 31 de junho de 2014, disponível em DEVMEDIA: <http://www.devmedia.com.br/orm-object-relational-mapper/19056>

Foundation, Apache Software. (2014). Acesso em 20 de junho de 2014, disponível em Apache Maven Project: <http://maven.apache.org/guides/mini/guide-proxies.html>

Foundation, T. A. (2014). Acesso em 26 de junho de 2014, disponível em Site de Apache Maven: <http://tomcat.apache.org/>

Gonçalves, E. (09 de fevereiro de 2010). *Conhecendo o comportamento do JavaServer Faces*. Acesso em 23 de junho de 2014, disponível em Tecnologia e Conhecimento ao Alcance de Todos : <http://www.edsongoncalves.com.br/category/jaserver-faces-2-0/>

Lohn, S., & Vahldick, A. (2013). FORMBUILDER: FRAMEWORK PARA GERAÇÃO AUTOMÁTICA DE TELAS PRIMEFACES COM BASE EM ANOTAÇÕES. *Revista Eletrônica do Vale do Itajaí* , 2 (1), 64-74.

Luckow, D. H., & Melo, A. A. (2010). *Programação Java para Web: Aprenda a desenvolver uma aplicação financeira pessoal com as ferramentas mais modernas da plataforma Java*. São Paulo: Novatec.

Medeiros, H. (2012). *Introdução à JPA - Java Persistence API*. Acesso em 02 de julho de 2014, disponível em DEVMEDIA: <http://www.devmedia.com.br/busca/?txtsearch=JPA>

Neves, T. J. (11 de janeiro de 2014). Ciclo de vida do JSF. *Ciclo de vida do JSF* .

Oracle. (10 de fevereiro de 2011). Acesso em 18 de junho de 2014, disponível em Oracle: <http://www.oracle.com/>

ORACLE. (2014). *JavaServer Faces API 2.2*. Acesso em 20 de junho de 2014, disponível em JAVA NET: <https://jaserverfaces.java.net/>

PrimeTek. (2014). *PrimeFaces*. Acesso em 27 de junho de 2014, disponível em Site da PrimeFaces: <http://www.primefaces.org/>

PrimeTek, PrimeFaces. (2014). *MOBILE SHOWCASE*. Acesso em 01 de juho de 2014, disponível em PrimeFaces: <http://www.primefaces.org/showcase/mobile/index.xhtml>

Rodrigues, F. (2014). *MvnRepository*. Acesso em 01 de julho de 2014, disponível em MvnRepository: <http://mvnrepository.com/>

Santos, W. F. (2008). Introdução ao Apache Maven. *Introdução ao Apache Maven* .

Sonatype. (2008). *Maven: The Complete Reference*. Acesso em 29 de junho de 2014, disponível em Sonatype: <http://www.sonatype.com/resources/books>

Souza, G. T., Pires, C. G., & Barros, M. O. (2003). PATI-MVC: Padrões MVC para Sistemas de Informação.

SUTIC. (05 de março de 2009). *Superintendência de Tecnologia da Informação e Comunicação*. Acesso em 01 de julho de 2014, disponível em <http://www2.ufersa.edu.br/portal/superintendencias/sutic>

UFERSA. (21 de janeiro de 2014). *CIência e Tecnologia: Inscrições de Disciplinas Eletivas*. Acesso em 22 de julho de 2014, disponível em [http://www2.ufersa.edu.br/portal/cursos/graduacao/ciencia\\_e\\_tecnologia/3753](http://www2.ufersa.edu.br/portal/cursos/graduacao/ciencia_e_tecnologia/3753)

## **ANEXO A**

O edital a baixo encontra-se d'oi dispon'ivel em julho de 2014 no site da UFERSA



SERVIÇO PÚBLICO FEDERAL

MINISTÉRIO DA EDUCAÇÃO

UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO

### **EDITAL DE SELEÇÃO PARA PREENCHIMENTO DAS VAGAS DOS CURSOS DE GRADUAÇÃO DE SEGUNDO CICLO NO SEMESTRE LETIVO DE 2014.2**

A UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO – UFERSA, por intermédio da Pró-Reitoria de Graduação (PROGRAD), torna público, para conhecimento dos interessados, as normas gerais para preenchimento das vagas dos cursos de Graduação segundo ciclo, vinculados ao curso de primeiro ciclo Bacharelado em Ciência e Tecnologia, com base nos termos da Resolução do CONSEPE 001/2011, 17 de junho de 2011.

#### **1. DA REALIZAÇÃO**

1.1.A realização da seleção ficará a cargo da Coordenação do Curso de Bacharelado em Ciência e Tecnologia, localizado no Campus Leste da UFERSA, sala 15 do prédio do DCEN, na Avenida Francisco Mota 572, Bairro Costa e Silva, Mossoró – RN, telefone (84) 3317-8253.

1.2.Poderão participar do processo seletivo:

- a) Concluintes do Curso de Bacharelado em Ciência e Tecnologia da UFERSA no período 2014.1;
- b) Graduados do Curso de Bacharelado em Ciência e Tecnologia da UFERSA em períodos anteriores a 2014.1;

- c) Graduados em Cursos de Bacharelado em Ciência e Tecnologia de outras Instituições de Ensino Superior, desde que os cursos sejam devidamente reconhecidos.

1.3.Os resultados do processo seletivo serão válidos, unicamente, para ingresso no semestre 2014.2.

## 2. DAS VAGAS

2.1.As vagas para os cursos de segundo ciclo devem ser oferecidas seguindo as recomendações da Resolução do CONSEPE 001/2011, 17 de junho de 2011, a citar:

- a) Seleção pelo score: 70% das vagas de cada curso de segundo ciclo são oferecidas com acesso de acordo com a classificação dos estudantes pelo Índice de Medida de Afinidade (IMA). Sendo a distribuição em relação aos campi diretamente proporcional ao número de concluintes do semestre 2014.1.
- b) Seleção por prova: 30% das vagas de cada curso de segundo ciclo são distribuídas, juntamente com as vagas remanescentes da seleção pelo score, por seleção através de provas, pautadas por conteúdos de disciplinas obrigatórios do curso de Bacharelado em Ciência e Tecnologia – UFERSA.
- c) A seleção por provas está dispensada no caso do número de vagas for igual ou superior ao número de candidatos inscritos por curso.

2.2.As vagas disponíveis para o semestre 2014.2 são:

| CAMPUS MOSSORÓ                  | TURNOS  | VAGAS |
|---------------------------------|---------|-------|
| Engenharia Civil                | Diurno  | 30    |
| Engenharia de Energia           | Diurno  | 30    |
| Engenharia Mecânica             | Diurno  | 30    |
| Engenharia Química              | Diurno  | 30    |
| Engenharia Agrícola e Ambiental | Diurno  | 05    |
| Engenharia de Petróleo          | Noturno | 30    |
| Engenharia de Produção          | Noturno | 30    |

| CAMPUS ANGICOS         | TURNOS  | VAGAS |
|------------------------|---------|-------|
| Engenharia Civil       | Diurno  | 30    |
| Sistemas de Informação | Noturno | 05    |
| Engenharia de Produção | Noturno | 30    |

| CAMPUS CARAÚBAS               | TURNO   | VAGAS |
|-------------------------------|---------|-------|
| Engenharia Elétrica – Noturno | Diurno  | 30    |
| Engenharia Mecânica – Noturno | Noturno | 30    |
| Engenharia Civil – Diurno     | Noturno | 30    |

### 3. DA INSCRIÇÃO E SELEÇÃO

3.1.Os concluintes deverão fazer sua inscrição em formulário eletrônico disponibilizado no sítio [www.ufersa.edu.br](http://www.ufersa.edu.br) em período indicado no ANEXO I. O ALUNO QUE NÃO SE INSCREVER NESTE PROCESSO SELETIVO ESTARÁ IMPEDIDO DE CONCORRER A UMA DAS VAGAS DISPONIBILIZADAS POR ESTE EDITAL.

3.2.No ato da inscrição o aluno deverá preencher corretamente os dados solicitados e anexar seu histórico acadêmico. No caso dos formados do semestre 2014.1 do curso de Bacharelado em Ciência e Tecnologia, anexar o histórico mais recente. Os outros candidatos deverão anexar o histórico de conclusão de curso de Ciência e Tecnologia. As informações prestadas pelos candidatos são de responsabilidade do mesmo.

#### 3.3. A PRIMEIRA FASE

3.3.1. A primeira fase da seleção será realizada utilizando o RANKING (classificação por ordem decrescente dos estudantes pelo valor do IMA) do Bacharelado em Ciência e Tecnologia e terá como público-alvo os concluintes 2014.1 do curso de Bacharelado em Ciência e Tecnologia da UFERSA. Será usado para quantitativo de uso das vagas o que está disposto em 2.1.

3.3.2. A divulgação do RANKING do curso de Bacharelado em Ciência e Tecnologia da UFERSA será realizada apenas quando o número de inscritos for superior ao quantitativo de vagas de cada curso do segundo ciclo.

3.3.3. A coordenação de cada curso de Ciência e Tecnologia da UFERSA é responsável pelo cálculo do IMA de seus respectivos alunos, no qual deverá enviar por memorando eletrônico aos responsáveis por esta seleção até 24 (vinte quatro) horas antes da divulgação do RANKING em período indicado no ANEXO I deste edital.

#### 3.4. A SEGUNDA FASE

3.4.1. A segunda fase da seleção também terá como público-alvo os concluintes 2014.1 do curso de Bacharelado em Ciência e Tecnologia da UFERSA, e só ocorrerá para

as engenharias que tiverem mais candidatos interessados que o número de vagas disponível. Essa fase constará de prova específica, detalhada no item 4.

### **3.5. A TERCEIRA FASE**

- 3.5.1. Uma terceira fase (REOPÇÃO) da seleção será destinada aos candidatos não classificados na segunda fase. A esses candidatos será disponibilizada a oportunidade de requerimento de uma das vagas remanescentes em um dos cursos descritos em 2.2.
- 3.5.2. O candidato interessado em participar da terceira fase deverá requerer sua participação em formulário disponibilizado pela coordenação do curso de Ciência e Tecnologia do campus Mossoró em data indicada no ANEXO I. O CANDIDATO QUE NÃO SOLICITAR SUA INSCRIÇÃO NA REOPÇÃO EM PERÍODO ESTABELECIDO ESTARÁ IMPEDIDO DE CONCORRER A UMA DAS VAGAS DISPONIBILIZADAS POR ESTE EDITAL.
- 3.5.3. Para preenchimento das vagas disponibilizadas na terceira fase, o critério de classificação utilizado será por ordem decrescente do IRA do curso de Bacharelado em Ciência e Tecnologia do candidato.
- 3.5.4. A terceira fase está condicionada a existência de vagas remanescentes das fases anteriores.

### **3.6. A QUARTA FASE**

- 3.6.1. Depois de finalizada a terceira fase, caso ainda existam vagas remanescentes, essas serão disponibilizadas em uma nova fase de seleção para preenchimento. Essa quarta fase do processo terá como público-alvo graduados em Ciência e Tecnologia de outras universidades, ou pela UFERSA em períodos anteriores ao 2014.1. Para preenchimento das vagas disponibilizadas na quarta fase, o critério de classificação utilizado será por ordem decrescente do rendimento acadêmico do curso de Ciência e Tecnologia do candidato.
- 3.6.2. O candidato interessado em participar da quarta fase da seleção deverá requerer sua participação em formulário eletrônico disponibilizado no sítio [www.ufersa.edu.br](http://www.ufersa.edu.br) em período indicado no ANEXO I.

## **4. DA PROVA ESPECÍFICA**

- 4.1.A prova específica (decorrente da segunda fase) será composta por 35 questões de múltipla escolha, e seus conteúdos estarão em conformidade com os conteúdos

especificados no Anexo II deste Edital. A prova terá duração de 04 (quatro) horas e será aplicada em dia, local e horários especificados no Anexo I deste edital, no campus de Mossoró.

4.2.O candidato deverá comparecer ao local da realização das provas com antecedência mínima de 30 (trinta) minutos, portando cédula de identidade, ou outro documento oficial de identificação com foto, e caneta esferográfica preta ou azul. O candidato não poderá portar telefone celular, boné (ou qualquer tipo de chapéu) e óculos escuros.

4.3.Usaremos como critério de desempate para preenchimento das vagas na segunda fase o Índice de Medida de Afinidade (IMA) do candidato, previamente calculado para a obtenção do ranking das engenharias. Caso persista o empate, será usado o Índice de Rendimento Acadêmico (IRA) para fins de desempate.

4.4.Caso o candidato tenha alguma deficiência física, ou necessite de tratamento especial por alguma condição adversa, favor comunicar com justificativa a coordenação do curso de Bacharelado em Ciência e Tecnologia do campus Mossoró, num período mínimo de 48 horas antes da realização da prova. A solicitação com justificativa deverá ser enviada ao e-mail [secretariabct@ufersa.edu.br](mailto:secretariabct@ufersa.edu.br).

4.5.Será excluído do Processo Seletivo o candidato que:

- a) Ausentar-se da sala de realização da prova sem o consentimento dos membros da comissão organizadora; Ausentar-se antes de uma hora e meia de realização da prova;
- b) Tentar comunicar-se com outros candidatos durante a execução da prova;
- c) Utilizar, ou tentar utilizar de meios fraudulentos para obter aprovação;
- d) Não entregar o caderno de questões e a folha de gabarito ao fiscal antes de se ausentar da sala de aplicação da prova específica.

4.6. Caso o candidato chegue atrasado, este será impedido de realizar a prova específica.

4.7. Os candidatos que, por qualquer razão, não realizarem a prova específica serão excluídos desta etapa, porém, poderão concorrer naturalmente a terceira etapa (REOPÇÃO – especificada na 3.5 deste edital), condicionada a existência de vagas.

## **5. DOS RESULTADOS**

5.1.Ao final de cada uma das fases, será divulgada uma lista com os candidatos que obtiveram aprovação na etapa, e outra com as vagas remanescentes que serão disponibilizadas para as fases seguintes.

5.2.O resultado final deste processo será divulgado no sítio [www.ufersa.edu.br](http://www.ufersa.edu.br) em período indicado no ANEXO I.

5.3. O calendário de matrículas dos alunos que obtiverem êxito nesse certame consta no Anexo I deste edital. Lembrando que a matrícula institucional só poderá ser efetivada para candidatos que já tenham colado grau.

## **6. INFORMAÇÕES ADICIONAIS**

6.1.Os candidatos inscritos poderão recorrer à Coordenação do curso de Bacharelado em Ciência e Tecnologia da UFERSA contra os atos desta coordenação quanto à lista de inscritos aptos ao processo seletivo, no prazo descrito no Anexo I deste edital, através do email [secretariabct@ufersa.edu.br](mailto:secretariabct@ufersa.edu.br).

6.2.Caso seja necessário, a coordenação poderá divulgar uma segunda lista de aprovação após o resultado final, para o caso de candidatos de tenham desistido de suas vagas. A desistência é definida pela não realização da matrícula institucional.

6.3. Casos omissos serão avaliados pela Pró-reitoria de Graduação da UFERSA.

Mossoró – RN, julho de 2014.

Augusto Carlos Pavão  
Pró-reitor de Graduação da UFERSA

## ANEXO A.1

### CRONOGRAMA DE ATIVIDADES

| ATIVIDADE                                                                                                                                                                                                                                                    | DATAS                                                        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|
| Inscrições por formulário eletrônico dos candidatos concluintes do período 2014.1 do curso de Ciência e Tecnologia da UFERSA e dos candidatos concorrentes da QUARTA FASE.                                                                                   | 13 a 16/08/2014                                              |
| Divulgação do deferimento das inscrições.                                                                                                                                                                                                                    | Até 19/08/2014                                               |
| Recurso contra os deferimentos das inscrições.                                                                                                                                                                                                               | 20/08/2014                                                   |
| Divulgação do RANKING do Bacharelado em Ciência e Tecnologia por campi e RESULTADO DA PRIMEIRA FASE.                                                                                                                                                         | Até 27/08/2014                                               |
| Recurso contra o RESULTADO DA PRIMEIRA FASE.                                                                                                                                                                                                                 | Dia útil seguinte a divulgação do resultado da primeira fase |
| Prova específica – SEGUNDA FASE<br>Local e horário: Auditório do DCEN do campus Mossoró, início da prova às 8h00min; chegada dos candidatos: 7h30min;                                                                                                        | 01/09/2014                                                   |
| Resultado da prova específica e RESULTADO DA SEGUNDA FASE.                                                                                                                                                                                                   | Até 03/09/2014                                               |
| Recurso contra o RESULTADO DA SEGUNDA FASE.                                                                                                                                                                                                                  | Dia útil seguinte a divulgação do resultado da segunda fase  |
| Solicitação de REOPÇÃO para vagas remanescentes da segunda fase por formulário eletrônico – TERCEIRA FASE.                                                                                                                                                   | 01/09/2014                                                   |
| Divulgação do resultado da REOPÇÃO – RESULTADO DA TERCEIRA FASE.                                                                                                                                                                                             | Até 03/09/2014                                               |
| Divulgação do resultado da QUARTA FASE e DIVULGAÇÃO DO RESULTADO FINAL.                                                                                                                                                                                      | Até 03/09/2014                                               |
| PERÍODO DE MATRÍCULAS INSTITUCIONAL <sup>1</sup> PARA OS CANDIDATOS APROVADOS NESTE EDITAL DE SELEÇÃO.<br>Local: Departamento de Registro Escolar do campus onde a Engenharia será cursada;<br>Horário: de 07h30min às 11h30min, e das 13h30min às 17h30min. | 04 e 05/09/2014                                              |
| PERÍODO DE MATRÍCULAS NAS DISCIPLINAS <sup>2</sup> PARA OS CANDIDATOS APROVADOS NESTE EDITAL DE SELEÇÃO VIA SIGAA.                                                                                                                                           | 09 e 10/09/2014                                              |

---

<sup>1</sup> Na matrícula institucional o candidato aprovado apresentará toda a documentação necessária para um novo vínculo à UFERSA, no caso, o novo vínculo será como estudante do curso de engenharia ao qual o candidato foi aprovado;

<sup>2</sup> A solicitação de matrícula nas disciplinas deverá ser feita via SIGAA pelo próprio aluno, em período estabelecido pelo calendário acadêmico;

## ANEXO A.2

### CONTEÚDOS PROGRAMÁTICOS PARA PROVA ESPECÍFICA

**INFORMÁTICA:** Noções de hardware; Comandos e recursos de edição de texto; Comandos e recursos de planilha eletrônica; Algoritmos; Programação em C.

**FÍSICA:** Trabalho e energia cinética; Momento linear; Impulso e colisões; Aplicação das Leis de Newton; Mecânica dos Fluidos; Temperatura e calor; Propriedades Térmicas da Matéria; Carga elétrica e campo elétrico; Circuitos de corrente contínua.

**MATEMÁTICA:** Funções e gráficos; Limite e continuidade; Derivação; Integração; Séries e sequências; Curvas de nível; Gradiente; Equações diferenciais lineares ordinárias de variáveis separadas de primeira e segunda ordem; Determinação de raízes de funções pelo método de Newton-Raphson, método de Gauss-Seidel.

**ANÁLISE E EXPRESSÃO TEXTUAL:** Linguagem, texto e discurso; Tipos textuais; Gêneros textuais; Competências necessárias para leitura e produção de texto; Paragrafação; Textos acadêmicos; Mecanismo de coesão textual; Coerência textual.

**RESISTÊNCIA DOS MATERIAIS:** Tensão normal média em uma barra com carga axial. Tensão de cisalhamento; Tensão admissível; Deformação; Deformação elástica de um elemento submetido à carga axial; Princípio da superposição; Elemento com carga axial estaticamente indeterminado; Tensão térmica; Fórmula de torção; Transmissão de potência; Ângulo de torção; Deformação por flexão de um elemento reto; Fórmula da flexão; Cisalhamento em elementos retos; Fórmula do cisalhamento; Tensões de cisalhamento em vigas.

**AMBIENTE, ENERGIA E SOCIEDADE:** Desenvolvimento sustentável; Ecossistemas; Gestão ambiental; Impacto ambiental; Legislação ambiental.

**QUÍMICA:** Cinética química; Termodinâmica química; Equilíbrio químico; Eletroquímica; Corrosão e proteção.