UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

VINICIUS RIGON

Sistema Web de Acompanhamento do Projeto Agroflorestas

Trabalho de Conclusão

Prof. Dr. Leandro Krug Wives
Orientador

Porto Alegre, dezembro de 2013

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Vinicius Rigon,

Sistema Web de Acompanhamento do Projeto Agroflorestas /

Vinicius Rigon. – Porto Alegre: Graduação em Ciência da Computação da UFRGS, 2013.

57 f.: il.

Trabalho de Conclusão – Universidade Federal do Rio Grande do Sul. Curso de Bacharelado em Ciência da Computação, Porto Alegre, BR–RS, 2013. Orientador: Leandro Krug Wives.

1. Agrofloresta. 2. Sistema Web. 3. PHP. 4. MVC. I. Wives, Leandro Krug. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Carlos Alexandre Netto
Vice-Reitor: Prof. Rui Vicente Oppermann
Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb
Coordenador do CIC: Prof. Raul Fernando Weber
Bibliotecória-chefe do Instituto de Informática: Alexsander Ribeiro

AGRADECIMENTOS

Agradeço aos meus pais pelo apoio durante todos os anos da faculdade e da vida, sem eles eu não teria conseguido.

Aos meus amigos pelas horas de diversão e por entenderem o meu desaparecimento durante os últimos meses.

Ao Prof. Leandro Krug Wives, pela orientação deste trabalho. A Universidade Federal do Rio Grande do Sul, em especial ao Instituto de Informática, pelo ensino de excelência, de qualidade e gratuito.

SUMÁRIO

SUMÁRIO 5

LISTA DE FIGURAS 7

RESUMO 9

ABSTRACT 11

1 INTRODUÇÃO 15

2 LEVANTAMENTO DE REQUISITOS 17

2.1 Descrição do processo atual 18

2.2 Levantamento de Requisitos 18

2.2.1 Requisitos Funcionais 18

2.2.2 Requisitos Não Funcionais 19

3 MODELAGEM, PROJETO E DESENVOLVIMENTO DO SISTEMA . . 21

3.1 Diagramas de caso de uso 21

3.1.1 Administrativa 21

3.1.2 Agroflorestas 24

3.2 Diagrama Entidade-Relacionamento 26

3.2.1 Descrição das entidades 27

3.3 Arquitetura do Sistema 30

3.4 Tecnologias 30

3.4.1 PHP 30

3.4.2 MySQL 31

3.4.3 Apache 31

3.4.4 Yii (Yes It Is) 31

3.4.5 MVC (Model View Controller) 31
3.5 Modelagem das Classes 33
3.5.1 Detalhes do Framework Yii 33
3.5.2 Classes principais 33

4 TELAS DO SISTEMA 39
4.1 Login 39
4.2 Listagem de SAF 40
4.3 Criação de SAF 42
4.4 Edição de SAF 43
4.5 Tela de informações dos produtores 44
4.5.1 Família e Integrantes 45
4.5.2 Propriedade 46
4.5.3 Beneficiamento 47
4.5.4 Comercialização 48
4.5.5 Manejo 49
4.5.6 Rede Sociotécnica 51
4.5.7 Perspectivas 52
4.5.8 Visita 53
4.6 Resultados 54

5 CONCLUSÃO 55

REFERÊNCIAS 57

LISTA DE FIGURAS

2.1 Modelo incremental. Fonte: adaptado de Pressman (2006) 17

3.1 Diagrama de caso de uso da parte administrativa 22
3.2 Diagrama de caso de uso da parte administrativa 22
3.3 Diagrama de caso de uso agroflorestas 25
3.4 Diagrama ER da parte administrativa 27
3.5 Diagrama ER da parte do sistema 28
3.6 Arquitetura do sistema 30
3.7 Modelo MVC 31
3.8 Diagrama de classes 34

4.1 Login 39
4.2 Login inválido 40
4.3 Tela inicial do sistema, após login 40
4.4 Listagem de agroflorestas 40
4.5 Listagem de agroflorestas - pesquisando 41
4.6 Listagem de agroflorestas - pesquisa 41
4.7 Criação de SAF 42
4.8 Edição de SAF 43
4.9 Entrar na edição de SAF, informações dos produtores 44
4.10 Entrar na edição de SAF, informações dos produtores 44
4.11 Família 45
4.12 Integrantes - listagem 45
4.13 Integrantes - edição/inclusão 46
4.14 Propriedade 46
4.15 Beneficiamento 47
4.16 Comercialização 48

4.17 Práticas de manejo 49
4.18 Integração com o manejo 49
4.19 Listagem de animais manejados 50
4.20 Inclusão/Edição de animais 50
4.21 Listagem de vegetais manejados 50
4.22 Inclusão/Edição de vegetais 51
4.23 Rede Sociotécnica 51
4.24 Perspectivas 52
4.25 Visita 53

LISTA DE ABREVIATURAS E SIGLAS

MVC	Model View Controller
PHP	PHP Hypertext Preprocessor
UML	Unified Modeling Language
CRUD	Create Retrieve Update Delete
ER	Entidade Relacionamento
HTML	HyperText Markup Language
PGDR	Programa de Pós-Graduação em Desenvolvimento Rural
ATER	Assistência Técnica e Extensão Rural

RESUMO

Este trabalho descreve o desenvolvimento de um sistema Web para acompanhamento do projeto "Fortalecimento das Agroflorestas no Rio Grande do Sul", ou, simplesmente, Agroflorestas. O trabalho consiste de uma introdução e contextualização do projeto Agroflorestas, após é feito a descrição da arquitetura do sistema, bem como as tecnologias, modelagem do banco e os diagramas em UML assim como a descrição de casos de uso. São apresentadas algumas telas presentes no sistema, e por fim, a conclusão do trabalho.

Palavras-chave: Agrofloresta, Sistema Web, PHP, MVC.

Support System of the Agrofloresta Project

ABSTRACT

This article describes the development of a Web system to support the project entitled "Fortalecimento das Agroflorestas no Rio Grande do Sul", or for short, project Agroflorestas. This article begins with an introduction and contextualization of project Agroflorestas, after a description of the system architecture, technologies, database modeling, UML diagrams and use cases. Then it describes some screens of the system, and finally, the conclusion.

Keywords: PHP, MVC, Web, Agroflorestas.

1 INTRODUÇÃO

Este trabalho tem como contexto o projeto intitulado "Fortalecimento das Agroflorestas no Rio Grande do Sul"(ou, simplesmente, Agroflorestas) e também o projeto "Observatório", ambos em desenvolvimento no Programa de Pós-Graduação em Desenvolvimento Rural (PGDR) da UFRGS. O primeiro projeto mapeia propriedades rurais do Rio Grande do Sul, enquanto o segundo reúne informações sobre a segurança alimentar e nutricional inicialmente do RS e, futuramente, de outros estados do Brasil. As informações coletadas no primeiro contribuirão para o segundo ao indicar a proveniência dos alimentos.

O objetivo do projeto Agroflorestas é desenvolver e difundir o conhecimento e fortalecer as ações de Assistência Técnica e Extensão Rural (ATER) sobre agroflorestas no RS, implantadas por agricultores familiares, povos indígenas e comunidades tradicionais (também conhecidos por AFPICT), contribuindo para a afirmação de sua autonomia, segurança alimentar e nutricional (SAN), inserção em mercados institucionais e articulação em redes, partindo dos princípios da agroecologia.

Com base nessas pesquisas e ações, os pesquisadores do projeto pretendem elaborar uma proposta de orientação, visando fortalecer a produção agroflorestal como estratégia para a SAN de AFPICT no RS. Ademais, o projeto mapeará e articulará uma rede com os diversos atores (agricultores, familiares, quilombolas, indígenas, assentados, técnicos, extensionistas, estudantes, professores e suas organizações), que desenvolvem diferentes trabalhos com agroflorestas no RS, subdividindo o estado em suas regiões fitoecológicas. A partir disso, serão desenvolvidas, em algumas regiões, atividades que propiciem a construção e o compartilhamento de conhecimento entre os atores, visando o aperfeiçoamento de experiências consolidadas e o incentivo a novas iniciativas no estado.

O objetivo do trabalho aqui apresentado é modelar e implementar um sistema Web onde serão inseridos os dados coletados pelos pesquisadores Agroflorestas. Posteriormente esses dados serão visualizados em um Sistema de Apoio à Decisão (SAD). Os dados processados e visualizados pelo SAD precisam ser cadastrados, mas, no entanto, esse cadastro é feito diretamente dos dados coletados pelos pesquisadores para o banco de dados do SAD. A opção de sistema Web deve-se à necessidade de acesso simultâneo ao sistema de diferentes locais e pelos usuários do sistema.

Este trabalho está estruturado da seguinte maneira. No capítulo seguinte são descritos os requisitos funcionais e não funcionais do sistema. No Capítulo 3 é descrita a modelagem, tecnologias e arquitetura utilizadas no desenvolvimento do sistema. No Capítulo 4 são apresentados as telas do sistema, constituindo-se num guia de uso. E, por fim, no

Capítulo 5 apresenta-se a conclusão do trabalho.

2 LEVANTAMENTO DE REQUISITOS

Neste capítulo são descritas o processo atual, contextualizando este trabalho, incluindo as necessidades e o levantamento dos requisitos funcionais e não funcionais para o sistema desenvolvido. Cabe salientar que seguiu-se uma metodologia clássica de desenvolvimento de software, descrita em [Pressman, 2006], envolvendo reuniões para a compreensão do problema e definição dos requisitos. Com base nisso, foram elaborados casos de uso, os quais foram posteriormente implementados.

A metodologia seguida segue o modelo incremental de desenvolvimento, onde aplicam-se elementos do modelo cascata porém de maneira iterativa, entregando versões do sistema com mais funcionalidades, conforme essas versões vão sendo implementadas. A figura 2.1 exemplifica esse modelo.

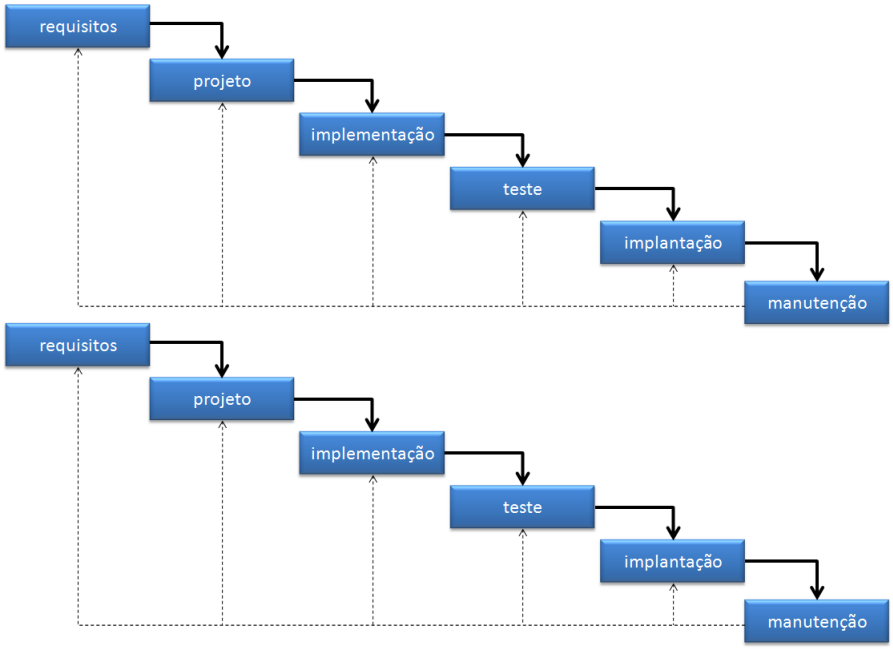

Figura 2.1: Modelo incremental.
Fonte: adaptado de Pressman (2006)

2.1 Descrição do processo atual

Conforme descrito na introdução deste documento, o objetivo consiste em desenvolver um sistema Web que permita aos pesquisadores envolvidos no projeto Agroflorestas possam cadastrar e gerenciar dados e informações relacionadas com o mesmo. O primeiro passo consistiu, portanto, na compreensão de como se dá o processo de coleta de dados e como os mesmos são cadastrados e analisados, para tanto, foram realizadas entrevistas com os pesquisadores envolvidos no projeto.

Com base nas entrevistas com os membros do projeto, identificou-se que o instrumento de coleta de dados usado por eles é um questionário que é aplicado durante entrevistas feitas com proprietários rurais. Nesse sentido, os pesquisadores ou bolsistas vão até as propriedades identificadas por eles como sendo relevantes para o projeto (i.e., aquelas que possuem sistemas agroflorestais) e entrevistam seus proprietários (i.e., os produtores rurais). A intenção primária de tal questionário é obter dados sobre as propriedades de tais produtores e, assim, verificar a produtividade de suas propriedades. As respostas dadas pelos produtores são anotadas e transcritas para um documento Word (.doc), em forma de questões abertas. Devido ao fato das respostas serem livres, escritas em linguagem natural – português, não há uma padronização das respostas, o que dificulta a aplicação de análise estatística tradicional (descritiva e quantitativa). Por esse motivo, os mesmos criaram uma planilha (Excel) onde os dados são novamente transcritos, sendo "transformados" em questões fechadas.

Pelo fato de haver um único arquivo Excel (uma única planilha) e também pelo fato de tal planilha não possuir recursos que facilitem a busca de informações, os pesquisadores apresentaram a necessidade de desenvolvimento de um sistema Web que os permitissem cadastrar, catalogar os dados coletados (um ambiente CRUD [1]). Dessa forma, vários pesquisadores poderiam fazer a inserção dos dados simultaneamente, além de poderem consultar e analisar os dados cadastrados.

2.2 Levantamento de Requisitos

Nessa seção são descritos o os requisitos funcionais e não-funcionais do sistema, os quais foram obtidos através de reuniões quinzenais com o grupo responsável pelo projeto Agroflorestas.

As reuniões, feitas quinzenalmente, foram propostas para o levantamento dos requisitos do sistema, onde eram feitas as anotações das necessidades dos usuários do sistema. De posse das anotações, os requisitos funcionais e não funcionais, os casos de uso, diagramas UML (Unified Modeling Language) foram feitos, conforme indicado por [Pressman, 2006].

2.2.1 Requisitos Funcionais

Para atender os requisitos do sistema Agroflorestas, apresentam-se listadas abaixo as principais características que o sistema deve fornecer.

[1] Acrônimo para Create Retrieve Update Delete, que são operações básicas para persistência de dados em um banco de dados. Mais informações em [Övergaard, 2005]

- O acesso ao sistema deverá ser feito apenas por usuários cadastrados. Cada usuário terá um perfil, que permitirá o acesso somente a determinadas funções do sistema;
- O sistema deve permitir cadastrar, listar, alterar e excluir agroflorestas, família, integrantes, comercialização, manejo, beneficiamento, visita, animais e vegetais;
- O sistema deverá emitir relatórios;
- O sistema deve fazer importação e exportação dos dados cadastrados;
- O sistema deve fazer backup do banco de dados;
- O sistema terá integração com o sistema desenvolvido por [Rigo, 2013];
- O sistema fará upload de fotos e do questionário com as respostas feitos na pesquisa de campo;

2.2.2 Requisitos Não Funcionais

Os requisitos não funcionais nesse sistema dizem respeito às tecnologias usadas, usabilidade, disponibilidade e manutenibilidade.

- O sistema deve ser desenvolvido no sistema Linux, devido a integração com outros sistemas desenvolvidos para o cliente.
- O sistema deve ser desenvolvido usando PHP.
- O sistema deve fazer uso de um framework PHP.
- O sistema deve se comunicar com banco de dados MySQL.
- O sistema deve usar o servidor de aplicação Apache.
- O sistema deve possuir uma interface de fácil utilização pelo usuário.

3 MODELAGEM, PROJETO E DESENVOLVIMENTO DO SISTEMA

A seguir são descritos os casos de uso do sistema através de diagramas de caso de uso. Também é descrita a modelagem do banco desenvolvido, através de um diagrama ER (Entidade Relacionamento) das tabelas de dados presentes no sistema. Finalmente, é descrita a arquitetura do sistema e as tecnologias usadas para implementá-lo.

3.1 Diagramas de caso de uso

O sistema pode ser divido em dois: a parte administrativa, onde são cadastrados os usuários, grupos de acesso e suas permissões; e a parte de gerenciamento de agroflorestas propriamente ditas.

Tendo em vista que o sistema possui basicamente operações de CRUD (Create Retrieve Update Delete) e são muito semelhantes entre si, as descrições dos casos de uso podem ser simplificadas usando o conceito de *template*.

A separação em duas áreas é necessária para permitir o controle de acesso dos usuários ao sistema, pois somente usuários com perfil específico podem cadastrar e alterar registros dos produtores e homologá-los antes da publicação.

Devido à essa separação, primeiramente detalha-se o controle de acesso e, após, o registro dos dados dos produtores.

3.1.1 Administrativa

Na Figura 3.1 está a representação dos casos de uso gerenciamento dos usuários, grupos e permissões. O ator será o administrador que terá permissão para gerenciar os usuários do sistema e fazer backup dos dados.

A seguir é descrito somente o caso de uso de usuário, pois gerenciar grupos e permissões são similares.

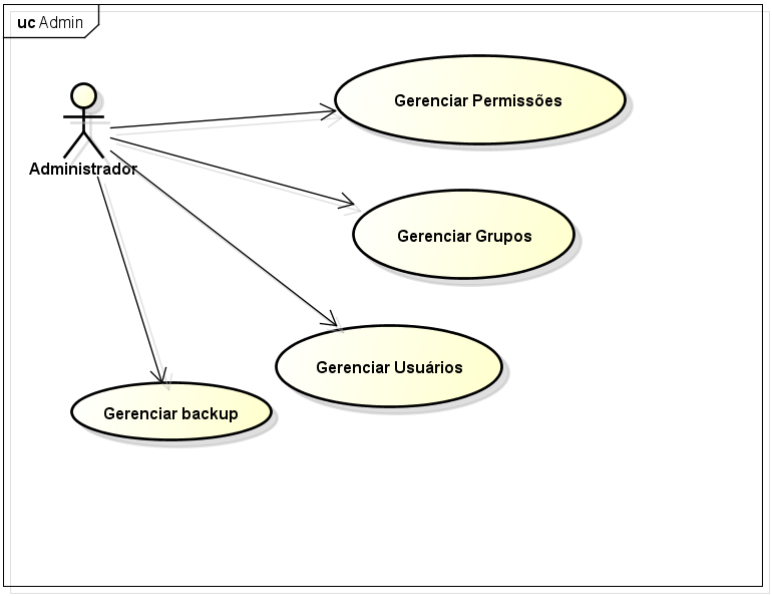

Figura 3.1: Diagrama de caso de uso da parte administrativa

3.1.1.1 Caso de uso: Gerenciar Usuários

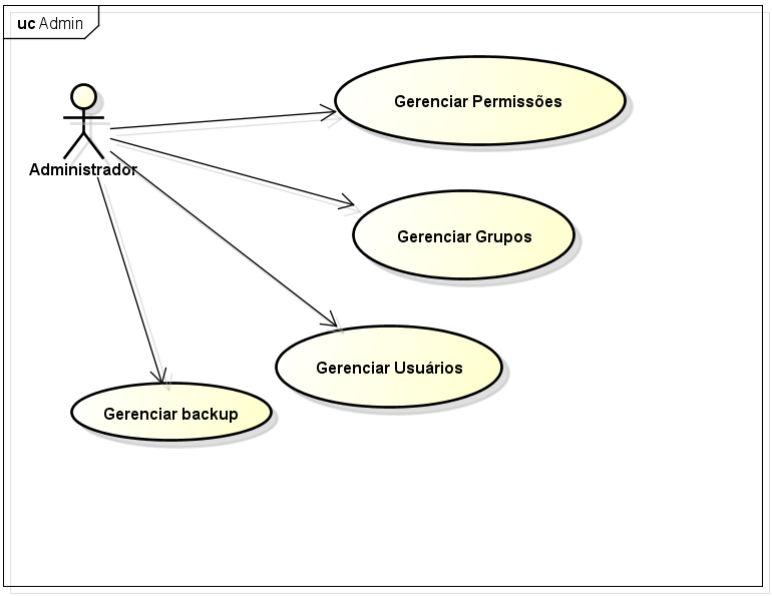

Figura 3.2: Diagrama de caso de uso da parte administrativa

3.1.1.2 Caso de uso: Gerenciar Usuários

Caso de uso: Gerenciar usuário
Ator: Administrador
Descrição resumida: Este caso de uso descreve o controle para listar, criar, alterar e excluir usuários.
Fluxo de eventos:
Fluxo básico

1. Administrador seleciona a opção Usuário do menu
2. Sistema apresenta a lista de usuários cadastrados

Fluxos alternativos

A **Cadastrar usuários**

1 Administrador clica no link 'Criar usuário'
2 Administrador entra com os dados do novo usuário
3 Sistema salva o novo usuário e retorna para a listagem

B **Consultar usuários**

1 Administrador entra com os filtros da pesquisa
2 Sistema retorna uma lista com os resultados

C **Atualizar usuários**

1 Administrador clica no ícone de edição na listagem
2 Na tela de edição, administrador altera os dados caso necessário
3 Administrador clica em salvar e sistema retorna para a listagem

D **Deletar usuários**

1 Administrador clica no ícone de deletar na listagem
3 Caso sim, sistema exclui o usuário e atualiza a listagem
4 Caso não, sistema não faz nada.

3.1.1.3 Caso de uso: Backup

Caso de uso: Gerenciar backup
Ator: Administrador
Descrição resumida: Este caso de uso descreve a criação e restauração de backups.
Fluxo de eventos:
Fluxo básico

1. Administrador seleciona a opção de backup no menu

2. Sistema apresenta botões de restaurar e criar backup

Fluxos alternativos

A **Criar backup**

1 Administrador clica no botão 'Criar'

2 Sistema cria o backup, salvando os dados em um arquivo .xlsx

B **Restaurar backup**

1 Administrador clica no botão 'Restaurar'

2 Administrador faz o upload de um arquivo .xlsx contendo o backup do sistema

3 Sistema faz a restauração do sistema

4 Sistema mostra mensagem de sucesso caso a restauração com sucesso

5 Sistema mostra mensagem de erro caso não consiga fazer o backup

3.1.2 Agroflorestas

A Figura 3.2 apresenta o diagrama dos casos de uso essenciais do sistema. Como os casos de uso são semelhantes, somente a descrição mais detalhada de alguns deles é necessária, pois os fluxos dos mesmos são operações CRUD.

3.1.2.1 Caso de uso: Gerenciar Agroflorestas

Caso de uso: Gerenciar agroflorestas
Ator: Pesquisador
Descrição resumida: Este caso de uso descreve o controle para listar, criar, alterar e excluir agroflorestas. Além de opções para cadastrar dados adicionais
Fluxo de eventos:
Fluxo básico

1. Pesquisador seleciona a opção SAF do menu

2. Sistema apresenta a lista de agroflorestas cadastradas

Fluxos alternativos

A **Cadastrar SAF**

1 Pesquisador clica no link 'Criar Agrofloresta'

2 Pesquisador entra com os dados da nova agrofloresta

3 Sistema salva a nova agrofloresta e retorna para a listagem

B **Consultar agroflorestas**

1 Pesquisador entra com os filtros da pesquisa

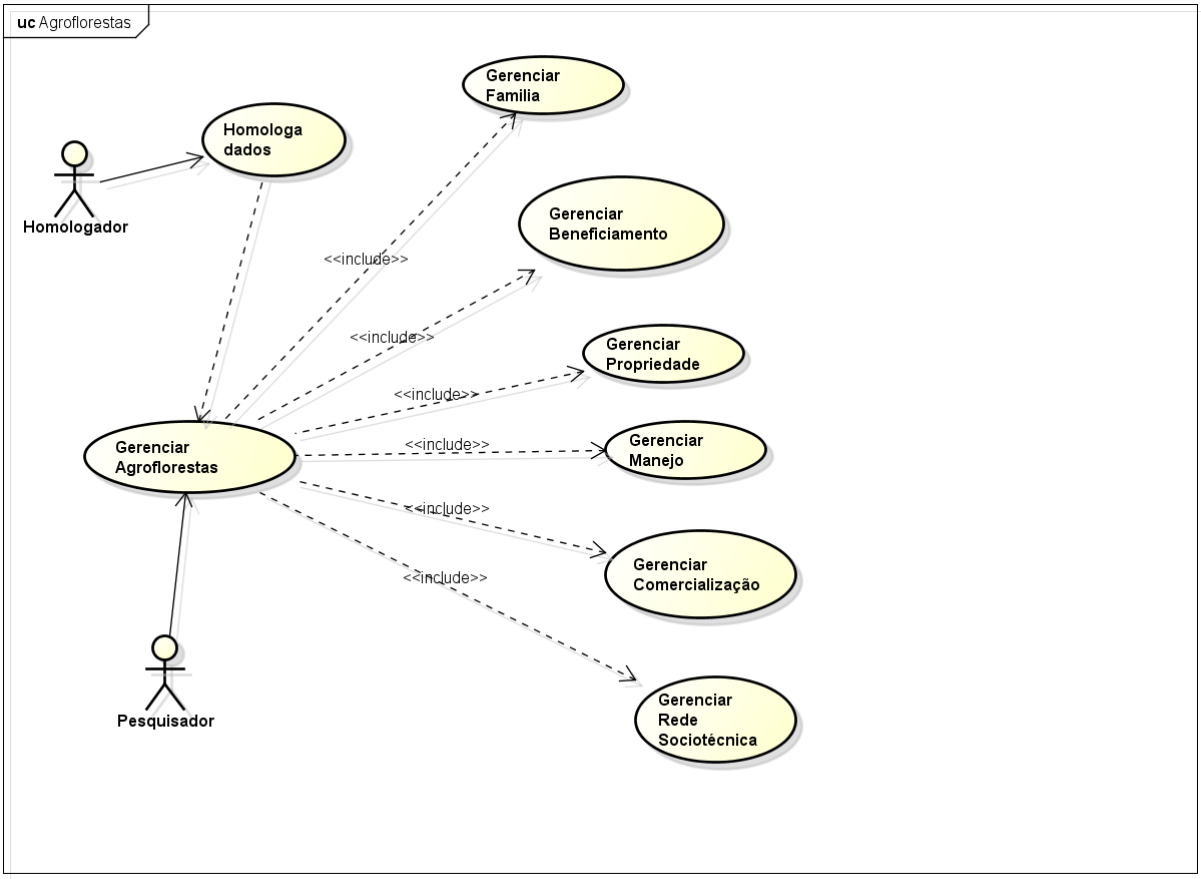

Figura 3.3: Diagrama de caso de uso agroflorestas

2 Sistema retorna uma lista com os resultados

C **Atualizar agroflorestas**

1 Pesquisador clica no ícone de edição na listagem

2 Na tela de edição, pesquisador altera os dados caso necessário

3 Pesquisador clica em salvar e sistema retorna para a listagem

D **Deletar agroflorestas**

1 Pesquisador clica no ícone de deletar na listagem

3 Caso sim, sistema exclui o usuário e atualiza a listagem

4 Caso não, sistema não faz nada.

E **Visualizar agroflorestas**

1 Pesquisador clica no ícone de visualizar na listagem

3 Sistema abre a tela de visualização com os dados da agrofloresta, exibe também opções de menu para incluir os dados adicionais

3.1.2.2 Caso de uso: Homologar Dados

Caso de uso: Homologar Dados
Ator: Homologador
Descrição resumida: Este caso de uso descreve a homologação dos dados antes de poder entrar na visualização pública
Fluxo de eventos:
Fluxo básico

1. Homologador seleciona a opção 'SAF' do menu
2. Sistema apresenta na listagem, agroflorestas não homologadas
3. Homologador clica no ícone de homologação
4. Sistema abre tela com a agrofloresta a ser homologada, exibindo todos os dados
5. Homologador clica no botão para homologar os dados, sistema exibe mensagem de confirmação

Fluxos alternativos

A **Confirmar homologação**

1 Sistema salva a agrofloresta como homologada
2 Sistema atualiza banco de dados públicos com os dados da nova agrofloresta homologada
3 Sistema retorna para a listagem de agroflorestas não homologadas

B **Cancelar homologação**

1 Sistema retorna para a listagem de agroflorestas não homologadas

3.2 Diagrama Entidade-Relacionamento

Na Figura 3.3 está representado o diagrama ER das tabelas do banco referente ao controle de acesso, ou seja, da parte administrativa do sistema, na figura 3.4 por sua vez, é a representação da parte dos registro dos dados dos produtores. A ferramenta usada para criar o modelo foi o MySQL Workbench, pois adota uma notação semelhante a *UML*.

A modelagem da figura 3.3 é composto pela tabela *Usuario* que contém o identificador único, o nome, login e a senha do usuário e grupo pela qual ele pertence, possibilitando o controle de acesso desse usuário conforme o grupo pelo qual ele pertence. Na tabela *Grupo* contém somente o identificador e o nome do grupo. A tabela *GrupoPermissao* relaciona os grupos com as permissões de acesso, já tabela *menu* permite montar os menus do sistema de acordo com as permissões do grupo. E finalmente, a tabela *Permissao* contém as permissões disponíveis no sistema.

A parte de sistema modelada na figura 3.4 possui como principal entidade a tabela *Agrofloresta* onde os dados de localização da propriedade são cadastrados, após esse cadastro, as outras informações do produtor podem ser inseridas no sistema.

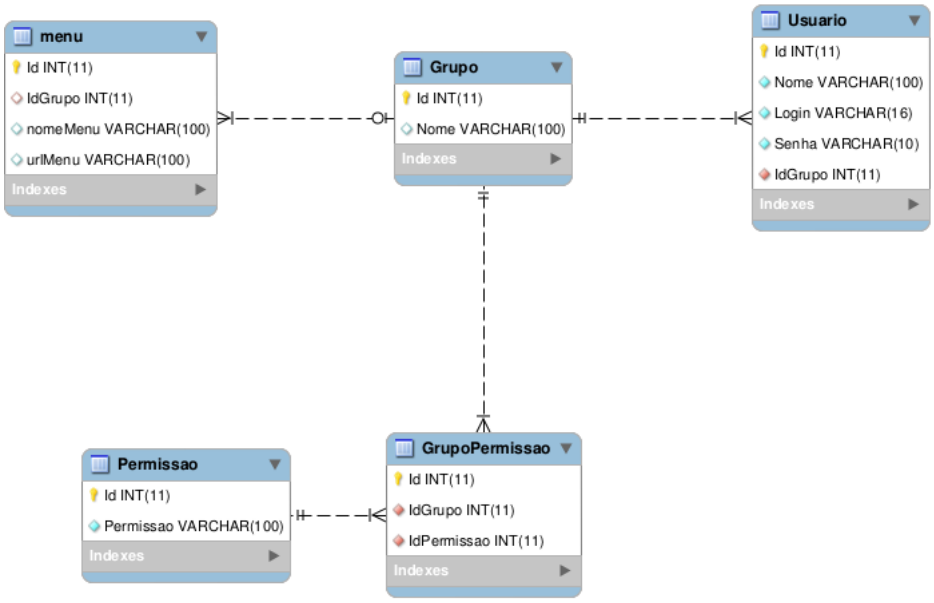

Figura 3.4: Diagrama ER da parte administrativa

3.2.1 Descrição das entidades

Abaixo segue e descrição e os objetivos das principais entidades vistas no diagrama da figura 3.4

3.2.1.1 Agroflorestas

Essa entidade é a entidade principal do sistema, nela está a informação da localização da agrofloresta, e é usada como entidade-pai das outras entidades vistas no diagrama. Portanto, a entidade permite saber os locais onde estão as agroflorestas no Rio Grande do Sul.

3.2.1.2 Propriedade

Nessa entidade é cadastrada as informações relativas à propriedade, nela podemos encontrar as informações sobre área, total e usadas como agroflorestas, sendo assim, os pesquisadores saberão a área total de agroflorestas no estado e quais as percentagem delas em relação à área total da propriedade.

3.2.1.3 Família e Integrantes

Essas entidades juntas fazem o cadastro de todas as informações relativas a família do proprietário, tais como, local de origem, renda, nome, parentesco com o proprietário, a ocupação, o tempo de residência na propriedade. Essa entidade permite ao pesquisador saber quais os locais onde possui mais integrantes, a renda deles, e assim permitir uma ação maior, caso a renda seja baixa, por exemplo.

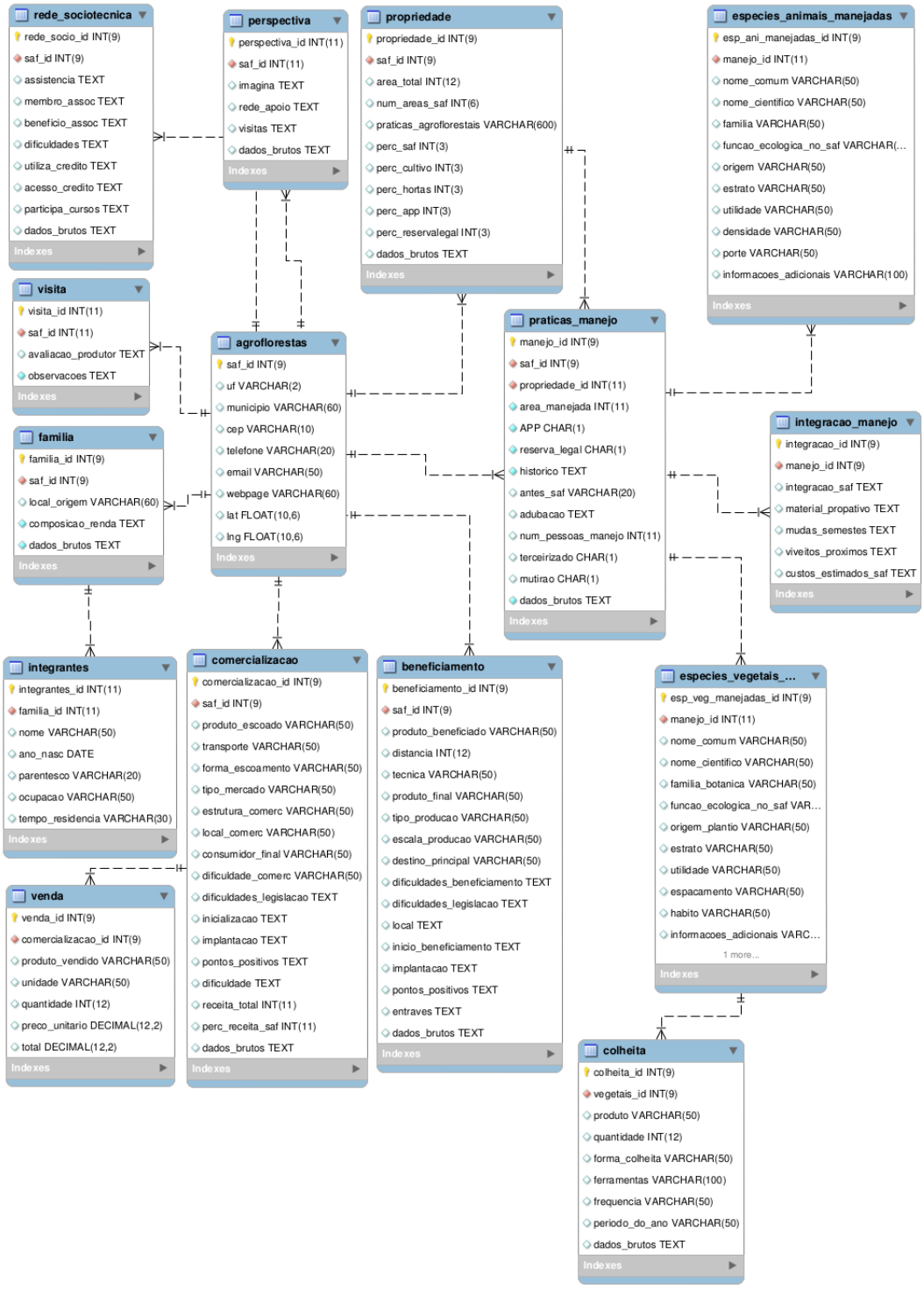

Figura 3.5: Diagrama ER da parte do sistema

3.2.1.4 Comercialização e Venda

As informações contidas nessas entidades são relativas a obtenção de renda a partir da venda dos produtos cultivados na propriedade, nela estão as informações sobre a receita advinda da venda, quais produtos vendidos, o total da venda. Essa entidade serve para os pesquisadores saberem quais os produtos que estão sendo comercializados e o total de renda obtido com a venda.

3.2.1.5 Manejo

As entidades *praticas_manejo*, *integracao_manejo*, *especies_animais_manejadas*, *especies_vegetais_manejadas* e *colheita* fornecem informações de como a propriedade é utilizada para gerar a renda. Nas entidades *especies_vegetais_manejadas* e *colheita*, por exemplo, possui as informações referentes aos vegetais presentes na propriedade e como elas geram a renda para a família. O mesmo acontece com a entidade de animais. Com as informações presentes nessas entidades os pesquisadores podem saber quais são as agroflorestas que cultivam banana, por exemplo.

3.3 Arquitetura do Sistema

Uma aplicação Web possui arquitetura cliente-servidor. Essa arquitetura é definida como uma máquina cliente requisitando acesso aos recursos de uma máquina servidor, o servidor então processa essas requisições e envia as respostas para as máquinas cliente. A arquitetura usada nesse sistema é a arquitetura cliente-servidor de 3 camadas: apresentação, servidor de aplicação e servidor de banco de dados. Para facilitar a gerência do sistema o servidor de banco de dados encontra-se na mesma máquina do servidor de aplicação, conforme ilustrado na Figura 3.5.

Os dados fornecidos pelos produtores são persistidos em um banco de dados, porém alguns dados dos produtores não podem ser exibidos no Sistema de Apoio à Decisão (SAD) conforme descrito em [Turban et al., 2005], por conter informações, no entender deles, sigilosos, com isso o sistema possui outro banco de dados usado para armazenar os registros para que então possam ser exibidos no SAD.

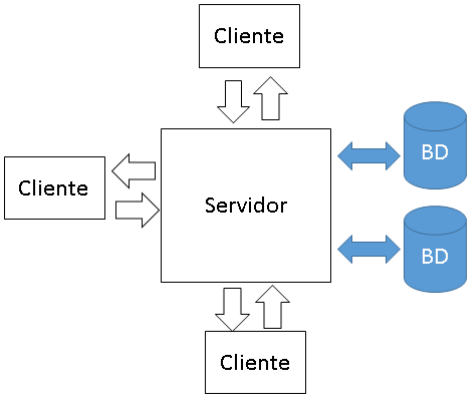

Figura 3.6: Arquitetura do sistema

Essa arquitetura é usada para facilitar a integração com outros sistemas já usados no projeto Agroflorestas, como por exemplo, o *data warehouse* usado para ajudar na tomada de decisões, conforme descrito em [Rigo, 2013].

3.4 Tecnologias

As tecnologias usadas no desenvolvimento desse projeto está descrita nas subseções abaixo. Para implementar uma arquitetura de 3 camadas, foi usado um framework PHP em MVC, com isso, a separação das camadas descrita na arquitetura é satisfeita. Tal arquitetura é explicada na seção 3.4.5.

3.4.1 PHP

A linguagem de programação PHP (PHP Hypertext Preprocessor), é uma linguagem de programação interpretada criada em 1995 usada principalmente para desenvolvimento web, porém é usada para outros fins, tais como processamento de strings. É uma linguagem usada para criar conteúdo dinâmico na web, por embutir código entre as tags HTML(HyperText Markup Language). Outros pontos fortes da linguagem são orienta-

ção à objetos, portabilidade, velocidade e tipagem dinâmica. Mais informações sobre a linguagem em [Welling, 2008].

3.4.2 MySQL

O MySQL é um sistema de banco de dados relacional (SGBD) de código aberto, fortemente ligado com o PHP, conhecido por ser um SGBD robusto, performático, e fácil de gerenciar.

3.4.3 Apache

Um servidor web, também conhecido como servidor de aplicação, é o responsável por fornecer as páginas, estáticas ou dinâmicas, requisitadas por clientes, geralmente navegadores. Isso é feito respondendo as requisições feitas por esses clientes. Normalmente, as páginas HTML possuem um script embutido, esse script é executado dentro do servidor, resultando em uma página que será enviada para o cliente.

3.4.4 Yii (Yes It Is)

Hoje em dia o desenvolvimento web está em um nível de maturidade muito elevado, com isso surgiu framework web, usados para facilitar o desenvolvimento fornecendo ferramentas e recursos criando aplicações rapidamente. O Yii é um desses frameworks, criado em 2008, conhecido por ser um framework eficiente, modular, fácil de desenvolver. Podendo ser encontrado em [Yii, 2013]

3.4.5 MVC (Model View Controller)

A metodologia descrita em [Krasner, 1988] foi proposta para modelar a interação do usuário em sistemas desenvolvidos em *Smalltalk-80*, o MVC, cuja sigla significa, *Model-View-Controller*, é uma metodologia onde a aplicação é dividida em três partes: *Model*, *View* e *Controller*, sendo a *Model* a parte responsável pelo domínio da aplicação, a *View* pela interação com o usuário e a *Controller* o envio de mensagens entre a Model a View. Abaixo veremos a descrição de cada uma das camadas, no contexto do framework usado, A Figura 3.6 ilustra essa separação.

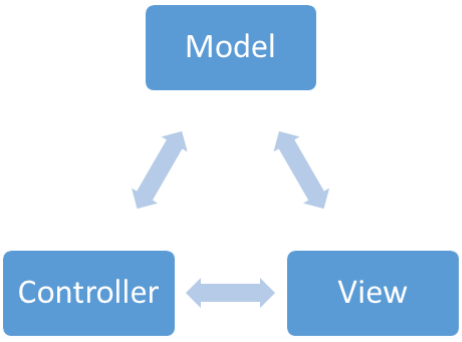

Figura 3.7: Modelo MVC

3.4.5.1 Model

No contexto do Yii, conforme a descrição encontrada em [Winesett, 2012], a *Model* é responsável por manter estado e encapsula as regras de negócio aplicadas os dados que definem esse estado, manipulando-os a partir da base. A *Model* é representado como uma série de atributos e validação dos mesmos, os dados dos atributos são, normalmente, um registro do banco de dados.

A implementação da *Model* em Yii é através de dois tipos: `CFormModel` onde os dados são obtidos de um *form*, encapsula toda a lógica para validação dos campos. Salva os dados na memória, ou através de um modelo *active record* salva os dados no banco.

A classe `CActiveRecord`, faz as mesmas coisas que a anterior, porém as tabelas são representadas como objetos, aproximando as tabelas do banco da orientação à objetos, sendo assim as colunas de uma tabela são representadas como propriedades de uma classe.

3.4.5.2 View

Responsável por renderizar a interface para o usuário, frequentemente uma representação dos dados da *Model*. Uma *View* é um *script* contendo código PHP simples. Por boas práticas, lógicas mais complexas o código deve ser colocado na *Model*, caso seja relacionado com os dados ou na *Controller* para regras de negócio mais gerais.

3.4.5.3 Controller

A *Controller* serve como camada intermediária entre a *Model* e a *View*. Também é responsável por mapear as requisições em ações dentro do sistema, sendo assim ela obtém as entradas do usuário, interagindo com a *Model* e atualizando as informações que serão mostradas na *View*.

3.5 Modelagem das Classes

A partir do modelo conceitual do banco de dados o próximo passo é a modelagem das classes usadas para implementar o sistema. As classes no sistema seguem o padrão definido pelo framework mencionado anteriormente.

3.5.1 Detalhes do Framework Yii

O framework utilizado na implementação do sistema fornece, atráves do modelo MVC, a separação do código das camadas envolvidas. Desse modo, as *Models*, responsáveis pelas tabelas do banco de dados herdam da classe `CActiveRecord`, e as *Controllers* herdam da classe `Controller`. Ambas as classes disponibilizam métodos e atributos que podem ser usados pelas classes filhas, algumas delas são de uso obrigatório para o funcionamento correto do sistema. Nos modelos do sistema, os atributos mais importantes são aqueles responsáveis por vincular as tabelas do banco de dados e estabelecer seus relacionamentos, e os que realizam a validação dos dados. No primeiro caso, tem-se o método `relations` onde: `HAS_MANY` define a relação *tem muitos*, quando a chave estrangeira está no outro modelo associado; `BELONGS_TO`, seu caso complementar, quando a chave estrangeira está no modelo atual, que define a relação *pertence a*. Para o segundo caso, o método `rules` define as validações dos dados, antes de serem salvos durante uma inclusão ou alteração. A *Controller* possui os métodos necessários para carregar os dados da *Model* e os métodos para a visualização através das *Views*, são eles: `loadModel`, usado para carregar o objeto; `actionView`, `actionEdit`, `actionCreate`, `actionUpdate`, `actionDelete`.

3.5.2 Classes principais

Na Figura 3.7 encontra-se o diagrama de classes do sistema, percebe-se que as classes principais são agroflorestas e praticas_manejo, apesar disso irei descrever somente a classe agroflorestas pois todas as classes do sistema são semelhantes. Por brevidade somente os atributos de chave primária e estrangeira são mostrados, porém esses atributos são exatamente os mostrados na modelagem do banco.

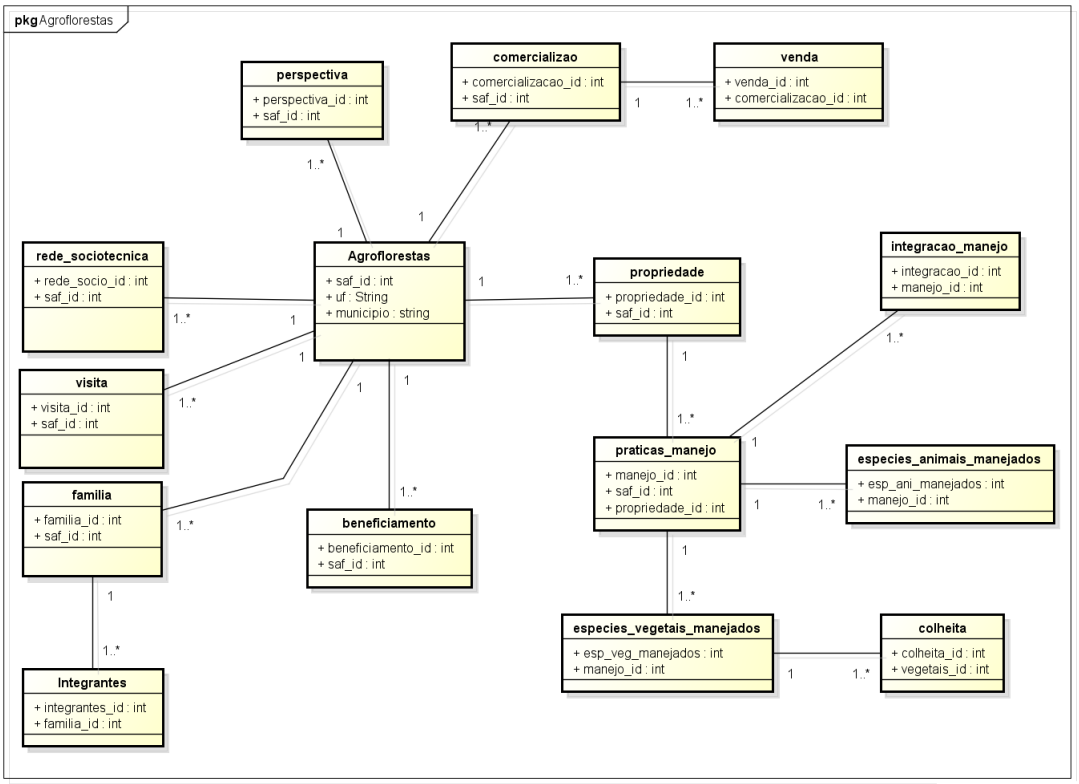

Figura 3.8: Diagrama de classes

3.5.2.1 Agroflorestas

Esta classe realiza todos as operações CRUD das agroflorestas. Essa implementação é feita através dos métodos: `actionEdit`, `actionCreate`, `actionUpdate`, `actionDelete`

No código abaixo, tem-se a definição da classe `Agroflorestas`, filha da classe `CActiveRecord`, na qual são definidos os relacionamentos com as demais tabelas e regras de validação durante a inclusão de uma nova tupla de agrofloresta.

```
class Agrofloresta extends CActiveRecord
{
  /**
   * @return array validation rules for model attributes.
   */
  public function rules()
  {
    // NOTE: you should only define rules for those attributes that
    // will receive user inputs.
    return array(
      array('lat,␣lng', 'numerical'),
      array('uf', 'length', 'max'=>2),
      array('municipio,␣webpage', 'length', 'max'=>60),
      array('cep', 'length', 'max'=>10),
      array('telefone', 'length', 'max'=>20),
      array('email', 'length', 'max'=>50),
      // The following rule is used by search().
      // Please remove those attributes that should not be searched.
```

```
      array('saf_id,␣uf,␣municipio,␣cep,␣telefone,␣email,␣webpage,␣lat,
          ␣lng', 'safe', 'on'=>'search'),
    );
  }

  /**
   * @return array relational rules.
   */
  public function relations()
  {
    // NOTE: you may need to adjust the relation name and the related
    // class name for the relations automatically generated below.
    return array(
      'beneficiamentos' => array(self::HAS_MANY, 'Beneficiamento', '
          saf_id'),
      'comercializacaos' => array(self::HAS_MANY, 'Comercializacao', '
          saf_id'),
      'especiesAnimaisManejadases' => array(self::HAS_MANY, '
          EspeciesAnimaisManejadas', 'saf_id'),
      'especiesVegetaisManejadases' => array(self::HAS_MANY, '
          EspeciesVegetaisManejadas', 'saf_id'),
      'familias' => array(self::HAS_MANY, 'Familia', 'saf_id'),
      'perpectivases' => array(self::HAS_MANY, 'Perpectivas', 'saf_id')
          ,
      'praticasManejos' => array(self::HAS_MANY, 'PraticasManejo', '
          saf_id'),
      'propriedades' => array(self::HAS_MANY, 'Propriedade', 'saf_id'),
      'redeSociotecnicas' => array(self::HAS_MANY, 'RedeSociotecnica',
          'saf_id'),
    );
  }

}
```

Na implementação da controller da agrofloresta a seguir, apresenta-se a definição da classe na qual, conforme mencionado anteriormente, é uma herança de `Controller`. Além disso, é definida a função *index*, a qual representa o método padrão da classe, que neste caso lista todas as agroflorestas cadastradas no sistema.

```
class AgroflorestaController extends Controller
{
  /**
  * Lists all models.
  */
  public function actionIndex()
  {
    $model=new Agrofloresta('search');
    $model->unsetAttributes();  // clear any default values

    if(isset($_GET['Agrofloresta']))
      $model->attributes=$_GET['Agrofloresta'];

    $this->render('index',array(
      'model'=>$model,
    ));
  }
}
```

Na sequência é definido os métodos responsáveis por criar, atualizar e excluir uma agrofloresta.

```
/**
 * Creates a new model.
 * If creation is successful, the browser will be redirected to the '
    view' page.
 */
public function actionCreate()
{
  $model=new Agrofloresta;

  // Uncomment the following line if AJAX validation is needed
  // $this->performAjaxValidation($model);

  if(isset($_POST['Agrofloresta']))
  {
    $model->attributes=$_POST['Agrofloresta'];
    if($model->save())
      $this->redirect(array('index','id'=>$model->saf_id));
  }

  $this->render('create',array(
    'model'=>$model,
  ));
}

/**
 * Updates a particular model.
 * If update is successful, the browser will be redirected to the '
    view' page.
 * @param integer $id the ID of the model to be updated
 */
public function actionUpdate($id)
{
  $model=$this->loadModel($id);

  // Uncomment the following line if AJAX validation is needed
  // $this->performAjaxValidation($model);

  if(isset($_POST['Agrofloresta']))
  {
    $model->attributes=$_POST['Agrofloresta'];
    if($model->save())
      $this->redirect(array('view','id'=>$model->saf_id));
  }

  $this->render('update',array(
    'model'=>$model,
  ));
}

/**
 * Deletes a particular model.
 * If deletion is successful, the browser will be redirected to the '
    admin' page.
 * @param integer $id the ID of the model to be deleted
```

```
 */
public function actionDelete($id)
{
  $this->loadModel($id)->delete();

  // if AJAX request (triggered by deletion via admin grid view), we
      should not redirect the browser
  if(!isset($_GET['ajax']))
    $this->redirect(isset($_POST['returnUrl']) ? $_POST['returnUrl']
        : array('admin'));
}
```

4 TELAS DO SISTEMA

Neste capítulo são apresentadas as principais telas do sistema desenvolvido, demonstrando seu fluxo normal de operação. De forma geral, são descritos os passos e ações a serem feitas pelos usuários para realizar as tarefas de gerenciar agroflorestas e demais informações dos produtores.

4.1 Login

Na figura 4.1 é mostrada a tela de login do sistema, necessária para controle de acesso. Entrando com o login e senha corretos, o sistema permite a entrada do usuário.

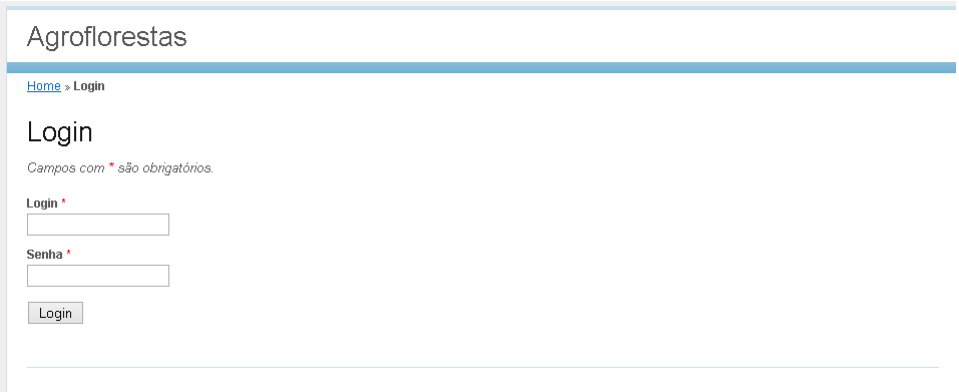

Figura 4.1: Login

Porém, se entrar com login/senha inválidos o sistema redireciona para a tela mostrada na figura 4.2.

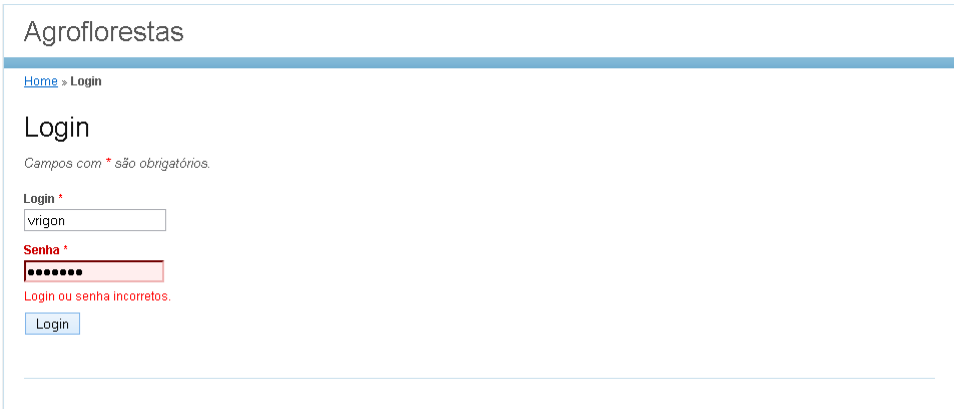

Figura 4.2: Login inválido

4.2 Listagem de SAF

Após o login feito com sucesso o sistema redireciona para a tela 4.3, nela vemos o menu contendo as opções de acesso.

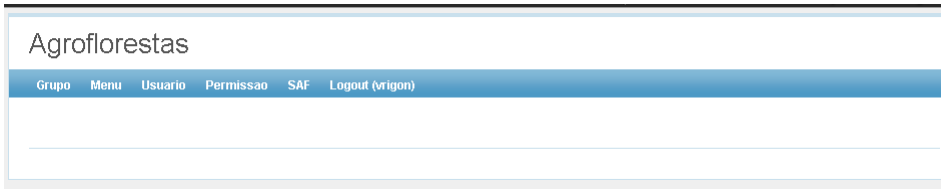

Figura 4.3: Tela inicial do sistema, após login

Ao clicar no item 'SAF', o sistema redireciona para a tela mostrada na figura 4.4, onde é possível cadastrar, editar e excluir agroflorestas. O menu está presente em todas as telas do sistema, sendo possível acessar as funcionalidades de todo o site.

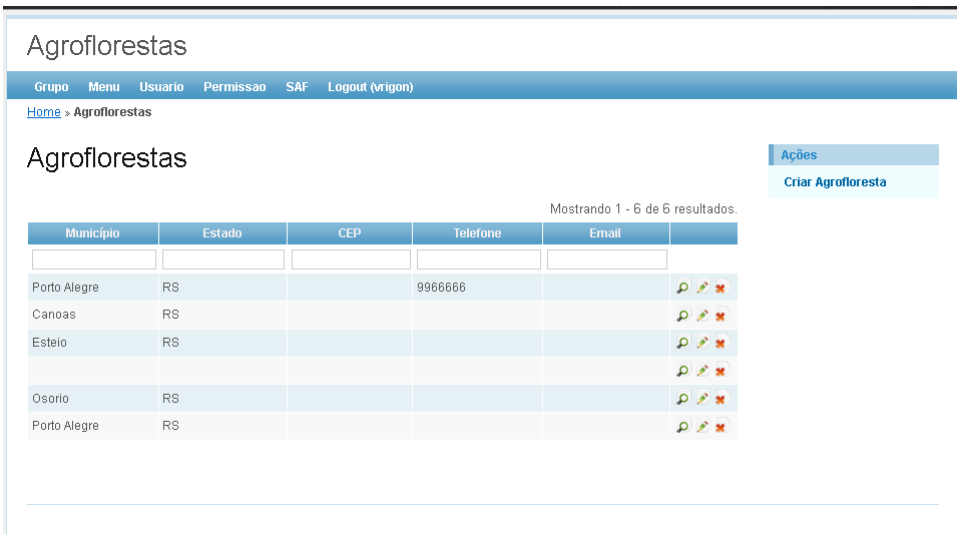

Figura 4.4: Listagem de agroflorestas

Para fazer uma pesquisa o usuário entra com o critério de pesquisa em qualquer uma das caixas de texto em cima da listagem, a figura 4.5 mostra essas caixas de texto.

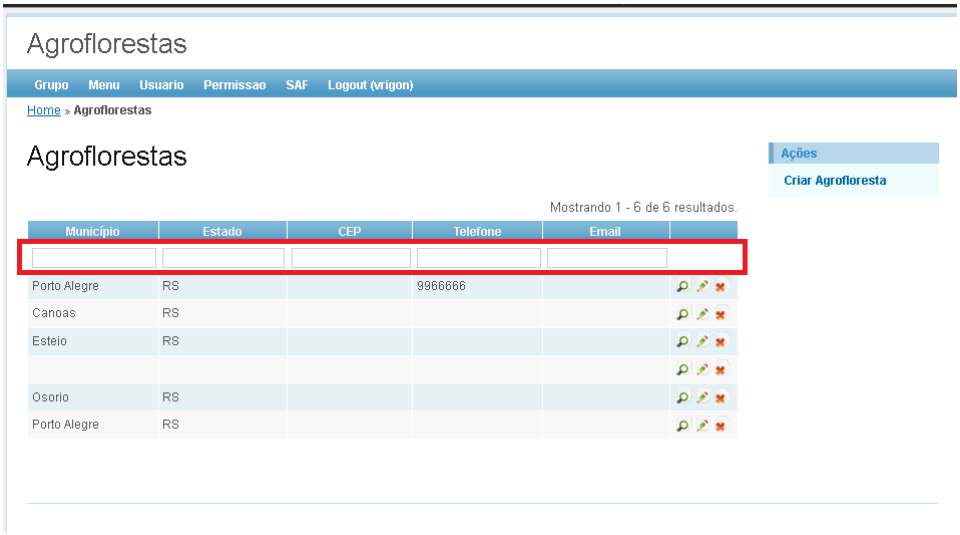

Figura 4.5: Listagem de agroflorestas - pesquisando

A pesquisa é feita depois que o usuário passa para a próxima caixa de texto, ou clica fora dessa caixa. O resultado de uma pesquisa é mostrado na Figura 4.6

Figura 4.6: Listagem de agroflorestas - pesquisa

4.3 Criação de SAF

A criação de uma nova SAF é demostrada na Figura 4.7, para poder entrar com os dados dos produtores, primeiramente cria-se um SAF, para então poder inserir as demais informações dos produtores.

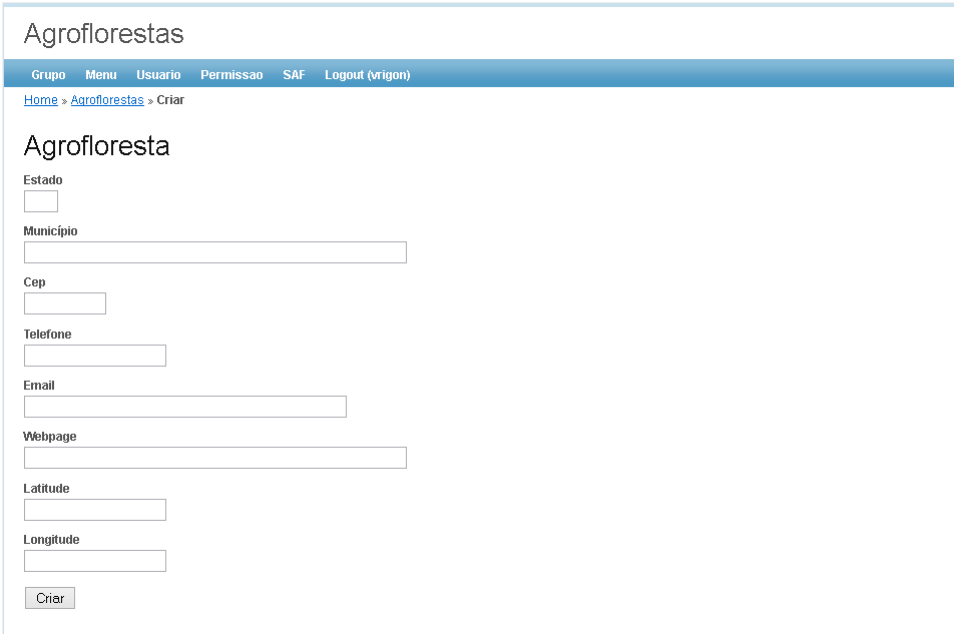

Figura 4.7: Criação de SAF

4.4 Edição de SAF

A edição, mostrada na figura 4.8, é muito simples. Ela mostra todos os dados do SAF em questão, permitindo ao usuário alterar os dados, caso necessário. Quando o usuário clica em 'Salvar', o sistema redireciona para a listagem.

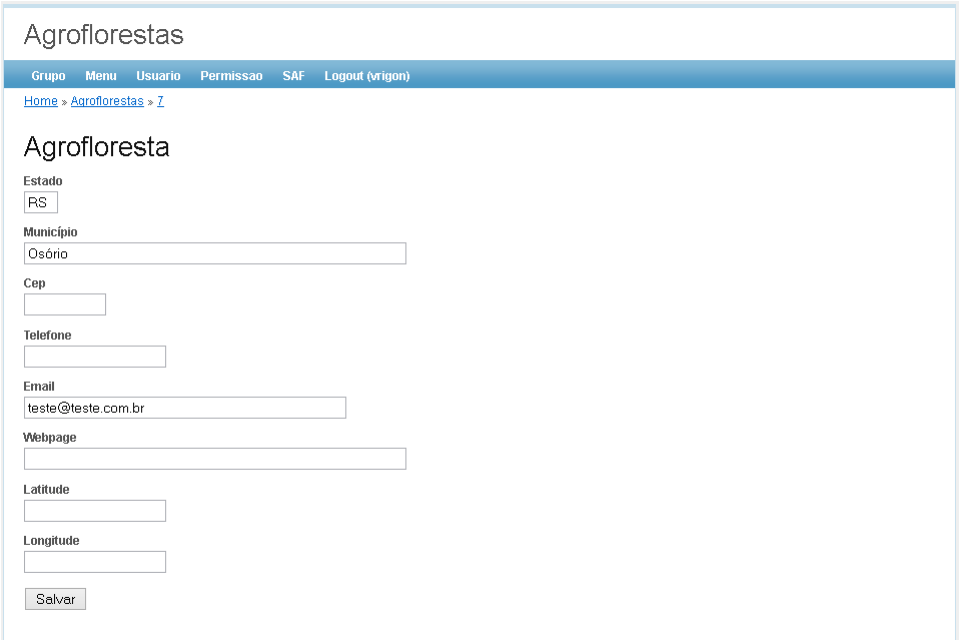

Figura 4.8: Edição de SAF

4.5 Tela de informações dos produtores

Para poder entrar com os demais dados dos produtores, ícone mostrado na figura 4.9 deve ser clicado. Com isso a tela seguinte exibe as informações gerais do SAF e um menu de ações no lado esquerdo contendo todas as demais telas do sistema. Essa tela é mostrada na Figura 4.10.

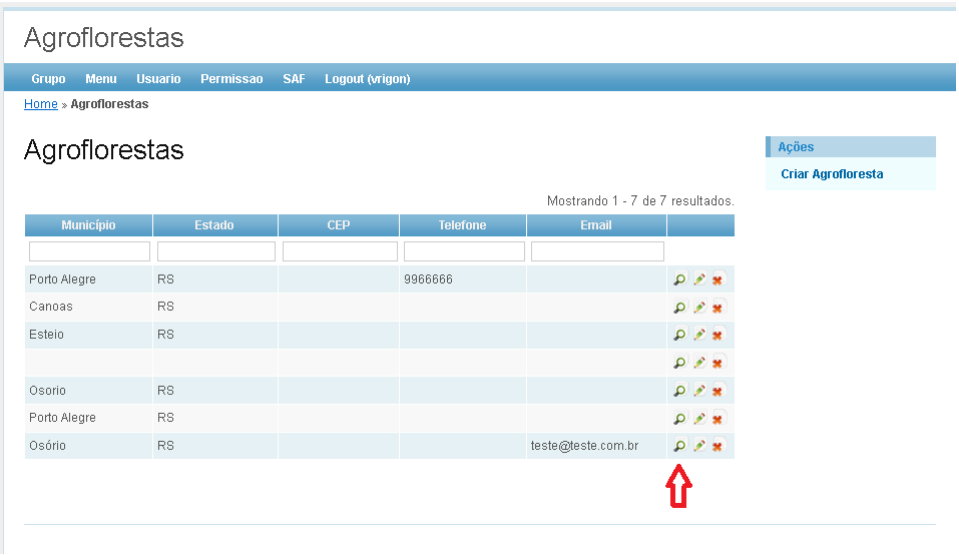

Figura 4.9: Entrar na edição de SAF, informações dos produtores

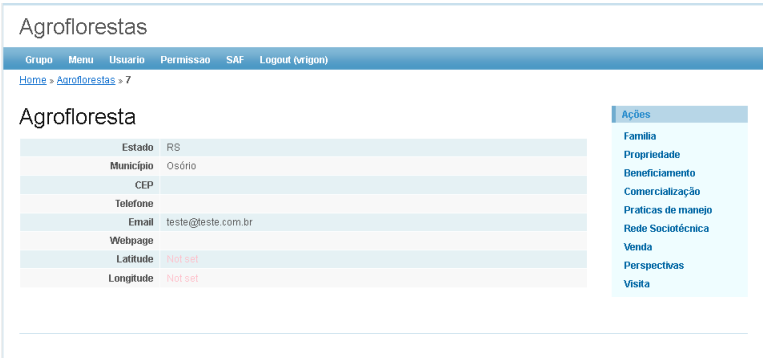

Figura 4.10: Entrar na edição de SAF, informações dos produtores

4.5.1 Família e Integrantes

Na Figura 4.11 é mostrada a tela de edição/inclusão das informações da família que reside na propriedade, já na figuras 4.12 e 4.13, tem-se a listagem e inclusão/edição dos integrantes da família

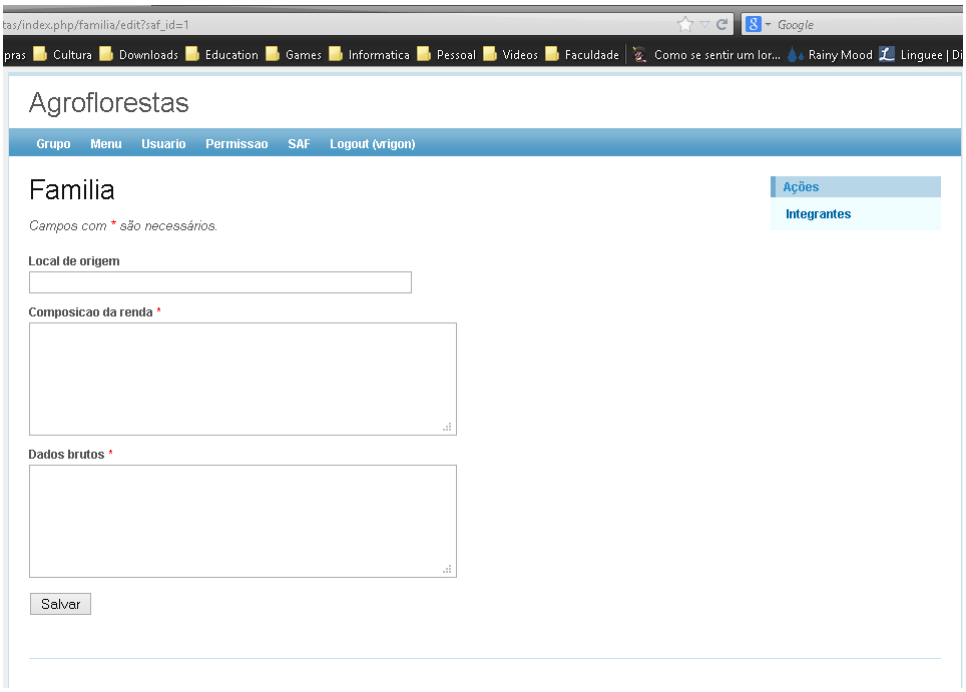

Figura 4.11: Família

Figura 4.12: Integrantes - listagem

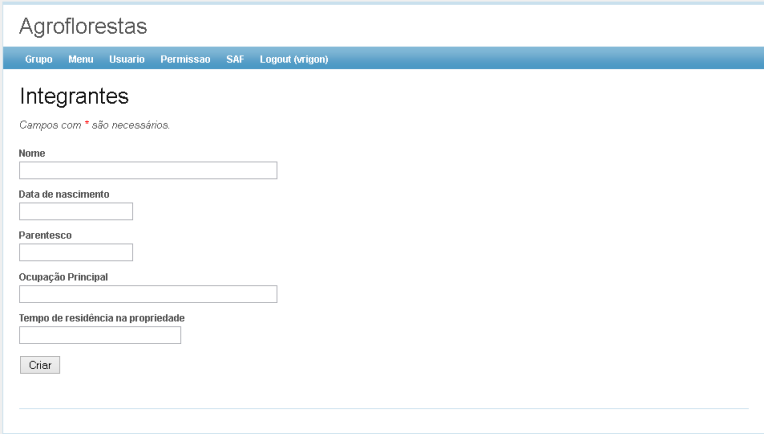

Figura 4.13: Integrantes - edição/inclusão

4.5.2 Propriedade

A tela de propriedade é mostrada na figura 4.14, nela são descritas as informações que podem ser inseridas.

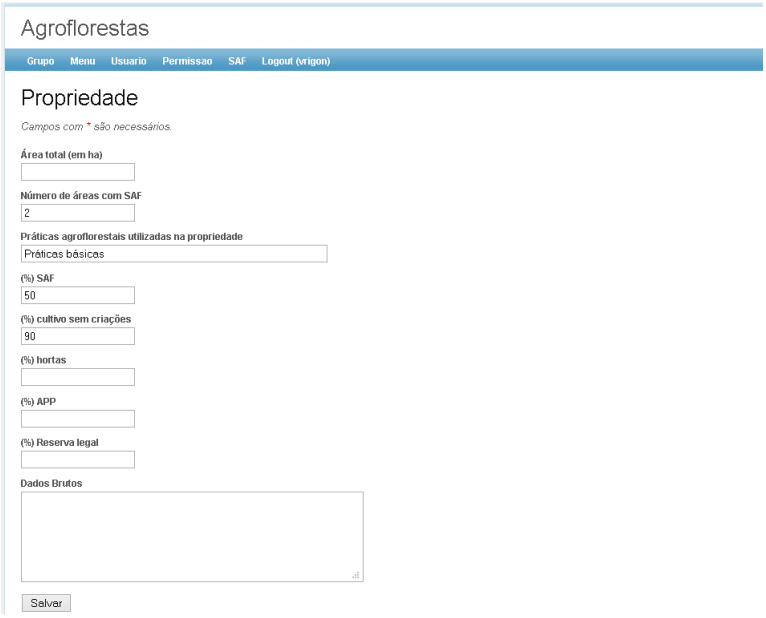

Figura 4.14: Propriedade

4.5.3 Beneficiamento

A Figura 4.15 descreve a tela de beneficiamento, porém está sendo mostrado a tela parcialmente.

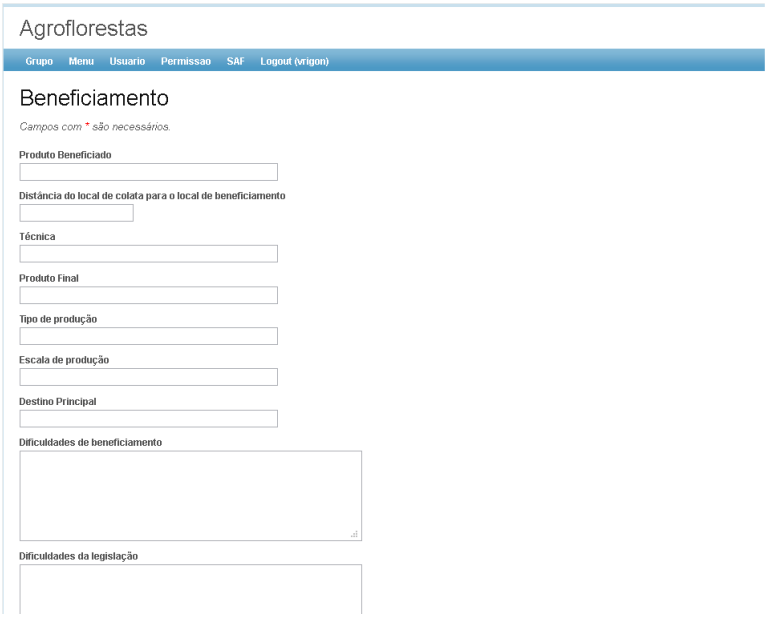

Figura 4.15: Beneficiamento

4.5.4 Comercialização

A tela de comercialização também será mostrada parcialmente, devido ao grande número de campos presentes na tela, como mostra a figura 4.16

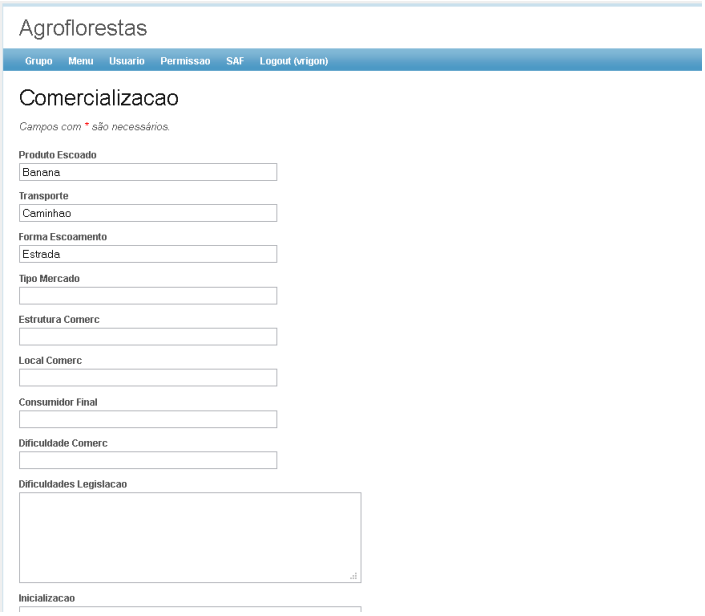

Figura 4.16: Comercialização

4.5.5 Manejo

As figuras abaixo mostram as telas acessadas a partir do item de menu 'Práticas de manejo', serão mostradas as telas de integração com o manejo, os animais e vegetais manejados na propriedade.

Figura 4.17: Práticas de manejo

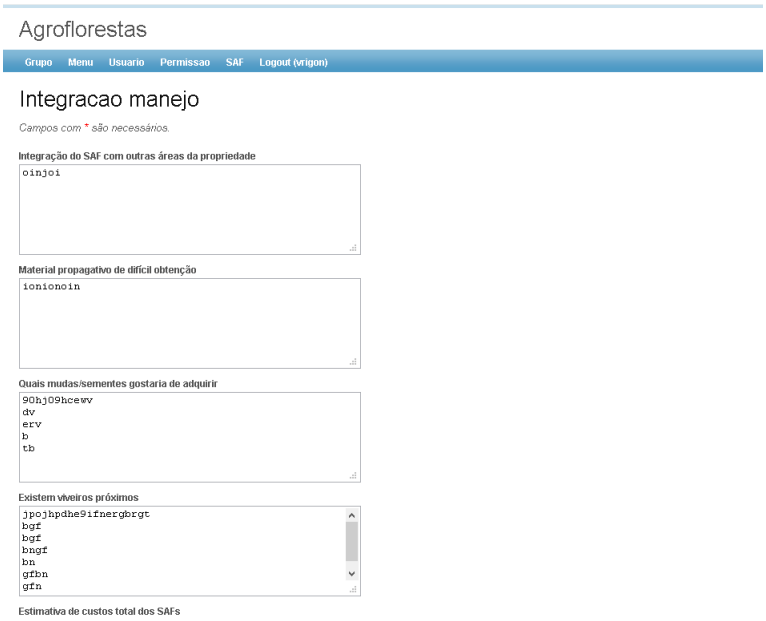

Figura 4.18: Integração com o manejo

Figura 4.19: Listagem de animais manejados

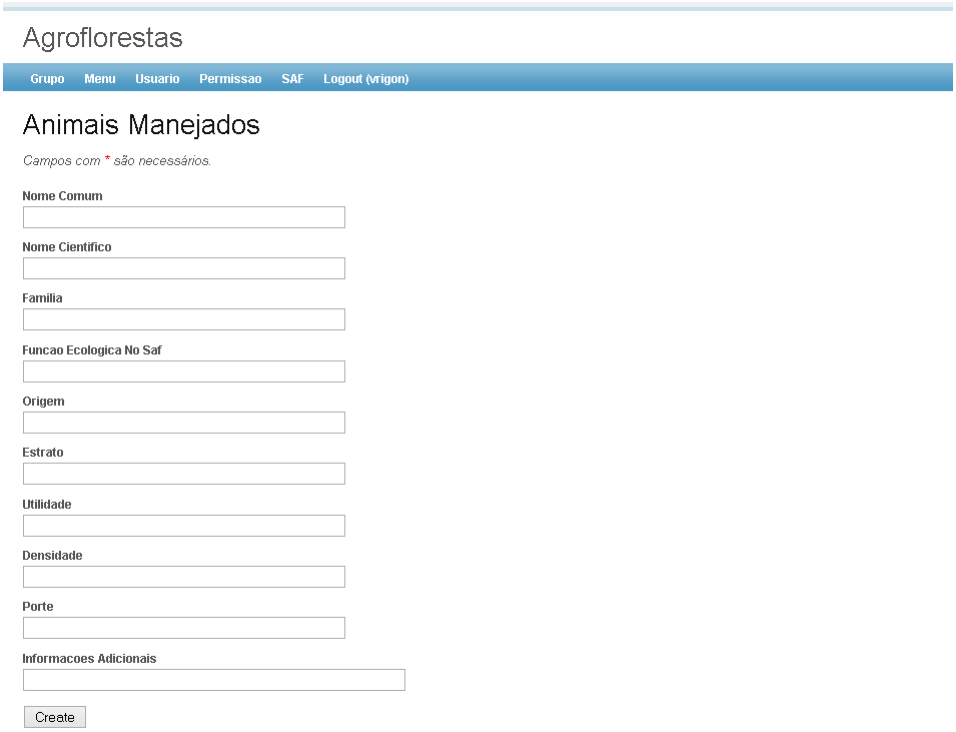

Figura 4.20: Inclusão/Edição de animais

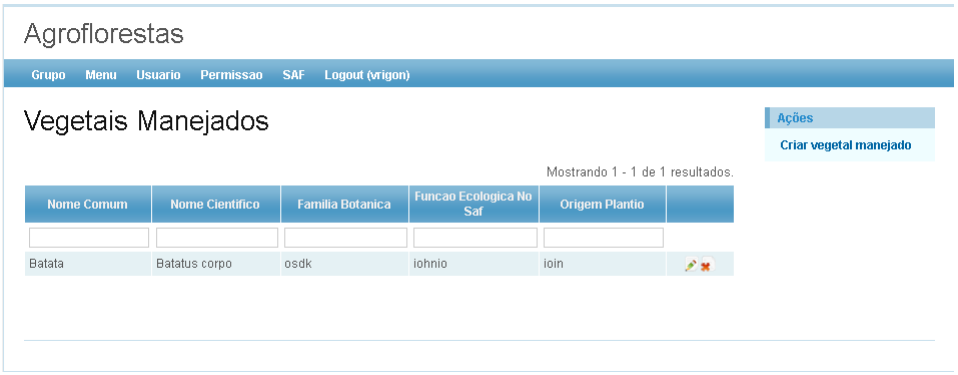

Figura 4.21: Listagem de vegetais manejados

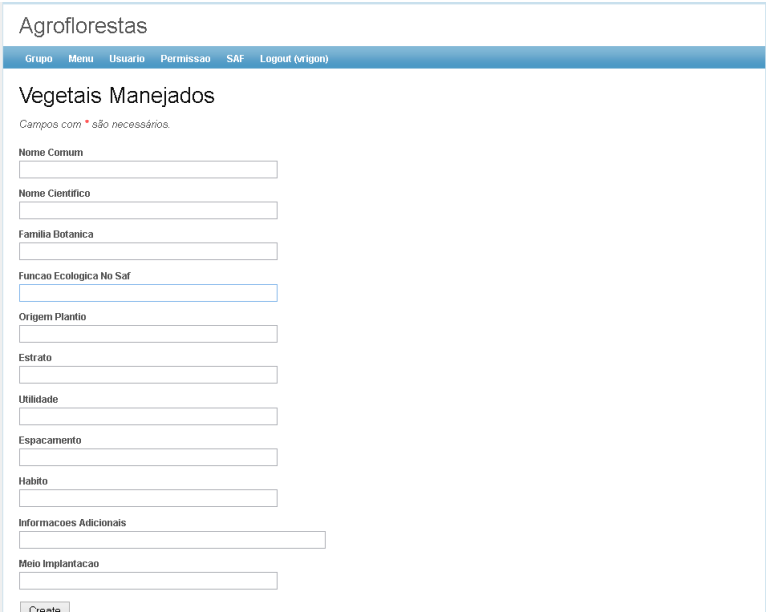

Figura 4.22: Inclusão/Edição de vegetais

4.5.6 Rede Sociotécnica

A tela de cadastro de informações sobre a rede sociotécnica que a agrofloresta está inserida é mostrada a seguir, essa tela é acessada a partir do item de menu 'Rede Sociotécnica'.

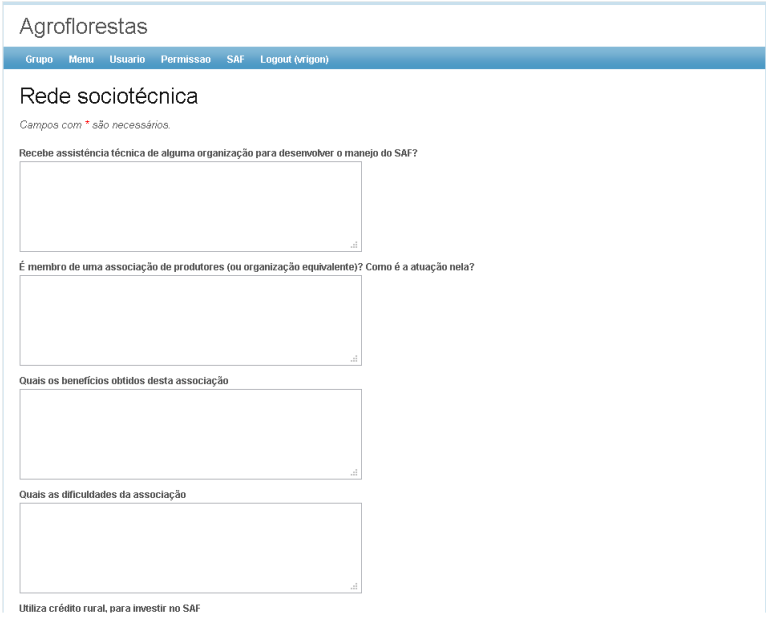

Figura 4.23: Rede Sociotécnica

4.5.7 Perspectivas

A partir do item 'Perspectivas' a tela que permite ao pesquisador cadastrar as informações sobre as perspectivas que o proprietário possui em relação ao projeto Agroflorestas.

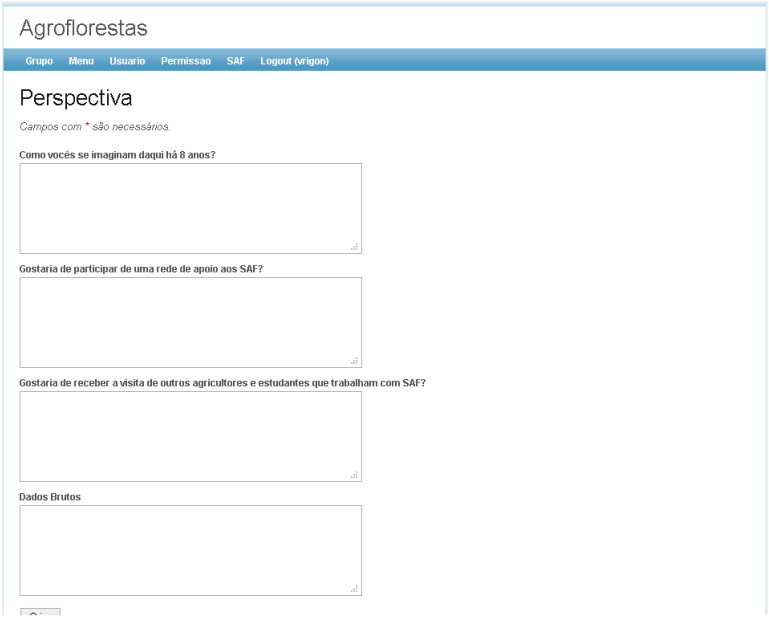

Figura 4.24: Perspectivas

4.5.8 Visita

A última tela que o pesquisador inclui informações é aquela onde o proprietário avalia a visita feita pelos pesquisadores na propriedade, essa tela é mostrada na figura seguinte.

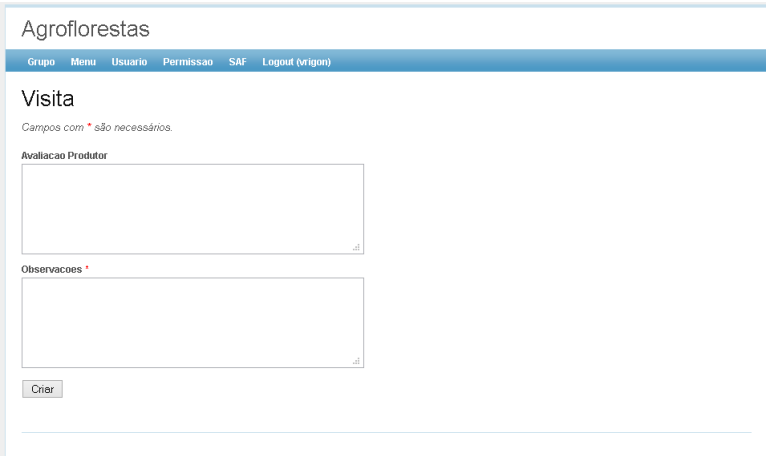

Figura 4.25: Visita

4.6 Resultados

Os usuários do sistema após uma demonstração, ficaram satisfeitos com o mesmo, ainda que as funcionalidades previstos nos requisitos funcionais foram atendidas, tais como: perfis de usuário com controle de acesso, exportação e importação de dados, relatórios, integração com sistema desenvolvido por [Rigo, 2013]. Além disso, sugeriram reformular a maneira como os dados são cadastrados no sistema. Na versão apresentada, os usuários devem acessar diferentes menus, um para cada entidade. No entanto, para oferecer melhor usabilidade, o mesmo deve ser feito em sequência, por exemplo, começando com agroflorestas, depois passando para a família e os seus integrantes, e assim por diante. Outra mudança consiste em possibilitar emissão de relatórios dos dados cadastrados, permitindo assim que o pesquisador tenha uma visão quantitativa e qualitativa dos dados, por exemplo, filtrando pelas SAF que contenham banana, por área de SAF, ou pela quantidade de pessoas que existem em cada família.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento de um sistema Web para auxiliar no cadastro dos dados dos produtores no projeto Agroflorestas. Para isso, inicialmente foi realizado um estudo com base na forma em que atualmente é feito os cadastros dos produtores envolvidos com o projeto, o qual identificou-se ser feito através de uma planilha Excel. Com essa planilha o modelo inicial do banco de dados foi desenvolvido. Através de reuniões com os pesquisadores do projeto o levantamento de requisitos do sistema foi feito, permitindo a elaboração de casos de uso. Após, foi feita a especificação detalhada da arquitetura do sistema junto com sua implementação. Para o desenvolvimento foi necessário o estudo de novas tecnologias, tais como, a linguagem de programação PHP orientado a objetos e o uso de um framework em MVC, o Yii.

Dentre os requisitos levantados, não foi possível concluir a implementação do controle de acesso com diferentes perfis, homologação das agroflorestas e a importação/exportação para planilha Excel, upload de fotos e questionário original, integração com outros sistemas do PGDR usando o banco de dados com informações não sigilosas. Todas essas, além das levantadas durante a reunião de apresentação do sistema serão implantadas futuramente.

REFERÊNCIAS

R.K.J. Caserta. *THE DATA WAREHOUSE ETL TOOLKIT*. Wiley India Pvt. Limited, 2004.

A. Cockburn. *Escrevendo Casos de Usos Eficazes: Um guia prático para desenvolvedores de software*. BOOKMAN COMPANHIA ED, 2005.

Stephen T. Krasner, Glenn E. e Pope. A cookbook for using the model-view controller user interface paradigm in smalltalk-80. *J. Object Oriented Program.*, 1(3):26–49, August 1988. ISSN 0896-8438. URL `http://dl.acm.org/citation.cfm?id=50757.50759`.

MySQL. Mysql :: Developer zone, Outubro 2013. URL `http://dev.mysql.com/`.

K. Övergaard, G. e Palmkvist. *Use cases: patterns and blueprints*. Addison-Wesley, 2005.

R.S. Pressman. *Engenharia de software*. McGraw-Hill, 2006.

Lucas Magrini Rigo. Sistema de apoio à decisão com apresentação georreferenciada de dados no âmbito do projeto agroflorestas. Trabalho de graduação, Instituto de Informática, Universidade Federal do Rio Grande do Sul, 2013.

E. Turban, J.E. Aronson, and T.P. Liang. *Decision support systems and intelligent systems*. Pearson/Prentice Hall, 2005.

L. Welling, L. e Thomson. *PHP and MySQL Web Development*. Pearson Education, 2008.

J. Winesett. *Web Application Development with Yii and PHP*. Packt Publishing, Limited, 2012.

Yii. Yii php framework: Best for web 2.0 development, Novembro 2013. URL `http://www.yiiframework.com/`.