

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS

DOUGLAS CASTRO TAUBE

**RECOMENDAÇÃO DE CALAGEM POR MEIO DE SISTEMA DE SUPORTE À  
DECISÃO**

TRABALHO DE DIPLOMAÇÃO

MEDIANEIRA

2016

DOUGLAS CASTRO TAUBE

**RECOMENDAÇÃO DE CALAGEM POR MEIO DE SISTEMA DE SUPORTE À  
DECISÃO**

Trabalho de Diplomação apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas – COADS – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Claudio Leones Bazzi.

Co-orientador: Prof. Paulo Lopes de Menezes.

MEDIANEIRA

2016



---

## TERMO DE APROVAÇÃO

### Recomendação de Calagem por meio de Sistema de Suporte à Decisão

Por

**Douglas Castro Taube**

Este Trabalho de Diplomação (TD) foi apresentado às 9:10 h do dia 18 de novembro de 2016 como requisito parcial para a obtenção do título de Tecnólogo no Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, *Campus* Medianeira. Os acadêmicos foram argüidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado com louvor e mérito.

---

Prof. Claudio Leones Bazzi  
UTFPR – *Campus* Medianeira  
(Orientador)

---

Prof. Paulo Lopes de Menezes  
UTFPR – *Campus* Medianeira  
(Convidado)

---

Prof. Ricardo Sobjak  
UTFPR – *Campus* Medianeira  
(Convidado)

---

Prof. Jorge Aikes Junior  
UTFPR – *Campus* Medianeira  
(Responsável pelas atividades de TCC)

## **AGRADECIMENTOS**

Agradeço primeiramente à Deus pelo dom da vida. Aos meus pais que sempre apoiaram meus estudos. Esse período de graduação na UTFPR foi para mim muito mais do que um curso superior, foi uma faculdade para a vida. Durante essa caminhada aprendi muito, conquistei amizades, aprendi lições, e tive grandes oportunidades como meu intercâmbio para os Estados Unidos.

Agradeço à minha namorada Gabriele pela compreensão e apoio nas horas difíceis, aos meus futuros sogros por todo o apoio, e à equipe do CCAA Missal pelo tempo concedido para que eu pudesse focar neste trabalho.

Aos amigos, colegas e professores, em especial Ricardo Sobjak por toda ajuda na implementação web deste trabalho. Agradeço também às pessoas que auxiliaram nas coletas de dados e trabalhos de campo em geral, para este e outros projetos desenvolvidos na UTFPR. E à todas as pessoas que de uma forma ou de outra contribuíram comigo nessa jornada, o meu sincero muito obrigado.

*The only way that the world can keep up with food production to the levels that are needed with a growing world population is by the improvement of science and technology, and with the right policies that permit the application of that science and technology. (BERLAUG, Norman, 2009).*

A única maneira que o mundo pode continuar com a produção de alimentos para os níveis que são necessários com uma crescente população mundial, é através da melhoria da ciência e tecnologia, e com as políticas certas que permitam a aplicação dessa ciência e tecnologia. (BERLAUG, Norman, 2009).

## RESUMO

TAUBE, Douglas C. Recomendação de calagem por meio de sistema de suporte à decisão. 2016. 56 f. Trabalho de Diplomação (Tecnologia em Análise e Desenvolvimento de Sistemas), Universidade Tecnológica Federal do Paraná. Medianeira, 2016.

O objetivo deste trabalho foi de desenvolver uma aplicação web para oferecer uma recomendação de calagem com sistema de suporte à decisão, que possa contribuir no gerenciamento de propriedades agrícolas. Baseando se em tecnologias de desenvolvimento de software, como a linguagem de programação Java, framework MVC VRaptor, Bootstrap e Sistema Gerenciador de Banco de Dados PostgreSQL para armazenamento de dados e gerenciamento de funções. A aplicação leva em consideração os dados do laudo de análise do solo para calcular recomendações de calagem, que pode contribuir para o gerenciamento de lavouras. Os frameworks mostraram-se como ferramentas ágeis de desenvolvimento, e o PostgreSQL como um bom sistema para manter e gerenciar dados. A aplicação fica disponibilizada gratuitamente à comunidade agrícola no endereço [ppat.md.utfpr.edu.br](http://ppat.md.utfpr.edu.br) trazendo benefícios ao gerenciamento agrícola.

**Palavras-chave:** Calagem. VRaptor. SGBD. PostgreSQL.

## ABSTRACT

TAUBE, Douglas C. Liming Recommendation through a decision support system. 2016. 56 f. Trabalho de Diplomação, FEDERAL UNIVERSITY OF TECHNOLOGY – PARANÁ. Medianeira 2016.

The aim of this study was to develop a web application to provide liming recommendations as a decision-making tool, in order to contribute on farm management. Based on software development technologies such as Java programming language, the MVC framework VRaptor, Bootstrap and PostgreSQL Database Management System for data storage and management of stored procedures. The application retrieves data from soil analysis reports to calculate liming recommendations, which can contribute in farm management. The frameworks proved to be agile development tools, and PostgreSQL as a good system to maintain and manage data. The web application is available for free at the link [ppat.md.utfpr.edu.br](http://ppat.md.utfpr.edu.br), benefiting the agricultural community and farming management as well.

**Keywords:** Liming. VRaptor. DBMS. PostgreSQL.

## LISTA DE SIGLAS

BD	<i>Banco de Dados</i>
CTC	<i>Capacidade de troca de cátions</i>
DAO	<i>Data-Access-Object</i>
DBMS	<i>Database Management System</i>
MER	<i>Modelo Entidade Relacionamento</i>
NC	<i>Necessidade de Calcário</i>
SGBD	<i>Sistema Gerenciador de Banco de Dados</i>
WEB	<i>World Wide Web</i>



## LISTA DE FIGURAS

Figura 1 – Exemplo de função utilizando a linguagem procedural PL/pgSQL.....	21
Figura 2 – Código do menu superior .....	23
Figura 3 – Código para incluir o conteúdo do menu.jsp na página list.jsp utilizando a <i>tag</i> JSTL <code>&lt;jsp:include&gt;</code> .....	24
Figura 4 – Arquivos do download do Bootstrap.....	25
Figura 5 – Arquitetura MVC .....	26
Figura 6 – <i>Tags</i> no arquivo pom.xml para configurar o VRaptor em um projeto.....	26
Figura 7 – Mapa da Área A .....	28
Figura 8 – Mapa da Área B .....	29
Figura 9 – Exemplo de um Laudo de Análise de Solo .....	30
Figura 10 – Arquitetura do projeto .....	31
Figura 11 – Estrutura do Projeto Web .....	33
Figura 12 – Estrutura do BD no ambiente pgAdmin III.....	34
Figura 13 – Função para calcular a NC pelo critério de saturação por bases.....	36
Figura 14 – Visão do arquivo persistence.xml .....	36
Figura 15 – Trecho de código da classe GenericDAO .....	37
Figura 16 – Trecho de código da classe modelo Produtor .....	38
Figura 17 – Código da classe ProdutorDAO .....	39
Figura 18 – Trecho de código da classe ProdutorController .....	40
Figura 19 – Trecho de código da página de listagem, com inclusão de bibliotecas.....	41
Figura 20 – Trecho do código para o corpo da página de listagem de produtores .....	42
Figura 21 – Trecho do código para o corpo da página de cadastro de produtores .....	43
Figura 22 – Tela inicial da aplicação Fonte: Autoria própria.....	43
Figura 23 – Tela de listagem de produtores .....	44
Figura 24 – Tela de cadastro de produtores.....	45
Figura 25 – Tela da listagem resumida dos dados do laudo .....	46
Figura 26 – Tela da listagem estendida dos dados do laudo .....	46
Figura 27 – Tela de cadastro de laudos .....	47
Figura 28 – Tela inicial de recomendações .....	48
Figura 29 – Tela de recomendações com filtros.....	49

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>9</b>
1.1	OBJETIVO GERAL.....	10
1.2	OBJETIVOS ESPECÍFICOS .....	10
1.3	JUSTIFICATIVA .....	10
1.4	ESTRUTURA DO TRABALHO .....	12
<b>2</b>	<b>REFERENCIAL TEÓRICO .....</b>	<b>13</b>
2.1	LEVANTAMENTO DE DADOS QUÍMICOS DO SOLO .....	13
2.1.1	Ph referência .....	13
2.1.2	Capacidade de troca de cátions (CTC) .....	15
2.1.3	Saturação da CTCefetiva .....	15
2.1.4	Saturação da CTCpH 7,0 .....	15
2.1.5	Critério da Saturação por bases .....	16
2.1.6	Critério para solos pouco tamponados.....	17
2.2	BANCO DE DADOS RELACIONAL.....	17
2.3	SISTEMAS GERENCIADORES DE BANCO DE DADOS .....	18
2.3.1	PostgreSQL.....	18
2.3.2	PL/pgSQL.....	20
2.3.3	PgAdmin .....	21
2.4	APLICAÇÃO WEB.....	22
2.5	PLATAFORMA JAVA ENTERPRISE EDITION .....	22
2.6	BIBLIOTECA JSTL.....	23
2.7	FRAMEWORK BOOTSTRAP .....	24
2.8	PADRÃO MVC.....	25
2.9	FRAMEWORK VRAPTOR.....	26
<b>3</b>	<b>MATERIAIS E MÉTODOS .....</b>	<b>27</b>
3.1	TECNOLOGIAS UTILIZADAS.....	27
3.2	DADOS UTILIZADOS E DESENVOLVIMENTO NO BANCO DE DADOS .....	28
3.3	LAUDE DE ANÁLISE DE SOLO.....	29
3.4	ANÁLISE DO SOFTWARE .....	30
3.4.1	Visão Geral do Sistema .....	30

3.5	ARQUITETURA DO PROJETO .....	31
3.5.1	Arquitetura do Projeto Web .....	31
3.5.2	Arquitetura da Aplicação no Banco de Dados .....	33
<b>4</b>	<b>RESULTADOS E DISCUSSÃO .....</b>	<b>35</b>
4.1	FUNÇÕES EM PL/PGSQL .....	35
4.2	APLICAÇÃO WEB DESENVOLVIDA .....	36
4.2.1	Desenvolvimento de um DAO genérico .....	36
4.2.2	Desenvolvimento de uma classe modelo .....	37
4.2.3	Apresentação das telas da Aplicação Web .....	43
4.2.4	Exibição de um laudo na Aplicação Web .....	45
4.2.5	Listagem estendida do laudo .....	46
4.2.6	Cadastro de um laudo na Aplicação .....	47
4.2.7	Exibição das recomendações de calagem .....	47
4.2.8	Exibição inicial sem filtros .....	48
4.2.9	Exibição com filtros .....	48
<b>5</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>50</b>
5.1	CONCLUSÃO .....	50
5.2	TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO .....	50
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>51</b>
	<b>APÊNDICE A – Requisitos do Sistema .....</b>	<b>54</b>

## 1 INTRODUÇÃO

O solo é um dos recursos que mais influenciam na produção agrícola, tendo em vista que investimentos em insumos e acompanhamento se perdem quando há desequilíbrio na distribuição e disponibilidade de seus elementos. Esta situação problemática até pode ser amenizada com aplicação de fertilizantes e corretivos para o solo, porém um grande fator aliado, muitas vezes esquecido é o pH do solo, que precisa atingir um nível de referência a fim de que as plantas possam aproveitar os macros e micronutrientes disponíveis no solo. Para realizar tal função, se faz necessário a aplicação de calcário, denominada calagem para regular o pH do solo. Segundo Caires e Joris (2016), a calagem reduz ou elimina os efeitos tóxicos de alumínio (Al) e manganês (Mn), aumenta a disponibilidade de nitrogênio (N), fósforo (P), cálcio (Ca), magnésio (Mg), enxofre (S) e molibdênio (Mo) no solo e também aumenta a atividade microbiana. Frustrações de safra acabam por ser comuns em todo o mundo, mesmo em anos com boas condições climáticas, devido a deficiência de calcário, que impossibilita as plantas de absorverem esses elementos no solo, prejudicando tanto o produtor como toda a logística e os aspectos econômicos relacionados.

Diante disso, verifica-se que manter uma terra agrícola com o pH referência exigido pela cultura a ser cultivada, é um fator importantíssimo para se obter bons resultados ano a ano e manter o solo homogêneo, fazendo o manejo da calagem sempre que necessário. Para manter esse equilíbrio se faz uso de um indicador de necessidade de calcário (NC) que pode ser determinado a partir do resultado do laudo de análise do solo.

Considerando a necessidade de calcular a NC, baseada em fórmulas matemáticas de metodologias científicas já desenvolvidas, o objetivo deste trabalho corresponde em desenvolver um modelo computacional para isso, visando à elaboração de um sistema de suporte a decisão. A partir de dados da análise de solo, fazendo uso de dados coletados em uma propriedade agrícola de 19,8 ha, localizada no município de Serranópolis do Iguaçu, nos anos de 2012 a 2016.

A partir do modelo proposto, foi desenvolvida uma aplicação, que possibilita ao produtor inserir dados das análises de solo, a fim de gerar uma recomendação da quantidade necessária para a correção dos níveis de calcário no solo, levando em conta as informações sobre disponibilidade de macro e micronutrientes no solo tais como: Ca, Mg, Mo, P, K, Cu, Zn, Fe, Mn, Al, H+Al, o pH do solo, o pH de equilíbrio de uma solução tamponada a pH 7,5

quando em contato com o solo (índice SMP), e a porcentagem dos valores em relação ao valor da capacidade de troca de cátions (CTC), providas pelo laudo de análise do solo.

### 1.1 OBJETIVO GERAL

Elaborar uma aplicação de suporte a decisão para cálculo de calagem, considerando dados químicos provenientes do laudo de análise do solo, a fim de fornecer uma recomendação da necessidade de calcário em uma terra agrícola.

### 1.2 OBJETIVOS ESPECÍFICOS

1. Construir uma aplicação em ambiente web que permita dar suporte à tomada de decisões para recomendação de calagem;
2. Validar a aplicação com um caso real;
3. Disponibilizar a aplicação gratuitamente à comunidade agrícola.

### 1.3 JUSTIFICATIVA

A atividade agrícola demanda de altos investimentos para retornar bons resultados de produtividade e retorno financeiro, e tendo isso em vista, a manutenção, correção e calagem do solo são práticas indispensáveis, em especial esta última, necessária para manter os nutrientes disponíveis para as plantas. Práticas essas que exigem a atenção do produtor, que precisa estar anualmente informado sobre as condições do solo, a fim de amenizar os riscos à sua produção e melhorar a gestão agrícola. De acordo com a Comissão de Química e Fertilidade do Solo - RS/SC (2004), “a acidificação do solo cultivado é um processo contínuo e sua intensidade depende de vários fatores. A utilização de corretivos da acidez do solo é, portanto, de grande importância para a produção agrícola”. Considerando isso, verifica-se que

diante dessas situações, o produtor rural precisa ser rápido na sua tomada de decisões, a fim de garantir maior segurança em seus investimentos futuros.

Desse modo, é necessário que o produtor faça anualmente coletas de solo para serem analisadas. A partir delas o produtor tem em mãos o laudo de análise de solo, contendo as informações atualizadas sobre a condição e disponibilidade dos macros e micronutrientes no solo. Com tais resultados é possível calcular, baseando-se em métodos e fórmulas aplicadas, a necessidade de calcário (NC), aplicado para elevar o pH do solo ao nível desejado pela cultura a ser cultivada nos próximos meses. Esse procedimento de aplicação de calcário é denominado calagem. Segundo Caires e Joris (2016) “A calagem pode ser considerada a prática que mais contribui para o aumento da eficiência do uso de fertilizantes”. Os autores Lopes, Silva e Guilherme (1991) complementam: “A calagem é considerada como uma das práticas que mais contribui para o aumento da eficiência dos adubos e, conseqüentemente, da produtividade e da rentabilidade agropecuária”. Quando há deficiência de calcário, o pH do solo fica abaixo do valor necessário para a cultura, impossibilitando a planta de se nutrir dos elementos disponíveis no solo. Com esta situação problemática, investimentos em adubos e fertilizantes acabam se perdendo, pois ficam impedidos de serem aproveitados pela cultura.

A necessidade que o produtor rural seja rápido na sua tomada de decisões pode ser auxiliada por meio das tecnologias de informação, utilizando-se de ferramentas computacionais que podem avaliar resultados, processar dados e efetuar cálculos de maneira muito mais rápida do que uma pessoa. Diversas vantagens podem ser apontadas quando aplicando tais soluções, como o aumento da segurança, eficiência, rapidez, agilidade, confiabilidade e fornecimento de informações em tempo real (CARVALHO, 2008). No mercado brasileiro existem poucos softwares voltados para a área de fertilidade do solo, e os que existem são comercializados com preços elevados, restringindo o acesso aos produtores, portanto há uma carência de softwares gratuitos (SILVA, 2012). Dentro desses aspectos, pode-se considerar que uma recomendação de calagem com sistema de suporte à decisão seria de grande contribuição no gerenciamento de propriedades agrícolas, também levando em conta que a agricultura é uma das principais atividades econômicas da região Oeste do Paraná. De acordo com o IPARDES (2003, p. 69) a mesorregião Oeste, dentre as regiões do Estado do Paraná, é talvez aquela na qual melhor se visualiza o processo de desenvolvimento tecnológico na produção agropecuária. A organização dos segmentos agroindustriais, e principalmente a organização de cooperativas e as importações primárias definiram o modelo econômico regional e a articulação do mesmo às economias estadual, nacional e mundial.

Deste modo, considerando os aspectos econômicos da região Oeste do Paraná, se justifica a escolha desse tema para este projeto que objetiva contribuir na gestão agrícola oferecendo uma recomendação de calagem aos seus usuários, bem como estimular o produtor rural a realizar anualmente a coleta de solo para análise, que possivelmente contribuirá no gerenciamento de suas lavouras.

#### 1.4 ESTRUTURA DO TRABALHO

O trabalho está dividido em nos seguintes capítulos: Introdução; Referencial Teórico; Materiais e Métodos; Resultados e discussão e Conclusões.

No referencial teórico estão abordados as tecnologias e estudos relevantes ao entendimento da aplicação.

Na seção Materiais e Métodos está descrito o que foi utilizado na construção da aplicação e uma análise simplificada.

No capítulo Desenvolvimento está descrito o sistema concluído, bem como uma abordagem ao mesmo em funcionamento.

## 2 REFERENCIAL TEÓRICO

### 2.1 LEVANTAMENTO DE DADOS QUÍMICOS DO SOLO

Para se gerar uma recomendação para tomada de decisão é necessário primeiramente o levantamento de dados. Para o caso proposto neste trabalho, os dados foram fornecidos pelo laudo de análise do solo, obtido após as coletas de solo. De acordo com Carvalho (2008), na administração de uma propriedade, é muito importante que o produtor tenha em mãos os dados necessários para definir a situação econômico-financeira e realizar um gerenciamento profissional de seu empreendimento.

Alguns trabalhos similares foram desenvolvidos na temática que prima pelo desenvolvimento de novas tecnologias para a recomendação de calagem. O autor Cerri (2001), desenvolveu um sistema para a aplicação localizada de calcário a taxas variáveis. O autor Silva (2012) desenvolveu um trabalho ainda mais similar ao presente, focando nas recomendações de adubação e calagem para o tomateiro no estado de Minas Gerais, e, o autor Carvalho (2008) semelhantemente desenvolveu com foco na cultura do Cafeeiro em produção.

Na literatura existem vários estudos que desenvolveram métodos utilizados para recomendação de calagem. De acordo com a Comissão de Química e Fertilidade do Solo - RS/SC (2004) “os principais critérios de recomendação de calagem são o pH do solo de referência da cultura e o percentual de saturação da Capacidade de troca de cátions (CTC) por cátions trocáveis de reação básica ( $\text{Ca}^{2+}$ ,  $\text{Mg}^{2+}$ ,  $\text{K}^+$  e  $\text{Na}^+$ ) ou de saturação da CTCefetiva por  $\text{Al}^{3+}$ ”.

#### 2.1.1 Ph referência

A Comissão de Química e Fertilidade do Solo - RS/SC (2004), define o pH referência como “o valor do pH do solo mais adequado ao desenvolvimento das culturas”, e ressaltam que com um pH acima do pH referência, não é observada resposta da



cultura à calagem. Também destacam que a maioria das culturas de grãos possuem o pH referência 6,0, exceto o arroz irrigado.

Para se definir a quantidade de corretivo a ser aplicado é necessário que o pH em água atinja o valor referência. Essa quantidade aumenta com a acidez potencial do solo expressa pelo índice SMP proposto por Shoemaker, McLean e Pratt (1961). A acidez potencial do solo diminui à medida que o índice SMP aumenta. Na Tabela 1 são apresentadas as quantidades de calcário em toneladas por hectare (t/ha) necessárias para atingir o pH referência de valores 5,5, 6,0 e 6,5. Ressalta-se que o corretivo deve possuir o indicador de qualidade de calcário, denominado poder relativo de neutralização total (PRNT) no valor 100% (COMISSÃO DE QUÍMICA E FERTILIDADE DO SOLO - RS/SC, 2004).

**Tabela 1 - Quantidades de calcário necessárias para elevar o pH em água do solo a 5,5, 6,0 e 6,5, estimadas pelo índice SMP.**

Índice SMP	pH desejado		
	5,5	6,0	6,5
	t/ha		
4,4	15,0	21,0	29,0
4,5	12,5	17,3	24,0
4,6	10,9	15,1	20,0
4,7	9,6	13,3	17,5
4,8	8,5	11,9	15,7
4,9	7,7	10,7	14,2
5,0	6,6	9,9	13,3
5,1	6,0	9,1	12,3
5,2	5,3	8,3	11,3
5,3	4,8	7,5	10,4
5,4	4,2	6,8	9,5
5,5	3,7	6,1	8,6
5,6	3,2	5,4	7,8
5,7	2,8	4,8	7,0
5,8	2,3	4,2	6,3
5,9	2,0	3,7	5,6
6,0	1,6	3,2	4,9
6,1	1,3	2,7	4,3
6,2	1,0	2,2	3,7
6,3	0,8	1,8	3,1
6,4	0,6	1,4	2,6
6,5	0,4	1,1	2,1
6,6	0,2	0,8	1,6
6,7	0	0,5	1,2
6,8	0	0,3	0,8
6,9	0	0,2	0,5
7,0	0	0	0,2
7,0	0	0	0

**Fonte: Quantidades necessárias de calcário para elevar o pH do solo a 5,5, 6,0 e 6,5 pelo índice SMP. Adaptado de (COMISSÃO DE QUÍMICA E FERTILIDADE DO SOLO - RS/SC, 2004).**

### 2.1.2 Capacidade de troca de cátions (CTC)

Segundo Van Raij (1969), “a CTC indica a quantidade de íons positivos que um solo é capaz de reter em determinadas condições e permutar por quantidades estequiometricamente equivalentes de outros íons do mesmo sinal”. A Capacidade de troca de cátions (CTC), é calculada pela soma dos cátions de reação básica trocáveis ( $K^+$ ,  $Ca^{2+}$ ,  $Mg^{2+}$ , e às vezes  $Na^+$ ) e dos cátions ácidos ( $H^+$  +  $Al^{3+}$ ) (COMISSÃO DE QUÍMICA E FERTILIDADE DO SOLO - RS/SC, 2004).

### 2.1.3 Saturação da $CTC_{efetiva}$

De acordo com a Comissão de Química e Fertilidade do Solo - RS/SC (2004), para o cálculo da CTC ao pH natural do solo (na análise), denominada CTC efetiva, é somado o cátion  $Al^{3+}$  aos cátions de reação básica, conforme a equação (1):

$$CTC_{efetiva} = Ca^{2+} + Mg^{2+} + K^+ + Na^+ + Al^{3+} \quad (1)$$

O valor da  $CTC_{efetiva}$  é utilizado para os cálculos da saturação por alumínio, obtidos conforme a equação (2):

$$m = \frac{Al \cdot 100}{CTC_{efetiva}} \quad (2)$$

Onde  $m$  é a saturação por Al.

### 2.1.4 Saturação da $CTC_{pH\ 7,0}$

A CTC a pH 7,0 é calculada pela equação (3):

$$CTC_{pH\ 7,0} = Ca^{2+} + Mg^{2+} + K^+ + Na^+ + Al^{3+} + (H^+ + Al^{3+}) \quad (3)$$

A Comissão de Química e Fertilidade do Solo - RS/SC (2004) destaca que é importante lembrar que o teor de  $Na^+$  nos solos ácidos em geral é baixo, e normalmente não é incluído no cálculo. O valor da  $CTC_{pH\ 7,0}$  é utilizado para os cálculos da saturação por bases.

A fração da CTC (calculada a pH 7,0) ocupada pelos cátions de reação básica representa a porcentagem das cargas negativas do solo neutralizadas por cátions de reação básica, denominada saturação por bases (valor V) (COMISSÃO DE QUÍMICA E FERTILIDADE DO SOLO - RS/SC, 2004), obtida conforme a equação (4):

$$V = \frac{S \cdot 100}{CTC_{pH\ 7,0}} \quad (4)$$

Onde S = soma dos cátions de reação básica ( $Ca^{2+} + Mg^{2+} + K^+ + Na^+$ ) em  $cmol_c/dm^3$  e V = saturação por bases.

#### 2.1.5 Critério da Saturação por bases

Segundo a Comissão de Química e Fertilidade do Solo - RS/SC (2004), a indicação da quantidade de calcário a utilizar pode ser feita pela saturação da  $CTC_{pH\ 7,0}$  por bases, conforme a equação (5):

$$NC = \frac{CTC(V2 - V1)}{100} \quad (5)$$

Onde NC é a necessidade de calcário, em t/ha (com PRNT 100%); V2 é a porcentagem de saturação por bases desejada; e, V1 é a porcentagem da saturação por bases do solo, fornecida no laudo de análise. A Comissão de Química e Fertilidade do Solo - RS/SC

(2004) esclarece que nos solos dos estados do Rio Grande do Sul e Santa Catarina, normalmente as porcentagens de saturação da CTC por bases de 65%, 80% e 85% correspondem aos valores de pH em água de 5,5, 6,0 e 6,5 respectivamente.

Segundo a Comissão de Química e Fertilidade do Solo - RS/SC (2004), o critério da saturação por bases é bastante utilizado em lavouras com rotações de culturas, particularmente no sistema plantio direto, sendo estas práticas conservacionistas as mais utilizadas na região oeste do Paraná (CASÃO JUNIOR; ARAÚJO; LLANILLO, 2012).

#### 2.1.6 Critério para solos pouco tamponados

Segundo a Comissão de Química e Fertilidade do Solo - RS/SC (2004) o índice SMP pode subestimar a necessidade de calcário em solos pouco tamponados. Nesses casos, a NC pode ser calculada pelos teores de matéria orgânica e de alumínio trocável do solo, conforme as equações (6), (7) e (8), dependendo do pH referência:

$$NC_{pH5,5} = -0,653 + 0,480MO + 1,937Al \quad (6)$$

$$NC_{pH6,0} = -0,516 + 0,805MO + 2,435Al \quad (7)$$

$$NC_{pH6,5} = -0,122 + 1,193MO + 2,713Al \quad (8)$$

Em que: NC é a necessidade de calcário em t/ha, (com PRNT 100%); MO é o teor de matéria orgânica (em %); e Al é o teor de alumínio trocável do solo (em  $\text{cmol}_c/\text{dm}^3$ ).

## 2.2 BANCO DE DADOS RELACIONAL

Segundo Ritchie (2002) “a invenção do banco de dados relacional é em grande parte atribuída a Edward Codd, que apresentou a ideia em um artigo seminal escrito em 1970”. De acordo com Coronel, Moris e Rob (2011) o modelo de dados relacional é implementado através de um Sistema Gerenciador de Banco de Dados (SGBD) sofisticado, que executa as mesmas funções básicas fornecidas pelos SGBDs hierárquicos e de rede, além de uma série de outras funções que tornam o modelo de dados relacional mais fácil de entender e implementar.

Uma das técnicas utilizadas para modelagem de um banco de dados (BD) é o modelo Entidade-Relacionamento. A Modelagem Entidade-Relacionamento (ER) foi introduzida por Peter Chen em 1976 e ainda é o método mais amplamente utilizado para a modelagem de dados. Ela retrata o mundo em termos de entidades que possuem atributos e participam de relacionamentos (HALPIN; MORGAN, 2008). Utilizar um banco de dados relacional é uma das bases desse projeto, pois a aplicação irá manipular dados, e estes precisam ser armazenados em uma base de dados. Para diagramar e facilitar a compreensão humana desses dados é utilizado a Modelagem entidade relacionamento.

## 2.3 SISTEMAS GERENCIADORES DE BANCO DE DADOS

Sistemas Gerenciadores de Banco de Dados (SGBDs), do inglês DataBase Management System (DBMS), são definidos por Pressman (2002) como “uma coleção de informação grande e organizada, a que se tem acesso por intermédio do software”. Um SGBD consiste de uma coleção de dados inter-relacionados e um conjunto de programas para acessá-los. O principal objetivo de um SGBD é proporcionar um ambiente que é conveniente e eficiente para uso na recuperação e armazenamento de informações de banco de dados (KEDAR, 2009). Em suma, Coronel, Moris e Rob (2011) esclarecem: “é um conjunto de programas que administra a estrutura do banco de dados e controla o acesso aos dados armazenados no banco de dados”.

### 2.3.1 PostgreSQL

O PostgreSQL é um SGBD objeto-relacional gratuito e de código fonte aberto, utilizado para armazenar informações de soluções de informática em todas as áreas de negócios existentes, bem como administrar o acesso a estas informações (MILANI, 2008). Baseado no POSTGRES, versão 4.2, desenvolvido na Universidade da Califórnia em Berkeley - Departamento de Ciência da Computação, o POSTGRES foi pioneiro em muitos conceitos que só se tornaram disponíveis em alguns sistemas de banco de dados comerciais mais tarde (THE POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2016). Ele fornece serviços de nível corporativo, incluindo um alto desempenho e escalabilidade (JUBA; VANNAHME; VOLKOV, 2015). Está em desenvolvimento ativo desde 1995, com uma arquitetura de confiabilidade, integridade de dados e conformidade a padrões (COMUNIDADE BRASILEIRA DE POSTGRESQL, 2016).

Na versão 9.6.0 (12 de outubro de 2016), o PostgreSQL apresenta recursos avançados, e é executável nos sistemas operacionais Unix, GNU/Linux e Microsoft Windows. Segundo Juba, Vannahme e Volkov (2015), o PostgreSQL possui vantagens também para os desenvolvedores. Entre as principais pode-se citar:

- Lançamento de uma versão principal a quase todos os anos;
- Boa documentação e uma comunidade ativa permite que os desenvolvedores encontrem e resolvam problemas rapidamente. O manual do PostgreSQL possui mais de 2.500 páginas;
- Rico repositório de extensões permite aos desenvolvedores concentrar-se na lógica de negócios. Além disso, ele permite que os desenvolvedores facilmente atendam às mudanças de requisitos;
- O código fonte está disponível gratuitamente;
- Tarefas de administração de banco de dados não exigem muito tempo, e podem ser automatizadas;
- O PostgreSQL pode ser integrado facilmente com outro SGBD, fornecendo uma boa flexibilidade à arquitetura de software.

Para manipular dados geográficos, é necessário um método de manipulação especial, pois estes são representados por meio de pontos, linhas ou polígonos. Para tal, o PostgreSQL oferece suporte à extensão espacial PostGIS que permite que objetos GIS (*Geographic Information Systems*) sejam armazenados no banco de dados (THE POSTGIS DEVELOPMENT GROUP, 2016).

### 2.3.2 PL/pgSQL

O PostgreSQL suporta várias linguagens procedurais, porém ele possui a sua própria linguagem procedural: o PL/pgSQL. PL significa linguagem procedural, e pgSQL é a forma abreviada para PostgreSQL (DAR, et al., 2015). Esta linguagem procedural, influenciada pelo PL/SQL da Oracle, possui estruturas de controle e plena integração com triggers do PostgreSQL, índices, regras, tipos de dados definidos pelo usuário e objetos operadores. (JUBA; VANNAHME; VOLKOV, 2015). Os autores Dar, et al., (2015) e Douglas e Douglas, (2005) destacam uma das principais diferenças do PL/pgSQL do SQL básico, onde o primeiro inclui elementos procedurais como comandos if/then else e loops, porém na utilização do PL/pgSQL quaisquer comandos SQL podem ser incluídos.

De acordo com a documentação do PostgreSQL (THE POSTGRES GLOBAL DEVELOPMENT GROUP, 2016), com PL/pgSQL é possível agrupar um bloco de comandos e uma série de consultas dentro do servidor de banco de dados, tendo assim o poder de uma linguagem procedural e a facilidade de uso do SQL, mas com considerável economia de sobrecarga de comunicação cliente/servidor. A documentação aponta os objetivos para os quais o PL/pgSQL foi desenvolvido:

- Ser usado para criar funções e procedimentos de funções e triggers;
- Adicionar estruturas de controle à linguagem SQL;
- Executar processamentos complexos;
- Herdar todos os tipos de dados, funções e operadores definidos pelo usuário;
- Ser confiável para o servidor;
- Ser fácil de usar.

Na Figura 1 é possível observar um exemplo de função utilizando o PL/pgSQL para executar a função que determina a NC através do critério para solos pouco tamponados descrito no item 2.1.6.

```

1 CREATE OR REPLACE FUNCTION f_solos_pouco_tamponados2(
2     p_lauid integer,
3     p_ph double precision)
4     RETURNS double precision AS
5     $BODY$
6     DECLARE
7         v_NC float;
8         v_MO float;
9         v_Al float;
10    BEGIN
11        EXECUTE 'select tb_laudo.mo, tb_laudo.al from tb_laudo where tb_laudo.id = ' || p_lauid
12        INTO v_MO, v_Al;
13
14        IF p_ph = 5.5 THEN
15            v_NC := -0.653 + 0.480*v_MO + 1.937*v_Al;
16        ELSIF p_ph = 6.0 THEN
17            v_NC := -0.516 + (0.805*v_MO) + (2.435*v_Al);
18        ELSIF p_ph = 6.5 THEN
19            v_NC := -0.122 + 0.193*v_MO + 2.713*v_Al;
20        ELSIF p_ph <> 5.5 OR p_ph <> 6.0 or p_ph <> 6.5 THEN
21            raise notice 'ph deve ser 5.5 ou 6.0 ou 6.5';
22            v_NC := NULL;
23        END IF;
24        RETURN v_NC;
25    END;
26    $BODY$
27    LANGUAGE plpgsql VOLATILE
28    COST 100;

```

**Figura 1 – Exemplo de função utilizando a linguagem procedural PL/pgSQL**  
**Fonte: Autoria própria.**

### 2.3.3 PgAdmin

O pgAdmin (pgAdmin III), é uma ferramenta de código fonte aberto e multiplataforma. Segundo Juba, Vannahme e Volkov (2015) o pgAdmin é uma boa ferramenta GUI para administração e desenvolvimento no PostgreSQL. Está disponível em várias línguas e suporta a maioria da codificação de dados PostgreSQL. Os autores Obe e Hsu (2014) apontam uma das grandes vantagens dessa ferramenta: “O pgAdmin é executado no ambiente de trabalho e pode se conectar a vários servidores PostgreSQL independentemente da versão ou sistema operacional”.

Na versão 1.22.1 (12 de outubro de 2016), o pgAdmin pode ser baixado separadamente do PostgreSQL, se ele já não estiver incluso no seu instalador (OBE e HSU, 2014). No instalador do PostgreSQL para o Windows, o pgAdmin é instalado por padrão. Os autores Obe e Hsu (2014) também esclarecem que o pgAdmin é a ferramenta de administração gráfica oficial do PostgreSQL, que tem de estar sempre em sincronia com as últimas versões deste SGBD, e também recomendam aos iniciantes que comecem com esta ferramenta antes de explorarem outras. “Se uma nova versão do PostgreSQL introduzir novas



funcionalidades, é possível contar com a versão mais recente do pgAdmin para auxiliar na utilização” (OBE e HSU, 2014).

## 2.4 APLICAÇÃO WEB

Segundo Nations (2016), “uma aplicação web é qualquer aplicação que usa um navegador web como cliente”. Uma aplicação web, para Shklar e Rosen (2004), é “a aplicação cliente/servidor que usa um navegador Web como seu programa cliente e executa um serviço interativo conectando-se com os servidores através da Internet (ou Intranet)”. Para Pressman (2011), as aplicações Web, também chamadas de WebApps, abarcam uma vasta gama de aplicações. Com o aparecimento da Web 2.0, elas têm evoluído e se transformado em sofisticados ambientes computacionais que não apenas fornecem recursos especializados, funções computacionais e conteúdo para o usuário final, como também estão integradas a bancos de dados corporativos e aplicações comerciais.

## 2.5 PLATAFORMA JAVA ENTERPRISE EDITION

A plataforma *Java Enterprise Edition* (Java EE) é um conjunto de especificações, destinadas a aplicações empresariais, implementadas por vários tipos diferentes de *containers*, que são ambientes de tempo de execução Java EE que fornecem determinados serviços para os componentes que eles hospedam, como gerenciamento do ciclo de vida, injeção de dependência, etc. (GONCALVES, 2013).

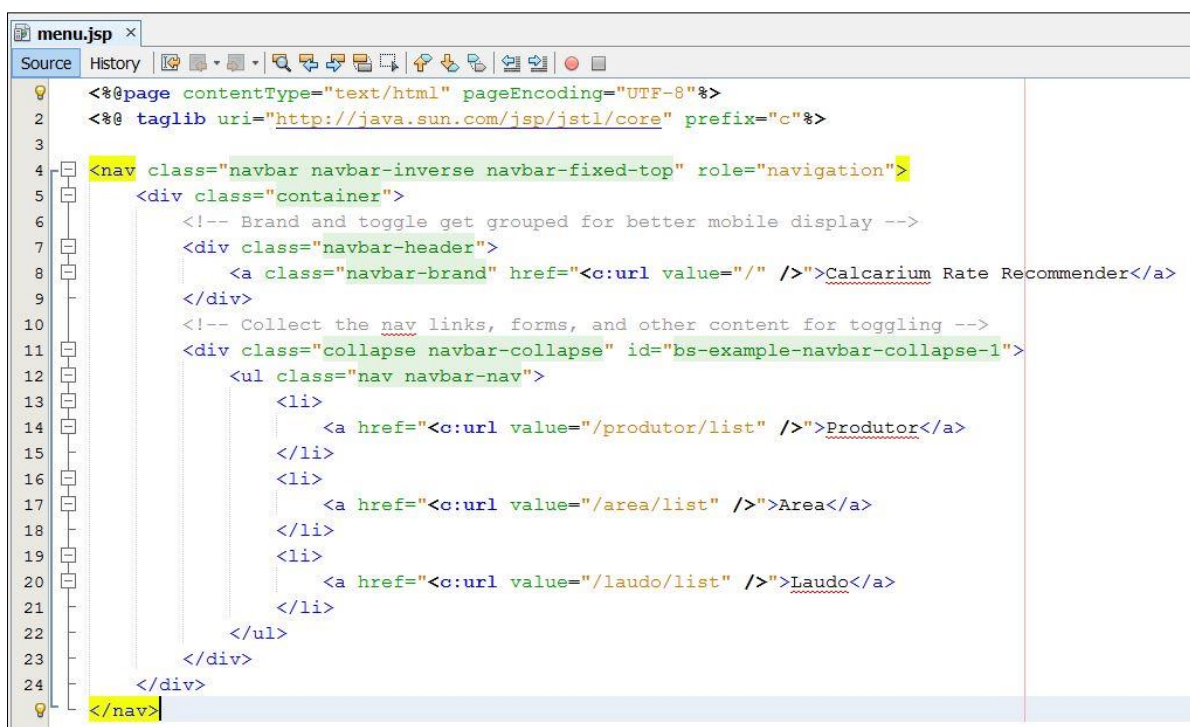
O objetivo da plataforma Java EE é fornecer aos desenvolvedores um conjunto de *Application Programming Interface* (APIs), melhorando o desempenho de uma aplicação e reduzindo a complexidade e tempo de desenvolvimento (JENDROCK et al., 2014).

Dentre as especificações do Java EE, pode-se destacar as utilizadas neste trabalho, como a *Java Servlet Pages* (JSP) e *Java Persistence API* (JPA).

## 2.6 BIBLIOTECA JSTL

A *JSP Standard Tag Library (JSTL)* é desenvolvida pelo *Java Specification Request-52*. Segundo Matha (2013) e Oracle (2016), a biblioteca JSTL é constituída por um conjunto de *tags* padrão que são usados para fornecer funcionalidades como a tomada de decisões, iteração e condicionais, manipulação de XML, internalização e localização, acesso a banco de dados e *tags* SQL, análise e formatação de números e datas, etc, e também fornece um framework para a integração de *tags* personalizadas existentes com as *tags* JSTL.

A utilização de *tags* JSTL na aplicação foi muito útil, tornando possível uma boa prática de programação que é o reuso de código. Com as *tags* JSTL algumas funcionalidades foram encapsuladas para serem utilizadas em vários arquivos JSP. Na aplicação, o arquivo `menu.jsp` contém o código para o menu superior, exibido em cada página. Para incluir o menu em cada página evitando a repetição de código pode ser utilizado a *tag* `<jsp:include>`. Nas figuras 2 e 3 é exibido o código para o menu, e, a utilização da *tag*, respectivamente.



```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
3
4 <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
5   <div class="container">
6     <!-- Brand and toggle get grouped for better mobile display -->
7     <div class="navbar-header">
8       <a class="navbar-brand" href="<c:url value="/" />">Calcarium Rate Recommender</a>
9     </div>
10    <!-- Collect the nav links, forms, and other content for toggling -->
11    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
12      <ul class="nav navbar-nav">
13        <li>
14          <a href="<c:url value="/produtor/list" />">Produtor</a>
15        </li>
16        <li>
17          <a href="<c:url value="/area/list" />">Area</a>
18        </li>
19        <li>
20          <a href="<c:url value="/laudo/list" />">Laudo</a>
21        </li>
22      </ul>
23    </div>
24  </div>
25 </nav>
```

**Figura 2 – Código do menu superior**

Fonte: Autoria própria.



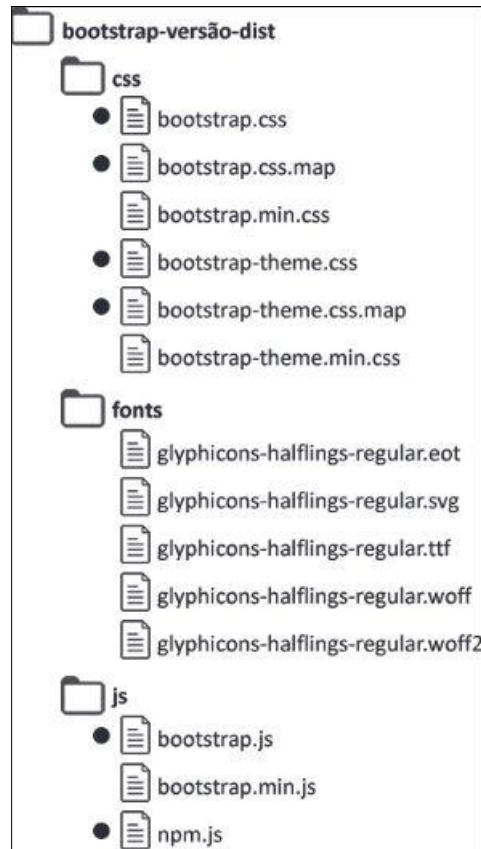
```
list.jsp x
Source History
31 |         </head>
32 |     <body>
33 |         <header class="bs-docs-nav navbar navbar-static-top">
34 |             <jsp:include page="/WEB-INF/jsp/inc/menu.jsp" />
35 |         </header>
```

**Figura 3 – Código para incluir o conteúdo do menu.jsp na página list.jsp utilizando a tag JSTL <jsp:include>**

**Fonte: Autoria própria.**

## 2.7 FRAMEWORK BOOTSTRAP

Criado em 2011, o Bootstrap surgiu como necessidade de padronizar os conjuntos de ferramentas de interface de engenheiros na empresa Twitter, onde na época trabalhavam os seus dois criadores, Mark Otto e Jacob Thornton (SPURLOCK, 2013). O Bootstrap é um framework para potencializar e aprimorar o design de interfaces web. O Bootstrap traz seus recursos empacotados, com soluções inteligentes e atributos que ajudam os desenvolvedores a realizar tarefas difíceis com facilidade e maior velocidade (SHENOY e SOSSOU, 2014). Com um design responsivo, o Bootstrap utiliza consultas CSS para medir a largura da janela do navegador e, em seguida, utilizando condicionais, determina quais partes das folhas de estilo serão carregadas (SPURLOCK, 2013), desta maneira adaptando o conteúdo para uma melhor visualização e conseqüentemente uma melhor interação humano-computador. Na Figura 4 visualiza-se a estrutura dos arquivos que compõem o pacote de download do Bootstrap.



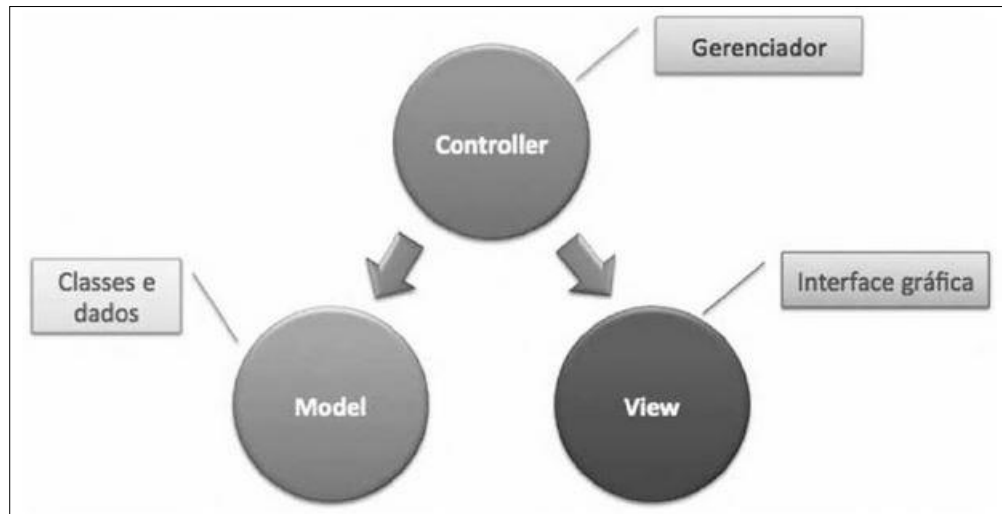
**Figura 4 – Arquivos do download do Bootstrap**  
**Fonte: Adaptado de (SILVA, 2015).**

## 2.8 PADRÃO MVC

Para Seshadri e Brad (2014), o padrão MVC (*Model-View-Controller*) evoluiu como uma maneira de separar unidades lógicas e responsabilidades no desenvolvimento de aplicações de grande porte, e oferece aos desenvolvedores um ponto de partida para decidir como e em que ponto as responsabilidades serão divididas.

De acordo com Seshadri e Brad (2014), o padrão de arquitetura MVC separa uma aplicação em três partes distintas e modulares: *model*, *view* e *controller*, representados na Figura 5:

- *Model*: corresponde aos dados da aplicação;
- *View*: corresponde à interface com a qual o usuário interage. Normalmente gerada de acordo com o *model*;
- *Controller*: representa a lógica de negócios e a camada de apresentação.



**Figura 5 – Arquitetura MVC**  
**Fonte: Adaptado de (MILANI, 2014).**

## 2.9 FRAMEWORK VRAPTOR

O VRaptor 4 é um framework MVC para a web focado no desenvolvimento ágil. De acordo com Cavalcanti (2014), o VRaptor é também útil para desenvolver APIs HTTP/REST para comunicação entre sistemas, e dá suporte à parte do Modelo — as classes que contém a lógica de negócio da aplicação. “Através da inversão de controle e injeção de dependências, ele diminui drasticamente o tempo de trabalho que seria perdido com o código repetitivo: validações, conversões, direcionamentos, ajax e lookups. ” (CAELUM ENSINO E INOVAÇÃO, 2016).

Na Figura 6 exibe-se a configuração para utilizar o VRaptor em um projeto, adicionando a dependência do VRaptor no arquivo pom.xml.

```
<dependency>
  <groupId>br.com.caelum</groupId>
  <artifactId>vraptor</artifactId>
  <version>4.2.0-RC4</version>
</dependency>
```

**Figura 6 – Tags no arquivo pom.xml para configurar o VRaptor em um projeto**  
**Fonte: Autoria própria.**

### 3 MATERIAIS E MÉTODOS

Nesta seção estão descritos os recursos utilizados no desenvolvimento do projeto, demonstrando a análise prevista para a construção da aplicação e a integração das tecnologias descritas anteriormente.

#### 3.1 TECNOLOGIAS UTILIZADAS

Para a construção da aplicação foram utilizados os seguintes recursos:

A codificação da parte de Banco de Dados do projeto foi desenvolvida no SGBD PostgreSQL versão 9.4, escolhido pelos aspectos de gratuidade e possibilidade de manipular dados geográficos. Esta etapa de desenvolvimento consistiu na criação da estrutura do BD e desenvolvimento das funções em linguagem PL/pgSQL, através da ferramenta de administração PgAdmin III.

A codificação da parte WEB do projeto foi feita em linguagem Java 8, utilizando o ambiente do Netbeans IDE versão 8.1. Com intuito de agilizar o desenvolvimento, a aplicação WEB foi construída a partir dos padrões do framework de desenvolvimento VRaptor 4.

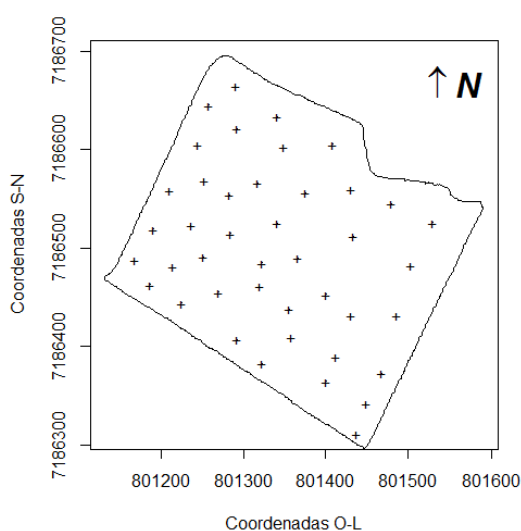
Frameworks utilizados para desenvolver o projeto:

- Maven: Framework de gerenciamento de projetos;
- Bootstrap 3.3.7: Framework *open-source* de *front-end* para construção de web sites;
- JQuery: É uma biblioteca JavaScript gratuita projetada para simplificar a criação de *scripts* HTML do lado do cliente;
- Bootstrap-select: é um *plugin* jQuery que utiliza o dropdown.js do Bootstrap para personalizar e trazer funcionalidades adicionais para os componentes padrões de seleção;
- GlassFish Server 4.1.1: Servidor da aplicação;
- VRaptor 4: O VRaptor é um framework MVC *open-source* voltado para desenvolvimento ágil e de alta produtividade. Implementa um controlador baseado no padrão Front Controller;

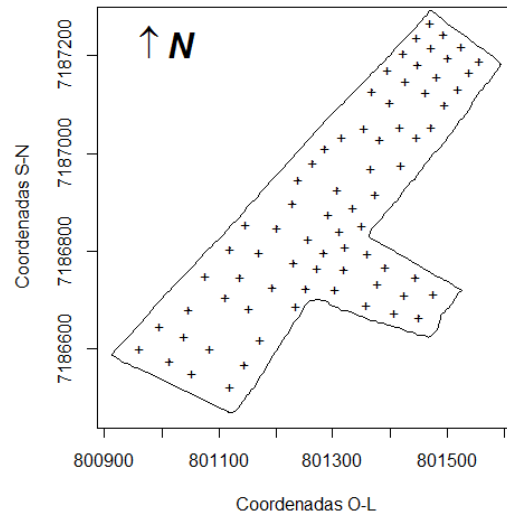
- EclipseLink: Framework responsável por controlar a persistência dos objetos no banco de dados bem como a comunicação com o mesmo;
- JDBC: Java Database Connectivity (JDBC) é a biblioteca de persistência em banco de dados relacionais do Java. Foi utilizado a versão do driver JDBC PostgreSQL 9.4-1201.

### 3.2 DADOS UTILIZADOS E DESENVOLVIMENTO NO BANCO DE DADOS

Após a modelagem do BD para armazenamento dos dados, foram implementados os cálculos de recomendação em linguagem PL/pgSQL, testadas com os dados reais de análises de solos coletados nos anos de 2012 a 2016 em duas áreas agrícolas no município de Serranópolis do Iguaçu-PR. A área A possui seu centro geográfico sob as coordenadas 25°24'30" S e 54°00'10" O e altitude média de 362,83 m, medindo aproximadamente 9,9 ha na qual foram definidos 42 pontos amostrais. A área B possui o centro geográfico sob as coordenadas 25°24'17" S e 54°00'18" O e altitude média de 363,58 m, medindo aproximadamente 19,8 ha, onde foram definidos 73 pontos amostrais. Nas Figuras 7 e 8, são exibidos os mapas com o contorno das áreas e seus respectivos pontos de coleta de solo para análise.



**Figura 7 – Mapa da Área A**  
**Fonte: Autoria própria.**

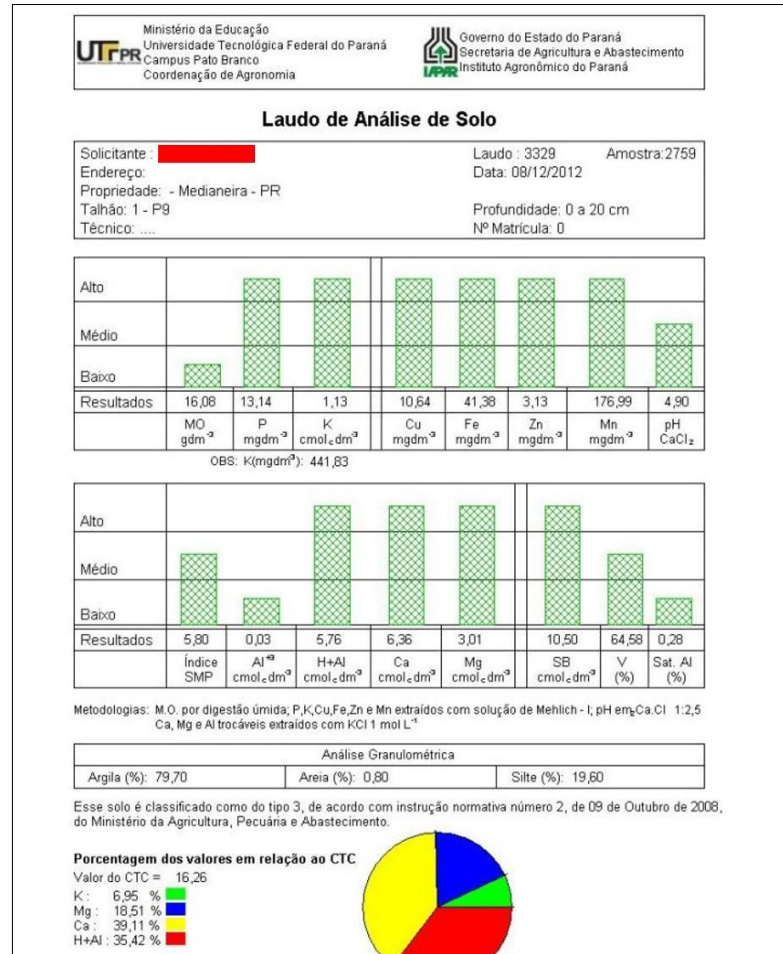


**Figura 8 – Mapa da Área B**  
**Fonte: Autoria própria.**

### 3.3 LAUDE DE ANÁLISE DE SOLO

Para obter resultados de recomendação, é necessário que o usuário tenha em mãos os dados de um laudo de análise de solo. Primeiramente deve-se cadastrar o laudo no sistema, para que seja possível realizar operações com os dados do mesmo, como a visualização e cálculos para recomendação de calagem. Na Figura 9 exibe-se um exemplo de laudo de análise de solo, de um dos pontos da área agrícola mencionada no item 3.2. Como aspecto de validação dos dados utilizados na aplicação, para calcular as recomendações de calagem foram utilizados os dados do laudo exibido na Figura 9.





**Figura 9 – Exemplo de um Laudo de Análise de Solo**  
**Fonte: UTFPR – Campus Pato Branco.**

### 3.4 ANÁLISE DO SOFTWARE

A fim de antecipar o aprimoramento no desenvolvimento do projeto, definindo claramente os recursos e objetivos do sistema, descreve-se a visão geral do sistema e seus requisitos.

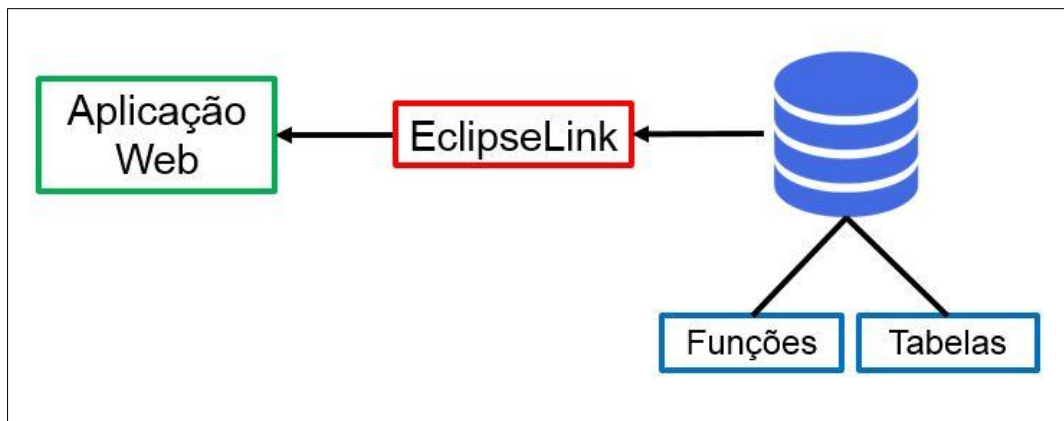
#### 3.4.1 Visão Geral do Sistema

O estudo proposto compreende no desenvolvimento de uma aplicação que seja possível fazer recomendações de calagem. O acesso à aplicação deve ser feito por meio da

Web. Os usuários, que podem ser agricultores e pessoas da área técnica devem cadastrar um produtor rural, uma ou mais áreas agrícolas e por fim uma ou mais análises de solo. Um produtor pode ter várias áreas, e cada área pode ter vários laudos de análise. Para cada laudo cadastrado o sistema deve gerar recomendações particulares. O resultado com as recomendações deve ser exibido em uma página web.

### 3.5 ARQUITETURA DO PROJETO

O projeto pode ser separado em duas partes principais, sendo uma a aplicação Web feita com Java e VRaptor, e a segunda sendo o Banco de Dados com o pacote de funções no PostgreSQL. Na Figura 10 é ilustrada esta visão geral da arquitetura do projeto, com as duas partes se comunicando através do framework de persistência EclipseLink.



**Figura 10 – Arquitetura do projeto**  
**Fonte: Autoria própria.**

#### 3.5.1 Arquitetura do Projeto Web

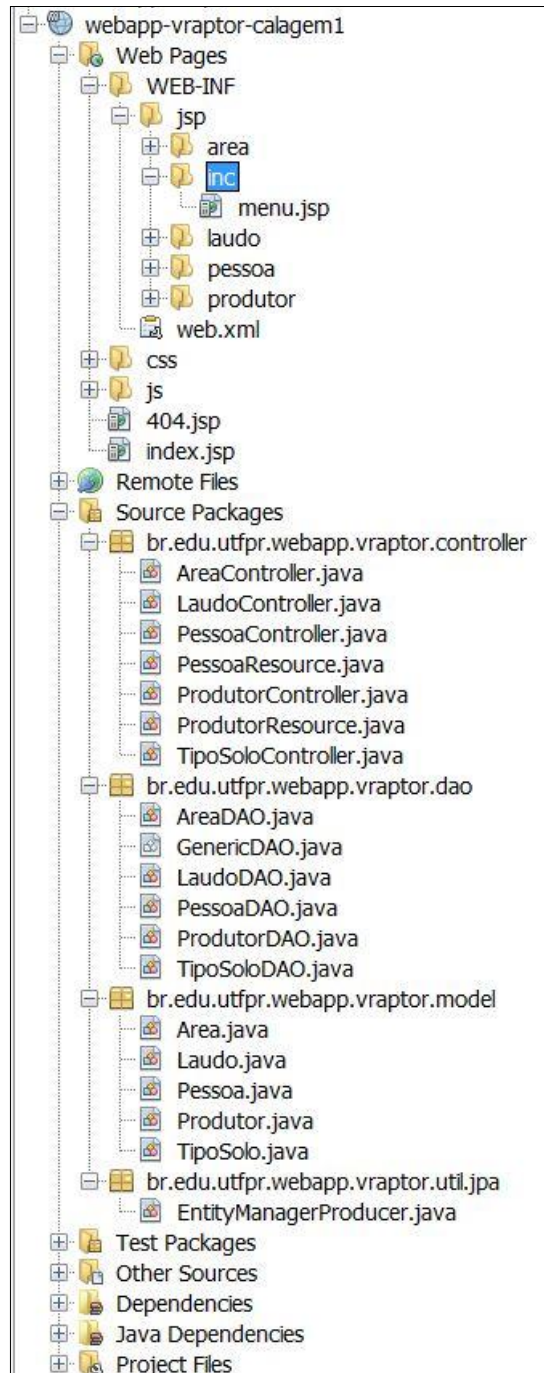
A arquitetura da aplicação Web está descrita.

- *WEB-INF*: nesta pasta estão organizadas as páginas JSP;
- *Remote files*: são arquivos JavaScript e css acessados pela web;

- Pacotes: as classes Java estão divididas em três pacotes: *controller*, *dao*, *model* e *util*;
- *Controller*: são as classes que contém a lógica de negócio da aplicação;
- *Dao*: classes responsáveis em pegar os dados do BD;
- *Model*: são as classes de modelo da aplicação;
- *Util*: este último pacote armazena a classe `EntityManagerProducer`, a qual não se encaixa ao escopo das outras classes;
- *Other sources*: nesta pasta encontra-se o arquivo `persistence.xml`, que é o arquivo padrão de configurações da JPA (JBoss Application Server 4.2.2 Community Documentation, 2008);
- *Dependencies*: contém os arquivos jar necessários para o projeto.

Na Figura 11 é exibida a estrutura do projeto web organizada dentro do Netbeans IDE

8.1.

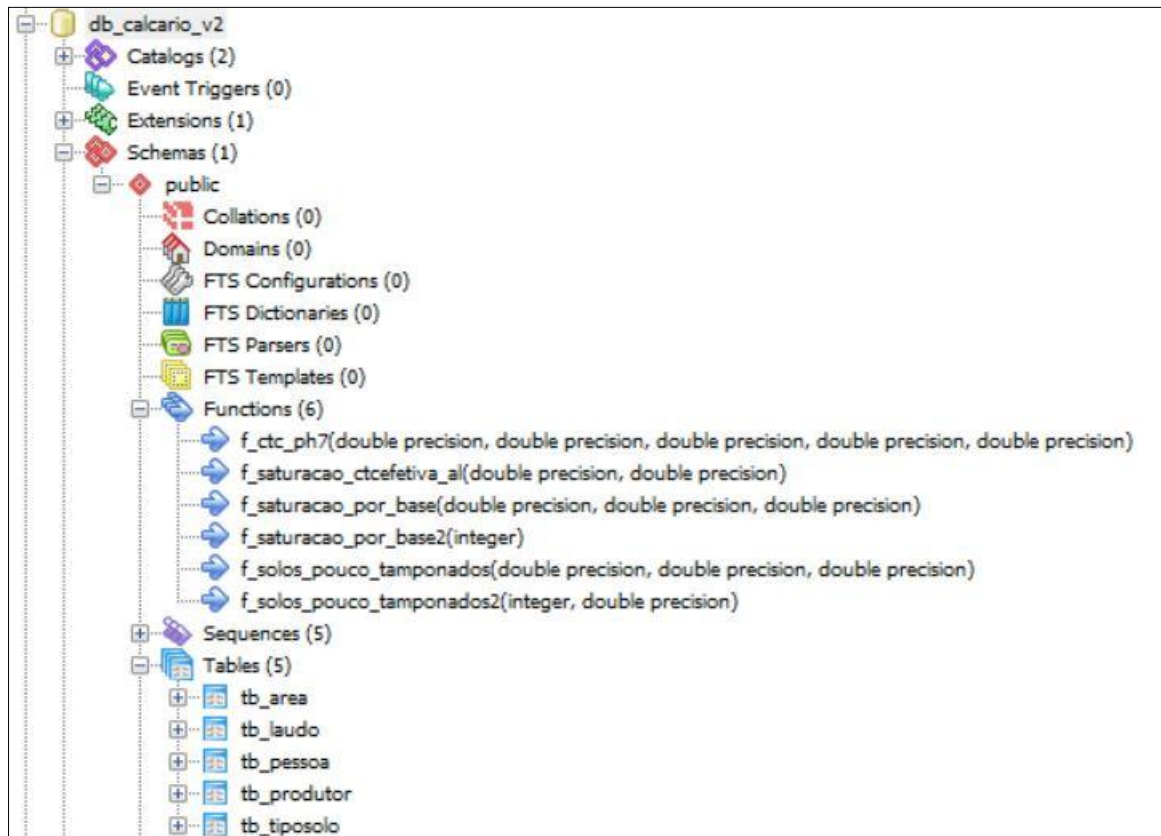


**Figura 11 – Estrutura do Projeto Web**  
**Fonte: Autoria própria.**

### 3.5.2 Arquitetura da Aplicação no Banco de Dados

No Banco de Dados existem duas divisões, sendo elas *Functions* e *Tables*, que armazenam as funções e tabelas, respectivamente. As funções foram implementadas no BD a

fim de estruturar o projeto para implementações futuras de novas funcionalidades, como operações espaciais, deste modo facilitando a integração destas. Na Figura 12 é exibida a estrutura do BD.



**Figura 12 – Estrutura do BD no ambiente pgAdmin III**

**Fonte: Autoria própria.**

## 4 RESULTADOS E DISCUSSÃO

Nesta seção são apresentados os resultados obtidos através da descrição do desenvolvimento do sistema e exibição do mesmo em funcionamento.

### 4.1 FUNÇÕES EM PL/PGSQL

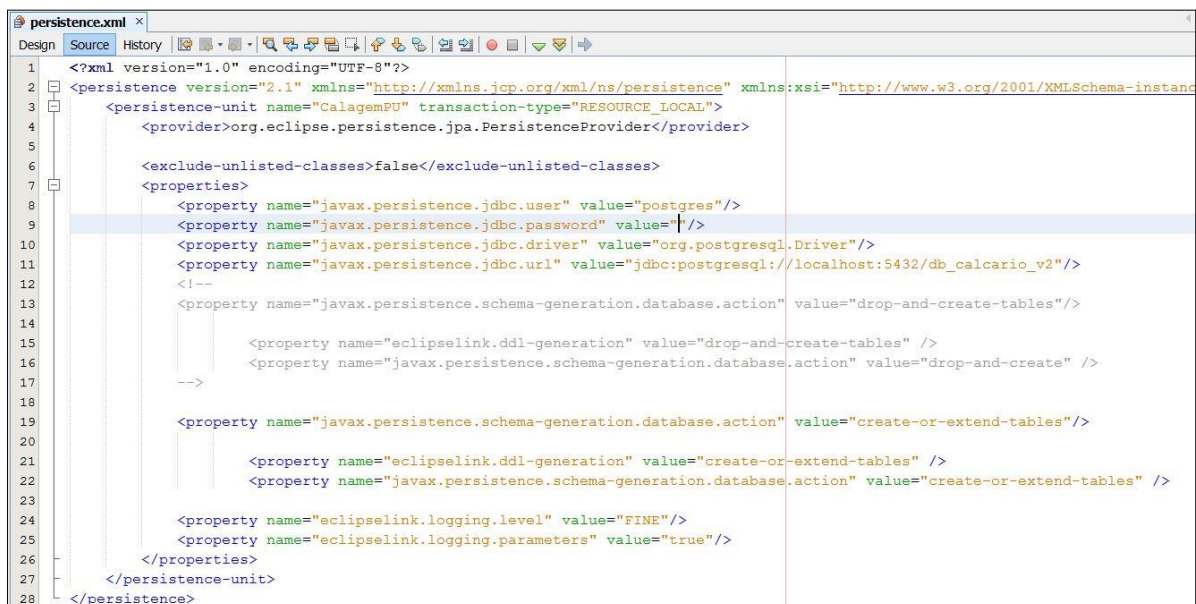
O projeto foi iniciado com a implementação das funções em PL/pgSQL. Foi escolhido iniciar com esta etapa pelo fato dela não necessitar a comunicação com outros softwares para testar sua funcionalidade. Posteriormente foi iniciado o desenvolvimento da aplicação em Java. O desenvolvimento foi feito no banco apenas durante a criação das funções, pois o framework de persistência EclipseLink tem a função de estabelecer a comunicação com o banco de dados e gerenciar a persistência, de modo que o desenvolvedor possa focar no desenvolvimento Java. Na Figura 13 visualiza-se a função que determina a NC pelo critério da saturação por bases.

```
1 CREATE OR REPLACE FUNCTION
2 f_saturacao_por_base2(p_laudo_id integer)
3
4 RETURNS double precision AS
5 $BODY$
6 DECLARE
7     v_CTC float;
8     v_V1 float;
9     v_V2 float := 80;
10    v_NC float;
11 BEGIN
12     EXECUTE 'select tb_laudo.ctc, tb_laudo.v
13             from tb_laudo
14             where tb_laudo.id = ' || p_laudo_id
15
16     INTO v_CTC, v_V1;
17
18     v_NC := (v_CTC * (v_V2 - v_V1)) / 100;
19
20     RETURN v_NC;
21 END;
22 $BODY$
23 LANGUAGE plpgsql VOLATILE
24 COST 100;
```

**Figura 13 – Função para calcular a NC pelo critério de saturação por bases**  
**Fonte: Autoria própria.**

## 4.2 APLICAÇÃO WEB DESENVOLVIDA

Inicialmente foi necessário a configuração no arquivo `persistence.xml` com os parâmetros necessários para estabelecer a comunicação com o SGBD PostgreSQL. Neste arquivo está definido o framework de persistência no caso o EclipseLink e também o modo de inicialização da estrutura do BD. Na Figura 14 visualiza-se a configuração do projeto no arquivo `persistence.xml`.



```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3  <persistence-unit name="CalagemPU" transaction-type="RESOURCE_LOCAL">
4  <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
5
6  <exclude-unlisted-classes>>false</exclude-unlisted-classes>
7  <properties>
8  <property name="javax.persistence.jdbc.user" value="postgres"/>
9  <property name="javax.persistence.jdbc.password" value="" />
10 <property name="javax.persistence.jdbc.driver" value="org.postgresql.Driver"/>
11 <property name="javax.persistence.jdbc.url" value="jdbc:postgresql://localhost:5432/db_calcario_v2"/>
12 <!--
13 <property name="javax.persistence.schema-generation.database.action" value="drop-and-create-tables"/>
14
15 <property name="eclipseLink.ddl-generation" value="drop-and-create-tables" />
16 <property name="javax.persistence.schema-generation.database.action" value="drop-and-create" />
17 -->
18
19 <property name="javax.persistence.schema-generation.database.action" value="create-or-extend-tables"/>
20
21 <property name="eclipseLink.ddl-generation" value="create-or-extend-tables" />
22 <property name="javax.persistence.schema-generation.database.action" value="create-or-extend-tables" />
23
24 <property name="eclipseLink.logging.level" value="FINE"/>
25 <property name="eclipseLink.logging.parameters" value="true"/>
26 </properties>
27 </persistence-unit>
28 </persistence>

```

**Figura 14 – Visão do arquivo persistence.xml**  
**Fonte: Autoria própria.**

### 4.2.1 Desenvolvimento de um DAO genérico

Para a comunicação com o BD, utiliza-se as classes *Data-access-object* (DAO). Como muitas das operações podem ser utilizadas igualmente pelas classes, pode se criar uma classe `GenericDAO`, possuindo métodos comuns, como operações `CREATE`, `READ`, `UPDATE`, `DELETE` (CRUD). Na Figura 15 demonstra-se um trecho da classe DAO genérica.



```

1 package br.edu.utfpr.webapp.vraptor.dao;
2
3 import java.lang.reflect.ParameterizedType;
4 import java.util.List;
5 import javax.inject.Inject;
6 import javax.persistence.EntityManager;
7 import javax.persistence.TypedQuery;
8
9
10 public abstract class GenericDAO<PK, T> {
11     @Inject
12     protected EntityManager entityManager;
13
14     public T getById(PK pk) { ...3 lines }
15
16
17     public long count() { ...5 lines }
18
19
20     public boolean contains(T entity) { ...3 lines }
21
22
23     public void save(T entity) { ...5 lines }
24
25
26     public void update(T entity) { ...5 lines }
27
28
29     public void delete(T entity) { ...5 lines }
30
31
32     public List<T> findAll() { ...3 lines }
33
34
35     private Class<?> getTypeClass() {
36         Class<?> clazz = (Class<?>) ((ParameterizedType) this.getClass()
37             .getGenericSuperclass()).getActualTypeArguments()[1];
38         return clazz;
39     }
40 }

```

**Figura 15 – Trecho de código da classe GenericDAO**  
**Fonte: Autoria própria.**

#### 4.2.2 Desenvolvimento de uma classe modelo

Para demonstrar o desenvolvimento desde uma classe modelo até sua exibição e manipulação na parte Web, foi utilizado a classe Produtor, onde são cadastrados os produtores que são os usuários do sistema.

A partir da Figura 16 pode-se observar a utilização de anotações como @Entity e @Table. A primeira define que a classe corresponde a uma tabela no BD, e na segunda podemos especificar parâmetros como o nome que deverá ser dado à tabela. Caso a anotação @Table seja omitida, a tabela será criada com o mesmo nome da classe. A última anotação exibida é @OneToMany, que define um relacionamento de um pra muitos com a tabela que



armazena os dados sobre as áreas agrícolas. Para o relacionamento ser efetivado, a classe correspondente à tabela de áreas deve possuir a anotação @ManyToOne.

```

1 package br.edu.utfpr.webapp.vraptor.model;
2
3 import ...10 lines
4
13
14 /**
15  *
16  * @author Douglas
17  */
18 @Entity
19 @Table(name = "tb_produtor")
20 public class Produtor implements Serializable {
21
22     @Id
23     @GeneratedValue(strategy = GenerationType.IDENTITY)
24     private int id;
25
26     @Column(nullable = false)
27     @NotNull
28     private String nome;
29
30     @Column(nullable = false)
31     @NotNull
32     private String fone;
33
34     @Column(nullable = false)
35     @NotNull
36     private String email;
37
38     @OneToMany(mappedBy = "produtor")
39     private Collection<Area> areas;
40
41     public int getId() {...3 lines }
42
43
44     public void setId(int id) {...3 lines }
45
46
47
48     public String getNome () {...3 lines }
49
50
51
52     public void setNome (String nome) {...3 lines }
53
54
55
56

```

**Figura 16 – Trecho de código da classe modelo Produtor**  
**Fonte: Autoria própria.**

#### 4.2.2.1 Desenvolvimento do DAO para a classe modelo

A classe que faz intermédio com o BD para os atributos do produtor é a classe ProdutorDAO que herda os métodos da classe GenericDAO. Na Figura 17 visualiza-se o código da classe ProdutorDAO.

```

1 package br.edu.utfpr.webapp.vraptor.dao;
2
3 import br.edu.utfpr.webapp.vraptor.model.Produtor;
4 import javax.enterprise.context.RequestScoped;
5
6 /**
7  *
8  * @author Douglas
9  */
10
11 @RequestScoped
12 public class ProdutorDAO extends GenericDAO<Integer, Produtor> {
13
14     public ProdutorDAO() {
15         super();
16     }
17
18 }

```

**Figura 17 – Código da classe ProdutorDAO**  
**Fonte: Autoria própria.**

#### 4.2.2.2 Desenvolvimento do Controller

Na classe controladora foram implementadas as regras de negócio. É importante ressaltar que com o VRaptor, em uma classe anotada com `@Controller`, os métodos públicos ficam disponíveis para receber requisições web através de chamadas do tipo GET ou POST a URLs específicas. No exemplo da Figura 18 com a classe `ProdutorController`, os métodos fazem com que as URLs `/produtor/list`, `/produtor/form`, `/produtor/save` sejam disponibilizadas, cada uma invocando o respectivo método em sua classe (DOCUMENTAÇÃO VRAPTOR, 2016).

```

1 package br.edu.utfpr.webapp.vraptor.controller;
2
3 import br.com.caelum.vraptor.Controller;
4 import br.com.caelum.vraptor.Result;
5 import br.edu.utfpr.webapp.vraptor.dao.ProdutorDAO;
6 import br.edu.utfpr.webapp.vraptor.model.Produtor;
7 import java.util.List;
8 import javax.inject.Inject;
9
10 /**...4 lines */
11
12
13
14
15 @Controller
16 public class ProdutorController {
17     @Inject
18     private ProdutorDAO produtorDAO;
19
20     @Inject
21     private Result result;
22
23     public List<Produtor> list() {
24         return produtorDAO.findAll();
25     }
26
27     public Produtor form(int id) {
28         return produtorDAO.getById(id);
29     }
30
31
32     public void save(Produtor produtor) {
33         if(produtor.getId() > 0)
34             produtorDAO.update(produtor);
35         else
36             produtorDAO.save(produtor);
37
38         // Redireciona para a página de listagem
39         result.redirectTo(ProdutorController.class).list();
40     }

```

**Figura 18 – Trecho de código da classe ProdutorController**  
**Fonte: Autoria própria.**

#### 4.2.2.3 Inclusão de bibliotecas no JSP

Nesta parte será apresentada a visão Web do produtor. Primeiramente devem ser incluídas as bibliotecas, conforme a Figura 19.

```

6
7 <@page contentType="text/html" pageEncoding="UTF-8"%>
8 <@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
9 <@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
10
11 <!DOCTYPE html>
12 <html>
13   <head>
14     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
15     <title>Recomendação de Calagem</title>
16
17     <!-- IMPORTANTE JQuery - Um local, e o outro online -->
18     <!-- Este link precisa estar antes dos links do Bootstrap! Previne o erro - "Bootstrap's JavaScript requires
19     <script type="text/javascript" src="../../js/jquery-3.1.0.js"></script>
20     <script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.js"></script>
21
22     <!-- Latest compiled and minified CSS -->
23     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity=
24
25     <!-- Optional theme -->
26     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css" in
27
28     <!-- Latest compiled and minified JavaScript -->
29     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity="sha384-Tc5IQib0
30   </head>

```

**Figura 19 – Trecho de código da página de listagem, com inclusão de bibliotecas**  
**Fonte: Autoria própria.**

#### 4.2.2.4 Página principal

No corpo da página é incluído uma tabela para exibir os atributos do produtor com as possibilidades de editar e excluir os dados, e um botão para acessar a página de cadastro de produtores. Na Figura 20 visualiza-se um trecho do código para o corpo da página de listagem de produtores.

```

37 <div class="container">
38 <div class="page-header">
39 <div class="jumbotron" style="text-align: center">
40 <h1>Recomendação de Calagem</h1>
41 </div>
42 <h1>Produtor <small>lista</small>
43 <a href="<c:url value="/produtor/form" />" class="btn btn-success"><span class="glyphicon glyphicon glyp
44 </h1>
45 </div>
46
47 <div>
48 <c:if test="<math>\$ \{ \text{not empty } \text{mensagem} \}>">
49 <div class="alert alert-warning" style="width: 90%; margin: 50px auto; "><math>\$ \{ \text{mensagem} \}</div>
50 </c:if>
51 <table class="table">
52 <tr>
53 <th><a href="<math>\$ \{ ? \text{ordem} = \text{id} \}>\#</a></th>
54 <th><a href="<math>\$ \{ ? \text{ordem} = \text{nome} \}>">Nome</a></th>
55 <th>Fone</th>
56 <th>Email</th>
57 </tr>
58 <c:forEach var="item" items="<math>\$ \{ \text{produtorList} \}" \text{ varStatus} = \text{"row"}>
59 <tr>
60 <td><math>\$ \{ \text{item.id} \}</td>
61 <td><math>\$ \{ \text{item.nome} \}</td>
62 <td><math>\$ \{ \text{item.fone} \}</td>
63 <td><math>\$ \{ \text{item.email} \}</td>
64 <td>
65 <a class="btn btn-default btn-xs" href="<math>\$ \{ ? \text{ordem} = \text{id} \}>" /><math>\$ \{ \text{item.id} \}><sp
66 <a class="btn btn-default btn-xs" href="<math>\$ \{ ? \text{ordem} = \text{delete} \}>" /><math>\$ \{ \text{item.id} \}><
67 </td>
68 </tr>
69 </c:forEach>
70 </table>
71 </div>
72 </div>

```

Figura 20 – Trecho do código para o corpo da página de listagem de produtores

Fonte: Autoria própria.

#### 4.2.2.5 Formulário de cadastro de produtores

O código do corpo da página de cadastro de produtores é exibido na Figura 21.

```

form.jsp x
Source History
44 <div>
45 <:if test="{not empty mensagem}">
46 <div class="alert alert-warning" style="width: 90%; margin: 50px auto;">${mensagem}</div>
47 </:if>
48 <section>
49 <form action="{c:url value='/produtor/save'/}" method="POST">
50 <:if test="{not empty produtor}">
51 <div class="form-group">
52 <label for="inputId">ID</label>
53 <input class="form-control" id="inputId" type="text" name="produtor.id" value="{produtor.id}" readonly="readonly" />
54 </div>
55 </:if>
56 <div class="form-group">
57 <label for="inputNome">Nome</label>
58 <input class="form-control" id="inputNome" type="text" placeholder="Nome" name="produtor.nome" value="{produtor.nome}" requi
59 </div>
60 <div class="form-group">
61 <label for="inputFone">Fone</label>
62 <input class="form-control" id="inputFone" type="tel" placeholder="Fone" name="produtor.fone" value="{produtor.fone}" requi
63 </div>
64 <div class="form-group">
65 <label for="inputEmail">Email</label>
66 <input class="form-control" id="inputEmail" type="email" placeholder="Email" name="produtor.email" value="{produtor.email}"
67 </div>
68 <button type="submit" class="btn btn-primary">Salvar</button>
69 <button type="reset" class="btn btn-default">Resetar</button>
70 <a href="list" class="btn btn-default">Cancelar</a>
71 </form>
72 </section>
73 </div>
74 </div>
75 </div>
76 </div>

```

**Figura 21 – Trecho do código para o corpo da página de cadastro de produtores**  
**Fonte: Autoria própria.**

### 4.2.3 Apresentação das telas da Aplicação Web

Nesta seção será apresentado a aplicação em funcionamento, com a descrição e ilustração das telas, primeiramente pela página inicial, conforme ilustrado na Figura 22.



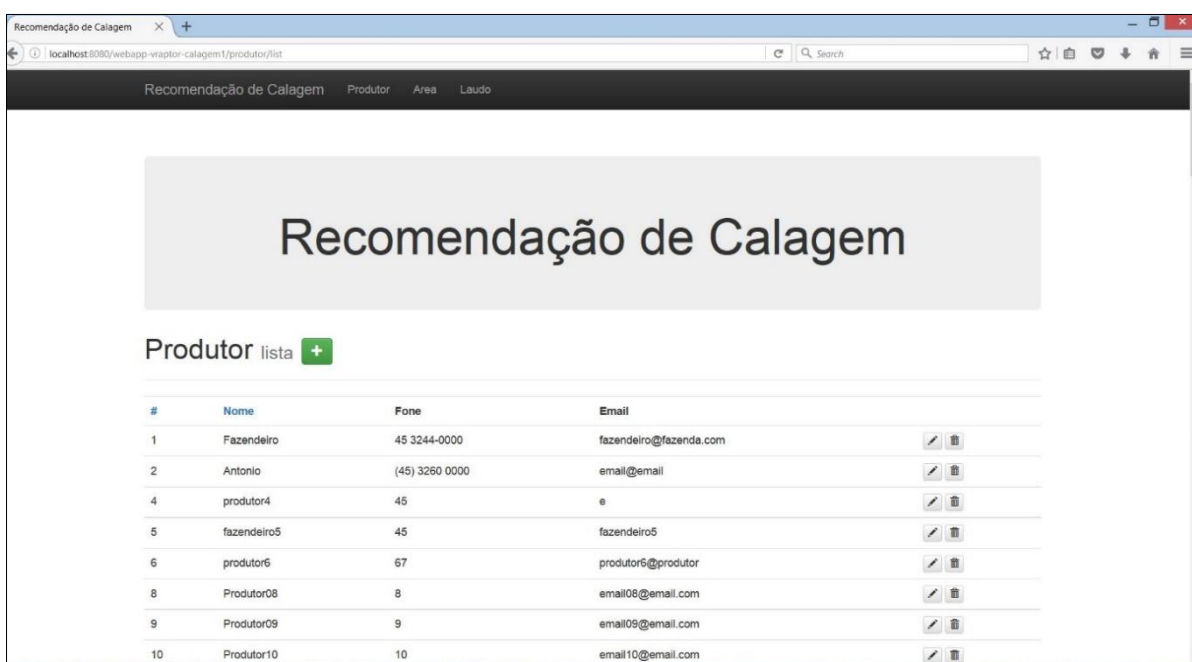
**Figura 22 – Tela inicial da aplicação**  
**Fonte: Autoria própria.**



















Pelo clássico layout da página, é possível perceber a utilização do Bootstrap. O menu superior contém *links* para a página inicial, página de produtores, áreas e laudos. Este é o menu incluído em todas as páginas através de *tags* JSTL, conforme descrito no item 2.6.

#### 4.2.3.1 Tela de listagem de produtores

Na tela de listagem pode-se observar os dados do produtor, dispostos em uma tabela, com botões para edição e deleção de dados. O botão com sinal de adição direciona para página de cadastros e foi adotado como padrão para todas as páginas de listagem. Na Figura 23 é exibida a página com a listagem de produtores.

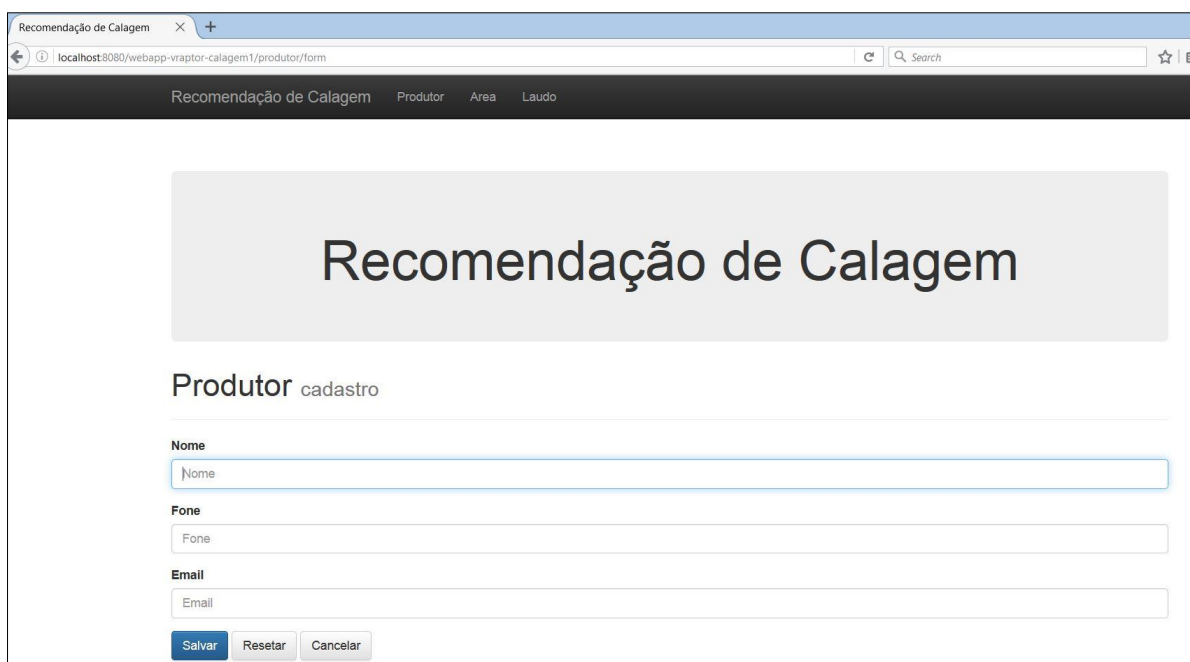


#	Nome	Fone	Email	
1	Fazendeiro	45 3244-0000	fazendeiro@fazenda.com	 
2	Antonio	(45) 3260 0000	email@email	 
4	produtor4	45	e	 
5	fazendeiro5	45	fazendeiro5	 
6	produtor6	67	produtor6@produtor	 
8	Produtor08	8	email08@email.com	 
9	Produtor09	9	email09@email.com	 
10	Produtor10	10	email10@email.com	 

**Figura 23 – Tela de listagem de produtores**  
**Fonte: Autoria própria.**

## Tela de cadastro de produtores

Para cadastrar um novo produtor, é necessário clicar no botão com sinal de adição descrito anteriormente, que direciona à página de cadastros, exibida na Figura 24.



Recomendação de Calagem

Produtor Area Laudo

# Recomendação de Calagem

Produtor cadastro

Nome

Fone

Email

Salvar Resetar Cancelar

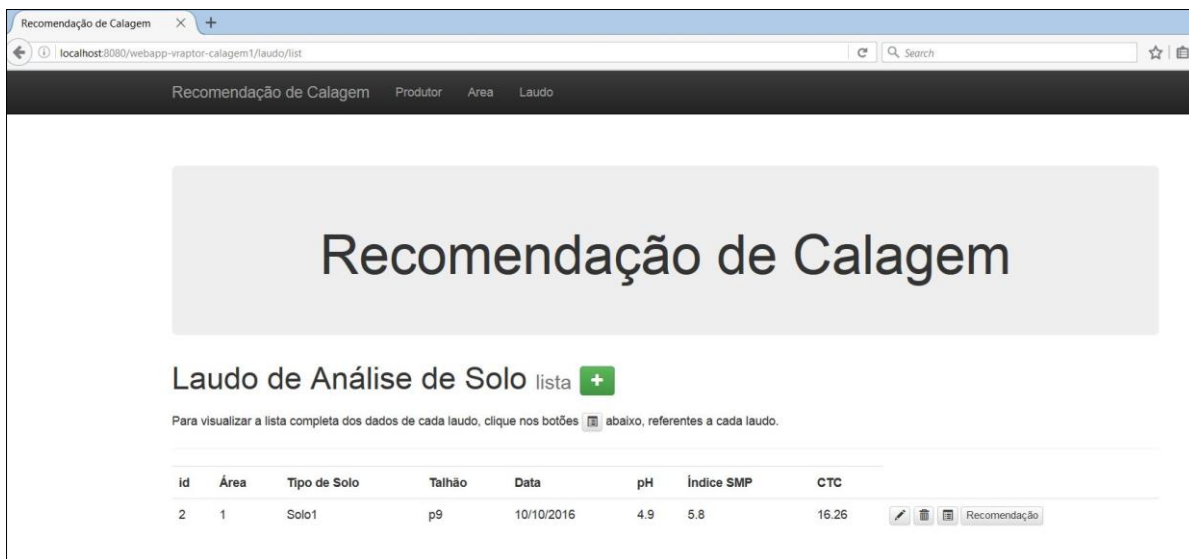
**Figura 24 – Tela de cadastro de produtores**

**Fonte: Autoria própria.**

### 4.2.4 Exibição de um laudo na Aplicação Web

A listagem de laudos foi separada em duas páginas devido à grande quantidade de atributos. Os dados mais importantes em uma primeira visualização foram dispostos na página direcionada pelo menu superior, conforme a Figura 25. A partir dessa página é possível visualizar todos os dados clicando no botão de listagem completa, ou acessar diretamente a página de recomendações. É possível também visualizar todos os dados de um laudo, conforme a página da aplicação indica.





**Figura 25 – Tela da listagem resumida dos dados do laudo**  
**Fonte: Autoria própria.**

#### 4.2.5 Listagem estendida do laudo

Na Figura 26 é exibida a listagem estendida do laudo.



**Figura 26 – Tela da listagem estendida dos dados do laudo**  
**Fonte: Autoria própria.**

#### 4.2.6 Cadastro de um laudo na Aplicação

A página de cadastro de laudos segue o mesmo padrão descrito anteriormente. Na Figura 27 é exibida a página de cadastro de laudos.

The screenshot shows a web browser window with the URL `localhost:8080/webapp-iraptor-calagem1/laudo/form`. The page title is "Recomendação de Calagem" and the breadcrumb is "Produtor > Área > Laudo". The main heading is "Recomendação de Calagem" and the sub-heading is "Laudo de Análise de Solo cadastro".

Form fields include:

- Dropdowns: "Selecione o proprietário da Área:" (with "Selecione um Produtor ..." option), "Selecione a Área:", and "Selecione o tipo de solo:" (with "Selecione o solo ..." option).
- Input fields: Talhão (with "Talhão" value), Data (with "ex. 24/10/2016" value), Matéria Orgânica(MO) (with "MO" value), Fósforo(P) (with "P" value), Potássio(K) (with "K" value), Cobre(Cu) (with "Cu" value), Ferro(Fe) (with "Fe" value), Zinco(Zn) (with "Zn" value), Manganês(Mn) (with "Mn" value), Acidez(pH) (with "ph" value), Índice SMP (with "SMP" value), Alumínio(Al) (with "Al" value), (H+Al) (with "H+Al" value), Cálcio(Ca) (with "Ca" value), Magnésio(Mg) (with "Mg" value), Saturação de Base(SB) (with "SB" value), (V%) (with "V%" value), (Sat. Al %) (with "Sat. Al" value), Argila (with "Argila" value), Areia (with "Areia" value), Silte (with "Silte" value), and CTC (with "CTC" value).
- Buttons: "Salvar" (blue), "Resetar", and "Cancelar" (grey).

**Figura 27 – Tela de cadastro de laudos**

**Fonte: Autoria própria.**

#### 4.2.7 Exibição das recomendações de calagem

O último passo da navegação pela aplicação é a visualização dos resultados obtidos das funções do BD e pela tabela que estima a NC pelo índice SMP, descrita no item 2.1.1. Nesta página o usuário pode visualizar alguns resultados sem mesmo interagir com a página. Para visualizar a recomendação conforme o critério do índice SMP, disposto em uma tabela, é possível filtrar resultados.

#### 4.2.8 Exibição inicial sem filtros

Os dados da tabela do índice SMP foram dispostos em uma tabela dinâmica, onde o produtor pode digitar o índice SMP, que é exibido no cabeçalho da página. Na Figura 28, visualiza-se a maneira que a página é exibida inicialmente.

The screenshot shows a web browser window with the URL `localhost:8080/webapp-iraptor-calagem/laudo/functionresult/laudoId=2`. The page title is "Recomendação de Calagem". The main content area is titled "Recomendações" and includes a "Voltar" button. Below this is a table with the following data:

Laudo#	Produtor	Índice SMP	Área	Talhão
2	Fazendeiro	5.8	1	p9

Below the table, there are three main sections:

- Recomendação - Critério Solos pouco tamponados:** A section with a blue header. It contains a label "Selecione o pH Referência da cultura a ser cultivada:" and three radio buttons for pH 5.5, pH 6.0 (selected), and pH 6.5. There is a "Calcular" button and a text field showing "pH=0.0 : Aplicar toneladas por hectare."
- Quantidades de calcário necessárias para elevar o pH em água do solo estimadas pelo índice SMP<sup>+</sup>:** A table with columns for Índice SMP, pH 5.5, pH 6.0, and pH 6.5. The rows show the required lime quantity in "toneladas/hectare\*\*" for various SMP values from <math>\leq 4.4</math> to 5.2.
- Recomendação - Critério Saturação por bases:** A green box containing the text "Aplicar 2,51 toneladas por hectare."

**Figura 28 – Tela inicial de recomendações**

**Fonte: Autoria própria.**

#### 4.2.9 Exibição com filtros

Conforme a Figura 28 é possível perceber que as colunas da tabela do índice SMP são ao mesmo tempo campos, onde pode-se digitar um valor para filtrar resultados. Na Figura 29 visualiza-se o filtro em ação com o valor do índice SMP e o resultado do critério de solos pouco tamponados, onde é necessário que o usuário selecione o pH e clique no botão "Calcular".

Recomendação de Calagem

Produtor Área Laudo

## Recomendações [Voltar](#)

Laudo#	Produtor	Índice SMP	Área	Talhão
2	Fazendeiro	5.8	1	p9

**Recomendação - Critério Solos pouco tamponados**

Selecione o pH Referência da cultura a ser cultivada:

pH 5.5

pH 6.0

pH 6.5

[Calcular](#)

pH=6.0 : Aplicar 12,5 toneladas por hectare.

**Quantidades de calcário necessárias para elevar o pH em água do solo estimadas pelo índice SMP\***

5.8	pH 5.5	pH 6.0	pH 6.5
toneladas/hectare**			
5.8	2.3	4.2	6.3

**Recomendação - Critério Saturação por bases**

Aplicar 2,51 toneladas por hectare.

**Figura 29 – Tela de recomendações com filtros**  
**Fonte: Autoria própria.**

Na Figura 29 é possível perceber os resultados. Para uma lavoura onde será plantado milho ou soja, o pH referência dessas culturas é de 6.0 (COMISSÃO DE QUÍMICA E FERTILIDADE DO SOLO - RS/SC, 2004). Considerando este caso típico da região Oeste do Paraná, o sistema oferece, portanto, o resultado de três critérios diferentes, sendo o critério para solos pouco tamponados, critério do índice SMP e o critério da saturação por bases, que, para elevar o pH ao valor 6.0 recomendam a aplicação de 12,5, 4,2 e 2,51 toneladas por hectare respectivamente.

## 5 CONSIDERAÇÕES FINAIS

### 5.1 CONCLUSÃO

A aplicação desenvolvida neste trabalho mostrou que o desenvolvimento Web pode ser mais simples e ágil com a utilização dos frameworks e bibliotecas específicos, como o VRaptor e o Bootstrap, que foram de grande importância na agilidade em tempo de desenvolvimento.

O SGBD PostgreSQL se mostrou como uma boa alternativa para o desenvolvimento deste trabalho, bem como abre a possibilidade para trabalhos futuros utilizando operações espaciais.

O projeto fica disponibilizado gratuitamente à toda comunidade agrícola, portanto dessa maneira contribuindo na gestão financeira e agrícola para os produtores rurais.

### 5.2 TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO

Após este trabalho, o desenvolvimento de novas funcionalidades continuará em foco. Várias possibilidades ficam em aberto. A partir desse projeto pretende-se continuar estudando novas tecnologias para serem incorporadas, como a integração com a extensão espacial PostGIS, que será estudada para implementar funções com métodos de interpolação, a fim de gerar mapas para aplicação de calagem a partir vários laudos de uma mesma área. De fato, o leque de possibilidades é grande, com o intuito de continuar o desenvolvimento em projetos futuros.

## REFERÊNCIAS BIBLIOGRÁFICAS

- BERLAUG, Norman E. **Academy of Achievement**, 2009. Disponível em: <<http://www.achievement.org/autodoc/page/bor0int-3#>>. Acesso em: 23 out. 2016.
- CAELUM ENSINO E INOVAÇÃO. **Desenvolvimento Ágil para a Web 2.0 com VRaptor, Hibernate e AJAX**. Caelum Ensino e Inovação, 2016. Cap. 7. Disponível em: <<https://www.caelum.com.br/apostila-vraptor-hibernate/vraptor/#7-1-sobre-o-vraptor>>. Acesso em: 20 out. 2016.
- CAIRES, Eduardo F.; JORIS, Helio A. W. Uso de Corretivos Granulados na Agricultura. **Informações Agronômicas**, Piracicaba, n. 154, p. 17-21, jun. 2016. ISSN 2311-5904.
- CARVALHO, Renato M. de. **Software como suporte para recomendação de calagem e adubação para cafeeiro em produção**. 2008. 47 f. Dissertação (Mestrado em Sistemas de Produção na Agropecuária) – Universidade José do Rosário Vellano, Alfenas, 2008.
- CASÃO JUNIOR, Ruy; ARAÚJO, Augusto G. de; LLANILLO, Rafael F. **Plantio direto no Sul do Brasil**. Londrina: Organização das Nações Unidas para a Agricultura e Alimentação e Instituto Agronômico do Paraná, 2012. ISBN 978-85-88184-40-4.
- CAVALCANTI, Lucas. **VRaptor Desenvolvimento Ágil para Web com Java**. São Paulo: Editora Casa do Código, 2014. ISBN 978-85-66250-26-8.
- CERRI, Domingos G. P. **Desenvolvimento de um Sistema de Aplicação Localizada de Calcário a Taxas Variáveis**. 2001. 72 f. Dissertação (Mestrado em Agronomia) - Escola Superior de Agricultura “Luiz de Queiroz”, Universidade de São Paulo, Piracicaba, 2001.
- COMISSÃO DE QUÍMICA E FERTILIDADE DO SOLO - RS/SC. **Manual de adubação e de calagem para os Estados do Rio Grande do Sul e de Santa Catarina**. 10. ed. Porto Alegre, RS, 2004. 400 p.
- COMUNIDADE BRASILEIRA DE POSTGRESQL. Sobre o PostgreSQL: Comunidade Brasileira de PostgreSQL. **Comunidade Brasileira de PostgreSQL Web site**, 2016. Disponível em: <<https://www.postgresql.org.br/sobre>>. Acesso em: 13 out. 2016.
- CORONEL, Carlos; MORRIS, Steven; ROB, Peter. **Database Systems: Design, Implementation, and Management**. 9. ed. Boston: Cengage Learning, 2011.
- DAR, Usama et al. **PostgreSQL Server Programming Second Edition**. 2. ed. Birmingham: Packt Publishing Ltd., 2015.
- DOCUMENTAÇÃO VRAPTOR. **Controllers Rest**, 2016. Disponível em: <<http://www.vraptor.org/pt/docs/controllers-rest/>>. Acesso em: 23 out. 2016.

DOUGLAS, Korry; DOUGLAS, Susan. **PostgreSQL, Second Edition**. 2. ed. Indianapolis: Sams Publishing, 2005.

GONCALVES, Antonio. **Beginning Java EE 7**. 1 ed. Berkely: Apress, 2013.

HALPIN, Terry; MORGAN, Tony. **Information Modeling and Relational Databases (2 ed.)**. 2. ed. San Francisco: Morgan Kaufmann Publishers Inc., 2008.

IPARDES. **Leituras regionais : Mesorregião Geográfica Oeste Paranaense**. Instituto Paranaense de Desenvolvimento Econômico e Social - IPARDES. Curitiba, PR, 2003.

JBOSS Application Server 4.2.2 Community Documentation, 2008. Disponível em: <[https://docs.jboss.org/jbossas/docs/Server\\_Configuration\\_Guide/4/html/ch01s02s01.html](https://docs.jboss.org/jbossas/docs/Server_Configuration_Guide/4/html/ch01s02s01.html)>. Acesso em: 23 out. 2016.

JENDROCK, Eric et al. **The Java EE 7 Tutorial**. 4. ed. Redwood City: Addison-Wesley Professional, v. 1, 2014.

JUBA, Salahaldin; VANNAHME, Achim; VOLKOV, Andrey. **Learning PostgreSQL**. Birmingham: Packt Publishing Ltd., 2015.

KEDAR, Seema. **Database Management System**. 1. ed. Pune: Technical Publications, 2009.

LOPES, Alfredo S.; SILVA, Marcelo de C.; GUILHERME, Luiz R. G. **Acidez do solo e calagem**. 3. ed. São Paulo: Associação Nacional para Difusão de Adubos - ANDA, 1991.

MATHA, Mahesh P. **JSP and SERVLETS: A Comprehensive Study**. Delhi: PHI Learning Pvt. Ltd., 2013. ISBN 9788120347458.

MILANI, André. **PostgreSQL - Guia do Programador**. São Paulo: Novatec Editora, 2008.

MILANI, André. **Programando para iPhone e iPad - 2ª Edição: Aprenda a construir aplicativos para o iOS**. 2. ed. São Paulo: Novatec Editora, 2014. ISBN 9788575223949.

NATIONS, Daniel. lifewire.com. **Lifewire**, 17 Outubro 2016. Disponível em: <<https://www.lifewire.com/what-is-a-web-application-3486637>>. Acesso em: 22 out. 2016.

OBE, Regina O.; HSU, Leo S. **PostgreSQL: Up and Running, Second Edition**. 2. ed. Sebastopol, CA: O'Reilly Media, Inc., 2014.

ORACLE. JavaServer Pages Standard Tag Library, 2016. Disponível em: <<http://www.oracle.com/technetwork/java/jstl-137486.html>>. Acesso em: 20 out. 2016.

PRESSMAN, Roger S. **Engenharia de Software**. 5. ed. Rio de Janeiro: Editora McGraw Hill, 2002.

PRESSMAN, Roger S. **Engenharia de Software Uma Abordagem Profissional**. Tradução de Ariovaldo Griesi e Mario M Fecchio. 7. ed. Porto Alegre: AMGH Editora Ltda, 2011.

RITCHIE, Colin. **Relational Database Principles**. 2. ed. London: Continuum, 2002.

SESHADRI, Shyam; BRAD, Green. **Desenvolvendo com AngularJS: Aumento de Produtividade com Aplicações Web Estruturadas**. São Paulo: Novatec Editora, 2014. ISBN 9788575224090.

SHENOY, Aravind; SOSSOU, Ulrich. **Learning Bootstrap**. Birmingham: Packt Publishing Ltd., 2014. ISBN 978-1-78216-184-4.

SHKLAR, Leon; ROSEN, Richard. **Web Application Architecture Principles, Protocols and Practices**. Chichester: John Wiley & Sons Ltd, 2004.

SHOEMAKER, H. E.; MCLEAN, E. O.; PRATT, P. F. Buffer Methods for Determining Lime Requirement of Soils With Appreciable Amounts of Extractable Aluminum. **Soil Science Society of America Journal**, Madison, WI, v. 25, p. 274-277, 1961.

SILVA, Maurício S. **Bootstrap 3.3.5 Aprenda a usar o framework Bootstrap para criar layouts CSS complexos e responsivos**. São Paulo: Novatec Editora Ltda. , 2015. ISBN 978-85-7522-460-1.

SILVA, Túlio M. D. **Software para recomendação de calagem e adubação do tomateiro no estado de Minas Gerais**. 84 f. Dissertação (Mestrado em Sistemas de Produção na Agropecuária) – Universidade José do Rosário Vellano, Alfenas, 2012

SPURLOCK, Jake. **Bootstrap**. 1. ed. Sebastopol: O'Reilly Media, Inc., 2013. ISBN 978-1-449-34391-0.

THE POSTGIS DEVELOPMENT GROUP. **PostGIS 2.2.1dev Manual**. 2016. 771 p. Disponível em: <<http://postgis.net/stuff/postgis-2.2.1.pdf>>. Acesso em: 20 out. 2016

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL 9.6.0 Documentation**. 2016. 3489 p. Disponível em: <<https://www.postgresql.org/docs/9.6/static/index.html>>. Acesso em: 20 out. 2016.

VAN RAIJ, Bernardo. A Capacidade De Troca De Cátions Das Frações Orgânica E Mineral Em Solos. **BRAGANTIA Boletim Científico do Instituto Agrônomo do Estado de São Paulo**. Campinas, v. 28, n. 8, mar. 1969.



**APÊNDICE A – Requisitos do Sistema**

Neste apêndice estão descritas funcionalidades, regras de negócio e restrições, ou seja, os requisitos implementados para a aplicação.

F1 Cadastrar produtores		Oculto ()		
<b>Descrição:</b> Faz o cadastro dos produtores no sistema				
<b>Requisitos não funcionais</b>				
Nome	Restrição	Categoria	Desejável	Permanente
NF 1.1 Identificação dos produtores	Cada produtor deve ser identificado pelo seu id	Interface	()	(x)
NF 1.2 Formulário	A tela de cadastro deve conter as seguintes informações para preencher, sendo todas obrigatórias: Nome, Fone e E-mail.	Interface	()	(x)

**Quadro 1 – Requisito Cadastrar produtores**

**Fonte:** Autoria própria.

F2 Cadastrar áreas		Oculto ()		
<b>Descrição:</b> Faz o cadastro das áreas no sistema				
<b>Requisitos não funcionais</b>				
Nome	Restrição	Categoria	Desejável	Permanente
NF 1.1 Identificação das áreas	Cada área deve ser identificada pelo seu id	Interface	()	(x)
NF 1.2 Formulário	A tela de cadastro deve conter um componente <i>select-list</i> para a seleção do nome do produtor que é o proprietário ou administrador da área em questão.	Interface	()	(x)

**Quadro 2 – Requisito Cadastrar áreas**

**Fonte:** Autoria própria.

F3 Inserir Laudos de Análise de Solo		Oculto ()		
<b>Descrição:</b> Faz o cadastro dos laudos de análise de solo no sistema				
<b>Requisitos não funcionais</b>				
Nome	Restrição	Categoria	Desejável	Permanente
NF 1.1 Identificação das áreas	Cada laudo deve ser identificado pelo seu id	Interface	()	(x)
NF 1.2 Formulário	A tela de cadastro deve conter três componentes <i>select-list</i> para a seleção da área onde foi coletada a amostra de solo, o nome do produtor que é o proprietário ou administrador da área em questão, e para a seleção do tipo de solo da análise.	Interface	()	(x)

**Quadro 3 – Requisito Cadastrar laudos**

**Fonte:** Autoria própria.

F4 Manter Base de dados	Oculto (x)
<b>Descrição:</b> Internamente os dados cadastrados pelo usuário, referentes à produtores, áreas e laudos, devem ser armazenados no banco de dados separados por tabelas respectivamente.	

**Quadro 4 – Manter Base de dados**

Fonte: Autoria própria.

F5 Visualizar Dados	Oculto ()			
<b>Descrição:</b> O sistema deve prover uma forma de exibir as tabelas criadas pelo usuário, com suas respectivas tuplas e linhas.				
<b>Requisitos não funcionais</b>				
Nome	Restrição	Categoria	Desejável	Permanente
NF 1.1 Exibição das tabelas	Para cada tabela do Banco de dados que se possui a necessidade de exibição, estas devem ser exibidas através das <i>tags</i> html <table> dentro de um arquivo jsp.	Interface	()	(x)
NF 1.2 Exibição de tabelas grandes	A fim de obter-se uma interface mais limpa com melhor visualização de dados, tabelas com muitos atributos como a tabela de Laudos, devem ter sua exibição dividida entre duas páginas, agrupando os dados relevantes para cada página.	Interface	(x)	()

**Quadro 5 – Visualizar Dados**

Fonte: Autoria própria.

F6 Visualizar Resultados dos cálculos para recomendação de calagem	Oculto ()			
<b>Descrição:</b> O sistema deve prover uma forma de exibir os resultados dos cálculos feitos pelas funções referentes à recomendação de calagem.				
<b>Requisitos não funcionais</b>				
Nome	Restrição	Categoria	Desejável	Permanente
NF 1.1 Exibição dos dados de cabeçalho	Os dados de cabeçalho (id do laudo, nome do produtor, área, talhão e índice SMP) devem ser exibidos no topo da página dentro de uma <i>tag</i> <table> pelos aspectos de fácil visualização e dinamismo.			
NF 1.2 Exibição do resultado gerado por cada critério de recomendação	O resultado de cada critério de recomendação deve ser exibido dentro de componentes do tipo Bootstrap panels.	Interface	()	(x)
NF 1.3 Exibição da tabela de recomendação pelo índice SMP	As informações da recomendação pelo índice SMP deve ser exibida dentro de uma <i>tag</i> <table>, e por ser um pouco extensa deve possuir os títulos das colunas em formato de campos onde o usuário possa digitar os valores e filtrar os resultados.	Interface	(x)	()

**Quadro 6 – Visualizar Resultados**

Fonte: Autoria própria.