



Universidade Estadual de Campinas
Faculdade de Engenharia de Alimentos
Departamento de Engenharia de Alimentos



**DESENVOLVIMENTO DE FERRAMENTAS
COMPUTACIONAIS PARA A AVALIAÇÃO EMERGÉTICA
DE SISTEMAS AGRÍCOLAS**

Fábio Takahashi

Engenheiro de Alimentos

Prof. Dr. Enrique Ortega Rodriguez

Orientador

Campinas - SP

2007

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA FEA – UNICAMP

T139d Takahashi, Fábio
Desenvolvimento de ferramentas computacionais para avaliação
emergética de sistemas agrícolas / Fábio Takahashi. -- Campinas, [s.n.],
2007.

Orientador: Enrique Ortega Rodriguez
Dissertação (mestrado) – Universidade Estadual de Campinas.
Faculdade de Engenharia de Alimentos

1. Avaliação emergética. 2. Java (Linguagem de programação de
computador). 3. XML. 4. Aplicativo web. 5. Struts. 6. JSP. I.
Ortega Rodriguez, Enrique. II. Universidade Estadual de
Campinas.Faculdade de Engenharia de Alimentos. III. Título.

(cars/fea)

Titulo em inglês: Development of computational tools for the emergy evaluation of agricultural
systems

Palavras-chave em inglês (Keywords): Emergy evaluation, Java (Computer program language),
XML, Web application, Struts, JSP

Titulação: Mestre em Engenharia de Alimentos

Banca examinadora: Enrique Ortega Rodriguez
José Maria Gusman Ferraz
Luis Alberto Ambrósio
Paulo Roberto Beskow

Programa de Pós Graduação: Programa em Engenharia de Alimentos



Universidade Estadual de Campinas
Faculdade de Engenharia de Alimentos
Departamento de Engenharia de Alimentos



DESENVOLVIMENTO DE FERRAMENTAS COMPUTACIONAIS PARA A AVALIAÇÃO EMERGÉTICA DE SISTEMAS AGRÍCOLAS

Dissertação apresentada a Faculdade de Engenharia de Alimentos da Universidade Estadual de Campinas, para a obtenção do Título de Mestre em Engenharia de Alimentos.

Fábio Takahashi

Engenheiro de Alimentos

Prof. Dr. Enrique Ortega Rodriguez

Orientador

Campinas - SP

2007

BANCA EXAMINADORA

Prof. Dr. Enrique Ortega Rodriguez
Universidade Estadual de Campinas - Unicamp
(Orientador)

Dr. José Maria Gusman Ferraz
EMBRAPA-CNPMA
(Membro)

Dr. Luis Alberto Ambrósio
Instituto Agronômico de Campinas - IAC
(Membro)

Prof. Dr. Paulo Roberto Beskow
Universidade Federal de São Carlos - UFSCar
(Membro)

Dedico
*aos meus pais **Tieko e Nelson,***
*ao meu irmão **William,***
e a todos que me apoiaram nessa caminhada.

AGRADECIMENTOS

À minha família por todo amor, incentivo e apoio que sempre me concederam;

Ao Prof. Dr. Enrique Ortega pela orientação;

Aos professores membros da banca pelas sugestões e contribuições apresentadas;

*Aos colegas do laboratório de Engenharia Ecológica e Informática Aplicada pela
companhia e ajuda durante a execução do trabalho;*

A todos os colegas do mestrado, pela companhia e momentos de estudo;

*Aos amigos Vívian, Geraldo e Rafael pela companhia, conversas e momentos de
descontração;*

*À Faculdade de Engenharia de Alimentos/Unicamp especialmente ao Departamento de
Engenharia de Alimentos pela oportunidade;*

À CNPq pela ajuda financeira;

Meus mais sinceros agradecimentos.

“O mundo é um lugar perigoso de se viver, não por causa daqueles que fazem o mal, mas sim por causa daqueles que observam e deixam o mal acontecer.”

Albert Einstein.

RESUMO

Este trabalho teve como objetivo o desenvolvimento de ferramentas computacionais para a realização da avaliação emergética de sistemas agrícolas utilizando as tecnologias XML, XSLT, Java, JSP, Struts e banco de dados MySQL. Na primeira etapa do trabalho, foram desenvolvidos documentos XML para representar os sistemas de agricultura química da Flórida e do Brasil. Um arquivo XSLT foi criado para processar e calcular o fluxo de energia para cada recurso e visualizar os resultados utilizando o navegador Internet Explorer. As informações registradas nos arquivos XML são: tipo de recurso (contribuição da natureza e da economia humana), quantidade utilizada, unidade de medida, fator de conversão de unidades e transformidades solares. A análise emergética dos sistemas agrícolas pode ser feita de maneira fácil, através da modificação dos campos “quantidade de recursos utilizados” nos arquivos XML. Na segunda etapa um aplicativo java foi desenvolvido para abrir os arquivos XML e modificar esses campos. Na terceira etapa foram desenvolvidas páginas web dinâmicas utilizando-se as tecnologias JSP e Struts. Com esta ferramenta o usuário pode escolher o sistema, alterar os valores dos recursos utilizados, gravar esses dados no banco de dados e visualizar os resultados. As principais conclusões deste trabalho são: (a) a utilização da tecnologia XML e Java facilita a análise emergética dos sistemas agrícolas; (b) disponibilizando estas ferramentas na internet, o diagnóstico emergético pode ser difundido e amplamente utilizado, sem perder seus padrões de qualidade; (c) esta metodologia pode ser adaptada para representar todos os sistemas agrícolas brasileiros, incluindo os modelos ecológicos como também os químicos.

Palavras Chave: Avaliação emergética, Java (linguagem de programação de computador), XML, aplicativo web, struts, JSP.

ABSTRACT

This work had the goal of developing computational tools to make the energy evaluation of agricultural systems using the technologies XML, XSLT, Java, JSP, Struts and the MySQL database. At the first stage of this work, XML documents were developed to represent the chemical agricultural systems of Florida and Brazil. A XSLT archive was created to process and to calculate the energy flows for each resource and to visualize the results using the Internet Explorer Navigator. The information stored on the XML archives are: type of resources (contribution of nature and economy), amount used, units of measure, units conversion factors and solar transformities. The energy analysis of agricultural systems can be done in an easy way, through the modification of the “quantity of resources used” fields on the XML archives. At the second stage, a java application was developed to open the XML archives and to modify these fields. At the third stage dynamic web pages were developed using the technologies JSP and Struts. With this tool the user can choose the system, modify the values of used resources, save these data on the database and visualize the results. The main conclusions of this work are: (a) the use of XML and Java technologies make easier the energy analysis of agricultural systems; (b) making these tools available at the internet, the energy diagnosis can be spread out and widely used, without losing its quality standards; (c) these methodologies can be adapted to represent all the Brazilian agricultural systems, including the ecological models as well as the chemical models.

Key Words: Energy evaluation, Java (computer program language), XML, web application, Struts, JSP.

Sumário

RESUMO.....	viii
ABSTRACT.....	ix
1. INTRODUÇÃO.....	1
1.1 Hipótese do trabalho.....	3
2. OBJETIVOS.....	4
2.1 Objetivos específicos.....	4
3. REVISÃO BIBLIOGRÁFICA.....	5
3.1 Agricultura e Sustentabilidade.....	5
3.1.1 Breve histórico da agricultura.....	5
3.1.2 Sustentabilidade e desenvolvimento sustentável.....	7
3.2 Análise Emergética.....	8
3.3 Páginas Web Dinâmicas.....	11
3.3.1 XML e XSLT.....	11
3.3.2 Java, banco de dados e aplicativos web.....	13
3.3.3 JavaServer Pages e Servlet.....	15
3.3.4 Struts <i>Framework</i>	16
4. MATERIAL E MÉTODOS.....	19
4.1 Análise emergética.....	19
4.1.1 1ª Etapa:Visão panorâmica pelos diagramas sistêmicos.....	19
4.1.2 2ª Etapa – Tabelas de avaliação emergética.....	20
4.1.3 3ª Etapa- Cálculo dos índices emergéticos.....	22
4.1.4 Emdólar.....	24
4.2 Construção dos arquivos XML.....	24
4.2.1 Arquivos XML para representar o sistema da Flórida.....	26
4.2.2 Arquivos XML para representar os sistemas agrícolas do Brasil.....	36
4.3 Construção do arquivo XSLT.....	37
4.4 Desenvolvimento do Aplicativo.....	40
4.5 Desenvolvimento de páginas web dinâmicas.....	41
4.5.1 Construção do Módulo Web.....	42
4.5.2 Construção do Banco de Dados.....	42
4.5.3 Criação dos arquivos JSP.....	45
4.5.4 Parte lógica das páginas JSP com o struts <i>framework</i>	46
5. RESULTADOS E DISCUSSÃO.....	53
5.1 O aplicativo desenvolvido.....	53
5.2 Páginas web dinâmicas desenvolvidas.....	58
5.2.1. Páginas de resposta.....	65
6. CONCLUSÕES.....	69
7. REFERENCIAS BIBLIOGRÁFICAS.....	71
Apêndice A – Unidades, fatores de correção e transformidades utilizados no sistema da Flórida.....	75
Cálculo do recurso sol (sun).....	75
Cálculo do recurso chuva (rain).....	75
Cálculo do recurso evapotranspiração (evapotranspiration).....	75

Cálculo do recurso perda de solo (net topsoil loss)	75
Cálculo do recurso combustível (fuel)	76
Cálculo do recurso eletricidade (electricity)	76
Cálculo do recurso cal (lime)	76
Cálculo do recurso pesticidas (pesticides)	76
Cálculo do recurso potássio (potash)	76
Cálculo do recurso fosfato (phosphate)	77
Cálculo do recurso nitrogênio (nitrogen)	77
Cálculo do recurso grãos (feed, grain)	77
Cálculo do recurso trabalho humano (human labor)	77
Cálculo do recurso serviços da economia (services from economy)	77
Apêndice B – Código do arquivo XSLT	78
Apêndice C – Arquivo DTD dos arquivos XML	84
Apêndice D – Código do arquivo CampoActionForm.java	85

LISTA DE FIGURAS

Figura 1	- Modelo MVC utilizado pelo struts framework.	18
Figura 2	- Diagrama de fluxos de energia do sistema.	20
Figura 3	- Esquema de organização de uma tabela de cálculo de fluxos de energia	20
Figura 4	- Esquema da tabela para calcular a energia produzida e o dinheiro recebido pelo sistema	21
Figura 5	- Representação simplificada de um sistema produtivo.	22
Figura 6	- Esquema de um documento XML.	25
Figura 7	- Inicialização do arquivo XML.	28
Figura 8	- Definição do sistema no arquivo XML	28
Figura 9	- Representação dos recursos renováveis no arquivo XML	29
Figura 10	- Representação dos recursos não renováveis, materiais e serviços no arquivo XML.	30
Figura 11	- Representação das variáveis as variáveis R, N, M, S, F e Y da metodologia emergética nos arquivos XML.	31
Figura 12	- Tags do arquivo XML que se referem a dados do produto estudado.	32
Figura 13	- Tags que representam os resultados	33
Figura 14	- Tags com as informações dos índices emergéticos.	35
Figura 15	- Tag utilizadas para referenciar os arquivos gráficos.	36
Figura 16	- Gráfico criado pelo arquivo XSLT para visualizar o resultado.	40
Figura 17	- Tags que importam as bibliotecas de tags personalizadas do struts framework.	45
Figura 18	- Parte do código do arquivo loginActionForm.java	46
Figura 19	- Campos da página login.jsp	47
Figura 20	- Esquema de encaminhamento de páginas com o struts.	48
Figura 21	- Parte do arquivo struts-config.xml referenciando as ações de login.jsp	48
Figura 22	- Tela do formulário de dados que pode ser preenchido pelo usuário.	49

Figura 23	- Esquema do encaminhamento de páginas do processo de criar uma novas análises.	50
Figura 24	- Parte do arquivo resposta.jsp	52
Figura 25	- Interface do aplicativo Emergy Table.	53
Figura 26	- Caixa de diálogo para abrir um arquivo XML.	54
Figura 27	- Aplicativo com o arquivo bellPepperData.xml aberto.	54
Figura 28	- Janela para modificar os dados do produto.	55
Figura 29	- Resultado apresentado no Internet Explorer.	55
Figura 30	- Esquema de todo o processo realizado na execução emergytable (TAKAHASHI et al. 2006)	56
Figura 31	- Esquema simplificado das páginas criadas.	58
Figura 32	- Parte da página de login.	59
Figura 33	- Parte da página de cadastro de usuário.	59
Figura 34	- Parte da página para recuperar a senha.	60
Figura 35	- Página de escolha do sistema desejado	60
Figura 36	- Parte da página inicial.	61
Figura 37	- Parte da página de escolha do produto.	61
Figura 38	- Formulário para alterar os valores dos recursos utilizados.	62
Figura 39	- Página informando que os dados foram gravados com sucesso.	63
Figura 40	- Parte da página que informa ao usuário os erros encontrados.	63
Figura 41	- Análises gravadas pelo usuário.	64
Figura 42	- Parte da página que mostra que uma análise foi aberta com sucesso.	64
Figura 43	- Parte da página para confirmar se o usuário realmente deseja apagar a análise.	65
Figura 44	- Tag do arquivo XSLT para criar a variável R1	66
Figura 45	- Tag do arquivo XSLT utilizada para criar a variável M do produto Trigo-1500.	67
Figura 46	- Tag do arquivo XSLT utilizada para criar a variável M do produto Banana.	67

LISTA DE TABELAS

Tabela 1	- Produtos da agricultura da Flórida estudados	26
Tabela 2	- Recursos utilizados na elaboração dos arquivos XML da Flórida.	27
Tabela 3	- Tabela construída pelo documento XSLT para mostrar os recursos utilizados.	37
Tabela 4	- Tabela construída pelo arquivo XSLT para mostrar os dados do produto.	38
Tabela 5	- Tabela criada pelo documento XSLT para representar o fluxo energético agregado.	38
Tabela 6	- Tabela criada pelo arquivo XSLT para mostrar o resultado referente ao produto.	39
Tabela 7	- Tabela criada pelo arquivo XSLT para mostrar os índices energéticos calculados.	39
Tabela 8	- Esquema para a tabela cadastro	43
Tabela 9	- Esquema para a tabela login.	43
Tabela 10	- Esquema para a tabela recursos	44

1. INTRODUÇÃO

Os seres humanos têm causado um impacto considerável sobre a Terra, principalmente devido ao crescimento populacional e ao desenvolvimento econômico. Nós enfrentamos uma série de contradições quando falamos de crescimento econômico e preservação dos recursos naturais. Essas contradições se devem principalmente ao crescimento populacional, redução dos recursos naturais (renováveis e não renováveis) e deterioração do meio ambiente (ZHAO; LI; LI, 2005).

O mundo enfrenta um importante desafio que é alimentar uma população que cresce mais a cada dia e ao mesmo tempo se adaptar a diminuição das fontes de energia e dos recursos naturais (MARTIN et al., 2006). O grande desafio é conciliar a produção de alimentos e a conservação ambiental na agricultura moderna, que depende dos recursos da natureza como a luz do sol, vento, água e solo e recursos da economia como fertilizantes, pesticidas, combustível, eletricidade, equipamentos mecânicos e outros produtos industriais (CHEN et al., 2006).

A atividade agrícola sempre produziu impactos negativos ao meio ambiente, mas esses impactos se agravaram a partir da década de setenta com a Revolução Verde. A Revolução Verde trouxe o aumento da produtividade agrícola com a mecanização e o uso intenso de fertilizantes químicos, cuja fabricação consome muito petróleo e libera muito dióxido de carbono. Isto foi feito sem se preocupar com os danos que essas práticas causariam. Hoje já se pode notar que elas trouxeram diversos problemas, tais como a

erosão, a poluição química, a perda de fertilidade do solo e a perda da biodiversidade, entre outros (EHLERS, 1996).

Os sistemas agroecológicos são aqueles que não agredem o meio ambiente e seriam uma boa alternativa para combater os males causados pela agricultura tradicional. Esses sistemas têm base no conceito de sustentabilidade ou desenvolvimento sustentável. (FAGNANI, 1997).

Para tornar a sustentabilidade uma realidade, nós devemos medir em que situação nós estamos agora e onde queremos chegar. Métodos de medição da sustentabilidade têm sido utilizados para se fazer análises qualitativas e quantitativas. A avaliação emergética proposta por Odum (1996) pode estimar a demanda de recursos utilizados pelo homem e verificar a situação dos recursos naturais utilizados (ZHAO; LI; LI, 2005).

Esta análise é feita através de um balanço termodinâmico integral dos sistemas agrícolas. Este balanço pode ser feito uma vez que todas as energias, dos serviços e produtos do sistema, são transformadas em uma mesma unidade. A energia solar é a energia solar requerida direta ou indiretamente para se produzir um serviço ou produto. Sua unidade é o emjoule (sej) (ODUM, 1996).

Considerando os graves problemas ambientais causados pelo homem e sabendo dos potenciais da metodologia emergética, este trabalho propõe a criação de ferramentas computacionais para a realização da análise emergética de sistemas agrícolas, com o objetivo de torná-la mais difundida e mais fácil de ser realizada.

Neste estudo foram utilizadas as tecnologias XML, XSLT, Java, JSP, Struts e banco de dados MySQL. Essas tecnologias foram escolhidas pois XML está se tornando o padrão de arquivos para troca de dados entre aplicativos web e através do XSLT podemos apresentar os dados dos arquivos XML em vários formatos. Java é uma linguagem de programação orientada a objetos que possui muitas tecnologias desenvolvidas para manipulação de arquivos XML. JSP oferece vários benefícios para a criação de páginas web com conteúdo dinâmico e através do Struts a criação dessas páginas pode ser simplificada. O banco de dados MySQL foi escolhido porque é um dos mais utilizados atualmente e por se tratar de um banco de dados livre.

1.1 Hipótese do trabalho

É possível armazenar dados de sistemas agrícolas em arquivos XML e construir aplicativos para utilizar essas informações, modificá-las e obter tanto pela internet quanto de modo local (sem conexão com a rede) os valores dos indicadores de desempenho emergético.

2. OBJETIVOS

O objetivo deste trabalho é a construção de ferramentas computacionais para a avaliação emergética de sistemas agrícolas.

2.1 Objetivos específicos

1. Criar arquivos XML e XSLT para armazenar e visualizar os dados dos sistemas agrícolas.
2. Desenvolver um aplicativo java que modifique os arquivos XML.
3. Desenvolver páginas web dinâmicas para a avaliação emergética dos sistemas agrícolas representados nos arquivos XML.
4. Obter os índices emergéticos de uma forma fácil e rápida.
5. Disponibilizar estes recursos para difundir amplamente a avaliação emergética.

3. REVISÃO BIBLIOGRÁFICA

3.1 Agricultura e Sustentabilidade

3.1.1 Breve histórico da agricultura

A prática do cultivo da terra, ou agricultura, teve início há mais ou menos dez mil anos, quando alguns povos do norte da África e do oeste asiático abandonaram progressivamente a caça e a coleta de alimentos e começaram a produzir seus próprios grãos (EHLERS, 1996).

No começo da agricultura, as tecnologias se caracterizavam pela rotação de culturas e a integração entre atividades de produção vegetal e animal, que respeitavam o meio ambiente e as limitações ecológicas. Atualmente o que se vê é que o homem usa o conhecimento e a tecnologia para tentar superar essas limitações (ASSIS, 2002).

As técnicas agrícolas foram aprimoradas após 1945 pelas descobertas científicas e o desenvolvimento de vários avanços tecnológicos que se consolidaram na década de 1970 com a chamada Revolução Verde, que utilizava maciçamente insumos industriais e energia fóssil, como os fertilizantes químicos, os agrotóxicos, os motores de combustão interna e as variedades vegetais de alto potencial produtivo. Isso resultou em grandes safras, mas trouxeram também uma série de problemas ambientais e sociais como a destruição das florestas e sua biodiversidade, a erosão dos solos e a contaminação dos recursos naturais e dos alimentos. No Brasil, além desses problemas, aumentou-se a concentração de terras e de riquezas provocando um intenso êxodo rural para os grandes centros industrializados (EHLERS, 1996).

A Revolução Verde fundamentava-se na melhoria do desempenho dos índices de produtividade agrícola, por meio da substituição dos moldes de produção locais ou tradicionais, por um conjunto bem mais homogêneo de práticas tecnológicas que criaram as condições necessárias para a produção em larga escala dos sistemas monoculturais (EHLERS, 1996).

Rapidamente a Revolução Verde se espalhou por vários países, quase sempre apoiada por órgãos governamentais, pela grande maioria da comunidade agrônômica e pelas empresas produtoras de insumos (EHLERS, 1996).

A notável expansão das técnicas e capacidades de produção e o crescimento acelerado da população mundial, especialmente durante a segunda metade do século XX, têm mostrado que os recursos naturais e os serviços derivados destes recursos não são ilimitados e que a falta e a escassez desses recursos são sérios problemas para a humanidade. Com isso pode-se verificar que, mesmo com o surgimento de novas tecnologias e o avanço da ciência, o homem não pode ultrapassar os limites impostos pela natureza (ESPINOSA, 1993).

A euforia das “grandes safras” propiciadas pelo padrão tecnológico da Revolução Verde logo cederia lugar a uma série de preocupações relacionadas tanto aos seus impactos sócio-ambientais quanto à sua viabilidade energética (EHLERS, 1996).

Em meados da década de 80, cresciam as preocupações relacionadas à qualidade de vida e aos problemas ambientais contemporâneo, como a poluição, o aquecimento global, a destruição da camada de ozônio, a erosão dos solos e a dilapidação das florestas e da biodiversidade genética (EHLERS, 1996).

Com o aumento da conscientização das pessoas devido aos movimentos ambientalistas, muitas pessoas começam a se voltar para as antigas práticas de agricultura. Surgiu, na década de 1970, a agroecologia, uma ciência que adota como princípios básicos a menor dependência possível de insumos externos e a conservação dos recursos naturais. Para isto os sistemas agroecológicos procuram maximizar a reciclagem de nutrientes e o aproveitamento de resíduos com potencial energético. Esta ciência resgata, sob novas bases tecnológicas e econômicas, a lógica da complexificação das sociedades camponesas tradicionais e seus conhecimentos desprezados pela agricultura moderna como forma de vencer o desafio de estabelecer uma agricultura sustentável (ASSIS, 2002).

3.1.2 Sustentabilidade e desenvolvimento sustentável

Sustentabilidade, segundo Hansen (1995) é a capacidade de um sistema continuar a existir no futuro. O conceito de sustentabilidade também tem sido aplicado para caracterizar o tipo de desenvolvimento: “desenvolvimento sustentável”. Em um estudo de prospecto, produzido pelo Instituto de Tecnologia de Massachusetts (MIT) no final da década de 60 demonstrou a exaustão dos recursos naturais e a incapacidade dos ecossistemas de se recuperar dos danos causados pelo homem, e anunciaram uma situação crítica que poderia ser alcançada em um curto período de tempo. Quatro décadas depois, um segundo relatório foi produzido pelo mesmo instituto e infelizmente os resultados confirmaram as mais pessimistas previsões (MORA, 2007).

Em 1987, a comissão Mundial sobre Meio Ambiente e Desenvolvimento publicava “Nosso futuro comum”, o famoso Relatório Brundtland, que ajudou a disseminar o ideal de um desenvolvimento sustentável para diferentes setores da sociedade moderna, como a

agricultura e a economia. A Conferência das Nações Unidas sobre Meio Ambiente e Desenvolvimento, a Rio-92, reafirmou este ideal (EHLERS, 1996).

Para Sachs (1986) o desenvolvimento sustentável é caracterizado pelo crescimento econômico baseado na justiça social e sustentabilidade no uso dos recursos naturais. Esses requisitos deveriam ser alcançados harmoniosamente, ou seja, as suas inter-relações deveriam ocorrer de forma balanceada. Para se conseguir isso é necessário ver o sistema como um todo, levando em conta os aspectos culturais, ecológicos, econômicos, sociais e espaciais dos sistemas.

3.2 Análise Emergética

A teoria e os conceitos básicos da metodologia emergética são fundamentados na termodinâmica e na teoria geral dos sistemas e é uma poderosa ferramenta para avaliar a sustentabilidade. A metodologia foi utilizada inicialmente para descrever sistemas naturais, como os relacionados à agricultura, florestas, economias regionais etc. (WANG; ZHANG; NI, 2005).

Atualmente a metodologia emergética já é utilizada em diversas áreas, Wang, Zhang e Ni (2005) utilizaram para avaliar eco-parques industriais e Björklund, Geber e Rydberg (2001) utilizaram a metodologia para avaliar sistemas de tratamentos de efluentes.

Muitos trabalhos utilizando a metodologia emergética têm sido realizados na China, entre eles podemos citar Liu e Chen (2007) que analisaram a eficiência e a sustentabilidade da produção de grãos em duas províncias da China e Chen et al. (2006) que realizaram a análise emergética da agricultura chinesa.

Entre as pesquisas envolvendo energia no Brasil, pode-se citar Comar (1998) que utilizou a análise emergética em projetos agrícolas e agroindustriais, Ortega et al. (2005) realizaram uma análise multi-variada para comparar os benefícios e os custos das principais modalidades de produção de soja no Brasil. Cavalett (2004) avaliou os aspectos energéticos e sócio-ambientais de dois importantes sistemas de aquicultura: a piscicultura integrada à criação de suínos no Estado de Santa Catarina e os pesque-pagues no Estado de São Paulo. E Agostinho (2005) utilizou a análise emergética em conjunto com o sistema de informações geográficas para estudar pequenas propriedades agrícolas.

A metodologia emergética estima os valores das energias naturais geralmente não contabilizadas, incorporadas em produtos, processos e serviços. Por meio de indicadores (índices emergéticos), esta abordagem desenvolve uma imagem dinâmica dos fluxos anuais dos recursos naturais e dos serviços ambientais providenciados pela natureza na geração de riqueza e o impacto das atividades antrópicas nos ecossistemas (COMAR, 1998).

Energia é definida como a energia de um tipo requerida em transformações para gerar um fluxo ou estoque. Nesta metodologia, a energia solar é utilizada. Energia solar de um fluxo ou estoque é a energia solar requerida para gerar este fluxo ou estoque. A sua unidade é o emjoule de energia solar, cuja abreviação é sej. (ODUM, 1996)

A energia avalia o trabalho previamente utilizado diretamente ou indiretamente para se produzir os produtos e serviços (ODUM, 1996).

Transformidade é definida como a quantidade de energia de um tipo requerida diretamente e indiretamente para gerar uma unidade de energia de outro tipo. Sua unidade é energia por unidade de energia (sej/J) (ODUM, 1988).

Os sistemas da natureza e a humanidade são partes de uma hierarquia de energia universal e estão imersos em uma rede de transformação de energia, que une os sistemas pequenos a grandes sistemas, e estes, a sistemas maiores ainda. Quanto maior a transformabilidade, mais acima no nível hierárquico de energia universal está o produto. Isto é baseado no princípio da máxima potência que diz que quanto maior a energia requerida para se produzir um produto ou serviço, maior é o valor da energia. Como o próprio nome diz, a transformabilidade pode ser usada para transformar um determinado tipo de energia em energia, multiplicando a energia de um determinado item pela sua correspondente transformabilidade (ODUM, 1996).

Geralmente, quanto maior é o trabalho da natureza na produção de recursos, menor é seu preço devido à sua abundância, ou seja, o preço em dinheiro não representa o valor do trabalho da natureza incorporado no recurso. Por outro lado, a energia expressa em em dólares (dólares do produto econômico bruto correspondente a uma dada contribuição de energia) consegue indicar a contribuição da natureza e da economia humana no recurso. Além disso, o uso do em dólar é recomendado porque é mais fácil para as pessoas pensarem em unidades monetárias do que em unidades de energia (ODUM, 2001).

Quando os recursos do ecossistema começam a ficar escassos e a demanda por esse recurso continua aumentando, o preço desse recurso tende a aumentar. Essa pressão de demanda coloca em risco a sustentabilidade do recurso. O trabalho da natureza deveria ser reconhecido e corretamente valorizado no mercado (ODUM, 2001).

3.3 Páginas Web Dinâmicas.

Nesta seção são apresentadas as principais tecnologias utilizadas para a construção de páginas web dinâmicas.

3.3.1 XML e XSLT

O XML (*Extensible Markup Language*) foi desenvolvido pela *XML Working Group* formado sob a supervisão do *World Wide Web Consortium* (W3C) em 1996. Os documentos XML são construídos através de unidades básicas chamadas “elementos” de acordo com uma série de restrições que são especificadas em um documento chamado *Document Type Definition* (DTD) (BRAY et al., 2006).

Os documentos XML são sempre heterogêneos e hierárquicos na estrutura e seguem uma ordem intrínseca. O modelo de dados de um documento XML pode ser visualizado como uma árvore (MELLO, 2006).

Diferentemente da HTML, que é uma linguagem que contém uma quantidade predefinida de tags descrevendo seus elementos, a XML permite a criação de tags próprias para compor os documentos (SILVA FILHO, 2004). Além disso, a XML possibilita a utilização de vários recursos importantes. A possibilidade de definir marcadores personalizados torna o documento “mais inteligente”, dando significado ao texto armazenado entre os marcadores (FURGERI, 2001).

Há muitas vantagens em se usar documentos XML. Esta linguagem é simples, regular e sua notação consistente torna fácil a construção de programas para processar dados XML. Muitas ferramentas têm sido desenvolvidas para ler e lidar com documentos

XML. O XML é extensível e qualquer um pode criar novas tags que podem ser compartilhadas com qualquer usuário. A XML também é auto-descritiva, e suas tags são compreensíveis tanto para os usuários quanto para os computadores. A XML foi construída para suportar aplicações internacionais incluindo textos de todos os alfabetos do mundo (MELLO, 2006).

Os arquivos XML têm se tornado o padrão de arquivos para troca de dados entre aplicativos web, promovendo inter-operabilidade e permitindo processamentos automáticos de recursos da web (SILVA FILHO, 2004; KAY, 2001). O XML satisfaz dois requisitos (KAY, 2001):

- **Separa os dados da apresentação:** A necessidade de separar a informação dos detalhes de como ele será apresentado.
- **Transmissão de dados entre aplicativos:** A necessidade de transmitir informação de uma organização para outra sem a necessidade de adaptar os softwares utilizados por cada empresa.

Com o objetivo de apresentar as informações de um arquivo XML para um usuário, deve ser criado um documento de apresentação como, por exemplo, um arquivo HTML ou PDF. Converter um arquivo XML para um HTML é provavelmente a mais comum aplicação para o XSLT (*Extensible Stylesheet Language Transformation*) (KAY, 2001).

O documento XSLT especifica a apresentação do documento XML descrevendo como ele deve ser transformado para um documento de tags que pode ser interpretado por um browser padrão, como os documentos da linguagem HTML (CLARK, 1999).

Há várias razões para se trabalhar com Java e XML (BEER et al., 2004):

1. Java e XML têm como objetivos comuns a simplicidade, a portabilidade e a flexibilidade;
2. Java é uma linguagem multiplataforma;
3. Java suporta pacotes para construção de aplicativos XML.

Ortega e Marchiori (2005) desenvolveram arquivos XML e XSLT para representar sistemas agrícolas da Flórida. Esses arquivos podem ser vistos em páginas da internet e são arquivos estáticos.

3.3.2 Java, banco de dados e aplicativos web.

Java é uma linguagem de programação muito conveniente para o desenvolvimento de software que funcione em conjunto com a Internet. Ela também é uma linguagem de programação orientada a objetos que utiliza uma metodologia que está se tornando cada vez mais útil no mundo do design de *software*. Além disso, ela é uma linguagem multiplataforma, o que significa que seus programas podem ser criados para executar do mesmo modo em diferentes sistemas operacionais (LEMAY; CADENHEAD, 1999).

Um banco de dados é uma coleção organizada de dados. Há muitas estratégias diferentes para organizar dados, para facilitar acesso e a manipulação. Um sistema de gerenciamento de banco de dados fornece mecanismos para armazenar, organizar, recuperar e modificar dados para muitos usuários. Os sistemas mais populares são os bancos de dados relacionais que são uma representação lógica de dados que permite o acesso aos dados sem considerar sua estrutura física. Um banco de dados relacional armazena dados em tabelas e a linguagem SQL (*Structured Query Language*) é a

linguagem padrão internacional utilizada quase universalmente com banco de dados relacionais para realizar consultas e manipular dados (DEITEL; DEITEL, 2005);

Uma importante característica da Java é sua Interface de Programação de Aplicações (API) para execução de instruções SQL, chamada Conectividade de Bancos de Dados em Java (JDBC). Esta API consiste de uma série de classes e interfaces escritas na linguagem de programação Java. Usando JDBC programadores podem escrever programas contendo instruções SQL que podem acessar bancos de dados relacionais, tais como Oracle, Sybase, Microsoft Access e Informix, MySQL. O acesso transparente a bancos de dados é uma característica muito importante para aplicações em tempo real, especialmente aplicações envolvendo acessos *on-line* a bases de dados (SALEH *et al.*, 1999).

Devido à facilidade de acesso da internet, os aplicativos web têm se tornado veículos populares para conduzir transações, serviços de compra e entrega e troca de informações (LIU, 2006). Java Servlets e JavaServer Pages (JSP) constituem a mais utilizada plataforma para desenvolvimento de aplicativos web. Aplicativos que são desenvolvidos utilizando-se estas tecnologias ou tecnologias semelhantes são tipicamente estruturados como fragmentos de coleções de programas (servlets ou páginas JSP) que recebem informações do usuário, produzem uma saída HTML ou XML, e interagem com banco de dados (KIRKEGAARD; MOLLER, 2006).

Kamiya (2005) desenvolveu um aplicativo web utilizando a tecnologia Java, especificamente applets, que realiza a avaliação emergética de sistemas agrícolas.

3.3.3 JavaServer Pages e Servlet

JavaServer Pages – JSP – é a tecnologia mais popular para desenvolvimento de aplicativos web com tecnologia Java (LIU, 2006). É uma tecnologia baseada em Java que simplifica o processo de desenvolvimento de sites dinâmicos. Com JSP, os designers da web e programadores podem rapidamente incorporar elementos dinâmicos em páginas da web usando Java embutido e algumas tags de marcação simples (FIELDS et al., 2002).

JavaServer Pages são arquivos de texto, normalmente com a extensão .jsp, que substituem as páginas HTML tradicionais. Os arquivos JSP contém código HTML tradicional embutido que permite que o designer de páginas acesse dados do código de Java rodando no servidor. Quando a página é solicitada por um usuário e processada pelo servidor de *HyperText Transport Protocol* (HTTP), a parte HTML da página é transmitida. No entanto, as partes de código da página são executadas no momento em que a solicitação é recebida e o conteúdo dinâmico gerado por este código é unido na página antes de ser enviado para o usuário (FIELDS et al., 2002).

Os programadores em JavaServer Page podem utilizar componentes e softwares especiais chamados JavaBeans e bibliotecas de tag personalizadas que encapsulam funcionalidades dinâmicas complexas. Um JavaBean é um componente reutilizável que segue determinadas convenções para design de classe especificadas pela Sun Microsystems (DEITEL; DEITEL, 2005).

A tecnologia JSP utiliza tags parecidas com as tags XML que encapsulam a lógica que gera o conteúdo da página. Todos os formatos de tags (HTML ou XML) são passados diretamente para a página de resposta (SUN MICROSYSTEMS, 2006).

Servlets são aplicações baseadas em Java para adicionar funcionalidade dinâmica a servidores da web (FIELDS et al., 2002). Os servlets têm a função de potencializar a funcionalidade de um servidor Web através da diminuição do overhead, da manutenção e do suporte.

Aplicativos web criados a partir de Java são geralmente uma coleção de servlets que rodam em um container servelet como Tomcat, JBoss ou WebSphere (DESMET et al., 2006)

Juntos, a tecnologia JSP e servlets fornecem uma atrativa alternativa para a criação de páginas web dinâmicas, oferecendo independência de plataforma, alta performance, separação da parte de lógica da parte de apresentação, fácil administração e a facilidade de uso (SUN MICROSYSTEMS, 2006).

3.3.4 Struts *Framework*

No desenvolvimento de um software, um *framework* é uma estrutura de suporte definida em que um outro projeto de software pode ser organizado e desenvolvido. Um *framework* pode incluir programas de suporte, bibliotecas de código, linguagens de script e outros softwares para ajudar a desenvolver e juntar diferentes componentes de um projeto de software.

O Struts é o framework livre mais difundido que combina o melhor de servlets e JSP e foi primeiramente desenvolvido em 2001 (DESMET et al. 2006; ZHANG et al. 2005).

Os aplicativos desenvolvidos com o Struts possuem as seguintes características:

- Uma estrutura de diretórios padrão.
- Arquivos de configuração padrão (web.xml).
- Funcionalidade dinâmica proporcionada pelas classes Java e páginas JSP.
- Um arquivo web padrão (.war) para desenvolvimento.

O Struts *framework* é baseado na arquitetura padrão MVC – *Model-View-Controller* (modelo-visualização-controle) – é um padrão que define a separação de maneira independente do *Model* (modelo) que são os objetos de Negócio, da *View* (Visualização) que compreende a interface com o usuário ou outro sistema e o *Controller* (Controle) que controla o fluxo da aplicação (SOUZA, 2002).

O struts torna fácil a criação de aplicativos flexíveis e a criação de tags JSP para a criação de páginas JSP. O modelo MCV do struts simplifica a construção de aplicativos web pela criação de um modelo onde podemos plugar componentes (ZHANG; WANG; CHEN, 2005).

Para aplicativos web, a arquitetura MVC pode ser descrita na Figura 1 em 5 passos a seguir (ZHANG; WANG; CHEN, 2005):

1. O usuário faz um requerimento http;
2. Os controladores (*controller*), na sua maioria servlets, recebem o requerimento e transferem as informações para os modelos;
3. Os modelos (*model*), que geralmente são Java Beans, recebem e processam os dados;

4. O controle (*controller*) recebe os resultados e disponibiliza para a Visualização (*View*). Os componentes de Visualização (*View*) geram páginas JSP com tags HTML embutidas;
5. A visualização (*View*) mostra os resultados no *browser* do usuário.

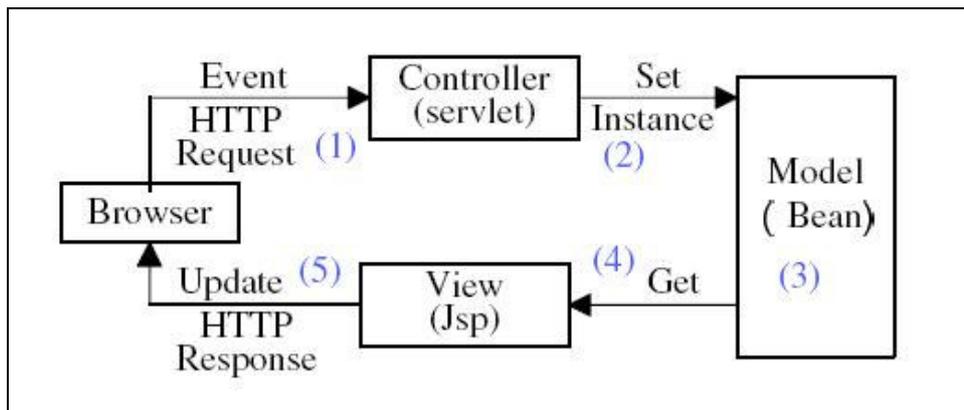


Figura 1 - Modelo MVC utilizado pelo struts framework.

4. MATERIAL E MÉTODOS

4.1 Análise emergética

A elaboração da análise emergética consiste nas seguintes etapas: construir o diagrama sistêmico para verificar e organizar todos os componentes e os relacionamentos existentes no sistema utilizando os símbolos de sistemas de energia; construir as tabelas emergéticas, com os fluxos quantitativos, baseados diretamente nos diagramas e, finalmente, calcular e os índices emergéticos que permitirão avaliar a situação econômica e ambiental do sistema.

4.1.1 1ª Etapa: Visão panorâmica pelos diagramas sistêmicos

Primeiramente deve-se construir um diagrama sistêmico panorâmico, identificando os componentes principais do sistema e os fluxos de entrada e saída. Com a representação dos componentes principais do sistema através dos símbolos, pode-se mostrar o caminho dos fluxos de massa, energia e serviços, incluindo as retro-alimentações. É necessário colocar no diagrama os limites do sistema para identificar todos os fluxos de entradas e saídas que cruzam as fronteiras do sistema escolhido. O diagrama inclui tanto a economia quanto o ambiente do sistema e mostra todas as interações relevantes.

A Figura 2 representa um exemplo de um diagrama de fluxos de energia de um sistema arbitrário.

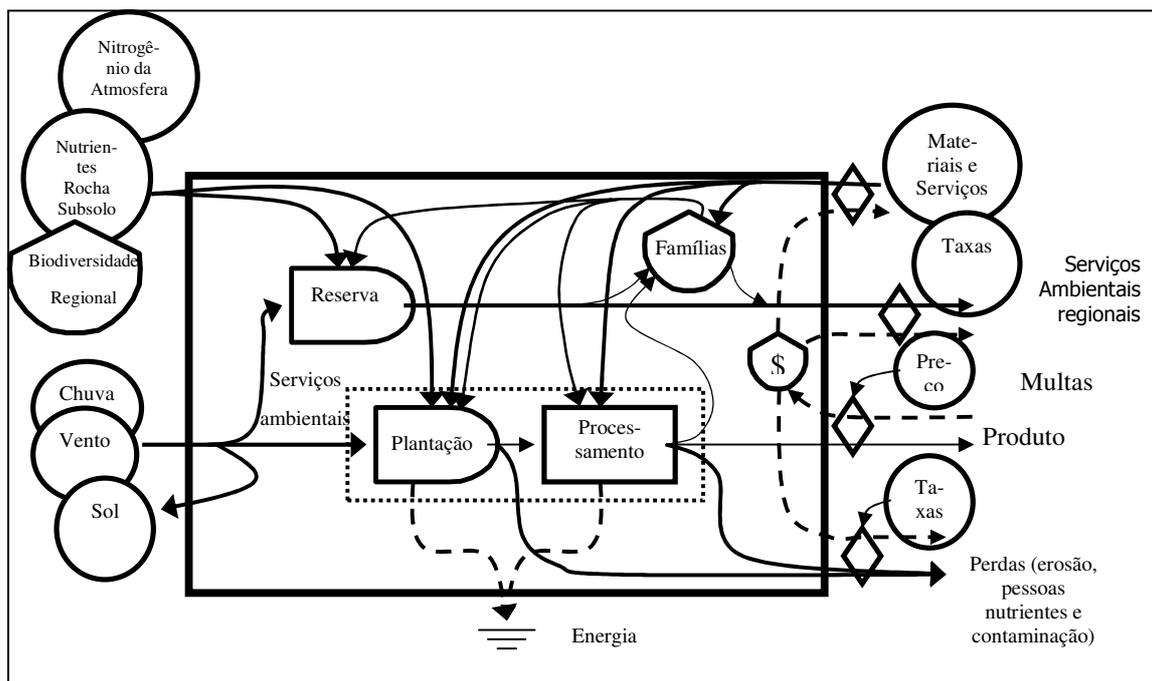


Figura 2 - Diagrama de fluxos de energia do sistema.

4.1.2 2ª Etapa – Tabelas de avaliação emergética

A segunda etapa da avaliação emergética consiste na construção de uma tabela de fluxos de energia, onde cada fluxo converte-se em uma linha de cálculo na tabela de avaliação de energia como mostrado na Figura 3.

Nota	Nome das contribuições	Valores	Unidade	Transformidade	Fluxo de energia
	R: Recursos da natureza renováveis				
	N: Recursos da natureza não-renováveis				
	M: Materiais da economia				
	S: Serviços da economia				
	TOTAL				

Figura 3 - Esquema de organização de uma tabela de cálculo de fluxos de energia.

A primeira coluna da tabela fornece a nota de pé-de-página onde se dão os detalhes do cálculo. A segunda coluna contém os nomes de todas as entradas do sistema. Na terceira coluna são apresentados os valores numéricos para cada uma das entradas e na quarta coluna, são apresentadas as unidades que são expressas por unidade de tempo (ano) e área (ha). A quinta coluna contém as transformidades (energia por unidade). Este valor consta na fonte de informação citada para essa linha na nota ao pé da página e são obtidos de estudos anteriores. Os fluxos de energia, calculados pela multiplicação dos fluxos de entrada pela transformidade correspondente, são apresentados na sexta coluna. Os valores obtidos correspondem ao fluxo de energia.

Existem também divisões na horizontal para facilitar a identificação dos tipos de recursos usados. Os primeiros fluxos colocados são os relacionados à contribuição da natureza (I), ou seja, $I = R + N$ onde os recursos naturais renováveis são representados pela letra (R) e os não-renováveis pela letra (N). Depois são colocados os recursos da economia (F), onde $F = M + S$. A letra (M) representa os materiais e a letras (S) os serviços. No final, o total de energia utilizado pelo sistema é representado pela letra (Y), onde $Y = I + F$.

Para o cálculo da energia dos produtos e do dinheiro recebido pelas vendas, é construída a tabela dos produtos do sistema mostrados na Figura 4. A Figura 5 apresenta um diagrama resumido com os fluxos agregados de energia descritos acima.

Produto	Unidade	Produção [Kg/ano]	Valor calórico do produto [Kcal/Kg]	Energia do produto [J/ha.ano]	Dinheiro recebido pelas vendas [R\$/ano]
TOTAL					

Figura 4 - Esquema da tabela para calcular a energia produzida e o dinheiro recebido pelo sistema.

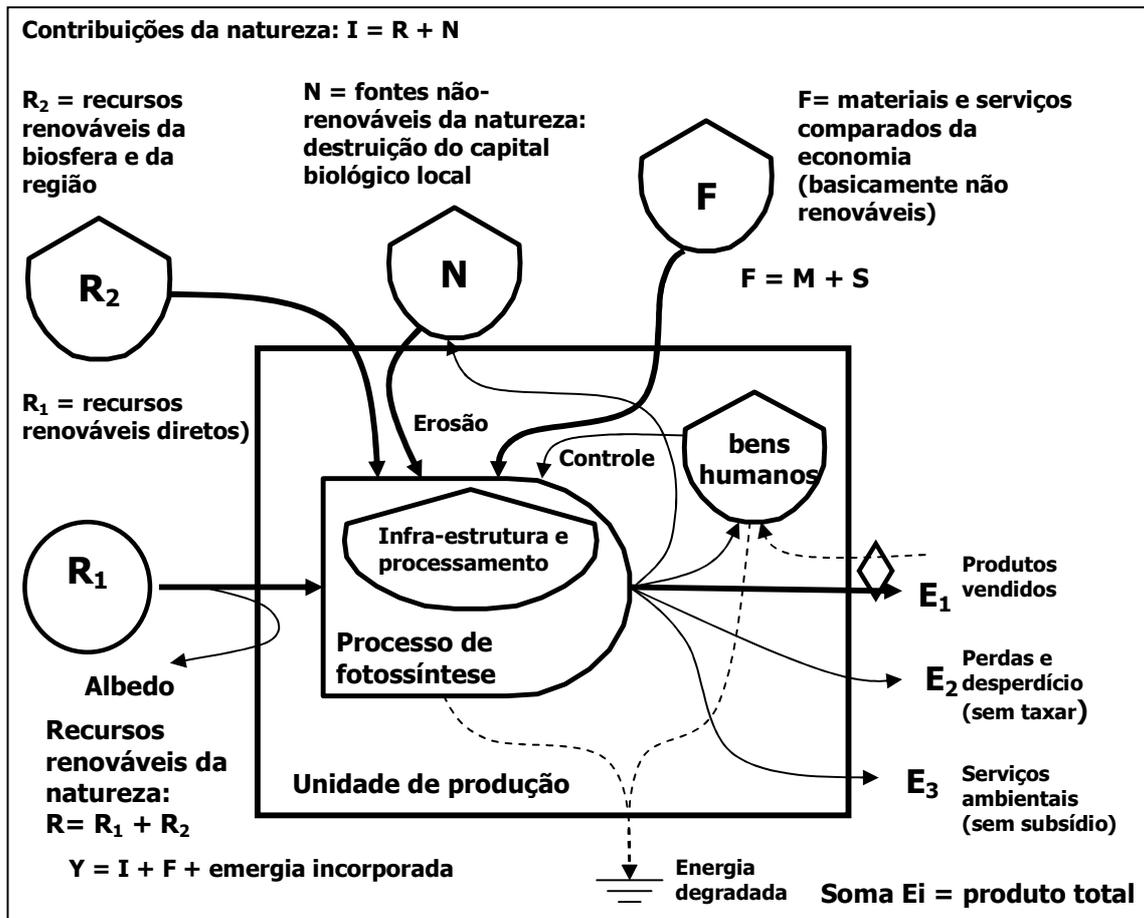


Figura 5 - Representação simplificada de um sistema produtivo.

4.1.3 3ª Etapa- Cálculo dos índices emergéticos

Os índices emergéticos são calculados com os resultados da tabela de avaliação de fluxos de energia e podem ser vistos a seguir:

1) Transformidade solar ($Tr=Y/E$): avalia a qualidade do fluxo de energia. Pode ser vista como um valor inverso da eficiência do agroecossistema. A transformidade solar do recurso gerado por um sistema é obtida dividindo a energia incorporada pelo sistema (Y) entre a energia do recurso (E). Sua unidade é expressa em energia por unidade de energia, massa, ou dinheiro, usualmente sej/J, sej/kg ou sej/US\$.

2) Renovabilidade emergética ($\%R=(R/Y) \times 100$): indica o grau de sustentabilidade do sistema. É calculado pela razão entre a energia dos recursos renováveis usados (R) e a energia total usada no sistema (Y).

3) Razão de investimento emergético ($EIR=F/I$): mede a proporção de energia comprada (F) em relação às entradas de energia do meio-ambiente (I). É um bom indicador da intensidade de uso de recursos econômicos na agricultura. O cálculo desta razão permite a escolha do modelo de agricultura compatível com o sistema econômico e ambiental analisado. Nesse sentido, a razão (F/I) se constitui em um bom indicador para auxiliar a elaboração de uma política agrícola sustentável.

4) Razão de rendimento emergético ($EYR=Y/F$): avalia o desempenho, ou eficiência, de uma unidade de produção ou processo. É obtida através da divisão da energia do produto (Y) pela energia das entradas que provém da economia (F). É uma medida de sua contribuição líquida, além do que o sistema de produção consome para sua própria operação (ODUM, 1996). Para que os sistemas de produção possam contribuir para a economia global, esta relação deveria ser maior do que 1, caso contrário o processo consome mais do que produz.

5) Razão de intercâmbio de energia (EER): é a proporção de energia do produto (Y) em relação ao valor da energia pela venda do produto através de uma transação comercial. Este índice é encontrado através da multiplicação da relação energia/dinheiro pelo dinheiro recebido pela venda, ou seja, $EER= Y/([produção \times preço \times (energia/dólar)])$. Esse índice avalia se o produtor está recebendo na venda dos produtos, toda energia necessária para a produção.

4.1.4 Emdólar

Emdólar é a relação [energia/dinheiro]. Os emdólares indicam o dinheiro circulante cujo poder de compra está estabelecido pelo uso de uma quantidade de energia. Os emdólares são equivalentes de energia. A relação (energia/dólar) é obtida dividindo-se a energia total do país em determinado ano pelo produto nacional bruto (PNB) expresso em dólares. (CAVALETT, 2004)

A energia expressa em emdólar consegue indicar a verdadeira contribuição da natureza e da economia humana no recurso. Quando os recursos do ecossistema passam a ser escassos, o preço aumenta e a pressão da demanda continua colocando em risco a sustentabilidade do recurso. As políticas públicas, independentemente do tamanho do sistema e do local, podem ter êxito, aumentando ao máximo os emdólares ou o fluxo de energia. Em outras palavras, isso significa que o trabalho da natureza deve ser reconhecido e corretamente valorizado no mercado (ODUM, 2001).

O emdólar para o Brasil pode ser calculado pela equação (1) (AGOSTINHO, 2005).

$$\frac{\text{energia}}{\text{dólar}} = \left(\frac{248 + 878,16 * e^{\frac{-(ano-1981)}{9,49}}}{100} \right) [\text{sej/US\$}] * 10^{12} \quad (1)$$

4.2 Construção dos arquivos XML.

A construção de documentos XML devem seguir as seguintes regras (FARIA, 2005):

- Um documento XML é formado por um ou mais elementos;
- Elementos são formados por duas tags, uma indicando o início, e outra, o fim;
- As tags que indicam o início e o fim possuem o mesmo nome, porém, a tag de fim é representada pelo caractere /. Toda tag é identificada pelo caracter <, seguido de seu nome e, logo após, o caractere >. Caso seja uma tag de finalização, o caractere / será colocado após o caractere < ;
- Tags podem ter atributos, colocados dentro da tag de inicio. O valor dos atributos deve ser especificado entre aspas simples ou entre aspas duplas, conforme o exemplo: <peessoa tipo="funcionário"> Maria Clara de Paula</peessoa>, em que tipo é o nome do atributo e funcionário, seu valor;
- O XML é case-sensitive, ou seja, diferencia entre caracteres maiúsculos e minúsculos;
- Tags devem ser abertas e fechadas na ordem correta.

A Figura 6 mostra o exemplo de um documento XML

```
<peessoa>
<nome> Maria Clara de Paula</nome>
<cargo>Diplomata</cargo>
<departamento>Direitos Humanos
  <sala>402</sala>
  <ramal>206</ramal>
</departamento>
</peessoa>
```

Figura 6 - Esquema de um documento XML.

Vamos dividir a construção dos arquivos XML em duas etapas. Na primeira foram construídos arquivos XML para os sistemas da agricultura da Flórida e na segunda etapa foram construídos arquivos XML para os sistemas da agricultura do Brasil.

4.2.1 Arquivos XML para representar o sistema da Flórida.

Para representar os sistemas agrícolas da Flórida foram construídos arquivos XML com dados do trabalho de Brandt-Williams (2002).

Foram criados 22 arquivos XML e cada um representa um produto da agricultura da Flórida. Na Tabela 1 podemos verificar os produtos estudados.

Tabela 1 - Produtos da agricultura da Flórida estudados

Item	Sistemas estudados
1	<i>Bell pepper</i>
2	<i>Cabbage</i>
3	<i>Corn (sweet)</i>
4	<i>Cucumber</i>
5	<i>Eggs</i>
6	<i>Green beans</i>
7	<i>Lettuce</i>
8	<i>Orange</i>
9	<i>Peanuts</i>
10	<i>Pecans</i>
11	<i>Potatoes</i>
12	<i>Tomatoes</i>
13	<i>Watermelon</i>
14	<i>Alligator</i>
15	<i>Beef</i>
16	<i>Corn (grain)</i>
17	<i>Milk</i>
18	<i>Oats</i>
19	<i>Soybeans</i>
20	<i>Sugarcane</i>
21	<i>Cotton</i>
22	<i>Bahia grass</i>

Todos os arquivos XML são padronizados, uma vez que todos os sistemas utilizam os mesmos recursos. A Tabela 2 mostra os recursos utilizados no sistema da agricultura da Flórida. Os arquivos XML foram construídos seguindo o mesmo esquema. A construção

do arquivo referente ao produto *bell pepper* será apresentado em detalhes, sendo que a mesma metodologia foi utilizada para construção de todos os demais arquivos XML. Para uma melhor compreensão vamos dividir a construção do arquivo em 8 partes: inicialização, definição do sistema, recursos utilizados, cálculo de variáveis, dados do produto, resultados, índices e gráficos.

Tabela 2 - Recursos utilizados na elaboração dos arquivos XML da Flórida.

Tipo do recurso	Recurso
Renovável	<i>Sun</i> (sol)
Renovável	<i>Rain</i> (chuva)
Renovável	<i>Evapotranspiration</i> (evapotranspiração)
Não Renovável	<i>Net Topsoil Loss</i> (perda de solo)
Material	<i>Fuel</i> (combustível)
Material	<i>Electricity</i> (eletricidade)
Material	<i>Machinery</i> (maquinário)
Material	<i>Lime</i> (calcário)
Material	<i>Pesticides</i> (pesticidas)
Material	<i>Potash</i> (potássio)
Material	<i>Phosphate</i> (fosfato)
Material	<i>Nitrogen</i> (nitrogênio)
Material	<i>Feed, grain</i> (alimentação, grãos)
Material	<i>Feed, Livestock</i> (alimentação, animal)
Serviço	<i>Human Labor</i> (trabalho humano)
Serviço	<i>Services From Economy</i> (serviços da economia)

A inicialização do documento XML é feita nas duas primeiras linhas do arquivo. A primeira *tag* do arquivo indica a versão do XML através do atributo *version* e é definido o atributo *encoding* que define qual será a seqüência de caracteres utilizados no documento. No nosso caso utilizamos a versão 1.0 e o valor ISO8859-1 é utilizado por que representa o idioma português. Na segunda linha, temos a definição de qual arquivo XSLT irá processar o arquivo XML para que o mesmo possa ser visto em um browser. Neste caso o nome do arquivo é *table.xslt*. A terceira linha o elemento TABLE é aberto. Este elemento irá

englobar todos os demais elementos do nosso arquivo XML. A inicialização do arquivo XML pode ser vista na Figura 7.

```
<?xml version="1.0" encoding="ISO8859-1"?>
<?xml-stylesheet href="table.xslt" type="text/xsl"?>
<TABLE>
```

Figura 7 - Inicialização do arquivo XML.

A segunda etapa é a definição do sistema representado pelo arquivo XML. Na Figura 8 podemos visualizar como este procedimento é realizado.

Primeiramente é definido a *tag* SYSTEM onde armazenamos o título, o autor, a instituição e o objeto de estudo do trabalho. Na *tag* título é definido o atributo foto com o endereço de foto que será carregada quando a página for visualizada. Nesta parte também é armazenada a informação do Emdolar.

```
<SYSTEM>
  <TITLE foto="bell_peppers.jpg">Folio # 4. Agricultural Systems of Florida</TITLE>
  <AUTHORS>Sherry L. Brandt-Williams</AUTHORS>
  <INSTITUTION>University of Florida, 2001 (revised 2004)</INSTITUTION>
  <OBJECTIVE>Bell Pepper (USA, 1981)</OBJECTIVE>
  <EMDOLLAR>270000000000</EMDOLLAR>
</SYSTEM>
```

Figura 8 - Definição do sistema no arquivo XML.

Na terceira etapa, são armazenados os dados dos recursos utilizados no sistema: Recursos Renováveis, Recursos Não-Renováveis, Materiais e Serviços.

A declaração dos recursos renováveis é feita como mostrado na Figura 9.

```
<RENEWABLE>
  <CODE>R1</CODE>
  <ITEM>Sun</ITEM>
  <QUANTITY>6350000000</QUANTITY>
  <UNITS>J/m2/year</UNITS>
  <FACTOR>10000</FACTOR>
  <TRANSFORMITY>1</TRANSFORMITY>
  <EMERGY>6.350000000</EMERGY>
</RENEWABLE>
```

Figura 9 - Representação dos recursos renováveis no arquivo XML.

É criada uma *tag* RENEWABLE que contém as seguintes *tags*: CODE, ITEM, QUANTITY, UNITS, FACTOR, TRANSFORMITY e EMERGY. A *tag* CODE representa o índice do item utilizado. Neste caso, por se tratar do primeiro item, a *tag* CODE tem o valor R1. O próximo item esta terá o valor R2, seguindo esta lógica até o último recurso renovável. Na *tag* ITEM é informado o nome do recurso utilizado, na *tag* QUANTITY é informado a quantidade desse recurso e a unidade é colocada na *tag* UNITS. Na *tag* FACTOR é colocado um fator de correção de unidades, na *tag* TRANSFORMITY é colocado o valor da transformidade para aquele item e na *tag* EMERGY o valor do fluxo emergético calculado.

Para os Recursos Não-Renováveis, Materiais e Serviços, a mesma lógica é utilizada, como podemos verificar na Figura 10.

Esta etapa é na verdade a segunda etapa da metodologia emergética, ou seja, o preenchimento das tabelas de fluxo de energia, com as contribuições da natureza (renovável e não renovável), material e serviços, conforme apresentado na Figura 4.

<pre> . . . <NONRENEWABLE> <CODE>N1</CODE> <ITEM>Net Topsoil Loss</ITEM> <QUANTITY>8504.98</QUANTITY> <UNITS>kg/ha/year</UNITS> <FACTOR>904176</FACTOR> <TRANSFORMITY>123984</TRANSFORMITY> <EMERGY>95.3436810783</EMERGY> </NONRENEWABLE> . . . <MATERIAL> <CODE>M1</CODE> <ITEM>Fuel</ITEM> <QUANTITY>1597.16</QUANTITY> <UNITS>litres/ha/year</UNITS> <FACTOR>34874504.62</FACTOR> <TRANSFORMITY>110880</TRANSFORMITY> <EMERGY>617.6034162020</EMERGY> </MATERIAL> </pre>	<pre> . . . <SERVICE> <CODE>S1</CODE> <ITEM>Human Labor</ITEM> <QUANTITY>16.32</QUANTITY> <UNITS>hours/ha/year</UNITS> <FACTOR>100464000</FACTOR> <TRANSFORMITY>4450000</TRANSFORMITY> <EMERGY>729.6097536000</EMERGY> </SERVICE> . . . </pre>
--	--

Figura 10 - Representação dos recursos não renováveis, materiais e serviços no arquivo XML.

A Figura 11 mostra a criação das *tags* AGREGATED que mostra como são calculados as variáveis R, N, M, S, F e Y na metodologia emergética. Nesta etapa são criadas as *tags* NAME e FORMULA para representar o nome da variável e a fórmula utilizada para o cálculo destas variáveis. Esta parte do arquivo XML é uma parte estática, ou seja, ela não é modificada e é sempre a mesma para todos os arquivos XML.

A próxima etapa da criação do arquivo XML se refere aos dados do produto estudado. Como podemos verificar na Figura 12 os dados são: massa (kg/ha/ano), umidade do produto, porcentagem de carboidratos, lipídios, proteínas e o preço do produto e são representados na tag PROPERTY.

```
<AGGREGATED tipo="R">
  <NAME>Renewable resources from Nature</NAME>
  <FORMULA>R=Sum(Renewables)</FORMULA>
  <VALUE />
</AGGREGATED>
<AGGREGATED tipo="N">
  <NAME>Non renewable resources from Nature</NAME>
  <FORMULA>N=Sum(Non renewables)</FORMULA>
  <VALUE />
</AGGREGATED>
<AGGREGATED tipo="I">
  <NAME />
  <FORMULA>I = R + N</FORMULA>
  <VALUE />
</AGGREGATED>
<AGGREGATED tipo="M">
  <NAME>Materials from Economy</NAME>
  <FORMULA>M=Sum(Materials)</FORMULA>
  <VALUE />
</AGGREGATED>
<AGGREGATED tipo="S">
  <NAME>Services from Economy</NAME>
  <FORMULA>S=Sum(Services)</FORMULA>
  <VALUE />
</AGGREGATED>
<AGGREGATED tipo="F">
  <NAME />
  <FORMULA>F = M + S</FORMULA>
  <VALUE />
</AGGREGATED>
<AGGREGATED>
  <NAME>Emergy used</NAME>
  <FORMULA>Y= I + F</FORMULA>
  <VALUE />
</AGGREGATED>
```

Figura 11 - Representação das variáveis as variáveis R, N, M, S, F e Y da metodologia emergética nos arquivos XML.

<pre> <PRODUCT line="06"> <PROPERTY>Mass (wet)</PROPERTY> <VALUE>28000</VALUE> <UNITS>kg/ha/year</UNITS> <AD1 /> <AD2 /> </PRODUCT> <PRODUCT line="06"> <PROPERTY>Moisture</PROPERTY> <VALUE>93.5</VALUE> <UNITS>%</UNITS> <AD1 /> <AD2 /> </PRODUCT> <PRODUCT line="06"> <PROPERTY>Carbohydrates</PROPERTY> <VALUE>63</VALUE> <UNITS>%</UNITS> <AD1>17000000</AD1> <AD2>J/kg</AD2> </PRODUCT> </pre>	<pre> <PRODUCT line="06"> <PROPERTY>Lipids</PROPERTY> <VALUE>11</VALUE> <UNITS>%</UNITS> <AD1>39000000</AD1> <AD2>J/kg</AD2> </PRODUCT> <PRODUCT line="06"> <PROPERTY>Protein</PROPERTY> <VALUE>26</VALUE> <UNITS>%</UNITS> <AD1>24000000</AD1> <AD2>J/kg</AD2> </PRODUCT> <PRODUCT line="06"> <PROPERTY>Price (wet product)</PROPERTY> <VALUE>0.5</VALUE> <UNITS>US\$/kg</UNITS> <AD1 /> <AD2 /> </PRODUCT> </pre>
---	---

Figura 12 - Tags do arquivo XML que se referem a dados do produto estudado.

A tag VALUE representa o valor da quantidade de cada atributo do produto e a tag UNITS representa a unidade utilizada. As tags AD1 e AD2 representam o fator de conversão, ou seja quantos Joules (J) temos em 1 quilo (kg) de carboidratos, lipídios e proteínas. Esta etapa ainda representa a segunda etapa da metodologia emergética, onde é calculado a energia do produto e a quantidade de dinheiro que é recebido pelas vendas conforme apresentado na Figura 4.

As próximas tag representam alguns resultados como pode ser visto na Figura 13. Na tag RESULT line="01" temos o total de massa seca (*Total dry mass*). O valor da tag VALUE é calculado pela equação (2).

$$TDM = M * \frac{(100 - Mo)}{100} \quad (2)$$

onde TDM é o Total de massa seca (*Total dry mass*), M é a massa e Mo é a umidade (*moisture*).

```

<RESULT line="01">
  <PROPERTY>Total dry mass</PROPERTY>
  <VALUE>1820</VALUE>
  <UNITS>kg/ha/year</UNITS>
</RESULT>
<RESULT line="02">
  <PROPERTY>Energy of product</PROPERTY>
  <VALUE>3,865</VALUE>
  <UNITS>E10 J/ha/year</UNITS>
</RESULT>
<RESULT line="03">
  <PROPERTY>Sale</PROPERTY>
  <VALUE>14000</VALUE>
  <UNITS>US$/ha/year</UNITS>
</RESULT>
<RESULT line="04">
  <PROPERTY>Emergy of Sale</PROPERTY>
  <VALUE>4.8</VALUE>
  <UNITS>E13 sej/ha/yr</UNITS>
</RESULT>
<RESULT line="05">
  <PROPERTY>Real value of product</PROPERTY>
  <VALUE />
  <UNITS>em-US$/ha/yr</UNITS>
</RESULT>

```

Figura 13 - Tags que representam os resultados

A tag RESULT line="02" representa a energia do produto (*Energy of product*) e o valor da tag VALUE é calculado pela equação (3).

$$E = TDM * \frac{(\%C * 17000000 + \%L * 39000000 + \%P * 24000000)}{100} \quad (3)$$

Onde E é a energia do produto, TDM é o total de massa seca, %C é a porcentagem de carboidrato, %L é a porcentagem de lipídios e %P é a porcentagem de proteína do produto.

A tag RESULT line="3" representa o valor obtido pela venda do produto (*Sale*) e o valor da tag VALUE é calculado pela equação (4).

$$Sale = M * Pr \quad (4)$$

onde M é a massa e Pr é o preço.

A tag RESULT line="4" representa o valor da energia das vendas (*Emergy of Sales*) e é calculado pela equação (5).

$$ES = Sale * EmDollar \quad (5)$$

A tag RESULT line="5" representa o Valor Real do Produto (*Real Value of Product*) e é calculado pela equação (6).

$$RV = \frac{Y}{ES} \quad (6)$$

onde RV é o Valor Real do produto, Y é a emergia total e ES é a Emergia das Vendas.

As próximas tags do arquivo XML representam os cálculos dos índices emergéticos. A Figura 14 mostra essas tags.

<pre> <INDICE tipo="TR"> <NAME>Transformity (sej/J)</NAME> <FORMULA>Tr=Y/E=Emergy/Energy</FORMULA> <VALUE /> </INDICE> <INDICE tipo="TRM"> <NAME>Transformity (sej/kg)</NAME> <FORMULA>Tr=Y/M=Emergy/Dry Mass</FORMULA> <VALUE /> </INDICE> <INDICE tipo="REN"> <NAME>Renewability</NAME> <FORMULA>Ren=(100)*(R/Y)</FORMULA> <VALUE /> </INDICE> </pre>	<pre> <INDICE tipo="EYR"> <NAME>Emergy Yield Ratio</NAME> <FORMULA>EYR=Y/F</FORMULA> <VALUE /> </INDICE> <INDICE tipo="EIR"> <NAME>Emergy Investment Ratio</NAME> <FORMULA>EIR=F/I</FORMULA> <VALUE /> </INDICE> <INDICE tipo="EER"> <NAME>Emergy Exchange Ratio</NAME> <FORMULA>EER=Y/EmS</FORMULA> <VALUE /> </INDICE> </pre>
---	---

Figura 14 - Tags com as informações dos índices emergéticos.

Cada *tag* INDICE possui um tipo que representa a abreviação do nome do índice emergético. A *tag* NAME indica o nome do índice e na *tag* FORMULA é informada a fórmula utilizada para se calcular o índice emergético. Estas *tag* são padrões em todos os arquivos XML e não podem ser modificadas.

As últimas *tags* do arquivo XML podem ser vistas na Figura 15.

Estas *tags* são escritas para referenciar arquivos gráficos que serão utilizados na construção de gráficos de barra que irão representar os resultados. Estas *tags* também são estáticas, e não podem ser alteradas. Para finalizar o nosso arquivo, fechamos a *tag* TABLE que foi aberta na inicialização.

Todos os cálculos, fatores de correção e transformidades utilizadas no sistema da Flórida podem ser vistos no Apêndice A.

```
<GRAPHIC>
  <CONCEPT picture="green-bar.jpg">Renewable resources</CONCEPT>
  <VALUE />
</GRAPHIC>
<GRAPHIC>
  <CONCEPT picture="brown-bar.jpg">Nature Non-Renewables</CONCEPT>
  <VALUE />
</GRAPHIC>
<GRAPHIC>
  <CONCEPT picture="red-bar.jpg">Economy Non-Renewables</CONCEPT>
  <VALUE />
</GRAPHIC>
<GRAPHIC>
  <CONCEPT picture="blue-bar.jpg">ESI = EYR/EIR</CONCEPT>
  <VALUE />
</GRAPHIC>
</TABLE>
```

Figura 15 - Tag utilizadas para referenciar os arquivos gráficos.

4.2.2 Arquivos XML para representar os sistemas agrícolas do Brasil.

Na construção dos arquivos XML que representam o sistema de agricultura do Brasil, foram utilizados os dados do AGRIANUAL 2004 (FNP Consutoria e Agroinformativos, 2004).

A metodologia utilizada para a construção desses arquivos é semelhante a metodologia utilizada na criação dos arquivos do sistema da Flórida. As mesmas *tags* são utilizadas, mas por se tratar de sistemas bem mais complexos, o número de recursos utilizados é bem maior. Por este motivo, os documentos não são padronizados, ou seja, para cada sistema há uma quantidade e tipos de recursos diferentes.

4.3 Construção do arquivo XSLT

A construção do documento XSLT se faz necessária para que os documentos XML possam ser visualizados em *browsers*. Para a visualização dos documentos que são processados pelo aplicativo desenvolvido foi necessária a criação de um único documento XSLT. Este documento cria tabelas que são apresentadas como resultado em uma página HTML. São criadas tabelas com os recursos utilizados pelo sistema, dados do sistema, fluxo emergético para cada tipo de recursos, os resultados e índices emergéticos.

A Tabela 3 mostra os recursos utilizados para o produto *Bell Pepper*. Para a construção desta tabela o arquivo XSLT copia dos dados dos *tags* CODE, ITEM, QUANTITY, UNIT, FACTOR e TRANSFORMITY do arquivo XML e cola nas células correspondentes. Para preencher a última célula, referente ao fluxo emergético, o arquivo XSLT processa o cálculo do dado multiplicando o valor de QUANTITY, FACTOR e TRANSFORMITY.

Tabela 3 - Tabela construída pelo documento XSLT para mostrar os recursos utilizados.

Code	Item	Quantity	Units	Conversion Factor	Transformities of inputs (sej/unit)	Emergy flow E13
R1	Sun	6350000000	J/m2/year	10000	1	6.4
R2	Rain	1.3716	M3/m2/year	45940000000	30240	190.5
R3	Evapotranspiration	5430000	J/m2/year	10000	25910.64	140.7
N1	Net Topsoil Loss	8504.98	Kg/ha/year	904176	123984	95.3
M1	Fuel	1597.16	litres/há/year	34874504.62	110880	617.6
M2	Electricity	208.06	KWH/ha/year	3600000	268800	20.1
M3	Machinery	0	Kg/ha/year	1	1120000000000	.0
M4	Lime	0.00	Kg/ha/year	1	1680000000000	.0
M5	Pesticides	131	Kg/ha/year	1	2486400000000	325.7
M6	Potash	172	Kg K/ha/year	1	2923200000000	50.3
M7	Phosphate	52.7	Kg P/ha/year	1	3696000000000	194.8
M8	Nitrogen	44	Kg N/ha/year	1	4048800000000	178.1
M9	-	0	-	0	0	.0
M10	-	0	-	0	0	.0
S1	Human Labor	16.32	hours/ha/year	100464000	4450000	729.6
S2	Services from economy	2110	US\$/há/year	1	4536000000000	957.1

A Tabela 4 construída pelo arquivo XSLT se refere aos dados do produto. Simplesmente são copiados os valores dos elementos PRODUCT do arquivo XML

Tabela 4 - Tabela construída pelo arquivo XSLT para mostrar os dados do produto.

Output data	Caloric value	
Mass (wet)	28000 kg/há/year	
Moisture	93.5 %	
Carbohydrates	63 %	17000000 J/kg
Lipids	11 %	39000000 J/kg
Protein	26 %	24000000 J/kg
Price (wet product)	0.5 US\$/kg	

A próxima tabela criada pelo arquivo XSLT mostra as contribuições R, N, M e S, os fluxos agregados de energia I e F e a energia total utilizada pelo sistema. Podemos visualizar os dados na Tabela 5 abaixo.

Tabela 5 - Tabela criada pelo documento XSLT para representar o fluxo energético agregado.

Inputs classification	Equation	Aggregated flows x E13 sej/ha/yr
Renewable resources from Nature	$R = \text{Max}(\text{Renewables})$	140.69
Non renewable resources from Nature	$N = \text{Sum}(\text{Non renewables})$	95.34
	$I = R + N$	236.04
Materials from Economy	$M = \text{Sum}(\text{Materials})$	1386.66
Services from Economy	$S = \text{Sum}(\text{Services})$	1686.71
	$F = M + S$	3073.37
Emergy used	$Y = I + F$	3309.41

Para construir a Tabela 5, o arquivo XSLT copia os valores dos elementos AGGREGATED do arquivo XML e cada valor é processado pelo arquivo XSLT de acordo com as equações apresentadas na tabela. O arquivo XSLT cria variáveis que recebem os

valores das equações mostradas na tabela. O valor destas variáveis é colocado na coluna *Aggregated flows*.

A próxima tabela criada pelo documento XSLT, Tabela 6, mostra o resultado referente ao produto. Os valores dos elementos PROPERTIES dos elementos RESULT são copiados e os valores são processados pelo arquivo XSLT de acordo com as equações 1, 2, 3, 4, 5.

Tabela 6 - Tabela criada pelo arquivo XSLT para mostrar o resultado referente ao produto.

Output results Bell Pepper (USA, 1981)	Value	Units
Total dry mass	1820	kg/há/year
Energy of product	386.57	E10 J/há/year
Sale	14000.00	US\$/há/year
Emergy of Sale	37.80	E13 sej/ha/yr
Real value of product	12257.06	em-US\$/ha/yr

A última tabela criada pelo arquivo XSLT, Tabela 7, mostra os índices emergéticos calculados. O arquivo XSLT copia os valores dos elementos NAME e FORMULA contidos em INDICE e transfere para as colunas *Emergy Indices* e *Equation* respectivamente. São criadas variáveis para cada um dos índices e os valores processados pelo arquivo XSLT são transferidos para a tabela na coluna *Value*.

Tabela 7 - Tabela criada pelo arquivo XSLT para mostrar os índices emergéticos calculados.

Emergy Índices	Equation	Value
Transformity (sej/J)	$Tr=Y/E=Emergy/Energy$	8560.99
Transformity (sej/kg)	$Tr=Y/M=Emergy/Dry\ Mass$	18.18 E12
Renewability	$Ren=(100)*(R/Y)$	4.25%
Emergy Yield Ratio	$EYR=Y/F$	1.08
Emergy Investment Ratio	$EIR=F/I$	13.02
Emergy Exchange Ratio	$EER=Y/sem$	0.88

Além de criar todas as tabelas, o arquivo XSLT ainda processa um título onde são mostrados os elementos da *tag* SYSTEM. Um gráfico também é criado para representar o resultado. O gráfico criado pode ser visto na Figura 16.

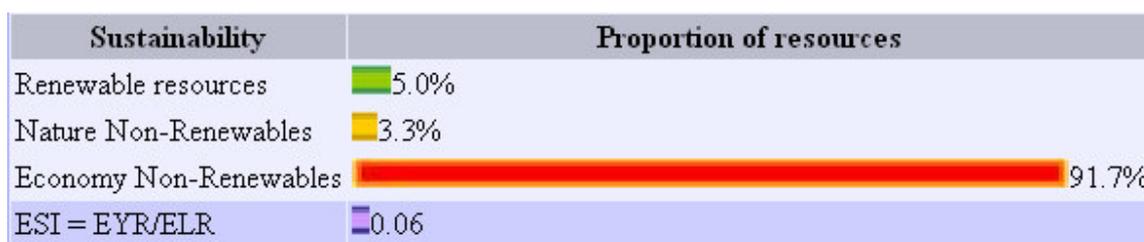


Figura 16 - Gráfico criado pelo arquivo XSLT para visualizar o resultado.

O código completo do arquivo XSLT pode ser visto no Apêndice B.

4.4 Desenvolvimento do Aplicativo

A análise energética dos sistemas pode ser feita facilmente modificando os campos quantidades dos recursos utilizados e os dados do produto nos arquivos XML. Com a alteração desses dados, o arquivo XSLT processa novos resultados.

Para desenvolver o aplicativo foi utilizado a linguagem de programação Java e o editor *Borland JBuilder 2005 Enterprise*.

A manipulação de arquivos XML pode ser feita de várias formas, e neste trabalho foi utilizado a API *BorlandXML*. Para utilizar esta API é necessário a criação de um documento de especificação do arquivo XML chamado *Document Type Definition* (DTD). Este documento .dtd é criado automaticamente pelo editor *JBuilder* quando fornecemos o documento XML.

O arquivo .dtd especifica todos os elementos do arquivo XML e como todos os documentos XML possuem o mesmos elementos, um único arquivo DTD é necessário. O arquivo DTD pode ser visto no Apêndice C.

A API BorlandXML lê os dados do arquivo .dtd e cria classes java para alterar todos os elementos do arquivo XML.

O aplicativo java foi desenvolvido utilizando-se essas classes. Foi criado uma interface gráfica que pode ser acessada através do arquivo emergytable.jar e os usuários podem efetuar as seguintes operações:

- Abrir o arquivo XML.
- Alterar os campos de quantidades dos recursos utilizados no sistema.
- Alterar os campos dos dados do produto.
- Gravar as alterações no arquivo XML.
- Visualizar o resultado.

4.5 Desenvolvimento de páginas web dinâmicas

Uma alternativa para a visualização dos arquivos XML na internet foi a criação de páginas web dinâmicas com o objetivo de tornar a análise emergética mais acessível.

Foram utilizadas as tecnologias JSP, Struts, e banco de dados MySQL e o editor *Borland JBuilder 2005 Enterprise* também foi utilizado nesta etapa do trabalho.

O *struts framework* torna mais fácil a criação de páginas web dinâmicas e o *JBuilder* possui algumas ferramentas que facilitam a utilização do *Struts framework*.

4.5.1 Construção do Módulo Web.

Primeiramente foi construído um Módulo Web (*Web Module*) que suportasse as tecnologias JSP 2.0, Servlet 2.4 e Struts 1.1. Esta etapa foi realizada através da ferramenta *Web Module Wizard* do *Borland JBuilder 2005*. Este módulo, que recebeu o nome de *emergytable*, tem a função de armazenar todos os arquivos com extensão *.jsp* que são criados. Dentro deste módulo é criada uma pasta chamada *WEB-INF* onde são armazenados os arquivos *struts-config.xml*, *web.xml* e arquivos de bibliotecas de *tags* utilizadas pelo *struts*. O arquivo *struts-config.xml* e *web.xml* são gerado automaticamente pelo *JBuilder*. O primeiro é responsável por controlar toda ação do *struts*, e o segundo contém as informações de configuração das páginas web geradas.

Ainda dentro do diretório *WEB-INF* encontramos a pasta *lib*, onde são armazenadas todas as bibliotecas de classes utilizadas, e a pasta *classes*, onde são armazenadas todas as classes java criadas.

4.5.2 Construção do Banco de Dados

A camada de dados foi desenvolvida utilizando-se o MySQL como sistema de banco de dados. O MySQL é um sistema de gerenciamento de dados SQL livre. Para fazer a conexão das classes java com o banco de dados é necessário um driver JDBC. O driver utilizado nesta aplicação foi o MySQL Connector/J que pode ser encontrado no site www.mysql.com/products/connector/j.

Foram criadas as seguintes tabelas no banco de dados:

- Cadastro
- Login
- Recursos
- DadosBr

A Tabela 8 mostra a tabela cadastro que possui 5 colunas. Essa tabela é preenchida quando o usuário se cadastra no sistema.

Tabela 8 - Esquema para a tabela cadastro.

Coluna	Tipo
Nome	varchar(20)
Mail	varchar(40)
Telddd	varchar(2)
Telnum	varchar(8)
Id	varchar(8)

A Tabela 9 mostra a tabela login que possui duas colunas: id e password.

Tabela 9 - Esquema para a tabela login.

Coluna	Tipo
Id	varchar(8)
Password	varchar(30)

Esta tabela é preenchida quando o usuário faz o cadastro, e pode ser modificada quando o usuário deseja alterar a senha ou pode ser alterada pelo sistema quando o usuário esquecer a senha.

A tabela recursos possui todos os dados dos sistemas da agricultura da Flórida cadastrados. O esquema da tabela recursos pode ser vista na Tabela 10.

Tabela 10 - Esquema para a tabela recursos.

Coluna	Tipo
<i>Foto</i>	varchar(20)
<i>Nome</i>	varchar(20)
<i>Sun</i>	varchar(20)
<i>Rain</i>	varchar(20)
<i>Evapo</i>	varchar(20)
<i>Topsoil</i>	varchar(20)
<i>Fuel</i>	varchar(20)
<i>Electricity</i>	varchar(20)
<i>Machinery</i>	varchar(20)
<i>Lime</i>	varchar(20)
<i>Pesticides</i>	varchar(20)
<i>Potash</i>	varchar(20)
<i>Phosphate</i>	varchar(20)
<i>Nitrogen</i>	varchar(20)
<i>Feed</i>	varchar(20)
<i>Livestock</i>	varchar(20)
<i>Humamlabor</i>	varchar(20)
<i>Services</i>	varchar(20)
<i>Identificador</i>	varchar(20)
<i>Id</i>	Int(11)
<i>Mass</i>	varchar(20)
<i>Moisture</i>	varchar(20)
<i>Carbohydrates</i>	varchar(20)
<i>Lipids</i>	varchar(20)
<i>Protein</i>	varchar(20)
<i>Price</i>	Varchar(20)
<i>Data</i>	Date

Os dados da quantidade de cada recurso são armazenados nos seus respectivos campos. Os dados originais de todos os produtos estudados da agricultura da Flórida são cadastrados nesta tabela com o identificador com o valor “original”. Quando um usuário grava sua análise, o campo identificador é preenchido com o id do usuário e o campo id da tabela é preenchido com um valor inteiro único para cada análise gravada pelo usuário. Este valor de id da tabela é utilizado para distinguir as análises gravadas pelo usuário.

A tabela dadosBr foi criada para armazenar os dados dos sistemas da agricultura do Brasil. Os sistemas da agricultura do Brasil não são padronizados. Cada sistema possui um número e tipos de recursos diferentes, o que ocasiona um problema para serem

armazenados em um banco de dados. Para solucionar esse problema, foram criadas colunas genéricas com o nome Di, onde i varia de 1 a 50. Ou seja, foram criadas as colunas D1, D2, D3, até a coluna D50. As colunas são preenchidas com as quantidades dos recursos utilizados no sistema seguindo a ordem de 1 a 50. Se um sistema utiliza 30 recursos, serão preenchidas as colunas D1 a D30 e as colunas restantes ficarão vazias. Foi criada uma coluna chamada contador, que informa o valor de quantos recursos são utilizados pelo sistema, no caso citado acima, o valor deste contador seria 30. Nesta tabela ainda foram criadas as seguintes colunas: nome, identificador, foto, mass, moisture, carbohydrates, lipids, protein, price, id e data.

4.5.3 Criação dos arquivos JSP.

A criação das páginas JSP utiliza tags JSP, HTML e tags personalizadas do struts framework. Os primeiros comandos de todos os arquivos JSP é importar as bibliotecas de tags personalizadas do struts. A Figura 17 mostra essas tags.

A maioria das páginas JSP possuem caixas de texto que são preenchidas pelo usuário e botões para enviar as informações. Também são utilizados links que permitem que o usuário vá para determinado local da página e caixas de seleção que permitem que o usuário escolha um item desejado.

```
<%@taglib uri="/WEB-INF/struts-bean.tld" prefix="bean"%>
<%@taglib uri="/WEB-INF/struts-html.tld" prefix="html"%>
<%@taglib uri="/WEB-INF/struts-logic.tld" prefix="logic"%>
<%@taglib uri="/WEB-INF/struts-nested.tld" prefix="nested"%>
<%@taglib uri="/WEB-INF/struts-template.tld" prefix="template"%>
<%@taglib uri="/WEB-INF/struts-tiles.tld" prefix="tiles"%>
```

Figura 17 - Tags que importam as bibliotecas de tags personalizadas do struts framework.

4.5.4 Parte lógica das páginas JSP com o struts *framework*.

Através do struts framework a parte de lógica das páginas é feita de uma forma fácil e simples. Primeiramente criamos um *ActionForm*, que é responsável por recolher todos os atributos digitados ou seleccionados pelo usuário e enviar para o processamento.

Na Figura 18 podemos ver parte do arquivo `loginActionForm.java`, criado para receber o login e a senha do usuário.

```
public class loginActionForm
    extends ActionForm {
    private String login;
    private String password;

    public String getLogin() {
        return login;}
    public void setLogin(String login) {
        this.login = login; }
    public void setPassword(String password) {
        this.password = password; }
    public String getPassword() {
        return password; }
```

Figura 18 - Parte do código do arquivo `loginActionForm.java`.

Esta classe é referenciada no arquivo JSP pela tag `<jsp:useBean id="loginActionForm" scope="session" class="emergytable.loginActionForm"/>`

A Figura 19 mostra uma parte da página `login.jsp`. Nesta página JSP temos os atributos `login` e `password`. Quando o botão `Submit` é pressionado, os valores digitados nas caixas de texto são passados para o `loginActionForm`.

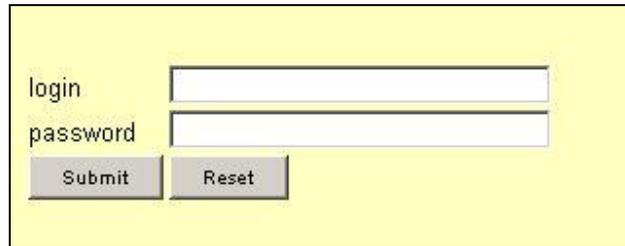
A screenshot of a login form on a yellow background. It features two text input fields: the top one is labeled 'login' and the bottom one is labeled 'password'. Below the input fields are two buttons: 'Submit' on the left and 'Reset' on the right.

Figura 19 - Campos da página login.jsp.

No struts, as ações são definidas em classes com o final `Action`. No nosso exemplo, a classe de ação é a `loginAction.java` e ela é referenciada na nossa página JSP por `loginAction.do`.

A classe `loginAction.java` obtém os valores dos atributos `login` e `password` através do `loginActionForm` e faz o processamento de verificar se o login e senha digitados conferem com os armazenados no banco de dados. Toda classe *Action* deve ter uma resposta, que deve ser retornada através da seguinte comando: `return actionMapping.findForward(destino)`.

Neste caso é retornada uma string destino, que pode receber o valor `good` ou `bad`. Se o login e usuário estiverem corretos, a string destino recebe o valor `good`. Caso contrário a string destino recebe o valor `bad`. Esse valor implicará em qual página JSP será encaminhada para o usuário. A Figura 20 mostra o esquema de como as páginas são encaminhadas dependendo da resposta recebida. Neste caso, se a resposta for `bad` o usuário será encaminhado para a página `loginError.jsp` e se o resultado for `good` o usuário será encaminhado para a página `principal0.jsp`. Nesta etapa, podemos clicar no nosso arquivo `jsp` e arrastá-lo até o campo `forward`. Digitamos a referência da resposta enviada pela classe *Action* e um novo encaminhamento é criado.

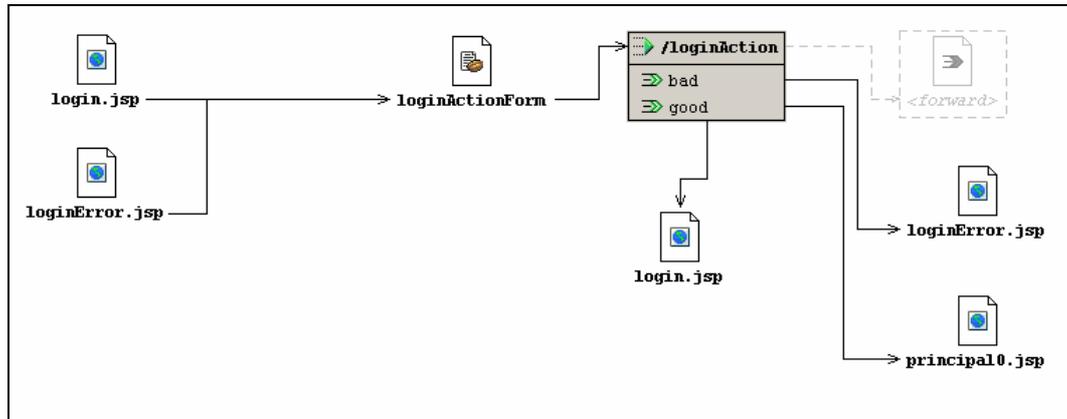


Figura 20 - Esquema de encaminhamento de páginas com o struts.

Este sistema de arrastar as páginas JSP para o campo *forward* é uma simplificação do *JBuilder*. Através desse processo, o arquivo de ação do struts, *struts-config.xml*, é modificado através da adição de *tags* que referenciam a ação. O elemento *form-beans* é adicionado, referenciando a adição do *ActionForm* e no elemento *action-mappings* é adicionada a ação de direcionamento das páginas. A Figura 21 mostra parte do arquivo *struts-config.xml*, onde podemos observar as *tags* que foram criadas automaticamente pelo *JBuilder* para gerenciar as ações da página de login.

```

<struts-config>
  <form-beans>
    <form-bean name="loginActionForm" type="emergytable.loginActionForm" />
    .
    .
  </form-beans>
  <action-mappings>
    <action input="/login.jsp" name="loginActionForm" path="/loginAction" scope="request"
    type="emergytable.loginAction">
      <forward name="bad" path="/loginError.jsp" />
      <forward name="good" path="/principal0.jsp" />
    </action>
  </action-mappings>
</struts-config>
  
```

Figura 21 - Parte do arquivo *struts-config.xml* referenciando as ações de *login.jsp*.

Para ilustrar melhor a utilização do *struts framework*, utilizaremos outro exemplo onde o usuário entra com dados em um formulário para criar uma nova análise. O arquivo responsável por essa parte do processo é o `campos1.jsp` e parte da visualização deste arquivo pode ser vista na Figura 22.

Modifique os valores de cada recurso do produto Bell Pepper

Sun :	<input type="text" value="123545"/>	J/m ² /year		
Rain :	<input type="text" value="1.3716"/>	m ³ /m ² /year		
Evapotranspiration :	<input type="text" value="54300000000"/>	J/m ² /year		
Net Topsoil Loss :	<input type="text" value="8500"/>	kg/ha/year		
Fuel :	<input type="text" value="57984"/>	litres/ha/year	Dados dos Produto	
Electricity :	<input type="text" value="208"/>	K Wh/ha/year	Mass	<input type="text" value="28000"/> kg/ha/year
Machinery :	<input type="text" value="0"/>	kg/ha/year	Moisture	<input type="text" value="93.5"/> %
Lime :	<input type="text" value="0"/>	kg/ha/year	Carbohydrates	<input type="text" value="63"/> %
Pesticides :	<input type="text" value="131"/>	kg/ha/year	Lipids	<input type="text" value="11"/> %
Potash :	<input type="text" value="172"/>	kg K/ha/year	Protein	<input type="text" value="26"/> %
Phosphate :	<input type="text" value="52.7"/>	kg P/ha/year	Price	<input type="text" value="0.5"/> US\$/kg
Nitrogen :	<input type="text" value="44"/>	kg N/ha/year		
Feed, Grain :	<input type="text" value="0"/>	J/ha/year		
Feed, Livestock :	<input type="text" value="0"/>	J/ha/year		
Human Labor :	<input type="text" value="1200"/>	hours/ha/year		
Services from Economy :	<input type="text" value="2110"/>	US\$/ha/year		

Figura 22 - Tela do formulário de dados que pode ser preenchido pelo usuário.

É necessário criar um *ActionForm*, que é responsável por recolher todos os dados digitados pelo usuário. Foi criado então o arquivo `recursosActionForm.java` que tem a mesma lógica do arquivo `loginActionForm.java` visto anteriormente na Figura 18. O arquivo `recursosActionForm.java` pode ser visualizado no Apêndice D.

Quando o usuário clicar no botão “Salvar”, a classe `resultadoCriarAction.java` irá recolher todos os dados fornecidos pelo usuário e irá verificar se há algum erro nas

informações digitadas pelo usuário. Um objeto *ActionErrors* chamado erro é criado e se for encontrado algum erro, o mesmo é adicionado a esse objeto. Se o objeto erro estiver vazio, significa que os dados estão corretos e podem ser gravados no banco de dados, caso contrário, uma tela informando os campos incorretos será apresentada ao usuário. Em ambos os casos, o usuário será direcionado para a mesma página JSP, campos1.jsp, através do comando `return actionMapping.findForward("sucesso")`, como pode ser visto na Figura 23.

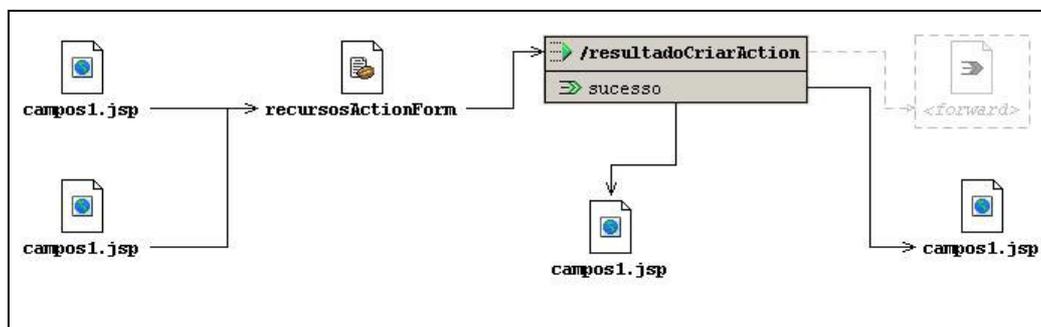


Figura 23 - Esquema do encaminhamento de páginas do processo de criar uma novas análises.

Este processo é possível porque a classe resultadoCriarAction.java envia dois atributos para a página JSP. Um atributo é a string status, que informa se os dados foram gravados no banco de dados ou não e o atributo erro que é o objeto *ActionErrors* criado pela classe resultadoCriarAction.java. O arquivo JSP verifica se a variável status possui o valor “gravou”. Se for encontrado esse valor, a informação que os dados foram gravados com sucesso será apresentada para o usuário e um link para a visualização do resultado aparecerá na tela. Caso a variável status não possuir o valor “gravou”, a página JSP lê o atributo erro e apresenta para o usuário quais campos contêm valores errados.

Esta mesma metodologia foi utilizada para criação de todas as páginas JSP e de toda a parte lógica das páginas.

Para os produtos da agricultura da Flórida, uma única página JSP de resultado, com tags XML, foi criada. Isto foi possível porque os arquivos XML da Flórida são padronizados. Um arquivo XML padrão foi criado e o resultado é obtido quando os valores gravados pelo usuário são adicionados nas tags XML do arquivo JSP.

A Figura 24 mostra parte do arquivo de resposta que possui as tags XML e como os valores gravados pelo usuário são adicionados a esse arquivo. Para construir esse arquivo, o atributo foto deve receber uma referencia, e ela é obtida pelo comando `<%=camp.getFoto()%>`. Neste caso camp é um objeto que foi criado para receber todas as informações do banco de dados, e `getFoto()` é o método utilizado para obter o valor da variável foto.

Esse mesmo processo acontece para o elemento OBJECTIVE com o comando `<%=camp.getNome()%>`, para o elemento QUANTITY de todos os recursos utilizados e para os dados relacionados ao produto, PRODUCT nos elementos VALUE.

Como podemos observar, a padronização dos arquivos do sistema da Flórida simplifica o processo. Uma vez que todos os produtos utilizam os mesmos recursos, foi possível criar um único arquivo de resposta. Para os produtos do Brasil, isso não foi possível porque cada produto possui um número diferente de recursos utilizados, então houve a necessidade de criar um arquivo de resposta para cada produto.

```
<?xml version="1.0" encoding="ISO8859-1"?>
<?xml-stylesheet href="table.xslt" type="text/xsl"?>
<TABLE>
  <SYSTEM>
    <TITLE foto="<%=camp.getFoto()%>">Folio # 4. Agricultural Systems of Florida</TITLE>
    <AUTHORS>Sherry L. Brandt-Williams</AUTHORS>
    <INSTITUTION>University of Florida, 2001 (revised 2004)</INSTITUTION>
    <OBJECTIVE><%=camp.getNome()%>
(USA, 1981)</OBJECTIVE>
    <EMDOLLAR>270000000000</EMDOLLAR>
  </SYSTEM>
  <RENEWABLE>
    <CODE>R1</CODE>
    <ITEM>Sun</ITEM>
    <QUANTITY><%=camp.getSun()%></QUANTITY>
.
.
.
<PRODUCT line="06">
  <PROPERTY>Mass (wet)</PROPERTY>
  <VALUE><%=camp.getMass()%></VALUE>
  <UNITS>kg/ha/year</UNITS>
  <AD1 />
  <AD2 />
</PRODUCT>
```

Figura 24 - Parte do arquivo resposta.jsp.

5. RESULTADOS E DISCUSSÃO

5.1 O aplicativo desenvolvido.

O aplicativo Energy Table foi desenvolvido para modificar os arquivos XML que representam os sistemas agrícolas. A interface com o usuário pode ser vista na Figura 25.

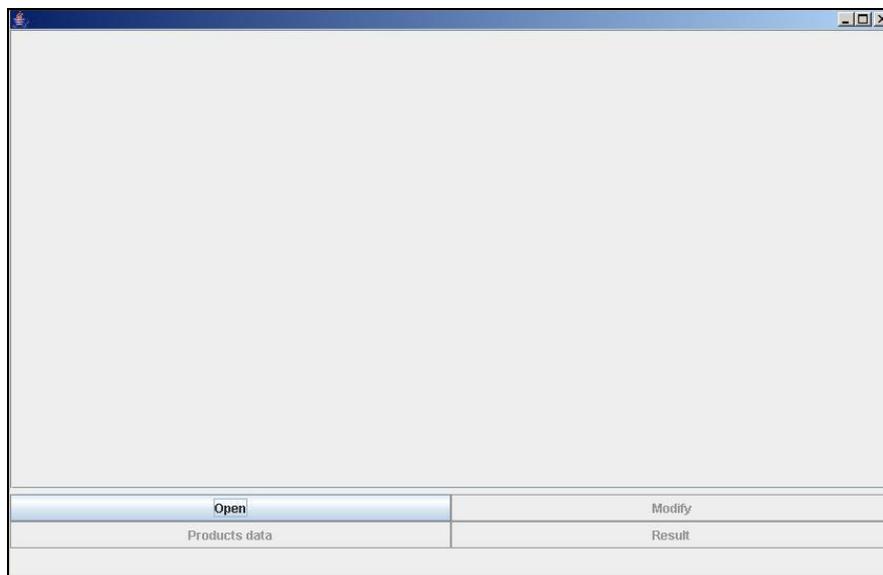


Figura 25 - Interface do aplicativo Energy Table.

O usuário deve clicar no botão “*Open*” para abrir um arquivo XML. Ao escolher essa opção, uma caixa de diálogo para a escolha de arquivos é aberta como podemos observar na Figura 26.

Ao abrir o arquivo, os botões “products data” e “modify” são habilitados para que o usuário possa clicar. Os dados do arquivo são mostrados na tela do aplicativo e as caixas de texto amarelas, que representam a quantidade de cada recurso, podem ser alteradas pelo usuário. Podemos observar na Figura 27 a tela de apresentação do arquivo bellPepper.xml aberto pelo usuário.

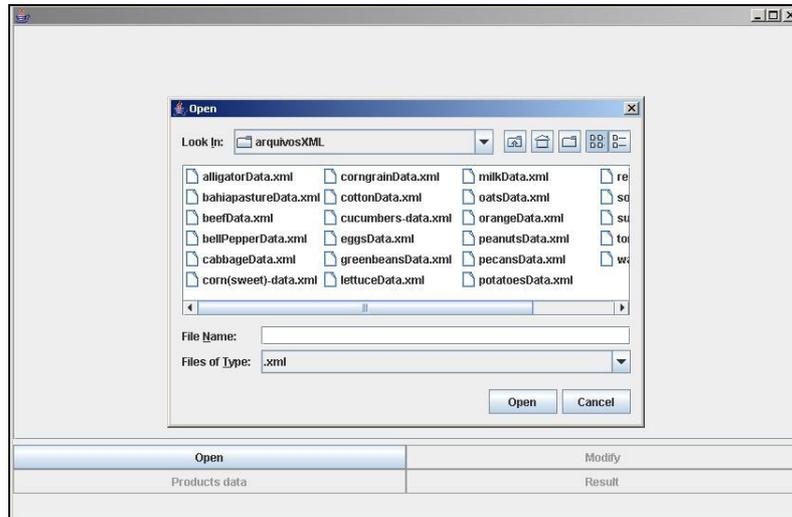


Figura 26 - Caixa de diálogo para abrir um arquivo XML.

Code	Item	Quantity	Units	ConversionFactor	Transformities of inputs(sej/unit)	Energy flow E13 sej/ha/yr
R1	Sun	6350000000	J/m ² /year	10000	1	6.3500000000
R2	Rain	1.3716	m ³ /m ² /year	459400000000	30240	190.5461832960
R3	Evapotranspiration	5430000	J/m ² /year	10000	25910.64	140.6947752000
N1	Net Topsoil Loss	8504.98	kg/ha/year	904176	123984	95.3436810783
M1	Fuel	1597.16	litres/ha/year	34874504.62	110880	617.6034162020
M2	Electricity	208.06	kWh/ha/year	3600000	268800	20.1335500800
M3	Machinery	0	kg/ha/year	1	11200000000000	0.0000000000
M4	Lime	0.00	kg/ha/year	1000	1680000000	0.0000000000
M5	Pesticides	131	kg/ha/year	1000	24864000000	325.7184000000
M6	Potash	172	kg K/ha/year	1000	2923200000	50.2790400000
M7	Phosphate	52.7	kg P/ha/year	1000	36960000000	194.7792000000
M8	Nitrogen	44	kg N /ha/year	1000	40488000000	178.1472000000
M9	-	820	-	0	0	0.0000000000
M10	-	0	-	0	0	0.0000000000
S1	Human Labor	16.32	hours/ha/year	100464000	4450000	729.6097536000
S2	Services from economy	2110	US\$/ha/year	1	4536000000000	957.0960000000

Figura 27 - Aplicativo com o arquivo bellPepperData.xml aberto.

Uma vez que o usuário altera as caixas de texto, ele deve clicar no botão “modify” para que os dados sejam gravados e ele ainda pode alterar os dados referentes ao produto clicando no botão “Products data”.

Ao clicar no botão “Products data” uma nova janela é aberta com os dados referentes ao produto, e da mesma maneira podemos alterar os dados em amarelo e clicar no botão “*Modify*” para gravar os dados. A Figura 28 mostra a janela aberta para modificar os dados do produto.

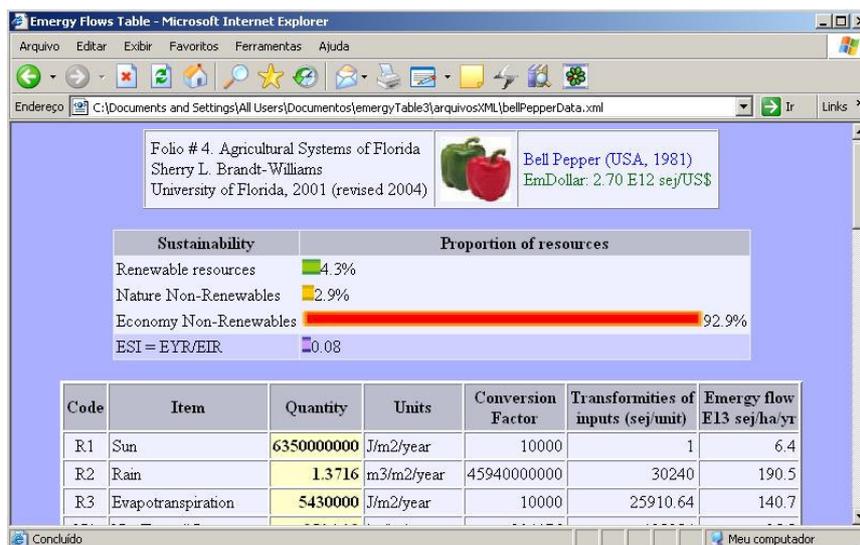


Property	Value	Units	CaloricValue	Unit
Mass (wet)	28000	kg/ha/year		
Moisture	93.5	%		
Carbohydrates	63	%	17000000	J/kg
Lipids	11	%	39000000	J/kg
Protein	26	%	24000000	J/kg
Price (wet product)	0.5	US\$/kg		

modify

Figura 28 - Janela para modificar os dados do produto.

Depois de clicar no botão “*Modify*”, o botão “*Result*” é habilitado. Ao clicar no botão “*Result*”, o arquivo XML é aberto em uma janela do Internet Explorer como mostrado na Figura 29.



Folio # 4. Agricultural Systems of Florida
Sherry L. Brandt-Williams
University of Florida, 2001 (revised 2004)

Bell Pepper (USA, 1981)
EmDollar: 2.70 E12 sej/US\$

Sustainability		Proportion of resources	
Renewable resources	4.3%		
Nature Non-Renewables	2.9%		
Economy Non-Renewables			92.9%
ESI = EYR/EIR	0.08		

Code	Item	Quantity	Units	Conversion Factor	Transformities of inputs (sej/unit)	Energy flow E13 sej/ha/yr
R1	Sun	6350000000	J/m ² /year	10000	1	6.4
R2	Rain	1.3716	m ³ /m ² /year	45940000000	30240	190.5
R3	Evapotranspiration	5430000	J/m ² /year	10000	25910.64	140.7

Figura 29 - Resultado apresentado no Internet Explorer.

Todas as etapas do processo podem ser resumidas pelo esquema mostrado na Figura 30, adaptado de Takahashi et al. (2006).

1. O usuário abre o emergytable.
2. O usuário abre o arquivo XML
3. O arquivo XML é apresentado para o usuário e pode ser modificado.
4. O usuário modifica o arquivo XML.
5. O usuário faz o requerimento de visualizar o resultado.
6. O Internet Explorer 6.0 é aberto e faz o requerimento do arquivo XML.
7. O arquivo XML é repassado para o Internet Explorer 6.0.
8. O documento XML faz a requisição do documento XSLT.
9. O usuário visualiza o resultado.

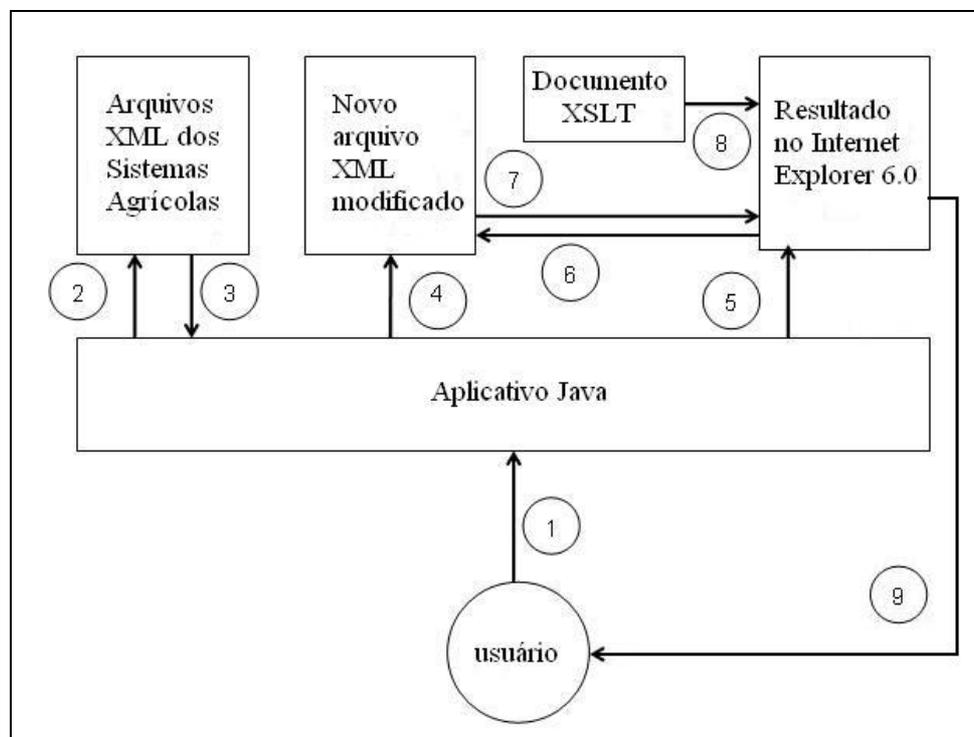


Figura 30 - Esquema de todo o processo realizado na execução emergytable (TAKAHASHI et al. 2006) .

Podemos verificar que através de um simples aplicativo de interface com o usuário é possível alterar os arquivos e visualizar o resultado de uma forma rápida e fácil. O mesmo aplicativo é utilizado para os arquivos da Flórida e do Brasil, pois o aplicativo lê todos os elementos dos arquivos XML e cria caixas de texto para cada um dos recursos utilizados, independente da quantidade. Durante o processamento da modificação do arquivo, além da modificação do elemento QUANTITY, o elemento EMERGY que representa o fluxo energético também é alterado com o novo valor do fluxo energético calculado pelo aplicativo. Este valor deve ser modificado porque o arquivo XSLT utiliza esses valores para processar os resultados. Na primeira tabela de resultados o arquivo XSLT calcula o fluxo energético multiplicando os elementos QUANTITY, FACTOR e TRANSFORMITY o que implica em um resultado correto. Mas para a tabela de fluxo agregado de energia, o arquivo XSLT utiliza uma variável que soma os valores dos elementos EMERGY. Por exemplo, para os recursos renováveis é criada uma variável R que soma todos os elementos EMERGY contidos nos elementos RENEWABLE (linha 5 do código do arquivo XSLT no Apêndice B). Se os valores da tag EMERGY não fossem alterados, o arquivo XSLT obteria os valores errados do fluxo energético e os resultados seriam incorretos. Desta forma, um único arquivo XSTL pode ser usado para apresentar os resultados.

Para as páginas web dinâmica do sistema da Flórida, foi utilizada uma versão única e com pequenas alterações no arquivo XSLT. Já para os sistemas do Brasil isso não foi possível, como será discutido posteriormente.

O aplicativo desenvolvido está disponível no site <http://www.unicamp.br/fea/ortega/em-folios/software/index.htm> .

5.2 Páginas web dinâmicas desenvolvidas.

Um esquema que representa as páginas web dinâmicas desenvolvidas pode ser visto na Figura 31.

- 1- O usuário acessa a página
- 2- A página faz requisição ao servlet que neste trabalho é controlado pelo struts.
- 3- O servlet recolhe informações do banco de dados.
- 4- As informações do banco de dados são passadas para as páginas JSP.
- 5- As páginas JSP são visualizadas pelo usuário.
- 6- As páginas de resposta em XML fazem o requerimento dos arquivos XSLT para que o usuário possa visualizar os resultados.

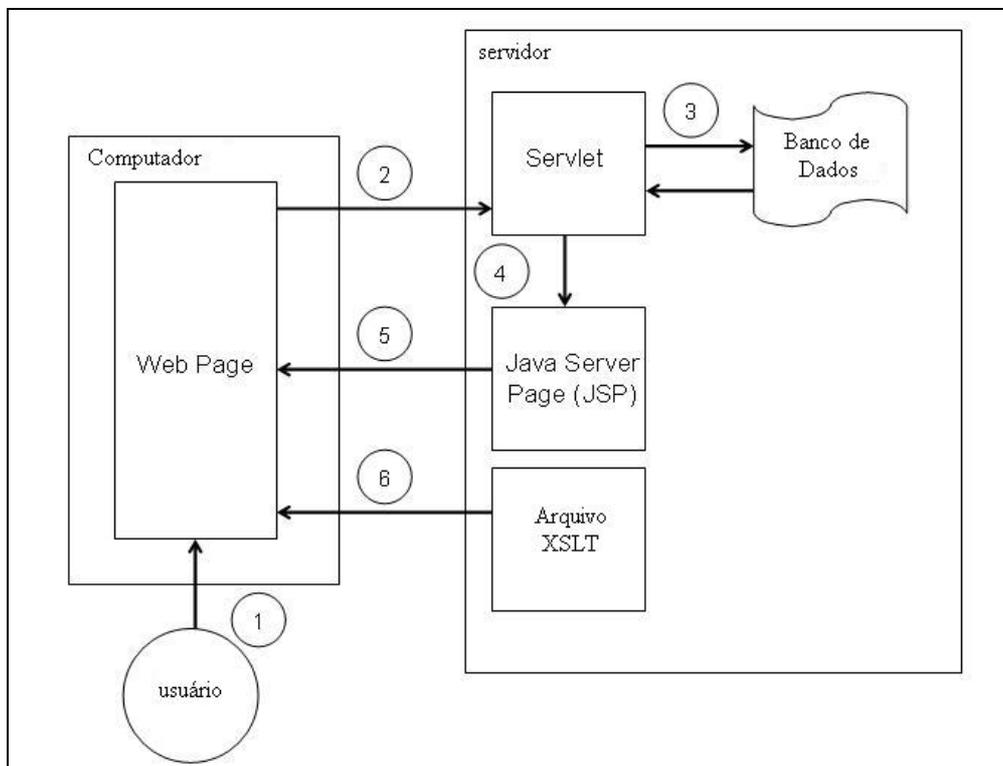


Figura 31 - Esquema simplificado das páginas criadas.

Para acessar as páginas web dinâmica é necessário efetuar um cadastro. Este cadastro é feito na página de login. Nesta página, além do cadastro, o usuário pode ter acesso as páginas inserindo o nome do usuário e a senha e recuperar sua senha caso tenha se esquecido. A Figura 32 mostra parte da página de login.

Bem vindo ao EMERGY TABLES

login:

password:

[Novo usuário](#) [Esqueceu a senha?](#)

Figura 32 - Parte da página de login.

Para efetuar o cadastro o usuário deve clicar no link “Novo usuário”. No cadastro o usuário deve fornecer os seguintes dados: Nome, login, e-mail, telefone, senha e a confirmação da senha. A Figura 33 mostra o formulário de cadastro de usuário.

Formulário de cadastro de usuário

nome:

login:

e-mail:

telefone: -

senha:

confirmarsenha:

Figura 33 - Parte da página de cadastro de usuário.

Ao se cadastrar, todos os dados são gravados no banco de dados, sendo que a senha é criptografada antes de ser gravada.

Caso o usuário esqueça a senha de acesso, ele pode recuperar a senha clicando no link “esqueceu a senha?”. Uma nova página será aberta onde o usuário deve digitar o login. Deste modo, uma nova senha será enviada para o e-mail do usuário. Esta página pode ser vista na Figura 34.

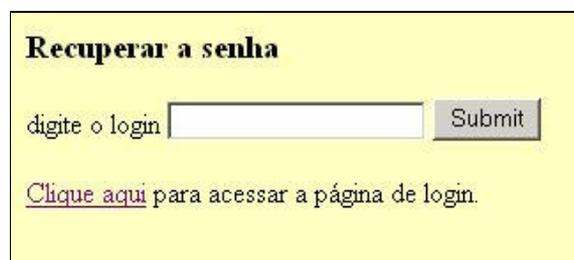
A imagem mostra uma interface web com o título "Recuperar a senha" em negrito. Abaixo do título, há o texto "digite o login" seguido de um campo de entrada de texto e um botão "Submit". Abaixo do campo de entrada, há um link azul com o texto "Clique aqui para acessar a página de login.".

Figura 34 - Parte da página para recuperar a senha.

Para acessar as páginas o usuário deve entrar com o nome de usuário e senha. Se os dados estiverem corretos o usuário será encaminhado para uma página onde ele pode escolher o sistema de agricultura da Flórida ou o sistema de agricultura do Brasil como pode ser visto na Figura 35.

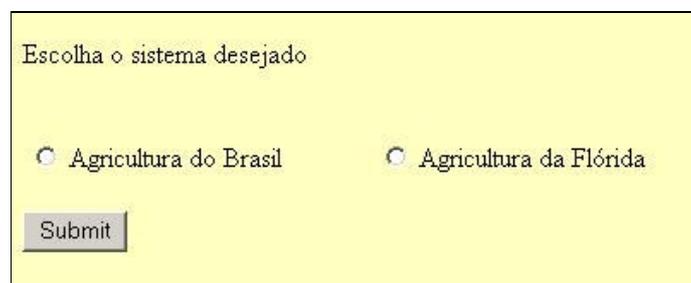
A imagem mostra uma interface web com o título "Escolha o sistema desejado". Abaixo do título, há duas opções de seleção com botões de rádio: "Agricultura do Brasil" e "Agricultura da Flórida". Abaixo das opções, há um botão "Submit".

Figura 35 - Página de escolha do sistema desejado

Uma vez escolhido o sistema o usuário pode efetuar os seguintes comandos através dessa página inicial: Acessar análises originais, criar nova análise, abrir uma análise que foi

gravada, trocar de sistema e ainda alterar a senha de acesso. Parte da página inicial pode ser vista na Figura 36 .

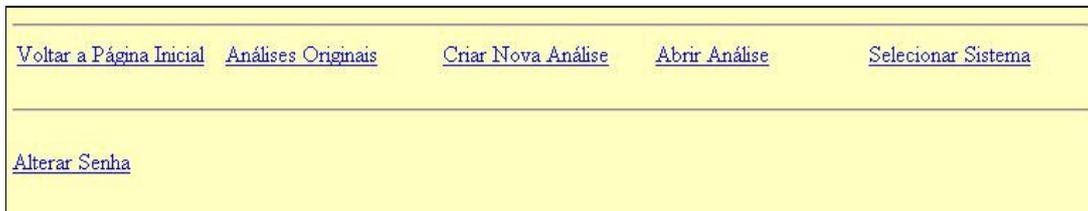
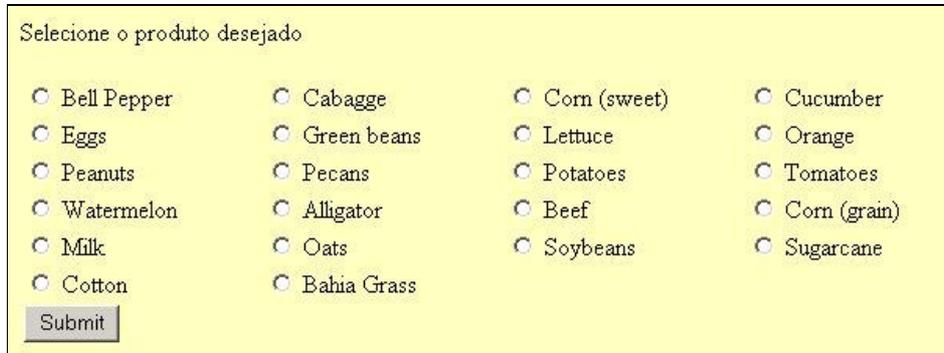


Figura 36 - Parte da página inicial.

Acessando o link “Análises Originais”, o usuário tem acesso a uma lista de produtos da agricultura para escolher. Ao selecionar um produto e apertar o botão submit, uma página com o arquivo XML é processada com os dados originais do produto. Esta página para o sistema da agricultura da Flórida é mostrada na Figura 37

A imagem mostra uma página amarela com o título "Selecione o produto desejado". Abaixo do título, há uma lista de produtos agrícolas em quatro colunas, cada um com um botão de seleção (radio button):

- Bell Pepper
- Cabagge
- Corn (sweet)
- Cucumber
- Eggs
- Green beans
- Lettuce
- Orange
- Peanuts
- Pecans
- Potatoes
- Tomatoes
- Watermelon
- Alligator
- Beef
- Corn (grain)
- Milk
- Oats
- Soybeans
- Sugarcane
- Cotton
- Bahia Grass

Na base da lista, há um botão cinza "Submit".

Figura 37 - Parte da página de escolha do produto.

Acessando o link “Criar Nova Análise”, o usuário tem acesso a uma lista de produtos da agricultura para escolher, a mesma lista mostrada na Figura 37. Ao selecionar um produto e apertar o botão submit, uma página com um formulário é criada, onde o usuário pode digitar o valor da quantidade de cada recurso utilizado nas caixas de texto como pode ser visto na Figura 38. Inicialmente as caixas de texto são preenchidas com os

valores originais de cada produto. O usuário modifica esses valores e clica no botão “Salvar”.

Modifique os valores de cada recurso do produto Bell Pepper

Sun :	<input type="text" value="123545"/>	J/m ² /year		
Rain :	<input type="text" value="1.3716"/>	m ³ /m ² /year		
Evapotranspiration :	<input type="text" value="54300000000"/>	J/m ² /year		
Net Topsoil Loss :	<input type="text" value="8500"/>	kg/ha/year		
Fuel :	<input type="text" value="57984"/>	litres/ha/year	Dados dos Produto	
Electricity :	<input type="text" value="208"/>	KWh/ha/year	Mass	<input type="text" value="28000"/> kg/ha/year
Machinery :	<input type="text" value="0"/>	kg/ha/year	Moisture	<input type="text" value="93.5"/> %
Lime :	<input type="text" value="0"/>	kg/ha/year	Carbohydrates	<input type="text" value="63"/> %
Pesticides :	<input type="text" value="131"/>	kg/ha/year	Lipids	<input type="text" value="11"/> %
Potash :	<input type="text" value="172"/>	kg K/ha/year	Protein	<input type="text" value="26"/> %
Phosphate :	<input type="text" value="52.7"/>	kg P/ha/year	Price	<input type="text" value="0.5"/> US\$/kg
Nitrogen :	<input type="text" value="44"/>	kg N/ha/year		
Feed, Grain :	<input type="text" value="0"/>	J/ha/year		
Feed, Livestock :	<input type="text" value="0"/>	J/ha/year		
Human Labor :	<input type="text" value="1200"/>	hours/ha/year		
Services from Economy :	<input type="text" value="2110"/>	US\$/ha/year		

Figura 38 - Formulário para alterar os valores dos recursos utilizados.

Se o usuário digitar valores corretos, uma informação de que os dados foram gravados com sucesso irá aparecer na tela e um *link* será criado para que o usuário tenha acesso à página de resultados, como pode ser visto na Figura 39. Caso o usuário tenha digitado algum valor incorreto nas caixas de texto, como letras, números mal formatados ou números negativos, uma tela indicando as caixas de texto com problemas será apresentada ao usuário e o mesmo pode fazer as devidas correções. Na Figura 40 podemos observar 3 erros. O primeiro erro se encontra no campo Sun onde o usuário digitou um valor negativo, o segundo erro no campo rain, onde o usuário digitou um valor não numérico, e o terceiro

erro no campo Evapotranspiration onde o usuário digitou um número mal formatado, utilizando vírgula (,) para separar as casas decimais no lugar do ponto (.).

Dados gravados com sucesso!!

[Clique aqui](#) para visualizar o resultado

Sun :	<input type="text" value="100000000"/>	J/m2/year		
Rain :	<input type="text" value="1.3716"/>	m3/m2/year		
Evapotranspiration :	<input type="text" value="54300000000"/>	J/m2/year		
Net Topsoil Loss :	<input type="text" value="8500"/>	kg/ha/year		
Fuel :	<input type="text" value="57984"/>	litres/ha/year	Dados dos Produto	
Electricity :	<input type="text" value="208"/>	KWh/ha/year	Mass	<input type="text" value="28000"/> kg/ha/year
Machinery :	<input type="text" value="0"/>	kg/ha/year	Moisture	<input type="text" value="93.5"/> %
Lime :	<input type="text" value="0"/>	kg/ha/year	Carbohydrates	<input type="text" value="63"/> %
Pesticides :	<input type="text" value="131"/>	kg/ha/year	Lipids	<input type="text" value="11"/> %
Potash :	<input type="text" value="172"/>	kg K/ha/year	Protein	<input type="text" value="26"/> %
Phosphate :	<input type="text" value="52.7"/>	kg P/ha/year	Price	<input type="text" value="0.5"/> US\$/kg
Nitrogen :	<input type="text" value="44"/>	kg N/ha/year		
Feed, Grain :	<input type="text" value="0"/>	J/ha/year		
Feed, Livestock :	<input type="text" value="0"/>	J/ha/year		
Human Labor :	<input type="text" value="1200"/>	hours/ha/year		
Services from Economy :	<input type="text" value="2110"/>	US\$/ha/year		

Figura 39 - Página informando que os dados foram gravados com sucesso.

Os Seguintes erros foram encontrados:

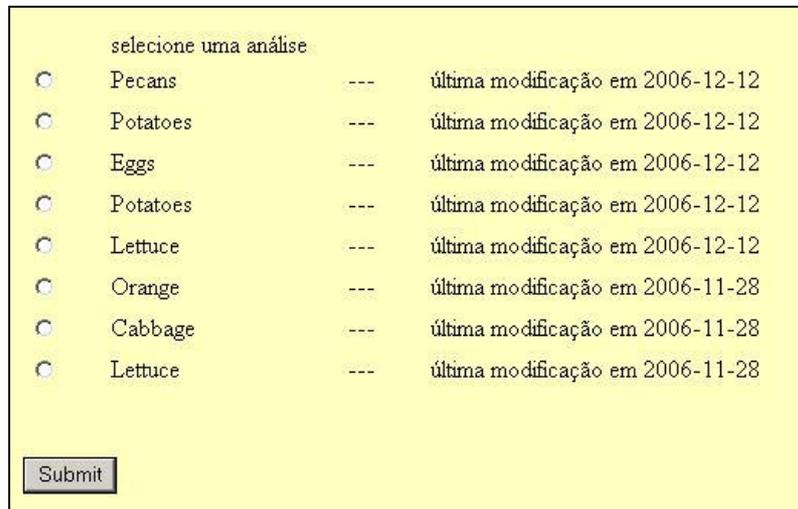
- O valor no campo sun deve ser numérico.
- O valor no campo rain deve ser numérico.
- O valor no campo evapotranspiration deve ser numérico.

Modifique os valores de cada recurso do produto Bell Pepper

Sun :	<input type="text" value="-56256"/>	J/m2/year
Rain :	<input type="text" value="ABC"/>	m3/m2/year
Evapotranspiration :	<input type="text" value="5.36985"/>	J/m2/year
Net Topsoil Loss :	<input type="text" value="8500"/>	kg/ha/year

Figura 40 - Parte da página que informa ao usuário os erros encontrados.

Acessando o *link* “Abrir Análise” o usuário tem acesso à lista de análises gravadas, como é mostrada na Figura 41.

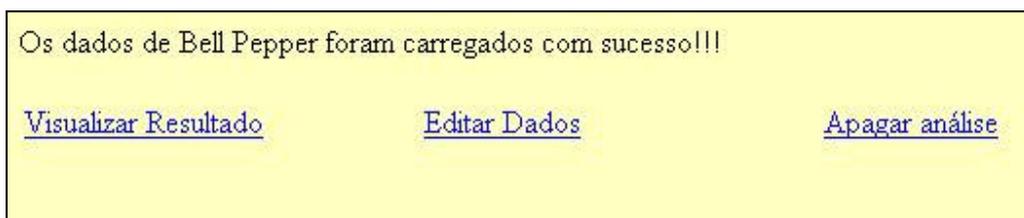


selecione uma análise

<input type="radio"/>	Pecans	---	última modificação em 2006-12-12
<input type="radio"/>	Potatoes	---	última modificação em 2006-12-12
<input type="radio"/>	Eggs	---	última modificação em 2006-12-12
<input type="radio"/>	Potatoes	---	última modificação em 2006-12-12
<input type="radio"/>	Lettuce	---	última modificação em 2006-12-12
<input type="radio"/>	Orange	---	última modificação em 2006-11-28
<input type="radio"/>	Cabbage	---	última modificação em 2006-11-28
<input type="radio"/>	Lettuce	---	última modificação em 2006-11-28

Figura 41 - Análises gravadas pelo usuário.

O usuário escolhe uma das análises e clica no botão submit. A análise é aberta e o usuário tem as opções de visualizar os resultados, editar os dados ou apagar a análise, como mostrado na Figura 42



Os dados de Bell Pepper foram carregados com sucesso!!!

[Visualizar Resultado](#) [Editar Dados](#) [Apagar análise](#)

Figura 42 - Parte da página que mostra que uma análise foi aberta com sucesso.

Ao clicar no *link* “Editar dados”, uma tela parecida com a Figura 38 será aberta. Os dados contidos nas caixas de texto são aqueles gravados pelo usuário. O usuário pode, da mesma maneira, alterar esses valores e gravar a análise no banco de dados.

O usuário pode clicar na opção “Apagar análise”, e uma tela perguntando se ele realmente deseja apagar a análise será apresentada. Caso o usuário confirme, a análise será apagada do banco de dados do servidor.

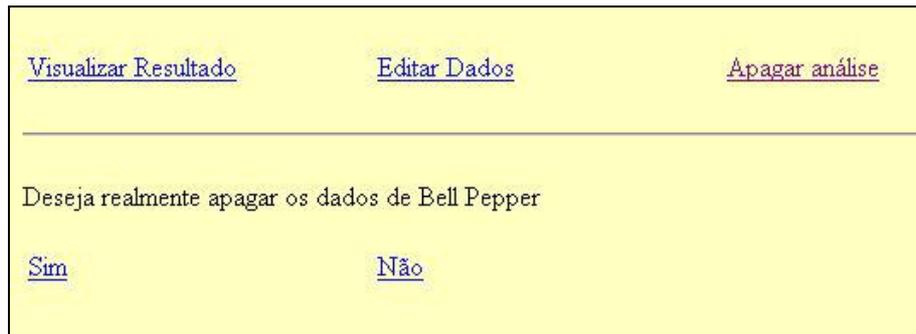


Figura 43 - Parte da página para confirmar se o usuário realmente deseja apagar a análise.

5.2.1. Páginas de resposta

As páginas de respostas são páginas JSP com as *tags* do arquivo XML inseridas. Para os produtos da agricultura da Flórida, uma única página JSP foi criada para visualizar os resultados. Esta página JSP possui as *tag* de um arquivo XML do sistema da Flórida padrão. Como foi visto anteriormente, somente as *tag* QUANTITY são preenchidas com valores do banco de dados. As *tags* TRANSFORMITY e EMERGY continuam com valores constantes. As *tags* TRANSFORMITY e EMERGY poderiam ser alteradas, mas isso resultaria em um aumento do processamento do sistema, uma vez que teríamos que condicionar as transformidades no banco de dados e teríamos que processar os cálculos. Para solucionar esse problema foi feita uma pequena modificação no arquivo XSLT.

Foram criadas novas variáveis para cada fluxo energético de um recurso. Ou seja, foi criada a variável R1 com resultado da multiplicação das *tags* QUANTITY, FACTOR e

TRANSFORMITY do primeiro recurso, a variável R2 para o resultado do fluxo energético do segundo recurso renovável e a mesma lógica foi seguida para os demais recursos.

Essas variáveis são criadas através das tags mostradas na Figura 44:

```
<xsl:variable name="R1"
select="(TABLE/RENEWABLE[1]/QUANTITY)*
(TABLE/RENEWABLE[1]/FACTOR)*(TABLE/RENEWABLE[1]/TRANSFORMITY)"/>
```

Figura 44 - Tag do arquivo XSLT para criar a variável R1

Para cada tipo de contribuição foi criada as variáveis R, N, M, S, que são as somas de todos os fluxo energéticos dos recursos utilizados como podemos ver nas equação (7), (8), (9) e (10).

$$R = \sum_{i=1}^n Ri \quad (7)$$

$$N = \sum_{i=1}^n Ni \quad (8)$$

$$M = \sum_{i=1}^n Mi \quad (9)$$

$$S = \sum_{i=1}^n Si \quad (10)$$

Está operação foi necessária porque no arquivo XSLT original, as variáveis R, N, M e S recebiam os valores da soma dos elementos EMERGY e isto não é possível nas páginas JSP visto que esses elementos são estáticos nessas páginas.

Para o sistema da Flórida, um só arquivo XSLT foi criado, uma vez que todos os produtos possuem o mesmo número de recursos, ou seja, o valor de "n" nas equações é sempre constante para cada tipo de contribuição.

Para o sistema do Brasil, uma página de resposta para cada produto foi criada. Isto foi necessário visto que cada produto utiliza recursos diferentes. Para o processamento das páginas, também foi necessário o desenvolvimento de um arquivo XSLT para cada produto porque para cada produto temos uma quantidade diferente de recursos, ou seja, o valor de n nas equações acima é diferente para cada produto. Para exemplificar melhor, o produto Trigo-1500, utiliza 7 materiais diferentes e o fluxo energético da contribuição M é calculado através da *tag* mostrada na Figura 45.

```
<xsl:variable name="M" select="($M1+$M2+$M3+$M4+$M5+$M6+$M7) div 10000000000000"/>.
```

Figura 45 - Tag do arquivo XSLT utilizada para criar a variável M do produto Trigo-1500.

Já para o produto Banana, são utilizados 11 materiais diferentes e o fluxo energético da contribuição M (materiais) é calculado através da tag mostrada na Figura 46.

```
<xsl:variable name="M" select="($M1+$M2+$M3+$M4+$M5+$M6+$M7+$M8+$M9+$M10+$M11) div 10000000000000"/>
```

Figura 46 - Tag do arquivo XSLT utilizada para criar a variável M do produto Banana.

Essas tags são diferentes para cada produto, por isso a necessidade da construção de arquivos XSLT diferentes para cada produto.

As páginas web podem ser acessada através do seguinte endereço:
<http://143.106.43.4:8080/WebModule1/login.jsp>.

6. CONCLUSÕES

Podemos concluir neste trabalho que os arquivos XML podem ser utilizados para representar sistemas agrícolas e que um mesmo modelo de arquivo XML pode ser utilizado para representar diversos sistemas de produção agrícola. Além disso, os arquivos XML possuem um tamanho reduzido, aproximadamente 9 KB, o que torna mais fácil o armazenamento, transferência e leitura dos mesmos.

O aplicativo desenvolvido se mostrou muito eficiente para alterar os documentos XML. Através da modificação dos arquivos XML podemos obter novos índices emergéticos de uma maneira rápida e prática de acordo com os valores das quantidades de recursos fornecidas pelo usuário. O aplicativo desenvolvido também possui um tamanho reduzido, aproximadamente 1,5MB, o que possibilita um menor tempo de download para usuários que queiram obter o aplicativo através da internet.

As páginas web dinâmicas desenvolvidas também se mostraram eficientes para o objetivo proposto. Porém, o tempo de programação gasto nas páginas JSP dos produtos da agricultura do Brasil foi bem maior que o tempo gasto nas páginas JSP da Flórida. Isto se deve a não padronização dos arquivos XML para os produtos do Brasil.

Os sistemas agrícolas brasileiros são bem mais complexos que os sistemas agrícolas da Flórida, o que torna difícil a padronização dos arquivos XML. Neste trabalho os sistemas agrícolas brasileiros não foram padronizados, mas podemos concluir pelos resultados obtidos no sistema da Flórida, que a padronização torna mais fácil a

programação das páginas web dinâmicas. Por este motivo, um grande esforço para a padronização dos arquivos XML deve ser realizado.

Como sugestão para futuras pesquisas, podemos deixar a elaboração de arquivos XML para a representação de sistemas de produção agrícolas mais complexos, como sítios, fazendas, com modelos de produção diferentes (químicos, biotecnológicos, orgânicos, agro-silvo-pastoris) onde temos mais de um tipo de produto final e a possibilidade de considerar serviços ambientais e externalidades negativas. Além de conseguir calcular os índices emergéticos, esses arquivos seriam uma ótima maneira de cadastrar propriedades agrícolas e ao mesmo tempo acompanhar a sustentabilidade de cada propriedade.

7. REFERENCIAS BIBLIOGRÁFICAS

AGOSTINHO, F. D. R. **Uso de análise emergética e sistemas de informações geográficas no estudo de pequenas propriedades agrícolas.** 2005. 226 f. Dissertação (Mestrado em Engenharia de Alimentos). Universidade Estadual de Campinas, Faculdade de Engenharia de Alimentos, Campinas.

ASSIS, R. L. de ; **Agroecologia no Brasil: análise no processo de difusão e perspectivas.** 2002. 150p. Tese (Programa de Pós-graduação em economia aplicada). Universidade Estadual de Campinas, Instituto de Economia, Campinas.

BEER, R. de; GRAVERON-DEMILLY, D.; NASTASE, S.; ORMONDT, D. van. A distributed computing system for magnetic resonance imaging: Java-based processing and binding of XML. **Computer Methods and Programs in Biomedicine.** V.73 p.221-231. 2004.

BJÖRKLUND, J.; GEBER, U.; RYDBERG, T. Emergy analysis of municipal wastewater treatment and generation of electricity by digestion of sewage sludge. **Resources Conservation & Recycling** v. 31 p. 293-316, 2001.

BRANDT-WILLIAMS, S. L. **Handbook of Emergy Evaluation: A Compendium of Data for Emergy Computation** Issued In a Series of Folios. Folio 4. Emergy of Florida Agriculture. 2002. Center for Environmental Policy. University of Florida, Gainesville, Florida, USA.

BROWN, M.T.; ARDING, J. Transformitis working paper. 1991. Center for Wetlands, University of Florida, Gainesville.

BRAY, T.; PAOLI, J.; SPERBERG-McQUEEN, C. M.; MALER, E.; YERGEAU, F. **Extensible markup language (xml) 1.0** (forth edition). 2006. Disponível em: <<http://www.w3.org/TR/REC-xml/>>. Acesso em: 14 dezembro 2006.

CAVALETT, O. **Análise emergética da piscicultura integrada à criação de suínos e de pesque-pagues.** 2004. 140 f. Dissertação (Mestrado em Engenharia de Alimentos.) Universidade Estadual de Campinas, Faculdade de Engenharia de Alimentos, Campinas.

CHEN, G.Q.; JIANG, M. M.; CHEN, B.; YANG, Z. F.; LIN, C. Emergy analysis of Chinese agriculture. **Agriculture Ecosystems & Environment.** V. 115 p. 161-173, 2006

CLARK, J. **XSL transformations (XSLT).** W3C Recommendation. 1999. Disponível em: <<http://www.w3.org/TR/xslt>>. Acesso em: 20 de setembro 2006.

COMAR, M. V. **Avaliação emergética de projetos agrícolas e agro-industriais**: a busca do desenvolvimento sustentável. 1998. 197 f. Tese (Doutorado em Engenharia de Alimentos.) Universidade Estadual de Campinas, Faculdade de Engenharia de Alimentos, Campinas.

DEITEL, H. M.; DEITEL, P.J. Java: como programar. Tradução Edson Furmankiewicz; revisão técnica Fábio Lucchini. – 6ª ed. – São Paulo. Pearson Prentice Hall, 2005

DESMET, L.; PIESSENS, F.; JOOSEN, W. VERBAETEN, P. Static verification of indirect data sharing in loosely-coupled component systems. **Lecture Notes in Computer Science**. V. 4089 p. 34-49, 2006.

EHLERS, E. **Agricultura sustentável: origens e perspectivas de um novo paradigma**. São Paulo: Livros da Terra, 1996.

ESPINOSA, H.R.M. Development and environment under a new optics. Environment 1993; In: SEIFFERT, M.A.B.; LOCH C. Systemic thinking in environmental management: suport for sustainable development. Journal of Cleaner Production v. 13 p. 1197–1202, 2005.

FAGNANI, M. A. **A Questão Ecológica na Formação do Engenheiro Agrícola**. 1997. 184f. Tese (Programa de Pós-Graduação em Educação). Universidade Estadual de Campinas, Faculdade de Educação, Campinas.

FARIA,R. A. **Treinamento avançado em XML**. São Paulo. Digerati Books, 2005.

FIELDS, D. K.; KOLB, M. A.; BAYERN, S. Web development with JavaServer Pages. 2. ed. Greenwich, CT: Manning Publications Co., 2002.

FNP Consultoria & AgroInformativos. Agriannual 2004. Anuário da agricultura brasileira. São Paulo. 2004

FURGERI, S. **Ensino Didático da Linguagem XML**. São Paulo: Érica, 2001.

HANSEN, J. W. Is Agricultural Sustainability a Useful Concept? **Agricultural Systems**. V. 50 p. 117-143, 1996

KAMIYA, D. S. **Análise emergética on-line para diagnóstico de sistemas agrícolas**. 2005. 163p. Dissertação (Mestrado em Engenharia de Alimentos). Universidade Estadual de Campinas, Faculdade de Engenharia de Alimentos, Campinas.

KAY, M. **XLST Programmer's Reference**. 2nd Edition. Acock's Green: Wrox Press, 2001.

KIRKEGAARD, C.; MOLLER, A; Static Analysis for Java Servlets and JSP. Lecture Notes in computer Science. V. 4134 p. 336-352, 2006

LEMAY, L.; CADENHEAD, R. **Sams teach yourself Java 2 in 21 days**. Indianapolis, Sams Publishing, 1999.

LI, X, Diversification and localization of energy systems for sustainable development and energy security. **Energy Policy**. v. 33 p. 2237-2243, 2005

LIU, C., Data flow analysis and testing of JSP-based Web applications. **Information and Software Technology**. V. 48 p. 1137-1147, 2006

LIU, X; CHEN, B. Efficiency and sustainability analysis of grain production in Jiangsu and Shaanxi Provinces of China. **Journal of Cleaner Production**. V. 15 p. 313-322, 2007

MARTIN, J. F.; DIEMONT, S. A. W.; POWEL, A.; STANTON, M.; LEVY-TACHER, S. Energy evaluation of the performance and sustainability of three agricultural systems with different scales and management. **Agriculture Ecosystems & Environment**. V.115 p. 128-140, 2006.

MELLO, T.; XU, L. Using XML to improve the productivity and robustness in application development in geosciences. **Computer & Geosciences**. V. 32 p. 1646-1653, 2006.

MORA, E. P.; Life cycle, sustainability and the transcendent quality of building materials. **Building and Environment** V. 42 p. 1329-1334, 2007.

ODUM, H.T. Self-Organization, Transformity, and Information. **Science** v.242, p. 1132-1139, 1988

ODUM, H.T. **Environmental Accounting: Energy and Environmental Decision Making**. John Wiley & Sons, Inc., New York, USA. 1996.

ODUM, H.T.; BROWN, M. T.; BRANDT-WILLIAMS, S. L. **Handbook of Energy Evaluation: A Compendium of Data for Energy Computation Issued In a Series of Folios**. Folio 1. Introduction and Global Budget. 2000. Center for Environmental Policy. University of Florida, Gainesville, Florida, USA.

ODUM, H.T.; An Energy Hierarchy Law for Biogeochemical Cycles. In: Energy Synthesis, ed. By M.T. Bown, Gainesville: Center for Environmental Policy. Univ. of Florida. Pág.235-247. 2001.

ORTEGA, E.; CAVALETT, O.; BONIFÁCIO, R.; WATANABE, M.; Brazilian soybean production: Emery analysis with an expanded scope. **Bulletin of Science, technology & society**. V.25 number 4 p.323–334, 2005

ORTEGA, E; MARCHIORI, F. Dianóstico emérgético da Agricultura da Flórida. 2006 Disponível em <<http://www.unicamp.br/fea/ortega/em-folios/index-frame.html>>. Acesso em 4 de janeiro 2007.

SACHS, I. **Ecodesenvolvimento**: crescer sem destruir. São Paulo: Vértice; 1986.

SALEH, K.; PROBERT, R.; KHANAFER, H. The distributed object computing paradigm: concepts and applications. **The Journal of Systems and Software**. V. 47, p. 125-131, 1999.

SILVA FILHO, A. M. da. Programando com XML. Rio de Janeiro: Elsevier, 2004

SOUZA, W. B. **Manual Básico de Struts**. 2002. Disponível em: <<http://www.jeebrasil.com.br/mostrat/30>>. Acesso em: 06 de dezembro 2006.

SUN MICROSYSTEMS. **JavaServer Pages Overview**. Disponível em: <<http://java.sun.com/products/jsp/overview.html>> . Acesso em: 14 de dezembro 2006.

TAKAHASHI, F; ZANGHETIN, M.; ORTEGA, E. **Use of Java application and XML database for emery evaluation of agricultural systems**. 5th Biennial International Workshop – Advances in energy studies. Porto Venere. 2006

WANG L.; ZHANG J.; NI W. Emery evaluation of Eco-Industrial Park with power plant. **Ecological Modelling**. v.189 p. 233-240, 2005

ZHANG, G.; WANG, P.; CHEN, W. Transaction of Web Services Based on Struts. **Lecture Notes in Computer Science**. V. 3756 p. 427-434, 2005.

ZHAO, S.; LI, Z.; LI, W. A modified method of ecological footprint calculation and its application. **Ecological Modelling**. V. 185 p. 65-75, 2005

Apêndice A – Unidades, fatores de correção e transformidades utilizados no sistema da Flórida

Cálculo do recurso sol (sun).

$$\text{Unidade de entrada} = \frac{J}{m^2 \text{ ano}}$$

$$\text{Fator de conversão} = \frac{10000m^2}{ha}$$

$$\text{Transformidade} = 1 \frac{sej}{J}$$

Cálculo do recurso chuva (rain)

$$\text{Unidade de entrada} = \frac{m^3}{m^2 \text{ ano}}$$

$$\text{Fator de conversão} = \frac{1 * 10^3 kg}{1m^3} * \frac{4940J}{1kg} * \frac{1 * 10^4 m^2}{ha} = \frac{4,94 * 10^{10} J}{m * ha}$$

$$\text{Transformidade} = 3,1 * 10^4 \frac{sej}{J} \text{ (ODUM et al., 2000)}$$

Cálculo do recurso evapotranspiração (evapotranspiration)

$$\text{Unidade de entrada} = \frac{J}{m^2 \text{ ano}}$$

$$\text{Fator de conversão} = \frac{10000m^2}{ha}$$

$$\text{Transformidade} = 25910,64 \frac{sej}{J} \text{ (ODUM, 1996) corrigido por Odum et al. (2000).}$$

Cálculo do recurso perda de solo (net topsoil loss)

$$\text{Unidade de entrada} = \frac{kg}{ha * \text{ano}}$$

$$\text{Energia do solo orgânico} \frac{5,4kcal}{g}$$

$$\text{Fator de conversão} = \frac{5,4kcal}{g} * \frac{1000g}{1kg} * \frac{4186J}{1kcal} = \frac{22604400J}{kg}$$

$$\text{Transformidade} = 123984 \frac{sej}{J} \text{ (ODUM, 1996) corrigido por Odum et al. (2000).}$$

Cálculo do recurso combustível (fuel)

$$\text{Unidade de entrada} = \frac{l}{ha * ano}$$

$$\text{Energia do combustível} = \frac{1,32 * 10^8 J}{gal\tilde{a}o}$$

$$\text{Fator de convers\~{a}o} = \frac{1 gal\tilde{a}o}{3,785 l} * \frac{1,32 * 10^8 J}{gal\tilde{a}o} = \frac{34874504,62 J}{l}$$

$$\text{Transformidade} = 110880 \frac{sej}{J} \text{ (ODUM, 1996) corrigido por Odum et al. (2000).}$$

Cálculo do recurso eletricidade (electricity)

$$\text{Unidade de entrada} = \frac{kWh}{ha * ano}$$

$$\text{Fator de convers\~{a}o} = \frac{3,6 * 10^6 J}{kWh}$$

$$\text{Transformidade} = 268800 \frac{sej}{J} \text{ (ODUM, 1996) corrigido por Odum et al. (2000).}$$

Cálculo do recurso cal (lime)

$$\text{Unidade de entrada} = \frac{kg}{ha * ano}$$

$$\text{Fator de convers\~{a}o} = 1$$

$$\text{Transformidade} = \frac{1,68 * 10^{12} sej}{kg} \text{ (ODUM, 1996) corrigido por Odum et al. (2000).}$$

Cálculo do recurso pesticidas (pesticides)

$$\text{Unidade de entrada} = \frac{kg}{ha * ano}$$

$$\text{Fator de convers\~{a}o} = 1$$

$$\text{Transformidade} = \frac{2,4864 * 10^{13} sej}{kg} \text{ (BROWN; ARDING, 1991) corrigido pelo fator 1,68 (ODUM et al., 2000).}$$

Cálculo do recurso potássio (potash)

$$\text{Unidade de entrada} = \frac{kg}{ha * ano}$$

$$\text{Fator de convers\~{a}o} = 1$$

$$\text{Transformidade} = \frac{2,9232 * 10^{12} sej}{kg} \text{ (ODUM, 1996) corrigido por Odum et al. (2000).}$$

Cálculo do recurso fosfato (phosphate)

$$\text{Unidade de entrada} = \frac{kg}{ha * ano}$$

$$\text{Fator de conversão} = 1$$

$$\text{Transformidade} = \frac{3,696 * 10^{13} \text{ sej}}{kg} \text{ (Brandt-Williams, 2002).}$$

Cálculo do recurso nitrogênio (nitrogen)

$$\text{Unidade de entrada} = \frac{kg}{ha * ano}$$

$$\text{Fator de conversão} = 1$$

$$\text{Transformidade} = \frac{4,0488 * 10^{13} \text{ sej}}{kg} \text{ (Brandt-Williams, 2002).}$$

Cálculo do recurso grãos (feed, grain)

$$\text{Unidade de entrada} = \frac{J}{ha * ano}$$

$$\text{Fator de conversão} = 1$$

$$\text{Transformidade} = \frac{1,43 * 10^5 \text{ sej}}{kg} \text{ (Brandt-Williams, 2002).}$$

Cálculo do recurso trabalho humano (human labor)

$$\text{Unidade de entrada} = \frac{\text{hora}}{ha * ano}$$

$$\text{Fator de conversão} = \frac{2500kcal}{dia} * \frac{4186J}{kcal} * \frac{dia}{8horas} = \frac{1308125J}{hora}$$

$$\text{Transformidade} = \frac{4,45 * 10^6 \text{ sej}}{J} \text{ (Brandt-Williams, 2002).}$$

Cálculo do recurso serviços da economia (services from economy)

$$\text{Unidade de entrada} = \frac{\text{dolar}}{ha * ano}$$

$$\text{Fator de conversão} = 1$$

$$\text{Transformidade} = \text{emdollar do ano de estudo.}$$

Apêndice B – Código do arquivo XSLT

```

<?xml version="1.0" encoding="ISO8859-1" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <xsl:variable name="emUS" select="/TABLE/SYSTEM/EMDOLLAR"/>
  <xsl:variable name="R1"
select="/TABLE/RENEWABLE[1]/QUANTITY*/(TABLE/RENEWABLE[1]/FACTOR)*(TABLE/RENE
WABLE[1]/TRANSFORMITY)/>
  <xsl:variable name="R2"
select="/TABLE/RENEWABLE[2]/QUANTITY*/(TABLE/RENEWABLE[2]/FACTOR)*(TABLE/RENE
WABLE[2]/TRANSFORMITY)/>
  <xsl:variable name="R3"
select="/TABLE/RENEWABLE[3]/QUANTITY*/(TABLE/RENEWABLE[3]/FACTOR)*(TABLE/RENE
WABLE[3]/TRANSFORMITY)/>
  <xsl:variable name="R" select="($R1+$R3) div 10000000000000"/>
  <xsl:variable name="N"
select="(/TABLE/NONRENEWABLE[1]/QUANTITY*/(TABLE/NONRENEWABLE[1]/FACTOR)*(TA
BLE/NONRENEWABLE[1]/TRANSFORMITY))div 10000000000000 "/>
  <xsl:variable name="I" select="$R + $N"/>
  <xsl:variable name="M1"
select="/TABLE/MATERIAL[1]/QUANTITY*/(TABLE/MATERIAL[1]/FACTOR)*(TABLE/MATERIA
L[1]/TRANSFORMITY)/>
  <xsl:variable name="M2"
select="/TABLE/MATERIAL[2]/QUANTITY*/(TABLE/MATERIAL[2]/FACTOR)*(TABLE/MATERIA
L[2]/TRANSFORMITY)/>
  <xsl:variable name="M3"
select="/TABLE/MATERIAL[3]/QUANTITY*/(TABLE/MATERIAL[3]/FACTOR)*(TABLE/MATERIA
L[3]/TRANSFORMITY)/>
  <xsl:variable name="M4"
select="/TABLE/MATERIAL[4]/QUANTITY*/(TABLE/MATERIAL[4]/FACTOR)*(TABLE/MATERIA
L[4]/TRANSFORMITY)/>
  <xsl:variable name="M5"
select="/TABLE/MATERIAL[5]/QUANTITY*/(TABLE/MATERIAL[5]/FACTOR)*(TABLE/MATERIA
L[5]/TRANSFORMITY)/>
  <xsl:variable name="M6"
select="/TABLE/MATERIAL[6]/QUANTITY*/(TABLE/MATERIAL[6]/FACTOR)*(TABLE/MATERIA
L[6]/TRANSFORMITY)/>
  <xsl:variable name="M7"
select="/TABLE/MATERIAL[7]/QUANTITY*/(TABLE/MATERIAL[7]/FACTOR)*(TABLE/MATERIA
L[7]/TRANSFORMITY)/>
  <xsl:variable name="M8"
select="/TABLE/MATERIAL[8]/QUANTITY*/(TABLE/MATERIAL[8]/FACTOR)*(TABLE/MATERIA
L[8]/TRANSFORMITY)/>
  <xsl:variable name="M9"
select="/TABLE/MATERIAL[9]/QUANTITY*/(TABLE/MATERIAL[9]/FACTOR)*(TABLE/MATERIA
L[9]/TRANSFORMITY)/>
  <xsl:variable name="M10"
select="/TABLE/MATERIAL[10]/QUANTITY*/(TABLE/MATERIAL[10]/FACTOR)*(TABLE/MATER
IAL[10]/TRANSFORMITY)/>

  <xsl:variable name="M" select="($M1+$M2+$M3+$M4+$M5+$M6+$M7+$M8+$M9+$M10) div
10000000000000"/>

```

```

<xsl:variable name="S1"
select="/TABLE/SERVICE[1]/QUANTITY)*(/TABLE/SERVICE[1]/FACTOR)*(/TABLE/SERVICE[1]/T
RANSFORMITY)"/>
<xsl:variable name="S2"
select="/TABLE/SERVICE[2]/QUANTITY)*(/TABLE/SERVICE[2]/FACTOR)*(/TABLE/SERVICE[2]/T
RANSFORMITY)"/>
<xsl:variable name="S" select="($S1+$S2) div 1000000000000000"/>
<xsl:variable name="F" select="$M + $S"/>
<xsl:variable name="Y" select="$I + $F"/>
<xsl:variable name="WM" select="/TABLE/PRODUCT[1]/VALUE"/>
<xsl:variable name="Moisture" select="/TABLE/PRODUCT[2]/VALUE div 100"/>
<xsl:variable name="DM" select="$WM * (1.0 - $Moisture)"/>
<xsl:variable name="Ech" select="/TABLE/PRODUCT[3]/VALUE) div 100 *
/TABLE/PRODUCT[3]/AD1"/>
<xsl:variable name="Efat" select="/TABLE/PRODUCT[4]/VALUE) div 100 *
/TABLE/PRODUCT[4]/AD1"/>
<xsl:variable name="Eprot" select="/TABLE/PRODUCT[5]/VALUE) div 100 *
/TABLE/PRODUCT[5]/AD1"/>
<xsl:variable name="Ep" select="$DM * ($Ech+$Efat+$Eprot) div 1000000000000000"/>
<xsl:variable name="Price" select="/TABLE/PRODUCT[6]/VALUE"/>
<xsl:variable name="Sales" select="$WM * $Price"/>
<xsl:variable name="EmSales" select="$emUS * $Sales div 1000000000000000"/>
<xsl:variable name="RealValue" select="$Y div $emUS * 1000000000000000"/>
<xsl:variable name="Tr" select="$Y div $Ep"/>
<xsl:variable name="Trm" select="$Y div $DM"/>
<xsl:variable name="Ren" select="$R div $Y"/>
<xsl:variable name="EYR" select="$Y div $F"/>
<xsl:variable name="EIR" select="$F div $I"/>
<xsl:variable name="EER" select="$Y *1000000000000000 div ($Sales * $emUS)"/>
<xsl:variable name="ELR" select="($F + $N) div $R"/>
<HTML><HEAD><TITLE>Emergency Flows Table</TITLE></HEAD>
<BODY bgcolor="#AAAAFF">
<CENTER>
<TABLE bgcolor="#EEEEFF" border="1" cellspacing="0" cellpadding="5">
<xsl:for-each select="TABLE/SYSTEM">
<TR bgcolor="#EEEEFF">
<TD>
<eml><xsl:value-of select="TITLE"/></eml><BR />
<xsl:value-of select="AUTHORS"/><BR />
<xsl:value-of select="INSTITUTION"/>
</TD>
<TD>
<xsl:element name="IMG">
<xsl:attribute name="SRC"><xsl:value-of select="TITLE/@foto"/></xsl:attribute>
<xsl:attribute name="height"><xsl:value-of select="60"/></xsl:attribute>
</xsl:element>
</TD>
<TD><span style="color: blue"><xsl:value-of select="OBJECTIVE"/></span><BR />
<span style="color: darkgreen">EmDollar: <xsl:value-of select="format-number($emUS div
1000000000000,##.00)"/> E12 sej/US$</span></TD>
</TR>
</xsl:for-each>
</TABLE>
<BR />

```

```

<TABLE bgcolor="#EEEEFF" border="0" cellspacing="1" cellpadding="2">
<TR bgcolor="#BBBCCC">
<TH>Sustainability</TH>
<TH>Proportion of resources</TH>
</TR>
<xsl:for-each select="/TABLE/GRAPHIC">
<TR>
<xsl:if test="position()=1">
<TD bgcolor="#EEEEFF" align="left"><xsl:value-of select="CONCEPT"/></TD>
</xsl:if>
<xsl:if test="position()=1">
<TD bgcolor="#EEEEFF" align="left">
<xsl:element name="IMG">
<xsl:attribute name="SRC"><xsl:value-of select="CONCEPT/@picture"/></xsl:attribute>
<xsl:attribute name="height"><xsl:value-of select="15"/></xsl:attribute>
<xsl:attribute name="width"><xsl:value-of select="400*$R div $Y"/></xsl:attribute>
</xsl:element>
<xsl:value-of select="format-number(100*$R div $Y, '#0.0')"/>%
</TD>
</xsl:if>
<xsl:if test="position()=2">
<TD bgcolor="#EEEEFF" align="left"><xsl:value-of select="CONCEPT"/></TD>
</xsl:if>
<xsl:if test="position()=2">
<TD bgcolor="#EEEEFF" align="left">
<xsl:element name="IMG">
<xsl:attribute name="SRC"><xsl:value-of select="CONCEPT/@picture"/></xsl:attribute>
<xsl:attribute name="height"><xsl:value-of select="15"/></xsl:attribute>
<xsl:attribute name="width"><xsl:value-of select="400*$N div $Y"/></xsl:attribute>
</xsl:element>
<xsl:value-of select="format-number(100*$N div $Y, '#0.0')"/>%
</TD>
</xsl:if>
<xsl:if test="position()=3">
<TD bgcolor="#EEEEFF" align="left"><xsl:value-of select="CONCEPT"/></TD>
<TD bgcolor="#EEEEFF" align="left">
<xsl:element name="IMG">
<xsl:attribute name="SRC"><xsl:value-of select="CONCEPT/@picture"/></xsl:attribute>
<xsl:attribute name="height"><xsl:value-of select="15"/></xsl:attribute>
<xsl:attribute name="width"><xsl:value-of select="400*$F div $Y"/></xsl:attribute>
</xsl:element>
<xsl:value-of select="format-number(100*$F div $Y, '#0.0')"/>%
</TD>
</xsl:if>
<xsl:if test="position()=4">
<TD bgcolor="#CCFFFF" align="left"><xsl:value-of select="CONCEPT"/></TD>
<TD bgcolor="#CCFFFF" align="left">
<xsl:element name="IMG">
<xsl:attribute name="SRC"><xsl:value-of select="CONCEPT/@picture"/></xsl:attribute>
<xsl:attribute name="height"><xsl:value-of select="15"/></xsl:attribute>
<xsl:attribute name="width"><xsl:value-of select="100*$EYR div $EIR"/></xsl:attribute>
</xsl:element>
<xsl:value-of select="format-number($EYR div $ELR, '#0.00')"/>
</TD>
</xsl:if>
</TR>

```



```

    <TD align="right"><xsl:value-of select="TRANSFORMITY"/></TD>
    <TD align="right"><xsl:value-of select="format-number((QUANTITY * FACTOR * TRANSFORMITY)
div 10000000000000, '#.0')"/></TD>
  </TR>
</xsl:for-each>
</TABLE>
<BR />
<TABLE bgcolor="#EEEEFF" border="0" cellspacing="1" cellpadding="2">
<TR bgcolor="#BBBCCC">
  <TH>Output data</TH>
  <TH></TH>
  <TH></TH>
  <TH>Caloric value</TH>
  <TH></TH>
</TR>
<xsl:for-each select="TABLE/PRODUCT">
<TR>
  <TD><xsl:value-of select="PROPERTY"/></TD>
  <TD bgcolor="FFFFCC" align="right"><b><xsl:value-of select="VALUE"/></b></TD>
  <TD align="left"><xsl:value-of select="UNITS"/></TD>
  <TD><xsl:value-of select="AD1"/></TD>
  <TD><xsl:value-of select="AD2"/></TD>
</TR>
</xsl:for-each>
</TABLE>
<BR />
<TABLE bgcolor="#EEEEFF" border="0" cellspacing="1" cellpadding="2">
<TR bgcolor="#BBBCCC">
  <TH>Inputs classification</TH>
  <TH>Equation</TH>
  <TH>Aggregated flows<BR />x E13 sej/ha/yr</TH>
</TR>
<xsl:for-each select="TABLE/AGGREGATED">
<TR>
  <TD><xsl:value-of select="NAME"/></TD>
  <TD><xsl:value-of select="FORMULA"/></TD>
  <xsl:if test="position()=1"> <TD bgcolor="#FFEFFF" align="right"><xsl:value-of select="format-
number($R , '#.00')"/></TD> </xsl:if>
  <xsl:if test="position()=2"> <TD bgcolor="#FFEFFF" align="right"><xsl:value-of select="format-
number($N , '#.00')"/></TD> </xsl:if>
  <xsl:if test="position()=3"> <TD bgcolor="#FFEFFF" align="right"><span style="color: red"><xsl:value-
of select="format-number($I , '#.00')"/></span></TD></xsl:if>
  <xsl:if test="position()=4"> <TD bgcolor="#FFEFFF" align="right"><xsl:value-of select="format-
number($M , '#.00')"/></TD> </xsl:if>
  <xsl:if test="position()=5"> <TD bgcolor="#FFEFFF" align="right"><xsl:value-of select="format-
number($S , '#.00')"/></TD> </xsl:if>
  <xsl:if test="position()=6"> <TD bgcolor="#FFEFFF" align="right"><span style="color: red"><xsl:value-
of select="format-number($F , '#.00')"/></span></TD></xsl:if>
  <xsl:if test="position()=7"> <TD bgcolor="#FFEFFF" align="right"><span style="color:
black"><b><xsl:value-of select="format-number($Y , '#.00')"/></b></span></TD></xsl:if>
</TR>
</xsl:for-each>
</TABLE>
<BR />
<TABLE bgcolor="#EEEEFF" border="0" cellspacing="1" cellpadding="4">
<TR bgcolor="#BBBCCC">

```

```

    <TH>Output results <span style="color: darkblue"><xsl:value-of
select="/TABLE/SYSTEM/OBJECTIVE"/></span></TH>
    <TH>Value</TH>
    <TH>Units</TH>
</TR>
<xsl:for-each select="TABLE/RESULT">
<TR>
<TD><xsl:value-of select="PROPERTY"/></TD>
<xsl:if test="position()=1"> <TD bgcolor="#FFFEFFF" align="right"><b><xsl:value-of select="format-
number($DM, '##.##')"/></b></TD> </xsl:if>
<xsl:if test="position()=2"> <TD bgcolor="#FFFEFFF" align="right"><b><xsl:value-of select="format-
number($Ep, '##.00')"/></b></TD> </xsl:if>
<xsl:if test="position()=3"> <TD bgcolor="#FFFEFFF" align="right"><b><xsl:value-of select="format-
number($Sales, '##.00')"/></b></TD> </xsl:if>
<xsl:if test="position()=4"> <TD bgcolor="#FFFEFFF" align="right"><b><xsl:value-of select="format-
number($EmSales, '##.00')"/></b></TD> </xsl:if>
<xsl:if test="position()=5"> <TD bgcolor="#FFFEFFF" align="right"><b><xsl:value-of select="format-
number($RealValue, '##.00')"/></b></TD> </xsl:if>
<TD align="left"><xsl:value-of select="UNITS"/></TD>
</TR>
</xsl:for-each>
</TABLE>
<BR />
<TABLE bgcolor="#EEEEFFF" border="0" cellspacing="1" cellpadding="3">
<TR bgcolor="#BBBCCCC">
<TH>Emergy Indices</TH>
<TH>Equation</TH>
<TH>Value</TH>
</TR>
<xsl:for-each select="TABLE/INDICE">
<TR>
<TD><xsl:value-of select="NAME"/></TD>
<TD><xsl:value-of select="FORMULA"/></TD>
<xsl:if test="position()=1"><TD bgcolor="#FFFEFFF" align="right"><b><xsl:value-of select="format-
number($Tr * 1000, '#0.00')"/></b></TD> </xsl:if>
<xsl:if test="position()=2"><TD bgcolor="#FFFEFFF" align="right"><b><xsl:value-of select="format-
number($Trm * 10, '#0.00')"/></b> E12</TD> </xsl:if>
<xsl:if test="position()=3"><TD bgcolor="#FFFEFFF" align="right"><b><xsl:value-of select="format-
number($Ren, '#0.00%')"/></b></TD> </xsl:if>
<xsl:if test="position()=4"><TD bgcolor="#FFFEFFF" align="right"><b><xsl:value-of select="format-
number($EYR, '#0.00')"/></b></TD> </xsl:if>
<xsl:if test="position()=5"><TD bgcolor="#FFFEFFF" align="right"><b><xsl:value-of select="format-
number($EIR, '#0.00')"/></b></TD> </xsl:if>
<xsl:if test="position()=6"><TD bgcolor="#FFFEFFF" align="right"><b><xsl:value-of select="format-
number($EER, '#0.00')"/></b></TD> </xsl:if>
</TR>
</xsl:for-each>
</TABLE>
<BR />
</CENTER></BODY></HTML>
</xsl:template>
</xsl:stylesheet>

```

Apêndice C – Arquivo DTD dos arquivos XML.

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!ELEMENT AD2 ( #PCDATA ) >
<!ELEMENT SERVICE ( CODE, ITEM, QUANTITY, UNITS, FACTOR, TRANSFORMITY, EMERGY )>
<!ELEMENT AD1 ( #PCDATA ) >
<!ELEMENT FORMULA ( #PCDATA ) >
<!ELEMENT GRAPHIC ( CONCEPT, VALUE ) >
<!ELEMENT MATERIAL ( CODE, ITEM, QUANTITY, UNITS, FACTOR, TRANSFORMITY,
EMERGY)>
<!ELEMENT EMDOLLAR ( #PCDATA ) >
<!ELEMENT PRODUCT ( PROPERTY, VALUE, UNITS, AD1, AD2 ) >
<!ATTLIST PRODUCT line NMTOKEN #REQUIRED >
<!ELEMENT INSTITUTION ( #PCDATA ) >
<!ELEMENT ITEM ( #PCDATA ) >
<!ELEMENT EMERGY ( #PCDATA ) >
<!ELEMENT VALUE ( #PCDATA ) >
<!ELEMENT UNITS ( #PCDATA ) >
<!ELEMENT NAME ( #PCDATA ) >
<!ELEMENT CODE ( #PCDATA ) >
<!ELEMENT RENEWABLE ( CODE, ITEM, QUANTITY, UNITS, FACTOR, TRANSFORMITY,
EMERGY ) >
<!ELEMENT TABLE ( SYSTEM, RENEWABLE+, NONRENEWABLE+, MATERIAL+, SERVICE+,
AGGREGATED+, PRODUCT+, RESULT+, INDICE+, GRAPHIC+ ) >
<!ELEMENT CONCEPT ( #PCDATA ) >
<!ATTLIST CONCEPT picture NMTOKEN #REQUIRED >
<!ELEMENT OBJECTIVE ( #PCDATA ) >
<!ELEMENT TITLE ( #PCDATA ) >
<!ATTLIST TITLE foto NMTOKEN #REQUIRED >
<!ELEMENT FACTOR ( #PCDATA ) >
<!ELEMENT SYSTEM ( TITLE, AUTHORS, INSTITUTION, OBJECTIVE, EMDOLLAR ) >
<!ELEMENT AGGREGATED ( NAME, FORMULA, VALUE ) >
<!ATTLIST AGGREGATED tipo NMTOKEN #IMPLIED >
<!ELEMENT INDICE ( NAME, FORMULA, VALUE ) >
<!ATTLIST INDICE tipo NMTOKEN #REQUIRED >
<!ELEMENT TRANSFORMITY ( #PCDATA ) >
<!ELEMENT NONRENEWABLE ( CODE, ITEM, QUANTITY, UNITS, FACTOR, TRANSFORMITY,
EMERGY ) >
<!ELEMENT PROPERTY ( #PCDATA ) >
<!ELEMENT RESULT ( PROPERTY, VALUE, UNITS ) >
<!ATTLIST RESULT line NMTOKEN #REQUIRED >
<!ELEMENT QUANTITY ( #PCDATA ) >
<!ELEMENT AUTHORS ( #PCDATA ) >
```

Apêndice D – Código do arquivo CampoActionForm.java

<pre> package emergytable; import javax.servlet.http.*; import org.apache.struts.action.*; public class recursosActionForm extends ActionForm { public recursosActionForm() { try { jbInit(); } catch (Exception ex) { ex.printStackTrace(); } } private String electricity; private String evapotranspiration; private String feed; private String foto; private String fuel; private String humanLabor; private String identificador; private String lime; private String livestock; private String machinery; private String netTopsoilLoss; private String nitrogen; private String nome; private String pesticides; private String phosphate; private String potash; private String rain; private String servicesFromEconomy; private String sun; private String mass; private String moisture; private String carbohydrates; private String lipids; private String protein; private String price; private String data; private int id; public String getElectricity() { return electricity; </pre>	<pre> public void setElectricity(String electricity) { this.electricity = electricity; } public void setSun(String sun) { this.sun = sun; } public void setServicesFromEconomy(String servicesFromEconomy) { this.servicesFromEconomy = servicesFromEconomy; } public void setRain(String rain) { this.rain = rain; } public void setPotash(String potash) { this.potash = potash; } public void setPhosphate(String phosphate) { this.phosphate = phosphate; } public void setPesticides(String pesticides) { this.pesticides = pesticides; } public void setNome(String nome) { this.nome = nome; } public void setNitrogen(String nitrogen) { this.nitrogen = nitrogen; } public void setNetTopsoilLoss(String netTopsoilLoss) { this.netTopsoilLoss = netTopsoilLoss; } public void setMachinery(String machinery) { this.machinery = machinery; } public void setLivestock(String livestock) { this.livestock = livestock; } public void setLime(String lime) { this.lime = lime; } public void setIdentificador(String identificador) { this.identificador = identificador; } public void setHumanLabor(String humanLabor) { this.humanLabor = humanLabor; } public void setFuel(String fuel) { this.fuel = fuel; } public void setFoto(String foto) { this.foto = foto; } public void setFeed(String feed) { </pre>
--	---

<pre> this.feed = feed; } public void setEvapotranspiration(String evapotranspiration) { this.evapotranspiration = evapotranspiration; } public void setMass(String mass) { this.mass = mass; } public void setMoisture(String moisture) { this.moisture = moisture; } public void setCarbohydrates(String carbohydrates) { this.carbohydrates = carbohydrates; } public void setLipids(String lipids) { this.lipids = lipids; } public void setProtein(String protein) { this.protein = protein; } public void setPrice(String price) { this.price = price; } public void setData(String data) { this.data = data; } public void setId(int id) { this.id = id; } public String getEvapotranspiration() { return evapotranspiration; } public String getFeed() { return feed; } public String getFoto() { return foto; } public String getFuel() { return fuel; } public String getHumanLabor() { return humanLabor; } public String getIdentificador() { return identificador; } public String getLime() { return lime; } public String getLivestock() { return livestock; } public String getMachinery() { return machinery; } public String getNetTopsoilLoss() { return netTopsoilLoss; } </pre>	<pre> public String getNitrogen() { return nitrogen; } public String getNome() { return nome; } public String getPesticides() { return pesticides; } public String getPhosphate() { return phosphate; } public String getPotash() { return potash; } public String getRain() { return rain; } public String getServicesFromEconomy() { return servicesFromEconomy; } public String getSun() { return sun; } public String getMass() { return mass; } public String getMoisture() { return moisture; } public String getCarbohydrates() { return carbohydrates; } public String getLipids() { return lipids; } public String getProtein() { return protein; } public String getPrice() { return price; } public String getData() { return data; } public int getId() { return id; } public ActionErrors validate(ActionMapping actionMapping, HttpServletRequest httpRequest) { return null; } public void reset(ActionMapping actionMapping, HttpServletRequest servletRequest) { } private void jbInit() throws Exception { } } </pre>
--	---