

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE INFORMATICA E ESTATISTICA

CLEBERSON CHARLES COLOMBO FACCIN

**SISTEMA DE INFORMAÇÃO PARA AGRICULTURA FAMILIAR**

FLORIANÓPOLIS

2010

Cleberson Charles Colombo Faccin

## **Sistema de Informação para agricultura familiar**

Trabalho de Conclusão de Curso apresentado ao Departamento de Informática e Estatística da Universidade Federal de Santa Catarina como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: José Eduardo De Lucca

Florianópolis

2010

Cleberson Charles Colombo Faccin

Sistema de Informação para agricultura familiar

Trabalho de Conclusão de Curso apresentado à disciplina INE5632 - Projetos II do Curso de Graduação Sistemas de Informação da Universidade Federal de Santa Catarina, submetido à Banca Examinadora e considerado APROVADO em \_\_/\_\_/\_\_.

Banca Examinadora

---

Profº. José Eduardo De Lucca  
Orientador

---

Profº. Antonio Carlos Mariani  
Membro da Banca

---

Profº. José Francisco Danilo de Guadalupe Correa Fletes  
Membro da Banca

## RESUMO

A agricultura familiar é composta por pequenos e médios produtores rurais que são responsáveis por produzir boa parte dos alimentos consumidos no Brasil. Estas propriedades possuem um grau de informatização muito baixo ou inexistente, que dificulta a gestão dos recursos.

Existem hoje no mercado brasileiro alguns softwares disponíveis para a administração de propriedades rurais. Analisando as funcionalidades destes softwares e a necessidade dos agricultores familiares surgiu a proposta do E-fagro.

O E-fagro é um projeto de *software* livre que foi desenvolvido a partir deste trabalho para auxiliar os agricultores familiares brasileiros na administração de suas propriedades rurais.

**Palavras chave:** agricultura familiar, web, Java, JBoss Seam Framework.

## **ABSTRACT**

The family agriculture is composed of small and medium rural producers who are responsible for producing much of the food consumed in Brazil. These properties have a very low level of computerization or nonexistent, making it difficult to resource management.

There are today in the Brazilian market some software available for the administration of rural properties. Analyzing the features of these software and the need of farmers emerged the proposal of the E-fagro.

E-fagro is a free software project that was developed from this monograph to help the Brazilian family farmers in managing their farms.

**Keywords:** family agriculture, web, Java, JBoss Seam Framework.

## **LISTA DE ABREVIATURAS**

AJAX - Asynchronous JavaScript And XML

CSS - Cascading Style Sheets

JPA - Java Persistence API

JSF - Java Server Faces

SGBD – Sistema de Gerenciamento de Banco de Dados

XHTML - eXtensible Hypertext Markup Language

XML - Extensible Markup Language

## LISTA DE CÓDIGO FONTE

Código Fonte 01 – Fragmento da classe br.ufsc.inf.efagro.entity.Usuario.java.....	40
Código Fonte 02 – Código do arquivo de template edit.xhtml.....	42
Código Fonte 03 – Fragmento da tela cadastraUsuario.xhtml.....	42

## LISTA DE DIAGRAMAS

Diagrama 01 – Diagrama do banco de Dados, parte 1.....	36
Diagrama 02 – Diagrama do banco de Dados parte 2.....	37
Diagrama 03 – Apresentação da arquitetura do JBoss Seam com JSF/EJB/JPA.....	38



## LISTA DE GRÁFICOS

Gráfico - 01 – Variação do valor das commodities agrícolas nos últimos cinco anos.....	20
Gráfico - 02 - Participação da agricultura Familiar na produção de alimentos.....	21
Gráfico - 03 - PIB das cadeias produtivas.....	22

## LISTA DE TABELAS

Tabela - 01 – Evolução dos estoques mundiais de alimento.....	18
Tabela - 02 - Produção e consumo no mundo.....	19

## LISTA DE TELAS

Tela 01 – Tela inicial do ADM Rural 4.0.....	25
Tela 02 – Tela inicial do PRORURAL.....	28
Tela 03 – Tela inicial do Agriness S2 Comercial .....	30
Tela 04 - Uma tela do OpenFarm 1.0.....	31
Tela 05 – Tela para cadastro de usuários .....	43
Tela 06 – Tela inicial do sistema após o usuário entrar no sistema.....	44
Tela 07 – Lista dos talhões do Ano/Safra selecionado .....	45
Tela 08 – Informações detalhadas do talhão selecionado na tela anterior.....	46
Tela 09 – Tela para registro de um plantio em um determinado talhão.....	47
Tela 10 – Tela com opções de relatório para safra.....	48
Tela 11 – Tela exibindo relatório da safra.....	49

## SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	Justificativa.....	15
1.2	Objetivos .....	16
1.2.1	Geral.....	16
1.2.2	Específicos .....	16
1.3	Metodologia.....	17
2	FUNDAMENTAÇÃO TÉCNICA .....	18
2.1	Cenário internacional de produção e consumo de alimentos .....	18
2.2	Commodities Agrícolas.....	19
2.3	Agricultura familiar no mercado brasileiro .....	20
2.4	Oportunidade de Negócio.....	22
3	ESTADO DA ARTE .....	24
3.1	ADM Rural 4.0.....	24
3.2	PRORURAL.....	25
3.3	Agriness S2 3.0.....	28
3.4	OpenFarm.....	30
4	SISTEMA PROPOSTO.....	32
4.1	Visão geral do sistema.....	32
4.2	Funcionalidades gerais do sistema .....	33
4.3	Atores do sistema .....	33
4.3.1	Ator Cliente.....	33
4.3.2	Ator Administrador .....	34
4.3.3	Ator Colaborador .....	34
4.4	Casos de Uso .....	34
4.5	Banco de dados.....	34
4.5.1	Diagrama Entidade Relacionamento.....	35
5	DESENVOLVIMENTO.....	38
5.1	Engenharia reversa do banco de dados.....	39
5.2	Classes de controle .....	41

5.3	Interface com o usuário .....	41
5.3.1	Relatórios .....	48
6	CONCLUSÃO.....	50
6.1	Trabalhos futuros.....	50
7	REFERÊNCIAS .....	52
	ANEXO I – REQUISITOS .....	55
	ANEXO II – REFINAMENTO DOS CASOS DE USO.....	90
	ANEXO III - OPENUP/BASIC .....	111
	ANEXO IV - ARTIGO .....	114
	ANEXO V – CÓDIGO FONTE.....	121

## 1 INTRODUÇÃO

A chamada agricultura familiar, constituída por pequenos e médios produtores, representa a imensa maioria de produtores rurais no Brasil. Em geral, são pessoas com baixo nível de escolaridade, que diversificam os produtos cultivados para diluir custos, aumentar a renda e aproveitar as oportunidades de oferta ambiental e disponibilidade de mão-de-obra. Este segmento tem um papel crucial na economia das pequenas cidades (PORTUGAL, 2004).

Segundo dados do IBGE<sup>1</sup>, o Brasil tem 5.564 municípios, dos quais, 3.942 possuem menos de 20 mil habitantes. Nestes municípios, os produtores rurais e seus familiares são responsáveis por movimentar a economia e, conseqüentemente, gerar inúmeros empregos no comércio e na prestação de serviços das pequenas cidades (PORTUGAL, (2004).

O governo federal, com apoio de estados e municípios, mantém um órgão oficial para desenvolver projetos de apoio à infraestrutura, assistência técnica, pesquisa agropecuária, capacitação e linhas de crédito diferenciadas. Trata-se da Secretaria da Agricultura Familiar do Ministério do Desenvolvimento Agrário (SAF/MDA)<sup>2</sup>. Entre os principais projetos desenvolvidos pela Secretaria está o Programa Nacional de Fortalecimento da Agricultura Familiar (Pronaf) e o Plano Safra da agricultura familiar.

O Pronaf<sup>3</sup> é um conjunto de políticas que visa facilitar o planejamento das atividades, permitir o acesso à melhor tecnologia e contribuir para o aproveitamento de novas oportunidades econômicas. O Programa possibilita, aos pequenos agricultores, ampliar a segurança alimentar de suas famílias e contribuí para o aumento da renda, com risco reduzido ou até inexistente (BRASIL, 2006). De acordo com os dados disponíveis no portal da SAF/MDA, de 1999 até 2008, foram aprovados e financiados quase 13 milhões de contratos Pronaf, totalizando um montante de aproximadamente

---

1 Dados de 2008. Disponível em: [ftp://ftp.ibge.gov.br/Perfil\\_Municipios/2008/Base2008.zip](ftp://ftp.ibge.gov.br/Perfil_Municipios/2008/Base2008.zip). Acessado em 15/10/2009

2 Disponível em: <<http://www.mda.gov.br/saf/>>. Acessado em 29/09/2010

3 Criado pelo Decreto Nº 1.946, de 28 de junho de 1996. Disponível em:<<http://www.mda.gov.br/saf/arquivos/DEC%201.946%20PRONAF.doc>>. Acessado em 15/09/2009

50 bilhões de reais<sup>4</sup>. Os financiamentos liberados pelo Programa possuem uma das menores taxas de inadimplência do sistema financeiro brasileiro, em torno de 2,5%<sup>5</sup>.

O Plano Safra tem como objetivo alavancar o crescimento do setor agrícola brasileiro. Anualmente, são disponibilizados bilhões de reais, de acordo com faixas de renda, tamanho das propriedades e localização geográfica. Além disso, há programas especiais para o semi-árido brasileiro, para projetos de agroecologia, para os jovens e para as mulheres<sup>6</sup>.

Para Pereira (2009), a aplicação da tecnologia da informação vem crescendo consideravelmente em todos os setores, permitindo o avanço científico em várias áreas do conhecimento. Os recursos tecnológicos facilitam cada vez mais o dia-a-dia das pessoas, incorporando novos hábitos e construindo novas relações sociais. No setor agropecuário, entretanto, a presença da tecnologia da informação ainda é tímida. Estudos indicam que dos 3,7 milhões de propriedades rurais no País, apenas entre 4 e 8% delas são informatizadas.

## 1.1 Justificativa

Por possuírem grau de informatização muito baixo ou inexistente, os agropecuaristas familiares encontram dificuldades para gerir os recursos de sua propriedade. Dessa forma é difícil gerar históricos de produtividade/lucratividade para aperfeiçoar o desempenho do negócio. E, neste caso, como as propriedades familiares geralmente diversificam sua produção, a administração da propriedade fica mais sujeita a falhas.

Sendo assim, a realização do presente estudo se justifica pela possibilidade de desenvolver um software que auxilie na administração da propriedade, facilitando o

---

4 Disponível em:

[http://smap.mda.gov.br/credito/anofiscal/rele\\_anofiscal.asp?cboAnoInicio=1999&cboAnoTermino=2008&cboUF=&SiglaDaUF=&NomeDaUF=&cboCDMunicipio=](http://smap.mda.gov.br/credito/anofiscal/rele_anofiscal.asp?cboAnoInicio=1999&cboAnoTermino=2008&cboUF=&SiglaDaUF=&NomeDaUF=&cboCDMunicipio=). Acesso em 23/10/2009.

5 Disponível em: <http://www.mda.gov.br/saf/index.php?sccid=813>. Acessado em: 16/10/2009.

6 Plano Safra da Agricultura Familiar 2008/2009. Disponível em:

[http://www.fpagropecuaria.com.br/images/site/PAP\\_2008\\_2009%5B1%5D.pdf](http://www.fpagropecuaria.com.br/images/site/PAP_2008_2009%5B1%5D.pdf). Acesso em 17/10/2009

trabalho do administrador na coleta e organização das informações, bem como no registro de informações, já que, no cenário atual, quando é feito um controle, este é feito de forma manual em 92 à 96%<sup>7</sup> dos casos. Além destas facilidades citadas o software pode manter históricos de produtividade e lucratividade, melhorando o uso dos recursos agropecuários, diminuindo riscos e possivelmente incrementando a renda familiar.

## 1.2 Objetivos

### 1.2.1 Geral

Auxiliar os agricultores familiares na administração de sua propriedade rural criando um *software* livre<sup>8</sup>.

### 1.2.2 Específicos

- Pesquisar a existência, funcionalidade e disponibilidade no mercado de softwares voltados à gestão agrícola.
- Identificar as necessidades dos agricultores familiares para a gestão de suas propriedades e produção.
- Projetar e desenvolver um software para gestão da produção em pequenas propriedades ou na agricultura familiar.

---

<sup>7</sup> Estudos indicam que das 3,7 milhões de propriedades rurais no País, apenas entre 4 e 8% delas são informatizadas (PEREIRA, 2009).

<sup>8</sup> Software Livre, ou Free Software, conforme a definição de software livre criada pela Free Software Foundation é o software que pode ser usado, copiado, estudado, modificado e redistribuído sem restrição. A forma usual de um software ser distribuído livremente é sendo acompanhado por uma licença de software livre e com a disponibilização do seu código fonte (FREE SOFTWARE FOUNDATION, 2009).



### 1.3 Metodologia

Na primeira etapa do trabalho foi realizada uma pesquisa nos órgãos oficiais do governo, através de seus portais na *web*, e institutos de pesquisa voltados para a agricultura, como a Embrapa e a Epagri, com na qual se buscou um melhor entendimento sobre a agricultura brasileira, com foco na agricultura familiar. Foram identificadas as principais atividades agrárias e sua importância no contexto sócio econômico brasileiro.

Posteriormente, já conhecendo as diversas formas da prática da agricultura, foi realizada uma busca na *web* para identificar a existência de *softwares* voltados para a gerência de propriedades rurais. A partir do conhecimento dos produtos disponíveis no mercado iniciou-se um estudo das funcionalidades dos *softwares* e opto-se por criar um sistema *web* para gerência de propriedades rurais, mais especificamente para a agricultura familiar. O período de levantamento destas informações foi de março de 2009 até julho do mesmo ano.

Tomada a decisão de se construir um novo sistema, foi estudado uma forma de coletar as necessidades de agricultores familiares. Utilizando algumas técnicas de levantamento de requisitos sugeridas pelo processo de desenvolvimento de software OpenUp/Basic<sup>9</sup>, como realização de entrevistas, trabalhar no ambiente alvo, sessões de *brainstorming* e estudar sistemas semelhantes, foram levantados os requisitos funcionais do sistema proposto. Os requisitos levantados estão detalhados no Anexo I deste trabalho. A partir dos requisitos levantados foram refinados os casos de uso descritos de forma textual no Anexo II.

A partir dos requisitos e dos casos de uso foi proposto um sistema que é apresentado no capítulo cinco deste trabalho.

---

<sup>9</sup> O processo de desenvolvimento de software OpenUp/Basic é brevemente descrito no Anexo III deste trabalho.

## 2 FUNDAMENTAÇÃO TÉCNICA

### 2.1 Cenário internacional de produção e consumo de alimentos

Segundo a *The Food and Agriculture Organization of the United Nations* (FAO, 2009), mais de um bilhão de pessoas no mundo passam fome, ou seja, uma a cada seis pessoas do globo não tem as necessidades básicas alimentares satisfeitas.

Podemos observar na Tabela 1, que os estoques mundiais de alguns dos principais produtos agrícolas, milho, arroz e trigo, tiveram uma grande redução nos últimos 10 anos. Isso causou uma elevação do preço dos alimentos ao consumidor final.

**Tabela - 01 – Evolução dos estoques mundiais de alimento em milhões de toneladas.**

<b>Evolução dos estoques mundiais de alimento</b>			
<b>Safra</b>	<b>Arroz</b>	<b>Milho</b>	<b>Trigo</b>
1999/2000	143,1	193,7	209,5
2008/2009	82,6	99	124

Fonte: USDA, 2009

A redução dos estoques se deu principalmente pelo fato da produção mundial de alimentos não acompanhar a demanda de consumo, como podemos ver na Tabela 2.

**Tabela – 02 - Produção e consumo no mundo, entre 2000/2001 e 2007/2008 em milhões de toneladas.**

<b>Produção e consumo no mundo</b>			
<b>Produtos</b>	<b>Produção</b>	<b>Consumo</b>	<b>Diferença</b>
Arroz	3.234,90	3.280,20	-45,50
Milho	5.322,50	5.390,60	-68,10
Soja	1.637,40	1.618,60	18,80
Trigo	4.733,50	4.815,20	-81,70
Total	14.982,30	15.104,60	-176,30

Fonte: USDA. 2009

No Brasil, segundo o Ministério do Desenvolvimento Agrário (2009), enquanto o índice dos preços agrícolas internacionais subiu 83% nos últimos 36 meses, o da cesta básica brasileira avançou 25% no mesmo período<sup>10</sup>.

## **2.2 Commodities Agrícolas**

O gráfico a seguir mostra a flutuação do valor das *commodities* agrícolas nos últimos cinco anos, sendo que em julho de 2008 atingiram o maior valor desde outubro de 1996<sup>11</sup>. Logo depois os valores dos produtos despencaram chegando a março de 2009 ao menor valor desde o final da década de 80<sup>12</sup>.

---

10 Disponível em: <[http://www.fpagropecuaria.com.br/images/site/PAP\\_2008\\_2009%5B1%5D.pdf](http://www.fpagropecuaria.com.br/images/site/PAP_2008_2009%5B1%5D.pdf)>  
Acessado em 15/11/2009

11 Disponível em: <<http://www.indexmundi.com/commodities/?commodity=agricultural-raw-materials-price-index&months=240>>. Acessado em 11/11/2009

12 Disponível em: <http://www.indexmundi.com/commodities/?commodity=agricultural-raw-materials-price-index&months=300>. Acessado em 11/11/2009

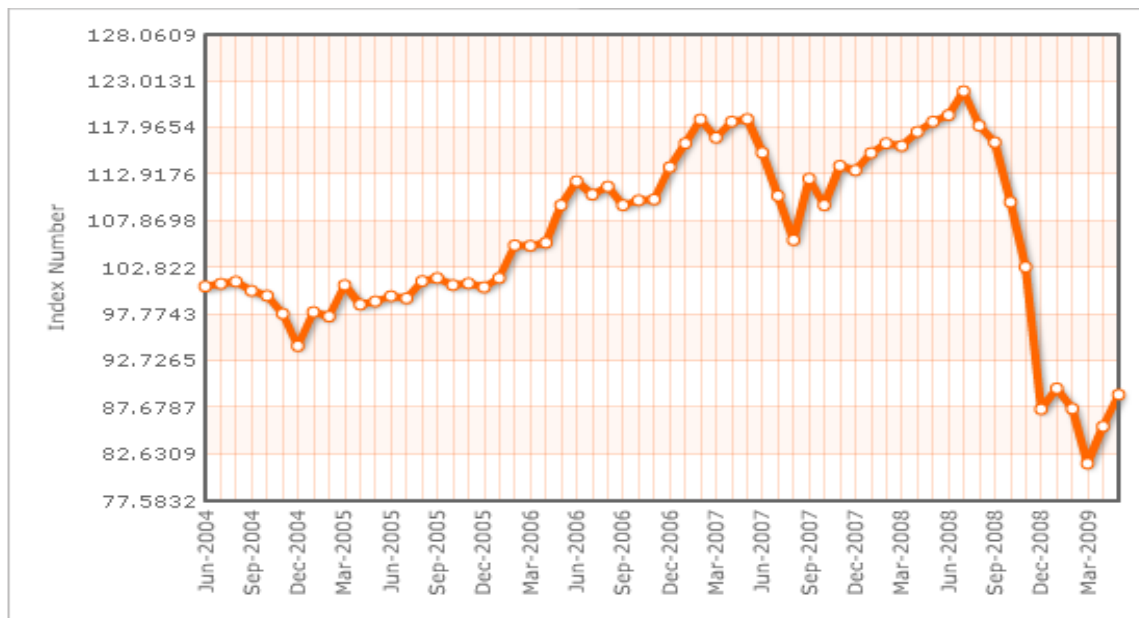


Gráfico - 01 – Variação do valor das *commodities* agrícolas nos últimos cinco anos.  
 Fonte: Disponível em: <<http://www.indexmundi.com>>. Acessado em 11/06/2009.

Porém, segundo Beting (2009), as cotações dos nossos principais produtos agrícolas de exportação encerram maio com elevação acumulada no ano de 13% no milho, 17% no café, 24% no suco de laranja, 28% na soja, 29% no algodão e 37% no açúcar. Isso é bom para quem produz ruim para quem consome. E, em relação a crise financeira mundial, iniciada em meados de 2007, no mercado norte-americano, pode-se afirmar que para as *commodities* minerais, metálicas e agrícolas a crise já passou.<sup>13</sup>

### 2.3 Agricultura familiar no mercado brasileiro

Segundo a Secretaria Nacional da Agricultura Familiar, a produção dos agricultores familiares supre 70% do consumo interno do país<sup>14</sup>. O gráfico a seguir mostra a participação da agricultura familiar na produção de alguns alimentos.

13 Disponível em:<<http://www.ppge.ufrgs.br/akb/dossie-crise.pdf>>. Acessado em 22/01/2010

14 Disponível em:<[http://www.fpagropecuaria.com.br/images/site/PAP\\_2008\\_2009%5B1%5D.pdf](http://www.fpagropecuaria.com.br/images/site/PAP_2008_2009%5B1%5D.pdf)>. Acessado em 28/09/2009

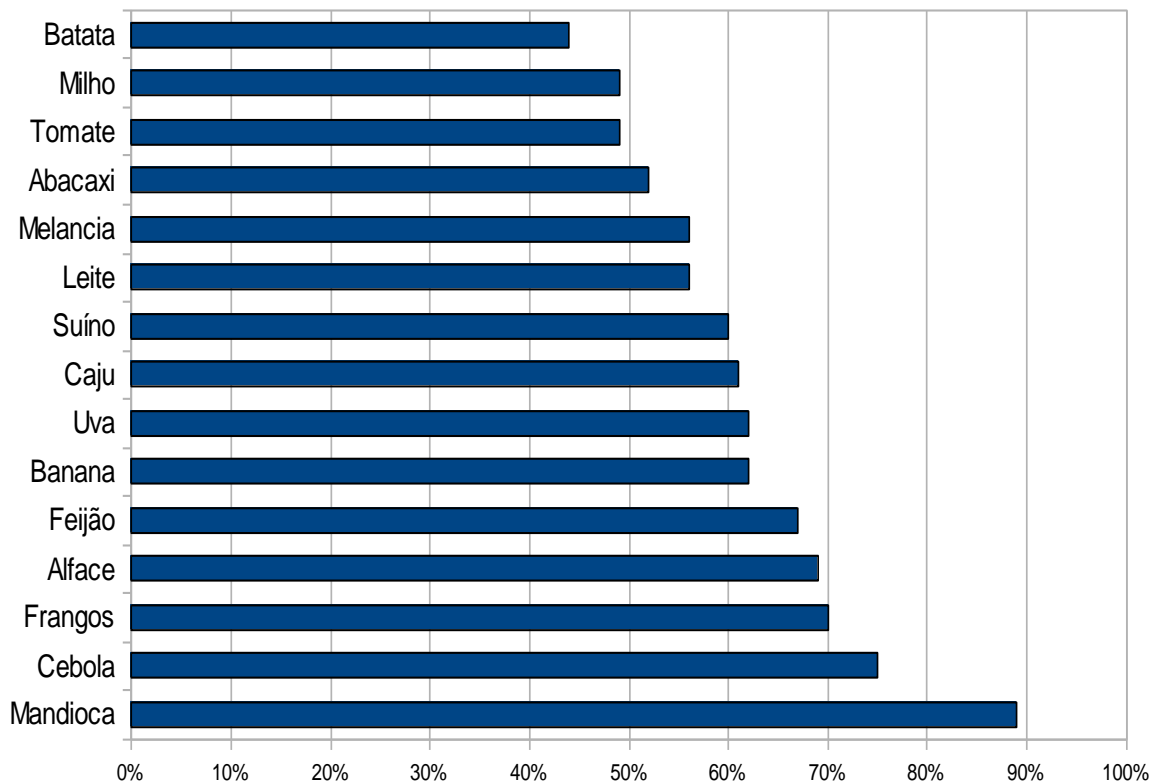


Gráfico - 02 - Participação da agricultura Familiar na produção de alimentos. Fonte: SAF/MDA, 2009.

Pode-se visualizar no gráfico seguinte a distribuição do PIB agropecuário nas cadeias produtivas, no ano de 2009. A agropecuária familiar brasileira representa 32% do PIB agropecuário, isto é, quase um terço de toda a produção agropecuária do Brasil. Vale ressaltar que a produção agrícola patronal responde por metade da riqueza gerada pela agricultura no Brasil. Isso se deve ao fato que a produção de soja, café, cana-de-açúcar e seus derivados, são principais produtos agrícolas de exportação<sup>15</sup>.

15 Fonte: DIEESE – Tabela 84.

## Distribuição do PIB das cadeias produtivas segundo o tipo de produtor Brasil 2005 (em %)

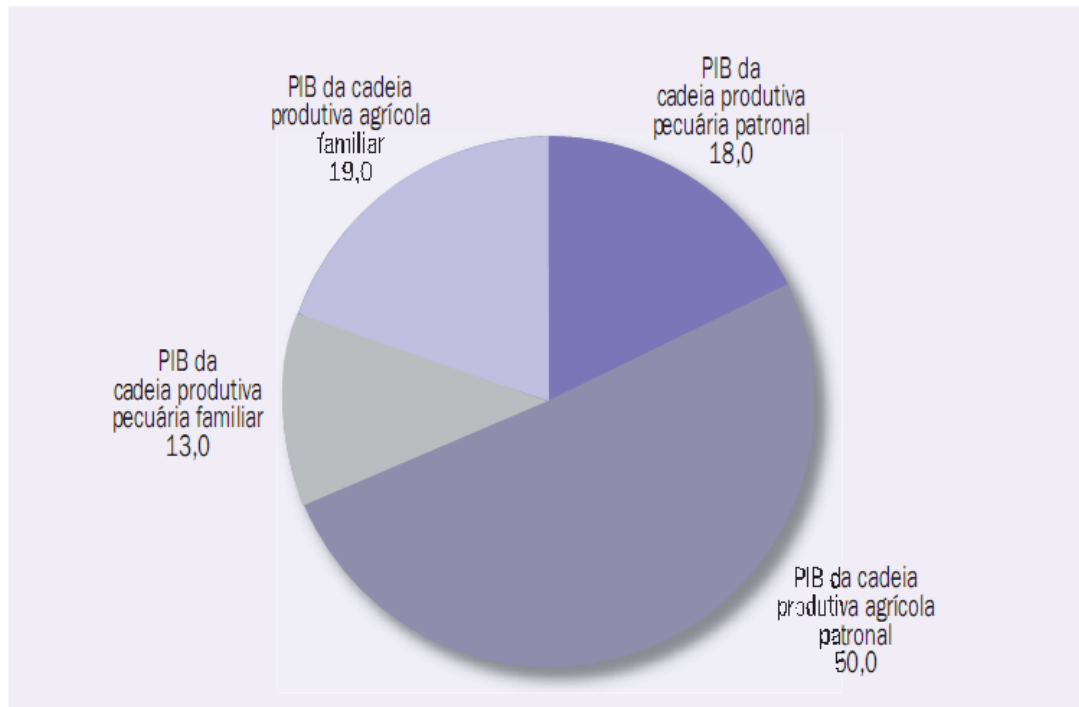


Gráfico - 03 - PIB das cadeias produtivas. Fonte: DIEESE.

### 2.4 Oportunidade de Negócio

Observando o mercado de software brasileiro voltado para agronegócio (no capítulo três deste trabalho são avaliados alguns destes softwares), pode-se dizer que não existem muitas opções, principalmente para a agricultura familiar. Os aplicativos comerciais hoje são voltados quase que exclusivamente para grandes e médias propriedades rurais.

Além de um custo inicial para aquisição do software e, possivelmente, de um computador, já que, segundo o Centro de Estudos sobre as Tecnologias da Informação e da Comunicação (Cetic), apenas 8% dos domicílios da zona rural possuem computadores (CETIC, 2008), tem-se o agravante de que a maioria dos softwares do

mercado só estão disponíveis para plataforma Windows, encarecendo ainda mais a aquisição desta ferramenta de auxílio na administração das propriedades.

Segundo o Cetic (2008), apenas 4% dos domicílios rurais tem acesso a internet. Proporcionalmente, é um número pequeno, mas quando tratamos de valores absolutos, estes 4%, segundo o IBGE (2008), são de aproximadamente 330 mil domicílios, representando um mercado em potencial bastante considerável. Deve-se levar em conta ainda que as perspectivas deste número crescer nos próximos anos é muito grande, principalmente devido à expansão de tecnologias como a cobertura de redes de celular de terceira geração, redes sem fio e a transmissão de sinal pela rede elétrica<sup>16</sup>.

Tendo em vista estes dados, a disponibilização de um serviço via *web* para gestão de agronegócio pode-se configurar um negócio viável. As vantagens de um serviço *web* em comparação a uma aplicação *desktop*, neste caso são:

- Retira do usuário a responsabilidade de backup e manutenção da base de dados;
- Acesso a partir de qualquer computador conectado a internet;
- Facilitada a atualização de versões;
- Independência do sistema operacional;
- Necessidade de computadores menos potentes, já que o único aplicativo necessário para utilizar o produto é um navegador;
- Possibilidade de acesso a partir de dispositivos móveis como celular e *palms*;
- Reduz os problemas de instalação.

Claro que um sistema *web* também possui desvantagens quando comparado a uma aplicação *desktop*, como por exemplo, a possibilidade de utilização somente conectado a internet e a dificuldade na construção de interfaces mais amigáveis.

Um dos desafios de desenvolver um software para o público definido, os agricultores familiares, é a capacitação dos usuários, que, conforme já citado, em geral, são agricultores com baixo nível de escolaridade e pouca intimidade com computadores.

---

<sup>16</sup> Disponível em:  
[http://www.aneel.gov.br/aplicacoes/noticias/Output\\_Noticias.cfm?Identidade=3127&id\\_area=90](http://www.aneel.gov.br/aplicacoes/noticias/Output_Noticias.cfm?Identidade=3127&id_area=90).  
Acessado em 02/09/2009.

### 3 ESTADO DA ARTE

A pesquisa por *software* com funcionalidades semelhantes ao sistema proposto foi realizada na Internet, em ferramentas de busca, fóruns e indicações de pessoas que atuam na área. A seguir são listados os *softwares* pesquisados bem como suas funcionalidades.

#### 3.1 ADM Rural 4.0

É um software para controle de custos de produção com a utilização de planos de contas gerenciais e centro de custos por atividade produtiva. Desenvolvido em Visual Basic, utilizando banco de dados Firebird, pela empresa Planejar Informática, o custo da licença é de R\$ 650,00. No site da empresa, está disponível uma versão para demonstração, que foi utilizada neste trabalho.

Analisando-se o software foram identificadas as seguintes funcionalidades:

- Configurar unidades de medida para cada tipo de cultura, por exemplo, arroba e saca;
- Configurar indexadores financeiros e de produção;
- Incluir, alterar e remover propriedades e contatos;
- Manter históricos das atividades agrárias;
- Manter inventário da propriedade;
- Gerência de custos;
- Controle da produção de gado leiteiro ou de corte;
- Controle da produção, armazenamento e venda de produtos agrícolas;
- Manutenção dos orçamentos da propriedade;
- Fluxo de caixa.
- Ferramentas de cópia de segurança e restauração da base de dados.
- Relatórios históricos de atividades.
- Controle de pastagens.
- Relatórios por safra ou produto.



- Históricos de vendas.

A imagem a seguir se refere à tela inicial do software ADM RURAL 4.0.



Tela 01 – Tela inicial do ADM Rural 4.0 – Versão Demonstrativa. Fonte: ADM RURAL 4.0.

## 3.2 PRORURAL

Este software apresenta dois módulos, um para o controle agrícola e outro para o controle pecuário. Utiliza o banco de dados Firebird, que é instalado no cliente durante a instalação do software. É desenvolvido pela Brazsoft Tecnologia em Agro Business, empresa sediada em Cuiabá, Mato Grosso, que desenvolve o software desde 2003. Possui diversos clientes espalhados por praticamente todo o Brasil. O investimento do

produtor rural para adquirir o software é de aproximadamente R\$ 400,00, segundo informação do site da empresa de junho de 2009.

Após a instalação o usuário efetua o cadastro da fazenda, as atividades, talhões, produtos, máquinas, contas bancárias e pessoas envolvidas. Várias funcionalidades são comuns aos dois módulos. A seguir são descritas as funcionalidades específicas de cada um deles.

As funcionalidades do módulo agrícola são:

- Cadastrar as atividades agrícolas realizadas durante o ciclo de produção, por exemplo, plantio, colheita, correção de solo e aplicação de defensivos. Cada fazenda cria e modifica suas próprias atividades.
- Cadastrar e modificar os itens produzidos pela fazenda. Estes produtos são os cultivares produzidos, por exemplo, milho, soja, sorgo trigo, etc.
- Cadastrar e modificar talhões. Cada talhão é uma fração da área da fazenda, que abriga durante certo período o cultivo de um determinado produto.
- Lançar no sistema as atividades agrícolas realizadas.
- Cadastrar ocorrência de pragas.
- Controle de armazenagem de grãos
- Controle de estoque de insumos.
- Relatórios climáticos com comparativos entre safras ou períodos definidos pelo usuário.
- Relatórios de colheita, de plantio, de acompanhamento de sementes no caso de produção de sementes, de atividades e de incidência de pragas.

No módulo pecuária, que no caso deste aplicativo cobre apenas a criação de mamíferos, temos as seguintes funcionalidades:

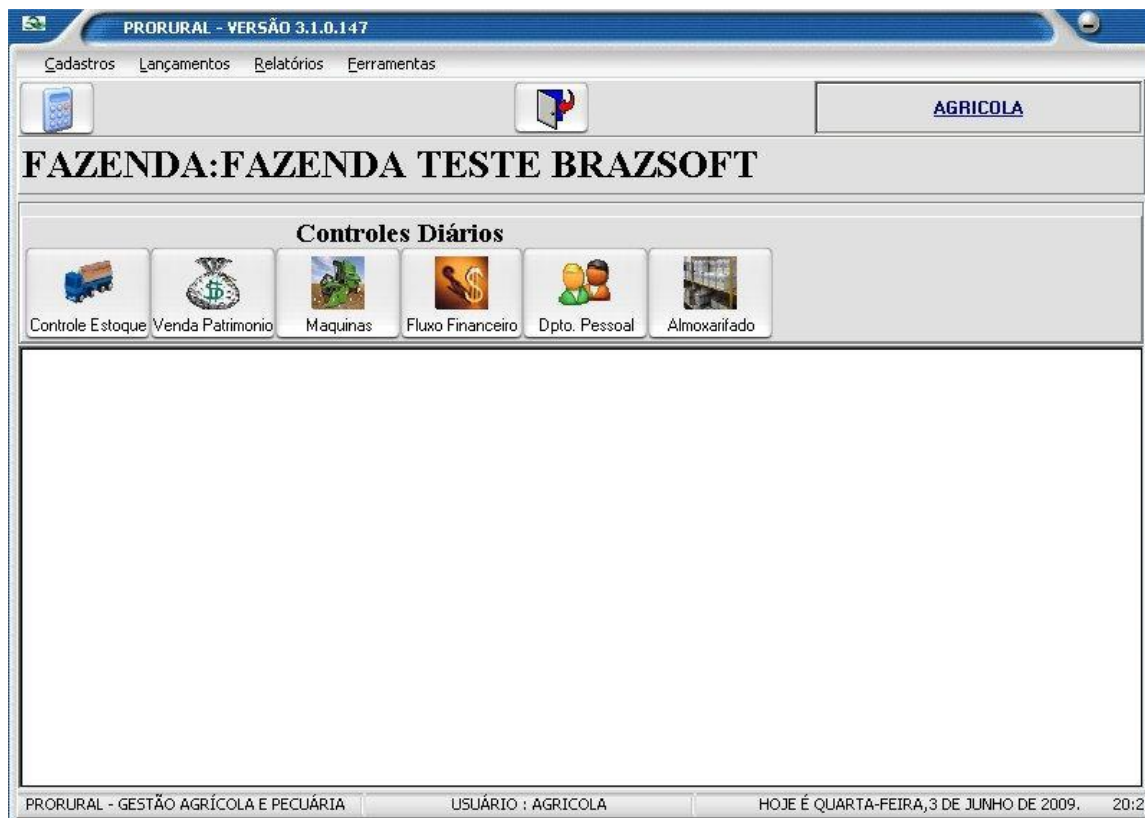
- Cadastro das espécies.
- Cadastro dos animais, podendo ser divididos em grupos ou lotes.
- Controle da produção de leite.
- Controle do ciclo produtivo dos animais, reprodução, desmama, engorda, descarte, venda e morte.
- Controle de vacinações.

- Formulário de rações e suplementação alimentar.
- Relatórios por fase de crescimento dos animais, ganho de peso, período de lactação e conversão de alimento em peso.
- Relatórios e estatísticas de morte e descarte.
- Histórico de pesagem dos animais.

Além das funcionalidades específicas, temos funcionalidades que são comuns aos dois módulos:

- Cadastrar insumos, tais como fertilizantes e sementes para agricultura e rações, vacinas e suplementos alimentares para a pecuária.
- Cadastrar pessoas, estas podem ser funcionários clientes ou fornecedores.
- Controle do fluxo financeiro.
- Controle de venda de patrimônio.
- Controle do departamento de pessoal
- Controle de estoque e almoxarifado
- Manter cadastro de máquinas, com históricos de compra, venda e manutenção destas.
- Acompanhamento climático através de lançamentos diários de índice pluviométrico, temperatura máxima e mínima, velocidade do vento e umidade relativa do ar.
- Ferramenta de backup e restauração da base de dados.
- Ferramenta para emissão de notas fiscais.

A imagem a seguir se refere à tela inicial do software PRORURAL.



Tela 02 – Tela inicial do PRORURAL– Versão Demonstrativa. Fonte: PRORURAL.

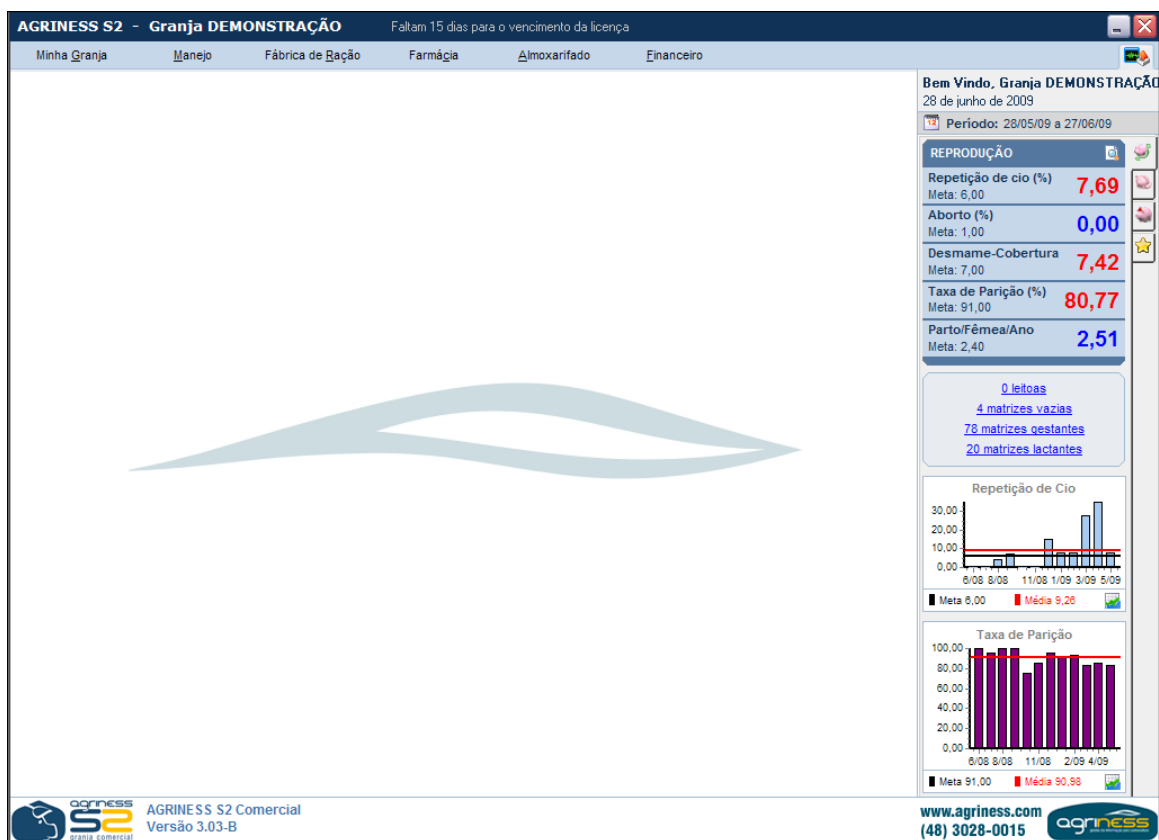
### 3.3 Agriness S2 3.0

O Agriness S2 3.0 é um software voltado para gerência da produção de suínos. Desenvolvido pela AGRINESS Sistemas e Tecnologias de Informação Ltda, empresa sediada em Florianópolis. Possui hoje, segundo informações do site da empresa, mais de mil unidades produtivas utilizando seus produtos, totalizando meio milhão de matrizes suínas gerenciadas (AGRINESS, 2009).

Está disponível nas versões S2 Comercial, CIA, Quarto Sítio e S2 Multiplicadora. As funcionalidades listadas a seguir são da versão S2 Comercial por tratar da produção de terminação de suínos, unidade de produção de leitões e produção de ciclo completo. A versão S2 Comercial apresenta as seguintes funcionalidades:

- Cadastro das granjas suínas;
- Agenda de compromissos;
- Controle de usuários
- Controle da fabricação de rações compra de ingredientes, formulário de rações, configurações especiais para cada propriedade e relatórios
- Controle do estoque de medicamentos e vacinas com disponibilidade de relatórios;
- Controle financeiro;
- Configurações do sistema como tipo de calendário utilizado, tipo de produção de suínos, forma de identificação dos animais e configurações sobre manejo;
- Cadastro das raças criadas;
- Cadastro dos tipos de rações utilizados;
- Agendamento de medicação e vacinação;
- Cadastro de funcionários;
- Cadastro de clientes e fornecedores;
- Cadastro de equipamentos e instalações;
- Tabelas configuráveis de acompanhamento de crescimento;
- Configurações de metas de produtividade da propriedade;
- Controle da reprodução, maternidade, creche e engorda;
- Controle de compra, venda, morte, desclassificação e transferência de leitões;
- Controles do processo de reprodução, parto, número de leitões nascidos, desmamados, peso e idade de desmame, taxas de parição e tipo de parto;

A imagem a seguir se refere à tela inicial do software Agriness S2 Comercial.



Tela 03 – Tela inicial do Agriness S2 Comercial – Versão Demonstrativa. Fonte: AGRINESS.

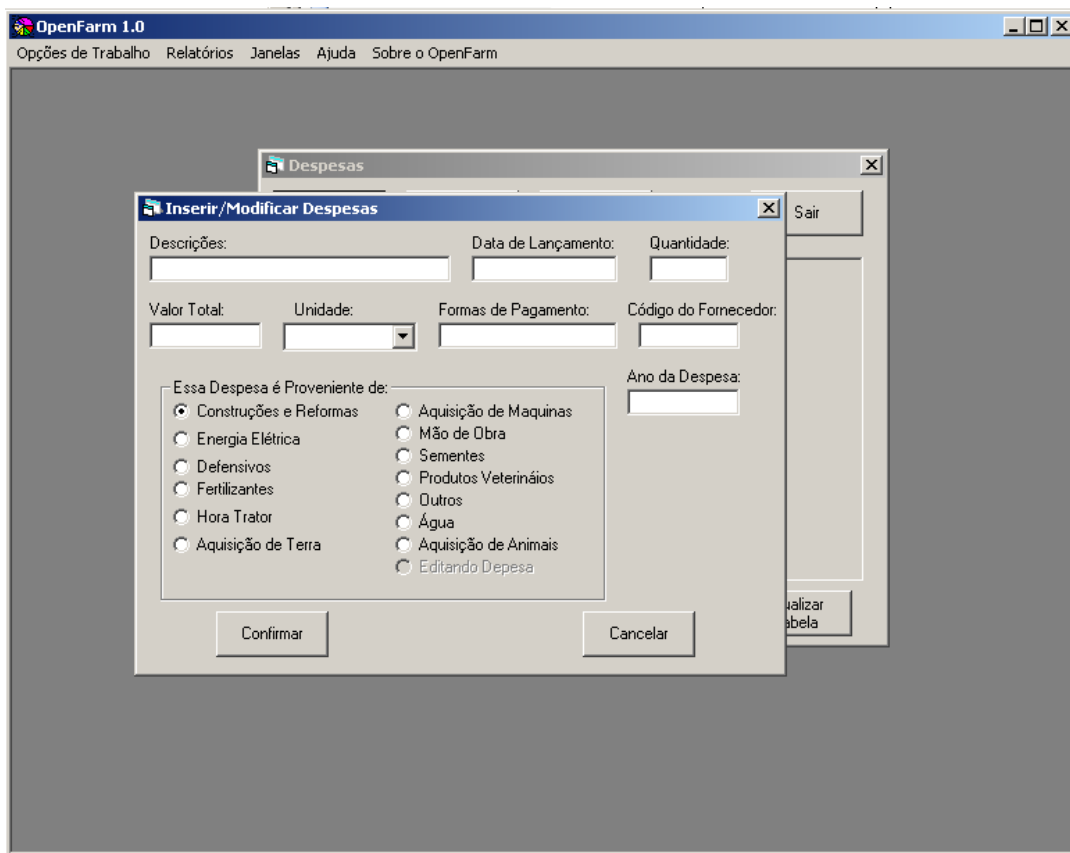
### 3.4 OpenFarm

Este é um projeto de software livre para administração de fazendas. Existem duas versões, uma para Windows, denominada 1.0, e outra para Linux, denominada 2.0. Segundo a documentação das versões, ambas apresentam as mesmas funcionalidades e características de interface. Sendo assim foi escolhida a versão 1.0, pelo fato da versão 2.0 ter apresentado problemas na instalação. A versão 1.0 foi escrita na linguagem Visual Basic utilizando banco de dados Access. Já na versão 2.0, o sistema foi, segundo o autor do projeto, traduzido para Python e o banco de dados passou a ser MySQL. Como é um projeto de software livre, é gratuito e está disponível para ser baixado, instalado e modificado sem qualquer custo.

Da versão 1.0 foram levantadas as seguintes funcionalidades:

- Cadastro de clientes;
- Cadastro de fornecedores
- Controle de receitas e despesas;
- Relatórios de Receitas;
- Relatório geral do inventário;
- Inserir, remover e alterar atividades agrícolas;
- Definir período de trabalho, ano/safra;
- Relatórios de Despesas;

A imagem a seguir se refere à tela de cadastro de despesas do projeto OpenFarm 1.0.



Tela 04 - Uma tela do OpenFarm 1.0 Fonte: PROJETO OPENFARM.

## 4 SISTEMA PROPOSTO

### 4.1 Visão geral do sistema

O sistema proposto foi desenvolvido de acordo com os requisitos levantados junto aos produtores rurais, com auxílio de profissionais atuantes no ramo da agricultura catarinense. O documento de requisitos encontra-se no anexo A deste trabalho. Este aplicativo é chamado E-fagro.

Um cliente acessa o *site* e realiza o cadastro informando os dados pessoais. Após preencher os dados, é criada uma propriedade no sistema e um usuário para o cliente, que receberá um email confirmando o cadastro. Neste email há uma senha gerada aleatoriamente, que o cliente usará para efetuar o primeiro acesso ao sistema. O cliente depois de entrar no sistema, utilizando seu email e a senha aleatória, pode alterar a sua senha e os dados da propriedade, bem como cadastrar outros usuários para acessar o sistema. Um usuário com perfil de administrador pode realizar o cadastro do maquinário da propriedade e definir o ano/safra<sup>17</sup>.

Cada talhão é uma fração da propriedade rural, que recebe por um período uma determinada cultura. Estes talhões são associados ao ano/safra definido. O usuário insere os talhões no sistema bem como as atividades agrícolas realizadas no talhão.

Cada ação realizada no talhão é registrada no sistema como uma atividade, que pode ser uma adubação, uma aplicação de defensivo, um plantio, uma colheita ou outras atividades não previstas. Para cada atividade é informado pelo usuário um valor total investido ou arrecadado, por exemplo, quando é realizado um plantio, o usuário insere o valor total investido em sementes. Estas atividades podem fazer uso de máquinas, onde o usuário informa a máquina utilizada e a quantidade de horas de trabalho da máquina.

Há uma opção para o usuário visualizar os talhões cadastrados para uma safra. Selecionando um dos talhões listados, o usuário recebe informações detalhadas do talhão, como as atividades realizadas e a operações com máquinas efetuadas.

---

<sup>17</sup> Exemplo: 2010/2011.



Através do total arrecadado, diminuído do total investido, incluindo as operações com máquinas é calculado o saldo do talhão. Este saldo é um valor aproximado do lucro de um talhão, já que o sistema não controla o fluxo de caixa, ativos e passivos da propriedade.

## **4.2 Funcionalidades gerais do sistema**

A seguir são listadas as funcionalidades gerais do sistema:

- Cadastrar a propriedade rural, sua localização, proprietário e área total.
- Cadastrar os usuários da propriedade com atribuições de administrador ou colaborador.
- Controlar as cultivares perenes e anuais, incidência de pragas, investimentos e resultados.
- Manter histórico das atividades agrícolas.
- Controlar o uso do maquinário da propriedade, o consumo de combustível e a manutenção.
- Gerar relatórios de produtividade e lucratividade por talhão, cultivar ou safra.
- Efetuar comparativos de produção ao longo do tempo.

## **4.3 Atores do sistema**

Quando uma propriedade é cadastrada, é cadastrado no mínimo um usuário, que terá o perfil de administrador de sistema. As tarefas que competem ao administrador ao descritas adiante.

### **4.3.1 Ator Cliente**

Este ator existe somente no cadastro inicial da propriedade. Ele executa os casos de uso cadastro de cliente e cadastro de propriedade.

#### 4.3.2 Ator Administrador

As atribuições do administrador são de nível gerencial, como o próprio nome sugere, tendo uma visão do todo de uma propriedade, além da visão no decorrer do tempo. O administrador tem privilégios para inserir e remover usuários com o perfil de colaborador ou mesmo de administrador. Tem a responsabilidade de gerenciar a propriedade, tendo acesso a todas as funcionalidades do sistema.

#### 4.3.3 Ator Colaborador

O perfil de colaborador é destinado a atualizar o sistema com as tarefas realizadas. Este usuário é responsável por manter a parte operacional do sistema. A seguir as principais tarefas deste usuário:

- Criar e atualizar as tarefas, estas sempre associadas a uma atividade;
- Atualizar diariamente as tarefas realizadas;
- Visualizar relatórios.

### 4.4 Casos de Uso

Seguindo as recomendações do processo de desenvolvimento de software OpenUp/Basic, a partir dos requisitos levantados, foram refinados os casos de uso, detalhando o fluxo de execução e os atores que executam cada funcionalidade do sistema. Os casos de uso refinados estão no Anexo II deste trabalho.

### 4.5 Banco de dados

Existe a necessidade em determinados *softwares* de se manter as informações ao longo do tempo, podendo as informações ser recuperadas quando for necessário. Com o intuito de facilitar a persistência e consulta destas informações foram desenvolvidas várias técnicas de persistência ao longo do tempo. Uma delas é a utilização de sistemas gerenciadores de banco de dados, também conhecidos como SGBDs, que podem ser relacionais, orientados a objetos ou objeto-relacional (HEUSER, 2004). No sistema proposto será utilizado um SGBD relacional.

O PostgreSQL<sup>18</sup> é um o sistema gerenciador de banco de dados de código aberto que possui versões para os principais sistemas operacionais existentes no mercado (GNU/Linux, Mac OS X, Unix e Windows). Tem suporte completo a chaves estrangeiras e primárias, junções, gatilhos e visões. Além de incluir a maior parte dos tipos de dados da norma ISO SQL:1999, suporta o armazenamento de objetos binários. Além disso, possui ferramentas para administração de banco de dados como, por exemplo, o pgAdmin. Tendo em vista que o PostgreSQL atende a todas as necessidades do projeto E-fagro, ele foi escolhido como o SGBD do projeto.

#### 4.5.1 Diagrama Entidade Relacionamento

Foi utilizando o DBDesigner 4<sup>19</sup>, para se fazer a modelagem relacional do sistema. As figuras a seguir mostram o resultado desta modelagem. Para melhorar a visualização o diagrama foi dividido em duas figuras neste texto.

Durante a modelagem do banco de dados foi preciso levar em conta que o sistema precisa manter várias propriedades cadastradas. Para isso todas as entidades e relacionamentos relativos aos dados da propriedade, devem estar vinculados a ela, direta ou indiretamente.

---

<sup>18</sup> <http://www.postgresql.org.br/sobre>. Acessado em 20/03/2010.

<sup>19</sup> <http://www.fabforce.net/dbdesigner4/>. Acesso em 10/01/2010.

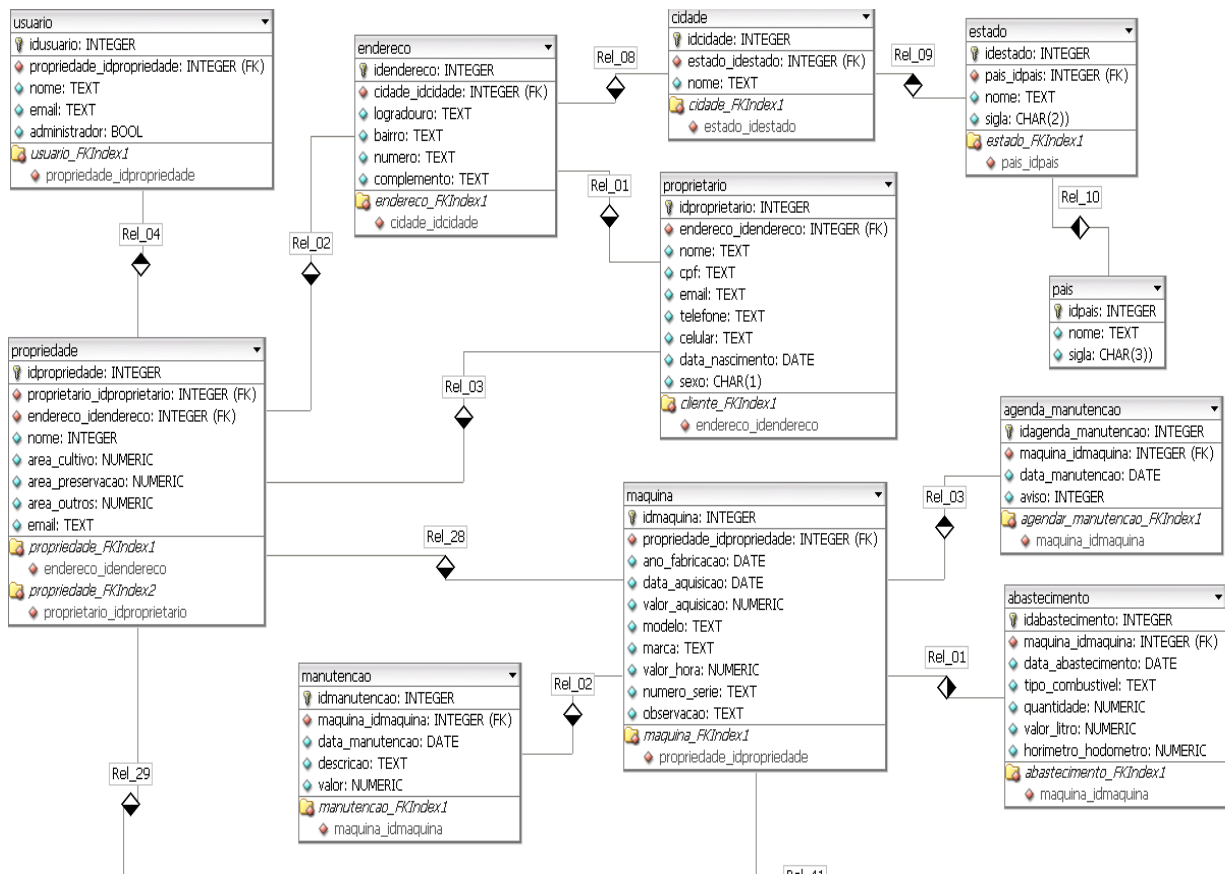


Diagrama 01 – Diagrama do banco de Dados, parte 1. Fonte: Elaboração do autor.

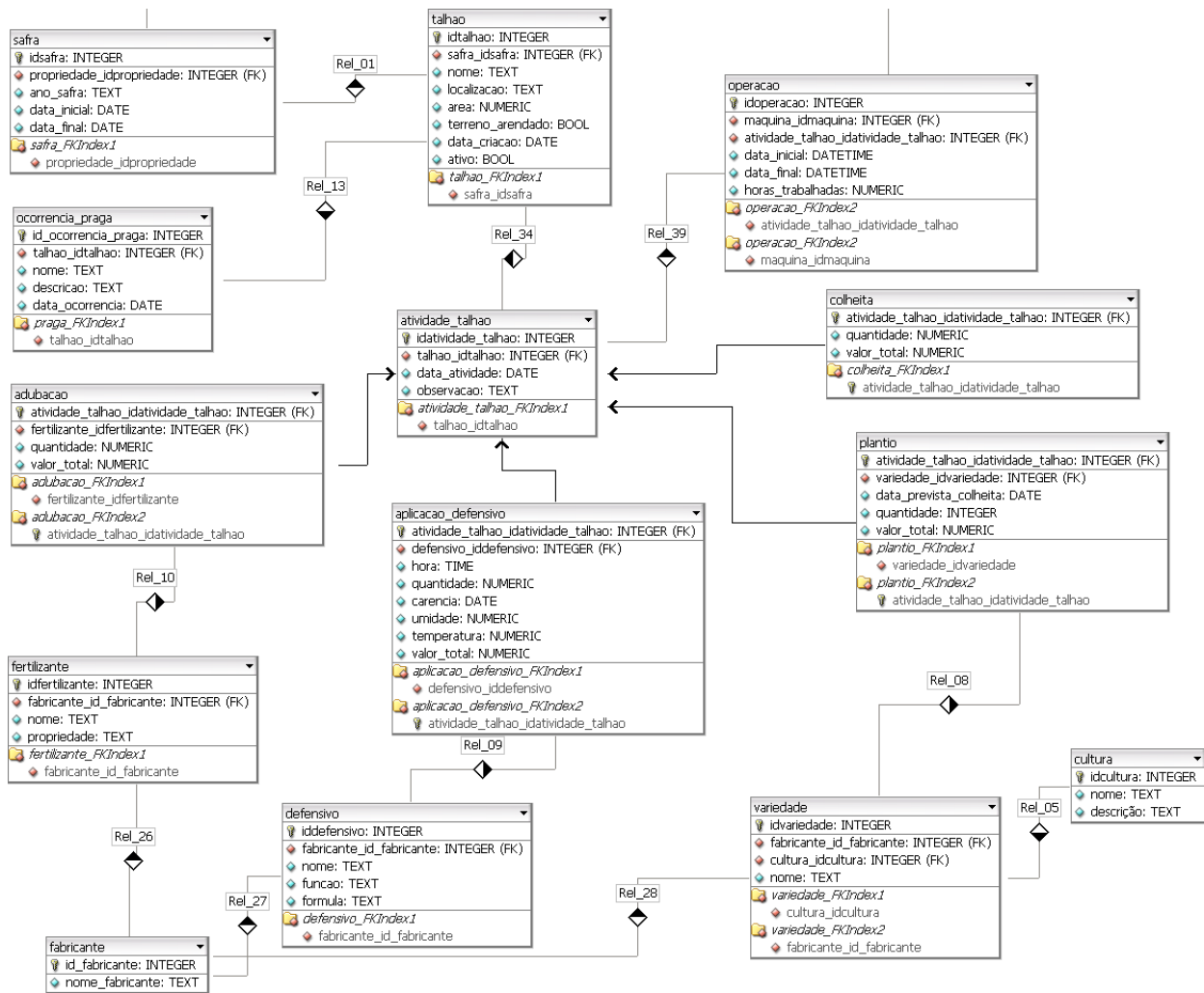


Diagrama 02 – Diagrama do banco de Dados parte 2. Fonte: Elaboração do autor.

## 5 DESENVOLVIMENTO

Para a implementação do E-fagro foi utilizado o framework JBoss Seam que é uma plataforma de desenvolvimento de código aberto para construções de aplicações web em Java. O Seam integra tecnologias como *Asynchronous JavaScript and XML* (AJAX), *Java Server Faces* (JSF), *Java Persistence* (JPA), *Enterprise Java Beans* (EJB 3.0) e *Business Process Management* (BPM). A versão utilizada neste trabalho foi a 2.1.2 datada de junho de 2009 (SEAM REFERENCE, 2009).

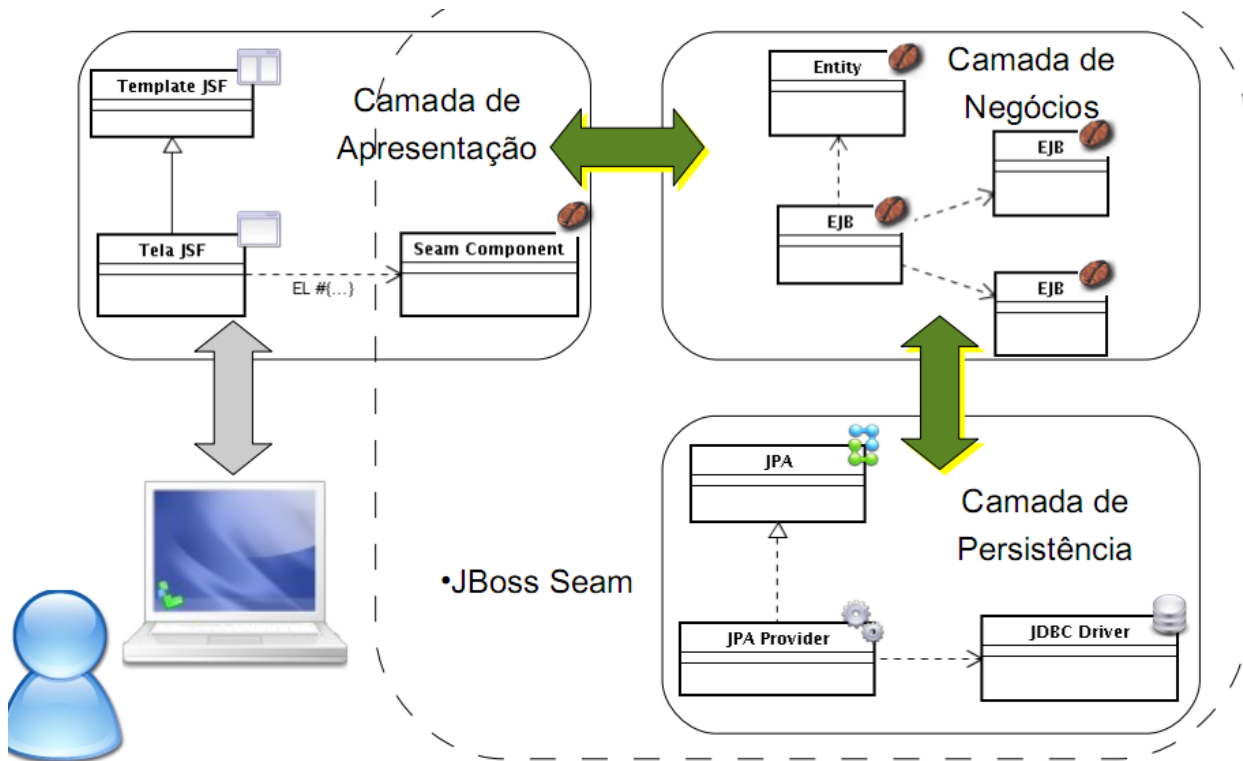


Diagrama 03 – Apresentação da arquitetura do JBoss Seam com JSF/EJB/JPA. Fonte: <http://www.thedevelopersconference.com.br/arquivos/TDC2008Floripa-seam-spock.pdf>. Acessado em 23/11/2009.

## 5.1 Engenharia reversa do banco de dados

Uma das ferramentas que acompanha o *framework* JBoss Seam é o seam-gen. Essa ferramenta gera através de linha de comando todo o projeto com os arquivos de configuração devidamente editados. Além disso, é possível efetuar a engenharia reversa a partir do banco de dados.

O programador deve inicialmente configurar um projeto por linha de comando indicando o nome do projeto, local onde será salvo, onde está instalado o servidor de aplicação JBoss, o endereço do SGBD utilizado, bem como o *driver* JDBC. Ainda existe a opção de montar o projeto com ou sem EJB. Pode-se também utilizar o JBoss Tools que é um *plugin* que integra o seam-gen ao ambiente de desenvolvimento Eclipse. Com este *plugin* as configurações do projeto são feitas a partir de utilitários gráficos. Neste trabalho foi utilizado o *plugin* do JBoss Tools. Uma vez configurado o utilitário de criação do projeto, as tabelas do banco de dados são mapeadas para classes Java, utilizando anotações. O seam-gen ainda gera as classes de controle e uma interface utilizando Java Server Faces coerente com os relacionamentos mapeados no modelo de dados entidade relacionamento. Dessa forma, quando o projeto é criado todas as entidades mapeadas já possuem as funcionalidades de criação, leitura, remoção e atualização implementadas automaticamente.

O resultado do mapeamento objeto relacional, do banco de dados foram as seguintes entidades mapeadas:

- Abastecimento.java
- Adubacao.java
- AgendaManutencao.java
- AplicacaoDefensivo.java
- AtividadeTalhao.java
- Cidade.java
- Cliente.java
- Colheita.java
- Cultura.java
- Defensivo.java

- Endereco.java
- Estado.java
- Fabricante.java
- Manutencao.java
- Maquina.java
- OcorrenciaPraga.java
- Operacao.java
- Pais.java
- Propriedade.java
- Safra.java
- Talhao.java
- Usuário.java
- Variedade.java

Nas entidades mapeadas, além das anotações de mapeamento do banco de dados, são utilizadas as anotações do Hibernate Validator, com a finalidade de validar as entradas do usuário no lado do servidor. Por exemplo, na entidade `Usuario` o atributo `email` além da anotação `@Column`, que relaciona este atributo com uma coluna da tabela `usuario` do banco de dados, está anotado com `@Email`, que obriga o atributo a estar preenchido com um valor válido para um email, isto é, destinatário@domínio. Por padrão do *framework* as anotações são colocadas nos métodos `get` que encapsulam os atributos.

```
@Column(name = "email", unique = true, nullable = false)
@email
public String getEmail() {
    return this.email;
}
```

Código Fonte 01 – Fragmento da classe `br.ufsc.inf.efagro.entity.Usuario.java`. Fonte: Elaboração do autor.



## 5.2 Classes de controle

Para cada entidade mapeada pelo seam-gen são geradas duas classes de controle. Elas são geradas com o mesmo nome da entidade acrescentando o sufixo “List” para as classes que tratam de pesquisas dos objetos e “Home” para as classes de tratam da inclusão, remoção e atualização dos objetos. As classes com o sufixo “List” herdam a classe EntityQuery<sup>20</sup>, que faz parte do *framework* JBoss Seam. A EntityQuery implementa o tratamento de listas de objetos retornados de pesquisas no banco de dados, como paginação de resultados, por exemplo. Já as classes com sufixo “Home” herdam a classe EntityHome, que também faz parte do *framework*. A EntityHome implementa os métodos que fazem a persistência dos objetos no banco de dados.

As classes de controle geradas foram modificadas, isto é, foram aplicadas regras de negócio para sistema. Além disso, outras classes de controle foram criadas, como por exemplo, a SenhaAction, que é responsável por alterar a senha dos usuários e a Authenticator, que valida as credenciais dos usuários no momento que efetuam a entrada no sistema.

## 5.3 Interface com o usuário

A interface com o usuário foi desenvolvida utilizando a tecnologia Java Server Faces com o auxílio das bibliotecas do RichFaces e do JBoss Seam Framework. Alguns *templates* gerados pelo seam-gen foram aproveitados no desenvolvimento, porém a maioria das telas geradas automaticamente pela ferramenta não foi aproveitada.

A utilização de *templates* facilita a padronização das telas e melhora a produtividade do projeto. Uma vez definido um *template*, basta associar um determinado item da tela ao *template* e o item assume o comportamento e as características visuais do *template*. A seguir se pode visualizar um *template* criado para os campos editáveis. Este arquivo foi gerado automaticamente pelo seam-gen, e não

---

<sup>20</sup> <http://docs.jboss.org/seam/2.1.2/api/org/jboss/seam/framework/EntityQuery.html> . Acessado em 01/04/2010.

precisou ser adaptado ao projeto. Apenas o arquivo que contém os estilos aplicados a este *template* foi alterado.

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:s="http://jboss.com/products/seam/taglib">

  <div class="prop">

    <s:label styleClass="name #{invalid?'errors':''}">
      <ui:insert name="label"/>
      <s:span styleClass="required" rendered="#{required}"*</s:span>
    </s:label>

    <span class="value #{invalid?'errors':''}">
      <s:validateAll>
        <ui:insert/>
      </s:validateAll>
    </span>

    <span class="error">
      <h:graphicImage value="/img/error.gif" rendered="#{invalid}"
        styleClass="errors"/>
      <s:message styleClass="errors"/>
    </span>

  </div>
</ui:composition>
```

Código Fonte 02 – Código do arquivo de *template* edit.xhtml. Fonte: Elaboração do autor.

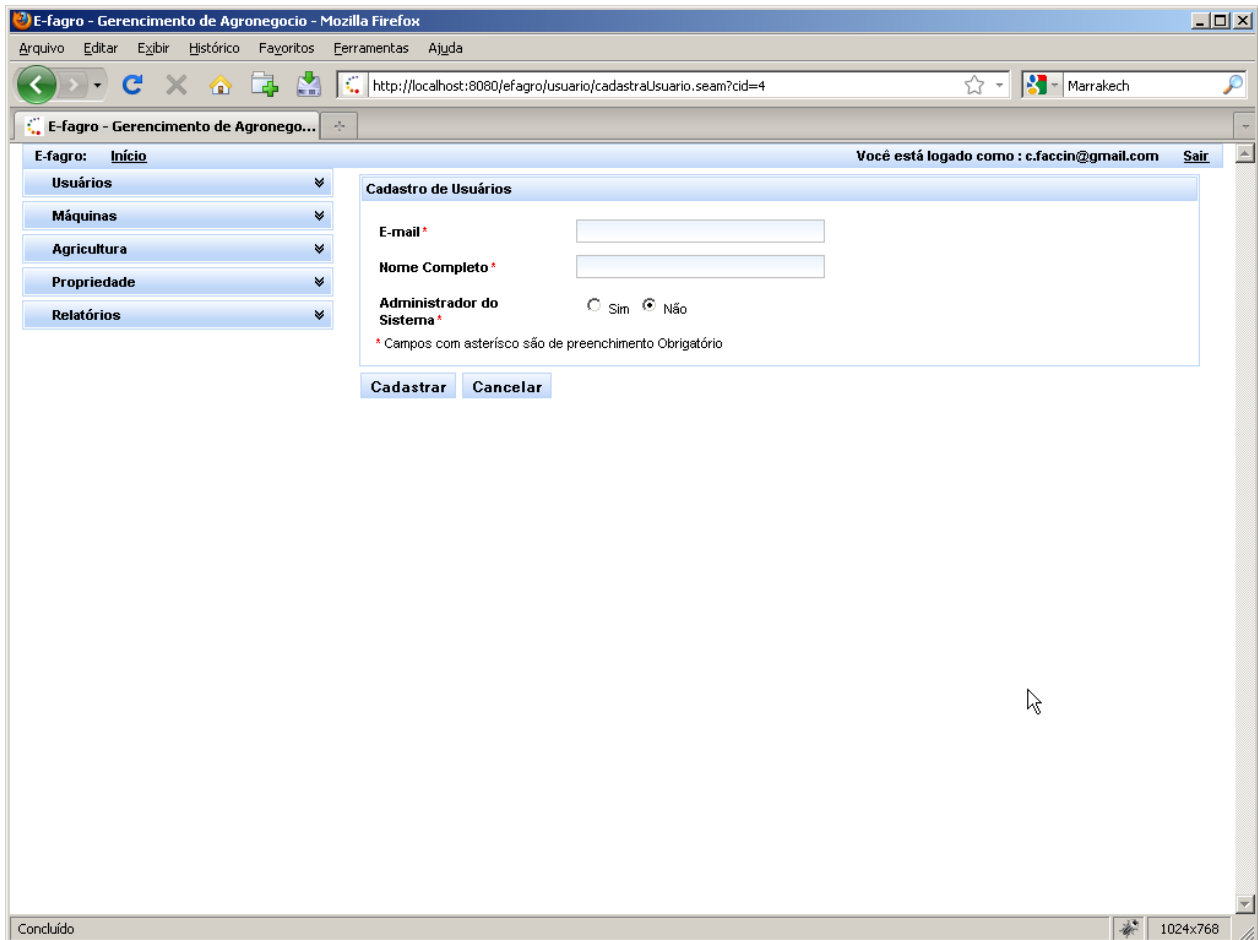
A aplicação de um estilo, Cascading Style Sheets, altera a aparência do item da tela, então os *template* são confeccionados de forma genérica, sem aplicar um tamanho padrão aos campos, por exemplo. No exemplo de código abaixo se pode observar a utilização do *template* */layout/edit.xhtml* e a aplicação de um estilo.

```
<s:decorate id="emailField" template="/layout/edit.xhtml">
  <ui:define name="label">#{messages.email}</ui:define>
  <h:inputText id="email"
    required="true"
    styleClass="inputMedio"
    value="#{usuarioHome.instance.email}">
  </h:inputText>
</s:decorate>
```

Código Fonte 03 – Fragmento da tela cadastraUsuario.xhtml. Fonte: Elaboração do autor.

Através da *unified expression language*<sup>21</sup> o valor que será digitado pelo usuário no campo email, será atribuído a instancia da classe usuarioHome, que herda a entidade Usuário. Com isso quando a página for submetida, os validadores que estão anotados no atributo email da entidade Usuario serão aplicados.

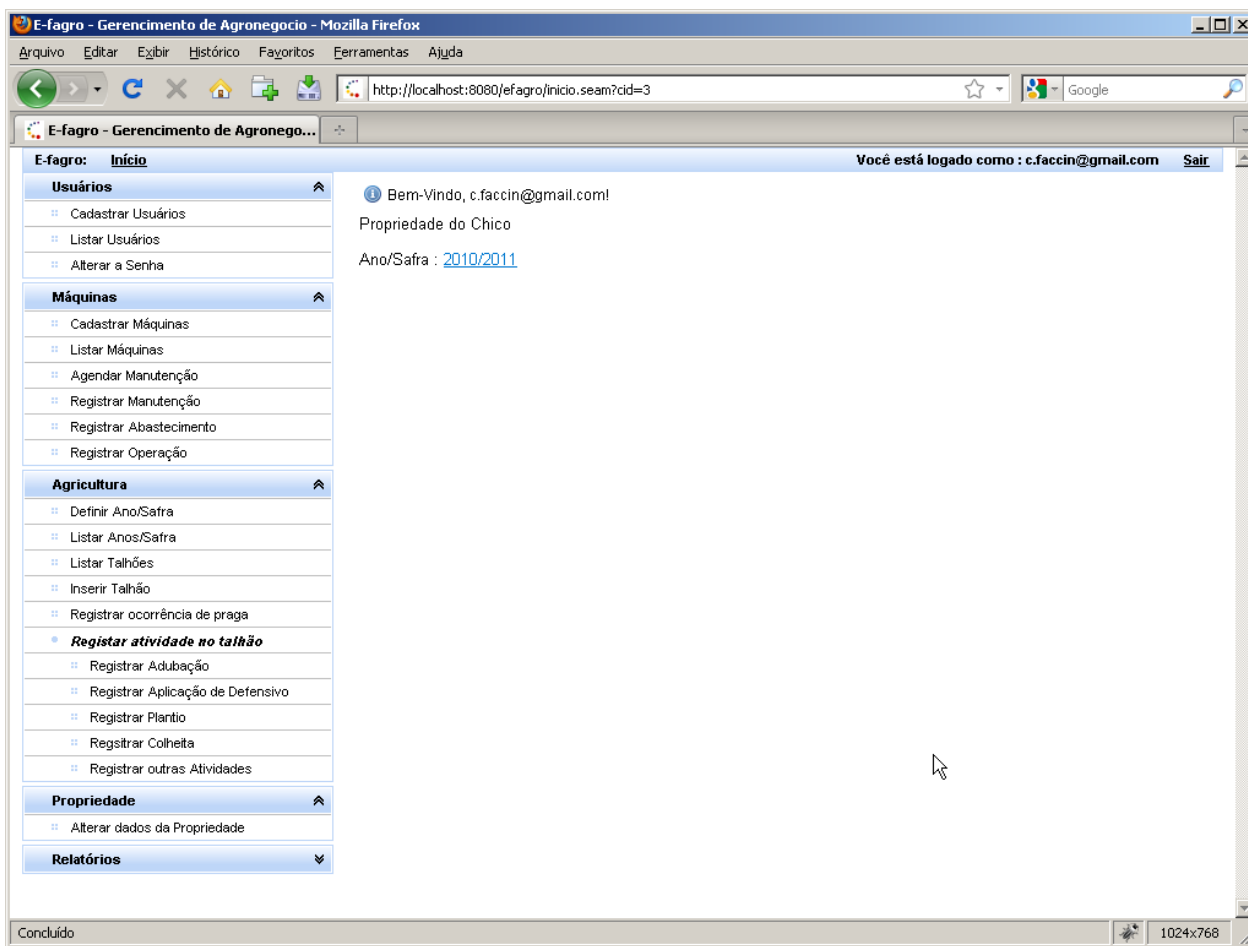
A tela a seguir mostra a interface de cadastro de usuários.



Tela 05 – Tela para cadastro de usuários. Fonte: Elaboração do autor.

<sup>21</sup> <http://today.java.net/pub/a/today/2006/03/07/unified-jsp-jsf-expression-language.html>. Acesso em 20/03/2010

A tela a seguir mostra a página inicial do usuário após efetuar a entrada no sistema, com todas as funcionalidades disponíveis no menu localizada na lateral esquerda do janelas.



Tela 06 – Tela inicial do sistema após o usuário entrar no sistema. Fonte: Elaboração do autor.

A tela a seguir é a interface com o usuário que lista os talhões cadastrados no sistema. São listados os Ano/Safra da propriedade e o usuário seleciona um deles. Quando a seleção ocorre, são listados os talhões daquela safra, conforme a imagem da tela abaixo. Clicando em um dos talhões da lista, será exibida a tela com os detalhes do talhão.

The screenshot shows a web browser window with the URL `http://localhost:8080/efagro/agricultura/listarTalhoes.seam?cid=10`. The application interface includes a navigation menu on the left with options like 'Usuários', 'Máquinas', 'Agricultura', 'Propriedade', and 'Relatórios'. The main content area displays a message 'Talhão cadastrado!' and a prompt 'Selecione o Ano/Safra para visualizar seus talhões'. A dropdown menu shows '2010/2011' selected. Below this, a table lists plots for the year/season '2010/2011'.

Nome	Área	Descrição da Localização
<a href="#">Talhão 2</a>	2.0	Localização do talhão 2
<a href="#">Talhão 6</a>	4.3	Localização do talhão 6
<a href="#">Talhão 3</a>	5.0	Localização do talhão 3
<a href="#">Talhão 5</a>	3.4	Localização do talhão 5
<a href="#">Talhão 4</a>	5.45	Localização do talhão 4
<a href="#">Talhão 1</a>	3.5	Localização do terreno

A mouse cursor is hovering over the 'Talhão 1' link, and a tooltip 'Detalhar informações' is visible.

Tela 07 – Lista dos talhões do Ano/Safra selecionado. Fonte: Elaboração do autor.

Após a seleção do talhão, mostrada na tela anterior, são recuperadas as informações do talhão no banco de dados. É efetuado um somatório dos totais de cada atividade realizada no talhão, adubação, aplicação de defensivos, plantios e operações com máquinas e diminuído dos totais de colheitas. Isso produz o saldo do talhão, que é um valor aproximado do lucro obtido no cultivo do talhão.

**E-fagro - Gerenciamento de Agronegócio - Mozilla Firefox**

Arquivo Editar Exibir Histórico Favoritos Ferramentas Ajuda

http://localhost:8080/efagro/agricultura/talhao.seam?idTalhao=166&cid=10

E-fagro - Gerenciamento de Agronegócio... Você está logado como : c.faccin@gmail.com Sair

E-fagro: **Início**

- Usuários
- Máquinas
- Agricultura
- Propriedade
- Relatórios

**Talhão: No topo do morro**

O saldo atual do talhão é: R\$ 4.650,00

**Atividades no talhão**

Adubações Efetuadas			
Data	Fertilizante	Quantidade	Valor total
08/01/2011	Ureia - AIP	500.0	R\$ 700,00
<b>Custo total de adubações</b>			<b>R\$ 700,00</b>

Aplicações de defensivos Efetuadas				
Data	Hora	Quantidade	Defensivo	Valor total
08/01/2011	16:00:00	20.0	Agimix	R\$ 200,00
01/09/2010	10:00:00	4.0	Roundup	R\$ 150,00
<b>Total dos valores de defensivos</b>				<b>R\$ 350,00</b>

Plantios Efetuados					
Data	Data Prevista da colheita	Quantidade	Variedade	Cultura	Valor total
15/09/2010	27/12/2010	35000.0	3021Y	Milho	R\$ 650,00
<b>Total dos valores de plantio</b>					<b>R\$ 650,00</b>

Colheitas Efetuadas		
Data	Quantidade	Valor total
08/01/2011	400.0	R\$ 6.800,00
<b>Totais dos valores de colheita</b>		<b>R\$ 6.800,00</b>

Operações Efetuadas			
Data do Início	Data do Final	Horas trabalhadas	Custo da operação
08/09/2010	08/09/2010	2.0	R\$ 100,00

Concluído 1024x768

Tela 08 – Informações detalhadas do talhão selecionado na tela anterior. Fonte: Elaboração do autor.

Através do menu “Agricultura” localizado na lateral esquerda da janela, se tem acesso ao submenu “Registrar atividade no talhão” que possui as seguintes opções: Registrar Adubação, Registrar Aplicação de Defensivo, Registrar Plantio, Registrar Colheita e Registrar outras Atividades. Selecionando uma das opções, é exibida para o usuário uma tela similar a mostrada abaixo. Essa é a tela para registro de plantio, onde o usuário preenche os seguintes dados: o talhão, a data da atividade, eventuais observações sobre a atividade, a cultura plantada, a variedade plantada, a data prevista da colheita, a quantidade de sementes/mudas utilizadas e o valor total do custo do plantio, sem considerar as operações com maquinário. Logo abaixo o usuário preenche a máquina que foi utilizada no plantio, a data inicial e final da operação e a quantidade de horas de máquina utilizadas na operação.

**Registrar Plantio**

Talhão \* T 1

Data da atividade \* 05/05/2010

Observação

Cultura \* Milho

Variedade \* 3021 - Pioneer Sementes

Data Prevista da colheita \*

Quantidade \*

Valor total \*

		Maio, 2010						
		Seg	Ter	Qua	Qui	Sex	Sáb	Dom
Máquinas utilizadas	19	3	4	5	6	7	8	9
Máquina *	20	10	11	12	13	14	15	16
Data do Início *	21	17	18	19	20	21	22	23
Data do Final *	22	24	25	26	27	28	29	30
Horas trabalhadas *	23	31	1	2	3	4	5	6

\* Campos com asterisco são de preenchimento Obrigatório

Cadastrar Cancelar

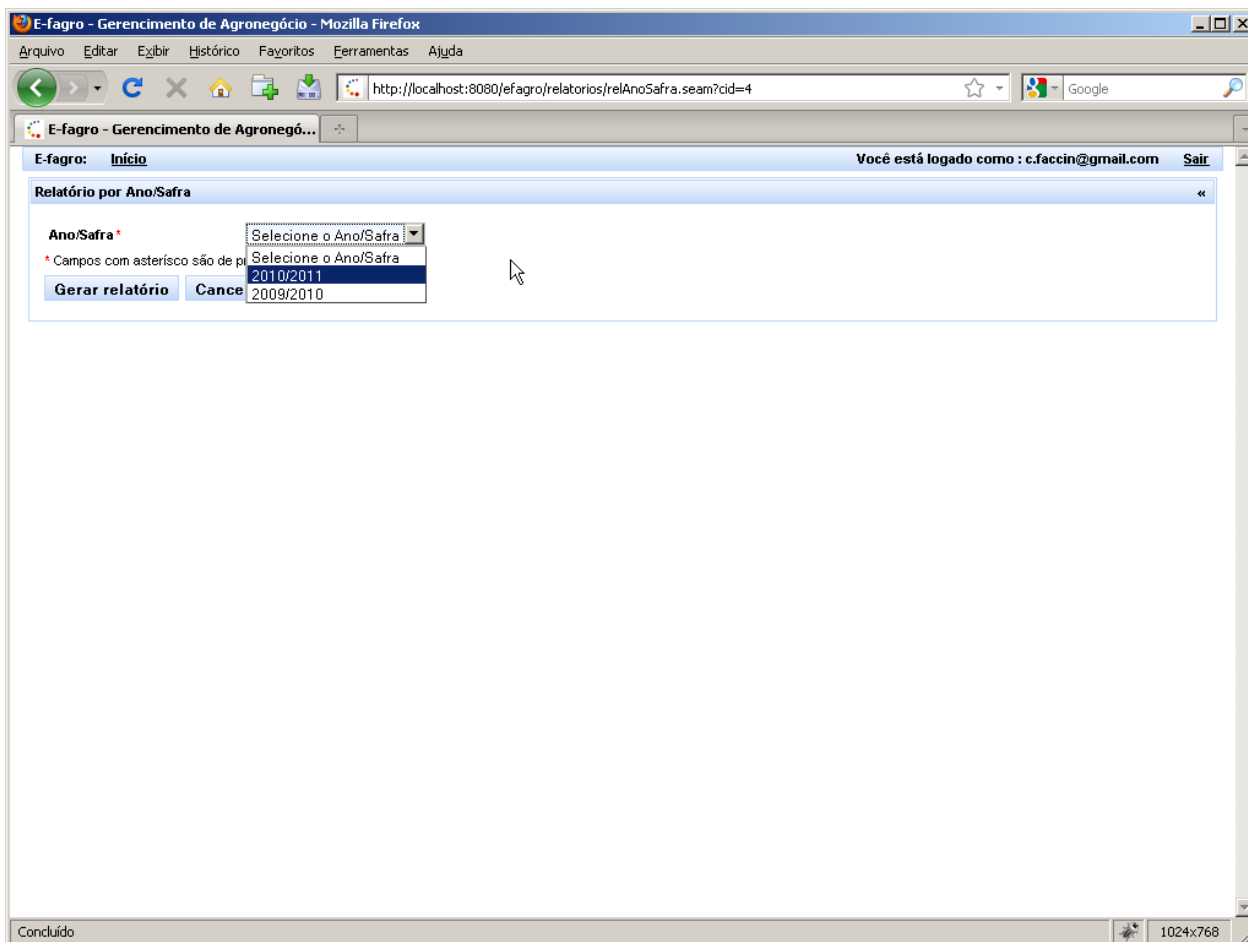
Tela 09 – Tela para registro de um plantio em um determinado talhão. Fonte: Elaboração do autor.

### 5.3.1 Relatórios

Através do menu “Relatórios” localizado a esquerda da janela, o usuário pode acessar os submenus “Relatórios de Maquinário” e “Relatórios Agrícolas”.

O primeiro possui as seguintes opções: Relatórios de Operações e Relatórios de Manutenções. Em ambas o usuário seleciona a máquina e o período. Caso o período não seja preenchido, todas as ocorrências da máquina são listadas no relatório.

O segundo possui as seguintes opções: Relatório por Ano/Safra, Relatório por Talhão, Relatório por Cultura, Relatório por Variedade e Relatório de Incidência de Pragas. Selecionando umas das opções é mostrada ao usuário uma tela similar a Tela 10.



Tela 10 – Tela com opções de relatório para safra. Fonte: Elaboração do autor.



Após preencher os dados pertinentes, é exibido ao usuário um relatório conforme a Tela 11, com as informações referentes ao que foi selecionado, no caso do exemplo, a safra 2010/2011.

Como no exemplo da Tela 11, para alguns relatórios haverá a opção do usuário obter informações detalhadas selecionando um dos talhões listados, apresentando então uma tela similar à Tela 08.

O saldo da safra é: R\$ 9.030,12

Nome do Talhão	Total Arrecadado	Total Investido	Saldo do Talhão
<a href="#">Ao norte dos pinheiros</a>	R\$ 0,00	R\$ 0,00	R\$ 0,00
<a href="#">Ao sul do aviário</a>	R\$ 11.500,12	R\$ 7.120,00	R\$ 4.380,12
<a href="#">No topo do mato</a>	R\$ 6.800,00	R\$ 2.150,00	R\$ 4.650,00
<a href="#">Ao oeste do açude</a>	R\$ 0,00	R\$ 0,00	R\$ 0,00
<b>Totais da Safra</b>	<b>R\$ 18.300,12</b>	<b>R\$ 9.270,00</b>	<b>R\$ 9.030,12</b>

Tela 11 – Tela exibindo relatório da safra. Fonte: Elaboração do autor.

## 6 CONCLUSÃO

Após as pesquisas realizadas sobre os softwares disponíveis no mercado voltados para a gerência de propriedades rurais, pode-se dizer que o projeto E-fagro é pioneiro no setor por ser um *software* livre voltado para a agricultura familiar. Para ser colocado em produção ainda são necessários alguns ajustes, por exemplo, construir um manual de utilização. Em breve será disponibilizada uma versão preliminar, para avaliar o desempenho e ter um retorno dos usuários quanto a utilização e funcionalidades do *software*.

A utilização do JBoss Seam Framework foi essencial para o desenvolvimento do projeto. As ferramentas dele deram um grande impulso inicial no desenvolvimento do *software* diminuindo significativamente a tarefa de implementar o código para operações básicas como incluir, buscar, alterar e remover registros do banco de dados. A integração do JSF com a os componentes do Seam, o controle de contextos, principalmente o contexto de conversação, e a adoção de uma arquitetura simples, sem muitas camadas, possibilitaram ao desenvolvedor a rápida criação do projeto.

Considera-se que este trabalho conseguiu atingir seus objetivos, desde a pesquisa realizada até o desenvolvimento do sistema, que resultou no protótipo E-fagro, de acordo com as necessidades observadas no levantamento de requisitos.

### 6.1 Trabalhos futuros

O sistema de informação para agricultura familiar, desenvolvido neste trabalho ainda pode ser melhorado de diversas formas. A seguir são listadas algumas sugestões de trabalhos futuros para ampliar as funcionalidades e o acesso ao sistema por parte dos agricultores familiares:

- Desenvolver a funcionalidade de gerencia de clientes e fornecedores de uma propriedade rural.

- Melhorar a interface gráfica atual e desenvolver uma interface acessível para dispositivos móveis.
- Desenvolver funcionalidades de fluxo de caixa e de controle de estoques.

## 7 REFERÊNCIAS

ADM RURAL 4.0. Versão demonstrativa. Disponível em <<http://www.admrural.com.br/>> Acesso em 16/10/2009.

AGRINESS S2. Versão demonstrativa. Disponível em <<http://www.agriness.com.br/index.jsp>>. Acesso em 27/09/2009.

BETING, Joelmir. Documento eletrônico disponível em <<http://www.joelmirbeting.com.br/noticiadetalhe.aspx?IdNews=32094&IdGnews=2>>. Acesso 01/06/2009.

BRASIL. Ministério da Agricultura, Pecuária e Abastecimento. **Plano Agrícola e Pecuário 2009/2010**. Documento Eletrônico. Disponível em <[http://www.agricultura.gov.br/images/MAPA/arquivos\\_portal/PAP2009\\_WEB.pdf](http://www.agricultura.gov.br/images/MAPA/arquivos_portal/PAP2009_WEB.pdf)>. Acesso em 23/06/2009

BRASIL. Ministério do Desenvolvimento Agrário. **Plano Safra da Agricultura Familiar 2008/2009**. Documento Eletrônico disponível em <[http://www.fpagropecuaria.com.br/images/site/PAP\\_2008\\_2009%5B1%5D.pdf](http://www.fpagropecuaria.com.br/images/site/PAP_2008_2009%5B1%5D.pdf)>. Acesso em 10/06/2009

\_\_\_\_\_. Ministério do Desenvolvimento Agrário. Secretaria da Agricultura Familiar. **Apresentação – Crédito Rural**. Documento eletrônico disponível em <<http://www.mda.gov.br/saf/index.php?sccid=813>>. Acesso em 01/06/2009.

\_\_\_\_\_. Ministério do Desenvolvimento Agrário. Secretaria da Agricultura Familiar. Documento eletrônico disponível em <<http://www.mda.gov.br/saf/index.php?sccid=742>>. Acesso em 15/06/2009.

\_\_\_\_\_. Ministério do Desenvolvimento Agrário. **Estatísticas do Meio Rural - 2008**. Documento eletrônico disponível em <<http://www.mda.gov.br/arquivos/estatisticas-rurais-2008.pdf>>. Acesso em 22/06/2009.

CETIC. Proporção de domicílios com acesso à internet. Documento eletrônico disponível em <<http://www.cetic.br/usuarios/tic/2008-total-brasil/rel-geral-04.htm>>. Acesso em 12/6/2009

CETIC. Proporção de domicílios com computador. Documento eletrônico disponível em <<http://www.cetic.br/usuarios/tic/2008-total-brasil/rel-geral-01.htm>>. Acesso em 12/6/2009

DIEESE. Departamento Intersindical de Estatística e Estudos Sócio-Econômicos. Documento eletrônico disponível em <<http://www.dieese.org.br/anu/estatisticasMeioRural2008.pdf>>. Acesso em 22/06/2009.

FAO. *One sixth of humanity undernourished - more than ever before*. Documento eletrônico disponível em <<http://www.fao.org/news/story/en/item/20568/icode/>>. Acesso em 19/06/2009.

FREE SOFTWARE FOUNDATION. Documento eletrônico disponível em <<http://www.fsf.org/>>. Acesso em 06/5/2009.

HEUSER, Carlos Alberto. **Projeto de Banco de dados**. Porto Alegre: Sagra Luzzato, 2004.

IBGE. Instituto Brasileiro de Geografia e Estatísticas. Documento eletrônico disponível em <<http://www.ibge.gov.br>>. Acesso em 01/06/2009.

JAVA SERVER FACES. Documento eletrônico disponível em <<http://java.sun.com/javaee/javaserverfaces/overview.html>>. Acessado em 22/01/2009.

PEREIRA, Nadir Rodrigues. **Embrapa Informática**. Documento eletrônico disponível em <<http://www.cnptia.embrapa.br/content/evento-quer-popularizar-tecnologia-da-informa-o-no-campo.html>>. Acesso em 20/06/2009.

PROJETO OPEN FARM. Documento eletrônico disponível em <<http://www.projetoopenfarm.cjb.net/>>. Acesso em 20/05/2009.

PRORURAL. Versão demonstrativa. Disponível em <<http://brazsoft.com.br/>>. Acesso em 10/06/2009.

PORTUGAL, Alberto Duque. **O Desafio da Agricultura Familiar**. Documento eletrônico disponível em <<http://www.embrapa.br/imprensa/artigos/2002/artigo.2004-12-07.2590963189/>> Acesso em 30/08/2008.

SEAM REFERENCE. Documento eletrônico disponível em <[http://docs.jboss.com/seam/2.1.1.GA/reference/en-US/pdf/seam\\_reference.pdf](http://docs.jboss.com/seam/2.1.1.GA/reference/en-US/pdf/seam_reference.pdf)>. Acesso em 21/01/2009.

SILVA, Ricardo Pereira. **UML2 em Modelagem Orientada a Objetos**. Florianópolis: Visual Books, 2007.

SILVA, Ricardo Pereira. **Como Modelar com UML2**. Florianópolis: Visual Books, 2009.

USDA, United States Department of Agriculture. Documento eletrônico disponível em <<http://www.usda.gov/wps/portal/usdahome>>. Acesso em 10/06/2009.

OPENUP/BASIC. Documento eletrônico disponível em <<http://epf.eclipse.org/wikis/openuppt/>>. Acesso em 02/02/2010.

## ANEXO I – REQUISITOS

<u>1</u>	<u>Introdução</u> .....	56
<u>2</u>	<u>Glossário</u> .....	57
<u>3</u>	<u>Visão Geral</u> .....	58
<u>3.1</u>	<u>Abrangência</u> .....	58
<u>3.1.1</u>	<u>Fluxo Geral</u> .....	58
<u>3.2</u>	<u>Descrição dos Atores</u> .....	59
<u>3.2.1</u>	<u>Ator Cliente</u> .....	59
<u>3.2.2</u>	<u>Ator Administrador</u> .....	59
<u>3.2.3</u>	<u>Ator Colaborador</u> .....	59
<u>4</u>	<u>Requisitos Funcionais</u> .....	59
<u>4.1</u>	<u>Cadastro de propriedade e usuários</u> .....	59
[RF01]	<u>Inserir Cliente no sistema</u> .....	60
[RF02]	<u>Inserir Propriedade do Cliente no sistema</u> .....	61
[RF03]	<u>Alterar dados da Propriedade</u> .....	62
[RF04]	<u>Inserir usuários no sistema</u> .....	63
[RF05]	<u>Alterar privilégio de usuário</u> .....	64
[RF06]	<u>Listar usuários do sistema</u> .....	64
[RF07]	<u>Alterar dados dos usuários do sistema</u> .....	65
[RF08]	<u>Remover usuários do sistema</u> .....	66
<u>4.2</u>	<u>Requisitos Gerais</u> .....	66
[RF09]	<u>Entrar no sistema</u> .....	66
[RF10]	<u>Sair do sistema</u> .....	67
[RF11]	<u>Alterar senha dos usuários</u> .....	67
<u>4.3</u>	<u>Requisitos do Maquinário</u> .....	68
[RF12]	<u>Cadastro das máquinas de uma propriedade</u> .....	68
[RF13]	<u>Alterar dados das máquinas de uma propriedade</u> .....	69
[RF14]	<u>Listar as máquinas de uma propriedade</u> .....	69
[RF15]	<u>Remover máquinas do sistema</u> .....	70

[RF16]	<a href="#">Registrar Abastecimento</a> .....	71
[RF17]	<a href="#">Calcular consumo de combustível</a> .....	72
[RF18]	<a href="#">Registrar manutenção</a> .....	72
[RF19]	<a href="#">Agendar manutenção</a> .....	73
[RF20]	<a href="#">Registrar operação</a> .....	74
4.4	<a href="#">Requisitos Agrícolas</a> .....	75
[RF21]	<a href="#">Definir calendário Agrícola</a> .....	75
[RF22]	<a href="#">Inserir talhões da propriedade</a> .....	76
[RF23]	<a href="#">Listar os talhões da propriedade</a> .....	77
[RF24]	<a href="#">Alteração dos talhões da propriedade</a> .....	77
[RF25]	<a href="#">Remover talhões da propriedade</a> .....	78
[RF26]	<a href="#">Registrar Adubação</a> .....	78
[RF27]	<a href="#">Registrar Aplicação de Defensivo</a> .....	79
[RF28]	<a href="#">Registrar Plantio</a> .....	81
[RF29]	<a href="#">Registrar Colheita</a> .....	82
[RF30]	<a href="#">Registrar outras operações</a> .....	83
[RF31]	<a href="#">Registrar Ocorrência de Praga</a> .....	83
4.5	<a href="#">Relatórios</a> .....	84
[RF32]	<a href="#">Relatório de operações de uma máquina</a> .....	84
[RF33]	<a href="#">Relatório de Manutenção de maquinário</a> .....	85
[RF34]	<a href="#">Relatório por Ano/Safra</a> .....	85
[RF35]	<a href="#">Relatório por Talhão</a> .....	86
[RF36]	<a href="#">Relatório por Cultura</a> .....	87
[RF37]	<a href="#">Relatório por Variedade</a> .....	88
[RF38]	<a href="#">Relatório de incidência de Pragas</a> .....	88

## Introdução

Os requisitos funcionais descritos a seguir foram levantados junto a agricultores familiares do município de Ouro, localizado no meio oeste de Santa Catarina.



Os requisitos foram capturados seguindo as recomendações do processo de desenvolvimento OpenUp/Basic.



Localização de Ouro no mapa de Santa Catarina. Fonte [http://pt.wikipedia.org/wiki/Ficheiro:SantaCatarina\\_Municip\\_Ouro.svg](http://pt.wikipedia.org/wiki/Ficheiro:SantaCatarina_Municip_Ouro.svg). Acessado em 10/05/2010.

## Glossário

Ano/Safra: Período agrícola definido com uma data inicial e uma data final, por exemplo, 2010/2011.

Talhão: Fração mínima da propriedade, vinculada a um ano/safra, ocupada por um determinado período por uma cultura. Também é conhecido como roça, lavoura ou campo.

Máquinas: Inclui máquinas e implementos agrícolas, como tratores, colheitadeiras, arados etc.

Cultura: Um tipo de cultivar agrícola, por exemplo, milho, soja, feijão etc.

Variedade: Um variante de uma cultura, ou um híbrido.

### **Visão Geral**

O sistema proposto tem como finalidade gerenciar propriedades rurais, controlando a entrada e saída de produtos, uso da terra, utilização do maquinário agrícola e gerenciamento da mão de obra.

### **Abrangência**

Qualquer proprietário de rural com acesso a internet pode cadastrar sua propriedade no sistema.

### **Fluxo Geral**

1. O cliente faz o cadastro no sistema com seus dados pessoais.
2. O cliente cadastra a sua propriedade rural.
3. É enviado um e-mail com a confirmação do cadastro e criado um usuário administrador com as mesmas credencias do cliente propriedade.
4. O administrador cadastra outros administradores ou colaboradores para a propriedade.
5. O administrador efetua o cadastro de todo o maquinário da propriedade.
6. À medida que os talhões forem sendo criados, os usuários adicionam os talhões ao sistema.
7. As operações realizadas nos talhões tais como plantio, colheita, adubação entre outras, são associadas a um talhão com valores associados aos insumos utilizados bem como o maquinário utilizado.

8. Os usuários solicitam relatórios ao sistema para obter informações de produtividade/lucratividade dos talhões, culturas, safras e variedades.
9. Os usuários solicitam relatórios de gastos com manutenção e/ou abastecimento de maquinário.

### ***Descrição dos Atores***

#### **Ator Cliente**

Efetua o cadastro inicial da propriedade. Logo após, é criado um usuário administrador para o cliente.

#### **Ator Administrador**

Pode executar todas as tarefas após efetuar login no sistema. Inclusive cadastrar outros usuários administradores ou colaboradores.

#### **Ator Colaborador**

Após efetuar login pode incluir, excluir e editar tarefas e transações. A intenção deste usuário é que para cada pessoa que trabalhe na propriedade exista um usuário colaborador e este de entrada no sistema dos trabalhos que realizou no dia.

### **Requisitos Funcionais**

#### ***Cadastro de propriedade e usuários***

Nesta seção estão agrupados os requisitos funcionais das primeiras interações do usuário com o sistema. Apenas os requisitos funcionais deste capítulo, [RF01] e [RF02], podem ser executados sem o usuário efetuar a entrada no sistema [RF03].

---

**[RF01]      Inserir Cliente no sistema**

---

**Ator:** Usuário Cliente

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja adicionar um novo cliente no sistema. Na página inicial deve haver um link para a página de cadastro de cliente.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Nome	Nome do Cliente
*Email	Email do Cliente
*Endereço	Endereço do Cliente
*Telefone	Telefone do Cliente com código de área
*Celular	Numero do telefone celular do Cliente com código de área
*CPF	CPF do Cliente
*Sexo	Masculino ou Feminino
*Data de Nascimento	Data de Nascimento do Cliente

*Tabela 1 - Campos para o Cadastro de Cliente*

*\* Esses campos são de preenchimento obrigatório*

# Esses campos não podem ser alterados

---

**[RF02] Inserir Propriedade do Cliente no sistema**

---

**Ator:** Usuário Cliente

**Prioridade:** Essencial

Este requisito funcional começa logo após a execução do [RF01]. É criada automaticamente uma propriedade com os valores padrões.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
Nome	Nome da Propriedade Rural. Valor padrão: "Propriedade do" concatenado com o nome do proprietário.
*Endereço	Endereço da Propriedade Rural. Valor padrão: usa o mesmo do proprietário.
Email	Email geral da Propriedade. Valor padrão: nulo.
*Área	Área total da propriedade em Hectares.
*Área cultivável	Área da propriedade que é utilizada para algum cultivo, seja agrícola, pecuário ou reflorestamento.
*Área preservação	Área de preservação de mata nativa e/ou reserva legal.

*Outras áreas	Área utilizada por construções, estradas, cede etc.
---------------	---

*Tabela 2 - Campos para o Cadastro de Propriedades Rurais*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

OBS: Após o cadastro o sistema deve enviar um e-mail para o cliente confirmar o cadastro e criar um usuário administrador com o mesmo login (e-mail) e uma senha aleatória.

---

**[RF03]      Alterar dados da Propriedade**

---

**Ator:** Usuário Administrador

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja alterar os dados da propriedade.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Nome	Nome da propriedade rural.
*Endereço	Endereço da Propriedade Rural.
Email	Email geral da Propriedade.
*Área	Área total da propriedade em Hectares.
*Área cultivável	Área da propriedade que é utilizada para algum cultivo, seja agrícola, pecuário ou reflorestamento.

*Área preservação	Área de preservação de mata nativa e/ou reserva legal.
*Outras áreas	Área utilizada por construções, estradas, cede etc.

*Tabela 3 - Campos para Alteração de Propriedades Rurais*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF04] Inserir usuários no sistema**

---

**Ator:** Usuário Administrador

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja adicionar mais usuários a uma propriedade rural. Os usuários podem ser administradores ou colaboradores.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Nome	Nome do usuário
*Senha	Senha do usuário, gerada automaticamente pelo sistema e enviada para o email cadastrado
*Ativo	Caso seja marcado como inativo, não permite o acesso do usuário ao sistema.
*Administrador	Se o usuário cadastrado possuir privilégios de administrador é

	marcado como verdadeiro, senão o usuário será definido como usuário Colaborador
--	---

*Tabela 4 - Campos para o Cadastro de Usuários no sistema*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF05]      Alterar privilégio de usuário**

---

**Ator:** Usuário Administrador

**Prioridade:** Desejável

Este requisito funcional começa quando o Ator deseja alterar os privilégios de um usuário de Colaborador para Administrador ou vice-versa.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Administrador	Novo privilégio, se antes administrador passa para colaborador.

*Tabela 5 - Campos para alteração de privilégios de usuários*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF06]      Listar usuários do sistema**

---

**Ator:** Usuário Administrador.

**Prioridade:** Importante



Este requisito funcional começa quando o Ator deseja visualizar os usuários do sistema. O Ator pode optar por visualizar somente os Usuários Administradores ou somente os Usuários Colaboradores ou ambos.

Nome do Campo	Descrição do campo
#Código	Código único gerado automaticamente pelo sistema
#Nome	Novo nome do usuário
#Email	Novo email do usuário
#Administrador	Se o usuário cadastrado possuir privilégios de administrador é marcado como verdadeiro, senão o usuário será definido como usuário Colaborador

*Tabela 6 - Campos para a visualização dos usuários do sistema  
# Esses campos apenas para visualização*

---

**[RF07]      Alterar dados dos usuários do sistema**

---

**Ator:** Usuário Administrador.

**Prioridade:** Importante

Este requisito funcional começa quando o Ator deseja alterar algum dado do cadastro do usuário.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Nome	Novo nome do usuário
*Ativo	Caso seja marcado como inativo, não permite o acesso do usuário ao sistema.

*Email	Novo email do usuário
--------	-----------------------

*Tabela 7 - Campos para a alteração dos dados dos usuários do sistema*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF08]      Remover usuários do sistema**

---

**Ator:** Usuário Administrador

**Prioridade:** Importante

Este requisito funcional começa quando o Ator deseja remover um usuário do sistema.

OBS 1: Cada propriedade rural deve ter no mínimo 1 (um) usuário administrador.

OBS 2: Um usuário não pode ser excluído por ele mesmo.

**Requisitos Gerais**

---

**[RF09]      Entrar no sistema**

---

**Ator:** Usuário Administrador, Usuário Colaborador

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja entrar no sistema. O usuário deve informar suas credenciais (email e senha). Estes são validados com o email e senha que estão cadastrados. Caso esteja correto, a entrada do usuário é habilitada no sistema.

Nome do Campo	Descrição do campo
*Login	Email do usuário

*Senha	Senha do Usuário
--------	------------------

*Tabela 8 - Campos para o usuário entrar no sistema*

*\* Esses campos são de preenchimento obrigatório*

OBS 1: Todos os requisitos a partir deste requerem que o usuário tenha entrado no sistema.

---

**[RF10] Sair do sistema**

---

**Ator:** Usuário Administrador, Usuário Colaborador

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja sair do sistema. O Ator clica no botão “sair” e sua sessão é terminada.

---

**[RF11] Alterar senha dos usuários**

---

**Ator:** Usuário Administrador, Usuário Colaborador

**Prioridade:** Importante

Este requisito funcional começa quando o Ator deseja alterar a sua senha.

Nome do Campo	Descrição do campo
*Senha Atual	Atual senha do Ator
*Nova Senha	Nova senha do usuário.
*Confirmação	Confirmação da nova senha

*Tabela 9 - Campos para alteração de senha dos usuários*

*\* Esses campos são de preenchimento obrigatório*

## Requisitos do Maquinário

---

### [RF12] Cadastro das máquinas de uma propriedade

---

**Ator:** Usuário Administrador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja cadastrar o maquinário de uma propriedade rural.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Ano fabricação	Ano em que a máquina foi fabricada
*Data de aquisição	Data da aquisição da máquina
*Valor de aquisição	Valor da compra
*Modelo	Modelo da máquina. Ex: TL 75E
*Marca	Empresa fabricante da máquina
*Valor hora	Valor da hora de trabalho desta máquina
Numero de série	Número de série/ número do chassi
Observação	Observações sobre a máquina

*Tabela 10 - Campos para o Cadastro de Máquinas*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF13]      Alterar dados das máquinas de uma propriedade**

---

**Ator:** Usuário Administrador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja alterar os dados do maquinário de uma propriedade rural.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Ano fabricação	Novo ano em que a máquina foi fabricada
*Data de aquisição	Nova data da aquisição da máquina
*Valor de aquisição	Novo valor da compra
*Modelo	Novo modelo da máquina. Ex: TL 75E
*Marca	Nova empresa fabricante da máquina
*Valor hora	Novo valor da hora de trabalho desta máquina
Numero de série	Número de série/ número do chassi
Observação	Observações sobre a máquina

*Tabela 11 - Campos para alteração de dados das máquinas*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF14]      Listar as máquinas de uma propriedade**

---

**Ator:** Usuário Administrador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja visualizar as máquinas de uma propriedade rural.

Nome do Campo	Descrição do campo
Código	Código único gerado automaticamente pelo sistema
Ano fabricação	Ano em que a máquina foi fabricada
Data de aquisição	Data da aquisição da máquina
Valor de aquisição	Valor da compra
Modelo	Modelo da máquina. Ex: TL 75E
Marca	Empresa fabricante da máquina
Valor hora	Valor da hora de trabalho desta máquina
Numero de série	Número de série/ número do chassi
Observação	Observações sobre a máquina

*Tabela 12 - Campos para visualização das máquinas*

---

**[RF15]      Remover máquinas do sistema**

---

**Ator:** Usuário Administrador

**Prioridade:** Importante

Este requisito funcional começa quando o Ator deseja remover uma máquina de uma propriedade. Primeiramente o ator executa o [RF17], escolhe uma máquina da lista e então o ator tem a opção de excluir a máquina.

---

**[RF16] Registrar Abastecimento**

---

**Ator:** Usuário Administrador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja registrar o abastecimento de combustível de uma máquina.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Código da Máquina	Máquina a qual a manutenção foi aplicada
*Data Abastecimento	Data em que o abastecimento foi realizado.
*Tipo do Combustível	Tipo do combustível utilizado (Diesel, gasolina, álcool,)
*Quantidade	Litros de combustível utilizado
*Valor do litro de Combustível	Valor em R\$ do litro de combustível utilizado.
*Horímetro / Hodômetro	Valor marcado pelo Hodômetro / Horímetro da máquina no momento do abastecimento

*Tabela 13 - Campos para o Registro de Manutenção*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF17]      Calcular consumo de combustível**

---

**Ator:** Usuário Administrador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja calcular o consumo de combustível de determinada máquina. O Ator escolhe uma máquina e o sistema deve retornar o consumo médio ao longo do tempo, em litros/Km ou litros/Hora, além de totalizar o consumo no período e também as horas/quilômetros realizados. Este período também pode ser definido pelo Ator.

---

**[RF18]      Registrar manutenção**

---

**Ator:** Usuário Administrador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja registrar a manutenção de uma máquina.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Código da Máquina	Máquina a qual a manutenção foi aplicada
*Data manutenção	Data em que a manutenção foi realizada.
*Descrição da manutenção	Descrição do que foi realizado na manutenção.
*Valor	Custo da manutenção. (Peças,



	lubrificantes, mão de obra)
--	-----------------------------

*Tabela 14 - Campos para o Registro de Manutenção*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF19] Agendar manutenção**

---

**Ator:** Usuário Administrador.

**Prioridade:** Desejável

Este requisito funcional começa quando o Ator deseja agendar a manutenção de uma máquina.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Código da Máquina	Máquina a qual a manutenção foi aplicada
*Data manutenção	Data em que a manutenção foi realizada.
*Avisar antes	Período de tempo anterior a data de manutenção que o Ator deve ser avisado da manutenção. (1 dia, 3 dias, 7 dias, 15 dias, 21 dias, 30 dias, 60 dias).
*Frequência	Frequência com que se repete a manutenção. (Nem uma, diária, semanal, quinzenal, mensal, trimestral, semestral ou anual)
Lembrete	Texto livre que será enviado para o email. Pode ser usado, por exemplo,

	para anotar eventuais peças de reposição que serão utilizadas na manutenção.
--	--

*Tabela 15 - Campos para o Registro de Agendamento de Manutenção*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF20] Registrar operação**

---

**Ator:** Usuário Administrador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja registrar uma operação realizada por uma máquina.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Máquina	Máquina que realizou a operação.
*Data inicial da Operação	Data em que a operação foi iniciada.
*Data final da Operação	Data em que a operação foi concluída.
*Tempo	Quantidade de horas gastas na operação.

*Tabela 16 - Campos para o Registro de Operação*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

## Requisitos Agrícolas

---

### [RF21] Definir calendário Agrícola

---

**Ator:** Usuário Administrador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja definir o calendário agrícola da propriedade, isto é, definir o início e o final de uma safra agrícola. Todos os

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Data Inicial	Data em que inicia um período de cultivo.
*Data Final	Data em que termina o período de cultivo.
*Ano-Safra	É um estereótipo de um período de cultivo. O calendário agrícola não acompanha o calendário gregoriano. Que contém o ano da data inicial do período de cultivo concatenado com o ano da data final do período de cultivo. Ex: 2010/2011,

*Tabela 17 - Definir calendário agrícola*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF22] Inserir talhões da propriedade**

---

**Ator:** Usuário Administrador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator subdivide a área total da propriedade em porções menores. Estas porções são denominadas talhões.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Nome da área	Nome para identificação da área
Observação	Anotações sobre os talhões feitas pelo usuário
*Área	Tamanho do talhão em hectares
*Localização	A localização é uma descrição do local onde se encontra tal área.
*Data da criação	Data em que foi definido o talhão
#Ano Safra	Ano/Safra atual da propriedade.
*Terreno arrendado	Caso o talhão seja arrendado deve ser marcado como verdadeiro. Por padrão o campo é falso.

*Tabela 18 - Campos para criação de talhões*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF23] Listar os talhões da propriedade**

---

**Ator:** Usuário Administrador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja visualizar os talhões da propriedade.

Nome do Campo	Descrição do campo
Código	Código único gerado automaticamente pelo sistema
Nome da área	Nome para identificação da área
Área	Tamanho do talhão em hectares
Localização	A localização é uma descrição do local onde se encontra tal área.
Ano Safra	Ano/Safra a qual o talhão pertence

*Tabela 19 - Campos para visualização dos talhões*

---

**[RF24] Alteração dos talhões da propriedade**

---

**Ator:** Usuário Administrador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator altera os talhões da propriedade.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Nome da área	Novo nome para identificação da área
Observação	Novas anotações sobre os talhões feitas pelo usuário

*Área	Novo tamanho do talhão em hectares
*Tipo de cultivo	Temporário ou permanente.
*Localização	Nova localização é uma descrição do local onde se encontra tal área.
*Terreno arrendado	Caso o talhão seja arrendado deve ser marcado como verdadeiro. Por padrão o campo é falso.

*Tabela 20 - Campos para alteração de talhões*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF25]      Remover talhões da propriedade**

---

**Ator:** Usuário Administrador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator remover os talhões da propriedade. Os talhões não são removidos da propriedade, mas são marcados como inativos.

---

**[RF26]      Registrar Adubação**

---

**Ator:** Usuário Administrador, Colaborador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja registrar a adubação de um talhão.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Talhão	Talhão em que foi realizada a

	adubação
*Data Adubação	Data em que o fertilizante foi aplicado
*Quantidade	Quantidade de fertilizante aplicado no talhão
*Valor Unitário	Valor unitário do adubo aplicado
*Fertilizante	Fertilizante aplicado no talhão
*Máquinas	Máquinas utilizadas na aplicação
*Horas de Máquina	Horas gastas por máquina para efetuar aplicação
Observação	Eventuais observações acerca da aplicação do fertilizante

*Tabela 21 - Campos para Registrar adubação*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

Observação: No momento da implantação, devem ser cadastrados os fertilizantes mais comuns utilizados na agricultura.

---

**[RF27] Registrar Aplicação de Defensivo**

---

**Ator:** Usuário Administrador, Colaborador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja registrar a aplicação de defensivos agrícolas.

Nome do Campo	Descrição do campo
# Código	Código único gerado

	automaticamente pelo sistema
*Talhão	Talhão em que foi realizada a adubação
*Data e hora da aplicação	Data e hora em que o defensivo foi aplicado
*Temperatura	Temperatura no momento da aplicação
*Umidade	Umidade relativa do ar no momento da aplicação
*Quantidade	Quantidade de defensivo aplicada
*Valor unitário	Valor unitário do defensivo aplicado
*Defensivo	Defensivo aplicado no talhão
*Data final da carência	Data em que termina o período de carência.
*Máquinas	Máquinas utilizadas na aplicação
*Horas de Máquina	Horas gastas por máquina para efetuar aplicação
Observação	Eventuais observações acerca da aplicação do fertilizante

*Tabela 22 - Campos para Registrar aplicação de defensivo*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

Observação: No momento da implantação, devem ser cadastrados os defensivos mais comuns utilizados na agricultura.



---

**[RF28] Registrar Plantio**

---

**Ator:** Usuário Administrador, Colaborador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja registrar o plantio em um talhão.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Talhão	Talhão em que foi realizada a adubação
*Data Plantio	Data em que o plantio foi realizado
*Data Prevista Colheita	Data Prevista para a colheita
*Variedade	Variedade da cultura plantada
*Quantidade	Quantidade de sementes/mudas utilizadas
Máquinas	Máquinas utilizadas na aplicação
Horas de Máquina	Horas gastas por máquina para efetuar aplicação
*Adubação Plantio	Caso haja, a quantidade e o adubo que foi utilizado no plantio
*Defensivo Plantio	Caso haja, a quantidade e o defensivo que foi utilizado no plantio
Observação	Eventuais observações acerca do plantio.

*Tabela 23 - Campos para Registrar plantio*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

Observação: No momento da implantação, devem ser cadastradas as variedades mais comuns utilizados na agricultura para cada cultura. Conseqüentemente, também devem ser cadastradas as culturas mais comuns.

---

**[RF29] Registrar Colheita**

---

**Ator:** Usuário Administrador, Colaborador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja registrar a colheita de um talhão.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Talhão	Talhão em que foi realizada a adubação
*Data da Colheita	Data em que a colheita foi realizada
*Quantidade Colhida	Quantidade colhida no talhão
Máquinas	Máquinas utilizadas na colheita
Horas de Máquina	Horas gastas por máquina para efetuar aplicação
Observação	Eventuais observações acerca do plantio.

*Tabela 24 - Campos para Registrar colheita*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF30] Registrar outras operações**

---

**Ator:** Usuário Administrador, Colaborador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja registrar alguma outra atividade que foi aplicada ao talhão que não seja plantio, adubação ou aplicação de defensivos. Um exemplo de uma atividade deste tipo seria efetuar a drenagem de algum ponto do talhão.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Talhão	Talhão em que foi realizada a adubação
*Data da operação	Data em que a operação foi realizada.
*Descrição	Descrição da operação realizada.
Máquinas	Máquinas utilizadas na operação
Horas de Máquina	Horas gastas por máquina para efetuar a operação

*Tabela 25 - Campos para Registrar plantio*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

---

**[RF31] Registrar Ocorrência de Praga**

---

**Ator:** Usuário Administrador, Colaborador.

**Prioridade:** Essencial

Este requisito funcional começa quando o Ator deseja registrar ocorrência de praga em determinado talhão.

Nome do Campo	Descrição do campo
# Código	Código único gerado automaticamente pelo sistema
*Talhão	Talhão em que foi realizada a adubação
*Data Ocorrência	Data em que ocorreu a praga
*Nome da praga	Nome da praga que atingiu o talhão
Observação	Eventuais observações acerca da praga.
Solução	Solução da praga. Quando houver

*Tabela 26 - Campos para Registrar plantio*

*\* Esses campos são de preenchimento obrigatório*

*# Esses campos não podem ser alterados*

## **Relatórios**

---

### **[RF32] Relatório de operações de uma máquina**

---

**Ator:** Usuário Administrador, Usuário Colaborador.

**Prioridade:** Importante

Este requisito funcional começa quando o Ator deseja visualizar um relatório contendo as operações efetuadas por uma máquina ao longo de um período.

Nome do Campo	Descrição do campo
*Máquina	Máquina
*Período	Data inicial e data final do período o qual se quer consultar
Operações	Operações realizadas pela máquina no período.

*Tabela 27 - Campos eu serão exibidos no relatório*

*\* Parâmetros definidos pelo Ator para se gerar o relatório*

---

**[RF33]      Relatório de Manutenção de maquinário**

---

**Ator:** Usuário Administrador, Usuário Colaborador.

**Prioridade:** Importante

Este requisito funcional começa quando o Ator deseja visualizar um relatório contendo as manutenções efetuadas em uma máquina ao longo de um período.

Nome do Campo	Descrição do campo
*Máquina	Máquina
*Período	Data inicial e data final do período o qual se quer consultar
Atividades	Atividades realizadas pelo trabalhador no período.

*Tabela 28 - Campos eu serão exibidos no relatório*

*\* Parâmetros definidos pelo Ator para se gerar o relatório*

---

**[RF34]      Relatório por Ano/Safra**

---

**Ator:** Usuário Administrador, Usuário Colaborador.

**Prioridade:** Importante

Este requisito funcional começa quando o Ator deseja visualizar o relatório do ano/safra.

Nome do Campo	Descrição do campo
*Ano/Safra	Ano safra que o Ator deseja visualizar
Talhões	Talhões do Ano/Safra, com informações básicas, como Data de plantio, data de colheita, variedade cultivada e quantidade colhida.

*Tabela 29 - Campos eu serão exibidos no relatório*

*\* Parâmetros definidos pelo Ator para se gerar o relatório*

---

**[RF35]      Relatório por Talhão**

---

**Ator:** Usuário Administrador, Usuário Colaborador.

**Prioridade:** Importante

Este requisito funcional começa quando o Ator deseja visualizar o relatório de um talhão específico.

-

Nome do Campo	Descrição do campo
*Talhão	Talhão selecionado pelo usuário.
Aplicação de Defensivo	Aplicações de Defensivos no talão
Adubação	Adubações efetuadas no talhão
Plantio	Detalhes do plantio no talhão

Colheita	Detalhes da colheita no talhão
Outras Operações	Outras operações Realizadas
Pragas	Incidência de pragas no talhão

*Tabela 30 - Campos eu serão exibidos no relatório*

*\* Parâmetros definidos pelo Ator para se gerar o relatório*

---

**[RF36]      Relatório por Cultura**

---

**Ator:** Usuário Administrador, Usuário Colaborador.

**Prioridade:** Importante

Este requisito funcional começa quando o Ator deseja visualizar o relatório de uma cultura específico.

-

Nome do Campo	Descrição do campo
*Cultura	Cultura selecionada pelo usuário
Área	Área total de cultivo
Produção total	Total produzido pela cultura
Produtividade	Kg / Hectare
Variedades	Variedades da cultura
Talhões	Lista dos talhões que abrigaram a cultura

*Tabela 31 - Campos eu serão exibidos no relatório*

*\* Parâmetros definidos pelo Ator para se gerar o relatório*

---

**[RF37] Relatório por Variedade**

---

**Ator:** Usuário Administrador, Usuário Colaborador.

**Prioridade:** Importante

Este requisito funcional começa quando o Ator deseja visualizar o relatório de um talhão específico.

-

Nome do Campo	Descrição do campo
*Variedade	Variedade selecionada
Área	Área total de cultivo
Produção total	Total produzido pela cultura
Talhões	Lista dos talhões que abrigaram a cultura
Produtividade	Kg / Hectare

*Tabela 32 - Campos eu serão exibidos no relatório*

*\* Parâmetros definidos pelo Ator para se gerar o relatório*

---

**[RF38] Relatório de incidência de Pragas**

---

**Ator:** Usuário Administrador, Usuário Colaborador.

**Prioridade:** Importante

Este requisito funcional começa quando o Ator deseja visualizar o relatório de incidência de pragas.

-



Nome do Campo	Descrição do campo
*Ano/Safra	Ano safra que se deseja pesquisar
Cultura	Cultura que foi afetada
Variedade	Variedade que foi afetada
Talhão	Talhão que foi afetado
Praga	Praga que afetou o talhão

*Tabela 33 - Campos eu serão exibidos no relatório*

*\* Parâmetros definidos pelo Ator para se gerar o relatório*

## ANEXO II – REFINAMENTO DOS CASOS DE USO.

Cadastro da Propriedade.....	90
Manter Usuários e propriedade.....	92
Funcionalidades Gerais do Sistema.....	96
Gerenciar Maquinário.....	98
Gerenciar Agricultura.....	103

### Cadastro da Propriedade

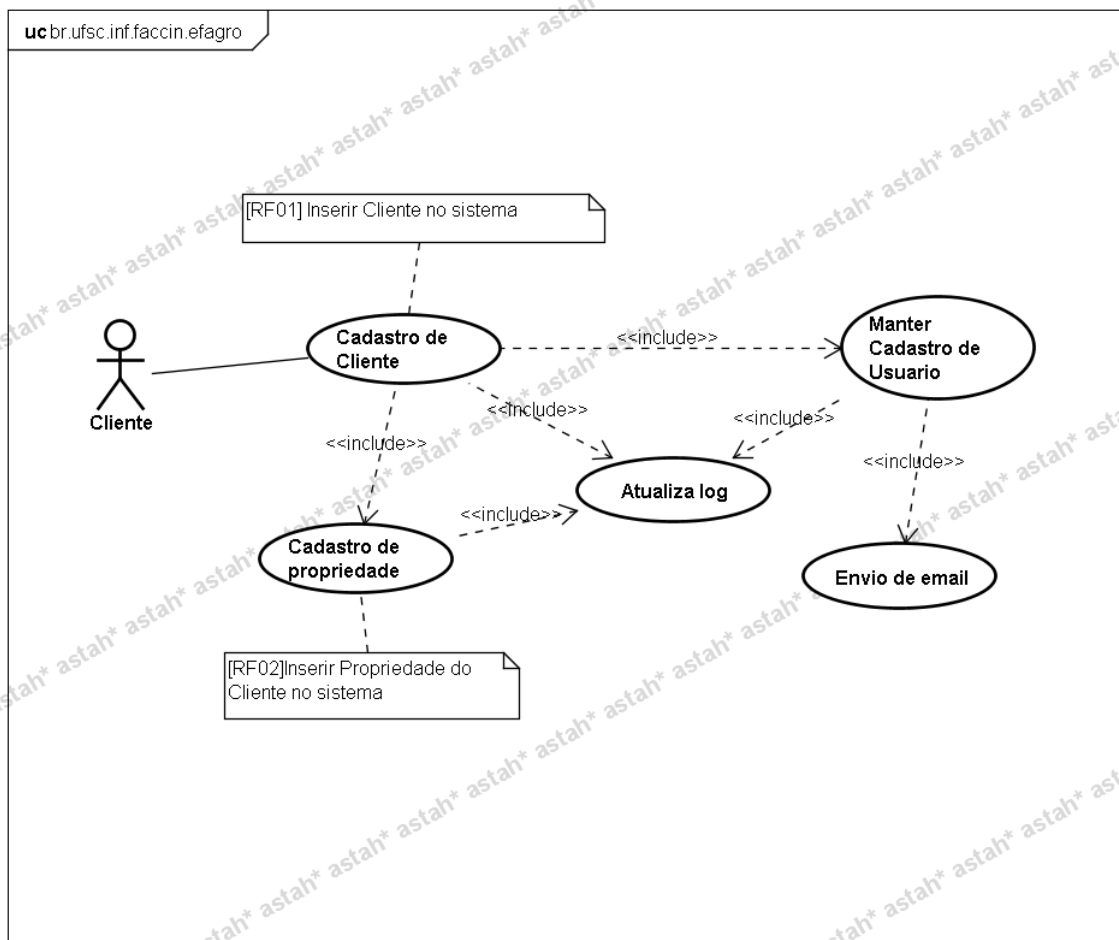


Diagrama 01 – Diagrama de Caso de Uso – Cadastro de Propriedade. Fonte: Elaboração do autor, 2010.

### **Caso de Uso: Cadastro de cliente**

Ator: Cliente

Objetivo: Cadastrar o cliente no sistema.

Pré-Condições: Não há.

Pós-Condições: Cliente cadastrado no sistema.

Fluxo principal:

1. O cliente informa nome, CPF, e-mail, telefone, telefone celular, data de nascimento e endereço.
2. O sistema registra as informações do cliente.
3. O sistema cria um usuário Administrador e envia um email para o endereço de email cadastrado contendo login e senha.
4. A transação é gravada no log.

Fluxo alternativo:

1. O cliente não informa um dos dados.
2. O sistema emite um aviso que os campos são obrigatórios.

### **Caso de Uso: Cadastro de propriedade**

Ator: Cliente

Objetivo: Cadastrar a propriedade no sistema

Pré-condições: O caso de uso cadastro do cliente deve ser executado com sucesso.

Pós-condições: Propriedade cadastrada no sistema.

Fluxo principal:

1. O sistema cria uma propriedade com os valores padrão e associa ao cliente.
2. A transação é gravada no log.

## Manter Usuários e propriedade

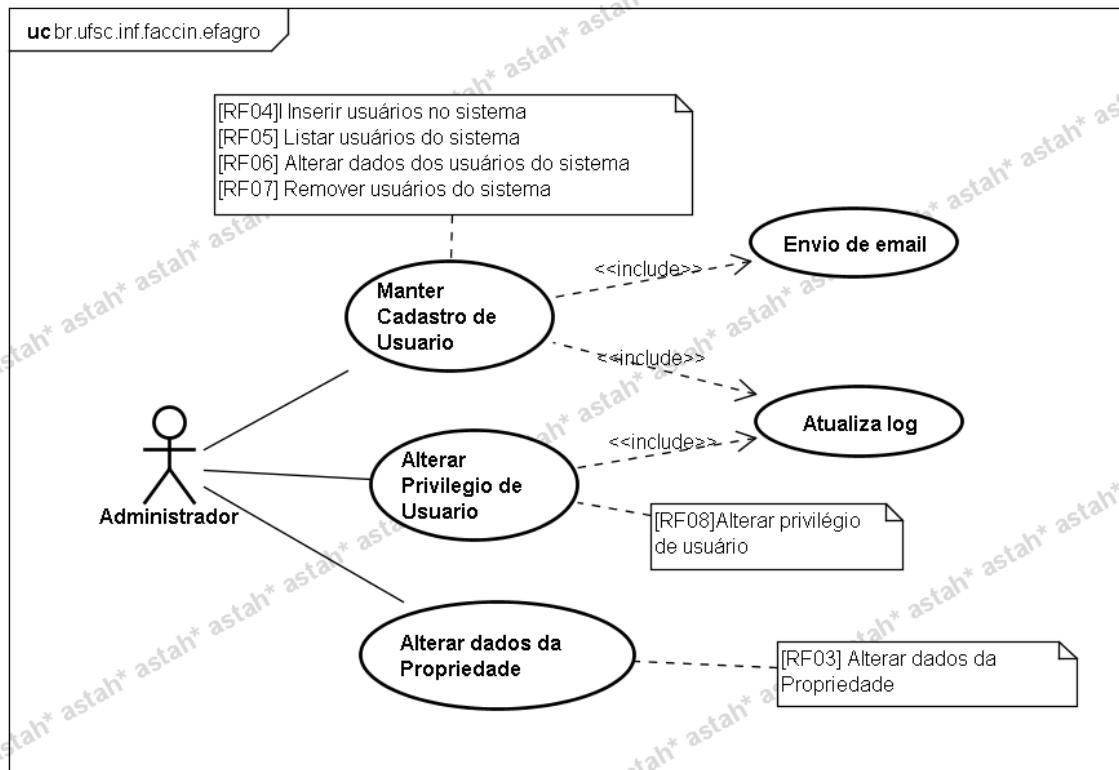


Diagrama 02 – Diagrama de Caso de Uso – Manter Usuários e propriedade. Fonte: Elaboração do autor, 2010.

### Caso de Uso: Inserir usuário

Ator: Administrador

Objetivo: Incluir um novo usuário na propriedade.

Pré-Condições: O ator deve estar autenticado no sistema.

Pós-Condições: Novo usuário cadastrado.

Fluxo principal:

1. O ator preenche os dados nome e email do usuário.
2. O sistema valida os dados.
3. O sistema insere o novo usuário. Quando inserido o usuário é automaticamente criado como ativo.
4. O sistema envia um email para o novo usuário cadastrado contendo uma senha gerada aleatoriamente.

5. A transação é gravada no log.

Fluxo alternativo:

1. O ator informa dados inconsistentes.
2. O sistema emite um aviso que os campos estão preenchidos de maneira incorreta.

### **Caso de Uso: Alterar usuário**

Ator: Administrador

Objetivo: Alterar os dados de um usuário da propriedade.

Pré-Condições: O ator deve estar autenticado no sistema.

Pós-Condições: Cadastro de usuário alterado.

Fluxo principal:

1. O sistema lista os usuários cadastrados.
2. O ator seleciona um usuário.
3. O sistema exibe os dados do usuário selecionado para o ator.
4. O ator pode alterar os seguintes dados do usuário: nome, e-mail e se o usuário está ou não ativo.
5. O sistema salva as alterações.
6. A transação é gravada no log.

Fluxo alternativo:

1. O ator informa dados inconsistentes.
2. O sistema emite um aviso que os campos estão preenchidos de maneira incorreta.

Fluxo alternativo 2:

1. O ator tenta desativar seu próprio usuário.
2. O sistema emite um aviso de que esta operação não é possível.

### **Caso de Uso: Excluir usuário**

Ator: Administrador

Objetivo: Remover um usuário da propriedade.

Pré-Condições: O ator deve estar autenticado no sistema.

Pós-Condições: Usuário removido do sistema.

Fluxo principal:

1. O sistema lista os usuários cadastrados.
2. O ator seleciona um usuário.
3. O sistema exibe os dados do usuário selecionado para o ator.
4. O ator confirma a exclusão do usuário.
5. O sistema exclui a conta do usuário.
6. A transação é gravada no log.

Fluxo alternativo:

1. O ator tenta excluir seu próprio usuário.
2. O sistema emite um aviso de que esta operação não pode ser efetuada.

### **Caso de Uso: Alterar privilégio de usuário**

Ator: Administrador

Objetivo: Alterar o privilégio de um usuário.

Pré-condições: O ator deve estar autenticado no sistema e ter privilégio de administrador.

Pós-condições: Usuário com privilégio alterado.

Fluxo principal:

1. O sistema lista os usuários cadastrados.
2. O ator seleciona um usuário.
3. O sistema exibe os dados do usuário selecionado para o ator.
4. O ator altera a opção de “colaborador” para “administrador” ou vice-versa.
5. O sistema grava as alterações.
6. A transação é gravada no log.

Fluxo alternativo:

1. O sistema verifica que existe apenas um usuário com os privilégios de administrador.

2. O sistema avisa o ator que deve existir no mínimo um usuário com privilégio de administrador cadastrado na propriedade.

### **Caso de Uso: Alterar dados da propriedade**

Ator: Administrador

Objetivo: Alterar os dados da propriedade.

Pré-condições: O ator deve estar autenticado no sistema e ter privilégio de administrador.

Pós-condições: propriedade com os dados atualizados.

Fluxo principal:

1. O sistema exibe os seguintes dados da propriedade para o ator: nome da propriedade, endereço, email, área total, área cultivável, área de preservação e outras áreas.
2. O ator altera os dados.
3. O sistema grava as alterações.

Fluxo alternativo:

1. O ator informa dados inconsistentes.
2. O sistema emite um aviso que os campos estão preenchidos de maneira incorreta.

## Funcionalidades Gerais do Sistema

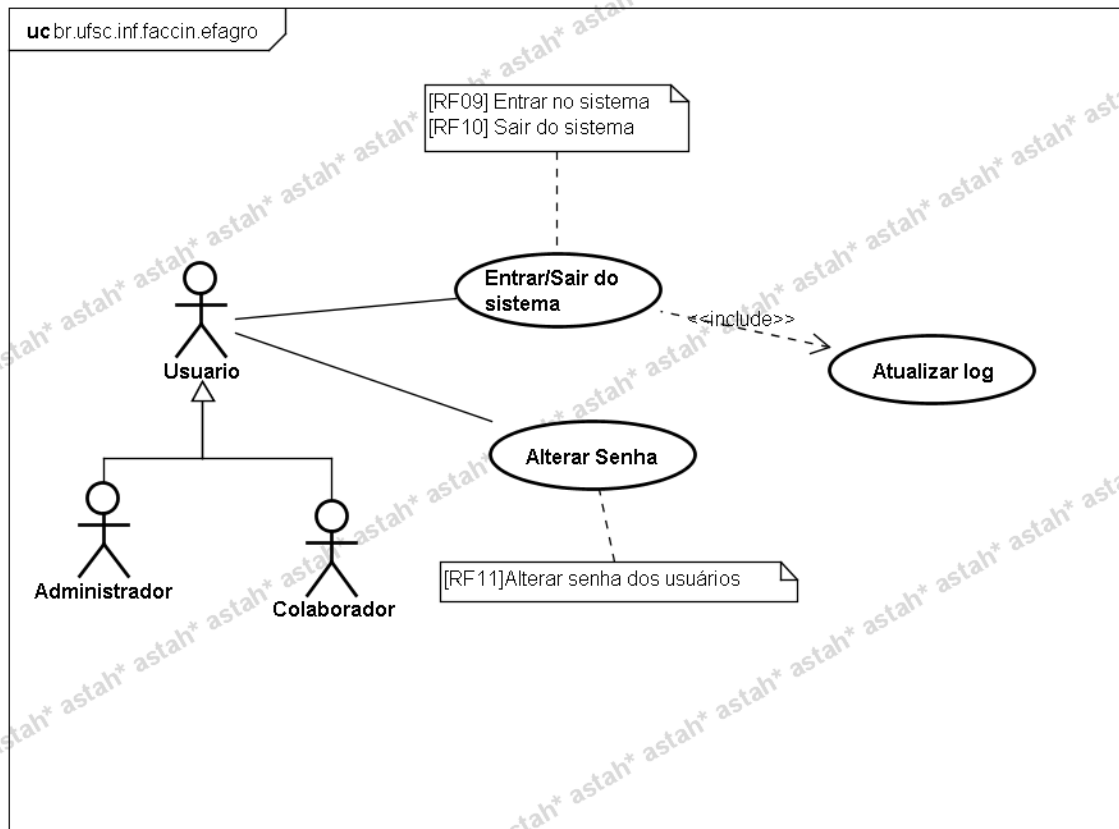


Diagrama 03 – Diagrama de Caso de Uso – Funcionalidades gerais. Fonte: Elaboração do autor, 2010.

### Caso de Uso: Entrar no sistema

Ator: Administrador, Colaborador

Objetivo: Efetuar *login* no sistema.

Pré-condições: O ator deve estar cadastrado no sistema.

Pós-condições: Ator autenticado no sistema.

Fluxo principal:

1. O ator preenche os campos de e-mail e senha.
2. O sistema verifica se os campos foram preenchidos corretamente.
3. O ator é autenticado no sistema.
4. O sistema exibe as opções conforme o privilégio do usuário (Administrador ou Colaborador)

Fluxo alternativo:



1. O ator informa dados inconsistentes.
2. O sistema emite um aviso que os campos estão preenchidos de maneira incorreta.

### **Caso de Uso: Sair do sistema**

Ator: Administrador, Colaborador

Objetivo: Efetuar *logout* do sistema.

Pré-condições: O ator deve estar autenticado no sistema.

Pós-condições: A sessão do ator deve ser finalizada.

Fluxo principal:

1. O ator clica na opção “Sair”.
2. O sistema encerra a sessão do usuário.

### **Caso de Uso: Alterar senha**

Ator: Administrador, Colaborador

Objetivo: Trocar a senha do usuário

Pré-condições: O ator deve estar autenticado no sistema.

Pós-condições: A senha do usuário deve ser alterada.

Fluxo principal:

1. O ator digita sua senha atual.
2. O ator digita a nova senha.
3. O ator redigita a nova senha
4. O sistema verifica se a senha atual está correta.
5. O sistema verifica se a nova senha, digitada no passo 2 e no passo 3 são iguais.
6. O sistema altera a senha do usuário.

Fluxo alternativo:

1. Caso o ator digite a senha atual incorreta, o sistema deve emitir a mensagem “Senha atual incorreta”.

Fluxo alternativo 2:

1. Caso a nova senha, digitada no passo 2 e no passo 3 sejam diferentes o sistema deve exibir a mensagem “Senhas não conferem”.

## Gerenciar Maquinário

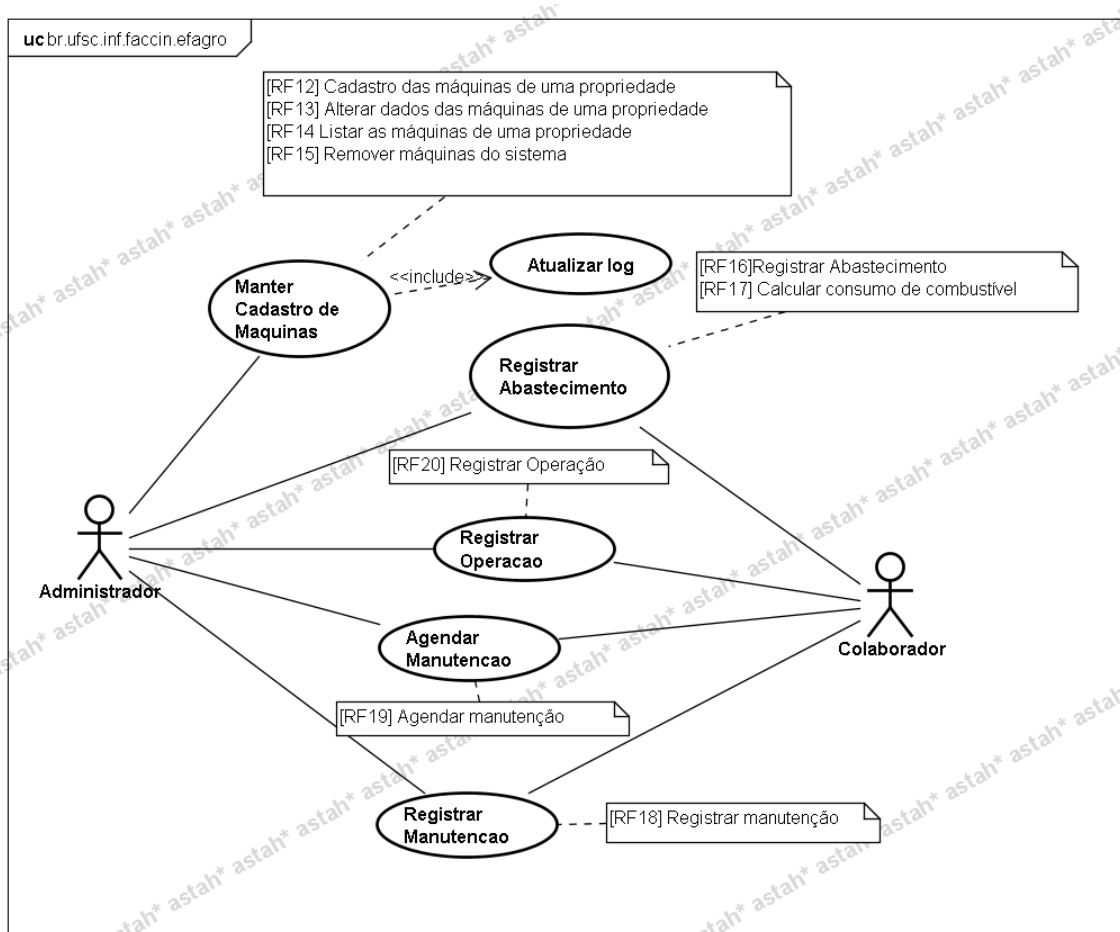


Diagrama 04 – Diagrama de Caso de Uso – Gerenciar Maquinário. Fonte: Elaboração do autor, 2010.

### Caso de Uso: Cadastrar máquina

Ator: Administrador

Objetivo: Inserir uma nova máquina no sistema.

Pré-condições: O ator deve estar autenticado no sistema.

Pós-condições: A máquina deve ser cadastrada no sistema.

Fluxo principal:

1. O ator preenche os seguintes dados da máquina: ano de fabricação, data da aquisição, valor da aquisição marca, modelo, numero de série e observação.
2. O sistema valida os dados digitados pelo ator.
3. É inserida a nova máquina no sistema.
4. A transação é gravada no log.

Fluxo alternativo:

1. O ator deixa de preencher algum dos seguintes dados: ano de fabricação, data da aquisição, valor da aquisição marca e modelo.
2. O sistema avisa que os campos são de preenchimento obrigatório.

### **Caso de Uso: Alterar o cadastro de uma máquina**

Ator: Administrador

Objetivo: Alterar uma máquina cadastrada no sistema.

Pré-condições: O ator deve estar autenticado no sistema e a máquina cadastrada.

Pós-condições: A máquina deve ser alterada no sistema.

Fluxo principal:

1. O sistema lista todas as máquinas da propriedade.
2. O ator seleciona uma máquina da lista.
3. O sistema exibe os seguintes dados da máquina: ano de fabricação, data da aquisição, valor da aquisição marca, modelo, numero de série e observação.
4. O ator altera os dados da máquina
5. O sistema valida os dados digitados pelo ator.
6. A máquina é alterada no sistema.
7. A transação é gravada no log.

Fluxo alternativo:

1. O ator deixa de preencher algum dos seguintes dados: ano de fabricação, data da aquisição, valor da aquisição marca e modelo.
2. O sistema avisa que os campos são de preenchimento obrigatório.

### **Caso de Uso: Remover uma máquina do cadastro**

Ator: Administrador

Objetivo: Remover uma máquina cadastrada no sistema.

Pré-condições: O ator deve estar autenticado no sistema e a máquina cadastrada.

Pós-condições: A máquina deve ser excluída do sistema.

Fluxo principal:

1. O sistema lista todas as máquinas da propriedade.
2. O ator seleciona uma máquina da lista.
3. O sistema exibe os seguintes dados da máquina: ano de fabricação, data da aquisição, valor da aquisição marca, modelo, numero de série e observação.
4. O ator confirma a exclusão da máquina do cadastro.
5. A máquina é removida do sistema.
6. A transação é gravada no log.

Fluxo alternativo:

1. O ator não confirma a exclusão da máquina.
2. O sistema volta a exibir a lista de máquinas.

### **Caso de Uso: Registrar abastecimento**

Ator: Administrador e Colaborador

Objetivo: Registrar o abastecimento de uma máquina

Pré-condições: O ator deve estar autenticado no sistema e a máquina cadastrada.

Pós-condições: Deve ser registrada a quantidade de combustível e a máquina que foi abastecida.

Fluxo principal:

1. O sistema lista todas as máquinas da propriedade.
2. O ator seleciona uma máquina da lista.

3. O ator preenche o tipo do combustível, a quantidade e o valor unitário do combustível.
4. (Passo opcional) O ator preenche no sistema o valor marcado pelo do hodômetro ou horímetro da máquina.
5. O sistema registra o abastecimento e exibe, caso tenha sido executado o passo 4, o consumo médio de combustível.

### **Caso de Uso: Agendar manutenção**

Ator: Administrador e Colaborador

Objetivo: Agendar a manutenção de uma máquina.

Pré-condições: O ator deve estar autenticado no sistema e a máquina cadastrada.

Pós-condições: Após o agendamento da manutenção o sistema deverá enviar um email para o usuário na data de aviso da manutenção.

Fluxo principal:

1. O sistema lista todas as máquinas da propriedade.
2. O ator seleciona uma máquina da lista.
3. O ator preenche os seguintes dados: máquina, data da manutenção, frequência, lembrete e quanto tempo antes da manutenção o usuário gostaria de ser lembrado da manutenção.
4. O sistema agenda a manutenção.

Fluxo alternativo:

1. O ator não preenche um dos dados: máquina, data da manutenção, frequência, lembrete e quanto tempo antes da manutenção o usuário gostaria de ser lembrado da manutenção.
2. O sistema avisa que os campos são de preenchimento obrigatório.

### **Caso de Uso: Registrar manutenção**

Ator: Administrador e Colaborador

Objetivo: Registrar a manutenção de uma máquina.

Pré-condições: O ator deve estar autenticado no sistema e a máquina cadastrada.

Pós-condições: Deve ser registrada a manutenção no sistema.

Fluxo principal:

1. O sistema lista todas as máquinas da propriedade.
2. O ator seleciona uma máquina da lista.
3. O ator preenche os seguintes dados: data da manutenção, descrição da manutenção e custo total da manutenção.
4. O sistema registra a manutenção.

Fluxo alternativo:

1. O ator não preenche um dos dados: data da manutenção, descrição e valor.
2. O sistema avisa que os campos são de preenchimento obrigatório.

### **Caso de Uso: Registrar operação**

Ator: Administrador e Colaborador

Objetivo: Registrar a operação de uma máquina, isto é, um trabalho realizado por ela.

Pré-condições: O ator deve estar autenticado no sistema e a máquina cadastrada.

Pós-condições: Deve ser registrada a operação no sistema.

Fluxo principal:

1. O sistema lista todas as máquinas da propriedade.
2. O ator seleciona uma máquina da lista.
3. O ator preenche os seguintes dados: data da operação, o tempo da operação e uma descrição da operação.
4. O sistema registra a operação.

Fluxo alternativo:

1. O ator não preenche um dos dados: data da operação e o tempo da operação.
2. O sistema avisa que os campos são de preenchimento obrigatório.

## Gerenciar Agricultura

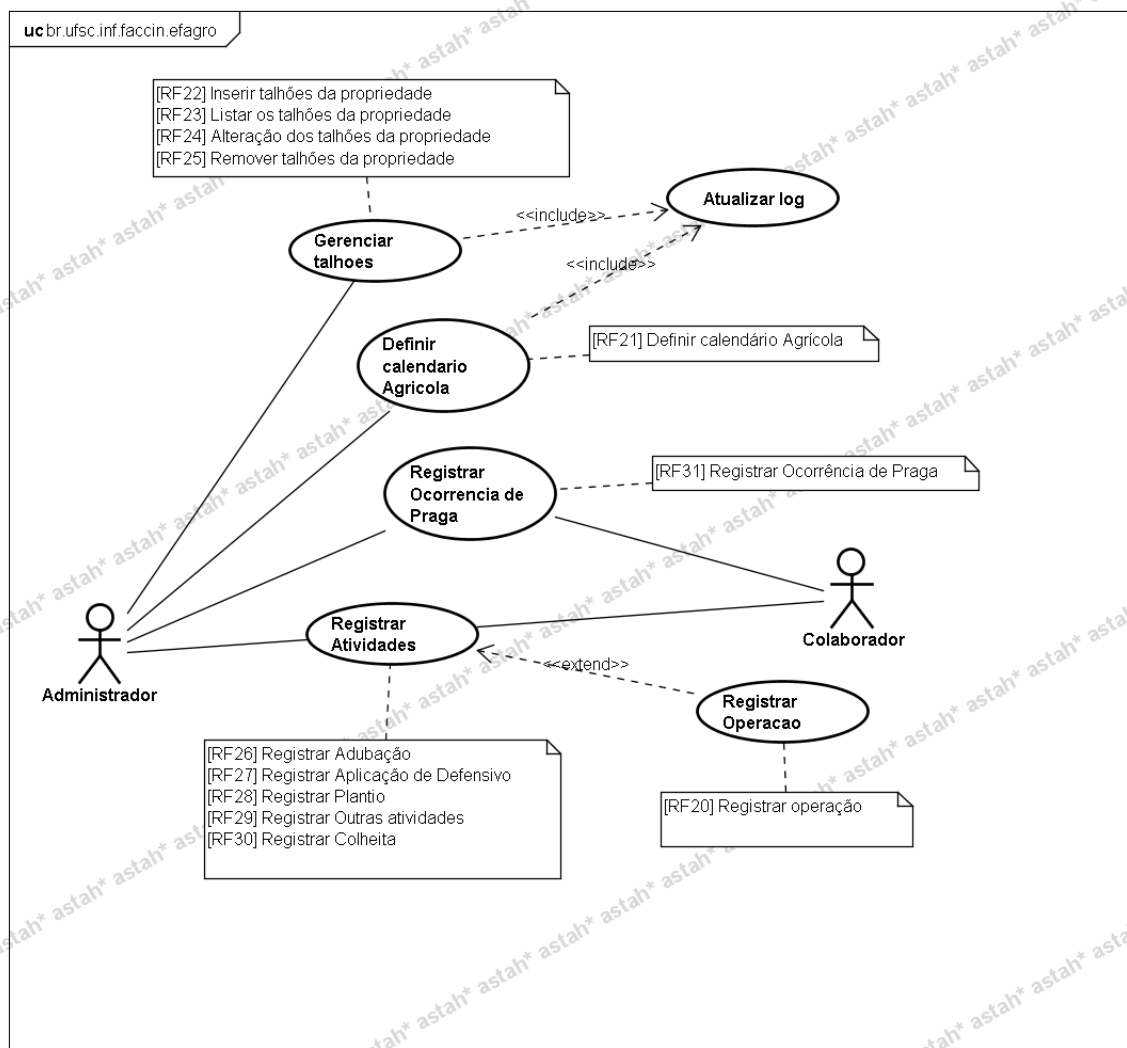


Diagrama 05 – Diagrama de Caso de Uso – Gerenciar Agricultura. Fonte: Elaboração do autor, 2010.

### **Caso de Uso: Definir Calendário Agrícola**

Ator: Administrador

Objetivo: Definir o Ano/Safra da propriedade.

Pré-condições: O ator deve estar autenticado no sistema.

Pós-condições: Definido o período do Ano/Safra atual.

Fluxo principal:

1. O ator preenche a data inicial, a data final do Ano/Safra e o ano/safra no formato AnoInicial/AnoFinal. Ex: "2009/2010".
2. O sistema insere um novo ano safra no sistema.
3. A partir do momento que foi definido o ano/safra, os talhões inseridos no período que entre a data inicial e a data final do Ano/Safra, são associados a ele.
4. A transação é gravada no log.

### **Caso de Uso: Inserir talhão**

Ator: Administrador

Objetivo: Inserir novo talhão no ano/safra

Pré-condições: O ator deve estar autenticado no sistema e o ano/safra definido.

Pós-condições: Deve ser incluído o novo talhão.

Fluxo principal:

1. O ator preenche os seguintes dados do talhão: nome da área, observação, área, localização, ativo, terreno arrendado.
2. O sistema inclui o novo talhão, inserido a data de inclusão e o Ano/Safra do período.
3. A transação é gravada no log.

Fluxo alternativo:

1. O ator deixa de preencher um dos seguintes dados do talhão: nome da área, área, localização, ativo, terreno arrendado.
2. O sistema avisa o ator que os campos são de preenchimento obrigatório.



### **Caso de Uso: Alterar talhão**

Ator: Administrador

Objetivo: Alterar os dados de um talhão.

Pré-condições: O ator deve estar autenticado no sistema e o talhão cadastrado.

Pós-condições: Os novos dados do talhão devem ser gravados.

Fluxo principal:

1. O sistema lista todos os talhões da propriedade do Ano/Safra atual.
2. O ator seleciona um talhão.
3. O sistema exibe seguintes dados do talhão: nome da área, observação, área, localização, ativo, terreno arrendado.
4. O ator altera os dados.
5. O sistema salva as alterações feitas pelo ator
6. A transação é gravada no log.

Fluxo alternativo:

1. O ator deixa de preencher um dos seguintes dados do talhão: nome da área, área, localização, ativo, terreno arrendado.
2. O sistema avisa o ator que os campos são de preenchimento obrigatório.

### **Caso de Uso: Remover talhão**

Ator: Administrador

Objetivo: Remover um talhão.

Pré-condições: O ator deve estar autenticado no sistema e o talhão cadastrado.

Pós-condições: O talhão deve ser removido.

Fluxo principal:

1. O sistema lista todos os talhões da propriedade do Ano/Safra atual.
2. O ator seleciona um talhão.
3. O ator confirma a exclusão do talhão.
4. A transação é gravada no log.

Fluxo alternativo:

1. O ator não confirma a exclusão do talhão.

2. O sistema volta e exibir a lista dos talhões da propriedade do Ano/Safra atual.

### **Caso de Uso: Registrar Ocorrência de praga**

Ator: Administrador, Colaborador

Objetivo: Registrar a ocorrência de uma praga em determinado talhão.

Pré-condições: O ator deve estar autenticado no sistema e o talhão cadastrado.

Pós-condições: A ocorrência da praga deve ser registrada no sistema e associada ao talhão.

Fluxo principal:

1. O sistema lista todos os talhões da propriedade do Ano/Safra atual.
2. O ator seleciona um talhão.
3. O ator clica na opção “Praga”
4. O ator digita o nome da praga, a data da ocorrência, uma breve descrição e descreve a solução adotada, quando houver.
5. O sistema registra a ocorrência da praga no talhão.

Fluxo alternativo:

1. O ator não digita o nome da praga e a data da ocorrência.
2. O sistema avisa o ator que os campos são de preenchimento obrigatório.

### **Caso de Uso: Registrar adubação**

Ator: Administrador, Colaborador

Objetivo: Registrar a aplicação de um fertilizante em um determinado talhão.

Pré-condições: O ator deve estar autenticado no sistema e o talhão cadastrado.

Pós-condições: Deve ser registrada no sistema a aplicação de fertilizante no talhão.

Fluxo principal:

1. O sistema lista todos os talhões da propriedade do Ano/Safra atual.
2. O ator seleciona um talhão.
3. O ator clica na opção “Adubação”.

4. O ator preenche os seguintes dados: data da adubação, quantidade de fertilizante, valor unitário, fertilizante e eventuais observações.
5. (Opcional) O ator registra uma operação. (Caso de Uso Registrar Operação)
6. O sistema registra a ocorrência da praga no talhão.

Fluxo alternativo:

1. O ator não preenche um dos seguintes campos: data da adubação, quantidade de fertilizante, valor unitário e fertilizante.
2. O sistema avisa o ator que os campos são de preenchimento obrigatório.

### **Caso de Uso: Registrar aplicação de defensivo**

Ator: Administrador, Colaborador

Objetivo: Registrar a aplicação de um defensivo em um determinado talhão.

Pré-condições: O ator deve estar autenticado no sistema e o talhão cadastrado.

Pós-condições: Deve ser registrada no sistema a aplicação do defensivo no talhão.

Fluxo principal:

1. O sistema lista todos os talhões da propriedade do Ano/Safra atual.
2. O ator seleciona um talhão.
3. O ator clica na opção “Aplicação de Defensivo”.
4. O ator preenche os seguintes dados: data da aplicação, quantidade aplicada, valor unitário, defensivo, temperatura ambiente, umidade relativa do ar, dias de carência e eventuais observações.
5. (Opcional) O ator registra uma operação. (Caso de Uso Registrar Operação).
6. O sistema registra a aplicação de um defensivo no talhão.

Fluxo alternativo:

1. O ator não preenche um dos seguintes campos: data da aplicação, quantidade aplicada, valor unitário, defensivo, máquinas utilizadas na

aplicação, horas de máquinas gastas, temperatura ambiente, umidade relativa do ar e dias de carência.

2. O sistema avisa o ator que os campos são de preenchimento obrigatório.

### **Caso de Uso: Registrar plantio**

Ator: Administrador, Colaborador

Objetivo: Registrar o plantio em um determinado talhão.

Pré-condições: O ator deve estar autenticado no sistema e o talhão cadastrado.

Pós-condições: O plantio deve ser registrado no sistema.

Fluxo principal:

1. O sistema lista todos os talhões da propriedade do Ano/Safra atual.
2. O ator seleciona um talhão.
3. O ator clica na opção "Plantio".
4. O ator preenche os seguintes dados: data do plantio, data prevista da colheita, variedade, quantidade, valor total e eventuais observações.
5. (Opcional) O ator registra uma operação. (Caso de Uso Registrar Operação).
6. (Opcional) O ator registra a aplicação de defensivo associado ao plantio.
7. (Opcional) O ator registra a adubação no plantio.
8. O sistema registra a aplicação de um defensivo no talhão.

Fluxo alternativo:

1. O ator não preenche um dos seguintes campos: data do plantio, data prevista da colheita, variedade, quantidade, valor total
2. O sistema avisa o ator que os campos são de preenchimento obrigatório.

Fluxo alternativo 2:

1. Quando executado o passo 3, caso já haja um plantio registrado para aquele talhão, o sistema informa o ator que já existe um plantio registrado para aquele talhão.

### **Caso de Uso: Registrar colheita**

Ator: Administrador, Colaborador

Objetivo: Registrar a colheita de um determinado talhão.

Pré-condições: O ator deve estar autenticado no sistema e o talhão cadastrado com um plantio registrado.

Pós-condições: Deve ser registrada no sistema a colheita de um determinado talhão.

Fluxo principal:

1. O sistema lista todos os talhões da propriedade do Ano/Safra atual.
2. O ator seleciona um talhão.
3. O ator clica na opção “Colheita”.
4. O ator preenche os seguintes dados: data da colheita, quantidade colhida, valor total e eventuais observações.
5. (Opcional) O ator registra uma operação. (Caso de Uso Registrar Operação)
6. O sistema registra a ocorrência da praga no talhão.

Fluxo alternativo:

1. O ator não preenche um dos seguintes campos: data da colheita, quantidade colhida e valor total.
2. O sistema avisa o ator que os campos são de preenchimento obrigatório.

### **Caso de Uso: Registrar outras atividades**

Ator: Administrador, Colaborador

Objetivo: Registrar uma atividade no talhão, que não seja adubação, colheita, plantio e aplicação de defensivo.

Pré-condições: O ator deve estar autenticado no sistema e o talhão cadastrado.

Pós-condições: A atividade deve ser registrada no sistema

Fluxo principal:

1. O sistema lista todos os talhões da propriedade do Ano/Safra atual.
2. O ator seleciona um talhão.
3. O ator clica na opção “Outras Atividades”.
4. O ator preenche os seguintes dados: data da atividade e descrição da atividade.
5. (Opcional) O ator registra uma operação. (Caso de Uso Registrar Operação).
6. O sistema registra a atividade no talhão.

Fluxo alternativo:

1. O ator não preenche um dos seguintes campos: data da atividade e descrição da atividade.
2. O sistema avisa o ator que os campos são de preenchimento obrigatório.

## ANEXO III - OPENUP/BASIC

### Introdução

O OpenUp/Basic<sup>22</sup> é um processo de desenvolvimento de software de código aberto. Ele é projetado para pequenas equipes de desenvolvimento que pretendem ter uma abordagem ágil no desenvolvimento. Segundo a definição do projeto, o OpenUp é um processo mínimo, completo e extensível, isto é, contém um processo com o mínimo necessário para as equipes ou então pode ser estendido e adaptado para atender propósitos específicos de um projeto ou de um grupo de trabalho.

Existe uma lista de trabalho, onde as tarefas são relacionadas, atribuídas e acompanhadas. Esta lista serve para que os membros da equipe registrar em um único lugar todas as tarefas executadas, incluído solicitações de mudança, pedidos de usuários e erros.

### Disciplinas

O OpenUp/Basic possui seis disciplinas, que fornecem a organização e as tarefas a serem executadas o ciclo de vida do projeto. São elas:

- Análise e *design* tem o propósito de:
  - Transformar os requisitos em um projeto.
  - Desenvolver uma arquitetura robusta para o sistema.
  - Adaptar o projeto para ser correspondente com o ambiente de desenvolvimento.
- Gerência de configuração tem o propósito de:

---

<sup>22</sup> <http://epf.eclipse.org/wikis/openuppt/>. Acessado em 10/12/2009.

- Manter os produtos de trabalho.
- Fornecer meios consistentes para adaptar se as mudanças.
- Fornecer dados para a medição do progresso.
- Implementação que tem o propósito de:
  - Construir o sistema de forma incremental.
  - Verificar se os artefatos construídos funcionem conforme o especificado.
- Gerência de projeto que tem o propósito de:
  - Priorizar a seqüência de trabalho.
  - Manter o foco da equipe.
  - Manter a equipe informada do progresso do projeto
  - Fornecer a estrutura de controle de riscos.
- Requisitos que tem o propósito de:
  - Compreender o problema a ser resolvido.
  - Entender o que os usuários desejam.
  - Definir os requisitos para a solução
  - Definir o escopo do sistema.
  - Identificar as restrições técnicas e as interfaces externas do sistema
  - Estimar um cronograma.
- Testes que tem o propósito de:
  - Encontrar e documentar os defeitos.
  - Validar o software de acordo com o projeto
  - Validar a implementação correta dos requisitos



## Ciclo de vida

O ciclo de vida de um projeto no OpenUp/Basic, é composto das seguintes etapas: concepção, elaboração, construção e transição. Na fase de concepção o objetivo é estabelecer uma visão geral do projeto. Na elaboração, são levantados e detalhados os requisitos. Na etapa de construção o software é construído e testado. Finalmente na fase de transição se realiza os ajustes finos na qualidade global do produto.

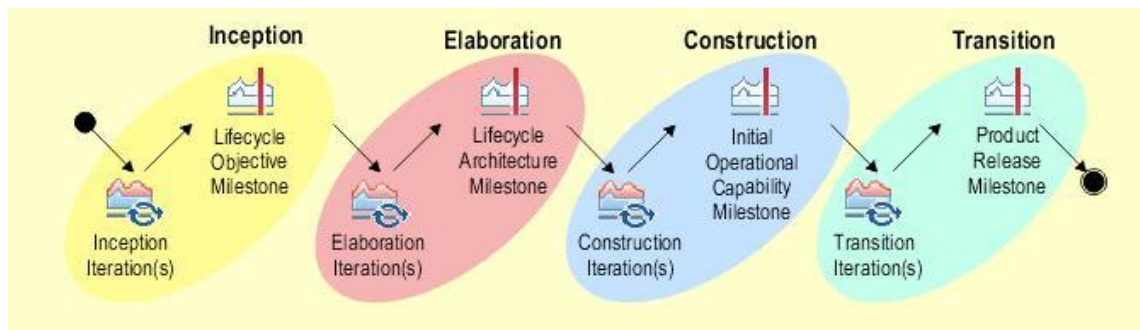


Diagrama 01 - O ciclo de vida do OpenUP. Fonte:

[http://epf.eclipse.org/wikis/openuppt/openup\\_basic/guidances/roadmaps/openup\\_basic\\_roadmap\\_vEruwN-rEdqiM\\_wFaqLjNg.html](http://epf.eclipse.org/wikis/openuppt/openup_basic/guidances/roadmaps/openup_basic_roadmap_vEruwN-rEdqiM_wFaqLjNg.html). Acessado em 15/11/2009.

## **ANEXO IV - ARTIGO**

# Sistema de Informação para Agricultura Familiar

Cleberon Charles Colombo Faccin

Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC - Brasil

[faccin@inf.ufsc.br](mailto:faccin@inf.ufsc.br)

***Abstract.** The family agriculture is composed of small and medium rural producers who are responsible for producing much of the food consumed in Brazil. These properties have a very low level of computerization or nonexistent. There are today in the Brazilian market some software available for the administration of rural properties. Analyzing the features of this software and the need of farmers emerged the proposal of the E-fagro, which is a free software project that was developed to help the family farmers in managing their farms.*

***Resumo.** A agricultura familiar é composta por pequenos e médios produtores rurais que são responsáveis por produzir boa parte dos alimentos consumidos no Brasil. Estas propriedades possuem um grau de informatização muito baixo ou inexistente. Existem hoje no mercado brasileiro alguns softwares disponíveis para a administração de propriedades rurais. Analisando as funcionalidades destes softwares e a necessidade dos agricultores familiares surgiu a proposta do E-fagro que é um projeto de software livre que foi desenvolvido para auxiliar os agricultores familiares na administração de suas propriedades.*

## 1. Introdução

A agricultura familiar, constituída por pequenos e médios produtores, representa a maioria dos produtores rurais brasileiros. Em geral, são pessoas com baixo nível de escolaridade, que diversificam os produtos cultivados para diluir custos, aumentar a renda e aproveitar as oportunidades de oferta ambiental e disponibilidade de mão-de-obra. Este segmento tem um papel crucial na economia das pequenas cidades (PORTUGAL, 2004).

O governo federal, com apoio de estados e municípios, mantém um órgão oficial para desenvolver projetos de apoio à infraestrutura, assistência técnica, pesquisa agropecuária, capacitação e linhas de crédito diferenciadas. Trata-se da Secretaria da Agricultura Familiar do Ministério do Desenvolvimento Agrário (SAF/MDA). Entre os principais projetos desenvolvidos pela Secretaria está o Programa Nacional de Fortalecimento da Agricultura Familiar (Pronaf) e o Plano Safra da agricultura familiar.

Para Pereira (2009), a aplicação da tecnologia da informação vem crescendo consideravelmente em todos os setores, permitindo o avanço científico em várias áreas do conhecimento. Os recursos tecnológicos facilitam cada vez mais o dia-a-dia das pessoas, incorporando novos hábitos e construindo novas relações sociais. No setor agropecuário, entretanto, a presença da tecnologia da informação ainda é tímida. Estudos indicam que dos 3,7 milhões de propriedades rurais no País, apenas entre 4 e 8% delas são informatizadas.

Sendo assim, a proposta deste trabalho foi desenvolver um software que auxilie na administração das propriedades rurais, facilitando o trabalho do administrador na coleta e organização das informações.

## 2. Fundamentação Técnica

Segundo a *The Food and Agriculture Organization of the United Nations* (FAO, 2009), mais de um bilhão de pessoas no mundo passam fome, ou seja, uma a cada seis pessoas do globo não tem as necessidades básicas alimentares satisfeitas. Conforme a tabela 1, os principais produtos agrícolas tiveram uma grande redução no estoque nos últimos 10 anos. Esta redução nos estoques deu-se pelo fato de que a produção não tem acompanhado a demanda de consumo.

**Tabela - 01 – Evolução dos estoques mundiais de alimento em milhões de toneladas.**

Evolução dos estoques mundiais de alimento			
Safra	Arroz	Milho	Trigo
1999/2000	143,1	193,7	209,5
2008/2009	82,6	99	124

Fonte: USDA, 2009

Segundo a Secretaria Nacional da Agricultura Familiar, a produção dos agricultores familiares supre 70% do consumo interno do país, além de representar 32% do PIB agropecuário. Vale ressaltar que a produção agrícola patronal responde por metade da riqueza gerada pela agricultura no Brasil. Isso se deve ao fato que a produção de soja, café, cana-de-açúcar e seus derivados, são principais produtos agrícolas de exportação.

Observando o mercado de software brasileiro voltado para agronegócio pode-se dizer que não existem muitas opções, principalmente para a agricultura familiar. Os aplicativos comerciais hoje são voltados quase que exclusivamente para grandes e médias propriedades rurais.

Segundo o Cetic (2008), apenas 4% dos domicílios rurais tem acesso a internet. Proporcionalmente, é um número pequeno, mas quando tratamos de valores absolutos, estes 4%, segundo o IBGE (2008), são de aproximadamente 330 mil domicílios, representando um mercado em potencial bastante considerável. Deve-se levar em conta ainda que as perspectivas deste número crescer nos próximos anos é muito grande, principalmente devido à expansão de tecnologias como a cobertura de redes de celular de terceira geração, redes sem fio e a transmissão de sinal pela rede elétrica. Por estes motivos foi optado por se desenvolver um *software web*.

## 3. Estado da Arte

Foram realizadas pesquisas por *software* com funcionalidades semelhantes ao sistema proposto em ferramentas de busca, fóruns e indicações de profissionais da área. Foram identificados os seguintes *softwares*: ADM Rural 4.0, Prorural, Agriness S2 3.0 e o Open Farm.

O Prorural apresenta dois módulos, um para o controle agrícola e outro para o controle pecuário. Utiliza o banco de dados Firebird, que é instalado no cliente durante a instalação do

software. É desenvolvido pela Brazsoft Tecnologia em Agro Business, empresa sediada em Cuiabá, Mato Grosso, que desenvolve o software desde 2003. Possui diversos clientes espalhados por praticamente todo o Brasil. O investimento do produtor rural para adquirir o software é de aproximadamente R\$ 400,00, segundo informação do site da empresa de junho de 2009.

O Agriness S2 3.0 é um software voltado para gerência da produção de suínos. Desenvolvido pela AGRINESS Sistemas e Tecnologias de Informação Ltda, empresa sediada em Florianópolis. Possui hoje, segundo informações do site da empresa, mais de mil unidades produtivas utilizando seus produtos, totalizando meio milhão de matrizes suínas gerenciadas (AGRINESS, 2009). Está disponível nas versões S2 Comercial, CIA, Quarto Sítio e S2 Multiplicadora. As funcionalidades listadas a seguir são da versão S2 Comercial por tratar da produção de terminação de suínos, unidade de produção de leitões e produção de ciclo completo.

O ADM Rural 4.0, é um software para controle de custos de produção com a utilização de planos de contas gerenciais e centro de custos por atividade produtiva. Desenvolvido em Visual Basic, utilizando banco de dados Firebird, pela empresa Planejar Informática, o custo da licença é de R\$ 650,00.

O Open Farm é um projeto de software livre para administração de fazendas. Existem duas versões, uma para Windows, denominada 1.0, e outra para Linux, denominada 2.0. Segundo a documentação das versões, ambas apresentam as mesmas funcionalidades e características de interface. Sendo assim foi escolhida a versão 1.0, pelo fato da versão 2.0 ter apresentado problemas na instalação. A versão 1.0 foi escrita na linguagem Visual Basic utilizando banco de dados Access. Já na versão 2.0, o sistema foi, segundo o autor do projeto, traduzido para Python e o banco de dados passou a ser MySQL. Como é um projeto de software livre, é gratuito e está disponível para ser baixado, instalado e modificado sem qualquer custo.

#### **4. Sistema Proposto**

O sistema proposto foi desenvolvido de acordo com os requisitos levantados junto aos produtores rurais, com auxílio de profissionais atuantes no ramo da agricultura catarinense.

Um agricultor acessa o site e efetua o cadastro da sua propriedade, incluindo o maquinário disponível. Após realizar estes cadastros, a área cultivável da propriedade é fracionada em partes, denominada talhões, que abrigam por um período definido uma certa variedade de uma determinada cultura.

A medida que são realizadas ações nestes talhões (plantio, colheita, adubação aplicação de defensivos...) o agricultor efetua lançamentos destas ações no sistema, bem como as máquinas utilizadas e o tempo gasto.

De posse destas informações, quando o agricultor necessitar ele pode solicitar ao sistema relatórios de um determinado talhão, ou de toda a safra. A seguir são listadas as funcionalidades gerais do sistema:

- Cadastrar a propriedade rural, sua localização, proprietário e área total.
- Cadastrar os usuários da propriedade com atribuições de administrador ou colaborador.
- Controlar as cultivares perenes e anuais, incidência de pragas, investimentos e resultados.

- Manter histórico das atividades agrícolas.
- Controlar o uso do maquinário da propriedade, o consumo de combustível e a manutenção.
- Gerar relatórios de produtividade e lucratividade por talhão, cultivar ou safra.
- Efetuar comparativos de produção ao longo do tempo.

## 5. Desenvolvimento

Após a conclusão do levantamento de requisitos, foi realizada a modelagem relacional do sistema que deu origem ao um banco de dados relacional. Durante a modelagem do banco de dados foi preciso levar em conta que o sistema precisa manter várias propriedades cadastradas. Para isso todas as entidades e relacionamentos relativos aos dados da propriedade, devem estar vinculados a ela, direta ou indiretamente.

Para a implementação do E-fagro foi utilizado o framework JBoss Seam que é uma plataforma de desenvolvimento de código aberto para construções de aplicações web em Java. O Seam integra tecnologias como Asynchronous JavaScript and XML (AJAX), Java Server Faces (JSF), Java Persistence (JPA), Enterprise Java Beans (EJB 3.0) e Business Process Management (BPM). A versão utilizada neste trabalho foi a 2.1.2 datada de junho de 2009 (SEAM REFERENCE, 2009).

Uma das ferramentas do JBoss Seam é o seam-gen, que possibilita realizar a engenharia reversa a partir do banco de dados. Uma vez configurado o seam-gen, as tabelas do banco de dados são mapeadas para classes Java, utilizando anotações. O seam-gen ainda gera as classes de controle e uma interface utilizando Java Server Faces coerente com os relacionamentos mapeados no modelo de dados entidade relacionamento. Dessa forma, quando o projeto é criado todas as entidades mapeadas já possuem as funcionalidades de criação, leitura, remoção e atualização implementadas automaticamente.

### 5.1. Interface com Usuário

A interface com o usuário foi desenvolvida utilizando a tecnologia Java Server Faces com o auxílio das bibliotecas do RichFaces e do JBoss Seam Framework. Alguns *templates* gerados pelo seam-gen foram aproveitados no desenvolvimento, porém a maioria das telas geradas automaticamente pela ferramenta não foi aproveitada. A utilização de *templates* facilita a padronização das telas e melhora a produtividade do projeto. Uma vez definido um *template*, basta associar um determinado item da tela ao *template* e o item assume o comportamento e as características visuais do *template*.

A seguir pode-se observar uma tela da interface com usuário desenvolvida:

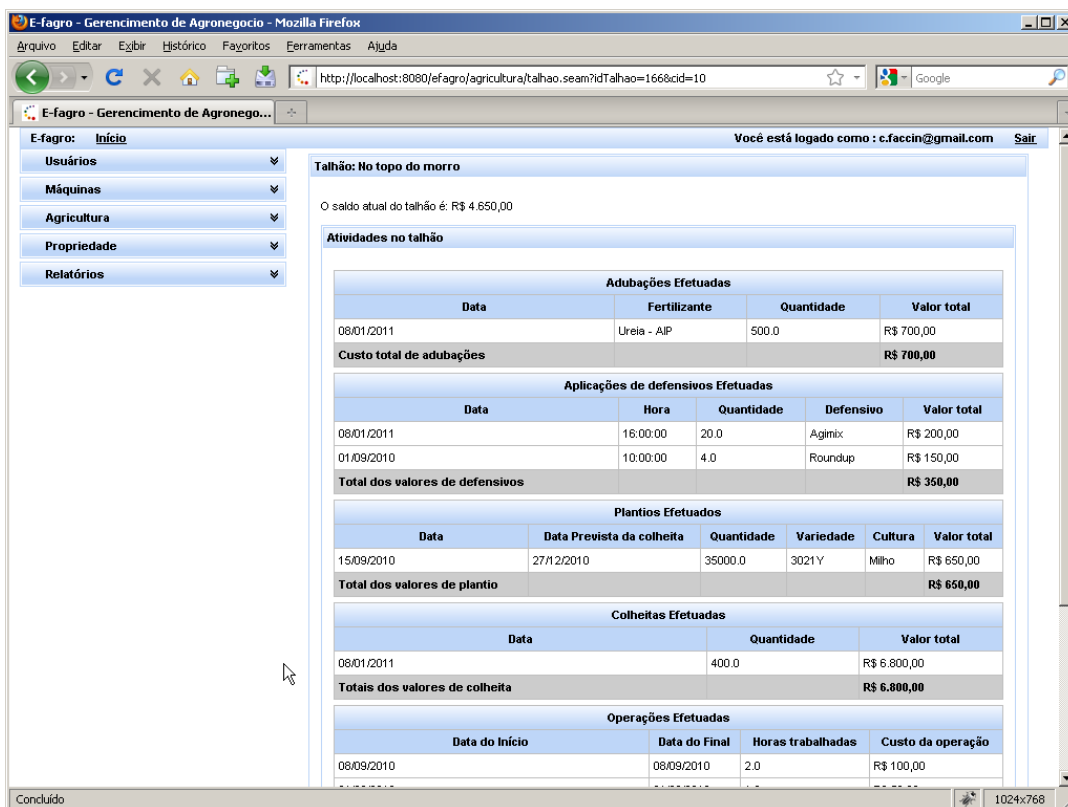


Figura 1. – Informações detalhadas de um talhão selecionado pelo usuário. Fonte: Elaboração do autor.

## 7. Conclusão

Após as pesquisas realizadas sobre os softwares disponíveis no mercado voltados para a gestão de propriedades rurais, pode-se dizer que o projeto E-fagro é pioneiro no setor por ser um *software* livre voltado para a agricultura familiar. Para ser colocado em produção ainda são necessários alguns ajustes, por exemplo, construir um manual de utilização. Em breve será disponibilizada uma versão preliminar, para avaliar o desempenho e ter um retorno dos usuários quanto a utilização e funcionalidades do *software*.

A utilização do JBoss Seam Framework foi essencial para o desenvolvimento do projeto. As ferramentas dele deram um grande impulso inicial no desenvolvimento do *software* diminuindo significativamente a tarefa de implementar o código para operações básicas como incluir, buscar, alterar e remover registros do banco de dados. A integração do JSF com a os componentes do Seam, o controle de contextos, principalmente o contexto de conversação, e a adoção de uma arquitetura simples, sem muitas camadas, possibilitaram ao desenvolvedor a rápida criação do projeto.

## References

- ADM RURAL 4.0. Versão demonstrativa. Disponível em <<http://www.admrural.com.br/>> Acesso em 16/10/2009.
- AGRINESS S2. Versão demonstrativa. Disponível em <<http://www.agriness.com.br/index.jsp>>. Acesso em 27/09/2009.

- BRASIL. Ministério da Agricultura, Pecuária e Abastecimento. Plano Agrícola e Pecuário 2009/2010. Documento Eletrônico. Disponível em <[http://www.agricultura.gov.br/images/MAPA/arquivos\\_portal/PAP2009\\_WEB.pdf](http://www.agricultura.gov.br/images/MAPA/arquivos_portal/PAP2009_WEB.pdf)>. Acesso em 23/06/2009
- CETIC. Proporção de domicílios com acesso à internet. Documento eletrônico disponível em <<http://www.cetic.br/usuarios/tic/2008-total-brasil/rel-geral-04.htm>>. Acesso em 12/6/2009
- DIEESE. Departamento Intersindical de Estatística e Estudos Sócio-Econômicos. Documento eletrônico disponível em <<http://www.dieese.org.br/anu/estatisticasMeioRural2008.pdf>>. Acesso em 22/06/2009.
- FAO. One sixth of humanity undernourished - more than ever before. Documento eletrônico disponível em <<http://www.fao.org/news/story/en/item/20568/icode/>>. Acesso em 19/06/2009.
- HEUSER, Carlos Alberto. Projeto de Banco de dados. Porto Alegre: Sagra Luzzato, 2004.
- IBGE. Instituto Brasileiro de Geografia e Estatísticas. Documento eletrônico disponível em <<http://www.ibge.gov.br>>. Acesso em 01/06/2009.
- JAVA SERVER FACES. Documento eletrônico disponível em <<http://java.sun.com/javaee/javaserverfaces/overview.html>>. Acessado em 22/01/2009.
- PEREIRA, Nadir Rodrigues. Embrapa Informática. Documento eletrônico disponível em <<http://www.cnptia.embrapa.br/content/evento-quer-popularizar-tecnologia-da-informa-o-no-campo.html>>. Acesso em 20/06/2009.
- PROJETO OPEN FARM. Documento eletrônico disponível em <<http://www.projetoopenfarm.cjb.net/>>. Acesso em 20/05/2009.
- PRORURAL. Versão demonstrativa. Disponível em <<http://brazsoft.com.br/>>. Acesso em 10/06/2009.
- PORTUGAL, Alberto Duque. O Desafio da Agricultura Familiar. Documento eletrônico disponível em <<http://www.embrapa.br/imprensa/artigos/2002/artigo.2004-12-07.2590963189/>> Acesso em 30/08/2008.
- SEAM REFERENCE. Documento eletrônico disponível em <[http://docs.jboss.com/seam/2.1.1.GA/reference/en-US/pdf/seam\\_reference.pdf](http://docs.jboss.com/seam/2.1.1.GA/reference/en-US/pdf/seam_reference.pdf)>. Acesso em 21/01/2009.
- USDA, United States Department of Agriculture. Documento eletrônico disponível em <<http://www.usda.gov/wps/portal/usdahome>>. Acesso em 10/06/2009.



## ANEXO V – CÓDIGO FONTE

## Introdução

Este documento contém o código fonte e os arquivos de configuração referentes ao trabalho de conclusão de curso Sistema de informação para Agricultura Familiar.

Este documento é dividido em quatro partes: configuração, SQL, XHTML e Java.

## Configuração

### Arquivo efagro\_v02-ds.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE datasources
  PUBLIC "-//JBoss//DTD JBOSS JCA Config 1.5//EN"
  "http://www.jboss.org/j2ee/dtd/jboss-ds_1_5.dtd">

<datasources>

  <local-tx-datasource>
    <jndi-name>efagro_v02Datasource</jndi-name>
    <connection-
url>jdbc:postgresql://localhost:5432/efagro_bd_v02</connection-url>
    <driver-class>org.postgresql.Driver</driver-class>
    <user-name>postgres</user-name>
    <password>postgres</password>
  </local-tx-datasource>

</datasources>
```

## Arquivo persistence.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd"
             version="1.0">

    <persistence-unit name="efagro_v02" transaction-type="JTA">
        <provider>org.hibernate.ejb.HibernatePersistence</provider>
        <jta-data-source>java:/efagro_v02Datasource</jta-data-source>
        <properties>
            <property name="hibernate.dialect"
value="org.hibernate.dialect.PostgreSQLDialect"/>
            <property name="hibernate.hbm2ddl.auto" value="validate"/>
            <property name="hibernate.show_sql" value="true"/>
            <property name="hibernate.format_sql" value="true"/>
            <property name="hibernate.transaction.manager_lookup_class"
value="org.hibernate.transaction.JBossTransactionManagerLookup"/>
        </properties>
    </persistence-unit>

</persistence>
```

## Arquivo web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://java.sun.com/xml/ns/javaee"
         xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID"
         version="2.5">
    <display-name>efagro_v02</display-name>
    <welcome-file-list>
```

```

    <welcome-file>index.html</welcome-file>
</welcome-file-list>
<servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.seam</url-pattern>
</servlet-mapping>
<context-param>
    <param-name>org.richfaces.SKIN</param-name>
    <param-value>blueSky</param-value>
</context-param>
<listener>
    <listener-class>org.jboss.seam.servlet.SeamListener</listener-class>
</listener>
<filter>
    <filter-name>Seam Filter</filter-name>
    <filter-class>org.jboss.seam.servlet.SeamFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>Seam Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
<servlet>
    <servlet-name>Seam Resource Servlet</servlet-name>
    <servlet-class>org.jboss.seam.servlet.SeamResourceServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>Seam Resource Servlet</servlet-name>
    <url-pattern>/seam/resource/*</url-pattern>
</servlet-mapping>
<context-param>
    <param-name>facelets.DEVELOPMENT</param-name>
    <param-value>>true</param-value>
</context-param>

```

```

<context-param>
  <param-name>javax.faces.DEFAULT_SUFFIX</param-name>
  <param-value>.<u>xhtml</u></param-value>
</context-param>
<security-constraint>
  <display-name>Restrict raw XHTML Documents</display-name>
  <web-resource-collection>
    <web-resource-name>XHTML</web-resource-name>
    <url-pattern>*<u>xhtml</u></url-pattern>
  </web-resource-collection>
  <auth-constraint/>
</security-constraint>
</web-app>

```

## Arquivo pages.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<pages xmlns="http://jboss.com/products/seam/pages"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://jboss.com/products/seam/pages
http://jboss.com/products/seam/pages-2.1.xsd"
  no-conversation-view-id="/home.xhtml"
  login-view-id="/login.xhtml">

  <page view-id="*">
    <navigation>
      <rule if-outcome="home">
        <redirect view-id="/home.xhtml"/>
      </rule>
      <!-- Usuarios -->
      <rule if-outcome="goAlterarSenha">
        <redirect view-id="/usuario/alteraSenha.xhtml"></redirect>
      </rule>
      <rule if-outcome="goListarUsuarios">
        <redirect view-id="/usuario/listaUsuarios.xhtml"></redirect>
      </rule>
      <rule if-outcome="goCadastrarUsuario">

```

```

        <redirect view-
id="/usuario/cadastraUsuario.xhtml"></redirect>
    </rule>

    <!-- Maquinas -->
    <rule if-outcome="goCadastrarMaquina">
        <redirect view-
id="/maquina/cadastrarMaquina.xhtml"></redirect>
    </rule>
    <rule if-outcome="goListarMaquinas">
        <redirect view-
id="/maquina/listarMaquinas.xhtml"></redirect>
    </rule>
    <rule if-outcome="goAgendarManutencao">
        <redirect view-
id="/maquina/agendarManutencao.xhtml"></redirect>
    </rule>
    <rule if-outcome="goRegistarManutencao">
        <redirect view-
id="/maquina/registrarManutencao.xhtml"></redirect>
    </rule>
    <rule if-outcome="goRegistarAbastecimento">
        <redirect view-
id="/maquina/registrarAbastecimento.xhtml"></redirect>
    </rule>
    <rule if-outcome="goRegistarOperacao">
        <redirect view-
id="/maquina/registrarOperacao.xhtml"></redirect>
    </rule>

    <!-- Propriedade -->
    <rule if-outcome="goAlterarPropriedade">
        <redirect view-id="/Propriedade.xhtml"></redirect>
    </rule>

    <!-- Agricultura -->
    <rule if-outcome="goDefinirAnoSafra">

```

```

        <redirect view-
id="/agricultura/cadastraSafra.xhtml"></redirect>
    </rule>
    <rule if-outcome="goListarAnoSafra">
        <redirect view-
id="/agricultura/listarSafras.xhtml"></redirect>
    </rule>
    <rule if-outcome="goListarTalhoes">
        <redirect view-
id="/agricultura/listarTalhoes.xhtml"></redirect>
    </rule>
    <rule if-outcome="goInserirTalhao">
        <redirect view-
id="/agricultura/cadastraTalhao.xhtml"></redirect>
    </rule>
    <rule if-outcome="goInserirOcorrPraga">
        <redirect view-
id="/agricultura/cadOcorrenciaPraga.xhtml"></redirect>
    </rule>

    <!-- Atividades Agrícolas -->
    <rule if-outcome="goInserirPlantio">
        <redirect view-
id="/agricultura/cadastraPlantio.xhtml"></redirect>
    </rule>
    <rule if-outcome="goInserirAplicacaoDefensivo">
        <redirect view-
id="/agricultura/cadastraApliDefensivo.xhtml"></redirect>
    </rule>
    <rule if-outcome="goInserirAdubacao">
        <redirect view-
id="/agricultura/cadastraAdubacao.xhtml"></redirect>
    </rule>
    <rule if-outcome="goInserirColheita">
        <redirect view-
id="/agricultura/cadastraColheita.xhtml"></redirect>
    </rule>
    <rule if-outcome="goRegistrarOutrasAtividades">

```

```

        <redirect view-
id="/agricultura/cadOutrasAtividades.xhtml"></redirect>
    </rule>

    <!-- Relatorios -->
    <rule if-outcome="goOperacoesMaquina">
        <redirect view-
id="/relatorios/relMaquinas.xhtml"></redirect>
    </rule>

    <!-- Relatorios -->
    <rule if-outcome="goRelAnoSafra">
        <redirect view-
id="/relatorios/relAnoSafra.xhtml"></redirect>
    </rule>

</navigation>
</page>

<exception class="org.jboss.seam.framework.EntityNotFoundException">
    <redirect view-id="/error.xhtml">
        <message severity="warn">Record not found</message>
    </redirect>
</exception>

<exception class="javax.persistence.EntityNotFoundException">
    <redirect view-id="/error.xhtml">
        <message severity="warn">Record not found</message>
    </redirect>
</exception>

<exception class="javax.persistence.EntityExistsException">
    <redirect view-id="/error.xhtml">
        <message severity="warn">Duplicate record</message>
    </redirect>
</exception>

<exception class="javax.persistence.OptimisticLockException">

```



```

        <end-conversation/>
        <redirect view-id="/error.xhtml">
            <message severity="warn">Another user changed the same data,
please try again</message>
        </redirect>
    </exception>

    <exception class="org.jboss.seam.security.AuthorizationException">
        <redirect view-id="/error.xhtml">
            <message
severity="error">#{messages.voceNaoTemPermissaoParaAcessarEsteRecurso}</messag
e>
        </redirect>
    </exception>

    <exception class="org.jboss.seam.security.NotLoggedInException">
        <redirect view-id="/login.xhtml">
            <message
severity="warn">#{messages['org.jboss.seam.NotLoggedIn']}</message>
        </redirect>
    </exception>

    <exception class="javax.faces.application.ViewExpiredException">
        <redirect view-id="/error.xhtml">
            <message severity="warn">#{messages.suaSessaoTerminou}</message>
        </redirect>
    </exception>

    <exception class="org.jboss.seam.ConcurrentRequestTimeoutException" log-
level="trace">
        <http-error error-code="503" />
    </exception>

    <exception>
        <redirect view-id="/error.xhtml">
            <message severity="error">#{messages.erroInesperado}</message>
        </redirect>
    </exception>

```

```
</pages>
```

## Arquivo jboss-web.xml

```
<!DOCTYPE jboss-web PUBLIC
    "-//JBoss//DTD Web Application 4.2//EN"
    "http://www.jboss.org/j2ee/dtd/jboss-web_4_2.dtd">

<jboss-web>
    <class-loading java2ClassLoadingCompliance="false">
        <loader-repository>
            seam.jboss.org:loader=efagro_v02
            <loader-repository-config>java2ParentDelegation=false</loader-
repository-config>
        </loader-repository>
    </class-loading>
    <context-root>/efagro</context-root>
</jboss-web>
```

## Arquivo faces-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<faces-config
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-facesconfig_1_2.xsd"
        version="1.2">
        <application>
            <view-handler>com.sun.facelets.FaceletViewHandler</view-handler>
            <locale-config>
                <default-locale>pt_BR</default-locale>
            </locale-config>
        </application>

</faces-config>

```

## Arquivo components.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<components xmlns="http://jboss.com/products/seam/components"
    xmlns:core="http://jboss.com/products/seam/core"
    xmlns:persistence="http://jboss.com/products/seam/persistence"
    xmlns:drools="http://jboss.com/products/seam/drools"
    xmlns:bpm="http://jboss.com/products/seam/bpm"
    xmlns:security="http://jboss.com/products/seam/security"
    xmlns:mail="http://jboss.com/products/seam/mail"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation=
        "http://jboss.com/products/seam/core
http://jboss.com/products/seam/core-2.1.xsd
        http://jboss.com/products/seam/persistence
http://jboss.com/products/seam/persistence-2.1.xsd
        http://jboss.com/products/seam/drools
http://jboss.com/products/seam/drools-2.1.xsd
        http://jboss.com/products/seam/bpm
http://jboss.com/products/seam/bpm-2.1.xsd
        http://jboss.com/products/seam/security
http://jboss.com/products/seam/security-2.1.xsd
        http://jboss.com/products/seam/mail
http://jboss.com/products/seam/mail-2.1.xsd

```

```

        http://jboss.com/products/seam/components
http://jboss.com/products/seam/components-2.1.xsd">

<core:init debug="true" jndi-pattern="@jndiPattern@"/>

<component name="org.jboss.seam.debug.hotDeployFilter">
    <property name="urlPattern">*.seam</property>
</component>

<core:manager concurrent-request-timeout="500"
    conversation-timeout="120000"
    conversation-id-parameter="cid"
    parent-conversation-id-parameter="pid"/>

<persistence:managed-persistence-context name="entityManager"
    auto-create="true"
    entity-manager-
factory="#{efagro_v02EntityManagerFactory}"/>

<persistence:entity-manager-factory name="efagro_v02EntityManagerFactory"
    persistence-unit-name="efagro_v02"/>

<drools:rule-base name="securityRules">
    <drools:rule-files><value>/security.drl</value></drools:rule-files>
</drools:rule-base>

<security:rule-based-permission-resolver security-
rules="#{securityRules}"/>

<security:identity authenticate-method="#{authenticator.authenticate}"
remember-me="true"/>

<event type="org.jboss.seam.security.notLoggedIn">
    <action execute="#{redirect.captureCurrentView}"/>
</event>

<event type="org.jboss.seam.security.loginSuccessful">
    <action execute="#{redirect.returnToCapturedView}"/>
</event>

```

```
<mail:mail-session host="smtp.gmail.com" username="teste.vproject@gmail.com"
password="Chico@faccin" />
```

```
</components>
```

## Arquivo messages\_pt\_BR.properties

```
up=\u2191
```

```
down=\u2193
```

```
left=\u2039
```

```
right=\u203A
```

```
validator.assertFalse=validation failed
```

```
validator.assertTrue=validation failed
```

```
validator.future=must be a future date
```

```
validator.length=length must be between {min} and {max}
```

```
validator.max=must be less than or equal to {value}
```

```
validator.min=must be greater than or equal to {value}
```

```
validator.notNull=may not be null
```

```
validator.past=must be a past date
```

```
validator.pattern=must match "{regex}"
```

```
validator.range=O valor deve ser entre {min} e {max}
```

```
validator.size=size must be between {min} and {max}
```

```
validator.email=Endere\u00E7o de E-mail Inv\u00E1lido
```

```
org.jboss.seam.loginFailed=Falha ao Entrar
```

```
org.jboss.seam.loginSuccessful=Bem-Vindo, \#0\!
```

```
org.jboss.seam.TransactionFailed=Transaction failed
```

```
org.jboss.seam.NoConversation=The conversation ended, timed out or was
processing another request
```

```
org.jboss.seam.IllegalNavigation=Illegal navigation
```

```
org.jboss.seam.ProcessEnded=Process #0 already ended
```

```
org.jboss.seam.ProcessNotFound=Process #0 not found
```

org.jboss.seam.TaskEnded=Task #0 already ended  
org.jboss.seam.TaskNotFound=Task #0 not found  
org.jboss.seam.NotLoggedIn=Por favor, entre no sistema primeiro  
propriedadeAlteradaSucesso=Propriedade alterada com sucesso\!  
outrasAreas=Outras \u00E1reas

javax.faces.component.UIInput.CONVERSION=value could not be converted to the expected type  
javax.faces.component.UIInput.REQUIRED=Campo Obrigat\u00F3rio  
javax.faces.component.UIInput.UPDATE=an error occurred when processing your submitted information  
javax.faces.component.UISelectOne.INVALID=Valor inv\u00E1lido  
javax.faces.component.UISelectMany.INVALID=Valor inv\u00E1lido

javax.faces.converter.BigDecimalConverter.DECIMAL=0 valor deve ser um n\u00F3mero  
javax.faces.converter.BigDecimalConverter.DECIMAL\_detail=value must be a signed decimal number consisting of zero or more digits, optionally followed by a decimal point and fraction, eg. {1}  
javax.faces.converter.BigIntegerConverter.BIGINTEGER=0 valor deve ser um n\u00F3mero inteiro  
javax.faces.converter.BigIntegerConverter.BIGINTEGER\_detail=value must be a signed integer number consisting of zero or more digits  
javax.faces.converter.BooleanConverter.BOOLEAN=value must be true or false  
javax.faces.converter.BooleanConverter.BOOLEAN\_detail=value must be true or false (any value other than true will evaluate to false)  
javax.faces.converter.ByteConverter.BYTE=value must be a number between 0 and 255  
javax.faces.converter.ByteConverter.BYTE\_detail=value must be a number between 0 and 255  
javax.faces.converter.CharacterConverter.CHARACTER=value must be a character  
javax.faces.converter.CharacterConverter.CHARACTER\_detail=value must be a valid ASCII character  
javax.faces.converter.DateTimeConverter.DATE=0 valor deve ser uma data v\u00E1lida  
javax.faces.converter.DateTimeConverter.DATE\_detail=value must be a date, eg. {1}  
javax.faces.converter.DateTimeConverter.TIME=0 valor deve ser um hor\u00E1rio

javax.faces.converter.DateTimeConverter.TIME\_detail=value must be a time, eg. **{1}**

javax.faces.converter.DateTimeConverter.DATETIME=value must be a date and time

javax.faces.converter.DateTimeConverter.DATETIME\_detail=value must be a date and time, eg. **{1}**

javax.faces.converter.DateTimeConverter.PATTERN\_TYPE=a pattern or type attribute must be specified to convert the value

javax.faces.converter.DoubleConverter.DOUBLE=0 valor precisa ser um n\u00F0FAmero

javax.faces.converter.DoubleConverter.DOUBLE\_detail=0 valor precisa ser um n\u00F0FAmero entre 4.9E-324 e 1.7976931348623157E308

javax.faces.converter.EnumConverter.ENUM=value must be convertible to an enum

javax.faces.converter.EnumConverter.ENUM\_detail=value must be convertible to an enum or from the enum that contains the constant **{1}**

javax.faces.converter.EnumConverter.ENUM\_NO\_CLASS=value must be convertible to an enum or from the enum, but no enum class provided

javax.faces.converter.EnumConverter.ENUM\_NO\_CLASS\_detail=value must be convertible to an enum or from the enum, but no enum class provided

javax.faces.converter.FloatConverter.FLOAT=0 valor deve ser um n\u00F0FAmero

javax.faces.converter.FloatConverter.FLOAT\_detail=0 valor precisa ser um n\u00F0FAmero entre 1.4E-45 e 3.4028235E38

javax.faces.converter.IntegerConverter.INTEGER=0 valor precisa ser um inteiro

javax.faces.converter.IntegerConverter.INTEGER\_detail=value must be an integer number between -2147483648 and 2147483647

javax.faces.converter.LongConverter.LONG=0 valor precisa ser um inteiro (long)

javax.faces.converter.LongConverter.LONG\_detail=value must be an integer number between -9223372036854775808 and 9223372036854775807

javax.faces.converter.NumberConverter.CURRENCY=0 valor deve ser um vaor monet\u00EElrio

javax.faces.converter.NumberConverter.CURRENCY\_detail=value must be a currency amount, eg. **{1}**

javax.faces.converter.NumberConverter.PERCENT=value must be a percentage amount

javax.faces.converter.NumberConverter.PERCENT\_detail=value must be a percentage amount, eg. **{1}**

javax.faces.converter.NumberConverter.NUMBER=0 valor deve ser um n\u00F0FAmero

javax.faces.converter.NumberConverter.NUMBER\_detail=value must be a number

javax.faces.converter.NumberConverter.PATTERN=value must be a number

javax.faces.converter.NumberConverter.PATTERN\_detail=value must be a number  
javax.faces.converter.ShortConverter.SHORT=value must be an integer  
javax.faces.converter.ShortConverter.SHORT\_detail=value must be an integer  
number between -32768 and 32767

javax.faces.validator.DoubleRangeValidator.MAXIMUM=value must be less than or  
equal to {0}  
javax.faces.validator.DoubleRangeValidator.MINIMUM=value must be greater than  
or equal to {0}  
javax.faces.validator.DoubleRangeValidator.NOT\_IN\_RANGE=value must be between  
{0} and {1}  
javax.faces.validator.DoubleRangeValidator.TYPE=value is not of the correct  
type  
javax.faces.validator.LengthValidator.MAXIMUM=value must be shorter than or  
equal to {0} characters  
javax.faces.validator.LengthValidator.MINIMUM=value must be longer than or  
equal to {0} characters  
javax.faces.validator.LongRangeValidator.MAXIMUM=value must be less than or  
equal to {0}  
javax.faces.validator.LongRangeValidator.MINIMUM=value must be greater than or  
equal to {0}  
javax.faces.validator.LongRangeValidator.NOT\_IN\_RANGE=value must be between  
{0} and {1}  
javax.faces.validator.LongRangeValidator.TYPE=value is not of the correct type

javax.faces.validator.NOT\_IN\_RANGE=value must be between {0} and {1}

javax.faces.converter.STRING=value could not be converted to a string

bemVindoAoEfgro=Bem-Vindo ao projeto E-fagro  
entrar=Entrar  
sair=Sair  
senhasDiferentes=As senhas n\u00E3o conferem  
senhaAlterada=Senha Alterada com sucesso\!  
senhaAtual=Senha Atual  
novaSenha=Nova Senha  
confirmaSenha=Confirme a nova senha  
senhaAtualInvalida=A senha atual \u00E9 incorreta  
inicio=In\u00E9cio  
copyRight=\u00A9 2009 E-fagro



politicaDePrivacidade=Pol\&u00EDtica de Privacidade  
sobre=Sobre  
selecioneIdioma=Selecione seu idioma  
mudaIdioma=Mudar idioma  
mapaDoSite=Mapa do site  
porFavorFacaOLogin=Por favor, entre com seu e-mail e senha para utilizar o sistema  
lembrarEmail=Lembrar o E-mail neste computador  
email=E-mail  
nome=Nome  
senhaConfirma=Redigite a senha  
senha=Senha  
tituloHome=O E-FAgro vai aumentar sua capacidade de gest&u00E3o.  
subTituloHome=Com ele voc&u00EA pode\:  
# Funcionalidades do Efagro  
funcionalidades1=Cadastrar os usu&u00Elrios da propriedade com atribui&u00E7&u00F5es de administrador ou colaborador  
funcionalidades2=Controlar das cultivares perenes e anuais, incid&u00EAncia de pragas, investimentos e resultados  
funcionalidades3=Manter hist&u00F3rico das atividades agr&u00EDcolas  
funcionalidades4=Controlar do maquin&u00Elrio da propriedade, o consumo de combust&u00EDvel e a manuten&u00E7&u00E3o  
funcionalidades5=Gerar relat&u00F3rios de produtividade e lucratividade por talh&u00E3o, cultivar ou safra.  
funcionalidades6=Controlar a presta&u00E7&u00E3o de servi&u00E7os a terceiros  
funcionalidades7=Efetuar comparativos de produ&u00E7&u00E3o ao longo do tempo.  
#---  
cadastre=Clique aqui e fa&u00E7a j&u00E1 seu cadastro  
cancelar=Cancelar  
cadastrar=Cadastrar  
clienteCadastrado=Cliente cadastrado com sucesso. Verifique sua caixa de email para maiores instru&u00E7&u00F5es  
erroEnvioEmail=Falha ao enviar o E-mail  
camposObrigatorios=Campos com aster&u00EDsco s&u00E3o de preenchimento Obrigat&u00F3rio  
dadosPessoais=Dados Pessoais  
preenchaOFormulario=Preencha o formul&u00Elrio abaixo para efetuar o cadastro  
senhaInvalida=Senha Incorreta

emailInvalido=Email n\u00E3o cadastrado  
titulo=E-fagro - Gerencimento de Agroneg\u00F3cio  
telefone=Telefone com DDD  
numTelefone=N\u00F0Amoro do telefone  
ddd=C\u00F3digo de \u00C1rea  
emailJaCadastrado=Este E-mail j\u00E1 est\u00E1 cadastrado no sistema  
celular=Celular com DDD  
nomeCompleto=Nome Completo  
pais=Pa\u00EDs  
estado=Estado  
municipio=Munic\u00EDpio  
selecioneOPais=Selecione o Pa\u00EDs...  
selecioneOEstado=Selecione o Estado...  
selecioneOMunicipio=Selecione o Munic\u00EDpio...  
CEP=CEP - somente n\u00F0Ameros  
rua=Rua  
bairro=Bairro  
numero=N\u00F0Amoro  
complemento=Complemento  
endereco=Endere\u00E7o  
voceEstaLogadoComo=Voc\u00EA est\u00E1 logado como \\  
servicos=Servi\u00E7os  
agricultura=Agricultura  
area=\u00C1rea  
talhao=Talh\u00E3o  
pecuaria=Pecu\u00E1ria  
sexo=Sexo  
masculino=Masculino  
feminino=Feminino  
fertilizante=Fertilizante  
cadastrarUsuarioComoAdministrador=Cadastrar este usu\u00E1rio como  
administrador do sistema ?  
sim=Sim  
nao=N\u00E3o  
usuarios=Usu\u00E1rios  
cadastrarUsuarios=Cadastrar Usu\u00E1rios  
listarUsuarios=Listar Usu\u00E1rios  
administradorSistema=Administrador do Sistema

privilegio=Privil\u00E9gio  
administrador=Administrador  
colaborador=Colaborador  
cadastroDeUsuarios=Cadastro de Usu\u00E1rios  
altereOsDados=Altere os Dados do Usu\u00E1rio  
alterar=Alterar  
acao=A\u00E7\u00E3o  
alteraSenha=Alterar a Senha  
esqueciSenha=Esqueci a Senha  
recupereASenha=Recupere sua senha\  
enviaSenhaParaEmail=Enviar a senha para o E-mail  
verifiqueSuaCaixaDeEmail=Foi enviado para o email fornecida a sua senha, por favor verifique sua caixa de e-mails  
emailNaoCadastrado=Email N\u00E3o cadastrado  
usuarioAlterado=Altera\u00E7\u00E3o dos dados do usu\u00E1rio efetuada com Sucesso  
propriedade=Propriedade  
editarPropriedade=Editar dados da Propriedade  
nomeDaPropriedade=Nome da Propriedade  
dadosDaPropriedade=Dados da Propriedade  
enderecoDaPropriedade=Endere\u00E7o da Propriedade  
areaTotal=\u00C1rea total (em Hectares)  
somatorioDosTalhoes=Somat\u00F3rio das \u00E1reas dos talh\u00F5es da safra\  
adicionarTalhao=Adicionar Talh\u00E3o  
alteracaoEfetuada=Altera\u00E7\u00E3o efetuada com Sucesso\  
maoDeObra=M\u00E3o-de-obra  
cadastrarTrabalhadores=Cadastrar Trabalhadores  
listarTrabalhadores=Listar Trabalhadores  
excluir=Excluir  
usuarioRemovido=Usu\u00E1rio removido com Sucesso\  
oEmailNaoPodeSerAlterado=O Email do usu\u00E1rio n\u00E3o pode ser alterado\  
usuarioCadastrado=Usu\u00E1rio cadastrado com sucesso - Foi Enviado um email para o usuario contendo a senha de acesso  
trabalhadorRemovido=Trabalhador removido com sucesso\  
trabalhadorAlterado=Trabalhador alterado com sucesso\  
trabalhadorCadastrado=Trabalhador cadastrado com sucesso\  
dataNasc=Data de Nascimento  
horasDisponiveisPorDia=Horas Dispon\u00EDveis por Dia

```

dataFormato=dd/MM/yyyy
horasDisponiveis=Horas Dispon\u00EDveis
semPermissao=Voc\u00EA n\u00E3o tem permiss\u00E3o para executar esta tarefa
erro=Erro
nenhumTrabalhadorcadastrado=Nenhum trabalhador cadastrado
nenhumUsuariocadastrado=Nenhum Usu\u00E1rio cadastrado
#-- Tarefas
tarefas=Tarefas
cadastrarTarefas=Cadastrar Tarefas
alterarTarefas=Alterar Tarefas
listarTarefas=Listar Tarefas
#-- Transacoes
transacoes=Transa\u00E7\u00F5es
compra=Compra
venda=Venda
relatorios=Relat\u00F3rios
maquinaCadastrada=M\u00E1quina cadastrada com sucesso\!
maquinaAlterada=M\u00E1quina alterada com sucesso\!
maquinaRemovida=M\u00E1quina removida com sucesso\!
maquinas=M\u00E1quinas
cadastrarMaquinas=Cadastrar M\u00E1quinas
listarMaquinas=Listar M\u00E1quinas
nenhumaMaquinaCadastrada=Nem uma m\u00E1quina cadastrada
cadastroDeTrabalhadores=Cadastro de trabalhadores
cadastroDeMaquinas=Cadastro de M\u00E1quinas
modelo=Modelo
marca=Marca
dataAquisicao=Data de aquisi\u00E7\u00E3o
anoFabricacao=Ano de Fabrica\u00E7\u00E3o
valorMaquina=Valor da M\u00E1quina
valorHoraMaquina=Valor da hora da M\u00E1quina
numSerie=N\u00FAmero de s\u00E9rie
editarMaquina=Editar M\u00E1quina
observacao=Observa\u00E7\u00E3o
editarDadosTrabalhador=Editar dados do Trabalhador
definirAnoSafra=Definir Ano/Safra
areaPreservacao=\u00C1rea de Preserva\u00E7\u00E3o
areaCultivo=\u00C1rea de Cultivo

```

```

numeroDeTalhoesQueDeseja=N\u00FAmero de talhoes que voc\u00EA deseja criar
safraAlterada=Ano/Safra Alterado com sucesso\!
safraCadastrada=Ano/Safra cadastrado com sucesso\!
anoSafra=Ano/Safra
dataInicio=Data do In\u00EDcio
dataFim=Data do Final
dataFimDeveSerPorteriorAoInicio=A data Final do ano/safra deve ser posterior a
data de in\u00EDcio
safraJaCadastrada=Ano/Safra j\u00E1 cadastrado
talhoesDaSafraAtual=Talh\u00F5es da safra atual
definirTalhoes=Definir os Talh\u00F5es
safraAtual=Safra Atual
nenhumTalhaoCadastrado=Nenhum talh\u00E3o cadastrado para esta safra
buscar=Buscar
selecioneASafra=Selecione o Ano/Safra
criar=Criar
localizacao=Descri\u00E7\u00E3o da Localiza\u00E7\u00E3o
nomeTalhao=Nome do Talh\u00E3o
processando=Processando
porFavoAguarde=Por favor aguarde...
descricao=Descri\u00E7\u00E3o
excluirEsteTalhaoDaSafra=Excluir este talh\u00E3o da safra ?
remover'=Remove
hectares=Hectares
ha=Ha
talhaoRemovido=Talh\u00E3o removido com sucesso\!
talhoesCadastrados=Talh\u00F5es Cadastrados
desejaRemoverOTalhao=Deseja remover o talh\u00E3o
interrogacao=\ ?
alterarTalhao=Alterar dados do Talh\u00E3o
insiraALocalizacaoDoTalhao=Insira a localiza\u00E7\u00E3o do Talh\u00E3o
terrenoProprio=Terreno pr\u00F3prio
cultivoPermanente=Cultivo permanente
talhaoAlterado=Talh\u00E3o Alterado com Sucesso\!
dadosDoTalhao=Dados do Talh\u00E3o
# Mensagens Abastecimento
abastecimentoRegistrado=Abastecimento Registrado
erroRegistrarAbastecimento=Erro ao registrar abastecimento

```

aducacaoRegistrado=Aduba\u00E7\u00E3o Registrada\!  
erroRegistrarAdubacao=Erro ao registrar aduba\u00E7\u00E3o\!  
manutencaoAgendada=Manuten\u00E7\u00E3o Agendada\!  
erroAgendarManutencao=Erro ao agendar Manuten\u00E7\u00E3o\!  
erroRegistrarAplicacaoDefensivo=Erro na aplica\u00E7\u00E3o do defensivo\!  
aplicacaoDefensivoRegistrado=Aplica\u00E7\u00E3o de defensivo registrado\!  
atividadeRegistrada=Atividade Registrada\!  
erroRegistrarAtividade=Erro registrar atividade  
manutencaoRegistrado=Manuten\u00E7\u00E3o registrada\!  
erroRegistrarManutencao=Erro registrar manuten\u00E7\u00E3o\!  
pragaRegistrada=Praga registrada\!  
erroRegistrarPraga=Erro registrar praga\!  
operacaoRegistrada=Opera\u00E7\u00E3o registrada\!  
erroRegistrarOperacao=Erro registrar opera\u00E7\u00E3o\!  
erroRegistrarPlantio=Erro registrar plantio\!  
plantioRegistrado=Plantio registrado\!  
erroCadastrarSafra=Erro cadastrar safra  
talhaoCadastrado=Talh\u00E3o cadastrado\!  
erroCadastrarTalhao=Erro cadastrar talh\u00E3o\!  
erroCadastrarUsuario=Erro cadastrar usu\u00E1rio\!  
erroCadastrarCliente=Erro ao cadastrar Cliente\!  
cpf=CPF  
colheitaRegistrada=Colheita Registrada\!  
erroRegistrarColheita=Erro ao registrar Colheita\!  
erroCadastrarMaquina=Erro ao cadastrar M\u00E1quina\!  
erroCadastrarPropriedade=Erro ao cadastrar propriedade\!  
propriedadeCadastrada=Propriedade Cadastrada\!  
cidade=Cidade  
selecioneACidade=Selecione a Cidade  
agendarManutencao=Agendar Manuten\u00E7\u00E3o  
registrarManutencao=Registrar Manuten\u00E7\u00E3o  
registrarAbastecimento=Registrar Abastecimento  
listarAnoSafra=Listar Anos/Safra  
listarTalhoes=Listar Talh\u00F5es  
inserirTalhao=Inserir Talh\u00E3o  
alterarDadosPropriedade=Alterar dados da Propriedade  
erroAlterarMaquina=Erro ao alterar M\u00E1quina\!  
erroAlterarPropriedade=Erro ao alterar dados da propriedade.

erroAlterarUsuario=Erro ao alterar dados do Usu\u00E1rio\!  
erroRemoverUsuario=Erro ao remover Usu\u00E1rio  
remover=Remover  
alterando=Alterando\:  
detalhes=Detalhar informa\u00E7\u00F5es  
abastecimentos=Abastecimentos\:  
manutencoesAgendadas=Manuten\u00E7\u00F5es Agendadas\:  
manutencoesEfetuadas=Manuten\u00E7\u00F5es Efetuadas\:  
data=Data  
combustivel=Combust\u00EDvel  
quantidade=Quantidade  
valorUnitario=Valor unit\u00E1rio  
horimetroHodometro=Hor\u00EDmetro/Hod\u00F4metro  
lembrete=Lembrete  
valorTotal=Valor total  
selecioneAMaquina=Selecione a M\u00E1quina  
maquina=M\u00E1quina  
dataManutencao=Data da Manuten\u00E7\u00E3o  
dataAviso=Data do Aviso  
frequencia=Frequ\u00EAncia  
aPropriedadeNaoTemSafraAtualDefinida=A propriedade n\u00E3o tem uma safra  
definida para o per\u00EDodo  
definirSafra=Definir Ano/Safra  
exemploAnoSafra=Ex\ : 2009/2010  
cadastraTalhao=Cadastrar Talh\u00E3o\:  
selecioneOAnoSafra=Selecione o Ano/Safra  
nenhumaSafraCadastrada=Nem uma safra cadastrada  
selecioneASafraParaVerOsTalhoes=Selecione o Ano/Safra para visualizar seus  
talh\u00F5es  
nenhumaTalhaoCadastrado=Nem um talh\u00E3o cadastrado para a safra  
talhoesDaSafra=Talh\u00F5es da Safra\:  
registrarAtividadeTalhao=Registrar atividade no talh\u00E3o  
voceNaoTemPermissaoParaAcessarEsteRecurso=Voc\u00EA n\u00E3o tem  
permiss\u00E3o para acessar este recurso\!  
suaSessaoTerminou=Sua sess\u00E3o terminou, por favor, tente novamente\!  
erroInesperado=Erro inesperado  
registrarOperacao=Registrar Opera\u00E7\u00E3o  
horasTrabalhadas=Horas trabalhadas

adubacao=Aduba\u00E7\u00E3o  
aplicacaoDefensivo=Aplica\u00E7\u00E3o de Defensivo  
plantio=Plantio  
ocorrenciaPraga=Ocorrencia de Praga  
colheita=Colheita  
operacoesEfetuadas=Opera\u00E7\u00F5es Efetuadas  
atividadesTalhao=Atividades no talh\u00E3o  
codISOMoeda=BRL  
simboloMoeda=R\$\n registrarOcorrenciaPraga=Registrar ocorr\u00Eancia de praga  
solucao=Solu\u00E7\u00E3o  
selecioneOTalhao=Selecione o talh\u00E3o  
anoSafraDaPropriedade=Ano/Safra \n :  
talhaodP=Talh\u00E3o\n :  
custoOperacao=Custo da opera\u00E7\u00E3o  
totaisDoTalhao=Totais do talh\u00E3o  
totaisDeOperacoesComMaquinas=Totais dos custos de opera\u00E7\u00F5es com  
m\u00E1quinas  
adubacoesEfetuadas=Aduba\u00E7\u00F5es Efetuadas  
totaisDeAdubacoes=Custo total de aduba\u00E7\u00F5es  
hora=Hora  
aplicacoesDefensivosEfetuadas=Aplica\u00E7\u00F5es de defensivos Efetuadas  
defensivo=Defensivo  
totaisDedefensivos=Total dos valores de defensivos  
totaisDePlantio=Total dos valores de plantio  
cultura=Cultura  
variedade=Variedade  
dataPrevistaColheita=Data Prevista da colheita  
plantiosEfetuados=Plantios Efetuados  
colheitasEfetuadas=Colheitas Efetuadas  
totaisDeColheita=Totais dos valores de colheita  
registrarAtividade=Registrar atividade no Talh\u00E3o\n :  
obeservacao=Observa\u00E7\u00E3o  
dataAtividade=Data da atividade  
dataOcorrencia=Data da ocorrencia  
registrarAdubacao=Registrar Aduba\u00E7\u00E3o  
saldoDoTalhaoEh=0 saldo atual do talh\u00E3o \u00E9\n :  
noTalhao=no talh\u00E3o\n :



registrarAplicacaoDefensivo=Registrar Aplica\u00E7\u00E3o de Defensivo  
registrarPlantio=Registrar Plantio  
registrarColheita=Regsitrar Colheita  
registrarOutrasOperacoes=Registrar outras opera\u00E7\u00F5es  
seleccioneOFertilizante=Selecione o fertilizante  
maquinasUtilizadas=M\u00E9quinas utilizadas  
dataFinalCarencia=Final do periodo de car\u00Eancia em\  
umidadeRelativa=Umidade relativa do ar  
temperatura=Temperatura  
registrarOutrasAtividades=Registrar outras Atividades  
seleccioneODefensivo=Selecione o defensivo  
seleccioneAVariiedade=Selecione a variedade  
seleccioneACultura=Selecione a cultura  
gerarRelatorio=Gerar relat\u00F3rio  
ReltorioDeOperacoesDaMaquina=Relat\u00F3rio de opera\u00E7\u00F5es da  
M\u00E9quina  
relatorioOperacoesMaquina=Relat\u00F3rio de Opera\u00E7\u00E3oes  
totaisDeOperacoesDaMaquina=Totais de opera\u00E7\u00E3o da m\u00E9quina  
relatorioAnoSafra=Relat\u00F3rio por Ano/Safra  
reltorioDosAnoSafra=Relat\u00F3rio do Ano/Safra  
custoDaOperacao=Custo da opera\u00E7\u00E3o  
relatoriosAgricolas=Relat\u00F3rios Agr\u00EDcolas  
relatoriosMaquinas=Relat\u00F3rios de Maquin\u00E9rio  
relatorioManutencaoMaquina=Relat\u00F3rio de Manuten\u00E7\u00F5es  
relatorioCultura=Relat\u00F3rio por Cultura  
relatorioVariiedade=Relat\u00F3rio po Variiedade  
relatorioPragas=Relat\u00F3rio de Incid\u00Eancia de Pragas  
relatorioTalhao=Relat\u00F3rio por Talh\u00E3o  
totalArrecadado=Total Arrecadado  
totalInvestimento=Total Investido  
saldoTalhao=Saldo do Talh\u00E3o  
saldoDaSafraEh=O saldo da safra \u00E9\  
totaisDaSafra=Totais da Safra

## SQL

Arquivo efagro\_bd\_v02.sql

```
-- Table: abastecimento

-- DROP TABLE abastecimento;

CREATE TABLE abastecimento
(
    id_abastecimento integer NOT NULL DEFAULT
nextval('id_abastecimento_seq'::regclass),
    cod_maquina integer NOT NULL,
    data_abastecimento date NOT NULL,
    tipo_combustivel text NOT NULL,
    quantidade double precision NOT NULL,
    valor_unitario double precision NOT NULL,
    horimetro_hodometro double precision,
    CONSTRAINT abastecimento_pkey PRIMARY KEY (id_abastecimento),
    CONSTRAINT abastecimento_cod_maquina_fkey FOREIGN KEY (cod_maquina)
REFERENCES maquina (id_maquina) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE abastecimento OWNER TO postgres;

-- Table: adubacao

-- DROP TABLE adubacao;

CREATE TABLE adubacao
(
    cod_fertilizante integer NOT NULL,
    quantidade double precision NOT NULL,
    valor_total double precision NOT NULL,
    cod_atividade_talhao bigint NOT NULL,
    CONSTRAINT adubacao_pkey PRIMARY KEY (cod_atividade_talhao),
```

```

    CONSTRAINT adubacao_cod_atividade_talhao_fkey FOREIGN KEY
(cod_atividade_talhao)
    REFERENCES atividade_talhao (id_atividade_talhao) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE CASCADE,
    CONSTRAINT adubacao_cod_fertilizante_fkey FOREIGN KEY (cod_fertilizante)
    REFERENCES fertilizante (id_fertilizante) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE adubacao OWNER TO postgres;

-- Table: agenda_manutencao

-- DROP TABLE agenda_manutencao;

CREATE TABLE agenda_manutencao
(
    id_agenda_manutencao integer NOT NULL DEFAULT
nextval('id_agenda_manutencao_seq'::regclass),
    data_manutencao date NOT NULL,
    lembrete text,
    data_aviso date NOT NULL,
    frequencia integer NOT NULL, -- 0,1,2... (Nem uma, di ria, semanal,
quinzenal, mensal, trimestral, semestral ou anual)
    cod_maquina integer NOT NULL,
    CONSTRAINT agenda_manutencao_pkey PRIMARY KEY (id_agenda_manutencao),
    CONSTRAINT agenda_manutencao_cod_maquina_fkey FOREIGN KEY (cod_maquina)
    REFERENCES maquina (id_maquina) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE agenda_manutencao OWNER TO postgres;
COMMENT ON COLUMN agenda_manutencao.frequencia IS '0,1,2... (Nem uma, di ria,
semanal, quinzenal, mensal, trimestral, semestral ou anual)';

-- Table: aplicacao_defensivo

-- DROP TABLE aplicacao_defensivo;

```

```

CREATE TABLE aplicacao_defensivo
(
  cod_atividade_talhao bigint NOT NULL,
  cod_defensivo integer NOT NULL,
  hora time without time zone NOT NULL,
  quantidade double precision NOT NULL,
  carencia date NOT NULL,
  umidade double precision,
  temperatura double precision,
  valor_total double precision NOT NULL,
  CONSTRAINT aplicacao_defensivo_pkey PRIMARY KEY (cod_atividade_talhao),
  CONSTRAINT aplicacao_defensivo_cod_atividade_talhao_fkey FOREIGN KEY
(cod_atividade_talhao)
  REFERENCES atividade_talhao (id_atividade_talhao) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE CASCADE,
  CONSTRAINT aplicacao_defensivo_cod_defensivo_fkey FOREIGN KEY
(cod_defensivo)
  REFERENCES defensivo (id_defensivo) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE aplicacao_defensivo OWNER TO postgres;
-- Table: atividade_talhao

-- DROP TABLE atividade_talhao;

CREATE TABLE atividade_talhao
(
  id_atividade_talhao bigint NOT NULL DEFAULT
nextval('id_atividade_talhao_seq'::regclass),
  cod_talhao bigint NOT NULL,
  data_atividade date NOT NULL,
  observacao text,
  CONSTRAINT atividade_talhao_pkey PRIMARY KEY (id_atividade_talhao),
  CONSTRAINT atividade_talhao_cod_talhao_fkey FOREIGN KEY (cod_talhao)
  REFERENCES talhao (id_talhao) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE CASCADE
)

```

```

)
WITH (OIDS=FALSE);
ALTER TABLE atividade_talhao OWNER TO postgres;
-- Table: cidade

-- DROP TABLE cidade;

CREATE TABLE cidade
(
    id_cidade integer NOT NULL DEFAULT nextval('id_cidade_seq'::regclass),
    nome text NOT NULL, -- nome da cidade
    cod_estado integer NOT NULL,
    CONSTRAINT cidade_pkey PRIMARY KEY (id_cidade),
    CONSTRAINT cidade_cod_estado_fkey FOREIGN KEY (cod_estado)
        REFERENCES estado (id_estado) MATCH SIMPLE
        ON UPDATE CASCADE ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE cidade OWNER TO postgres;
COMMENT ON COLUMN cidade.nome IS 'nome da cidade';

-- Table: cliente

-- DROP TABLE cliente;

CREATE TABLE cliente
(
    id_cliente integer NOT NULL DEFAULT nextval('id_cliente_seq'::regclass),
    nome text NOT NULL,
    email text NOT NULL,
    sexo character(1),
    data_nascimento date NOT NULL,
    cpf text NOT NULL,
    cod_endereco integer NOT NULL,
    telefone text NOT NULL,
    celular text NOT NULL,
    CONSTRAINT cliente_pkey1 PRIMARY KEY (id_cliente),
    CONSTRAINT cliente_cod_endereco_fkey FOREIGN KEY (cod_endereco)

```

```

        REFERENCES endereco (id_endereco) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE CASCADE,
        CONSTRAINT cliente_email_key UNIQUE (email)
    )
    WITH (OIDS=FALSE);
ALTER TABLE cliente OWNER TO postgres;
-- Table: colheita

-- DROP TABLE colheita;

CREATE TABLE colheita
(
    cod_atividade_talhao bigint NOT NULL,
    quantidade double precision NOT NULL,
    valor_total double precision NOT NULL,
    CONSTRAINT colheita_pkey PRIMARY KEY (cod_atividade_talhao),
    CONSTRAINT colheita_cod_atividade_talhao_fkey FOREIGN KEY
(cod_atividade_talhao)
        REFERENCES atividade_talhao (id_atividade_talhao) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE colheita OWNER TO postgres;
-- Table: cultura

-- DROP TABLE cultura;

CREATE TABLE cultura
(
    id_cultura integer NOT NULL DEFAULT nextval('id_cultura_seq'::regclass),
    nome text NOT NULL,
    descricao text,
    CONSTRAINT cultura_pkey PRIMARY KEY (id_cultura)
)
WITH (OIDS=FALSE);
ALTER TABLE cultura OWNER TO postgres;
-- Table: defensivo

```

```

-- DROP TABLE defensivo;

CREATE TABLE defensivo
(
  id_defensivo integer NOT NULL DEFAULT nextval('id_defensivo_seq'::regclass),
  nome text NOT NULL,
  funcao text NOT NULL,
  formula text,
  cod_fabricante integer,
  CONSTRAINT defensivo_pkey PRIMARY KEY (id_defensivo),
  CONSTRAINT defensivo_cod_fabricante_fkey FOREIGN KEY (cod_fabricante)
    REFERENCES fabricante (id_fabricante) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE defensivo OWNER TO postgres;
-- Table: endereço

-- DROP TABLE endereço;

CREATE TABLE endereço
(
  id_endereco integer NOT NULL DEFAULT nextval('id_endereco_seq'::regclass),
  logradouro text,
  complemento text,
  bairro text,
  codigo_postal text,
  numero text,
  cod_cidade integer NOT NULL,
  CONSTRAINT logradouro_pkey PRIMARY KEY (id_endereco),
  CONSTRAINT logradouro_cod_cidade_fkey FOREIGN KEY (cod_cidade)
    REFERENCES cidade (id_cidade) MATCH SIMPLE
    ON UPDATE CASCADE ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE endereço OWNER TO postgres;
-- Table: estado

```

```

-- DROP TABLE estado;

CREATE TABLE estado
(
  id_estado integer NOT NULL DEFAULT nextval('id_estado_seq'::regclass),
  nome text NOT NULL, -- nome do estado ou provincia
  sigla text, -- sigla do estado quando houver
  cod_pais integer NOT NULL,
  CONSTRAINT estado_pkey PRIMARY KEY (id_estado),
  CONSTRAINT estado_cod_pais_fkey FOREIGN KEY (cod_pais)
    REFERENCES pais (id_pais) MATCH SIMPLE
    ON UPDATE CASCADE ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE estado OWNER TO postgres;
COMMENT ON TABLE estado IS 'estado ou provincia';
COMMENT ON COLUMN estado.nome IS 'nome do estado ou provincia
';
COMMENT ON COLUMN estado.sigla IS 'sigla do estado quando houver
';

-- Table: fabricante

-- DROP TABLE fabricante;

CREATE TABLE fabricante
(
  id_fabricante integer NOT NULL DEFAULT
nextval('id_fabricante_seq'::regclass),
  nome_fabricante text NOT NULL,
  CONSTRAINT fabricante_pkey PRIMARY KEY (id_fabricante)
)
WITH (OIDS=FALSE);
ALTER TABLE fabricante OWNER TO postgres;
-- Table: fertilizante

-- DROP TABLE fertilizante;

```



```

CREATE TABLE fertilizante
(
  id_fertilizante integer NOT NULL DEFAULT
nextval('id_fertilizante_seq'::regclass),
  nome text NOT NULL,
  propriedades text,
  cod_fabricante integer,
  CONSTRAINT fertilizante_pkey PRIMARY KEY (id_fertilizante),
  CONSTRAINT fertilizante_cod_fabricante_fkey FOREIGN KEY (cod_fabricante)
  REFERENCES fabricante (id_fabricante) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE fertilizante OWNER TO postgres;
-- Table: manutencao

-- DROP TABLE manutencao;

CREATE TABLE manutencao
(
  id_manutencao integer NOT NULL DEFAULT
nextval('id_manutencao_seq'::regclass),
  cod_maquina integer NOT NULL,
  data_manutencao date NOT NULL,
  descricao text NOT NULL,
  valor_total double precision NOT NULL,
  CONSTRAINT manutencao_pkey PRIMARY KEY (id_manutencao),
  CONSTRAINT manutencao_cod_maquina_fkey FOREIGN KEY (cod_maquina)
  REFERENCES maquina (id_maquina) MATCH SIMPLE
  ON UPDATE NO ACTION ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE manutencao OWNER TO postgres;
-- Table: maquina

-- DROP TABLE maquina;

CREATE TABLE maquina

```

```

(
  id_maquina integer NOT NULL DEFAULT nextval('id_maquina_seq'::regclass),
  modelo text NOT NULL,
  marca text NOT NULL,
  data_aquisicao date NOT NULL,
  ano_fabricacao integer NOT NULL,
  numero_serie text,
  observacao text,
  cod_propriedade integer NOT NULL,
  valor_maquina double precision NOT NULL,
  valor_hora double precision NOT NULL,
  nome text NOT NULL,
  CONSTRAINT maquina_pkey PRIMARY KEY (id_maquina),
  CONSTRAINT maquina_cod_propriedade_fkey FOREIGN KEY (cod_propriedade)
    REFERENCES propriedade (id_propriedade) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (OIDS=FALSE);
ALTER TABLE maquina OWNER TO postgres;
-- Table: ocorrencia_praga

-- DROP TABLE ocorrencia_praga;

CREATE TABLE ocorrencia_praga
(
  id_ocorrencia_praga integer NOT NULL DEFAULT
nextval('id_ocorrencia_praga_seq'::regclass),
  cod_talhao bigint NOT NULL,
  nome text NOT NULL,
  descricao text,
  solucao text,
  data_ocorrencia date NOT NULL,
  CONSTRAINT ocorrencia_praga_pkey PRIMARY KEY (id_ocorrencia_praga),
  CONSTRAINT ocorrencia_praga_cod_talhao_fkey FOREIGN KEY (cod_talhao)
    REFERENCES talhao (id_talhao) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (OIDS=FALSE);

```

```

ALTER TABLE ocorrencia_praga OWNER TO postgres;
-- Table: operacao

-- DROP TABLE operacao;

CREATE TABLE operacao
(
    id_operacao bigint NOT NULL DEFAULT nextval('id_operacao_seq'::regclass),
    cod_maquina integer NOT NULL,
    data_inicial timestamp without time zone NOT NULL,
    data_final timestamp without time zone NOT NULL,
    horas_trabalhadas double precision NOT NULL,
    descricao text,
    cod_atividade_talhao bigint,
    CONSTRAINT operacao_pkey PRIMARY KEY (id_operacao),
    CONSTRAINT operacao_cod_atividade_talhao_fkey FOREIGN KEY
(cod_atividade_talhao)
    REFERENCES atividade_talhao (id_atividade_talhao) MATCH SIMPLE
    ON UPDATE CASCADE ON DELETE NO ACTION,
    CONSTRAINT operacao_cod_maquina_fkey FOREIGN KEY (cod_maquina)
    REFERENCES maquina (id_maquina) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE operacao OWNER TO postgres;
-- Table: pais

-- DROP TABLE pais;

CREATE TABLE pais
(
    id_pais integer NOT NULL DEFAULT nextval('id_pais_seq'::regclass),
    nome text NOT NULL,
    sigla text, -- ISO-3166-3
    CONSTRAINT pais_pkey PRIMARY KEY (id_pais)
)
WITH (OIDS=FALSE);
ALTER TABLE pais OWNER TO postgres;

```

```

COMMENT ON COLUMN pais.sigla IS 'ISO-3166-3';

-- Table: plantio

-- DROP TABLE plantio;

CREATE TABLE plantio
(
    cod_atividade_talhao bigint NOT NULL,
    cod_variedade integer NOT NULL,
    data_prevista_colheita date NOT NULL,
    quantidade double precision NOT NULL,
    valor_total double precision NOT NULL,
    CONSTRAINT plantio_pkey PRIMARY KEY (cod_atividade_talhao),
    CONSTRAINT plantio_cod_atividade_talhao_fkey FOREIGN KEY
(cod_atividade_talhao)
    REFERENCES atividade_talhao (id_atividade_talhao) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE CASCADE,
    CONSTRAINT plantio_cod_variedade_fkey FOREIGN KEY (cod_variedade)
    REFERENCES variedade (id_variedade) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE plantio OWNER TO postgres;
-- Table: propriedade

-- DROP TABLE propriedade;

CREATE TABLE propriedade
(
    id_propriedade integer NOT NULL DEFAULT
nextval('id_propriedade_seq'::regclass),
    area_total double precision NOT NULL,
    nome text,
    cod_cliente integer NOT NULL,
    cod_endereco integer,
    area_preservacao double precision NOT NULL,
    area_cultivo double precision NOT NULL,

```

```

area_outras double precision NOT NULL,
email text,
CONSTRAINT propriedade_pkey PRIMARY KEY (id_propriedade),
CONSTRAINT propriedade_cod_cliente_fkey FOREIGN KEY (cod_cliente)
    REFERENCES cliente (id_cliente) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE CASCADE,
CONSTRAINT propriedade_cod_endereco_fkey FOREIGN KEY (cod_endereco)
    REFERENCES endereco (id_endereco) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE propriedade OWNER TO postgres;
-- Table: safra

-- DROP TABLE safra;

CREATE TABLE safra
(
    id_safra bigint NOT NULL DEFAULT nextval('id_safra_seq'::regclass),
    ano_safra text NOT NULL,
    data_inicio date NOT NULL,
    data_fim date,
    atual boolean NOT NULL DEFAULT false,
    cod_propriedade integer NOT NULL,
    CONSTRAINT safra_pkey PRIMARY KEY (id_safra),
    CONSTRAINT safra_cod_propriedade_fkey FOREIGN KEY (cod_propriedade)
        REFERENCES propriedade (id_propriedade) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE CASCADE,
    CONSTRAINT safra_ano_safra_key UNIQUE (ano_safra, cod_propriedade)
)
WITH (OIDS=FALSE);
ALTER TABLE safra OWNER TO postgres;
-- Table: senha

-- DROP TABLE senha;

CREATE TABLE senha
(

```

```

id_senha integer NOT NULL DEFAULT nextval('id_senha_seq'::regclass),
senha text NOT NULL,
email text NOT NULL,
CONSTRAINT senha_pkey PRIMARY KEY (id_senha),
CONSTRAINT senha_email_key UNIQUE (email)
)
WITH (OIDS=FALSE);
ALTER TABLE senha OWNER TO postgres;
-- Table: talhao

-- DROP TABLE talhao;

CREATE TABLE talhao
(
nome text NOT NULL,
area double precision NOT NULL,
descricao text,
localizacao text NOT NULL,
terreno proprio boolean NOT NULL DEFAULT true,
id_talhao bigint NOT NULL DEFAULT nextval('id_talhao_seq'::regclass),
cod_safra bigint NOT NULL,
cultivo permanente boolean NOT NULL DEFAULT false,
CONSTRAINT talhao_pkey PRIMARY KEY (id_talhao),
CONSTRAINT talhao_cod_safra_fkey FOREIGN KEY (cod_safra)
REFERENCES safra (id_safra) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE CASCADE
)
WITH (OIDS=FALSE);
ALTER TABLE talhao OWNER TO postgres;
-- Table: usuario

-- DROP TABLE usuario;

CREATE TABLE usuario
(
id_usuario bigint NOT NULL DEFAULT nextval('id_usuario_seq'::regclass),
nome text NOT NULL,
email text NOT NULL,

```

```

administrador boolean NOT NULL DEFAULT false,
cod_propriedade integer NOT NULL,
CONSTRAINT usuario_pkey PRIMARY KEY (id_usuario),
CONSTRAINT usuario_cod_propriedade_fkey FOREIGN KEY (cod_propriedade)
    REFERENCES propriedade (id_propriedade) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE CASCADE,
CONSTRAINT usuario_email_key UNIQUE (email)
)
WITH (oids=FALSE);
ALTER TABLE usuario OWNER TO postgres;
-- Table: variedade

-- DROP TABLE variedade;

CREATE TABLE variedade
(
    id_variedade integer NOT NULL DEFAULT nextval('id_variedade_seq'::regclass),
    cod_cultura integer NOT NULL,
    nome text NOT NULL,
    informacoes text,
    cod_fabricante integer,
CONSTRAINT variedade_pkey PRIMARY KEY (id_variedade),
CONSTRAINT variedade_cod_cultura_fkey FOREIGN KEY (cod_cultura)
    REFERENCES cultura (id_cultura) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE CASCADE,
CONSTRAINT variedade_cod_fabricante_fkey FOREIGN KEY (cod_fabricante)
    REFERENCES fabricante (id_fabricante) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE CASCADE
)
WITH (oids=FALSE);
ALTER TABLE variedade OWNER TO postgres;

```

## XHTML

## Arquivo home.xhtml

```
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:rich="http://richfaces.org/rich"
    template="layout/template.xhtml">

    <ui:define name="body">

        <h1>#{messages.bemVindoAoEfagro}</h1>
        <rich:panel style="width:70%;">
            <h:panelGrid columns="2">
                <s:div styleClass="info">
                    <p><strong>#{messages.tituloHome}</strong>
                    #{messages.subTituloHome}</p>
                    <ul class="bullets">
                        <li>#{messages.funcionalidades1}</li>
                        <li>#{messages.funcionalidades2}</li>
                        <li>#{messages.funcionalidades3}</li>
                        <li>#{messages.funcionalidades4}</li>
                        <li>#{messages.funcionalidades5}</li>
                        <li>#{messages.funcionalidades6}</li>
                        <li>#{messages.funcionalidades7}</li>
                    </ul>
                    <p>
                        <h:form>

                            <s:button view="/cadastroCliente.xhtml"
                                propagation="none" value="#{messages.cadastre}" style="height:50px;font-
                                size:1.2em;" />

                        </h:form>
                    </p>
                </s:div>
            </h:panelGrid>
        </rich:panel>
    </ui:define>
</ui:composition>
```



```
        </s:div>
    </h:panelGrid>
</rich:panel>
</ui:define>
</ui:composition>
```

## Arquivo index.html

```
<html>
<head>
  <meta http-equiv="Refresh" content="0; URL=home.seam">
</head>
</html>
```

## Arquivo inicio.xhtml

```
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
  xmlns:s="http://jboss.com/products/seam/taglib"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:rich="http://richfaces.org/rich"
  template="/layout/templateLogado.xhtml">

  <ui:define name="body">

    <h:form>

      <h:outputLabel value="#{propriedadeHome.instance.nome}"/>

    <p/>
```

```

        <h:outputLabel value="#{messages.anoSafraDaPropriedade}"
rendered="#{propriedadeHome.safraAtual != null}"/>
        <s:link view="/agricultura/safra.xhtml"
value="#{propriedadeHome.safraAtual.anoSafra}"
rendered="#{propriedadeHome.safraAtual != null}">
            <f:param name="idSafra"
value="#{propriedadeHome.safraAtual.idSafra}"/>
        </s:link>
    </h:form>
</ui:define>
</ui:composition>

```

## Arquivo cadastroCliente.xhtml

```

<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:rich="http://richfaces.org/rich"
    template="layout/template.xhtml">

    <ui:define name="body">

        <h:form id="cliente" styleClass="edit">

            <rich:panel id="cadUser" style="width:100%;">
                <f:facet name="header">#{messages.dadosPessoais}</f:facet>

                <s:decorate id="emailField" template="layout/edit.xhtml">
                    <ui:define name="label">#{messages.email}</ui:define>
                    <h:inputText id="email"
                        cols="80"
                        required="true"

```

```

        styleClass="inputMedio"
        value="#{clienteHome.instance.email}"
    </h:inputText>
</s:decorate>

<s:decorate id="nomeField" template="layout/edit.xhtml">
    <ui:define name="label">#{messages.nomeCompleto}</ui:define>
    <h:inputText id="nome"
        cols="80"
        required="true"
        styleClass="inputMedio"
        value="#{clienteHome.instance.nome}"/>
</s:decorate>

<s:decorate id="sexoField" template="layout/edit.xhtml">
    <ui:define name="label">#{messages.sexo}</ui:define>
    <h:selectOneRadio id="chaSex"
value="#{clienteHome.instance.sexo}" required="true" >
        <f:selectItem
itemLabel="#{messages.masculino}" itemValue="M"/>
        <f:selectItem
itemLabel="#{messages.feminino}" itemValue="F"/>
    </h:selectOneRadio>
</s:decorate>

<s:decorate id="telefone" template="layout/edit.xhtml">
    <ui:define name="label">#{messages.telefone}</ui:define>
    <h:inputText id="telefone"
        maxLength="8"
        title="#{messages.numTelefone}"
        required="true"
        size="8"
        value="#{clienteHome.instance.telefone}"/>
    </h:inputText>
</s:decorate>

```

```

<s:decorate id="celular" template="layout/edit.xhtml">
  <ui:define name="label">#{messages.celular}</ui:define>
  <h:inputText id="num_celular"
    maxlength="8"
    title="#{messages.numTelefone}"
    required="true"
    size="8"
    value="#{clienteHome.instance.celular}">
  </h:inputText>
</s:decorate>

<s:decorate id="cpf" template="layout/edit.xhtml">
  <ui:define name="label">#{messages.cpf}</ui:define>
  <h:inputText id="cpfCampo"
    maxlength="11"
    required="true"
    size="8"
    value="#{clienteHome.instance.cpf}">
  </h:inputText>
</s:decorate>

<s:decorate id="dataNascimento" template="layout/edit.xhtml">
  <ui:define name="label">#{messages.dataNasc}</ui:define>
  <rich:calendar id="dataNascCampo"

value="#{clienteHome.instance.dataNascimento}"
    datePattern="#{messages.dataFormato}" />
  </s:decorate>

<a:region id="regCadPais">
  <a:outputPanel id="outcadPais">
    <s:decorate id="paisField" template="layout/edit.xhtml">
      <ui:define
name="label">#{messages.pais}</ui:define>
      <h:selectOneMenu value="#{paisHome.paisIdPais}"
id="selectPais" required="true" >
        <s:selectItems var="_pais"
value="#{paisList.resultList}"

```

```

itemValue="#{_pais.idPais}"

noSelectionLabel="#{messages.seleccioneOPais}"
label="#{_pais.nome}">
    </s:selectItems>
    <a:support id="supPais" event="onchange"
reRender="outEstado"/>
    </h:selectOneMenu>
    </s:decorate>
    </a:outputPanel>
</a:region>

<a:region id="regCadEstado">
<a:outputPanel id="outEstado">
    <s:decorate id="estadoField" template="layout/edit.xhtml">
        <ui:define name="label">#{messages.estado}</ui:define>
        <h:selectOneMenu
value="#{estadoHome.estadoIdEstado}" required="true" id="selectEstado">
            <s:selectItems id="selectEstados" var="_estado"
value="#{paisHome.estados}"

noSelectionLabel="#{messages.seleccioneOEstado}"
label="#{_estado.nome}"

itemValue="#{_estado.idEstado}">
    </s:selectItems>
    <a:support id="supEstado" event="onchange"
reRender="outCidade" />
    </h:selectOneMenu>
    </s:decorate>
    </a:outputPanel>
</a:region>

    <a:outputPanel id="outCidade">
<s:decorate id="cidadeField" template="layout/edit.xhtml">
    <ui:define name="label">#{messages.cidade}</ui:define>

```

```

        <h:selectOneMenu value="#{enderecoHome.instance.cidade}"
required="true" id="selectCidade">
            <s:selectItems var="_cidade" value="#{estadoHome.cidades}"

noSelectionLabel="#{messages.selecioneACidade}"
                    label="#{_cidade.nome}">
                </s:selectItems>
                <s:convertEntity/>
            </h:selectOneMenu>
        </s:decorate>
    </a:outputPanel>

    <s:decorate id="cepField" template="layout/edit.xhtml">
        <ui:define name="label">#{messages.CEP}</ui:define>
        <h:inputText id="CEP"
            maxlength="8"
            cols="8"
            required="true"
            value="#{enderecoHome.instance.codigoPostal}">
        </h:inputText>
    </s:decorate>

    <s:decorate id="ruaField" template="layout/edit.xhtml">
        <ui:define name="label">#{messages.rua}</ui:define>
        <h:inputText id="rua"
            cols="80"
            required="true"
            styleClass="inputGrande"
            value="#{enderecoHome.instance.logradouro}">
        </h:inputText>
    </s:decorate>

    <s:decorate id="bairroField" template="layout/edit.xhtml">
        <ui:define name="label">#{messages.bairro}</ui:define>
        <h:inputText id="bairro"
            cols="80"
            styleClass="inputGrande"
            value="#{enderecoHome.instance.bairro}">
    </s:decorate>

```

```

        </h:inputText>
    </s:decorate>

    <s:decorate id="numeroField" template="layout/edit.xhtml">
        <ui:define name="label">#{messages.numero}</ui:define>
        <h:inputText id="numero"
            cols="5"
            value="#{enderecoHome.instance.numero}">
        </h:inputText>
    </s:decorate>

    <s:decorate id="complementoField" template="layout/edit.xhtml">
        <ui:define name="label">#{messages.complemento}</ui:define>
        <h:inputText id="numero"
            cols="80"
            styleClass="inputGrande"
            value="#{enderecoHome.instance.complemento}">
        </h:inputText>
    </s:decorate>

    <div style="clear:both">
        <span class="required">*</span>
        #{messages.camposObrigatorios}
    </div>

</rich:panel>

<div class="actionButtons">

    <h:commandButton id="save"
        value="#{messages.cadastrar}"
        action="#{clienteHome.inserir()}">

    <s:button id="cancelAdd"
        value="#{messages.cancelar}"
        propagation="end"
        view="/home.xhtml">

```

```

        </div>
    </h:form>

</ui:define>

</ui:composition>

```

## Arquivo Propriedade.xhtml

```

<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:rich="http://richfaces.org/rich"
    template="layout/templateLogado.xhtml">

    <ui:define name="body">

        <h:form id="propriedade" styleClass="edit">

            <rich:panel>
                <f:facet name="header">#{propriedadeHome.instance.nome}</f:facet>

                <s:decorate id="nomeField" template="layout/edit.xhtml">
                    <ui:define name="label">#{messages.nome}</ui:define>
                    <h:inputText id="nome" styleClass="inputGrande"
                        value="#{propriedadeHome.instance.nome}"/>
                </s:decorate>

                <s:decorate id="areaTotalField" template="layout/edit.xhtml">

```



```

        <ui:define name="label">#{messages.areaTotal}</ui:define>
        <h:inputText id="areaTotal"
            required="true"
            value="#{propriedadeHome.instance.areaTotal}"
            <a:support event="onblur" reRender="areaTotalField"
bypassUpdates="true" ajaxSingle="true"/>
        </h:inputText>
    </s:decorate>

    <s:decorate id="areaPreservacaoField"
template="layout/edit.xhtml">
        <ui:define
name="label">#{messages.areaPreservacao}</ui:define>
        <h:inputText id="areaPreservacao"
            required="true"

value="#{propriedadeHome.instance.areaPreservacao}"
            <a:support event="onblur" reRender="areaPreservacaoField"
bypassUpdates="true" ajaxSingle="true"/>
        </h:inputText>
    </s:decorate>

    <s:decorate id="areaCultivoField" template="layout/edit.xhtml">
        <ui:define name="label">#{messages.areaCultivo}</ui:define>
        <h:inputText id="areaCultivo"
            required="true"
            value="#{propriedadeHome.instance.areaCultivo}"
            <a:support event="onblur" reRender="areaCultivoField"
bypassUpdates="true" ajaxSingle="true"/>
        </h:inputText>
    </s:decorate>

    <s:decorate id="areaOutrasField" template="layout/edit.xhtml">
        <ui:define name="label">#{messages.outrasAreas}</ui:define>
        <h:inputText id="areaOutras"
            required="true"

```

```

        value="#{propriedadeHome.instance.areaOutras}">
    <a:support event="onblur" reRender="areaOutrasField"
bypassUpdates="true" ajaxSingle="true"/>
        </h:inputText>
    </s:decorate>

    <s:decorate id="emailField" template="layout/edit.xhtml">
        <ui:define name="label">#{messages.email}</ui:define>
        <h:inputText id="email" styleClass="inputGrande"
            value="#{propriedadeHome.instance.email}"/>
    </s:decorate>

    <div style="clear:both">
        <span class="required">*</span>
            #{messages.camposObrigatorios}
    </div>

</rich:panel>

<div class="actionButtons">

    <h:commandButton id="save"
        value="#{messages.alterar}"
        action="#{propriedadeHome.alterar()}"/>

    <s:button id="cancelEdit"
        value="#{messages.cancelar}"
        propagation="end"
        view="/inicio.xhtml"/>

</div>
</h:form>

</ui:define>

```

```
</ui:composition>
```

## Arquivo login.xhtml

```
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:rich="http://richfaces.org/rich"
    template="layout/template.xhtml">

<ui:define name="body">

    <rich:panel styleClass="edit">
        <f:facet name="header">#{messages.entrar}</f:facet>

        <h:form id="login">

        <p>#{messages.porFavorFacaOLogin}</p>

        <div class="dialog">
            <h:panelGrid columns="2" rowClasses="prop"
columnClasses="name,value">
                <h:outputLabel
for="username">#{messages.email}</h:outputLabel>
                <h:inputText id="username" value="#{credentials.username}"
styleClass="inputMedio" />
                <h:outputLabel
for="password">#{messages.senha}</h:outputLabel>
                <h:inputSecret id="password"
value="#{credentials.password}" styleClass="inputMedio" />
            </h:panelGrid>
        </div>
    </rich:panel>
</ui:define>
```

```

        <h:outputLabel
for="rememberMe">#{messages.lembrarEmail}</h:outputLabel>
        <h:selectBooleanCheckbox id="rememberMe"
                                value="#{identity.rememberMe}"/>
    </h:panelGrid>
</div>
    <div class="actionButtons">
        <h:commandButton value="#{messages.entrar}"
action="#{identity.login}"/>

    </div>

    </h:form>
</rich:panel>

</ui:define>
</ui:composition>

```

## Arquivo error.xhtml

```

<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    template="/layout/template.xhtml">

    <ui:param name="showGlobalMessages" value="false"/>
    <ui:define name="body">

        <h1>Error</h1>
        <p>Something bad happened :-(</p>

```

```

        <h:messages id="errorMessage" styleClass="message"/>

    </ui:define>
</ui:composition>

```

## Arquivo edit.xhtml

```

<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:s="http://jboss.com/products/seam/taglib">

    <div class="prop">

        <s:label styleClass="name #{invalid?'errors':''}">
            <ui:insert name="label"/>
            <s:span styleClass="required" rendered="#{required}">*</s:span>
        </s:label>

        <span class="value #{invalid?'errors':''}">
            <s:validateAll>
                <ui:insert/>
            </s:validateAll>
        </span>

        <span class="error">
            <h:graphicImage value="/img/error.gif" rendered="#{invalid}"
styleClass="errors"/>
            <s:message styleClass="errors"/>
        </span>

    </div>

</ui:composition>

```

## Arquivo menu.xhtml

```
<rich:toolBar
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ui="http://java.sun.com/jsf/facelets"
  xmlns:h="http://java.sun.com/jsf/html"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:s="http://jboss.com/products/seam/taglib"
  xmlns:rich="http://richfaces.org/rich">
  <rich:toolBarGroup>
    <h:outputText value="#{projectName}:"/>
    <s:link view="/inicio.xhtml" value="#{messages.inicio}"
propagation="none" rendered="#{identity.loggedIn}"/>
    <s:link view="/home.xhtml" value="#{messages.inicio}"
propagation="none" rendered="#{not identity.loggedIn}"/>
  </rich:toolBarGroup>

  <!-- @newMenuItem@ -->
  <rich:toolBarGroup location="right">
    <h:outputText value="#{messages.voceEstaLogadoComo}
#{identity.username}" rendered="#{identity.loggedIn}"/>
    <s:link view="/login.xhtml" value="#{messages.entrar}" rendered="#{not
identity.loggedIn}" propagation="none"/>
    <s:link view="/home.xhtml" action="#{identity.logout}"
value="#{messages.sair}" rendered="#{identity.loggedIn}" propagation="none"/>
  </rich:toolBarGroup>
</rich:toolBar>
```

## Arquivo menuEsquerdo.xhtml

```
<ui:composition
  xmlns="http://www.w3.org/1999/xhtml"
```

```

xmlns:ui="http://java.sun.com/jsf/facelets"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:s="http://jboss.com/products/seam/taglib"
xmlns:rich="http://richfaces.org/rich">

<h:form id="form">
<rich:panelMenu mode="ajax" width="250px"
    iconExpandedGroup="disc" iconCollapsedGroup="disc"
    iconExpandedTopGroup="chevronUp" iconGroupTopPosition="right"
    iconCollapsedTopGroup="chevronDown" iconCollapsedTopPosition="right"
expandSingle="false"
    expandMode="none">
    <rich:panelMenuGroup label="#{messages.usuarios}">
        <rich:panelMenuItem label="#{messages.cadastrarUsuarios}"
rendered="#{identity.hasRole('admin')}}"
                                action="goCadastrarUsuario">
        </rich:panelMenuItem>
        <rich:panelMenuItem label="#{messages.listarUsuarios}"
rendered="#{identity.hasRole('admin')}}"
                                action="goListarUsuarios">
        </rich:panelMenuItem>
        <rich:panelMenuItem label="#{messages.alteraSenha}"
                                action="goAlterarSenha">
        </rich:panelMenuItem>
    </rich:panelMenuGroup>
    <rich:panelMenuGroup label="#{messages.maquinas}">
        <rich:panelMenuItem label="#{messages.cadastrarMaquinas}"
rendered="#{identity.hasRole('admin')}}"
                                action="goCadastrarMaquina">
        </rich:panelMenuItem>
        <rich:panelMenuItem label="#{messages.listarMaquinas}"
                                action="goListarMaquinas">
        </rich:panelMenuItem>
        <rich:panelMenuItem label="#{messages.agendarManutencao}"
                                action="goAgendarManutencao">
        </rich:panelMenuItem>
        <rich:panelMenuItem label="#{messages.registrarManutencao}"

```

```

                action="goRegistrarManutencao">
</rich:panelMenuItem>
<rich:panelMenuItem label="#{messages.registrarAbastecimento}"
                action="goRegistrarAbastecimento">

</rich:panelMenuItem>
<rich:panelMenuItem label="#{messages.registrarOperacao}"
                action="goRegistrarOperacao">

</rich:panelMenuItem>
</rich:panelMenuGroup>
<rich:panelMenuGroup label="#{messages.agricultura}">
    <rich:panelMenuItem label="#{messages.definirAnoSafra}"
rendered="#{identity.hasRole('admin')}}"
                action="goDefinirAnoSafra">

</rich:panelMenuItem>
<rich:panelMenuItem label="#{messages.listarAnoSafra}"
                action="goListarAnoSafra">

</rich:panelMenuItem>
<rich:panelMenuItem label="#{messages.listarTalhoes}"
                action="goListarTalhoes">

    </rich:panelMenuItem>
<rich:panelMenuItem label="#{messages.inserirTalhao}"
                action="goInserirTalhao">

    </rich:panelMenuItem>
<rich:panelMenuItem label="#{messages.registrarOcorrenciaPraga}"
                action="goInserirOcorrPraga">

    </rich:panelMenuItem>
<rich:panelMenuGroup
label="#{messages.registrarAtividadeTalhao}">
    <rich:panelMenuItem label="#{messages.registrarAdubacao}"
                action="goInserirAdubacao">

        </rich:panelMenuItem>
    <rich:panelMenuItem
label="#{messages.registrarAplicacaoDefensivo}"

                action="goInserirAplicacaoDefensivo">
        </rich:panelMenuItem>
    <rich:panelMenuItem label="#{messages.registrarPlantio}"
                action="goInserirPlantio">

```



```

        </rich:panelMenuItem>
    <rich:panelMenuItem label="#{messages.registrarColheita}"
        action="goInserirColheita">
        </rich:panelMenuItem>
    <rich:panelMenuItem
label="#{messages.registrarOutrasAtividades}"
        action="goRegistrarOutrasAtividades">
    <s:conversationPropagation type="none"/>
    </rich:panelMenuItem>
</rich:panelMenuGroup>
</rich:panelMenuGroup>
<rich:panelMenuGroup label="#{messages.propriedade}"
rendered="#{identity.hasRole('admin')}">
    <rich:panelMenuItem label="#{messages.alterarDadosPropriedade}"
        action="goAlterarPropriedade">
    </rich:panelMenuItem>
</rich:panelMenuGroup>
<rich:panelMenuGroup label="#{messages.relatorios}">
    <rich:panelMenuGroup label="#{messages.relatoriosMaquinas}">
    <rich:panelMenuItem
label="#{messages.relatorioOperacoesMaquina}"
        action="goOperacoesMaquina">
    </rich:panelMenuItem>
    <rich:panelMenuItem
label="#{messages.relatorioManutencaoMaquina}"
        action="goRelManutencaoMaquina">
    </rich:panelMenuItem>
</rich:panelMenuGroup>
<rich:panelMenuGroup label="#{messages.relatoriosAgricolas}">
    <rich:panelMenuItem label="#{messages.relatorioAnoSafra}"
        action="goRelAnoSafra">
    </rich:panelMenuItem>
    <rich:panelMenuItem label="#{messages.relatorioTalhao}"
        action="goRelTalhao">
    </rich:panelMenuItem>
    <rich:panelMenuItem label="#{messages.relatorioCultura}"
        action="goRelCultura">

```

```

        </rich:panelMenuItem>
        <rich:panelMenuItem label="#{messages.relatorioVariedade}"
                           action="goRelVariedade">
        </rich:panelMenuItem>
        <rich:panelMenuItem label="#{messages.relatorioPragas}"
                           action="goRelPragas">
        </rich:panelMenuItem>
    </rich:panelMenuGroup>
</rich:panelMenuGroup>
</rich:panelMenu>
</h:form>
</ui:composition>

```

## Arquivo moeda.xhtml

```

<ui:composition xmlns="http://www.w3.org/1999/xhtml"
                xmlns:ui="http://java.sun.com/jsf/facelets"
                xmlns:f="http://java.sun.com/jsf/core">

    <f:convertNumber currencySymbol="#{messages.simboloMoeda}"
                    currencyCode="#{messages.codISOMoeda}" type="currency" />

</ui:composition>

```

## Arquivo template.xhtml

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
                "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<f:view contentType="text/html"
        xmlns="http://www.w3.org/1999/xhtml"
        xmlns:ui="http://java.sun.com/jsf/facelets"
        xmlns:h="http://java.sun.com/jsf/html"

```

```

    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:s="http://jboss.com/products/seam/taglib">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>#{messages.titulo}</title>
    <link rel="shortcut icon" href="#{request.contextPath}/favicon.ico"/>
    <a:loadStyle src="resource:///stylesheet/theme.xcss"/>
    <a:loadStyle src="/stylesheet/theme.css"/>
    <ui:insert name="head"/>
</head>

<body>
    <div id="tudo">
        <ui:include src="menu.xhtml">
            <ui:param name="projectName" value="E-fagro"/>
        </ui:include>

        <div class="body">
            <h:messages id="messages" globalOnly="true" styleClass="message"
                errorClass="errormsg" infoClass="infomsg"
warnClass="warnmsg"
                rendered="#{showGlobalMessages != 'false'}"/>
            <ui:insert name="body"/>
        </div>
    </div>
</body>
</html>
</f:view>

```

## Arquivo templateLogado.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```

        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<f:view contentType="text/html"
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:s="http://jboss.com/products/seam/taglib">
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>#{messages.titulo}</title>
    <link rel="shortcut icon" href="#{request.contextPath}/favicon.ico"/>
    <a:loadStyle src="resource:///stylesheet/theme.xcss"/>
    <a:loadStyle src="/stylesheet/theme.css"/>
    <ui:insert name="head"/>
</head>

<body>
    <div id="tudo">

        <ui:include src="menu.xhtml">
            <ui:param name="projectName" value="E-fagro"/>
        </ui:include>

        <div id="container">
            <div id="menuEsquerdo">
                <ui:include src="menuEsquerdo.xhtml">
                </ui:include>
            </div>
            <div id="conteudo" class="body">
                <h:messages id="messages" globalOnly="true"
styleClass="message"
                    errorClass="errormsg" infoClass="infomsg"
warnClass="warnmsg"
                    rendered="#{showGlobalMessages != 'false'}/>

```

```

        <ui:insert name="body"/>
    </div>
</div>
</div>
</body>
</html>
</f:view>

```

## Arquivo operacao.xhtml

```

<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:rich="http://richfaces.org/rich">

    <rich:panel>
        <f:facet name="header">#{messages.maquinasUtilizadas}</f:facet>

        <s:decorate id="maquina" template="/layout/edit.xhtml">
            <ui:define name="label">#{messages.maquina}</ui:define>
            <h:selectOneMenu value="#{maquinaHome.maquinaIdMaquina}"
                id="selectMaquina"
                required="true">
                <s:selectItems var="_maq"
                    value="#{propriedadeHome.maquinas}"

                    noSelectionLabel="#{messages.seleccioneAMaquina}"
                    itemValue="#{_maq.idMaquina}"
                    label="#{_maq.nome} -
                    #{_maq.modelo}">
                </s:selectItems>
            </h:selectOneMenu>
        </s:decorate>

```

```

<s:decorate id="dataInicialField" template="/layout/edit.xhtml">
  <ui:define name="label">#{messages.dataInicio}</ui:define>
  <rich:calendar id="dataInicial"
    required="true"
    value="#{operacaoHome.instance.dataInicial}"
datePattern="#{messages.dataFormato}"/>
</s:decorate>

<s:decorate id="dataFinalField" template="/layout/edit.xhtml">
  <ui:define name="label">#{messages.dataFim}</ui:define>
  <rich:calendar id="dataFinal"
    required="true"
    value="#{operacaoHome.instance.dataFinal}"
datePattern="#{messages.dataFormato}"/>
</s:decorate>

<s:decorate id="horasTrabalhadasField"
template="/layout/edit.xhtml">
  <ui:define
name="label">#{messages.horasTrabalhadas}</ui:define>
  <h:inputText id="horasTrabalhadas"
    required="true"
    value="#{operacaoHome.instance.horasTrabalhadas}">
    <a:support event="onblur" reRender="horasTrabalhadasField"
bypassUpdates="true" ajaxSingle="true"/>
  </h:inputText>
</s:decorate>

<div style="clear:both"/>
</rich:panel>

</ui:composition>

```

## Arquivo atividade.xhtml

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:rich="http://richfaces.org/rich">

    <s:decorate id="talhao" template="/layout/edit.xhtml">
    <ui:define name="label">#{messages.talhao}</ui:define>
    <h:selectOneMenu value="#{talhaoHome.talhaoIdTalhao}"
id="selectTalhao"
                                required="true">
    <s:selectItems var="_talhao"
value="#{propriedadeHome.safraAtual.talhaos}"

    noSelectionLabel="#{messages.selecioneOTalhao}"
                                itemValue="#{_talhao.idTalhao}"
                                label="#{_talhao.nome}">
    </s:selectItems>
    </h:selectOneMenu>
    </s:decorate>

    <s:decorate id="dataAtividadeField" template="/layout/edit.xhtml">
    <ui:define name="label">#{messages.dataAtividade}</ui:define>
    <rich:calendar id="dataAtividade"
                                required="true"

value="#{atividadeTalhaoHome.instance.dataAtividade}"
datePattern="#{messages.dataFormato}" />
    </s:decorate>

    <s:decorate id="observacaoField" template="/layout/edit.xhtml">
```

```

        <ui:define name="label">#{messages.observacao}</ui:define>
        <h:inputTextarea id="observacao" styleClass="inputGrande"
            rows="3"
            value="#{atividadeTalhaoHome.instance.observacao}"/>
        </s:decorate>
    </ui:composition>

```

## Arquivo cadastroAdubacao.xhtml

```

<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:rich="http://richfaces.org/rich"
    template="/layout/templateLogado.xhtml">
    <ui:define name="body">
        <h:form id="adubacao" styleClass="edit">
            <rich:panel>
                <f:facet name="header">#{messages.registrarAdubacao}</f:facet>
                <ui:include src="/agricultura/fragmento/atividade.xhtml" />
                <s:decorate id="fertilizante" template="/layout/edit.xhtml">
                    <ui:define name="label">#{messages.fertilizante}</ui:define>
                    <h:selectOneMenu
            value="#{fertilizanteHome.fertilizanteIdFertilizante}" id="selectFertilizante"
                required="true">

```



```

        <s:selectItems var="_fertilizante"
value="#{fertilizanteList.resultList}"

noSelectionLabel="#{messages.seleccioneOFertilizante}"

itemValue="#{_fertilizante.idFertilizante}"
label="#{_fertilizante.nome} -
#{_fertilizante.propriedades}">
        </s:selectItems>
    </h:selectOneMenu>
</s:decorate>

<s:decorate id="quantidadeField" template="/layout/edit.xhtml">
    <ui:define name="label">#{messages.quantidade}</ui:define>
    <h:inputText id="quantidade"
        required="true"
        value="#{adubacaoHome.instance.quantidade}">
    <a:support event="onblur" reRender="quantidadeField"
bypassUpdates="true" ajaxSingle="true"/>
    </h:inputText>
</s:decorate>

<s:decorate id="valorTotalField" template="/layout/edit.xhtml">
    <ui:define name="label">#{messages.valorTotal}</ui:define>
    <h:inputText id="valorTotal"
        required="true"
        value="#{adubacaoHome.instance.valorTotal}">
    <a:support event="onblur" reRender="valorTotalField"
bypassUpdates="true" ajaxSingle="true"/>
    </h:inputText>
</s:decorate>

    <rich:spacer height="10px"/>

<div style="clear:both"/>

<ui:include src="/agricultura/fragmento/operacao.xhtml" />

```

```

        <div style="clear:both">
            <span class="required">*</span>
            #{messages.camposObrigatorios}
        </div>

</rich:panel>

<div class="actionButtons">

    <h:commandButton id="save"
        value="#{messages.cadastrar}"
        action="#{adubacaoHome.inserir()}" />

    <s:button id="cancelAdd"
        value="#{messages.cancelar}"
        propagation="end"
        view="/inicio.xhtml" />

</div>
</h:form>
</ui:define>

</ui:composition>

```

## Arquivo cadastraApliDefensivo.xhtml

```

<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"

```

```

xmlns:a="http://richfaces.org/a4j"
xmlns:rich="http://richfaces.org/rich"
template="/layout/templateLogado.xhtml">

<ui:define name="body">

    <h:form id="aplicacaoDefensivo" styleClass="edit">

        <rich:panel>
            <f:facet
name="header">#{messages.registrarAplicacaoDefensivo}</f:facet>

                <ui:include src="/agricultura/fragmento/atividade.xhtml" />

                <s:decorate id="defensivo" template="/layout/edit.xhtml">
                    <ui:define name="label">#{messages.defensivo}</ui:define>
                    <h:selectOneMenu
value="#{defensivoHome.defensivoIdDefensivo}" id="selectDefensivo"
                        required="true">
                        <s:selectItems var="_defensivo"
value="#{defensivoList.resultList}"

                            noSelectionLabel="#{messages.selecioneODefensivo}"

                            itemValue="#{_defensivo.idDefensivo}"
                                label="#{_defensivo.nome} -
#{_defensivo.fabricante.nomeFabricante}">
                                </s:selectItems>
                            </h:selectOneMenu>
                        </s:decorate>

                <s:decorate id="horaField" template="/layout/edit.xhtml">
                    <ui:define name="label">#{messages.hora}</ui:define>
                    <h:inputText id="hora" styleClass="inputPequeno"
                        required="true"

```

```

        value="#{aplicacaoDefensivoHome.instance.hora}">
        <s:convertDateTime type="time"/>
        <a:support event="onblur" reRender="horaField"
bypassUpdates="true" ajaxSingle="true"/>
        </h:inputText>
    </s:decorate>

    <s:decorate id="quantidadeField" template="/layout/edit.xhtml">
        <ui:define name="label">#{messages.quantidade}</ui:define>
        <h:inputText id="quantidade"
            required="true"

value="#{aplicacaoDefensivoHome.instance.quantidade}">
            <a:support event="onblur" reRender="quantidadeField"
bypassUpdates="true" ajaxSingle="true"/>
            </h:inputText>
        </s:decorate>

    <s:decorate id="carenciaField" template="/layout/edit.xhtml">
        <ui:define
name="label">#{messages.dataFinalCarencia}</ui:define>
        <rich:calendar id="carencia"
            required="true"
            value="#{aplicacaoDefensivoHome.instance.carencia}"
datePattern="#{messages.dataFormato}" />
        </s:decorate>

    <s:decorate id="umidadeField" template="/layout/edit.xhtml">
        <ui:define
name="label">#{messages.umidadeRelativa}</ui:define>
        <h:inputText id="umidade"
            value="#{aplicacaoDefensivoHome.instance.umidade}">
            <a:support event="onblur" reRender="umidadeField"
bypassUpdates="true" ajaxSingle="true"/>
        </h:inputText>

```

```

</s:decorate>

<s:decorate id="temperaturaField" template="/layout/edit.xhtml">
  <ui:define name="label">#{messages.temperatura}</ui:define>
  <h:inputText id="temperatura"

value="#{aplicacaoDefensivoHome.instance.temperatura}">
    <a:support event="onblur" reRender="temperaturaField"
bypassUpdates="true" ajaxSingle="true"/>
    </h:inputText>
</s:decorate>

<s:decorate id="valorTotalField" template="/layout/edit.xhtml">
  <ui:define name="label">#{messages.valorTotal}</ui:define>
  <h:inputText id="valorTotal"
    required="true"

value="#{aplicacaoDefensivoHome.instance.valorTotal}">
    <a:support event="onblur" reRender="valorTotalField"
bypassUpdates="true" ajaxSingle="true"/>
    </h:inputText>
</s:decorate>

<rich:spacer height="10px"/>

<div style="clear:both"/>

<ui:include src="/agricultura/fragmento/operacao.xhtml" />

<div style="clear:both">
  <span class="required">*</span>
  #{messages.camposObrigatorios}
</div>

</rich:panel>

```

```

<div class="actionButtons">

    <h:commandButton id="save"
        value="#{messages.cadastrar}"
        action="#{aplicacaoDefensivoHome.inserir()}" />

    <s:button id="cancelAdd"
        value="#{messages.cancelar}"
        propagation="end"
        view="/inicio.xhtml" />

</div>
</h:form>

</ui:define>

</ui:composition>

```

## Arquivo cadastraColheita.xhtml

```

<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:rich="http://richfaces.org/rich"
    template="/layout/templateLogado.xhtml">

    <ui:define name="body">

        <h:form id="colheita" styleClass="edit">

            <rich:panel>

```

```

<f:facet name="header">#{messages.registrarColheita}</f:facet>

    <ui:include src="/agricultura/fragmento/atividade.xhtml" />

<s:decorate id="quantidadeField" template="/layout/edit.xhtml">
    <ui:define name="label">#{messages.quantidade}</ui:define>
    <h:inputText id="quantidade" styleClass="inputPequeno"
        required="true"
        value="#{colheitaHome.instance.quantidade}">
        <a:support event="onblur" reRender="quantidadeField"
bypassUpdates="true" ajaxSingle="true"/>
    </h:inputText>
</s:decorate>

<s:decorate id="valorTotalField" template="/layout/edit.xhtml">
    <ui:define name="label">#{messages.valorTotal}</ui:define>
    <h:inputText id="valorTotal" styleClass="inputPequeno"
        required="true"
        value="#{colheitaHome.instance.valorTotal}">
        <a:support event="onblur" reRender="valorTotalField"
bypassUpdates="true" ajaxSingle="true"/>
    </h:inputText>
</s:decorate>

    <rich:spacer height="10px"/>

    <div style="clear:both"/>

    <ui:include src="/agricultura/fragmento/operacao.xhtml" />

<div style="clear:both">
    <span class="required">*</span>
    #{messages.camposObrigatorios}
</div>

</rich:panel>

```

```

<div class="actionButtons">

    <h:commandButton id="save"
        value="#{messages.cadastrar}"
        action="#{colheitaHome.inserir()}" />

    <s:button id="cancelAdd"
        value="#{messages.cancelar}"
        propagation="end"
        view="/inicio.xhtml" />

</div>
</h:form>
</ui:define>

</ui:composition>

```

## Arquivo cadastraPlantio.xhtml

```

<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:rich="http://richfaces.org/rich"
    template="/layout/templateLogado.xhtml">

<ui:define name="body">

    <h:form id="plantio" styleClass="edit">

        <rich:panel>

```



```

<f:facet name="header">#{messages.registrarPlantio}</f:facet>

    <ui:include src="/agricultura/fragmento/atividade.xhtml" />

        <a:region id="regSelCultura">
            <a:outputPanel id="outSelcultura">
                <s:decorate id="culturaField" template="/layout/edit.xhtml">
                    <ui:define
name="label">#{messages.cultura}</ui:define>
                    <h:selectOneMenu
value="#{culturaHome.culturaIdCultura}" id="selectCultura" required="true" >
                        <s:selectItems var="_cultura"
value="#{culturaList.resultList}"

itemValue="#{_cultura.idCultura}"

noSelectionLabel="#{messages.selecioneACultura}"

label="#{_cultura.nome}">
                            </s:selectItems>
                            <a:support id="supCultura" event="onchange"
reRender="outVariedade"/>
                        </h:selectOneMenu>
                    </s:decorate>
                </a:outputPanel>
            </a:region>

            <a:outputPanel id="outVariedade">
                <s:decorate id="variedadeField" template="/layout/edit.xhtml">
                    <ui:define
name="label">#{messages.variedade}</ui:define>
                    <h:selectOneMenu
value="#{variedadeHome.variedadeIdVariedade}" required="true"
id="selectVariedade">
                        <s:selectItems var="_variedade"
value="#{culturaHome.variedades}"

```

```

noSelectionLabel="#{messages.seleccioneAVariiedade}"

label="#{_variedade.nome} - #{_variedade.fabricante.nomeFabricante}"

itemValue="#{_variedade.idVariiedade}"
        </s:selectItems>
    </h:selectOneMenu>
</s:decorate>
</a:outputPanel>

    <s:decorate id="dataPrevistaColheitaField"
template="/layout/edit.xhtml">
    <ui:define
name="label">#{messages.dataPrevistaColheita}</ui:define>
    <rich:calendar id="dataPrevistaColheita"
        required="true"
        value="#{plantioHome.instance.dataPrevistaColheita}"
datePattern="#{messages.dataFormato}" />
</s:decorate>

<s:decorate id="quantidadeField" template="/layout/edit.xhtml">
    <ui:define name="label">#{messages.quantidade}</ui:define>
    <h:inputText id="quantidade" styleClass="inputPequeno"
        required="true"
        value="#{plantioHome.instance.quantidade}"
    <a:support event="onblur" reRender="quantidadeField"
bypassUpdates="true" ajaxSingle="true"/>
    </h:inputText>
</s:decorate>

<s:decorate id="valorTotalField" template="/layout/edit.xhtml">
    <ui:define name="label">#{messages.valorTotal}</ui:define>
    <h:inputText id="valorTotal"
        required="true"

```

```

        value="#{plantioHome.instance.valorTotal}">
        <a:support event="onblur" reRender="valorTotalField"
bypassUpdates="true" ajaxSingle="true"/>
        </h:inputText>
    </s:decorate>

    <rich:spacer height="10px"/>

    <div style="clear:both"/>

    <ui:include src="/agricultura/fragmento/operacao.xhtml" />

    <div style="clear:both">
        <span class="required">*</span>
        #{messages.camposObrigatorios}
    </div>

</rich:panel>

<div class="actionButtons">

    <h:commandButton id="save"
        value="#{messages.cadastrar}"
        action="#{plantioHome.inserir()}" />

    <s:button id="cancelAdd"
        value="#{messages.cancelar}"
        propagation="end"
        view="/inicio.xhtml" />

</div>
</h:form>

</ui:define>

</ui:composition>

```

## Arquivo cadastraSafra.xhtml

```
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:rich="http://richfaces.org/rich"
    template="/layout/templateLogado.xhtml">

<ui:define name="body">

    <h:form id="safra" styleClass="edit">

        <rich:panel>
            <f:facet name="header">#{messages.definirSafra}</f:facet>

            <s:decorate id="anoSafraField" template="/layout/edit.xhtml">
                <ui:define name="label">#{messages.anoSafra}</ui:define>
                <h:inputText id="anoSafra"
                    required="true"
                    value="#{safraHome.instance.anoSafra}"/>
                <rich:toolTip value="#{messages.exemploAnoSafra}"
for="anoSafraField"/>
            </s:decorate>

            <s:decorate id="dataInicioField" template="/layout/edit.xhtml">
                <ui:define name="label">#{messages.dataInicio}</ui:define>
                <rich:calendar id="dataInicio"
                    required="true"
                    value="#{safraHome.instance.dataInicio}"
datePattern="#{messages.dataFormato}" />
            </s:decorate>


```

```

<s:decorate id="dataFimField" template="/layout/edit.xhtml">
  <ui:define name="label">#{messages.dataFim}</ui:define>
  <rich:calendar id="dataFim"
    value="#{safraHome.instance.dataFim}"
datePattern="#{messages.dataFormato}" />
</s:decorate>

<div style="clear:both">
  <span class="required">*</span>
  #{messages.camposObrigatorios}
</div>

</rich:panel>

<div class="actionButtons">

  <h:commandButton id="save"
    value="#{messages.cadastrar}"
    action="#{safraHome.inserir()}" />

  <s:button id="cancelAdd"
    value="#{messages.cancelar}"
    propagation="end"
    view="/inicio.xhtml" />

</div>
</h:form>

</ui:define>

</ui:composition>

```

### Arquivo cadastrTalhao.xhtml

```

<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"

```

```

xmlns:s="http://jboss.com/products/seam/taglib"
xmlns:ui="http://java.sun.com/jsf/facelets"
xmlns:f="http://java.sun.com/jsf/core"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:a="http://richfaces.org/a4j"
xmlns:rich="http://richfaces.org/rich"
template="/layout/templateLogado.xhtml">

<ui:define name="body">

    <h:form id="talhao" styleClass="edit">

        <rich:panel>
            <f:facet name="header">#{messages.cadastraTalhao}</f:facet>

            <s:decorate id="maquina" template="/layout/edit.xhtml">
                <ui:define name="label">#{messages.anoSafra}</ui:define>
                <h:selectOneMenu value="#{safraHome.safraIdSafra}"
id="selectSafra"
                    required="true">
                    <s:selectItems var="_safra"
value="#{propriedadeHome.safras}"
noSelectionLabel="#{messages.seleccioneOAnoSafra}"
                    itemValue="#{_safra.idSafra}"
                    label="#{_safra.anoSafra}"
                </s:selectItems>
                </h:selectOneMenu>
            </s:decorate>

            <s:decorate id="nomeField" template="/layout/edit.xhtml">
                <ui:define name="label">#{messages.nome}</ui:define>
                <h:inputText id="nome" styleClass="inputMedio"
                    required="true"
                    value="#{talhaoHome.instance.nome}"/>
            </s:decorate>
        </rich:panel>
    </h:form>
</ui:define>

```

```

</s:decorate>

<s:decorate id="areaField" template="/layout/edit.xhtml">
  <ui:define name="label">#{messages.area}</ui:define>
  <h:inputText id="area"
    required="true"
    value="#{talhaoHome.instance.area}">
  <a:support event="onblur" reRender="areaField"
bypassUpdates="true" ajaxSingle="true"/>
  </h:inputText>
</s:decorate>

<s:decorate id="descricaoField" template="/layout/edit.xhtml">
  <ui:define name="label">#{messages.descricao}</ui:define>
  <h:inputTextarea id="descricao" styleClass="inputGrande"
    rows="3"
    value="#{talhaoHome.instance.descricao}">
</s:decorate>

<s:decorate id="localizacaoField" template="/layout/edit.xhtml">
  <ui:define name="label">#{messages.localizacao}</ui:define>
  <h:inputTextarea id="localizacao" styleClass="inputGrande"
    rows="3"
    required="true"
    value="#{talhaoHome.instance.localizacao}">
</s:decorate>

<s:decorate id="terrenoProprioField"
template="/layout/edit.xhtml">
  <ui:define name="label">#{messages.terrenoProprio}</ui:define>
  <h:selectBooleanCheckbox id="terrenoProprio"
    required="true"

value="#{talhaoHome.instance.terrenoProprio}">

```

```

        </s:decorate>

        <s:decorate id="cultivoPermanenteField"
template="/layout/edit.xhtml">
            <ui:define
name="label">#{messages.cultivoPermanente}</ui:define>
            <h:selectBooleanCheckbox id="cultivoPermanente"
                required="true"

value="#{talhaoHome.instance.cultivoPermanente}"/>
            </s:decorate>

            <div style="clear:both">
                <span class="required">*</span>
                #{messages.camposObrigatorios}
            </div>

        </rich:panel>

        <div class="actionButtons">

            <h:commandButton id="save"
                value="#{messages.cadastrar}"
                action="#{talhaoHome.inserir()}"/>

            <s:button id="cancelAdd"
                value="#{messages.cancelar}"
                propagation="end"
                view="/inicio.xhtml"/>

        </div>
    </h:form>

</ui:define>

```



```
</ui:composition>
```

## Arquivo cadOcorrenciaPraga.xhtml

```
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:rich="http://richfaces.org/rich"
    template="/layout/templateLogado.xhtml">

    <ui:define name="body">

        <h:form id="ocorrenciaPraga" styleClass="edit">

            <rich:panel>
                <f:facet name="header">#{messages.registrarOcorrenciaPraga}
                #{messages.noTalhao} #{talhaoHome.instance.nome} </f:facet>

                <s:decorate id="talhao" template="/layout/edit.xhtml">
                <ui:define name="label">#{messages.talhao}</ui:define>
                <h:selectOneMenu value="#{talhaoHome.talhaoIdTalhao}"
                id="selectTalhao"
                    required="true">
                    <s:selectItems var="_talhao"
                value="#{propriedadeHome.safraAtual.talhaos}"

                noSelectionLabel="#{messages.seleccioneOTalhao}"
                    itemValue="#{_talhao.idTalhao}"
                    label="#{_talhao.nome}">
                </s:selectItems>
            </rich:panel>
        </h:form>
    </ui:define>
</ui:composition>
```

```

        </h:selectOneMenu>
    </s:decorate>

    <s:decorate id="nomeField" template="/layout/edit.xhtml">
        <ui:define name="label">#{messages.nome}</ui:define>
        <h:inputText id="nome"
            cols="80"
            required="true"
            value="#{ocorrenciaPragaHome.instance.nome}"/>
    </s:decorate>

    <s:decorate id="descricaoField" template="/layout/edit.xhtml">
        <ui:define name="label">#{messages.descricao}</ui:define>
        <h:inputTextarea id="descricao"
            cols="80"
            rows="3"
            value="#{ocorrenciaPragaHome.instance.descricao}"/>
    </s:decorate>

    <s:decorate id="solcaoField" template="/layout/edit.xhtml">
        <ui:define name="label">#{messages.solucao}</ui:define>
        <h:inputTextarea id="solucao"
            cols="80"
            rows="3"
            value="#{ocorrenciaPragaHome.instance.solucao}"/>
    </s:decorate>

    <s:decorate id="dataOcorrenciaField"
template="/layout/edit.xhtml">
        <ui:define name="label">#{messages.dataOcorrencia}</ui:define>
        <rich:calendar id="dataOcorrencia"
            required="true"

```

```

value="#{ocorrenciaPragaHome.instance.dataOcorrencia}"
datePattern="#{messages.dataFormato}" />
    </s:decorate>

    <div style="clear:both">
        <span class="required">*</span>
        #{messages.camposObrigatorios}
    </div>

</rich:panel>

<div class="actionButtons">

    <h:commandButton id="save"
        value="#{messages.cadastrar}"
        action="#{ocorrenciaPragaHome.inserir()}" />

    <s:button id="cancelAdd"
        value="#{messages.cancelar}"
        propagation="end"
        view="/inicio.xhtml" />

</div>
</h:form>

</ui:define>

</ui:composition>

```

## Arquivo cadOutrasAtividades.xhtml

```

<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"

```

```
xmlns:ui="http://java.sun.com/jsf/facelets"  
xmlns:f="http://java.sun.com/jsf/core"  
xmlns:h="http://java.sun.com/jsf/html"  
xmlns:a="http://richfaces.org/a4j"  
xmlns:rich="http://richfaces.org/rich"  
template="/layout/templateLogado.xhtml">
```

```
<ui:define name="body">
```

```
<h:form id="atividadeTalhao" styleClass="edit">
```

```
<rich:panel>
```

```
<f:facet name="header">#{messages.registrarAtividade}</f:facet>
```

```
<ui:include src="/agricultura/fragmento/atividade.xhtml" />
```

```
<rich:spacer height="5px"/>
```

```
<div style="clear:both"/>
```

```
<ui:include src="/agricultura/fragmento/operacao.xhtml" />
```

```
<rich:spacer height="5px"/>
```

```
<div style="clear:both">
```

```
<span class="required">*</span>
```

```
#{messages.camposObrigatorios}
```

```
</div>
```

```
</rich:panel>
```

```
<div class="actionButtons">
```

```
<h:commandButton id="save"
```

```
value="#{messages.cadastrar}"
```

```
action="#{atividadeTalhaoHome.inserir()}" />
```

```

        <s:button id="cancelAdd"
            value="#{messages.cancelar}"
            propagation="end"
            view="/inicio.xhtml"/>

    </div>
</h:form>
</ui:define>

</ui:composition>

```

## Arquivo listarSafras.xhtml

```

<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:rich="http://richfaces.org/rich"
    template="/layout/templateLogado.xhtml">

    <ui:define name="body">

        <h:form id="listMaq">

            <h:outputLabel value="#{messages.nenhumaSafraCadastrada}"
                rendered="#{empty propriedadeHome.safras}"/>

            <div class="actionButtons">
                <s:button id="cancelAdd" rendered="#{empty
                    propriedadeHome.safras}"
                    value="#{messages.cadastrar}"

```

```

        propagation="end"
        view="/agricultura/cadastraSafra.xhtml"/>

</div>

<rich:dataTable var="_safra" value="#{propriedadeHome.safras}"
rendered="#{not empty propriedadeHome.safras}">
    <f:facet name="header">
        <rich:columnGroup>
            <rich:column>
                <h:outputLabel value="#{messages.anoSafra}" />
            </rich:column>
            <rich:column>
                <h:outputLabel value="#{messages.dataInicio}" />
            </rich:column>
            <rich:column>
                <h:outputLabel value="#{messages.dataFim}" />
            </rich:column>
        </rich:columnGroup>
    </f:facet>
    <rich:columnGroup>
        <rich:column>
            <s:link value="#{_safra.anoSafra}"
view="/agricultura/safra.xhtml"
                <rich:tooltip
value="#{messages.detalhes}"/>
                <f:param name="idSafra"
value="#{_safra.idSafra}"/>
            </s:link>
        </rich:column>
        <rich:column>
            <h:outputText value="#{_safra.dataInicio}" />
        </rich:column>
        <rich:column>
            <h:outputText value="#{_safra.dataFim}"/>
        </rich:column>
    </rich:columnGroup>
</rich:dataTable>

```

```

    </h:form>

</ui:define>

</ui:composition>

```

## Arquivo listarTalhoes.xhtml

```

<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:rich="http://richfaces.org/rich"
    template="/layout/templateLogado.xhtml">

<ui:define name="body">

    <h:form id="formListTalhoes">

        <rich:panel>
            <f:facet
name="header">#{messages.seleccioneASafraParaVerOsTalhoes}</f:facet>

            <a:region id="regAnoSafra">
                <a:outputPanel id="outAnoSafra">
                    <s:decorate id="anoSafra" template="/layout/display.xhtml">
                        <ui:define name="label">#{messages.anoSafra}</ui:define>
                            <h:selectOneMenu value="#{safraHome.safraIdSafra}"
id="selectSafra" required="true">
                                <s:selectItems id="selectItensSafras"
                                    var="_safra"
value="#{propriedadeHome.safras}"

```

```

noSelectionLabel="#{messages.seleccioneASafra}"
                                itemValue="#{_safra.idSafra}"
                                label="#{_safra.anoSafra}"
        </s:selectItems>
        <a:support id="supAnoSafra" event="onchange"
reRender="infAnoSafra"/>
        </h:selectOneMenu>
</s:decorate>
</a:outputPanel>
    </a:region>

<p/>

<div style="clear:both"/>

    <rich:panel id="infAnoSafra">
        <f:facet name="header">#{messages.talhoesDaSafra}
#{safraHome.instance.anoSafra}</f:facet>

        <p/>

        <h:outputLabel id="msgNenhumTalhao"
value="#{messages.nenhumaTalhaoCadastrado}" rendered="#{empty
safraHome.talhaos}"/>

        <p/>

        <rich:dataTable id="listaTalhoes" var="_talhao"
value="#{safraHome.talhaos}" rendered="#{not empty safraHome.talhaos}">
            <f:facet name="header">
                <rich:columnGroup>
                    <rich:column>
                        <h:outputLabel value="#{messages.nome}" />
                    </rich:column>
                    <rich:column>

```



```

        <h:outputLabel value="#{messages.area}" />
    </rich:column>
    <rich:column>
        <h:outputLabel value="#{messages.localizacao}" />
    </rich:column>
</rich:columnGroup>
</f:facet>
    <rich:columnGroup>
        <rich:column>
            <s:link value="#{_talhao.nome}"
view="/agricultura/talhao.xhtml">
                <rich:toolTip
value="#{messages.detalhes}"/>
                <f:param name="idTalhao"
value="#{_talhao.idTalhao}"/>
            </s:link>
        </rich:column>
        <rich:column>
            <h:outputText value="#{_talhao.area}" />
        </rich:column>
        <rich:column>
            <h:outputText value="#{_talhao.localizacao}" />
        </rich:column>
    </rich:columnGroup>
</rich:dataTable>

</rich:panel>

</rich:panel>

</h:form>

</ui:define>

</ui:composition>

```

## Arquivo safra.xhtml

```
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:rich="http://richfaces.org/rich"
    template="/layout/templateLogado.xhtml">

<ui:define name="body">

    <h:form id="safra" styleClass="edit">

        <rich:panel>
            <f:facet name="header">#{messages.anoSafra}
            #{safraHome.instance.anoSafra}</f:facet>

            <rich:panel>
                <f:facet name="header">#{messages.talhoesDaSafra}
                #{safraHome.instance.anoSafra}</f:facet>

                <s:button view="/agricultura/cadastraTalhao.xhtml"
                value="#{messages.cadastraTalhao}"/>

            <p/>

            <rich:dataTable var="_talhao"
            value="#{safraHome.talhaos}" rendered="#{not empty safraHome.talhaos}">
                <f:facet name="header">
                    <rich:columnGroup>
                        <rich:column>
                            <h:outputLabel
                            value="#{messages.nome}" />
                        </rich:column>
                    </rich:columnGroup>
                </f:facet>
            </rich:dataTable>
        </h:form>
    </ui:define>
</ui:composition>
```

```

        <rich:column>
            <h:outputLabel
value="#{messages.area}" />
        </rich:column>
        <rich:column>
            <h:outputLabel
value="#{messages.localizacao}" />
        </rich:column>
    </rich:columnGroup>
</f:facet>
    <rich:columnGroup>
        <rich:column>
            <s:link value="#{_talhao.nome}"
view="/agricultura/talhao.xhtml"
            <f:param name="idTalhao"
value="#{_talhao.idTalhao}"/>
            </s:link>
        </rich:column>
        <rich:column>
            <h:outputText
value="#{_talhao.area}" />
        </rich:column>
        <rich:column>
            <h:outputText
value="#{_talhao.localizacao}" />
        </rich:column>
    </rich:columnGroup>
</rich:dataTable>

</rich:panel>

<rich:spacer height="5px"/>

<div style="clear:both"/>

</rich:panel>

</h:form>

```

```
</ui:define>
```

```
</ui:composition>
```

## Arquivo talhao.xhtml

```
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:s="http://jboss.com/products/seam/taglib"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    xmlns:f="http://java.sun.com/jsf/core"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:a="http://richfaces.org/a4j"
    xmlns:rich="http://richfaces.org/rich"
    template="/layout/templateLogado.xhtml">

<ui:define name="body">

    <h:form id="talhaoForm" styleClass="edit">

        <rich:panel>
            <f:facet name="header">#{messages.talhaodP}
            #{talhaoHome.instance.nome}</f:facet>

            <p/>
            <h:outputLabel value="#{messages.saldoDoTalhaoEh}" />
            <h:outputText value="#{talhaoHome.saldoTalhao}">
                <ui:include src="/layout/moeda.xhtml" />
            </h:outputText>
            <p/>

            <div style="clear:both"/>

            <rich:panel>
```

```

        <f:facet
name="header">#{messages.atividadesTalhao}</f:facet>

        <p/>

        <rich:dataTable var="_atividade"
value="#{talhaoHome.atividadeTalhaos}">
            <f:facet name="header">
                <rich:columnGroup>
                    <rich:column colspan="4">
                        <h:outputText
value="#{messages.adubacoesEfetuadas}" />
                    </rich:column>
                    <rich:column breakBefore="true">
                        <h:outputLabel
value="#{messages.data}" />
                    </rich:column>
                    <rich:column>
                        <h:outputLabel
value="#{messages.fertilizante}" />
                    </rich:column>
                    <rich:column>
                        <h:outputLabel
value="#{messages.quantidade}" />
                    </rich:column>
                    <rich:column>
                        <h:outputLabel
value="#{messages.valorTotal}" />
                    </rich:column>
                </rich:columnGroup>
            </f:facet>
            <rich:columnGroup rendered="#{null !=
_atividade.adubacao}">
                <rich:column>
                    <h:outputText
value="#{_atividade.dataAtividade}" />

```

```

                                <f:convertDateTime
pattern="#{messages.dataFormato}" />
                                </h:outputText>
                                </rich:column>
                                <rich:column>
                                    <h:outputText
value="#{_atividade.adubacao.fertilizante.nome}" />
                                </rich:column>
                                <rich:column>
                                    <h:outputText
value="#{_atividade.adubacao.quantidade}" />
                                </rich:column>
                                <rich:column>
                                    <h:outputText
value="#{_atividade.adubacao.valorTotal}">
                                <ui:include
src="/layout/moeda.xhtml" />
                                </h:outputText>
                                </rich:column>
                                </rich:columnGroup>
                                <f:facet name="footer">
                                <rich:columnGroup>
                                    <rich:column>
                                        <h:outputText
value="#{messages.totaisDeAdubacoes}" />
                                    </rich:column>
                                    <rich:column>
                                        <h:outputText value="" />
                                    </rich:column>
                                    <rich:column>
                                        <h:outputText value="">
                                        </h:outputText>
                                    </rich:column>
                                    <rich:column>
                                        <h:outputText
value="#{talhaoHome.totalCustoAdubacao}">
                                <ui:include
src="/layout/moeda.xhtml" />

```

```

                </h:outputText>
            </rich:column>
        </rich:columnGroup>
    </f:facet>
</rich:dataTable>

<rich:spacer height="5px"/>

<div style="clear:both"/>

    <rich:dataTable var="_atividade"
value="#{talhaoHome.atividadeTalhaos}">
        <f:facet name="header">
            <rich:columnGroup>
                <rich:column colspan="5">
                    <h:outputText
value="#{messages.aplicacoesDefensivosEfetuadas}" />
                </rich:column>
                <rich:column breakBefore="true">
                    <h:outputLabel
value="#{messages.data}" />
                </rich:column>
                <rich:column>
                    <h:outputLabel
value="#{messages.hora}" />
                </rich:column>
                <rich:column>
                    <h:outputLabel
value="#{messages.quantidade}" />
                </rich:column>
                <rich:column>
                    <h:outputLabel
value="#{messages.defensivo}" />
                </rich:column>
                <rich:column>
                    <h:outputLabel
value="#{messages.valorTotal}" />
            </rich:columnGroup>
        </f:facet>
    </rich:dataTable>

```

```

        </rich:column>

    </rich:columnGroup>
</f:facet>
    <rich:columnGroup rendered="#{null !=
_atividade.aplicacaoDefensivo}">
        <rich:column>
            <h:outputText
value="#{_atividade.dataAtividade}"
            <f:convertDateTime
pattern="#{messages.dataFormato}" />
            </h:outputText>
        </rich:column>
        <rich:column>
            <h:outputText
value="#{_atividade.aplicacaoDefensivo.hora}" />
        </rich:column>
        <rich:column>
            <h:outputText
value="#{_atividade.aplicacaoDefensivo.quantidade}" />
        </rich:column>
        <rich:column>
            <h:outputText
value="#{_atividade.aplicacaoDefensivo.defensivo.nome}" />
        </rich:column>
        <rich:column>
            <h:outputText
value="#{_atividade.aplicacaoDefensivo.valorTotal}">
                <ui:include
src="/layout/moeda.xhtml" />
            </h:outputText>
        </rich:column>
    </rich:columnGroup>
</f:facet name="footer">
    <rich:columnGroup>
        <rich:column>

```



```

                <h:outputText
value="#{messages.totaisDedefensivos}"/>
            </rich:column>
            <rich:column>
                <h:outputText value=""/>
            </rich:column>
            <rich:column>
                <h:outputText value=""/>
            </rich:column>
            <rich:column>
                <h:outputText value=""/>
            </rich:column>
            <rich:column>
                <h:outputText value=""/>
            </rich:column>
            <rich:column>
                <h:outputText
value="#{talhaoHome.totalCustoDefensivo}">
                <ui:include
src="/layout/moeda.xhtml" />
            </h:outputText>
        </rich:column>
    </rich:columnGroup>
</f:facet>
</rich:dataTable>
<rich:spacer height="5px"/>

<div style="clear:both"/>

    <rich:dataTable var="_atividade"
value="#{talhaoHome.atividadeTalhaos}">
        <f:facet name="header">
            <rich:columnGroup>
                <rich:column colspan="6">
                    <h:outputText
value="#{messages.plantiosEfetuados}" />
                </rich:column>
                <rich:column breakBefore="true">
                    <h:outputLabel
value="#{messages.data}" />

```

```

        </rich:column>
        <rich:column>
            <h:outputLabel
value="#{messages.dataPrevistaColheita}" />
        </rich:column>
        <rich:column>
            <h:outputLabel
value="#{messages.quantidade}" />
        </rich:column>
        <rich:column>
            <h:outputLabel
value="#{messages.variedade}" />
        </rich:column>
        <rich:column>
            <h:outputLabel
value="#{messages.cultura}" />
        </rich:column>
        <rich:column>
            <h:outputLabel
value="#{messages.valorTotal}" />
        </rich:column>
    </rich:columnGroup>
</f:facet>
    <rich:columnGroup rendered="#{null !=
_atividade.plantio}">
        <rich:column>
            <h:outputText
value="#{_atividade.dataAtividade}">
                <f:convertDateTime
pattern="#{messages.dataFormato}" />
            </h:outputText>
        </rich:column>
        <rich:column>
            <h:outputText
value="#{_atividade.plantio.dataPrevistaColheita}" />
        </rich:column>
        <rich:column>

```

```

                <h:outputText
value="#{_atividade.plantio.quantidade}" />
            </rich:column>
            <rich:column>
                <h:outputText
value="#{_atividade.plantio.variedade.nome}" />
            </rich:column>
            <rich:column>
                <h:outputText
value="#{_atividade.plantio.variedade.cultura.nome}" />
            </rich:column>
            <rich:column>
                <h:outputText
value="#{_atividade.plantio.valorTotal}">
                <ui:include
src="/layout/moeda.xhtml" />
            </h:outputText>
        </rich:column>
    </rich:columnGroup>
    <f:facet name="footer">
        <rich:columnGroup>
            <rich:column>
                <h:outputText
value="#{messages.totaisDePlantio}"/>
            </rich:column>
            <rich:column>
                <h:outputText value=""/>
            </rich:column>
            <rich:column>
                <h:outputText value=""/>
            </rich:column>
            <rich:column>
                <h:outputText value=""/>
            </rich:column>
            <rich:column>
                <h:outputText value=""/>
            </rich:column>
            <rich:column>
                <h:outputText value=""/>
            </rich:column>
            <rich:column>
                <h:outputText value=""/>
            </rich:column>
        </rich:columnGroup>
    </f:facet>

```

```

                <rich:column>
                    <h:outputText
value="#{talhaoHome.totalCustoPlantio}">
                        <ui:include
src="/layout/moeda.xhtml" />
                    </h:outputText>
                </rich:column>
            </rich:columnGroup>
        </f:facet>
    </rich:dataTable>
    <rich:spacer height="5px"/>

    <div style="clear:both"/>

    <rich:dataTable var="_atividade"
value="#{talhaoHome.atividadeTalhaos}">
        <f:facet name="header">
            <rich:columnGroup>
                <rich:column colspan="3">
                    <h:outputText
value="#{messages.colheitasEfetuadas}" />
                </rich:column>
                <rich:column breakBefore="true">
                    <h:outputLabel
value="#{messages.data}" />
                </rich:column>
                <rich:column>
                    <h:outputLabel
value="#{messages.quantidade}" />
                </rich:column>
                <rich:column>
                    <h:outputLabel
value="#{messages.valorTotal}" />
                </rich:column>
            </rich:columnGroup>
        </f:facet>

```

```

        <rich:columnGroup rendered="#{null !=
_atividade.colheita}">
            <rich:column>
                <h:outputText
value="#{_atividade.dataAtividade}">
                    <f:convertDateTime
pattern="#{messages.dataFormato}" />
                </h:outputText>
            </rich:column>
            <rich:column>
                <h:outputText
value="#{_atividade.colheita.quantidade}" />
            </rich:column>
            <rich:column>
                <h:outputText
value="#{_atividade.colheita.valorTotal}">
                    <ui:include
src="/layout/moeda.xhtml" />
                </h:outputText>
            </rich:column>
        </rich:columnGroup>
    </f:facet name="footer">
        <rich:columnGroup>
            <rich:column>
                <h:outputText
value="#{messages.totaisDeColheita}" />
            </rich:column>
            <rich:column>
                <h:outputText value=""/>
            </rich:column>
            <rich:column>
                <h:outputText
value="#{talhaoHome.totalBrutoColheita}">
                    <ui:include
src="/layout/moeda.xhtml" />
                </h:outputText>
            </rich:column>
        </rich:columnGroup>
    </f:facet name="total">

```

```

        </f:facet>
    </rich:dataTable>
    <rich:spacer height="5px"/>

    <div style="clear:both"/>

        <rich:dataTable var="_oper"
value="#{talhaoHome.operacoesDoTalhao}">
            <f:facet name="header">
                <rich:columnGroup>
                    <rich:column colspan="4">
                        <h:outputText
value="#{messages.operacoesEfetuadas}" />
                    </rich:column>
                    <rich:column breakBefore="true">
                        <h:outputLabel
value="#{messages.dataInicio}" />
                    </rich:column>
                    <rich:column>
                        <h:outputLabel
value="#{messages.dataFim}" />
                    </rich:column>
                    <rich:column>
                        <h:outputLabel
value="#{messages.horasTrabalhadas}" />
                    </rich:column>
                    <rich:column>
                        <h:outputLabel
value="#{messages.custoOperacao}" />
                    </rich:column>
                </rich:columnGroup>
            </f:facet>
            <rich:columnGroup>
                <rich:column>
                    <h:outputText
value="#{_oper.dataInicial}">
                    <hr style="width: 50%; margin-left: auto; margin-right: 0; border: 0.5px solid black;"/>
                    <f:convertDateTime
pattern="#{messages.dataFormato}" />
                </rich:column>
            </rich:columnGroup>
        </rich:dataTable>

```

```

                </h:outputText>
            </rich:column>
        </rich:column>
        <rich:column>
            <h:outputText
value="#{_oper.dataFinal}">
                _____ <f:convertDateTime
pattern="#{messages.dataFormato}" />
            </h:outputText>
        </rich:column>
        <rich:column>
            <h:outputText
value="#{_oper.horasTrabalhadas}" />
                _____ </rich:column>
            </rich:column>
            <h:outputText
value="#{_oper.horasTrabalhadas * _oper.maquina.valorHora}">
                <ui:include
src="/layout/moeda.xhtml" />
            </h:outputText>
        </rich:column>
    </rich:columnGroup>
    <f:facet name="footer">
        <rich:columnGroup>
            <rich:column>
                <h:outputText
value="#{messages.totaisDeOperacoesComMaquinas}"/>
            </rich:column>
            <rich:column>
                <h:outputText value=""/>
            </rich:column>
            <rich:column>
                <h:outputText
value="#{talhaoHome.totalHorasTrabalhadas}">
            </h:outputText>
        </rich:column>
        <rich:column>
            <h:outputText
value="#{talhaoHome.totalCustoOperacoes}">

```

```

        <ui:include
src="/layout/moeda.xhtml" />
        </h:outputText>
    </rich:column>
</rich:columnGroup>
</f:facet>
</rich:dataTable>

</rich:panel>

<rich:spacer height="5px"/>

<div style="clear:both"/>

<rich:panel>
<f:facet name="header">#{messages.ocorrenciaPraga}</f:facet>

    <s:button view="/agricultura/cadOcorrenciaPraga.xhtml"
value="#{messages.registrarOcorrenciaPraga}" >
        <f:param name="idTalaho"
value="#{talhaoHome.instance.idTalhao}"/>
    </s:button>

<p/>

    <rich:dataTable var="_praga"
value="#{talhaoHome.ocorrenciaPragas}" rendered="#{not empty
talhaoHome.ocorrenciaPragas}">
        <f:facet name="header">
            <rich:columnGroup>
                <rich:column>
                    <h:outputLabel
value="#{messages.data}" />
                </rich:column>
                <rich:column>
                    <h:outputLabel
value="#{messages.nome}" />
                </rich:column>
            </rich:columnGroup>
        </f:facet>
    </rich:dataTable>

```



```

        <rich:column>
            <h:outputLabel
value="#{messages.descricao}" />
        </rich:column>
        <rich:column>
            <h:outputLabel
value="#{messages.solucao}" />
        </rich:column>
    </rich:columnGroup>
</f:facet>
    <rich:columnGroup>
        <rich:column>
            <h:outputText
value="#{_praga.dataOcorrencia}"
            <f:convertDateTime
pattern="#{messages.dataFormato}" />
            </h:outputText>
        </rich:column>
        <rich:column>
            <h:outputText value="#{_praga.nome}"
/>
        </rich:column>
        <rich:column>
            <h:outputText
value="#{_praga.descricao}" />
        </rich:column>
        <rich:column>
            <h:outputText
value="#{_praga.solucao}">
            </h:outputText>
        </rich:column>
    </rich:columnGroup>
</rich:dataTable>

</rich:panel>

<rich:spacer height="5px"/>

```

```
        <div style="clear:both"/>

        </rich:panel>
    </h:form>
</ui:define>

</ui:composition>
```

## Java

```
package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import org.hibernate.validator.NotNull;
```

```

/**
 * Variedade generated by hbm2java
 */
@Entity
@Table(name = "variedade", schema = "public")
public class Variedade implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 5742597442745803440L;
    private int idVariedade;
    private Cultura cultura;
    private Fabricante fabricante;
    private String nome;
    private String informacoes;
    private Set<Plantio> plantios = new HashSet<Plantio>(0);

    public Variedade() {
    }

    public Variedade(int idVariedade, Cultura cultura, String nome) {
        this.idVariedade = idVariedade;
        this.cultura = cultura;
        this.nome = nome;
    }

    public Variedade(int idVariedade, Cultura cultura, Fabricante
fabricante,
        String nome, String informacoes, Set<Plantio> plantios) {
        this.idVariedade = idVariedade;
        this.cultura = cultura;
        this.fabricante = fabricante;
        this.nome = nome;
        this.informacoes = informacoes;
        this.plantios = plantios;
    }
}

```

```

@Id
@Column(name = "id_variedade", unique = true, nullable = false)
@SequenceGenerator( name = "VARIEDADE_ID", sequenceName =
"id_variedade_seq", allocationSize = 1 )
@GeneratedValue( strategy = GenerationType.AUTO, generator =
"VARIEDADE_ID" )
@NotNull
public int getIdVariedade() {
    return this.idVariedade;
}

public void setIdVariedade(int idVariedade) {
    this.idVariedade = idVariedade;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_cultura", nullable = false)
@NotNull
public Cultura getCultura() {
    return this.cultura;
}

public void setCultura(Cultura cultura) {
    this.cultura = cultura;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_fabricante")
public Fabricante getFabricante() {
    return this.fabricante;
}

public void setFabricante(Fabricante fabricante) {
    this.fabricante = fabricante;
}

@Column(name = "nome", nullable = false)

```

```

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

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

@Column(name = "informacoes")
public String getInformacoes() {
    return this.informacoes;
}

public void setInformacoes(String informacoes) {
    this.informacoes = informacoes;
}

@OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"variedade")
public Set<Plantio> getPlantios() {
    return this.plantios;
}

public void setPlantios(Set<Plantio> plantios) {
    this.plantios = plantios;
}
}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;

```

```

import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.UniqueConstraint;

import org.hibernate.validator.Email;
import org.hibernate.validator.NotNull;

/**
 * Usuario generated by hbm2java
 */
@Entity
@Table(name = "usuario", schema = "public", uniqueConstraints =
@UniqueConstraint(columnNames = "email"))
public class Usuario implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -4160204248440836694L;
    private long idUsuario;
    private Propriedade propriedade;
    private String nome;
    private String email;
    private boolean administrador;

    public Usuario() {
    }

    public Usuario(long idUsuario, String nome, String email,
        boolean administrador) {
        this.idUsuario = idUsuario;
        this.nome = nome;
        this.email = email;
    }

```

```

        this.administrador = administrador;
    }

    public Usuario(long idUsuario, Propriedade propriedade, String nome,
        String email, boolean administrador) {
        this.idUsuario = idUsuario;
        this.propriedade = propriedade;
        this.nome = nome;
        this.email = email;
        this.administrador = administrador;
    }

    @Id
    @Column(name = "id_usuario", unique = true, nullable = false)
    @SequenceGenerator( name = "USUARIO_ID", sequenceName =
        "id_usuario_seq", allocationSize = 1 )
    @GeneratedValue( strategy = GenerationType.AUTO, generator =
        "USUARIO_ID" )
    @NotNull
    public long getIdUsuario() {
        return this.idUsuario;
    }

    public void setIdUsuario(long idUsuario) {
        this.idUsuario = idUsuario;
    }

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "cod_propriedade")
    @NotNull
    public Propriedade getPropriedade() {
        return this.propriedade;
    }

    public void setPropriedade(Propriedade propriedade) {
        this.propriedade = propriedade;
    }
}

```

```

@Column(name = "nome", nullable = false)
@NotNull
public String getNome() {
    return this.nome;
}

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

@Column(name = "email", unique = true, nullable = false)
@NotNull
>Email
public String getEmail() {
    return this.email;
}

public void setEmail(String email) {
    this.email = email;
}

@Column(name = "administrador", nullable = false)
@NotNull
public boolean isAdministrador() {
    return this.administrador;
}

public void setAdministrador(boolean administrador) {
    this.administrador = administrador;
}

}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.HashSet;
import java.util.Set;

```



```

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.Transient;

import org.hibernate.validator.NotNull;

/**
 * Talhao generated by hbm2java
 */
@Entity
@Table(name = "talhao", schema = "public")
public class Talhao implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -4017066866282871126L;
    private long idTalhao;
    private Safra safra;
    private String nome;
    private double area;
    private String descricao;
    private String localizacao;
    private boolean terrenoProprio;
    private boolean cultivoPermanente;
    private double saldoTalhao;
    private double custoTalhao;
    private double arrecadacaoTalhao;

```

```

        private Set<AtividadeTalhao> atividadeTalhaos = new
HashSet<AtividadeTalhao>(0);
        private Set<OcorrenciaPraga> ocorrenciaPragas = new
HashSet<OcorrenciaPraga>(0);

    public Talhao() {
    }

    public Talhao(long idTalhao, Safra safra, String nome, double area,
        String localizacao, boolean terrenoProprio,
        boolean cultivoPermanente) {
        this.idTalhao = idTalhao;
        this.safra = safra;
        this.nome = nome;
        this.area = area;
        this.localizacao = localizacao;
        this.terrenoProprio = terrenoProprio;
        this.cultivoPermanente = cultivoPermanente;
    }

    public Talhao(long idTalhao, Safra safra, String nome, double area,
        String descricao, String localizacao, boolean
terrenoProprio,
        boolean cultivoPermanente, Set<AtividadeTalhao>
atividadeTalhaos,
        Set<OcorrenciaPraga> ocorrenciaPragas) {
        this.idTalhao = idTalhao;
        this.safra = safra;
        this.nome = nome;
        this.area = area;
        this.descricao = descricao;
        this.localizacao = localizacao;
        this.terrenoProprio = terrenoProprio;
        this.cultivoPermanente = cultivoPermanente;
        this.atividadeTalhaos = atividadeTalhaos;
        this.ocorrenciaPragas = ocorrenciaPragas;
    }

```

```

@Id
@Column(name = "id_talhao", unique = true, nullable = false)
@SequenceGenerator( name = "TALHAO_ID", sequenceName = "id_talhao_seq",
allocationSize = 1 )
@GeneratedValue( strategy = GenerationType.AUTO, generator = "TALHAO_ID"
)

@NotNull
public long getIdTalhao() {
    return this.idTalhao;
}

public void setIdTalhao(long idTalhao) {
    this.idTalhao = idTalhao;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_safra", nullable = false)
@NotNull
public Safra getSafra() {
    return this.safra;
}

public void setSafra(Safra safra) {
    this.safra = safra;
}

@Column(name = "nome", nullable = false)
@NotNull
public String getNome() {
    return this.nome;
}

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

@Column(name = "area", nullable = false, precision = 17, scale = 17)
@NotNull

```

```

public double getArea() {
    return this.area;
}

public void setArea(double area) {
    this.area = area;
}

@Column(name = "descricao")
public String getDescricao() {
    return this.descricao;
}

public void setDescricao(String descricao) {
    this.descricao = descricao;
}

@Column(name = "localizacao", nullable = false)
@NotNull
public String getLocalizacao() {
    return this.localizacao;
}

public void setLocalizacao(String localizacao) {
    this.localizacao = localizacao;
}

@Column(name = "terreno_proprio", nullable = false)
@NotNull
public boolean isTerrenoProprio() {
    return this.terrenoProprio;
}

public void setTerrenoProprio(boolean terrenoProprio) {
    this.terrenoProprio = terrenoProprio;
}

@Column(name = "cultivo_permanente", nullable = false)

```

```

@NotNull
public boolean isCultivoPermanente() {
    return this.cultivoPermanente;
}

public void setCultivoPermanente(boolean cultivoPermanente) {
    this.cultivoPermanente = cultivoPermanente;
}

@OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"talhao")
public Set<AtividadeTalhao> getAtividadeTalhaos() {
    return this.atividadeTalhaos;
}

public void setAtividadeTalhaos(Set<AtividadeTalhao> atividadeTalhaos) {
    this.atividadeTalhaos = atividadeTalhaos;
}

@OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"talhao")
public Set<OcorrenciaPraga> getOcorrenciaPragas() {
    return this.ocorrenciaPragas;
}

public void setOcorrenciaPragas(Set<OcorrenciaPraga> ocorrenciaPragas) {
    this.ocorrenciaPragas = ocorrenciaPragas;
}

@Transient
public double getSaldoTalhao() {
    return saldoTalhao;
}

public void setSaldoTalhao(double saldoTalhao) {
    this.saldoTalhao = saldoTalhao;
}

```

```

    @Transient
    public double getCustoTalhao() {
        return custoTalhao;
    }

    public void setCustoTalhao(double custoTalhao) {
        this.custoTalhao = custoTalhao;
    }

    @Transient
    public double getArrecadacaoTalhao() {
        return arrecadacaoTalhao;
    }

    public void setArrecadacaoTalhao(double resultadoTalhao) {
        this.arrecadacaoTalhao = resultadoTalhao;
    }

}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.UniqueConstraint;
import org.hibernate.validator.NotNull;

/**
 * Senha generated by hbm2java
 */
@Entity

```

```

@Table(name = "senha", schema = "public", uniqueConstraints =
@UniqueConstraint(columnNames = "email"))
public class Senha implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 432156253498996064L;
    private int idSenha;
    private String senha;
    private String email;

    public Senha() {
    }

    public Senha(int idSenha, String senha, String email) {
        this.idSenha = idSenha;
        this.senha = senha;
        this.email = email;
    }

    @Id
    @Column(name = "id_senha", unique = true, nullable = false)
    @SequenceGenerator( name = "SENHA_ID", sequenceName = "id_senha_seq",
allocationSize = 1 )
    @GeneratedValue( strategy = GenerationType.AUTO, generator = "SENHA_ID"
)
    @NotNull
    public int getIdSenha() {
        return this.idSenha;
    }

    public void setIdSenha(int idSenha) {
        this.idSenha = idSenha;
    }

    @Column(name = "senha", nullable = false)
    @NotNull

```

```

    public String getSenha() {
        return this.senha;
    }

    public void setSenha(String senha) {
        this.senha = senha;
    }

    @Column(name = "email", unique = true, nullable = false)
    @NotNull
    public String getEmail() {
        return this.email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.Date;
import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;

```



```

import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.persistence.UniqueConstraint;
import org.hibernate.validator.NotNull;

/**
 * Safra generated by hbm2java
 */
@Entity
@Table(name = "safra", schema = "public", uniqueConstraints =
@UniqueConstraint(columnNames = {"ano_safra", "cod_propriedade" }))
public class Safra implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 8360012527091845123L;
    private long idSafra;
    private Propriedade propriedade;
    private String anoSafra;
    private Date dataInicio;
    private Date dataFim;
    private boolean atual;
    private Set<Talhao> talhaos = new HashSet<Talhao>(0);

    public Safra() {
    }

    public Safra(long idSafra, Propriedade propriedade, String anoSafra,
        Date dataInicio, boolean atual) {
        this.idSafra = idSafra;
        this.propriedade = propriedade;
        this.anoSafra = anoSafra;
        this.dataInicio = dataInicio;
        this.atual = atual;
    }

    public Safra(long idSafra, Propriedade propriedade, String anoSafra,

```

```

        Date dataInicio, Date dataFim, boolean atual, Set<Talhao>
talhaos) {
    this.idSafra = idSafra;
    this.propriedade = propriedade;
    this.anoSafra = anoSafra;
    this.dataInicio = dataInicio;
    this.dataFim = dataFim;
    this.atual = atual;
    this.talhaos = talhaos;
}

@Id
@Column(name = "id_safra", unique = true, nullable = false)
@SequenceGenerator( name = "SAFRA_ID", sequenceName = "id_safra_seq",
allocationSize = 1 )
@GeneratedValue( strategy = GenerationType.AUTO, generator = "SAFRA_ID"
)

@NotNull
public long getIdSafra() {
    return this.idSafra;
}

public void setIdSafra(long idSafra) {
    this.idSafra = idSafra;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_propriedade", nullable = false)
@NotNull
public Propriedade getPropriedade() {
    return this.propriedade;
}

public void setPropriedade(Propriedade propriedade) {
    this.propriedade = propriedade;
}

@Column(name = "ano_safra", nullable = false)

```

```

@NotNull
public String getAnoSafra() {
    return this.anoSafra;
}

public void setAnoSafra(String anoSafra) {
    this.anoSafra = anoSafra;
}

@Temporal(TemporalType.DATE)
@Column(name = "data_inicio", nullable = false, length = 13)
@NotNull
public Date getDataInicio() {
    return this.dataInicio;
}

public void setDataInicio(Date dataInicio) {
    this.dataInicio = dataInicio;
}

@Temporal(TemporalType.DATE)
@Column(name = "data_fim", length = 13)
public Date getDataFim() {
    return this.dataFim;
}

public void setDataFim(Date dataFim) {
    this.dataFim = dataFim;
}

@Column(name = "atual", nullable = false)
@NotNull
public boolean isAtual() {
    return this.atual;
}

public void setAtual(boolean atual) {
    this.atual = atual;
}

```

```

    }

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"safra")
    public Set<Talhao> getTalhaos() {
        return this.talhaos;
    }

    public void setTalhaos(Set<Talhao> talhaos) {
        this.talhaos = talhaos;
    }
}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.HashSet;
import java.util.Set;

import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;

import org.hibernate.validator.NotNull;

/**
 * Propriedade generated by hbm2java
 */

```

```

@Entity
@Table(name = "propriedade", schema = "public")
public class Propriedade implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -6427261149717206982L;
    private int idPropriedade;
    private Endereco endereco;
    private Cliente cliente;
    private double areaTotal;
    private String nome;
    private double areaPreservacao;
    private double areaCultivo;
    private double areaOutras;
    private String email;
    private Set<Usuario> usuarios = new HashSet<Usuario>(0);
    private Set<Safras> safras = new HashSet<Safras>(0);
    private Set<Maquina> maquinas = new HashSet<Maquina>(0);

    public Propriedade() {
    }

    public Propriedade(int idPropriedade, Cliente cliente, double areaTotal,
        double areaPreservacao, double areaCultivo, double
areaOutras) {
        this.idPropriedade = idPropriedade;
        this.cliente = cliente;
        this.areaTotal = areaTotal;
        this.areaPreservacao = areaPreservacao;
        this.areaCultivo = areaCultivo;
        this.areaOutras = areaOutras;
    }

    public Propriedade(int idPropriedade, Endereco endereco, Cliente
cliente,
        double areaTotal, String nome, double areaPreservacao,

```

```

        double areaCultivo, double areaOutras, String email,
        Set<Usuario> usuarios, Set<Safra> safras, Set<Maquina>
maquinas) {
    this.idPropriedade = idPropriedade;
    this.endereco = endereco;
    this.cliente = cliente;
    this.areaTotal = areaTotal;
    this.nome = nome;
    this.areaPreservacao = areaPreservacao;
    this.areaCultivo = areaCultivo;
    this.areaOutras = areaOutras;
    this.email = email;
    this.usuarios = usuarios;
    this.safras = safras;
    this.maquinas = maquinas;
}

@Id
@Column(name = "id_propriedade", unique = true, nullable = false)
@SequenceGenerator( name = "PROPRIEDADE_ID", sequenceName =
"id_propriedade_seq", allocationSize = 1 )
@GeneratedValue( strategy = GenerationType.AUTO, generator =
"PROPRIEDADE_ID" )
@NotNull
public int getIdPropriedade() {
    return this.idPropriedade;
}

public void setIdPropriedade(int idPropriedade) {
    this.idPropriedade = idPropriedade;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_endereco")
public Endereco getEndereco() {
    return this.endereco;
}

```

```

public void setEndereco(Endereco endereco) {
    this.endereco = endereco;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_cliente", nullable = false)
@NotNull
public Cliente getCliente() {
    return this.cliente;
}

public void setCliente(Cliente cliente) {
    this.cliente = cliente;
}

@Column(name = "area_total", nullable = false, precision = 17, scale =
17)
@NotNull
public double getAreaTotal() {
    return this.areaTotal;
}

public void setAreaTotal(double areaTotal) {
    this.areaTotal = areaTotal;
}

@Column(name = "nome")
public String getNome() {
    return this.nome;
}

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

@Column(name = "area_preservacao", nullable = false, precision = 17,
scale = 17)
@NotNull

```

```

    public double getAreaPreservacao() {
        return this.areaPreservacao;
    }

    public void setAreaPreservacao(double areaPreservacao) {
        this.areaPreservacao = areaPreservacao;
    }

@Column(name = "area_cultivo", nullable = false, precision = 17, scale =
17)
@NotNull
    public double getAreaCultivo() {
        return this.areaCultivo;
    }

    public void setAreaCultivo(double areaCultivo) {
        this.areaCultivo = areaCultivo;
    }

@Column(name = "area_outras", nullable = false, precision = 17, scale =
17)
@NotNull
    public double getAreaOutras() {
        return this.areaOutras;
    }

    public void setAreaOutras(double areaOutras) {
        this.areaOutras = areaOutras;
    }

@Column(name = "email")
    public String getEmail() {
        return this.email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

```



```

        @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"propriedade")
        public Set<Usuario> getUsuarios() {
            return this.usuarios;
        }

        public void setUsuarios(Set<Usuario> usuarios) {
            this.usuarios = usuarios;
        }

        @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"propriedade")
        public Set<Safras> getSafras() {
            return this.safras;
        }

        public void setSafras(Set<Safras> safras) {
            this.safras = safras;
        }

        @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"propriedade")
        public Set<Maquina> getMaquinas() {
            return this.maquinas;
        }

        public void setMaquinas(Set<Maquina> maquinas) {
            this.maquinas = maquinas;
        }
    }

    package br.ufsc.inf.faccin.efagro.entity;

    // Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

    import java.util.Date;
    import javax.persistence.Column;

```

```

import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import org.hibernate.validator.NotNull;

/**
 * Plantio generated by hbm2java
 */
@Entity
@Table(name = "plantio", schema = "public")
public class Plantio implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -785338822378372850L;
    private long codAtividadeTalhao;
    private AtividadeTalhao atividadeTalhao;
    private Variedade variedade;
    private Date dataPrevistaColheita;
    private double quantidade;
    private double valorTotal;

    public Plantio() {
    }

    public Plantio(long codAtividadeTalhao, AtividadeTalhao atividadeTalhao,
        Variedade variedade, Date dataPrevistaColheita, double
quantidade,
        double valorTotal) {
        this.codAtividadeTalhao = codAtividadeTalhao;
        this.atividadeTalhao = atividadeTalhao;

```

```

        this.variedade = variedade;
        this.dataPrevistaColheita = dataPrevistaColheita;
        this.quantidade = quantidade;
        this.valorTotal = valorTotal;
    }

    @Id
    @Column(name = "cod_atividade_talhao", unique = true, nullable = false)
    @NotNull
    public long getCodAtividadeTalhao() {
        return this.codAtividadeTalhao;
    }

    public void setCodAtividadeTalhao(long codAtividadeTalhao) {
        this.codAtividadeTalhao = codAtividadeTalhao;
    }

    @OneToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "cod_atividade_talhao", unique = true, nullable =
false, insertable = false, updatable = false)
    @NotNull
    public AtividadeTalhao getAtividadeTalhao() {
        return this.atividadeTalhao;
    }

    public void setAtividadeTalhao(AtividadeTalhao atividadeTalhao) {
        this.atividadeTalhao = atividadeTalhao;
    }

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "cod_variedade", nullable = false)
    @NotNull
    public Variedade getVariedade() {
        return this.variedade;
    }

    public void setVariedade(Variedade variedade) {
        this.variedade = variedade;
    }

```

```

    }

    @Temporal(TemporalType.DATE)
    @Column(name = "data_prevista_colheita", nullable = false, length = 13)
    @NotNull
    public Date getDataPrevistaColheita() {
        return this.dataPrevistaColheita;
    }

    public void setDataPrevistaColheita(Date dataPrevistaColheita) {
        this.dataPrevistaColheita = dataPrevistaColheita;
    }

    @Column(name = "quantidade", nullable = false, precision = 17, scale =
17)
    @NotNull
    public double getQuantidade() {
        return this.quantidade;
    }

    public void setQuantidade(double quantidade) {
        this.quantidade = quantidade;
    }

    @Column(name = "valor_total", nullable = false, precision = 17, scale =
17)
    @NotNull
    public double getValorTotal() {
        return this.valorTotal;
    }

    public void setValorTotal(double valorTotal) {
        this.valorTotal = valorTotal;
    }

}

package br.ufsc.inf.faccin.efagro.entity;

```

```
// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1
```

```
import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import javax.persistence.OrderBy;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import org.hibernate.validator.NotNull;

/**
 * Pais generated by hbm2java
 */
@Entity
@Table(name = "pais", schema = "public")
public class Pais implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 644485046746478721L;
    private int idPais;
    private String nome;
    private String sigla;
    private Set<Estado> estados = new HashSet<Estado>(0);

    public Pais() {
    }

    public Pais(int idPais, String nome) {
        this.idPais = idPais;
    }
}
```

```

        this.nome = nome;
    }

    public Pais(int idPais, String nome, String sigla, Set<Estado> estados)
    {
        this.idPais = idPais;
        this.nome = nome;
        this.sigla = sigla;
        this.estados = estados;
    }

    @Id
    @Column(name = "id_pais", unique = true, nullable = false)
    @SequenceGenerator( name = "PAIS_ID", sequenceName = "id_pais_seq",
allocationSize = 1 )
    @GeneratedValue( strategy = GenerationType.AUTO, generator = "PAIS_ID" )
    @NotNull
    public int getIdPais() {
        return this.idPais;
    }

    public void setIdPais(int idPais) {
        this.idPais = idPais;
    }

    @Column(name = "nome", nullable = false)
    @NotNull
    @OrderBy
    public String getNome() {
        return this.nome;
    }

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

    @Column(name = "sigla")
    public String getSigla() {

```

```

        return this.sigla;
    }

    public void setSigla(String sigla) {
        this.sigla = sigla;
    }

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"pais")
    public Set<Estado> getEstados() {
        return this.estados;
    }

    public void setEstados(Set<Estado> estados) {
        this.estados = estados;
    }

}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.Date;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import org.hibernate.validator.NotNull;

/**

```

```

* Operacao generated by hbm2java
*/
@Entity
@Table(name = "operacao", schema = "public")
public class Operacao implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -1480211564078107244L;
    private long idOperacao;
    private Maquina maquina;
    private AtividadeTalhao atividadeTalhao;
    private Date dataInicial;
    private Date dataFinal;
    private String descricao;
    private double horasTrabalhadas;

    public Operacao() {
    }

    public Operacao(long idOperacao, Maquina maquina, AtividadeTalhao
atividadeTalhao,
                    Date dataInicial, Date dataFinal, double horasTrabalhadas,
String descricao) {
        this.idOperacao = idOperacao;
        this.maquina = maquina;
        this.atividadeTalhao = atividadeTalhao;
        this.dataInicial = dataInicial;
        this.dataFinal = dataFinal;
        this.horasTrabalhadas = horasTrabalhadas;
        this.setDescricao(descricao);
    }

    @Id
    @Column(name = "id_operacao", unique = true, nullable = false)
    @SequenceGenerator( name = "OPERACAO_ID", sequenceName =
"id_operacao_seq", allocationSize = 1 )

```



```

    @GeneratedValue( strategy = GenerationType.AUTO, generator =
"OPERACAO_ID" )
    @NotNull
    public long getIdOperacao() {
        return this.idOperacao;
    }

    public void setIdOperacao(long idOperacao) {
        this.idOperacao = idOperacao;
    }

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "cod_maquina", nullable = false)
    @NotNull
    public Maquina getMaquina() {
        return this.maquina;
    }

    public void setMaquina(Maquina maquina) {
        this.maquina = maquina;
    }

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "cod_atividade_talhao", nullable = true)
    public AtividadeTalhao getAtividadeTalhao() {
        return this.atividadeTalhao;
    }

    public void setAtividadeTalhao(AtividadeTalhao atividadeTalhao) {
        this.atividadeTalhao = atividadeTalhao;
    }

    @Temporal(TemporalType.TIMESTAMP)
    @Column(name = "data_inicial", nullable = false, length = 29)
    @NotNull
    public Date getDataInicial() {
        return this.dataInicial;
    }
}

```

```

    public void setDataInicial(Date dataInicial) {
        this.dataInicial = dataInicial;
    }

    @Temporal(TemporalType.TIMESTAMP)
    @Column(name = "data_final", nullable = false, length = 29)
    @NotNull
    public Date getDataFinal() {
        return this.dataFinal;
    }

    public void setDataFinal(Date dataFinal) {
        this.dataFinal = dataFinal;
    }

    @Column(name = "horas_trabalhadas", nullable = false, precision = 17,
scale = 17)
    @NotNull
    public double getHorasTrabalhadas() {
        return this.horasTrabalhadas;
    }

    public void setHorasTrabalhadas(double horasTrabalhadas) {
        this.horasTrabalhadas = horasTrabalhadas;
    }

    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }

    @Column(name = "descricao", nullable = true)
    public String getDescricao() {
        return descricao;
    }

}

package br.ufsc.inf.faccin.efagro.entity;

```

```
// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1
```

```
import java.util.Date;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import org.hibernate.validator.NotNull;

/**
 * OcorrenciaPraga generated by hbm2java
 */
@Entity
@Table(name = "ocorrencia_praga", schema = "public")
public class OcorrenciaPraga implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -1576194275197218208L;
    private int idOcorrenciaPraga;
    private Talhao talhao;
    private String nome;
    private String descricao;
    private String solucao;
    private Date dataOcorrencia;

    public OcorrenciaPraga() {
    }
}
```

```

    public OcorrenciaPraga(int idOcorrenciaPraga, Talhao talhao, String
nome,
        Date dataOcorrencia) {
        this.idOcorrenciaPraga = idOcorrenciaPraga;
        this.talhao = talhao;
        this.nome = nome;
        this.dataOcorrencia = dataOcorrencia;
    }

    public OcorrenciaPraga(int idOcorrenciaPraga, Talhao talhao, String
nome,
        String descricao, String solucao, Date dataOcorrencia) {
        this.idOcorrenciaPraga = idOcorrenciaPraga;
        this.talhao = talhao;
        this.nome = nome;
        this.descricao = descricao;
        this.solucao = solucao;
        this.dataOcorrencia = dataOcorrencia;
    }

    @Id
    @Column(name = "id_ocorrencia_praga", unique = true, nullable = false)
    @SequenceGenerator( name = "OCORRENCIA_PRAGA_ID", sequenceName =
"id_ocorrencia_praga_seq", allocationSize = 1 )
    @GeneratedValue( strategy = GenerationType.AUTO, generator =
"OCORRENCIA_PRAGA_ID" )
    @NotNull
    public int getIdOcorrenciaPraga() {
        return this.idOcorrenciaPraga;
    }

    public void setIdOcorrenciaPraga(int idOcorrenciaPraga) {
        this.idOcorrenciaPraga = idOcorrenciaPraga;
    }

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "cod_talhao", nullable = false)

```

```

@NotNull
public Talhao getTalhao() {
    return this.talhao;
}

public void setTalhao(Talhao talhao) {
    this.talhao = talhao;
}

@Column(name = "nome", nullable = false)
@NotNull
public String getNome() {
    return this.nome;
}

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

@Column(name = "descricao")
public String getDescricao() {
    return this.descricao;
}

public void setDescricao(String descricao) {
    this.descricao = descricao;
}

@Column(name = "solucao")
public String getSolucao() {
    return this.solucao;
}

public void setSolucao(String solucao) {
    this.solucao = solucao;
}

@Temporal(TemporalType.DATE)

```

```

@Column(name = "data_ocorrencia", nullable = false, length = 13)
@NotNull
public Date getDataOcorrencia() {
    return this.dataOcorrencia;
}

public void setDataOcorrencia(Date dataOcorrencia) {
    this.dataOcorrencia = dataOcorrencia;
}

}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.Date;
import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import org.hibernate.validator.NotNull;

/**
 * Maquina generated by hbm2java
 */
@Entity

```

```

@Table(name = "maquina", schema = "public")
public class Maquina implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -2332834929752562381L;
    private int idMaquina;
    private Propriedade propriedade;
    private String modelo;
    private String marca;
    private Date dataAquisicao;
    private int anoFabricacao;
    private String numeroSerie;
    private String observacao;
    private double valorMaquina;
    private double valorHora;
    private String nome;
    private Set<Abastecimento> abastecimentos = new
HashSet<Abastecimento>(0);
    private Set<Operacao> operacoes = new HashSet<Operacao>(0);
    private Set<AgendaManutencao> agendaManutencaos = new
HashSet<AgendaManutencao>(0);
    private Set<Manutencao> manutencaos = new HashSet<Manutencao>(0);

    public Maquina() {
    }

    public Maquina(int idMaquina, Propriedade propriedade, String modelo,
        String marca, Date dataAquisicao, int anoFabricacao,
        double valorMaquina, double valorHora, String nome) {
        this.idMaquina = idMaquina;
        this.propriedade = propriedade;
        this.modelo = modelo;
        this.marca = marca;
        this.dataAquisicao = dataAquisicao;
        this.anoFabricacao = anoFabricacao;
        this.valorMaquina = valorMaquina;
    }
}

```

```

        this.valorHora = valorHora;
        this.nome = nome;
    }

    public Maquina(int idMaquina, Propriedade propriedade, String modelo,
        String marca, Date dataAquisicao, int anoFabricacao,
        String numeroSerie, String observacao, double valorMaquina,
        double valorHora, String nome, Set<Abastecimento>
abastecimentos,
        Set<Operacao> operacoes, Set<AgendaManutencao>
agendaManutencaos,
        Set<Manutencao> manutencaos) {
        this.idMaquina = idMaquina;
        this.propriedade = propriedade;
        this.modelo = modelo;
        this.marca = marca;
        this.dataAquisicao = dataAquisicao;
        this.anoFabricacao = anoFabricacao;
        this.numeroSerie = numeroSerie;
        this.observacao = observacao;
        this.valorMaquina = valorMaquina;
        this.valorHora = valorHora;
        this.nome = nome;
        this.abastecimentos = abastecimentos;
        this.operacoes = operacoes;
        this.agendaManutencaos = agendaManutencaos;
        this.manutencaos = manutencaos;
    }

    @Id
    @Column(name = "id_maquina", unique = true, nullable = false)
    @SequenceGenerator( name = "MAQUINA_ID", sequenceName =
" id_maquina_seq", allocationSize = 1 )
    @GeneratedValue( strategy = GenerationType.AUTO, generator =
"MAQUINA_ID" )
    @NotNull
    public int getIdMaquina() {
        return this.idMaquina;
    }

```



```

}

public void setIdMaquina(int idMaquina) {
    this.idMaquina = idMaquina;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_propriedade", nullable = false)
@NotNull
public Propriedade getPropriedade() {
    return this.propriedade;
}

public void setPropriedade(Propriedade propriedade) {
    this.propriedade = propriedade;
}

@Column(name = "modelo", nullable = false)
@NotNull
public String getModelo() {
    return this.modelo;
}

public void setModelo(String modelo) {
    this.modelo = modelo;
}

@Column(name = "marca", nullable = false)
@NotNull
public String getMarca() {
    return this.marca;
}

public void setMarca(String marca) {
    this.marca = marca;
}

@Temporal(TemporalType.DATE)

```

```

@Column(name = "data_aquisicao", nullable = false, length = 13)
@NotNull
public Date getDataAquisicao() {
    return this.dataAquisicao;
}

public void setDataAquisicao(Date dataAquisicao) {
    this.dataAquisicao = dataAquisicao;
}

@Column(name = "ano_fabricacao", nullable = false)
@NotNull
public int getAnoFabricacao() {
    return this.anoFabricacao;
}

public void setAnoFabricacao(int anoFabricacao) {
    this.anoFabricacao = anoFabricacao;
}

@Column(name = "numero_serie")
public String getNumeroSerie() {
    return this.numeroSerie;
}

public void setNumeroSerie(String numeroSerie) {
    this.numeroSerie = numeroSerie;
}

@Column(name = "observacao")
public String getObservacao() {
    return this.observacao;
}

public void setObservacao(String observacao) {
    this.observacao = observacao;
}

```

```

    @Column(name = "valor_maquina", nullable = false, precision = 17, scale
= 17)
    @NotNull
    public double getValorMaquina() {
        return this.valorMaquina;
    }

    public void setValorMaquina(double valorMaquina) {
        this.valorMaquina = valorMaquina;
    }

    @Column(name = "valor_hora", nullable = false, precision = 17, scale =
17)
    @NotNull
    public double getValorHora() {
        return this.valorHora;
    }

    public void setValorHora(double valorHora) {
        this.valorHora = valorHora;
    }

    @Column(name = "nome", nullable = false)
    @NotNull
    public String getNome() {
        return this.nome;
    }

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

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"maquina")
    public Set<Abastecimento> getAbastecimentos() {
        return this.abastecimentos;
    }

```

```

    public void setAbastecimentos(Set<Abastecimento> abastecimentos) {
        this.abastecimentos = abastecimentos;
    }

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"maquina")
    public Set<Operacao> getOperacoes() {
        return this.operacoes;
    }

    public void setOperacoes(Set<Operacao> operacoes) {
        this.operacoes = operacoes;
    }

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"maquina")
    public Set<AgendaManutencao> getAgendaManutencaos() {
        return this.agendaManutencaos;
    }

    public void setAgendaManutencaos(Set<AgendaManutencao>
agendaManutencaos) {
        this.agendaManutencaos = agendaManutencaos;
    }

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"maquina")
    public Set<Manutencao> getManutencaos() {
        return this.manutencaos;
    }

    public void setManutencaos(Set<Manutencao> manutencaos) {
        this.manutencaos = manutencaos;
    }
}

package br.ufsc.inf.faccin.efagro.entity;

```

```
// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1
```

```
import java.util.Date;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import org.hibernate.validator.NotNull;

/**
 * Manutencao generated by hbm2java
 */
@Entity
@Table(name = "manutencao", schema = "public")
public class Manutencao implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -3359581157747027845L;
    private int idManutencao;
    private Maquina maquina;
    private Date dataManutencao;
    private String descricao;
    private double valorTotal;

    public Manutencao() {
    }
}
```

```

    public Manutencao(int idManutencao, Maquina maquina, Date
dataManutencao,
        String descricao, double valorTotal) {
    this.idManutencao = idManutencao;
    this.maquina = maquina;
    this.dataManutencao = dataManutencao;
    this.descricao = descricao;
    this.valorTotal = valorTotal;
}

@Id
@Column(name = "id_manutencao", unique = true, nullable = false)
@SequenceGenerator( name = "MANUTENCAO_ID", sequenceName =
"id_manutencao_seq", allocationSize = 1 )
@GeneratedValue( strategy = GenerationType.AUTO, generator =
"MANUTENCAO_ID" )
@NotNull
public int getIdManutencao() {
    return this.idManutencao;
}

public void setIdManutencao(int idManutencao) {
    this.idManutencao = idManutencao;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_maquina", nullable = false)
@NotNull
public Maquina getMaquina() {
    return this.maquina;
}

public void setMaquina(Maquina maquina) {
    this.maquina = maquina;
}

@Temporal(TemporalType.DATE)
@Column(name = "data_manutencao", nullable = false, length = 13)

```

```

@NotNull
public Date getDataManutencao() {
    return this.dataManutencao;
}

public void setDataManutencao(Date dataManutencao) {
    this.dataManutencao = dataManutencao;
}

@Column(name = "descricao", nullable = false)
@NotNull
public String getDescricao() {
    return this.descricao;
}

public void setDescricao(String descricao) {
    this.descricao = descricao;
}

@Column(name = "valor_total", nullable = false, precision = 17, scale =
17)
@NotNull
public double getValorTotal() {
    return this.valorTotal;
}

public void setValorTotal(double valorTotal) {
    this.valorTotal = valorTotal;
}

}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;

```

```

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import org.hibernate.validator.NotNull;

/**
 * Fertilizante generated by hbm2java
 */
@Entity
@Table(name = "fertilizante", schema = "public")
public class Fertilizante implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -3348624825636372198L;
    private int idFertilizante;
    private Fabricante fabricante;
    private String nome;
    private String propriedades;
    private Set<Adubacao> adubacoes = new HashSet<Adubacao>(0);

    public Fertilizante() {
    }

    public Fertilizante(int idFertilizante, String nome) {
        this.idFertilizante = idFertilizante;
        this.nome = nome;
    }
}

```



```

    public Fertilizante(int idFertilizante, Fabricante fabricante, String
nome,
        String propriedades, Set<Adubacao> adubacoes) {
    this.idFertilizante = idFertilizante;
    this.fabricante = fabricante;
    this.nome = nome;
    this.propriedades = propriedades;
    this.adubacoes = adubacoes;
}

@Id
@Column(name = "id_fertilizante", unique = true, nullable = false)
@SequenceGenerator( name = "FERTILIZANTE_ID", sequenceName =
"id_fertilizante_seq", allocationSize = 1 )
@GeneratedValue( strategy = GenerationType.AUTO, generator =
"FERTILIZANTE_ID" )
@NotNull
public int getIdFertilizante() {
    return this.idFertilizante;
}

public void setIdFertilizante(int idFertilizante) {
    this.idFertilizante = idFertilizante;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_fabricante")
public Fabricante getFabricante() {
    return this.fabricante;
}

public void setFabricante(Fabricante fabricante) {
    this.fabricante = fabricante;
}

@Column(name = "nome", nullable = false)
@NotNull
public String getNome() {

```

```

        return this.nome;
    }

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

    @Column(name = "propriedades")
    public String getPropriedades() {
        return this.propriedades;
    }

    public void setPropriedades(String propriedades) {
        this.propriedades = propriedades;
    }

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"fertilizante")
    public Set<Adubacao> getAdubacoes() {
        return this.adubacoes;
    }

    public void setAdubacoes(Set<Adubacao> adubacoes) {
        this.adubacoes = adubacoes;
    }

}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;

```

```

import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToMany;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import org.hibernate.validator.NotNull;

/**
 * Fabricante generated by hbm2java
 */
@Entity
@Table(name = "fabricante", schema = "public")
public class Fabricante implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 4803402979105283198L;
    private int idFabricante;
    private String nomeFabricante;
    private Set<Variedade> variedades = new HashSet<Variedade>(0);
    private Set<Fertilizante> fertilizantes = new HashSet<Fertilizante>(0);
    private Set<Defensivo> defensivos = new HashSet<Defensivo>(0);

    public Fabricante() {
    }

    public Fabricante(int idFabricante, String nomeFabricante) {
        this.idFabricante = idFabricante;
        this.nomeFabricante = nomeFabricante;
    }

    public Fabricante(int idFabricante, String nomeFabricante,
        Set<Variedade> variedades, Set<Fertilizante> fertilizantes,
        Set<Defensivo> defensivos) {
        this.idFabricante = idFabricante;
        this.nomeFabricante = nomeFabricante;
        this.variedades = variedades;
    }
}

```

```

        this.fertilizantes = fertilizantes;
        this.defensivos = defensivos;
    }

    @Id
    @Column(name = "id_fabricante", unique = true, nullable = false)
    @SequenceGenerator( name = "FABRICANTE_ID", sequenceName =
        "id_fabricante_seq", allocationSize = 1 )
    @GeneratedValue( strategy = GenerationType.AUTO, generator =
        "FABRICANTE_ID" )
    @NotNull
    public int getIdFabricante() {
        return this.idFabricante;
    }

    public void setIdFabricante(int idFabricante) {
        this.idFabricante = idFabricante;
    }

    @Column(name = "nome_fabricante", nullable = false)
    @NotNull
    public String getNomeFabricante() {
        return this.nomeFabricante;
    }

    public void setNomeFabricante(String nomeFabricante) {
        this.nomeFabricante = nomeFabricante;
    }

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
        "fabricante")
    public Set<Variedade> getVariedades() {
        return this.variedades;
    }

    public void setVariedades(Set<Variedade> variedades) {
        this.variedades = variedades;
    }
}

```

```

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"fabricante")
    public Set<Fertilizante> getFertilizantes() {
        return this.fertilizantes;
    }

    public void setFertilizantes(Set<Fertilizante> fertilizantes) {
        this.fertilizantes = fertilizantes;
    }

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"fabricante")
    public Set<Defensivo> getDefensivos() {
        return this.defensivos;
    }

    public void setDefensivos(Set<Defensivo> defensivos) {
        this.defensivos = defensivos;
    }
}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;

```

```

import javax.persistence.OrderBy;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import org.hibernate.validator.NotNull;

/**
 * Estado generated by hbm2java
 */
@Entity
@Table(name = "estado", schema = "public")
public class Estado implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 5974058038537488280L;
    private int idEstado;
    private Pais pais;
    private String nome;
    private String sigla;
    private Set<Cidade> cidades = new HashSet<Cidade>(0);

    public Estado() {
    }

    public Estado(int idEstado, Pais pais, String nome) {
        this.idEstado = idEstado;
        this.pais = pais;
        this.nome = nome;
    }

    public Estado(int idEstado, Pais pais, String nome, String sigla,
        Set<Cidade> cidades) {
        this.idEstado = idEstado;
        this.pais = pais;
        this.nome = nome;
        this.sigla = sigla;
        this.cidades = cidades;
    }

```

```

    }

    @Id
    @Column(name = "id_estado", unique = true, nullable = false)
    @SequenceGenerator( name = "ESTADO_ID", sequenceName = "id_estado_seq",
allocationSize = 1 )
    @GeneratedValue( strategy = GenerationType.AUTO, generator = "ESTADO_ID"
)

    @NotNull
    public int getIdEstado() {
        return this.idEstado;
    }

    public void setIdEstado(int idEstado) {
        this.idEstado = idEstado;
    }

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "cod_pais", nullable = false)
    @NotNull
    public Pais getPais() {
        return this.pais;
    }

    public void setPais(Pais pais) {
        this.pais = pais;
    }

    @Column(name = "nome", nullable = false)
    @NotNull
    @OrderBy
    public String getNome() {
        return this.nome;
    }

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

```

```

@Column(name = "sigla")
public String getSigla() {
    return this.sigla;
}

public void setSigla(String sigla) {
    this.sigla = sigla;
}

@OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"estado")
public Set<Cidade> getCidades() {
    return this.cidades;
}

public void setCidades(Set<Cidade> cidades) {
    this.cidades = cidades;
}
}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.SequenceGenerator;

```



```

import javax.persistence.Table;
import org.hibernate.validator.NotNull;

/**
 * Endereco generated by hbm2java
 */
@Entity
@Table(name = "endereco", schema = "public")
public class Endereco implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -6472267181531518206L;
    private int idEndereco;
    private Cidade cidade;
    private String logradouro;
    private String complemento;
    private String bairro;
    private String codigoPostal;
    private String numero;
    private Set<Propriedade> propriedades = new HashSet<Propriedade>(0);
    private Set<Cliente> clientes = new HashSet<Cliente>(0);

    public Endereco() {
    }

    public Endereco(int idEndereco, Cidade cidade) {
        this.idEndereco = idEndereco;
        this.cidade = cidade;
    }

    public Endereco(int idEndereco, Cidade cidade, String logradouro,
        String complemento, String bairro, String codigoPostal,
        String numero, Set<Propriedade> propriedades, Set<Cliente>
clientes) {
        this.idEndereco = idEndereco;
        this.cidade = cidade;
    }

```

```

        this.logradouro = logradouro;
        this.complemento = complemento;
        this.bairro = bairro;
        this.codigoPostal = codigoPostal;
        this.numero = numero;
        this.propriedades = propriedades;
        this.clientes = clientes;
    }

    @Id
    @Column(name = "id_endereco", unique = true, nullable = false)
    @SequenceGenerator( name = "ENDERECO_ID", sequenceName =
        "id_endereco_seq", allocationSize = 1 )
    @GeneratedValue( strategy = GenerationType.AUTO, generator =
        "ENDERECO_ID" )
    @NotNull
    public int getIdEndereco() {
        return this.idEndereco;
    }

    public void setIdEndereco(int idEndereco) {
        this.idEndereco = idEndereco;
    }

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "cod_cidade", nullable = false)
    @NotNull
    public Cidade getCidade() {
        return this.cidade;
    }

    public void setCidade(Cidade cidade) {
        this.cidade = cidade;
    }

    @Column(name = "logradouro")
    public String getLogradouro() {
        return this.logradouro;
    }

```

```

}

public void setLogradouro(String logradouro) {
    this.logradouro = logradouro;
}

@Column(name = "complemento")
public String getComplemento() {
    return this.complemento;
}

public void setComplemento(String complemento) {
    this.complemento = complemento;
}

@Column(name = "bairro")
public String getBairro() {
    return this.bairro;
}

public void setBairro(String bairro) {
    this.bairro = bairro;
}

@Column(name = "codigo_postal")
public String getCodigoPostal() {
    return this.codigoPostal;
}

public void setCodigoPostal(String codigoPostal) {
    this.codigoPostal = codigoPostal;
}

@Column(name = "numero")
public String getNumero() {
    return this.numero;
}

```

```

    public void setNumero(String numero) {
        this.numero = numero;
    }

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"endereco")
    public Set<Propriedade> getPropriedades() {
        return this.propriedades;
    }

    public void setPropriedades(Set<Propriedade> propriedades) {
        this.propriedades = propriedades;
    }

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"endereco")
    public Set<Cliente> getClientes() {
        return this.clientes;
    }

    public void setClientes(Set<Cliente> clientes) {
        this.clientes = clientes;
    }

}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

```

```

import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToOne;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import org.hibernate.validator.NotNull;

/**
 * Defensivo generated by hbm2java
 */
@Entity
@Table(name = "defensivo", schema = "public")
public class Defensivo implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -3837015189260003600L;
    private int idDefensivo;
    private Fabricante fabricante;
    private String nome;
    private String funcao;
    private String formula;
    private Set<AplicacaoDefensivo> aplicacaoDefensivos = new
HashSet<AplicacaoDefensivo>(
        0);

    public Defensivo() {
    }

    public Defensivo(int idDefensivo, String nome, String funcao) {
        this.idDefensivo = idDefensivo;
        this.nome = nome;
        this.funcao = funcao;
    }

    public Defensivo(int idDefensivo, Fabricante fabricante, String nome,
        String funcao, String formula,

```

```

        Set<AplicacaoDefensivo> aplicacaoDefensivos) {
    this.idDefensivo = idDefensivo;
    this.fabricante = fabricante;
    this.nome = nome;
    this.funcao = funcao;
    this.formula = formula;
    this.aplicacaoDefensivos = aplicacaoDefensivos;
}

@Id
@Column(name = "id_defensivo", unique = true, nullable = false)
@SequenceGenerator( name = "DEFENSIVO_ID", sequenceName =
"id_defensivo_seq", allocationSize = 1 )
@GeneratedValue( strategy = GenerationType.AUTO, generator =
"DEFENSIVO_ID" )
@NotNull
public int getIdDefensivo() {
    return this.idDefensivo;
}

public void setIdDefensivo(int idDefensivo) {
    this.idDefensivo = idDefensivo;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_fabricante")
public Fabricante getFabricante() {
    return this.fabricante;
}

public void setFabricante(Fabricante fabricante) {
    this.fabricante = fabricante;
}

@Column(name = "nome", nullable = false)
@NotNull
public String getNome() {
    return this.nome;
}

```

```

    }

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

    @Column(name = "funcao", nullable = false)
    @NotNull
    public String getFuncao() {
        return this.funcao;
    }

    public void setFuncao(String funcao) {
        this.funcao = funcao;
    }

    @Column(name = "formula")
    public String getFormula() {
        return this.formula;
    }

    public void setFormula(String formula) {
        this.formula = formula;
    }

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"defensivo")
    public Set<AplicacaoDefensivo> getAplicacaoDefensivos() {
        return this.aplicacaoDefensivos;
    }

    public void setAplicacaoDefensivos(
        Set<AplicacaoDefensivo> aplicacaoDefensivos) {
        this.aplicacaoDefensivos = aplicacaoDefensivos;
    }

}

package br.ufsc.inf.faccin.efagro.entity;

```

```

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import org.hibernate.validator.NotNull;

/**
 * Cultura generated by hbm2java
 */
@Entity
@Table(name = "cultura", schema = "public")
public class Cultura implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 2443150215153071582L;
    private int idCultura;
    private String nome;
    private String descricao;
    private Set<Variedade> variedades = new HashSet<Variedade>(0);

    public Cultura() {
    }

    public Cultura(int idCultura, String nome) {
        this.idCultura = idCultura;
    }
}

```



```

        this.nome = nome;
    }

    public Cultura(int idCultura, String nome, String descricao,
        Set<Variedade> variedades) {
        this.idCultura = idCultura;
        this.nome = nome;
        this.descricao = descricao;
        this.variedades = variedades;
    }

    @Id
    @Column(name = "id_cultura", unique = true, nullable = false)
    @SequenceGenerator( name = "CULTURA_ID", sequenceName =
        "id_cultura_seq", allocationSize = 1 )
    @GeneratedValue( strategy = GenerationType.AUTO, generator =
        "CULTURA_ID" )
    @NotNull
    public int getIdCultura() {
        return this.idCultura;
    }

    public void setIdCultura(int idCultura) {
        this.idCultura = idCultura;
    }

    @Column(name = "nome", nullable = false)
    @NotNull
    public String getNome() {
        return this.nome;
    }

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

    @Column(name = "descricao")
    public String getDescricao() {

```

```

        return this.descricao;
    }

    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"cultura")
    public Set<Variedade> getVariedades() {
        return this.variedades;
    }

    public void setVariedades(Set<Variedade> variedades) {
        this.variedades = variedades;
    }
}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.OneToOne;
import javax.persistence.Table;

import org.hibernate.validator.NotNull;

/**
 * Colheita generated by hbm2java
 */
@Entity
@Table(name = "colheita", schema = "public")
public class Colheita implements java.io.Serializable {

```

```

/**
 *
 */
private static final long serialVersionUID = 4860647935056253685L;
private long codAtividadeTalhao;
private AtividadeTalhao atividadeTalhao;
private double quantidade;
private double valorTotal;

public Colheita() {
}

public Colheita(long codAtividadeTalhao, AtividadeTalhao
atividadeTalhao,
        double quantidade, double valorTotal) {
    this.codAtividadeTalhao = codAtividadeTalhao;
    this.atividadeTalhao = atividadeTalhao;
    this.quantidade = quantidade;
    this.valorTotal = valorTotal;
}

@Id
@Column(name = "cod_atividade_talhao", unique = true, nullable = false)
@NotNull
public long getCodAtividadeTalhao() {
    return this.codAtividadeTalhao;
}

public void setCodAtividadeTalhao(long codAtividadeTalhao) {
    this.codAtividadeTalhao = codAtividadeTalhao;
}

@OneToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_atividade_talhao", unique = true, nullable =
false, insertable = false, updatable = false)
@NotNull
public AtividadeTalhao getAtividadeTalhao() {

```

```

        return this.atividadeTalhao;
    }

    public void setAtividadeTalhao(AtividadeTalhao atividadeTalhao) {
        this.atividadeTalhao = atividadeTalhao;
    }

    @Column(name = "quantidade", nullable = false, precision = 17, scale =
17)
    @NotNull
    public double getQuantidade() {
        return this.quantidade;
    }

    public void setQuantidade(double quantidade) {
        this.quantidade = quantidade;
    }

    @Column(name = "valor_total", nullable = false, precision = 17, scale =
17)
    @NotNull
    public double getValorTotal() {
        return this.valorTotal;
    }

    public void setValorTotal(double valorTotal) {
        this.valorTotal = valorTotal;
    }
}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.Date;
import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;

```

```

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.persistence.UniqueConstraint;
import org.hibernate.validator.NotNull;

/**
 * Cliente generated by hbm2java
 */
@Entity
@Table(name = "cliente", schema = "public", uniqueConstraints =
@UniqueConstraint(columnNames = "email"))
public class Cliente implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 5791694564303369806L;
    private int idCliente;
    private Endereco endereco;
    private String nome;
    private String email;
    private String celular;
    private String telefone;
    private char sexo;
    private Date dataNascimento;
    private String cpf;
    private Set<Propriedade> propriedades = new HashSet<Propriedade>(0);

```

```

public Cliente() {
}

public Cliente(int idCliente, Endereco endereco, String nome, String
email,
        Date dataNascimento, String cpf) {
    this.idCliente = idCliente;
    this.endereco = endereco;
    this.nome = nome;
    this.email = email;
    this.dataNascimento = dataNascimento;
    this.cpf = cpf;
}

public Cliente(int idCliente, Endereco endereco, String nome, String
email,
        String celular, String telefone, char sexo,
        Date dataNascimento, String cpf, Set<Propriedade>
propriedades) {
    this.idCliente = idCliente;
    this.endereco = endereco;
    this.nome = nome;
    this.email = email;
    this.celular = celular;
    this.telefone = telefone;
    this.sexo = sexo;
    this.dataNascimento = dataNascimento;
    this.cpf = cpf;
    this.propriedades = propriedades;
}

@Id
@Column(name = "id_cliente", unique = true, nullable = false)
@SequenceGenerator( name = "CLIENTE_ID", sequenceName =
"id_cliente_seq", allocationSize = 1 )
@GeneratedValue( strategy = GenerationType.AUTO, generator =
"CLIENTE_ID" )

```

```

@NotNull
public int getIdCliente() {
    return this.idCliente;
}

public void setIdCliente(int idCliente) {
    this.idCliente = idCliente;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_endereco", nullable = false)
@NotNull
public Endereco getEndereco() {
    return this.endereco;
}

public void setEndereco(Endereco endereco) {
    this.endereco = endereco;
}

@Column(name = "nome", nullable = false)
@NotNull
public String getNome() {
    return this.nome;
}

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

@Column(name = "email", unique = true, nullable = false)
@NotNull
public String getEmail() {
    return this.email;
}

public void setEmail(String email) {
    this.email = email;
}

```

```

}

@Column(name = "celular")
public String getCelular() {
    return this.celular;
}

public void setCelular(String celular) {
    this.celular = celular;
}

@Column(name = "telefono")
public String getTelefono() {
    return this.telefono;
}

public void setTelefono(String telefono) {
    this.telefono = telefono;
}

@Column(name = "sexo", length = 1)
public char getSexo() {
    return this.sexo;
}

public void setSexo(char sexo) {
    this.sexo = sexo;
}

@Temporal(TemporalType.DATE)
@Column(name = "data_nascimento", nullable = false, length = 13)
@NotNull
public Date getDataNascimento() {
    return this.dataNascimento;
}

public void setDataNascimento(Date dataNascimento) {
    this.dataNascimento = dataNascimento;
}

```



```

    }

    @Column(name = "cpf", nullable = false)
    @NotNull
    public String getCpf() {
        return this.cpf;
    }

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

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"cliente")
    public Set<Propriedade> getPropriedades() {
        return this.propriedades;
    }

    public void setPropriedades(Set<Propriedade> propriedades) {
        this.propriedades = propriedades;
    }
}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;

```

```

import javax.persistence.OneToOne;
import javax.persistence.OrderBy;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import org.hibernate.validator.NotNull;

/**
 * Cidade generated by hbm2java
 */
@Entity
@Table(name = "cidade", schema = "public")
public class Cidade implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 5880828211494386565L;
    private int idCidade;
    private Estado estado;
    private String nome;
    private Set<Endereco> enderecos = new HashSet<Endereco>(0);

    public Cidade() {
    }

    public Cidade(int idCidade, Estado estado, String nome) {
        this.idCidade = idCidade;
        this.estado = estado;
        this.nome = nome;
    }

    public Cidade(int idCidade, Estado estado, String nome,
        Set<Endereco> enderecos) {
        this.idCidade = idCidade;
        this.estado = estado;
        this.nome = nome;
        this.enderecos = enderecos;
    }
}

```

```

@Id
@Column(name = "id_cidade", unique = true, nullable = false)
@SequenceGenerator( name = "CIDADE_ID", sequenceName = "id_cidade_seq",
allocationSize = 1 )
@GeneratedValue( strategy = GenerationType.AUTO, generator = "CIDADE_ID"
)

@NotNull
public int getIdCidade() {
    return this.idCidade;
}

public void setIdCidade(int idCidade) {
    this.idCidade = idCidade;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_estado", nullable = false)
@NotNull
public Estado getEstado() {
    return this.estado;
}

public void setEstado(Estado estado) {
    this.estado = estado;
}

@Column(name = "nome", nullable = false)
@NotNull
@OrderBy
public String getNome() {
    return this.nome;
}

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

```

```

        @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"cidade")
        public Set<Endereco> getEnderecos() {
            return this.enderecos;
        }

        public void setEnderecos(Set<Endereco> enderecos) {
            this.enderecos = enderecos;
        }
    }

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.Date;
import java.util.HashSet;
import java.util.Set;
import javax.persistence.CascadeType;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToMany;
import javax.persistence.OneToOne;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import org.hibernate.validator.NotNull;

/**
 * AtividadeTalhao generated by hbm2java
 */

```

```

@Entity
@Table(name = "atividade_talhao", schema = "public")
public class AtividadeTalhao implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 6079155061949401718L;
    private long idAtividadeTalhao;
    private Talhao talhao;
    private Date dataAtividade;
    private String observacao;
    private AplicacaoDefensivo aplicacaoDefensivo;
    private Plantio plantio;
    private Colheita colheita;
    private Adubacao adubacao;
    private Set<Operacao> operacoes = new HashSet<Operacao>(0);

    public AtividadeTalhao() {
    }

    public AtividadeTalhao(long idAtividadeTalhao, Talhao talhao,
        Date dataAtividade) {
        this.idAtividadeTalhao = idAtividadeTalhao;
        this.talhao = talhao;
        this.dataAtividade = dataAtividade;
    }

    public AtividadeTalhao(long idAtividadeTalhao, Talhao talhao,
        Date dataAtividade, String observacao,
        AplicacaoDefensivo aplicacaoDefensivo, Plantio plantio,
        Colheita colheita, Adubacao adubacao, Set<Operacao>
operacoes) {
        this.idAtividadeTalhao = idAtividadeTalhao;
        this.talhao = talhao;
        this.dataAtividade = dataAtividade;
        this.observacao = observacao;
        this.aplicacaoDefensivo = aplicacaoDefensivo;
    }
}

```

```

        this.plantio = plantio;
        this.colheita = colheita;
        this.adubacao = adubacao;
        this.operacoes = operacoes;
    }

    @Id
    @Column(name = "id_atividade_talhao", unique = true, nullable = false)
    @SequenceGenerator( name = "ATIVIDADE_TALHAO_ID", sequenceName =
        "id_atividade_talhao_seq", allocationSize = 1 )
    @GeneratedValue( strategy = GenerationType.AUTO, generator =
        "ATIVIDADE_TALHAO_ID" )
    @NotNull
    public long getIdAtividadeTalhao() {
        return this.idAtividadeTalhao;
    }

    public void setIdAtividadeTalhao(long idAtividadeTalhao) {
        this.idAtividadeTalhao = idAtividadeTalhao;
    }

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "cod_talhao", nullable = false)
    @NotNull
    public Talhao getTalhao() {
        return this.talhao;
    }

    public void setTalhao(Talhao talhao) {
        this.talhao = talhao;
    }

    @Temporal(TemporalType.DATE)
    @Column(name = "data_atividade", nullable = false, length = 13)
    @NotNull
    public Date getDataAtividade() {
        return this.dataAtividade;
    }
}

```

```

public void setDataAtividade(Date dataAtividade) {
    this.dataAtividade = dataAtividade;
}

@Column(name = "observacao")
public String getObservacao() {
    return this.observacao;
}

public void setObservacao(String observacao) {
    this.observacao = observacao;
}

@OneToOne(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"atividadeTalhao")
public AplicacaoDefensivo getAplicacaoDefensivo() {
    return this.aplicacaoDefensivo;
}

public void setAplicacaoDefensivo(AplicacaoDefensivo aplicacaoDefensivo)
{
    this.aplicacaoDefensivo = aplicacaoDefensivo;
}

@OneToOne(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"atividadeTalhao")
public Plantio getPlantio() {
    return this.plantio;
}

public void setPlantio(Plantio plantio) {
    this.plantio = plantio;
}

@OneToOne(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"atividadeTalhao")
public Colheita getColheita() {

```

```

        return this.colheita;
    }

    public void setColheita(Colheita colheita) {
        this.colheita = colheita;
    }

    @OneToOne(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"atividadeTalhao")
    public Adubacao getAdubacao() {
        return this.adubacao;
    }

    public void setAdubacao(Adubacao adubacao) {
        this.adubacao = adubacao;
    }

    @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy =
"atividadeTalhao")
    public Set<Operacao> getOperacoes() {
        return this.operacoes;
    }

    public void setOperacoes(Set<Operacao> operacoes) {
        this.operacoes = operacoes;
    }
}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.Date;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;

```



```

import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToOne;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

import org.hibernate.validator.NotNull;

/**
 * AplicacaoDefensivo generated by hbm2java
 */
@Entity
@Table(name = "aplicacao_defensivo", schema = "public")
public class AplicacaoDefensivo implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -6495503899273300449L;
    private long codAtividadeTalhao;
    private AtividadeTalhao atividadeTalhao;
    private Defensivo defensivo;
    private Date hora;
    private double quantidade;
    private Date carencia;
    private Double umidade;
    private Double temperatura;
    private double valorTotal;

    public AplicacaoDefensivo() {
    }

    public AplicacaoDefensivo(long codAtividadeTalhao,
        AtividadeTalhao atividadeTalhao, Defensivo defensivo, Date
hora,
        double quantidade, Date carencia, double valorTotal) {
        this.codAtividadeTalhao = codAtividadeTalhao;

```

```

        this.atividadeTalhao = atividadeTalhao;
        this.defensivo = defensivo;
        this.hora = hora;
        this.quantidade = quantidade;
        this.carencia = carencia;
        this.valorTotal = valorTotal;
    }

    public AplicacaoDefensivo(long codAtividadeTalhao,
        AtividadeTalhao atividadeTalhao, Defensivo defensivo, Date
hora,
        double quantidade, Date carencia, Double umidade,
        Double temperatura, double valorTotal) {
        this.codAtividadeTalhao = codAtividadeTalhao;
        this.atividadeTalhao = atividadeTalhao;
        this.defensivo = defensivo;
        this.hora = hora;
        this.quantidade = quantidade;
        this.carencia = carencia;
        this.umidade = umidade;
        this.temperatura = temperatura;
        this.valorTotal = valorTotal;
    }

    @Id
    @Column(name = "cod_atividade_talhao", unique = true, nullable = false)
    @NotNull
    public long getCodAtividadeTalhao() {
        return this.codAtividadeTalhao;
    }

    public void setCodAtividadeTalhao(long codAtividadeTalhao) {
        this.codAtividadeTalhao = codAtividadeTalhao;
    }

    @OneToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "cod_atividade_talhao", unique = true, nullable =
false, insertable = false, updatable = false)

```

```

@NotNull
public AtividadeTalhao getAtividadeTalhao() {
    return this.atividadeTalhao;
}

public void setAtividadeTalhao(AtividadeTalhao atividadeTalhao) {
    this.atividadeTalhao = atividadeTalhao;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_defensivo", nullable = false)
@NotNull
public Defensivo getDefensivo() {
    return this.defensivo;
}

public void setDefensivo(Defensivo defensivo) {
    this.defensivo = defensivo;
}

@Temporal(TemporalType.TIME)
@Column(name = "hora", nullable = false, length = 15)
@NotNull
public Date getHora() {
    return this.hora;
}

public void setHora(Date hora) {
    this.hora = hora;
}

@Column(name = "quantidade", nullable = false, precision = 17, scale =
17)
@NotNull
public double getQuantidade() {
    return this.quantidade;
}

```

```

public void setQuantidade(double quantidade) {
    this.quantidade = quantidade;
}

@Temporal(TemporalType.DATE)
@Column(name = "carencia", nullable = false, length = 13)
@NotNull
public Date getCarencia() {
    return this.carencia;
}

public void setCarencia(Date carencia) {
    this.carencia = carencia;
}

@Column(name = "umidade", precision = 17, scale = 17)
public Double getUmidade() {
    return this.umidade;
}

public void setUmidade(Double umidade) {
    this.umidade = umidade;
}

@Column(name = "temperatura", precision = 17, scale = 17)
public Double getTemperatura() {
    return this.temperatura;
}

public void setTemperatura(Double temperatura) {
    this.temperatura = temperatura;
}

@Column(name = "valor_total", nullable = false, precision = 17, scale =
17)
@NotNull
public double getValorTotal() {
    return this.valorTotal;
}

```

```

    }

    public void setValorTotal(double valorTotal) {
        this.valorTotal = valorTotal;
    }

}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import java.util.Date;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import org.hibernate.validator.NotNull;

/**
 * AgendaManutencao generated by hbm2java
 */
@Entity
@Table(name = "agenda_manutencao", schema = "public")
public class AgendaManutencao implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 4726699474224636844L;
    private int idAgendaManutencao;

```

```

private Maquina maquina;
private Date dataManutencao;
private String lembrete;
private Date dataAviso;
private int frequencia;

public AgendaManutencao() {
}

public AgendaManutencao(int idAgendaManutencao, Maquina maquina,
    Date dataManutencao, Date dataAviso, int frequencia) {
    this.idAgendaManutencao = idAgendaManutencao;
    this.maquina = maquina;
    this.dataManutencao = dataManutencao;
    this.dataAviso = dataAviso;
    this.frequencia = frequencia;
}

public AgendaManutencao(int idAgendaManutencao, Maquina maquina,
    Date dataManutencao, String lembrete, Date dataAviso, int
frequencia) {
    this.idAgendaManutencao = idAgendaManutencao;
    this.maquina = maquina;
    this.dataManutencao = dataManutencao;
    this.lembrete = lembrete;
    this.dataAviso = dataAviso;
    this.frequencia = frequencia;
}

@Id
@Column(name = "id_agenda_manutencao", unique = true, nullable = false)
@SequenceGenerator( name = "AGENDA_MANUTENCAO_ID", sequenceName =
"id_agenda_manutencao_seq", allocationSize = 1 )
@GeneratedValue( strategy = GenerationType.AUTO, generator =
"AGENDA_MANUTENCAO_ID" )
@NotNull
public int getIdAgendaManutencao() {
    return this.idAgendaManutencao;
}

```

```

}

public void setIdAgendaManutencao(int idAgendaManutencao) {
    this.idAgendaManutencao = idAgendaManutencao;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_maquina", nullable = false)
@NotNull
public Maquina getMaquina() {
    return this.maquina;
}

public void setMaquina(Maquina maquina) {
    this.maquina = maquina;
}

@Temporal(TemporalType.DATE)
@Column(name = "data_manutencao", nullable = false, length = 13)
@NotNull
public Date getDataManutencao() {
    return this.dataManutencao;
}

public void setDataManutencao(Date dataManutencao) {
    this.dataManutencao = dataManutencao;
}

@Column(name = "lembrete")
public String getLembrete() {
    return this.lembrete;
}

public void setLembrete(String lembrete) {
    this.lembrete = lembrete;
}

@Temporal(TemporalType.DATE)

```

```

@Column(name = "data_aviso", nullable = false, length = 13)
@NotNull
public Date getDataAviso() {
    return this.dataAviso;
}

public void setDataAviso(Date dataAviso) {
    this.dataAviso = dataAviso;
}

@Column(name = "frequencia", nullable = false)
@NotNull
public int getFrequencia() {
    return this.frequencia;
}

public void setFrequencia(int frequencia) {
    this.frequencia = frequencia;
}
}

package br.ufsc.inf.faccin.efagro.entity;

// Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.OneToOne;
import javax.persistence.Table;

import org.hibernate.validator.NotNull;

/**
 * Aduacao generated by hbm2java

```



```

    */
@Entity
@Table(name = "adubacao", schema = "public")
public class Adubacao implements java.io.Serializable {

    /**
     *
     */
    private static final long serialVersionUID = -316566037091566945L;
    private long codAtividadeTalhao;
    private AtividadeTalhao atividadeTalhao;
    private Fertilizante fertilizante;
    private double quantidade;
    private double valorTotal;

    public Adubacao() {
    }

    public Adubacao(long codAtividadeTalhao, AtividadeTalhao
atividadeTalhao,
        Fertilizante fertilizante, double quantidade, double
valorTotal) {
        this.codAtividadeTalhao = codAtividadeTalhao;
        this.atividadeTalhao = atividadeTalhao;
        this.fertilizante = fertilizante;
        this.quantidade = quantidade;
        this.valorTotal = valorTotal;
    }

    @Id
    @Column(name = "cod_atividade_talhao", unique = true, nullable = false)
    @NotNull
    public long getCodAtividadeTalhao() {
        return this.codAtividadeTalhao;
    }

    public void setCodAtividadeTalhao(long codAtividadeTalhao) {
        this.codAtividadeTalhao = codAtividadeTalhao;
    }

```

```

    }

    @OneToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "cod_atividade_talhao", unique = true, nullable =
false, insertable = false, updatable = false)
    @NotNull
    public AtividadeTalhao getAtividadeTalhao() {
        return this.atividadeTalhao;
    }

    public void setAtividadeTalhao(AtividadeTalhao atividadeTalhao) {
        this.atividadeTalhao = atividadeTalhao;
    }

    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "cod_fertilizante", nullable = false)
    @NotNull
    public Fertilizante getFertilizante() {
        return this.fertilizante;
    }

    public void setFertilizante(Fertilizante fertilizante) {
        this.fertilizante = fertilizante;
    }

    @Column(name = "quantidade", nullable = false, precision = 17, scale =
17)
    @NotNull
    public double getQuantidade() {
        return this.quantidade;
    }

    public void setQuantidade(double quantidade) {
        this.quantidade = quantidade;
    }

    @Column(name = "valor_total", nullable = false, precision = 17, scale =
17)

```

```

        @NotNull
        public double getValorTotal() {
            return this.valorTotal;
        }

        public void setValorTotal(double valorTotal) {
            this.valorTotal = valorTotal;
        }
    }

    package br.ufsc.inf.faccin.efagro.entity;

    // Generated 21/04/2010 18:08:44 by Hibernate Tools 3.2.4.CR1

    import java.util.Date;
    import javax.persistence.Column;
    import javax.persistence.Entity;
    import javax.persistence.FetchType;
    import javax.persistence.GeneratedValue;
    import javax.persistence.GenerationType;
    import javax.persistence.Id;
    import javax.persistence.JoinColumn;
    import javax.persistence.ManyToOne;
    import javax.persistence.SequenceGenerator;
    import javax.persistence.Table;
    import javax.persistence.Temporal;
    import javax.persistence.TemporalType;
    import org.hibernate.validator.NotNull;

    /**
     * Abastecimento generated by hbm2java
     */
    @Entity
    @Table(name = "abastecimento", schema = "public")
    public class Abastecimento implements java.io.Serializable {

        /**
         *

```

```

    */
    private static final long serialVersionUID = 2228294041135761955L;
    private int idAbastecimento;
    private Maquina maquina;
    private Date dataAbastecimento;
    private String tipoCombustivel;
    private double quantidade;
    private double valorUnitario;
    private Double horimetroHodometro;

    public Abastecimento() {
    }

    public Abastecimento(int idAbastecimento, Maquina maquina,
        Date dataAbastecimento, String tipoCombustivel, double
quantidade,
        double valorUnitario) {
        this.idAbastecimento = idAbastecimento;
        this.maquina = maquina;
        this.dataAbastecimento = dataAbastecimento;
        this.tipoCombustivel = tipoCombustivel;
        this.quantidade = quantidade;
        this.valorUnitario = valorUnitario;
    }

    public Abastecimento(int idAbastecimento, Maquina maquina,
        Date dataAbastecimento, String tipoCombustivel, double
quantidade,
        double valorUnitario, Double horimetroHodometro) {
        this.idAbastecimento = idAbastecimento;
        this.maquina = maquina;
        this.dataAbastecimento = dataAbastecimento;
        this.tipoCombustivel = tipoCombustivel;
        this.quantidade = quantidade;
        this.valorUnitario = valorUnitario;
        this.horimetroHodometro = horimetroHodometro;
    }
}

```

```

@Id
@SequenceGenerator( name = "ABASTECIMIENTO_ID", sequenceName =
" id_abastecimento_seq", allocationSize = 1 )
@GeneratedValue( strategy = GenerationType.AUTO, generator =
"ABASTECIMIENTO_ID" )
@Column(name = "id_abastecimento", unique = true, nullable = false)
@NotNull
public int getIdAbastecimento() {
    return this.idAbastecimento;
}

public void setIdAbastecimento(int idAbastecimento) {
    this.idAbastecimento = idAbastecimento;
}

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "cod_maquina", nullable = false)
@NotNull
public Maquina getMaquina() {
    return this.maquina;
}

public void setMaquina(Maquina maquina) {
    this.maquina = maquina;
}

@Temporal(TemporalType.DATE)
@Column(name = "data_abastecimento", nullable = false, length = 13)
@NotNull
public Date getDataAbastecimento() {
    return this.dataAbastecimento;
}

public void setDataAbastecimento(Date dataAbastecimento) {
    this.dataAbastecimento = dataAbastecimento;
}

@Column(name = "tipo_combustivel", nullable = false)

```

```

@NotNull
public String getTipoCombustivel() {
    return this.tipoCombustivel;
}

public void setTipoCombustivel(String tipoCombustivel) {
    this.tipoCombustivel = tipoCombustivel;
}

@Column(name = "quantidade", nullable = false, precision = 17, scale =
17)
@NotNull
public double getQuantidade() {
    return this.quantidade;
}

public void setQuantidade(double quantidade) {
    this.quantidade = quantidade;
}

@Column(name = "valor_unitario", nullable = false, precision = 17, scale
= 17)
@NotNull
public double getValorUnitario() {
    return this.valorUnitario;
}

public void setValorUnitario(double valorUnitario) {
    this.valorUnitario = valorUnitario;
}

@Column(name = "horimetro_hodometro", precision = 17, scale = 17)
public Double getHorimetroHodometro() {
    return this.horimetroHodometro;
}

public void setHorimetroHodometro(Double horimetroHodometro) {
    this.horimetroHodometro = horimetroHodometro;
}

```

```

    }

}

package br.ufsc.inf.faccin.efagro.tools;

import org.jboss.seam.annotations.Name;

@Name("constantes")
public class Constantes {

    public static final String ABASTECIMENTO_INCLUIDO_ERRO =
"ABASTECIMENTO_INCLUIDO_ERRO";
    public static final String ABASTECIMENTO_INCLUIDO_OK =
"ABASTECIMENTO_INCLUIDO_OK";
    public static final String ABASTECIMENTO_REMOVIDO_ERRO =
"ABASTECIMENTO_REMOVIDO_ERRO";
    public static final String ABASTECIMENTO_REMOVIDO_OK =
"ABASTECIMENTO_REMOVIDO_OK";
    public static final String ABASTECIMENTO_ALTERADO_ERRO =
"ABASTECIMENTO_ALTERADO_ERRO";
    public static final String ABASTECIMENTO_ALTERADO_OK =
"ABASTECIMENTO_ALTERADO_OK";
    public static final String ADUBACAO_INCLUIDO_ERRO =
"ADUBACAO_INCLUIDO_ERRO";
    public static final String ADUCACAO_INCLUIDO_OK =
"ADUBACAO_INCLUIDO_ERRO";
    public static final String AGENDA_MANUTENCAO_INCLUIDO_ERRO =
"AGENDA_MANUTENCAO_INCLUIDO_ERRO";
    public static final String AGENDA_MANUTENCAO_INCLUIDO_OK =
"AGENDA_MANUTENCAO_INCLUIDO_OK";
    public static final String AP_DEFENSIVO_INCLUIDO_ERRO =
"AP_DEFENSIVO_INCLUIDO_ERRO";
    public static final String AP_DEFENSIVO_INCLUIDO_OK =
"AP_DEFENSIVO_INCLUIDO_OK";
    public static final String ATIVIDADE_INCLUIDO_ERRO =
"ATIVIDADE_INCLUIDO_ERRO";

```

```

    public static final String ATIVIDADE_INCLUIDO_OK =
"ATIVIDADE_INCLUIDO_OK";
    public static final String COLHEITA_INCLUIDO_ERRO =
"COLHEITA_INCLUIDO_ERRO";
    public static final String COLHEITA_INCLUIDO_OK =
"COLHEITA_INCLUIDO_OK";
    public static final String MANUTENCAO_INCLUIDO_ERRO =
"MANUTENCAO_INCLUIDO_ERRO";
    public static final String MANUTENCAO_INCLUIDO_OK =
"MANUTENCAO_INCLUIDO_OK";
    public static final String MAQUINA_INCLUIDO_ERRO =
"MAQUINA_INCLUIDO_ERRO";
    public static final String MAQUINA_INCLUIDO_OK = "MAQUINA_INCLUIDO_OK";
    public static final String OCORRENCIA_PRAGA_INCLUIDO_ERRO =
"OCORRENCIA_PRAGA_INCLUIDO_ERRO";
    public static final String OCORRENCIA_PRAGA_INCLUIDO_OK =
"OCORRENCIA_PRAGA_INCLUIDO_OK";
    public static final String OPERACAO_INCLUIDO_ERRO =
"OPERACAO_INCLUIDO_ERRO";
    public static final String OPERACAO_INCLUIDO_OK =
"OPERACAO_INCLUIDO_OK";
    public static final String PLANTIO_INCLUIDO_ERRO =
"PLANTIO_INCLUIDO_ERRO";
    public static final String PLANTIO_INCLUIDO_OK = "PLANTIO_INCLUIDO_OK";
    public static final String PROPRIEDADE_INCLUIDO_ERRO =
"PROPRIEDADE_INCLUIDO_ERRO";
    public static final String PROPRIEDADE_INCLUIDO_OK =
"PROPRIEDADE_INCLUIDO_OK";
    public static final String SAFRA_INCLUIDO_ERRO = "SAFRA_INCLUIDO_ERRO";
    public static final String SAFRA_INCLUIDO_OK = "SAFRA_INCLUIDO_OK";
    public static final String TALHAO_INCLUIDO_ERRO =
"TALHAO_INCLUIDO_ERRO";
    public static final String TALHAO_INCLUIDO_OK = "TALHAO_INCLUIDO_OK";
    public static final String USUARIO_INCLUIDO_ERRO =
"USUARIO_INCLUIDO_ERRO";
    public static final String USUARIO_INCLUIDO_OK = "USUARIO_INCLUIDO_OK";
    public static final String CLIENTE_INCLUIDO_ERRO =
"CLIENTE_INCLUIDO_ERRO";

```



```

    public static final String CLIENTE_INCLUIDO_OK = "CLIENTE_INCLUIDO_OK";
    public static final String SENHA_ALTERADA = "SENHA_ALTERADA";
    public static final String PROPRIEDADE_ALTERADA_ERRO =
"PROPRIEDADE_ALTERADA_ERRO";
    public static final String PROPRIEDADE_ALTERADA_OK =
"PROPRIEDADE_ALTERADA_OK";
    public static final String USUARIO_ALTERADO_ERRO =
"USUARIO_ALTERADO_ERRO";
    public static final String USUARIO_ALTERADO_OK = "USUARIO_ALTERADO_OK";
    public static final String USUARIO_REMOVIDO_ERRO =
"USUARIO_REMOVIDO_ERRO";
    public static final String USUARIO_REMOVIDO_OK = "USUARIO_REMOVIDO_OK";
    public static final String MAQUINA_ALTERADA_ERRO =
"MAQUINA_ALTERADA_ERRO";
    public static final String MAQUINA_ALTERADO_OK = "MAQUINA_ALTERADO_OK";
    public static final String REL_OPERACOES_DA_MAQUINA_OK =
"REL_OPERACOES_DA_MAQUINA_OK";
    public static final String REL_ANO_SAFRA_OK = "REL_ANO_SAFRA_OK";

}

package br.ufsc.inf.faccin.efagro.tools;

import java.util.Random;

public class ChaveAleatoria {

    private String [] elementos =
{"A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","
T","U","V","W","x","Y","Z",

"a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t
","u","v","w","x","y","z",

"1","2","3","4","5","6","7","8","9","0",};

    private String chave = "";
    private int tamanhoChave = 32;

```

```

/**
 *
 * Contrutor da chave aleatoria, tamanho 32 caracteres
 *
 * */
public ChaveAleatoria(){
    this.criaChave();
}
/**
 * contrutor passando o tamanho da chave a ser gerada aleatoriamente;
 *
 *
 * @param int; tamanho da chave aleatoria;
 * */
public ChaveAleatoria(int tamanho){
    this.setTamanhoChave(tamanho);
    this.criaChave();
}

public String getChave() {
    return chave;
}

public int getTamanhoChave() {
    return tamanhoChave;
}

public void setTamanhoChave(int tamanhoChave) {
    this.tamanhoChave = tamanhoChave;
}

private void criaChave() {

    Random rand = new Random();

```

```

        for(int i = 0;i<tamanhoChave;i++){
            chave += elementos[(rand.nextInt(elementos.length))];
        }

    }

    public void naoFazNada() {

    }

}

package br.ufsc.inf.faccin.efagro.session.relatorios;

import java.util.ArrayList;
import java.util.Iterator;

import javax.persistence.EntityManager;

import org.jboss.seam.ScopeType;
import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Logger;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.annotations.Scope;
import org.jboss.seam.annotations.security.Restrict;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.log.Log;

import br.ufsc.inf.faccin.efagro.entity.Talhao;
import br.ufsc.inf.faccin.efagro.session.SafraHome;
import br.ufsc.inf.faccin.efagro.session.TalhaoHome;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

@Name("relAnoSafraAction")
@Restrict("#{identity.loggedIn}")
@Scope(ScopeType.PAGE)
public class RelatorioAnoSafraAction{

    /**

```

```

*
*/

@In(create=true)
SafraHome safraHome;

@In(create=true)
TalhaoHome talhaoHome;

@Logger
Log log;

@In
FacesMessages facesMessages;

@In
EntityManager entityManager;

private ArrayList<Talhao> talhoes = new ArrayList<Talhao>();

private double saldoSafra = 0;

private double totalInvestido = 0;

private double totalArrecadado = 0;

public double getTotalInvestido() {
    return totalInvestido;
}

public void setTotalInvestido(double totalInvestido) {
    this.totalInvestido = totalInvestido;
}

public double getTotalArrecadado() {
    return totalArrecadado;
}

```

```

}

public void setTotalArrecadado(double totalArrecadado) {
    this.totalArrecadado = totalArrecadado;
}

public void setSaldoSafra(double saldoSafra) {
    this.saldoSafra = saldoSafra;
}

public double getSaldoSafra() {
    return this.saldoSafra;
}

public void setTalhoes(ArrayList<Talhao> talhoes) {
    this.talhoes = talhoes;
}

public ArrayList<Talhao> getTalhoes() {
    return this.talhoes;
}

public String geraRelatorioDoAnoSafra(){

    talhoes.clear();

    saldoSafra = 0;

    this.somatorioDosTalhoes();

    return Constantes.REL_ANO_SAFRA_OK;
}

private void somatorioDosTalhoes() {

    Iterator<Talhao> it = safraHome.getTalhoes().iterator();

```

```

        while (it.hasNext()) {
            Talhao talhao = (Talhao) it.next();

            talhaoHome.setInstance(talhao);

            talhaoHome.calculaCustosTalhao();

            talhao.setSaldoTalhao(talhaoHome.getSaldoTalhao());

            talhao.setCustoTalhao(talhaoHome.getTotalCustoAdubacao() +
            talhaoHome.getTotalCustoDefensivo() + talhaoHome.getTotalCustoOperacoes() +
            talhaoHome.getTotalCustoPlantio());

            talhao.setArrecadacaoTalhao(talhaoHome.getTotalBrutoColheita());

            totalInvestido = totalInvestido + talhao.getCustoTalhao();

            totalArrecadado = totalArrecadado +
            talhao.getArrecadacaoTalhao();

            saldoSafra = saldoSafra + talhaoHome.getSaldoTalhao();

            talhoes.add(talhao);

        }

    }

}

package br.ufsc.inf.faccin.efagro.session.relatorios;

import java.util.ArrayList;
import java.util.Date;
import java.util.Iterator;

```

```

import java.util.List;

import javax.persistence.EntityManager;

import org.jboss.seam.ScopeType;
import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Logger;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.annotations.Scope;
import org.jboss.seam.annotations.security.Restrict;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.log.Log;

import br.ufsc.inf.faccin.efagro.entity.Operacao;
import br.ufsc.inf.faccin.efagro.session.MaquinaHome;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

@Name("relMaquinaAction")
@Restrict("#{identity.loggedIn}")
@Scope(ScopeType.PAGE)
public class RelatorioMaquinaAction{

    /**
     *
     */

    @In(create=true)
    MaquinaHome maquinaHome;

    @Logger
    Log log;

    @In
    FacesMessages facesMessages;

    @In

```

```

EntityManager entityManager;

private Date dataInicial = null;

private Date dataFinal = null;

private ArrayList<Operacao> operacoes = new ArrayList<Operacao>();

private double totalHoras;

public void setTotalHoras(double totalHoras) {
    this.totalHoras = totalHoras;
}

public double getTotalHoras() {
    return this.totalHoras;
}

public void setDataFinal(Date dataFinal) {
    this.dataFinal = dataFinal;
}

public Date getDataFinal() {
    return this.dataFinal;
}

public void setDataInicial(Date dataInicial) {
    this.dataInicial = dataInicial;
}

public Date getDataInicial() {
    return this.dataInicial;
}

public void setOperacoes(ArrayList<Operacao> operacoes) {
    this.operacoes = operacoes;
}

```



```

public List<Operacao> getOperacoes() {
    return this.operacoes;
}

public String geraRelatorioDeOperacoesDaMaquina(){

    operacoes.clear();

    if (dataFinal == null && dataInicial == null){
        operacoes = new
ArrayList<Operacao>(maquinaHome.getOperacoes());
    } else {
        operacoes =
this.getOperacoesNoPeriodo(maquinaHome.getInstance().getIdMaquina());

    }

    this.somatorioHoras();

    return Constantes.REL_OPERACOES_DA_MAQUINA_OK;
}

private void somatorioHoras() {

    totalHoras = 0;

    Iterator<Operacao> it = operacoes.iterator();

    while (it.hasNext()) {
        Operacao operacao = (Operacao) it.next();
        totalHoras = totalHoras + operacao.getHorasTrabalhadas();
    }

}

@SuppressWarnings("unchecked")

```

```

private ArrayList<Operacao> getOperacoesNoPeriodo(int idMaquina) {

    List<Operacao> oper = null;

    try {
        oper = entityManager.createQuery("from Operacao where
cod_maquina =: pidMaquina " +
"
and data_inicial > pDataInicial " +
"
and data_inicial < pDataFinal")

.setParameter("pDataInicial", dataInicial)

.setParameter("pDataFinal", dataFinal)

.setParameter("pidMaquina", idMaquina)

.getResultList();

    } catch (Exception e) {
        log.error("Relatorio Maquina Action - Erro busca operacoes
da maquina");
        log.error("Relatorio Maquina Action - " + e.getMessage());
    }

    return new ArrayList<Operacao>(oper);
}

}

package br.ufsc.inf.faccin.efagro.session;

```

```

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import br.ufsc.inf.faccin.efagro.entity.*;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Logger;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;
import org.jboss.seam.log.Log;

@Name("abastecimentoHome")
public class AbastecimentoHome extends EntityHome<Abastecimento> {

    /**
     *
     */
    private static final long serialVersionUID = 7689699046937502383L;

    @In
    EntityManager entityManager;

    @In
    FacesMessages facesMessages;

    @In(create = true)
    MaquinaHome maquinaHome;

    @Logger
    private Log log;

    public void setAbastecimentoIdAbastecimento(Integer id) {
        setId(id);
    }
}

```

```

public Integer getAbastecimentoIdAbastecimento() {
    return (Integer) getId();
}

@Override
protected Abastecimento createInstance() {
    Abastecimento abastecimento = new Abastecimento();
    return abastecimento;
}

public void wire() {
    getInstance();
    Maquina maquina = maquinaHome.getDefinedInstance();
    if (maquina != null) {
        getInstance().setMaquina(maquina);
    }
}

public boolean isWired() {
    if (getInstance().getMaquina() == null)
        return false;
    return true;
}

public Abastecimento getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

public String inserir(){

    this.getInstance().setMaquina(maquinaHome.getInstance());

    try {
        this.persist();
        facesMessages.clear();
    }
}

```

```

        facesMessages.add(Severity.INFO, "#{messages.abastecimentoRegistrado}");

        } catch (PersistenceException e) {

            log.info("Abastecimento Erro - inserir - " + e.getMessage()
);

            facesMessages.clear();

        facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarAbastecimento}
");

            return Constantes.ABASTECIMENTO_INCLUIDO_ERRO;
        }

        return Constantes.ABASTECIMENTO_INCLUIDO_OK;
    }

}

package br.ufsc.inf.faccin.efagro.session;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import br.ufsc.inf.faccin.efagro.entity.*;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Logger;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;
import org.jboss.seam.log.Log;

```

```

@Name("adubacaoHome")
public class AdubacaoHome extends EntityHome<Adubacao> {

    /**
     *
     */

    private static final long serialVersionUID = -6185247975058774823L;

    @In(create = true)
    AtividadeTalhaoHome atividadeTalhaoHome;

    @In(create = true)
    FertilizanteHome fertilizanteHome;

    @In
    EntityManager entityManager;

    @In
    FacesMessages facesMessages;

    @Logger
    private Log log;

    public void setAdubacaoCodAtividadeTalhao(Integer id) {
        setId(id);
    }

    public Integer getAdubacaoCodAtividadeTalhao() {
        return (Integer) getId();
    }

    @Override
    protected Adubacao createInstance() {
        Adubacao adubacao = new Adubacao();
        return adubacao;
    }

    public void wire() {

```

```

        getInstance();
        AtividadeTalhao atividadeTalhao = atividadeTalhaoHome
            .getDefinedInstance();
        if (atividadeTalhao != null) {
            getInstance().setAtividadeTalhao(atividadeTalhao);
        }
        Fertilizante fertilizante = fertilizanteHome.getDefinedInstance();
        if (fertilizante != null) {
            getInstance().setFertilizante(fertilizante);
        }
    }

    public boolean isWired() {
        if (getInstance().getAtividadeTalhao() == null)
            return false;
        if (getInstance().getFertilizante() == null)
            return false;
        return true;
    }

    public Aduacao getDefinedInstance() {
        return isIdDefined() ? getInstance() : null;
    }

    public String inserir(){

        if
        (atividadeTalhaoHome.inserir().equals(Constants.ATIVIDADE_INCLUIDO_ERRO
    )){

            log.error(Constants.ATIVIDADE_INCLUIDO_ERRO);
            log.error("Aduacao Home" + "Erro inserir Atividade");

            facesMessages.clear();

            facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarAduacao}");

```

```

        return Constantes.ADUBACAO_INCLUIDO_ERRO;
    }

    this.getInstance().setFertilizante(fertilizanteHome.getInstance());

    this.getInstance().setAtividadeTalhao(atividadeTalhaoHome.getInstance())
;

    this.getInstance().setCodAtividadeTalhao(atividadeTalhaoHome.getInstance
().getIdAtividadeTalhao());

    try {

        this.persist();

    } catch (PersistenceException e) {
        log.error(Constantes.ADUBACAO_INCLUIDO_ERRO);
        log.error("Adubacao Home" + "Erro inserir adubacao");
        facesMessages.clear();

        facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarAdubacao}");

        return Constantes.ADUBACAO_INCLUIDO_ERRO;
    }

    try {

atividadeTalhaoHome.getInstance().setAdubacao(this.getInstance());
        atividadeTalhaoHome.update();
        facesMessages.clear();

        facesMessages.add(Severity.INFO, "#{messages.aducacaoRegistrado}");

```



```

    } catch (PersistenceException e){

        log.error(Constants.OPERACAO_INCLUIDO_ERRO);
        log.error("Atividade Home" + "Erro atualizar Atividade");
        facesMessages.clear();

        facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarAdubacao}");

        return Constantes.ADUBACAO_INCLUIDO_ERRO;
    }

    return Constantes.ADUCACAO_INCLUIDO_OK;
}

}

package br.ufsc.inf.faccin.efagro.session;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import br.ufsc.inf.faccin.efagro.entity.*;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;

@Name("agendaManutencaoHome")
public class AgendaManutencaoHome extends EntityHome<AgendaManutencao> {

    /**
     *
     */
    private static final long serialVersionUID = -5166575627216799067L;

    @In(create = true)

```

```

MaquinaHome maquinaHome;

@In
EntityManager entityManager;

@In
FacesMessages facesMessages;

public void setAgendaManutencaoIdAgendaManutencao(Integer id) {
    setId(id);
}

public Integer getAgendaManutencaoIdAgendaManutencao() {
    return (Integer) getId();
}

@Override
protected AgendaManutencao createInstance() {
    AgendaManutencao agendaManutencao = new AgendaManutencao();
    return agendaManutencao;
}

public void wire() {
    getInstance();
    Maquina maquina = maquinaHome.getDefinedInstance();
    if (maquina != null) {
        getInstance().setMaquina(maquina);
    }
}

public boolean isWired() {
    if (getInstance().getMaquina() == null)
        return false;
    return true;
}

public AgendaManutencao getDefinedInstance() {

```

```

        return isIdDefined() ? getInstance() : null;
    }

    public String inserir(){

        this.getInstance().setMaquina(maquinaHome.getInstance());

        try {

            this.persist();
            facesMessages.clear();

            facesMessages.add(Severity.INFO, "#{messages.manutencaoAgendada}");

        } catch (PersistenceException e) {

            facesMessages.clear();

            facesMessages.add(Severity.ERROR, "#{messages.erroAgendarManutencao}");

            return Constantes.AGENDA_MANUTENCAO_INCLUIDO_ERRO;
        }

        return Constantes.AGENDA_MANUTENCAO_INCLUIDO_OK;
    }
}

package br.ufsc.inf.faccin.efagro.session;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import br.ufsc.inf.faccin.efagro.entity.*;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Logger;

```

```

import org.jboss.seam.annotations.Name;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;
import org.jboss.seam.log.Log;

@Name("aplicacaoDefensivoHome")
public class AplicacaoDefensivoHome extends EntityHome<AplicacaoDefensivo> {

    /**
     *
     */
    private static final long serialVersionUID = -5966757966258079926L;

    @In(create = true)
    AtividadeTalhaoHome atividadeTalhaoHome;
    @In(create = true)
    DefensivoHome defensivoHome;

    @In
    EntityManager entityManager;

    @In
    FacesMessages facesMessages;

    @Logger
    private Log log;

    public void setAplicacaoDefensivoCodAtividadeTalhao(Integer id) {
        setId(id);
    }

    public Integer getAplicacaoDefensivoCodAtividadeTalhao() {
        return (Integer) getId();
    }

    @Override

```

```

protected AplicacaoDefensivo createInstance() {
    AplicacaoDefensivo aplicacaoDefensivo = new AplicacaoDefensivo();
    return aplicacaoDefensivo;
}

public void wire() {
    getInstance();
    AtividadeTalhao atividadeTalhao = atividadeTalhaoHome
        .getDefinedInstance();
    if (atividadeTalhao != null) {
        getInstance().setAtividadeTalhao(atividadeTalhao);
    }
    Defensivo defensivo = defensivoHome.getDefinedInstance();
    if (defensivo != null) {
        getInstance().setDefensivo(defensivo);
    }
}

public boolean isWired() {
    if (getInstance().getAtividadeTalhao() == null)
        return false;
    if (getInstance().getDefensivo() == null)
        return false;
    return true;
}

public AplicacaoDefensivo getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

public String inserir(){

    if
(atividadeTalhaoHome.inserir().equals(Constants.ATIVIDADE_INCLUIDO_ERRO
)) {

        log.error(Constants.ATIVIDADE_INCLUIDO_ERRO);
    }
}

```

```

        log.error("AplicacaoDefensivo Home" + "Erro inserir
Atividade");

        facesMessages.clear();

        facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarAplicacaoDefen
sivo}");

        return Constantes.AP_DEFENSIVO_INCLUIDO_ERRO;

    }

    this.getInstance().setDefensivo(defensivoHome.getInstance());

    this.getInstance().setAtividadeTalhao(atividadeTalhaoHome.getInstance())
;

    this.getInstance().setCodAtividadeTalhao(atividadeTalhaoHome.getInstance
().getIdAtividadeTalhao());

    try {

        this.persist();

    } catch (PersistenceException e) {

        facesMessages.clear();

        facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarAplicacaoDefen
sivo}");

        return Constantes.AP_DEFENSIVO_INCLUIDO_ERRO;

    }

    try {

```

```

        atividadeTalhaoHome.getInstance().setAplicacaoDefensivo(this.getInstance
    ());

        atividadeTalhaoHome.update();
        facesMessages.clear();

        facesMessages.add(Severity.INFO, "#{messages.aplicacaoDefensivoRegistrado
    }");

        } catch (PersistenceException e){

            log.error(Constants.OPERACAO_INCLUIDO_ERRO);
            log.error("AplicacaoDefensivo Home" + "Erro atualizar
    Atividade");

            facesMessages.clear();

            facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarAplicacaoDefen
    sivo}");

            return Constants.AP_DEFENSIVO_INCLUIDO_ERRO;
        }

        return Constants.AP_DEFENSIVO_INCLUIDO_OK;
    }
}

package br.ufsc.inf.faccin.efagro.session;

import java.util.ArrayList;
import java.util.List;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Logger;
import org.jboss.seam.annotations.Name;

```

```

import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;
import org.jboss.seam.log.Log;

import br.ufsc.inf.faccin.efagro.entity.Adubacao;
import br.ufsc.inf.faccin.efagro.entity.AplicacaoDefensivo;
import br.ufsc.inf.faccin.efagro.entity.AtividadeTalhao;
import br.ufsc.inf.faccin.efagro.entity.Colheita;
import br.ufsc.inf.faccin.efagro.entity.Operacao;
import br.ufsc.inf.faccin.efagro.entity.Plantio;
import br.ufsc.inf.faccin.efagro.entity.Talhao;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

@Name("atividadeTalhaoHome")
public class AtividadeTalhaoHome extends EntityHome<AtividadeTalhao> {

    /**
     *
     */
    private static final long serialVersionUID = 262981235448763342L;

    @In(create = true)
    TalhaoHome talhaoHome;

    @In(create = true)
    OperacaoHome operacaoHome;

    @In
    EntityManager entityManager;

    @In
    FacesMessages facesMessages;

    @Logger
    private Log log;

```



```

public void setAtividadeTalhaoIdAtividadeTalhao(Long id) {
    setId(id);
}

public Long getAtividadeTalhaoIdAtividadeTalhao() {
    return (Long) getId();
}

@Override
protected AtividadeTalhao createInstance() {
    AtividadeTalhao atividadeTalhao = new AtividadeTalhao();
    return atividadeTalhao;
}

public void wire() {
    getInstance();
    Talhao talhao = talhaoHome.getDefinedInstance();
    if (talhao != null) {
        getInstance().setTalhao(talhao);
    }
}

public boolean isWired() {
    if (getInstance().getTalhao() == null)
        return false;
    return true;
}

public AtividadeTalhao getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

public AplicacaoDefensivo getAplicacaoDefensivo() {
    return getInstance() == null ? null :
getInstance().getAplicacaoDefensivo();
}

public Plantio getPlantio() {

```

```

        return getInstance() == null ? null : getInstance().getPlantio();
    }

    public Colheita getColheita() {
        return getInstance() == null ? null : getInstance().getColheita();
    }

    public Adubacao getAdubacoes() {
        return getInstance() == null ? null : getInstance().getAdubacao();
    }

    public List<Operacao> getOperacoes() {
        return getInstance() == null ? null : new ArrayList<Operacao>(
            getInstance().getOperacoes());
    }

    public String inserir(){

        this.getInstance().setTalhao(talhaoHome.getInstance());

        try {

            this.persist();
            facesMessages.clear();

            facesMessages.add(Severity.INFO, "#{messages.atividadeRegistrada}");

        } catch (PersistenceException e) {

            facesMessages.clear();

            facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarAtividade}");

            return Constantes.ATIVIDADE_INCLUIDO_ERRO;
        }

        operacaoHome.getInstance().setAtividadeTalhao(this.getInstance());
    }

```

```

        if
(operacaoHome.inserir().equals(Constants.OPERACAO_INCLUIDO_ERRO)) {

    log.error(Constants.OPERACAO_INCLUIDO_ERRO);
    log.error("Atividade Home" + "Erro inserir Atividade");

    facesMessages.clear();

    facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarAtividade}");

    return Constants.ATIVIDADE_INCLUIDO_ERRO;
}

return Constants.ATIVIDADE_INCLUIDO_OK;
}

}

package br.ufsc.inf.faccin.efagro.session;

import javax.persistence.EntityManager;
import javax.persistence.NoResultException;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Logger;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.contexts.Contexts;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.log.Log;
import org.jboss.seam.security.Credentials;
import org.jboss.seam.security.Identity;

import br.ufsc.inf.faccin.efagro.entity.Senha;
import br.ufsc.inf.faccin.efagro.entity.Usuario;

@Name ("authenticator")

```

```

public class Authenticator
{
    @Logger private Log log;

    @In Identity identity;

    @In Credentials credentials;

    @In
    EntityManager entityManager;

    @In
    FacesMessages facesMessages;

    public boolean authenticate()
    {
        log.info("authenticating {0}", credentials.getUsername());

        if (this.validaUsuario(credentials.getUsername())){
            return true;
        }

        return false;
    }

    private boolean validaUsuario(String email) {

        Senha senha = null;
        Usuario user = null;

        try {

            senha = (Senha) entityManager.createQuery("from Senha where
email = :pEmail")

            .setParameter("pEmail", email)

```

```

        .getSingleResult();

        } catch (NoResultException e) {
            return false;
        }

        try {

            user = (Usuario) entityManager.createQuery("from Usuario
where email = :pEmail")

                .setParameter("pEmail", email)

                .getSingleResult();

            } catch (NoResultException e) {
                return false;
            }

            if (senha.getSenha().equals(credentials.getPassword())){
                this.atribuiPrivilegio(user);
                this.setSessao(user);
                return true;
            }

            return false;
        }

        private void setSessao(Usuario user) {

            Contexts.getSessionContext().set("idUser", user.getIdUsuario());
            Contexts.getSessionContext().set("idProp",
user.getPropriedade().getIdPropriedade());

        }

```

```

private void atribuiPrivilegio(Usuario user) {

    if (user.isAdministrador()){
        identity.addRole("admin");
        log.info(user.getNome() + " Role = admin");
    }
    else {
        identity.addRole("colab");
        log.info(user.getNome() + " Role = colab");
    }
}

}

package br.ufsc.inf.faccin.efagro.session;

import br.ufsc.inf.faccin.efagro.entity.*;
import java.util.ArrayList;
import java.util.List;
import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.framework.EntityHome;

@Name("cidadeHome")
public class CidadeHome extends EntityHome<Cidade> {

    /**
     *
     */
    private static final long serialVersionUID = -5777281783069565288L;
    @In(create = true)
    EstadoHome estadoHome;

    public void setCidadeIdCidade(Integer id) {
        setId(id);
    }
}

```

```

public Integer getCidadeIdCidade() {
    return (Integer) getId();
}

@Override
protected Cidade createInstance() {
    Cidade cidade = new Cidade();
    return cidade;
}

public void wire() {
    getInstance();
    Estado estado = estadoHome.getDefinedInstance();
    if (estado != null) {
        getInstance().setEstado(estado);
    }
}

public boolean isWired() {
    if (getInstance().getEstado() == null)
        return false;
    return true;
}

public Cidade getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

public List<Endereco> getEnderecos() {
    return getInstance() == null ? null : new ArrayList<Endereco>(
        getInstance().getEnderecos());
}

}

package br.ufsc.inf.faccin.efagro.session;

import java.util.ArrayList;

```

```

import java.util.List;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Logger;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;
import org.jboss.seam.log.Log;

import br.ufsc.inf.faccin.efagro.entity.Cliente;
import br.ufsc.inf.faccin.efagro.entity.Endereco;
import br.ufsc.inf.faccin.efagro.entity.Propriedade;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

@Name("clienteHome")
public class ClienteHome extends EntityHome<Cliente> {

    /**
     *
     */
    private static final long serialVersionUID = 5304916097741947548L;

    @Logger private Log log;

    @In(create = true)
    EnderecoHome enderecoHome;

    @In(create = true)
    PropriedadeHome propriedadeHome;

    @In(create = true)
    UsuarioHome usuarioHome;

```



```

@In
EntityManager entityManager;

@In
FacesMessages facesMessages;

public void setClienteIdCliente(Integer id) {
    setId(id);
}

public Integer getClienteIdCliente() {
    return (Integer) getId();
}

@Override
protected Cliente createInstance() {
    Cliente cliente = new Cliente();
    return cliente;
}

public void wire() {
    getInstance();
    Endereco endereco = enderecoHome.getDefinedInstance();
    if (endereco != null) {
        getInstance().setEndereco(endereco);
    }
}

public boolean isWired() {
    if (getInstance().getEndereco() == null)
        return false;
    return true;
}

public Cliente getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

```

```

}

public List<Propriedade> getPropriedades() {
    return getInstance() == null ? null : new ArrayList<Propriedade>(
        getInstance().getPropriedades());
}

/**
 * Cadastrar novo Cliente.
 *
 * Com base nas informações inseridas pelo usuário, é criado um cadastro
de cliente, um usuário e uma propriedade para o
 * cliente.
 *
 * @return CLIENTE_INCLUIDO_OK se cadatro ok, senão
CLIENTE_INCLUIDO_ERRO;
 * */
public String inserir(){

    try {

        enderecoHome.persist();

    } catch (Exception e) {
        log.error(e.getMessage());
        return Constantes.CLIENTE_INCLUIDO_ERRO;
    }

    this.getInstance().setEndereco(enderecoHome.getInstance());

    try {

        this.persist();
    } catch (PersistenceException e) {

```

```

        log.error(e.getMessage());
        facesMessages.clear();

        facesMessages.add(Severity.ERROR, "#{messages.erroCadastrarCliente}");

        return Constantes.CLIENTE_INCLUIDO_ERRO;
    }

    propriedadeHome.getInstance().setCliente(this.getInstance());

    propriedadeHome.getInstance().setEndereco(enderecoHome.getInstance());

    if
    (!propriedadeHome.inserir().equals(Constantes.PROPRIEDADE_INCLUIDO_OK)) {

        log.error("Cliente Home - " +
Constantes.PROPRIEDADE_INCLUIDO_ERRO);
        facesMessages.clear();

        facesMessages.add(Severity.ERROR, "#{messages.erroCadastrarCliente}");
        return Constantes.CLIENTE_INCLUIDO_ERRO;

    }

    usuarioHome.getInstance().setAdministrador(true);
    usuarioHome.getInstance().setEmail(this.getInstance().getEmail());
    usuarioHome.getInstance().setNome(this.getInstance().getNome());

    usuarioHome.getInstance().setPropriedade(propriedadeHome.getInstance());

    if
    (!usuarioHome.inserir().equals(Constantes.USUARIO_INCLUIDO_OK)) {

        log.error("Cliente Home - " +
Constantes.USUARIO_INCLUIDO_ERRO);
        facesMessages.clear();

```

```

        facesMessages.add(Severity.ERROR, "#{messages.erroCadastrarCliente}");
        return Constantes.CLIENTE_INCLUIDO_ERRO;
    }

    facesMessages.clear();
    facesMessages.add(Severity.INFO, "#{messages.clienteCadastrado}");

    return Constantes.CLIENTE_INCLUIDO_OK;
}
}
package br.ufsc.inf.faccin.efagro.session;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import br.ufsc.inf.faccin.efagro.entity.*;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Logger;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;
import org.jboss.seam.log.Log;

@Name("colheitaHome")
public class ColheitaHome extends EntityHome<Colheita> {

    /**
     *
     */
    private static final long serialVersionUID = -901571210804351593L;
    @In(create = true)
    AtividadeTalhaoHome atividadeTalhaoHome;

```

```

@In
EntityManager entityManager;

@In
FacesMessages facesMessages;

@Logger
private Log log;

public void setColheitaCodAtividadeTalhao(Integer id) {
    setId(id);
}

public Integer getColheitaCodAtividadeTalhao() {
    return (Integer) getId();
}

@Override
protected Colheita createInstance() {
    Colheita colheita = new Colheita();
    return colheita;
}

public void wire() {
    getInstance();
    AtividadeTalhao atividadeTalhao =
atividadeTalhaoHome.getDefinedInstance();
    if (atividadeTalhao != null) {
        getInstance().setAtividadeTalhao(atividadeTalhao);
    }
}

public boolean isWired() {
    if (getInstance().getAtividadeTalhao() == null)
        return false;
    return true;
}

```

```

public Colheita getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

public String inserir(){

    if
(atividadeTalhaoHome.inserir().equals(Constants.ATIVIDADE_INCLUIDO_ERRO
)){

        log.error(Constants.ATIVIDADE_INCLUIDO_ERRO);
        log.error("Colheita Home" + "Erro inserir Atividade");

        facesMessages.clear();

        facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarColheita}");

        return Constants.COLHEITA_INCLUIDO_ERRO;

    }

    this.getInstance().setAtividadeTalhao(atividadeTalhaoHome.getInstance())
;

    this.getInstance().setCodAtividadeTalhao(atividadeTalhaoHome.getInstance
().getIdAtividadeTalhao());

    try {

        this.persist();

    } catch (PersistenceException e) {

        facesMessages.clear();

```

```

facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarColheita}");

        return Constantes.COLHEITA_INCLUIDO_ERRO;
    }

    try {

atividadeTalhaoHome.getInstance().setColheita(this.getInstance());
        atividadeTalhaoHome.update();
        facesMessages.clear();

facesMessages.add(Severity.INFO, "#{messages.colheitaRegistrada}");

        } catch (PersistenceException e){

            log.error(Constantes.OPERACAO_INCLUIDO_ERRO);
            log.error("Colheita Home" + "Erro atualizar Atividade");
            facesMessages.clear();

facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarColheita}");

            return Constantes.COLHEITA_INCLUIDO_ERRO;
        }

        return Constantes.COLHEITA_INCLUIDO_OK;
    }
}

package br.ufsc.inf.faccin.efagro.session;

import br.ufsc.inf.faccin.efagro.entity.*;
import java.util.ArrayList;
import java.util.List;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.framework.EntityHome;

```

```

@Name("culturaHome")
public class CulturaHome extends EntityHome<Cultura> {

    /**
     *
     */
    private static final long serialVersionUID = 2890663814316295768L;

    public void setCulturaIdCultura(Integer id) {
        setId(id);
    }

    public Integer getCulturaIdCultura() {
        return (Integer) getId();
    }

    @Override
    protected Cultura createInstance() {
        Cultura cultura = new Cultura();
        return cultura;
    }

    public void wire() {
        getInstance();
    }

    public boolean isWired() {
        return true;
    }

    public Cultura getDefinedInstance() {
        return isIdDefined() ? getInstance() : null;
    }

    public List<Variedade> getVariedades() {
        return getInstance() == null ? null : new ArrayList<Variedade>(
            getInstance().getVariedades());
    }
}

```



```

}
package br.ufsc.inf.faccin.efagro.session;

import br.ufsc.inf.faccin.efagro.entity.*;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.framework.EntityQuery;
import java.util.Arrays;

@Name("culturaList")
public class CulturaList extends EntityQuery<Cultura> {

    /**
     *
     */
    private static final long serialVersionUID = -1402445139002303357L;

    private static final String EJBQL = "select cultura from Cultura
cultura";

    private static final String[] RESTRICTIONS = {
        "lower(cultura.nome) like
concat(lower("#{culturaList.cultura.nome}),'%')",
        "lower(cultura.descricao) like
concat(lower("#{culturaList.cultura.descricao}),'%')", };

    private Cultura cultura = new Cultura();

    public CulturaList() {
        setEjbql(EJBQL);
        setRestrictionExpressionStrings(Arrays.asList(RESTRICTIONS));
        setMaxResults(25);
    }

    public Cultura getCultura() {
        return cultura;
    }
}

```

```

package br.ufsc.inf.faccin.efagro.session;

import br.ufsc.inf.faccin.efagro.entity.*;
import java.util.ArrayList;
import java.util.List;
import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.framework.EntityHome;

@Name("defensivoHome")
public class DefensivoHome extends EntityHome<Defensivo> {

    /**
     *
     */
    private static final long serialVersionUID = -867511153881199615L;
    @In(create = true)
    FabricanteHome fabricanteHome;

    public void setDefensivoIdDefensivo(Integer id) {
        setId(id);
    }

    public Integer getDefensivoIdDefensivo() {
        return (Integer) getId();
    }

    @Override
    protected Defensivo createInstance() {
        Defensivo defensivo = new Defensivo();
        return defensivo;
    }

    public void wire() {
        getInstance();
        Fabricante fabricante = fabricanteHome.getDefinedInstance();
        if (fabricante != null) {
            getInstance().setFabricante(fabricante);
        }
    }
}

```

```

        }
    }

    public boolean isWired() {
        return true;
    }

    public Defensivo getDefinedInstance() {
        return isIdDefined() ? getInstance() : null;
    }

    public List<AplicacaoDefensivo> getAplicacaoDefensivos() {
        return getInstance() == null ? null
            : new ArrayList<AplicacaoDefensivo>(getInstance()
                .getAplicacaoDefensivos());
    }
}

package br.ufsc.inf.faccin.efagro.session;

import br.ufsc.inf.faccin.efagro.entity.*;
import java.util.ArrayList;
import java.util.List;
import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.framework.EntityHome;

@Name("enderecoHome")
public class EnderecoHome extends EntityHome<Endereco> {

    /**
     *
     */
    private static final long serialVersionUID = -5108009309993702302L;

    @In(create = true)
    CidadeHome cidadeHome;

```

```

public void setEnderecoIdEndereco(Integer id) {
    setId(id);
}

public Integer getEnderecoIdEndereco() {
    return (Integer) getId();
}

@Override
protected Endereco createInstance() {
    Endereco endereco = new Endereco();
    return endereco;
}

public void wire() {
    getInstance();
    Cidade cidade = cidadeHome.getDefinedInstance();
    if (cidade != null) {
        getInstance().setCidade(cidade);
    }
}

public boolean isWired() {
    if (getInstance().getCidade() == null)
        return false;
    return true;
}

public Endereco getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

public List<Propriedade> getPropriedades() {
    return getInstance() == null ? null : new ArrayList<Propriedade>(
        getInstance().getPropriedades());
}

public List<Cliente> getClientes() {

```

```

        return getInstance() == null ? null : new ArrayList<Cliente>(
            getInstance().getClientes());
    }

}

package br.ufsc.inf.faccin.efagro.session;

import br.ufsc.inf.faccin.efagro.entity.*;
import java.util.ArrayList;
import java.util.List;
import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.framework.EntityHome;

@Name("estadoHome")
public class EstadoHome extends EntityHome<Estado> {

    /**
     *
     */
    private static final long serialVersionUID = 8915654707693692215L;
    @In(create = true)
    PaisHome paisHome;

    public void setEstadoIdEstado(Integer id) {
        setId(id);
    }

    public Integer getEstadoIdEstado() {
        return (Integer) getId();
    }

    @Override
    protected Estado createInstance() {
        Estado estado = new Estado();
        return estado;
    }
}

```

```

    public void wire() {
        getInstance();
        Pais pais = paisHome.getDefinedInstance();
        if (pais != null) {
            getInstance().setPais(pais);
        }
    }

    public boolean isWired() {
        if (getInstance().getPais() == null)
            return false;
        return true;
    }

    public Estado getDefinedInstance() {
        return isIdDefined() ? getInstance() : null;
    }

    public List<Cidade> getCidades() {
        return getInstance() == null ? null : new ArrayList<Cidade>(
            getInstance().getCidades());
    }
}

package br.ufsc.inf.faccin.efagro.session;

import br.ufsc.inf.faccin.efagro.entity.*;
import java.util.ArrayList;
import java.util.List;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.framework.EntityHome;

@Name("fabricanteHome")
public class FabricanteHome extends EntityHome<Fabricante> {

    /**
     *
     */
}

```

```

private static final long serialVersionUID = -8757859256743822383L;

public void setFabricanteIdFabricante(Integer id) {
    setId(id);
}

public Integer getFabricanteIdFabricante() {
    return (Integer) getId();
}

@Override
protected Fabricante createInstance() {
    Fabricante fabricante = new Fabricante();
    return fabricante;
}

public void wire() {
    getInstance();
}

public boolean isWired() {
    return true;
}

public Fabricante getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

public List<Variedade> getVariedades() {
    return getInstance() == null ? null : new ArrayList<Variedade>(
        getInstance().getVariedades());
}

public List<Fertilizante> getFertilizantes() {
    return getInstance() == null ? null : new ArrayList<Fertilizante>(
        getInstance().getFertilizantes());
}

```

```

    public List<Defensivo> getDefensivos() {
        return getInstance() == null ? null : new ArrayList<Defensivo>(
            getInstance().getDefensivos());
    }
}

package br.ufsc.inf.faccin.efagro.session;

import br.ufsc.inf.faccin.efagro.entity.*;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.framework.EntityQuery;
import java.util.Arrays;

@Name("fabricanteList")
public class FabricanteList extends EntityQuery<Fabricante> {

    /**
     *
     */
    private static final long serialVersionUID = -3113279904466931755L;

    private static final String EJBQL = "select fabricante from Fabricante
fabricante";

    private static final String[] RESTRICTIONS = {
"lower(fabricante.nomeFabricante) like
concat(lower("#{fabricanteList.fabricante.nomeFabricante}),'%')", };

    private Fabricante fabricante = new Fabricante();

    public FabricanteList() {
        setEjbql(EJBQL);
        setRestrictionExpressionStrings(Arrays.asList(RESTRICTIONS));
        setMaxResults(25);
    }

    public Fabricante getFabricante() {
        return fabricante;
    }
}

```



```

    }
}
package br.ufsc.inf.faccin.efagro.session;

import br.ufsc.inf.faccin.efagro.entity.*;
import java.util.ArrayList;
import java.util.List;
import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.framework.EntityHome;

@Name("fertilizanteHome")
public class FertilizanteHome extends EntityHome<Fertilizante> {

    /**
     *
     */
    private static final long serialVersionUID = -3554380750326679842L;
    @In(create = true)
    FabricanteHome fabricanteHome;

    public void setFertilizanteIdFertilizante(Integer id) {
        setId(id);
    }

    public Integer getFertilizanteIdFertilizante() {
        return (Integer) getId();
    }

    @Override
    protected Fertilizante createInstance() {
        Fertilizante fertilizante = new Fertilizante();
        return fertilizante;
    }

    public void wire() {
        getInstance();
        Fabricante fabricante = fabricanteHome.getDefinedInstance();

```

```

        if (fabricante != null) {
            getInstance().setFabricante(fabricante);
        }
    }

    public boolean isWired() {
        return true;
    }

    public Fertilizante getDefinedInstance() {
        return isIdDefined() ? getInstance() : null;
    }

    public List<Adubacao> getAdubacoes() {
        return getInstance() == null ? null : new ArrayList<Adubacao>(
            getInstance().getAdubacoes());
    }
}

package br.ufsc.inf.faccin.efagro.session;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import br.ufsc.inf.faccin.efagro.entity.*;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;

@Name("manutencaoHome")
public class ManutencaoHome extends EntityHome<Manutencao> {

    /**
     *

```

```

    */
    private static final long serialVersionUID = 563954741067030497L;
    @In(create = true)
    MaquinaHome maquinaHome;

    @In
    EntityManager entityManager;

    @In
    FacesMessages facesMessages;

    public void setManutencaoIdManutencao(Integer id) {
        setId(id);
    }

    public Integer getManutencaoIdManutencao() {
        return (Integer) getId();
    }

    @Override
    protected Manutencao createInstance() {
        Manutencao manutencao = new Manutencao();
        return manutencao;
    }

    public void wire() {
        getInstance();
        Maquina maquina = maquinaHome.getDefinedInstance();
        if (maquina != null) {
            getInstance().setMaquina(maquina);
        }
    }

    public boolean isWired() {
        if (getInstance().getMaquina() == null)
            return false;
        return true;
    }

```

```

    }

    public Manutencao getDefinedInstance() {
        return isIdDefined() ? getInstance() : null;
    }

    public String inserir(){

        this.getInstance().setMaquina(maquinaHome.getInstance());

        try {

            this.persist();
            facesMessages.clear();

            facesMessages.add(Severity.INFO, "#{messages.manutencaoRegistrado}");

        } catch (PersistenceException e) {

            facesMessages.clear();

            facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarManutencao}");

            return Constantes.MANUTENCAO_INCLUIDO_ERRO;
        }

        return Constantes.MANUTENCAO_INCLUIDO_OK;
    }
}

package br.ufsc.inf.faccin.efagro.session;

import java.util.ArrayList;
import java.util.List;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import org.jboss.seam.annotations.In;

```

```

import org.jboss.seam.annotations.Name;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;

import br.ufsc.inf.faccin.efagro.entity.Abastecimento;
import br.ufsc.inf.faccin.efagro.entity.AgendaManutencao;
import br.ufsc.inf.faccin.efagro.entity.Manutencao;
import br.ufsc.inf.faccin.efagro.entity.Maquina;
import br.ufsc.inf.faccin.efagro.entity.Operacao;
import br.ufsc.inf.faccin.efagro.entity.Propriedade;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

@Name("maquinaHome")
public class MaquinaHome extends EntityHome<Maquina> {

    /**
     *
     */
    private static final long serialVersionUID = 5958602371713798271L;

    @In(create = true)
    PropriedadeHome propriedadeHome;

    @In
    EntityManager entityManager;

    @In
    FacesMessages facesMessages;

    public void setMaquinaIdMaquina(Integer id) {
        setId(id);
    }

    public Integer getMaquinaIdMaquina() {
        return (Integer) getId();
    }
}

```

```

@Override
protected Maquina createInstance() {
    Maquina maquina = new Maquina();
    return maquina;
}

public void wire() {
    getInstance();
    Propriedade propriedade = propriedadeHome.getDefinedInstance();
    if (propriedade != null) {
        getInstance().setPropriedade(propriedade);
    }
}

public boolean isWired() {
    if (getInstance().getPropriedade() == null)
        return false;
    return true;
}

public Maquina getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

public List<Abastecimento> getAbastecimentos() {
    return getInstance() == null ? null : new
ArrayList<Abastecimento>(
        getInstance().getAbastecimentos());
}

public List<Operacao> getOperacoes() {
    return getInstance() == null ? null : new ArrayList<Operacao>(
        getInstance().getOperacoes());
}

public List<AgendaManutencao> getAgendaManutencaos() {

```

```

        return getInstance() == null ? null : new
ArrayList<AgendaManutencao>(
            getInstance().getAgendaManutencaos());
    }

    public List<Manutencao> getManutencaos() {
        return getInstance() == null ? null : new ArrayList<Manutencao>(
            getInstance().getManutencaos());
    }

    public String inserir(){

        propriedadeHome.carregaPropriedade();
        this.getInstance().setPropriedade(propriedadeHome.getInstance());

        try {

            this.persist();
            facesMessages.clear();

            facesMessages.add(Severity.INFO, "#{messages.maquinaCadastrada}");

            entityManager.flush();
        } catch (PersistenceException e) {

            facesMessages.clear();

            facesMessages.add(Severity.ERROR, "#{messages.erroCadastrarMaquina}");

            return Constantes.MAQUINA_INCLUIDO_ERRO;
        }

        entityManager.refresh(propriedadeHome.getInstance());
        return Constantes.MAQUINA_INCLUIDO_OK;
    }

    public String alterar(){

```

```

        try {

            this.update();
            facesMessages.clear();

            facesMessages.add(Severity.INFO, "#{messages.maquinaAlterada}");

            entityManager.flush();
        } catch (PersistenceException e) {

            facesMessages.clear();

            facesMessages.add(Severity.ERROR, "#{messages.erroAlterarMaquina}");

            return Constantes.MAQUINA_ALTERADA_ERRO;
        }

        entityManager.refresh(propriedadeHome.getInstance());

        return Constantes.MAQUINA_ALTERADO_OK;
    }
}

package br.ufsc.inf.faccin.efagro.session;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import br.ufsc.inf.faccin.efagro.entity.*;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;

@Name("ocorrenciaPragaHome")
public class OcorrenciaPragaHome extends EntityHome<OcorrenciaPraga> {

```



```

/**
 *
 */
private static final long serialVersionUID = -1511723861901205560L;
@In(create = true)
TalhaoHome talhaoHome;

@In
EntityManager entityManager;

@In
FacesMessages facesMessages;

public void setOcorrenciaPragaIdOcorrenciaPraga(Integer id) {
    setId(id);
}

public Integer getOcorrenciaPragaIdOcorrenciaPraga() {
    return (Integer) getId();
}

@Override
protected OcorrenciaPraga createInstance() {
    OcorrenciaPraga ocorrenciaPraga = new OcorrenciaPraga();
    return ocorrenciaPraga;
}

public void wire() {
    getInstance();
    Talhao talhao = talhaoHome.getDefinedInstance();
    if (talhao != null) {
        getInstance().setTalhao(talhao);
    }
}

public boolean isWired() {

```

```

        if (getInstance().getTalhao() == null)
            return false;
        return true;
    }

    public OcorrenciaPraga getDefinedInstance() {
        return isIdDefined() ? getInstance() : null;
    }

    public String inserir(){

        this.getInstance().setTalhao(talhaoHome.getInstance());

        try {

            this.persist();
            facesMessages.clear();

            facesMessages.add(Severity.INFO, "#{messages.pragaRegistrada}");

        } catch (PersistenceException e) {

            facesMessages.clear();

            facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarPraga}");

            return Constantes.OCORRENCIA_PRAGA_INCLUIDO_ERRO;
        }

        entityManager.refresh(talhaoHome.getInstance());

        return Constantes.OCORRENCIA_PRAGA_INCLUIDO_OK;
    }
}

package br.ufsc.inf.faccin.efagro.session;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

```

```

import br.ufsc.inf.faccin.efagro.entity.*;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Logger;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;
import org.jboss.seam.log.Log;

@Name("operacaoHome")
public class OperacaoHome extends EntityHome<Operacao> {

    /**
     *
     */
    private static final long serialVersionUID = 6560399526344979628L;

    @In(create = true)
    MaquinaHome maquinaHome;

    @In(create = true)
    AtividadeTalhaoHome atividadeTalhaoHome;

    @In
    EntityManager entityManager;

    @In
    FacesMessages facesMessages;

    @Logger
    private Log log;

    public void setOperacaoIdOperacao(Long id) {
        setId(id);
    }

```

```

}

public Long getOperacaoIdOperacao() {
    return (Long) getId();
}

@Override
protected Operacao createInstance() {
    Operacao operacao = new Operacao();
    return operacao;
}

public void wire() {
    getInstance();
    Maquina maquina = maquinaHome.getDefinedInstance();
    if (maquina != null) {
        getInstance().setMaquina(maquina);
    }
    AtividadeTalhao atividadeTalhao =
atividadeTalhaoHome.getDefinedInstance();

    if (atividadeTalhao != null) {
        getInstance().setAtividadeTalhao(atividadeTalhao);
    }
}

public boolean isWired() {
    if (getInstance().getMaquina() == null)
        return false;
    if (getInstance().getAtividadeTalhao() == null)
        return false;
    return true;
}

public Operacao getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

```

```

public String inserir(){

    this.getInstance().setMaquina(maquinaHome.getInstance());

    try {

        this.persist();
        facesMessages.clear();

facesMessages.add(Severity.INFO, "#{messages.operacaoRegistrada}");

    } catch (PersistenceException e) {

        log.error(e.getMessage());
        log.error("Operacao Home" + "Erro inserir Operacao");

        facesMessages.clear();

facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarOperacao}");

        return Constantes.OPERACAO_INCLUIDO_ERRO;
    }

    return Constantes.OPERACAO_INCLUIDO_OK;
}

}

package br.ufsc.inf.faccin.efagro.session;

import br.ufsc.inf.faccin.efagro.entity.*;
import java.util.ArrayList;
import java.util.List;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.framework.EntityHome;

@Name("paisHome")

```

```

public class PaisHome extends EntityHome<Pais> {

    /**
     *
     */
    private static final long serialVersionUID = 7295082236595433399L;

    public void setPaisIdPais(Integer id) {
        setId(id);
    }

    public Integer getPaisIdPais() {
        return (Integer) getId();
    }

    @Override
    protected Pais createInstance() {
        Pais pais = new Pais();
        return pais;
    }

    public void wire() {
        getInstance();
    }

    public boolean isWired() {
        return true;
    }

    public Pais getDefinedInstance() {
        return isIdDefined() ? getInstance() : null;
    }

    public List<Estado> getEstados() {
        return getInstance() == null ? null : new ArrayList<Estado>(
            getInstance().getEstados());
    }
}

```

```

}
package br.ufsc.inf.faccin.efagro.session;

import br.ufsc.inf.faccin.efagro.entity.*;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.framework.EntityQuery;
import java.util.Arrays;

@Name("paisList")
public class PaisList extends EntityQuery<Pais> {

    /**
     *
     */
    private static final long serialVersionUID = 2974080202746159631L;

    private static final String EJBQL = "select pais from Pais pais";

    private static final String[] RESTRICTIONS = {
        "lower(pais.nome) like
concat(lower("#{paisList.pais.nome}),'%')",
        "lower(pais.sigla) like
concat(lower("#{paisList.pais.sigla}),'%')", };

    private Pais pais = new Pais();

    public PaisList() {
        setEjbql(EJBQL);
        setRestrictionExpressionStrings(Arrays.asList(RESTRICTIONS));
        //setMaxResults(25);
    }

    public Pais getPais() {
        return pais;
    }
}
package br.ufsc.inf.faccin.efagro.session;

```

```

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import br.ufsc.inf.faccin.efagro.entity.*;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Logger;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;
import org.jboss.seam.log.Log;

@Name("plantioHome")
public class PlantioHome extends EntityHome<Plantio> {

    /**
     *
     */
    private static final long serialVersionUID = -8438401567727697254L;
    @In(create = true)
    AtividadeTalhaoHome atividadeTalhaoHome;
    @In(create = true)
    VariedadeHome variedadeHome;

    @In
    EntityManager entityManager;

    @In
    FacesMessages facesMessages;

    @Logger
    private Log log;

    public void setPlantioCodAtividadeTalhao(Integer id) {

```



```

        setId(id);
    }

    public Integer getPlantioCodAtividadeTalhao() {
        return (Integer) getId();
    }

    @Override
    protected Plantio createInstance() {
        Plantio plantio = new Plantio();
        return plantio;
    }

    public void wire() {
        getInstance();
        AtividadeTalhao atividadeTalhao = atividadeTalhaoHome
            .getDefinedInstance();
        if (atividadeTalhao != null) {
            getInstance().setAtividadeTalhao(atividadeTalhao);
        }
        Variedade variedade = variedadeHome.getDefinedInstance();
        if (variedade != null) {
            getInstance().setVariedade(variedade);
        }
    }

    public boolean isWired() {
        if (getInstance().getAtividadeTalhao() == null)
            return false;
        if (getInstance().getVariedade() == null)
            return false;
        return true;
    }

    public Plantio getDefinedInstance() {
        return isIdDefined() ? getInstance() : null;
    }
}

```

```

public String inserir(){

    if
(atividadeTalhaoHome.inserir().equals(Constants.ATIVIDADE_INCLUIDO_ERRO
)){

        log.error(Constants.ATIVIDADE_INCLUIDO_ERRO);
        log.error("Plantio Home" + "Erro inserir Atividade");

        facesMessages.clear();

        facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarPlantio}");

        return Constants.PLANTIO_INCLUIDO_ERRO;

    }

    this.getInstance().setVariedade(variedadeHome.getInstance());

    this.getInstance().setAtividadeTalhao(atividadeTalhaoHome.getInstance())
;

    this.getInstance().setCodAtividadeTalhao(atividadeTalhaoHome.getInstance
().getIdAtividadeTalhao());

    try {
        this.persist();
    } catch (PersistenceException e) {

        facesMessages.clear();

        facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarPlantio}");

        return Constants.PLANTIO_INCLUIDO_ERRO;

    }

    try {

```

```

        atividadeTalhaoHome.getInstance().setPlantio(this.getInstance());
        atividadeTalhaoHome.update();
        facesMessages.clear();

        facesMessages.add(Severity.INFO, "#{messages.plantioRegistrado}");

        } catch (PersistenceException e){

            log.error(Constants.ATIVIDADE_INCLUIDO_ERRO);
            log.error("Plantio Home" + "Erro atualizar Atividade");
            facesMessages.clear();

            facesMessages.add(Severity.ERROR, "#{messages.erroRegistrarPlantio}");

            return Constantes.PLANTIO_INCLUIDO_ERRO;
        }

        return Constantes.PLANTIO_INCLUIDO_OK;
    }
}

package br.ufsc.inf.faccin.efagro.session;

import java.util.ArrayList;
import java.util.List;

import javax.persistence.EntityManager;
import javax.persistence.NoResultException;
import javax.persistence.PersistenceException;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Logger;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.contexts.Contexts;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;
import org.jboss.seam.log.Log;

```

```

import br.ufsc.inf.faccin.efagro.entity.Cliente;
import br.ufsc.inf.faccin.efagro.entity.Endereco;
import br.ufsc.inf.faccin.efagro.entity.Maquina;
import br.ufsc.inf.faccin.efagro.entity.Propriedade;
import br.ufsc.inf.faccin.efagro.entity.Safra;
import br.ufsc.inf.faccin.efagro.entity.Usuario;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

@Name("propriedadeHome")
public class PropriedadeHome extends EntityHome<Propriedade> {

    /**
     *
     */
    private static final long serialVersionUID = 4975892845841211734L;

    @In(create = true)
    EnderecoHome enderecoHome;

    @In(create = true)
    ClienteHome clienteHome;

    @In
    EntityManager entityManager;

    @In
    FacesMessages facesMessages;

    @Logger
    private Log log;

    public void setPropriedadeIdPropriedade(Integer id) {
        setId(id);
    }

    public Integer getPropriedadeIdPropriedade() {
        return (Integer) getId();
    }
}

```

```

}

@Override
protected Propriedade createInstance() {
    Propriedade propriedade = new Propriedade();
    return propriedade;
}

public void wire() {
    getInstance();
    Endereco endereco = enderecoHome.getDefinedInstance();
    if (endereco != null) {
        getInstance().setEndereco(endereco);
    }
    Cliente cliente = clienteHome.getDefinedInstance();
    if (cliente != null) {
        getInstance().setCliente(cliente);
    }
}

public boolean isWired() {
    if (getInstance().getCliente() == null)
        return false;
    return true;
}

public Propriedade getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

public List<Usuario> getUsuarios() {
    return getInstance() == null ? null : new ArrayList<Usuario>(
        getInstance().getUsuarios());
}

public List<Safra> getSafras() {
    return getInstance() == null ? null : new ArrayList<Safra>(

```

```

        getInstance().getSafras());
    }

    public List<Maquina> getMaquinas() {
        return getInstance() == null ? null : new ArrayList<Maquina>(
            getInstance().getMaquinas());
    }

    public Safra getSafraAtual(){

        Safra safraAtual = null;

        try {
            safraAtual = (Safra) entityManager.createQuery("from Safra
where cod_propriedade =:pcodPropriedade and atual = true")

                .setParameter("pcodPropriedade", this.getInstance().getIdPropriedade())
                    .getSingleResult();
        } catch (NoResultException e) {
            log.error( "PROPRIIDADE HOME - getSafraAtual -" +
e.getMessage());
            facesMessages.clear();

            facesMessages.add(Severity.INFO, "#{messages.aPropriedadeNaoTemSafraAtual
Definida}");
        }

        return safraAtual;
    }

    public void carregaPropriedade() {

        this.setPropriedadeIdPropriedade((Integer)
Contexts.getSessionContext().get("idProp"));
    }
}

```

```

public String inserir(){

    this.getInstance().setAreaCultivo(0);
    this.getInstance().setAreaOutras(0);
    this.getInstance().setAreaPreservacao(0);
    this.getInstance().setAreaTotal(0);
    this.getInstance().setNome("Nome da propriedade");

    try {

        this.persist();
        facesMessages.clear();

facesMessages.add(Severity.INFO, "#{messages.propriedadeCadastrada}");

    } catch (PersistenceException e) {

        log.error( "PROPRIEDADE HOME - incluir -" + e.getMessage());

        facesMessages.clear();

facesMessages.add(Severity.ERROR, "#{messages.erroCadastrarPropriedade}")
;

        return Constantes.PROPRIEDADE_INCLUIDO_ERRO;
    }

    return Constantes.PROPRIEDADE_INCLUIDO_OK;
}

public String alterar(){

    try {

        this.update();

```

```

        facesMessages.clear();

        facesMessages.add(Severity.INFO, "#{messages.propriedadeAlteradaSucesso}"
    );
        } catch (PersistenceException e) {

            log.error( "PROPRIIDADE HOME - alterar - " +
e.getMessage());

            facesMessages.clear();

            facesMessages.add(Severity.ERROR, "#{messages.erroAlterarPropriedade}");

            return Constantes.PROPRIIDADE_ALTARADA_ERRO;
        }

        return Constantes.PROPRIIDADE_ALTARADA_OK;
    }

}

package br.ufsc.inf.faccin.efagro.session;

import br.ufsc.inf.faccin.efagro.entity.*;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;

```



```

@Name("safraHome")
public class SafraHome extends EntityHome<Safra> {

    /**
     *
     */
    private static final long serialVersionUID = 3887415980858522195L;

    @In(create = true)
    PropriedadeHome propriedadeHome;

    @In
    EntityManager entityManager;

    @In
    FacesMessages facesMessages;

    public void setSafraIdSafra(Long id) {
        setId(id);
    }

    public Long getSafraIdSafra() {
        return (Long) getId();
    }

    @Override
    protected Safra createInstance() {
        Safra safra = new Safra();
        return safra;
    }

    public void wire() {
        getInstance();
        Propriedade propriedade = propriedadeHome.getDefinedInstance();
        if (propriedade != null) {
            getInstance().setPropriedade(propriedade);
        }
    }
}

```

```

public boolean isWired() {
    if (getInstance().getPropriedade() == null)
        return false;
    return true;
}

public Safra getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

public List<Talhao> getTalhaos() {
    return getInstance() == null ? null : new ArrayList<Talhao>(
        getInstance().getTalhaos());
}

public String inserir(){

    propriedadeHome.carregaPropriedade();
    this.getInstance().setPropriedade(propriedadeHome.getInstance());

    if
    (this.getInstance().getDataInicio().compareTo(this.getInstance().getData
Fim()) > -1){

        facesMessages.clear();
        facesMessages.add(Severity.INFO
, "#{messages.dataFimDeveSerPorteriorAoInicio}");
        return "";
    }

    this.getInstance().setAtual(this.isAnoSafraAtual(this.getInstance().getD
ataInicio(), this.getInstance().getDataFim()));

    try {

        this.persist();
    }
}

```

```

        facesMessages.clear();

facesMessages.add(Severity.INFO, "#{messages.safraCadastrada}");

    } catch (PersistenceException e) {

        facesMessages.clear();

facesMessages.add(Severity.ERROR, "#{messages.erroCadastrarSafra}");

        return Constantes.SAFRA_INCLUIDO_ERRO;
    }

    return Constantes.SAFRA_INCLUIDO_OK;
}

/**
 * Se a data inicio da safra for anterior que a data atual e a data fim
posterior que a atual, significa que o Ano/Safra Ã© o atual
 * */
private boolean isAnoSafraAtual(Date inicio, Date fim){

    Date dataAtual = new Date();

    if (inicio.compareTo(dataAtual) <= 0){
        if (fim.compareTo(dataAtual) >= 0){
            return true;
        }
    }

    return false;
}

}

package br.ufsc.inf.faccin.efagro.session;

import javax.persistence.EntityManager;

import org.jboss.seam.annotations.In;

```

```

import org.jboss.seam.annotations.Logger;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.annotations.security.Restrict;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.international.StatusMessage.Severity;
import org.jboss.seam.log.Log;
import org.jboss.seam.security.Credentials;

import br.ufsc.inf.faccin.efagro.entity.Senha;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

@Name("senhaAction")
@Restrict("#{identity.loggedIn}")
public class SenhaAction{

    private String senhaAtual = "";
    private String senhaNova = "";
    private String senhaConfirma = "";

    /**
     *
     */

    @Logger
    Log log;

    @In
    Credentials credentials;

    @In
    FacesMessages facesMessages;

    @In
    EntityManager entityManager;

```

```

@In(create=true)
UsuarioHome usuarioHome;

public String getSenhaAtual() {
    return senhaAtual;
}

public void setSenhaAtual(String senhaAtual) {
    this.senhaAtual = senhaAtual;
}

public String getSenhaNova() {
    return senhaNova;
}

public void setSenhaNova(String senhaNova) {
    this.senhaNova = senhaNova;
}

public String getSenhaConfirma() {
    return senhaConfirma;
}

public void setSenhaConfirma(String senhaConfirma) {
    this.senhaConfirma = senhaConfirma;
}

public String alteraSenha(){

    Senha senhaAtualBD =
this.buscaSenhaAtual(credentials.getUsername());

    if (senhaAtual.equals(senhaAtualBD.getSenha())){
        if (senhaNova.equals(senhaConfirma)){
            senhaAtualBD.setSenha(senhaNova);
            entityManager.persist(senhaAtualBD);
            entityManager.flush();
        }
    }
}

```

```

        facesMessages.clear();
        facesMessages.add(Severity.INFO
, "#{messages.senhaAlterada}");
        return Constantes.SENHA_ALTERADA;
    }
    else {
        facesMessages.add(Severity.WARN
, "#{messages.senhasDiferentes}");
    }
    }else {
        facesMessages.add(Severity.WARN
, "#{messages.senhaAtualInvalida}");
    }

    return null;
}

private Senha buscaSenhaAtual(String email){

    Senha senha = (Senha) entityManager.createQuery("from Senha where
email =:pemail")
                                                .setParameter("pemail",
email)
                                                .getSingleResult();

    return senha;
}

}

package br.ufsc.inf.faccin.efagro.session;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import org.jboss.seam.annotations.In;

```

```

import org.jboss.seam.annotations.Name;
import org.jboss.seam.annotations.security.Restrict;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;

import br.ufsc.inf.faccin.efagro.entity.AtividadeTalhao;
import br.ufsc.inf.faccin.efagro.entity.OcorrenciaPraga;
import br.ufsc.inf.faccin.efagro.entity.Operacao;
import br.ufsc.inf.faccin.efagro.entity.Safra;
import br.ufsc.inf.faccin.efagro.entity.Talhao;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

@Name("talhaoHome")
@Restrict("#{identity.loggedIn}")
public class TalhaoHome extends EntityHome<Talhao> {

    /**
     *
     */
    private static final long serialVersionUID = 4719189533144762010L;

    @In(create = true)
    SafraHome safraHome;

    @In
    EntityManager entityManager;

    @In
    FacesMessages facesMessages;

    private double totalCustoAdubacao;
    private double totalCustoDefensivo;
    private double totalCustoPlantio;
    private double totalBrutoColheita;
    private double saldoTalhao;
    private double totalHorasTrabalhadas;

```

```

private double totalCustoOperacoes;

private ArrayList<Operacao> operacoesDoTalhao = new
ArrayList<Operacao>();

public void setTalhaoIdTalhao(Long id) {
    setId(id);
}

public Long getTalhaoIdTalhao() {
    return (Long) getId();
}

@Override
protected Talhao createInstance() {
    Talhao talhao = new Talhao();
    return talhao;
}

public void wire() {
    getInstance();
    Safra safra = safraHome.getDefinedInstance();
    if (safra != null) {
        getInstance().setSafra(safra);
    }
}

public boolean isWired() {
    if (getInstance().getSafra() == null)
        return false;
    return true;
}

public Talhao getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

```



```

    public List<AtividadeTalhao> getAtividadeTalhaos() {
        return getInstance() == null ? null : new
ArrayList<AtividadeTalhao>(
            getInstance().getAtividadeTalhaos());
    }

    public List<OcorrenciaPraga> getOcorrenciaPragas() {
        return getInstance() == null ? null : new
ArrayList<OcorrenciaPraga>(
            getInstance().getOcorrenciaPragas());
    }

    public double getTotalCustoAdubacao() {
        return totalCustoAdubacao;
    }

    public void setTotalCustoAdubacao(double totalCustoAdubacao) {
        this.totalCustoAdubacao = totalCustoAdubacao;
    }

    public double getTotalCustoDefensivo() {
        return totalCustoDefensivo;
    }

    public void setTotalCustoDefensivo(double totalCustoDefensivo) {
        this.totalCustoDefensivo = totalCustoDefensivo;
    }

    public double getTotalCustoPlantio() {
        return this.totalCustoPlantio;
    }

    public void setTotalCustoPlantio(double totalCustoPlantio) {
        this.totalCustoPlantio = totalCustoPlantio;
    }

```

```

public double getTotalBrutoColheita() {
    return this.totalBrutoColheita;
}

public void setTotalBrutoColheita(double totalBrutoColheita) {
    this.totalBrutoColheita = totalBrutoColheita;
}

public void setSaldoTalhao(double totalLiquidoColheita) {
    this.saldoTalhao = totalLiquidoColheita;
}

public double getSaldoTalhao() {
    return this.saldoTalhao;
}

public void setOperacoesDoTalhao(ArrayList<Operacao> operacoesDoTalhao)
{
    this.operacoesDoTalhao = operacoesDoTalhao;
}

public ArrayList<Operacao> getOperacoesDoTalhao() {
    return this.operacoesDoTalhao;
}

public double getTotalHorasTrabalhadas() {
    return totalHorasTrabalhadas;
}

public void setTotalHorasTrabalhadas(double totalHorasTrabalhadas) {
    this.totalHorasTrabalhadas = totalHorasTrabalhadas;
}

public double getTotalCustoOperacoes() {
    return totalCustoOperacoes;
}

```

```

public void setTotalCustoOperacoes(double totalCustoOperacoes) {
    this.totalCustoOperacoes = totalCustoOperacoes;
}

public String inserir(){

    this.getInstance().setSafra(safraHome.getInstance());

    try {

        this.persist();
        facesMessages.clear();

        facesMessages.add(Severity.INFO, "#{messages.talhaoCadastrado}");

        entityManager.flush();

    } catch (PersistenceException e) {

        facesMessages.clear();

        facesMessages.add(Severity.ERROR, "#{messages.erroCadastrarTalhao}");

        return Constantes.TALHAO_INCLUIDO_ERRO;
    }

    entityManager.refresh(safraHome.getInstance());

    return Constantes.TALHAO_INCLUIDO_OK;
}

/**
 * Efetua o somatório dos custos do talhao
 * **/
public void calculaCustosTalhao(){

    this.setaZeroParaOsValores();
}

```

```

        Iterator<AtividadeTalhao> it =
this.getInstance().getAtividadeTalhaos().iterator();

        while (it.hasNext()) {

            AtividadeTalhao atividade = (AtividadeTalhao) it.next();

            if (atividade.getAdubacao() != null){
                totalCustoAdubacao = totalCustoAdubacao +
atividade.getAdubacao().getValorTotal();
            }

            if (atividade.getAplicacaoDefensivo() != null){
                totalCustoDefensivo = totalCustoDefensivo +
atividade.getAplicacaoDefensivo().getValorTotal();
            }

            if (atividade.getPlantio() != null){
                totalCustoPlantio = totalCustoPlantio +
atividade.getPlantio().getValorTotal();;
            }

            if (atividade.getColheita() != null){
                totalBrutoColheita = totalBrutoColheita +
atividade.getColheita().getValorTotal();
            }

            this.calculaTotaisOperacao(atividade.getOperacoes().iterator());

        }

        saldoTalhao = totalBrutoColheita - (totalCustoAdubacao +
totalCustoDefensivo + totalCustoPlantio + totalCustoOperacoes);

    }

```

```

private void setaZeroParaOsValores() {

    entityManager.refresh(this.getInstance());

    operacoesDoTalhao = new ArrayList<Operacao>();

    totalCustoAdubacao           =0;
    totalCustoDefensivo          =0;
    totalCustoPlantio            =0;
    totalBrutoColheita           =0;
    saldoTalhao                  =0;
    totalHorasTrabalhadas        =0;
    totalCustoOperacoes         =0;

}

public void calculaTotaisOperacao(Iterator<Operacao> iterator){

    while (iterator.hasNext()) {

        Operacao operacao = (Operacao) iterator.next();

        operacoesDoTalhao.add(operacao);
        totalHorasTrabalhadas = totalHorasTrabalhadas +
operacao.getHorasTrabalhadas();
        totalCustoOperacoes = totalCustoOperacoes +
(operacao.getHorasTrabalhadas() * operacao.getMaquina().getValorHora());
    }

}

}

package br.ufsc.inf.faccin.efagro.session;

import javax.persistence.EntityManager;
import javax.persistence.PersistenceException;

import br.ufsc.inf.faccin.efagro.entity.*;

```

```

import br.ufsc.inf.faccin.efagro.tools.ChaveAleatoria;
import br.ufsc.inf.faccin.efagro.tools.Constantes;

import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Logger;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.faces.FacesMessages;
import org.jboss.seam.faces.Renderer;
import org.jboss.seam.framework.EntityHome;
import org.jboss.seam.international.StatusMessage.Severity;
import org.jboss.seam.log.Log;

@Name("usuarioHome")
public class UsuarioHome extends EntityHome<Usuario> {

    /**
     *
     */
    private static final long serialVersionUID = -8125592490620031093L;

    /**
     * Atributo provisório
     */
    private String ale;

    @In(create = true)
    PropriedadeHome propriedadeHome;

    @In(create=true)
    private Renderer renderer;

    @Logger
    private Log log;

    @In
    EntityManager entityManager;

```

```

@In
FacesMessages facesMessages;

public void setUsuarioIdUsuario(Long id) {
    setId(id);
}

public Long getUsuarioIdUsuario() {
    return (Long) getId();
}

@Override
protected Usuario createInstance() {
    Usuario usuario = new Usuario();
    return usuario;
}

public void wire() {
    getInstance();
    Propriedade propriedade = propriedadeHome.getDefinedInstance();
    if (propriedade != null) {
        getInstance().setPropriedade(propriedade);
    }
}

public boolean isWired() {
    return true;
}

public Usuario getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

public void setAle(String ale) {
    this.ale = ale;
}

public String getAle() {

```

```

        return ale;
    }

    public String inserir(){

        if (this.getInstance().getPropriedade() == null){
            this.getPropriedadeDaSessao();
        }

        // Persiste usuario
        try {

            this.persist();

            entityManager.flush();

        } catch (PersistenceException e) {

            log.error(e.getMessage());
            log.error("Usuario Home" + "Erro criar Usuario");
            facesMessages.clear();

            facesMessages.add(Severity.ERROR, "#{messages.erroCadastrarUsuario}");

            return Constantes.USUARIO_INCLUIDO_ERRO;
        }

        // Cria e persiste senha aleatoria
        try {

            Senha senha = new Senha();

            senha.setEmail(this.getInstance().getEmail());
            senha.setSenha(new ChaveAleatoria(10).getChave());

```



```

        entityManager.persist(senha);
        entityManager.flush();

    } catch (PersistenceException e) {

        log.error(e.getMessage());
        log.error("Usuario Home" + "Erro criar Senhal");
        facesMessages.clear();

facesMessages.add(Severity.ERROR, "#{messages.erroCadastrarUsuario}");

        return Constantes.USUARIO_INCLUIDO_ERRO;

    }

// Envia email com senha aleatoria
try {

    this.enviaEmailParaUsuario();

} catch (Exception e) {

    log.error(e.getMessage());
    log.error("Usuario Home" + "Erro envio de email");
    facesMessages.clear();

facesMessages.add(Severity.ERROR, "#{messages.erroCadastrarUsuario}");

        return Constantes.USUARIO_INCLUIDO_ERRO;

    }

entityManager.refresh(propriedadeHome.getInstance());

facesMessages.clear();
facesMessages.add(Severity.INFO, "#{messages.usuarioCadastrado}");

```

```

        return Constantes.USUARIO_INCLUIDO_OK;
    }

    public String alterar(){

        try {

            this.update();

            entityManager.flush();
        } catch (PersistenceException e) {

            log.error("Usuario Home - alterar usuario - "+
e.getMessage());

            facesMessages.clear();

            facesMessages.add(Severity.ERROR, "#{messages.erroAlterarUsuario}");

            return Constantes.USUARIO_ALTERADO_ERRO;
        }

        facesMessages.clear();
        facesMessages.add(Severity.INFO, "#{messages.usuarioAlterado}");
        return Constantes.USUARIO_ALTERADO_OK;
    }

    public String deletar(){

        try {

            this.remove();

        } catch (PersistenceException e) {

            log.error("Usuario Home - remover usuario - "+
e.getMessage());

```

```

        facesMessages.clear();

facesMessages.add(Severity.ERROR, "#{messages.erroRemoverUsuario}");

        return Constantes.USUARIO_REMOVIDO_ERRO;
    }

    this.removerSenha();

    facesMessages.clear();
    facesMessages.add(Severity.ERROR, "#{messages.usuarioAlterado}");
    return Constantes.USUARIO_REMOVIDO_OK;
}

private void removerSenha() {

    try {
        Senha senha = (Senha) entityManager.createQuery("from Senha
where email =:pemail")

        .setParameter("pemail", this.getInstance().getEmail())

        .getSingleResult();

        entityManager.remove(senha);
        entityManager.flush();

    } catch (Exception e) {
        log.error("Usuario Home - remover Senha usuario - "+
e.getMessage());
    }

}

private void getPropriedadeDaSessao() {

```

```

        propriedadeHome.carregaPropriedade();
        this.getInstance().setPropriedade(propriedadeHome.getInstance());
    }

    private void enviaEmailParaUsuario() throws Exception{

        renderer.render("/emails/cadUsuario.xhtml");

    }

}

package br.ufsc.inf.faccin.efagro.session;

import br.ufsc.inf.faccin.efagro.entity.*;
import java.util.ArrayList;
import java.util.List;
import org.jboss.seam.annotations.In;
import org.jboss.seam.annotations.Name;
import org.jboss.seam.framework.EntityHome;

@Name("variedadeHome")
public class VariedadeHome extends EntityHome<Variedade> {

    /**
     *
     */
    private static final long serialVersionUID = -8310913244373462864L;
    @In(create = true)
    CulturaHome culturaHome;
    @In(create = true)
    FabricanteHome fabricanteHome;

    public void setVariedadeIdVariedade(Integer id) {
        setId(id);
    }
}

```

```

public Integer getVariedadeIdVariedade() {
    return (Integer) getId();
}

@Override
protected Variedade createInstance() {
    Variedade variedade = new Variedade();
    return variedade;
}

public void wire() {
    getInstance();
    Cultura cultura = culturaHome.getDefinedInstance();
    if (cultura != null) {
        getInstance().setCultura(cultura);
    }
    Fabricante fabricante = fabricanteHome.getDefinedInstance();
    if (fabricante != null) {
        getInstance().setFabricante(fabricante);
    }
}

public boolean isWired() {
    if (getInstance().getCultura() == null)
        return false;
    return true;
}

public Variedade getDefinedInstance() {
    return isIdDefined() ? getInstance() : null;
}

public List<Plantio> getPlantios() {
    return getInstance() == null ? null : new ArrayList<Plantio>(
        getInstance().getPlantios());
}
}

```