
Ontologias e DSLs na geração de sistemas de apoio à decisão, caso de estudo SustenAgro

John Freddy Garavito Suárez

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

John Freddy Garavito Suárez

Ontologias e DSLs na geração de sistemas de apoio à decisão, caso de estudo SustenAgro

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional.

VERSÃO REVISADA

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Dilvan de Abreu Moreira

USP – São Carlos
Junho de 2017

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

L965c Garavito S., John Freddy
 Ontologias e DSLs na Geração de Sistemas de
Apoio à Decisão, caso de estudo SustenAgro /
John Freddy Garavito S.; orientador Dilvan de
Abreu Moreira. - São Carlos, 2017.
 139 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2017.

1. SAD. 2. Web Semântica e DSLs. 3. Framework
Decisioner. 4. SustenAgro. 5. SAD SustenAgro
6. Avaliação. I. Moreira, Dilvan de Abreu , orient.

John Freddy Garavito Suárez

**Ontologies and DSLs in the generation of
decision support systems,
SustenAgro study case**

Master dissertation submitted to the Instituto de
Ciências Matemáticas e de Computação - ICMC-
USP, in partial fulfillment of the requirements for the
degree of the Master Program in Computer Science
And Computational Mathematics.

FINAL VERSION

Concentration Area: Computer Science and
Computational Mathematics

Advisor Prof. Dr. Dilvan de Abreu Moreira

**USP – São Carlos
June 2017**

***Aos meus pais, Salustriano e Dioselina pelo
amor compartilhado.***

Agradecimentos.

Agradeço a Deus pela vida,

Aos meus pais e família pelo excelente exemplo de vida e apoio,

A Mayra Bonfim pela agradável companhia, ótimos conselhos e apoio incondicional, que me permitiram crescer como pessoa e profissional.

Ao professor Dr. Dilvan Moreira pela orientação, pelo compartilhamento de sabedoria, conhecimento, tempo, conversas e risadas, e especialmente pela paciência e apoio durante cada uma das etapas do mestrado.

Aos meus colegas do laboratório Intermídia
pelo exemplo de perseverança.

A Universidade de São Paulo
pelo ensino.

Agradecimentos técnicos.

Um especial agradecimento à Dra. Katia Regina Evaristo de Jesus, pesquisadora da Embrapa Meio Ambiente, coordenadora do Projeto: ‘Avaliação da Sustentabilidade de Sistemas de Produção da Cana-de-açúcar e Soja na Região Centro-Sul do Brasil: Uma proposta metodológica e de modelo conceitual SustenAgro’, financiado pela Embrapa. Os dados técnicos empregados no desenvolvimento do Software foram desenvolvidos no escopo do projeto supra citado.

Também um especial agradecimento ao pesquisador Sérgio Alves Torquato da Agência Paulista de Tecnologia do Agronegócio – APTA/ UPD Tietê/SP.

***“Live as if you were to die tomorrow.
Learn as if you were to live forever.”***

***“Viva como se você fosse morrer amanhã.
Aprenda como se você fosse viver para sempre.”***

***“Vive como si fueras a morir mañana.
Aprende como si fueras a vivir siempre”***

Resumo

GARAVITO, J. F. Ontologias e DSLs na geração de sistemas de apoio à decisão, caso de estudo SustenAgro. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

Os Sistemas de Apoio à Decisão (SAD) organizam e processam dados e informações para gerar resultados que apoiem a tomada de decisão em um domínio específico. Eles integram conhecimento de especialistas de domínio em cada um de seus componentes: modelos, dados, operações matemáticas (que processam os dados) e resultado de análises. Nas metodologias de desenvolvimento tradicionais, esse conhecimento deve ser interpretado e usado por desenvolvedores de software para implementar os SADs. Isso porque especialistas de domínio não conseguem formalizar esse conhecimento em um modelo computável que possa ser integrado aos SADs. O processo de modelagem de conhecimento é realizado, na prática, pelos desenvolvedores, parcializando o conhecimento do domínio e dificultando o desenvolvimento ágil dos SADs (já que os especialistas não modificam o código diretamente). Para solucionar esse problema, propõe-se um método e ferramenta web que usa ontologias, na *Web Ontology Language* (OWL), para representar o conhecimento de especialistas, e uma *Domain Specific Language* (DSL), para modelar o comportamento dos SADs. Ontologias, em OWL, são uma representação de conhecimento computável, que permite definir SADs em um formato entendível e acessível a humanos e máquinas. Esse método foi usado para criar o *Framework Decisioner* para a instanciação de SADs. O *Decisioner* gera automaticamente SADs a partir de uma ontologia e uma descrição na DSL, incluindo a interface do SAD (usando uma biblioteca de *Web Components*). Um editor online de ontologias, que usa um formato simplificado, permite que especialistas de domínio possam modificar aspectos da ontologia e imediatamente ver as consequência de suas mudanças no SAD. Uma validação desse método foi realizada, por meio da instanciação do SAD SustenAgro no *Framework Decisioner*. O SAD SustenAgro avalia a sustentabilidade de sistemas produtivos de cana-de-açúcar na região centro-sul do Brasil. Avaliações, conduzidas por especialistas em sustentabilidade da Embrapa Meio ambiente (parceiros neste projeto), mostraram que especialistas são capazes de alterar a ontologia e DSL usadas, sem a ajuda de programadores, e que o sistema produz análises de sustentabilidade corretas.

Palavras Chave: *Ontologias, Linguagem de Domínio Específico, Web Semântica, Representação de Conhecimento, Framework Decisioner, Sistema de apoio à decisão, SustenAgro*

Abstract

GARAVITO, J. F. Ontologies and DSLs in the generation of decision support systems, SustenAgro study case. Master dissertation (Master Program in Computer Science And Computational Mathematics.) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

Decision Support Systems (DSSs) organize and process data and information to generate results to support decision making in a specific domain. They integrate knowledge from domain experts in each of their components: models, data, mathematical operations (that process the data) and analysis results. In traditional development methodologies, this knowledge must be interpreted and used by software developers to implement DSSs. That is because domain experts cannot formalize this knowledge in a computable model that can be integrated into DSSs. The knowledge modeling process is carried out, in practice, by the developers, biasing domain knowledge and hindering the agile development of DSSs (as domain experts cannot modify code directly). To solve this problem, a method and web tool is proposed that uses ontologies, in the Web Ontology Language (OWL), to represent expert's knowledge, and a Domain Specific Language (DSL), to model DSS behavior. Ontologies, in OWL, are a computable knowledge representations, which allow the definition of DSSs in a format understandable and accessible to humans and machines. This method was used to create the Decisioner Framework for the instantiation of DSSs. Decisioner automatically generates DSSs from an ontology and a description in its DSL, including the DSS interface (using a Web Components library). An online ontology editor, using a simplified format, allows that domain experts change the ontology and immediately see the consequences of their changes in the in the DSS. A validation of this method was done through the instantiation of the SustenAgro DSS, using the Decisioner Framework. The SustenAgro DSS evaluates the sustainability of sugarcane production systems in the center-south region of Brazil. Evaluations, done by by sustainability experts from Embrapa Environment (partners in this project), showed that domain experts are capable of changing the ontology and DSL program used, without the help of software developers, and that the system produced correct sustainability analysis.

Keywords: *Ontologies, Domain-Specific Language, Semantic Web, Knowledge Representation, Decisioner Framework, Decision Support System, SustenAgro*

Lista de ilustrações

Figura 1 – Componentes de um SAD.	36
Figura 2 – Arquitetura inicial do SAD SustenAgro	39
Figura 3 – História da <i>Web Semântica</i>	46
Figura 4 – Arquitetura em camadas da <i>Web Semântica</i>	47
Figura 5 – <i>Smart data continuum</i> : níveis de representação de dados na forma de conhecimento processável por máquinas.	49
Figura 6 – OWL2 Profiles.	51
Figura 7 – Arquitetura do Decisioner	58
Figura 8 – Metodologia de desenvolvimento do Decisioner	60
Figura 9 – Modelagem abstrata do SAD	61
Figura 10 – Editor da ontologia SustenAgro	63
Figura 11 – Widget gerada a partir da definição da DSL	65
Figura 12 – Editor da DSL	68
Figura 13 – Web UI com web components	70
Figura 14 – Arquitetura do <i>DSL Interpreter</i>	71
Figura 15 – Metodologia de definição da ontologia SustenAgro.	75
Figura 16 – Mapa conceitual da ontologia SustenAgro.	77
Figura 17 – Modelagem da classe de unidade produtiva (<i>ProductionUnit</i>).	79
Figura 18 – Modelagem de Value	80
Figura 19 – Modelagem de microrregiões.	81
Figura 20 – Modelagem de indicador	82
Figura 21 – Mapa conceitual - Método de Avaliação.	84
Figura 22 – <i>Storyboard</i> definição da localização.	89
Figura 23 – <i>Storyboard</i> seleção da unidade produtiva.	89
Figura 24 – <i>Storyboard</i> mostrando o preenchimento dos indicadores.	90
Figura 25 – <i>Storyboard</i> sobre a avaliação de unidade produtiva	90
Figura 26 – <i>Storyboard</i> para cadastro de novo indicador.	90
Figura 27 – <i>Storyboard</i> mostrando a apresentação de resultados	91
Figura 28 – <i>Mockup</i> da tela inicial do SustenAgro.	92
Figura 29 – <i>Mockup</i> da tela de indicadores do SustenAgro.	93
Figura 30 – Protótipo do SustenAgro – Tela inicial.	94
Figura 31 – Cadastro de indicadores	95
Figura 32 – Planilhas do resultado da avaliação	96

Figura 33 – Matriz de sustentabilidade mostrando um índice de sustentabilidade de 2,3 e um índice de eficiência de 2,4.	97
Figura 34 – Semáforo de Sustentabilidade mostrando um índice de sustentabilidade geral de 4 numa escala de -100 até +100.	98
Figura 35 – Matriz de Sustentabilidade com valores mínimos.	107
Figura 36 – Dimensões da sustentabilidade	127

Lista De Códigos-Fonte

4.1	Definição do Evaluation Object	64
4.2	Definição de Features	65
4.3	Definição da lógica de avaliação.	66
4.4	Definição dos componentes visuais do relatório.	67
5.1	DSL que define o Evaluation Object	99
5.2	Definição de Features	100
C.1	Instalação do Java	131
C.2	Instalação do Apache Tomcat	132
C.3	Configuração dos Users	132
C.4	Exportar variáveis de entorno	132
C.5	Iniciar o servidor	132
C.6	Instalação do wkhtmltopdf	133
C.7	Execução do wkhtmltopdf	133
C.8	Instalação do Blazegraph	133
C.9	Execução do sustenagro	134
D.1	Código de novo indicador	137
D.2	Adição de feature na DSL	138

Lista de tabelas

Tabela 1 – Avaliação dos usuários finais do SAD SustenAgro	108
Tabela 2 – Avaliação dos especialistas do SAD SustenAgro	109

Lista De Abreviaturas e Siglas

AGROVOC Agricultural vocabulary

APTA Agência Paulista de Tecnologia dos Agronegócios

CEPAL Economic Commission for Latin America and the Caribbean

CSS Cascading Style Sheets

DSL Domain Specific Language

HTML HyperText Markup Language

ICMC Instituto de Ciências Matemáticas e de Computação

KOS Knowledge Organization System

LOD Linked Open Data

OWL Web Ontology Language

PDF Portable Document Format

PNUMA Programa das Nações Unidas para o Meio Ambiente

RDF Resource Description Framework

SAD Sistema de Apoio à Decisão

SPARQL SPARQL Protocol and RDF Query Language

UI User Interface

UML Unified Modeling Language

URI Uniform Resource Identifier

W3C World Wide Web Consortium

WAR Web Application Archive

XML Extensible Markup Language

YAML YAML Ain't Markup Language

Sumário

Lista de ilustrações	17
Lista de tabelas	21
Sumário	25
1 INTRODUÇÃO	29
1.1 Motivação	30
1.2 Objetivo	31
1.3 Resultados principais	32
1.4 Organização da dissertação	32
2 SAD	35
2.1 Definição de Sistema de Apoio à Decisão	35
2.2 Arquitetura para Sistemas de Apoio à Decisão	36
2.3 Taxonomia dos Sistemas de Apoio à Decisão	37
2.4 Sistema de Apoio à Decisão SustenAgro	38
2.5 Trabalhos relacionados	41
2.6 Considerações finais	43
3 WEB SEMÂNTICA E DSLS	45
3.1 Web Semântica	45
3.2 Linguagem de domínio específico	54
3.3 Considerações finais	56
4 FRAMEWORK DECISIONER	57
4.1 Arquitetura do Framework Decisioner	57
4.2 Metodologia	59
4.3 Ontologia Decisioner	60
4.4 Editor de Ontologias	62
4.5 DSL do Framework Decisioner	63
4.6 Editor da DSL	67
4.7 Web Components	68
4.8 Web UI	69

4.9	DSL Interpreter	70
4.10	Considerações finais	71
5	SAD SUSTENAGRO	73
5.1	Arquitetura do SustenAgro	73
5.2	Metodologia	74
5.3	Ontologia de domínio: SustenAgro	76
5.4	SustenAgro Web UI	84
5.5	<i>Web Components</i>	96
5.6	DSL: code	99
5.7	Considerações Finais	101
6	AVALIAÇÃO	103
6.1	Avaliação das <i>Web UI</i>	103
6.2	Avaliação da ontologia de domínio de avaliação da sustentabilidade	104
6.3	Avaliação do protótipo funcional com dados	105
6.4	Avaliação do SAD SustenAgro e do <i>Framework Decisioner</i>	107
6.5	<i>Workshop</i> : validação do <i>software</i> SustenAgro v1.0 com equipe de especialistas	110
6.6	Avaliação de SAD SustenAgro nos servidores da Embrapa	111
6.7	Conclusões	112
7	CONCLUSÕES	113
7.1	Resultados	114
7.2	Dificuldades	115
7.3	Trabalhos futuros	116
7.4	Agradecimentos	118
	REFERÊNCIAS	119
A	MÉTODO SUSTENAGRO DE AVALIAÇÃO DE SUSTENTABI- LIDADE	125
A.1	Sustentabilidade	125
A.2	Dimensões da sustentabilidade	126
A.3	Critérios de sustentabilidade	127
A.4	Método SustenAgro	128
A.5	Conclusões	128
B	INDICADORES DE SUSTENTABILIDADE	129
B.1	Índice de Sustentabilidade	129
B.2	Indicadores de Sustentabilidade	129

B.3	Dados fornecidos pela Embrapa Meio Ambiente	130
B.4	Considerações finais	130
C	INSTALAÇÃO	131
C.1	Configuração do servidor	131
C.2	<i>Deploy</i> do arquivo WAR	134
D	FORMULÁRIOS DE AVALIAÇÃO	135
D.1	Formulário de avaliação do SAD SustenAgro	135
D.2	Formulário de avaliação do <i>Framework Decisioner</i>	136
D.3	Formulário <i>Delphi</i> para <i>Workshop</i> SustenAgro	139

Introdução

Os Sistemas de Apoio à Decisão (SAD) organizam e processam os dados e informações para gerar resultados de valor que apoiem e melhorem a tomada de decisão em um área de conhecimento, também denominada como domínio específico (TURBAN; ARONSON; LIANG, 2004). Os SADs integram conhecimento desenvolvido pelos especialistas do domínio que fica implícito nos dados, informações e processos usados durante o processo de decisão (POWER, 2002). Tal conhecimento, específico de um domínio, não é familiar para desenvolvedores de software. Eles têm que usar técnicas diversas para o levantamento de requisitos para entender o domínio dos especialistas e assim implementar o *software* corretamente (GAVRILOVA; ANDREEVA, 2012). Quanto mais especializado for o domínio dos especialistas, mais esforço adicional será necessário aos desenvolvedores do sistema, o que leva a ampliação do tempo e custo de desenvolvimento.

Adicionalmente, os especialistas do domínio, em geral, não têm conhecimento suficiente em matéria de desenvolvimento de sistemas de *software* para realizar o processo por eles mesmos. Além disso, os dois domínios, tanto dos especialistas de domínio como dos desenvolvedores de *software*, são tão amplos que precisam de perfis particulares para realizar os processos corretamente (ROUSSEY et al., 2010). Dentro deste contexto, foi identificado o problema que os especialistas do domínio não tinham um meio de representação de conhecimento para definir SADs, que tenha um formato computável, entendível e acessível.

Como exemplo do problema anterior, podemos expor o caso dos especialistas em sustentabilidade da Embrapa Meio Ambiente, que desenvolveram o projeto SustenAgro (Apêndice A). Nesse projeto, foi desenvolvido um método de avaliação de sustentabilidade no sistema produtivo da cana-de-açúcar do centro-sul do Brasil (CARDOSO, 2013). Tais especialistas precisavam implementar um SAD para disponibilizar o método SustenAgro à comunidade interessada em realizar avaliações de sustentabilidade em cana-de-açúcar. Nesse caso, foi identificado que eles possuíam o conhecimento do domínio de avaliação de sustentabilidade da cana-de-açúcar, mas não tinham um meio para definir esse conhecimento de

maneira computável em um SAD.

1.1 Motivação

A pesquisa em representação e organização de conhecimento tem alto impacto devido ao fato de fornecer métodos e ferramentas para entender e gerenciar o conhecimento em diversos domínios (TUDHOPE; NIELSEN, 2006). Especificamente nos SADs, ela pode fornecer meios de integração de conhecimento que aumentam as funcionalidades e a eficiência desses sistemas, inclusive trazendo vantagens no processo de desenvolvimento dos SAD (SAXENA, 1991).

A análise do projeto SustenAgro permitiu identificar as seguintes motivações relacionadas com a definição de conhecimento dos especialistas de domínio:

A primeira delas é a necessidade de um método e ferramenta para que os especialistas do domínio definam o conhecimento deles nos SADs, principalmente as características particulares que requerem profundo conhecimento. Isso permite a participação deles como descritores de conhecimento específico fornecendo, aos desenvolvedores de *software*, tempo adicional para dedicar-se aos assuntos próprios da computação e assim agilizar o processo de desenvolvimento de SADs. Essa abordagem facilita a definição de SADs com menos intervenção por parte dos não especialistas do domínio, fazendo que a definição do conhecimento fique em termos conhecidos pelos especialistas e seja gerenciada por eles mesmos.

A segunda refere-se a fornecer meios computáveis de representação desse conhecimento, que facilitem a comunicação entre os especialistas do domínio e os desenvolvedores de software e que adaptem-se às mudanças do domínio. Domínios de conhecimento estão em contínua mudança, como é o caso do domínio avaliação da sustentabilidade da cultura de cana-de-açúcar (CARDOSO, 2013).

A terceira é o impacto que gera o correto uso dos SAD (LEE; WAGNER; SHIN, 2008). O desenvolvimento do SAD SustenAgro, em cooperação com a Embrapa Meio Ambiente, poderia gerar impacto positivo na produção agrícola, já que ele fornece a funcionalidade de avaliar a sustentabilidade na cana-de-açúcar. Este SAD em particular é uma contribuição inédita que também permite a aplicação em outros sistemas agrícolas, o que poderia suportar melhoras e correções na produção de alimentos e produtos neste setor (MATTHEWS et al., 2008).

Para definir um meio computável de representação de conhecimento, foram analisados vários tipos de sistemas de organização de conhecimento ou *Knowledge Organization System* (KOS). Um dos KOS mais completos são as ontologias porque permitem definir, classificar, relacionar e inferir conhecimento. Por este motivo as ontologias foram selecionadas para representar a modelagem do domínio dos especialistas.

Numa parceria com a Embrapa Meio Ambiente, foi desenvolvida uma ontologia de

avaliação da sustentabilidade em cana-de-açúcar junto com os especialistas desta instituição. Nas áreas de biologia e ecologia as ontologias são comuns, desta forma eles não tiveram dificuldade com o conceito. Eles usaram uma *Domain Specific Language* (DSL) (FOWLER, 2010) que permite formatar as perguntas aos usuários finais, definir o método de avaliação e gerar o relatório de cada análise. Os especialistas apreciaram muito a utilização de uma ontologia em um SAD, foi a primeira vez que fizeram isso, e particularmente, o fato de que as mudanças na ontologia, façam mudanças imediatamente nos componentes do SAD.

Além das motivações vinculadas com o domínio dos especialistas em sustentabilidade existe a motivação de desenvolver novas tecnologias. Uma contribuição tecnologia neste sentido na pesquisa foi o *Framework Decisioner* composto por ontologias e DSLs, cujo propósito é facilitar a implementação de SADs, sendo este *framework* instanciado para produzir o SAD SustenAgro.

1.2 Objetivo

Desenvolver um método e ferramenta *web*, baseados em ontologias, que permitam representar o conhecimento de especialistas do domínio para suportar a definição de SADs.

Objetivos específicos

Para alcançar o objetivo proposto, foi necessário cumprir os seguintes objetivos específicos:

- Definir a arquitetura e o código de uma ferramenta que permita gerar SADs baseados em conhecimento de domínios específicos. Os SADs gerados usam uma ontologia, desenvolvida por especialistas do domínio e uma DSL, para gerar as interfaces e funcionalidades do sistema.
- Definir uma ontologia sobre avaliação da sustentabilidade nos sistemas produtivos de cana-de-açúcar do centro-sul do Brasil.
- Definir uma DSL que permita a definição da interface de usuário, das fórmulas do modelo, usado pelos especialistas, e do formato do relatório final de cada análise.
- Demonstrar que o método e ferramenta permitem a criação de SADs funcionais que podem ser modificados por especialistas de domínio. Esses especialistas devem definir a ontologia e modificar o comportamento do sistema com pouca ou nenhuma intervenção de desenvolvedores.

1.3 Resultados principais

As principais contribuições da pesquisa foram:

- Método e *Framework Decisioner* para definir SADs por parte dos especialistas do domínio, baseado do *Framework Grails*.
- O SAD SustenAgro: sistema de apoio à decisão para suportar a avaliação da sustentabilidade em cana-de-açúcar, na região centro-sul do Brasil, instanciado por meio do *Framework Decisioner*.
- Ontologia SustenAgro sobre avaliação de sustentabilidade em cana-de-açúcar, representando os principais conceitos desse domínio: indicadores, índices e métodos de avaliação. Ela foi desenvolvida em parceria com os especialistas de domínio de sustentabilidade da Embrapa.
- Ontologia sobre controles visuais para suportar a geração automática das interfaces gráficas do SAD SustenAgro. Ela permite associar tipos de dados aos *web components* que constituem as interfaces dos SADs.
- Uma DSL baseada na linguagem *Groovy* que permite a definição da interface de usuário, do formato do dados a processar, das fórmulas do modelo e do formato do relatório final de cada análise.
- Artigo (co-autor) “Sustainability assessment of sugarcane production systems: SustenAgro Method” submetido ao periódico da Elsevier “Energy for sustainable Development” ISSN: 0973-0826 (Artigo não aceito).
- Artigo (primeiro autor) “SustenAgro Sistema de Apoio à Decisão baseado em Ontologias e definido por uma Linguagem de Domínio Específico” submetido ao periódico “Revista Brasileira de Sistemas de Informação” ISSN Eletrônico: 1984-2902.
- Avaliações do *Framework Decisioner*, do método de geração de SADs e do Sistema SustenAgro que satisfizerem os objetivos da pesquisa.

1.4 Organização da dissertação

A presente dissertação está estruturada da seguinte forma:

Capítulo 2: Apresenta a definição de Sistemas de Apoio à Decisão, as características do SAD SustenAgro e os trabalhos relacionados.

Capítulo 3: Apresenta as ontologias da web semântica e DSLs, com a finalidade de descrever as principais tecnologias e a teoria necessária para desenvolver a presente pesquisa.

Capítulo 4: Apresenta o protótipo do *Framework Decisioner*, que suporta a geração de Sistemas de Apoio à Decisão.

Capítulo 5: Apresenta o SAD SustenAgro. Desenvolvido neste trabalho como caso de uso do *Framework Decisioner*. Ele permitiu o desenvolvimento da arquitetura do *SAD SustenAgro* e serviu para avaliar a funcionalidade do mesmo.

Capítulo 6: Apresenta a avaliação realizada pelos especialistas de domínio em cada um dos sistemas software desenvolvidos e no método de geração de SADs.

Capítulo 7: Apresenta as conclusões do presente trabalho, dificuldades, trabalhos futuros e uma discussão sobre a pesquisa em geral.

Ao final, são apresentados os anexos que descrevem conceitos de terceiros usados no trabalho e informações técnicas dos sistemas desenvolvidos.

A construção de sistemas que sejam capazes de fornecer suporte ao gestor em um processo de tomada de decisões tem sido um desafio ao longo dos anos. Um dos problemas principais, são as dificuldades relacionadas com a modelagem e o entendimento do conhecimento do domínio. Especialistas de domínio e desenvolvedores de software têm *Backgrounds* e culturas diferentes. Isso faz com que o processo de modelagem desses sistemas seja um processo, na maioria das vezes, lento e custoso, dificultando os processos de desenvolvimento e testes. Por esta razão, foram pesquisados modelos que facilitem o entendimento entre esses dois tipos de profissionais e que permitam uma maior liberdade e participação no desenvolvimento dos especialistas de domínio.

Neste capítulo serão abordados a definição e arquitetura de SADs, o projeto SustenAgro da Embrapa e os trabalhos relacionados.

2.1 Definição de Sistema de Apoio à Decisão

Os Sistemas de Apoio à Decisão (SAD) são uma área de conhecimento ampla e em contínua evolução. A definição de SAD deriva da definição de sistema, pois eles contêm um conjunto de partes organizadas para um propósito comum, existem múltiplas definições do termo SAD, a seguir serão apresentadas algumas definições que permitem explicar o propósito deste tipo de sistemas.

Tweedale, Phillips-Wren e Jain (2016) define os SADs como sistemas *software* que visam melhorar a tomada de decisão individual ou grupal, combinando o conhecimento do(s) tomador(es) de decisão com dados relevantes de fontes confiáveis, nos quais são aplicados métodos e modelos matemáticos para suportar a análise, comparação e escolha de alternativas no processo de decisão.

Heinzle, Gauthier e Fialho (2010) define que os SADs apoiam o entendimento de processos complexos, auxiliam na comparação dos fenômenos envolvidos e suportam a

análise e escolha de alternativas no processo de decisão. Este entendimento do domínio surge da combinação das habilidades e metodologias dos especialistas (humanos) à capacidade dos computadores de acessar dados, estruturá-los em modelos, interpretar, formular e avaliar alternativas e cenários distintos.

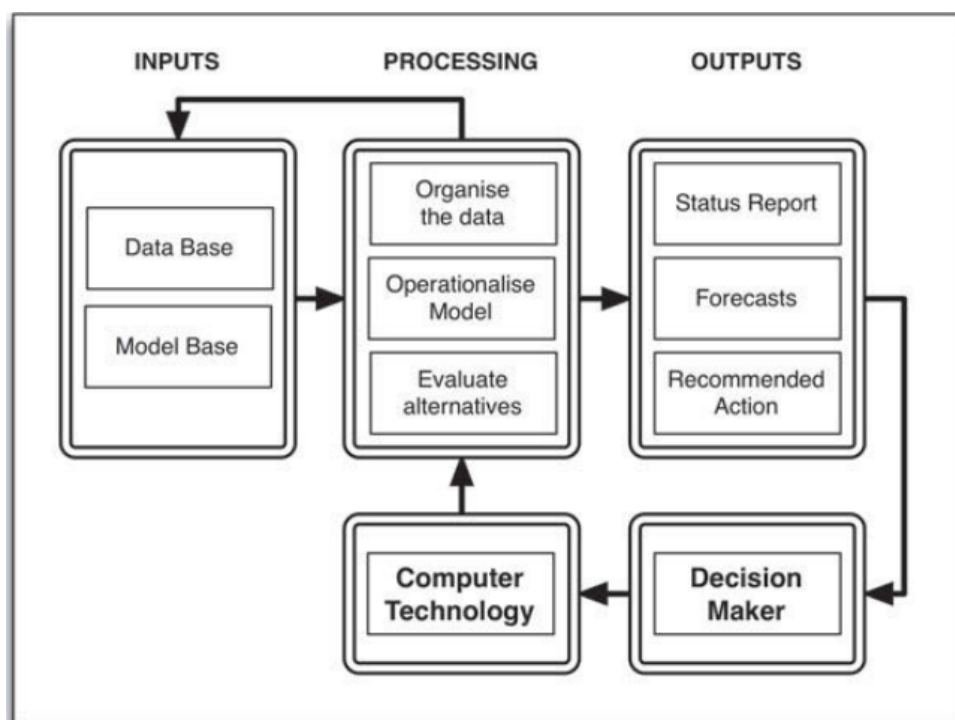
O conhecimento dos especialistas do domínio está implícito nos SADs. Evans (2003) explica que existe uma necessidade de modelar o conhecimento chave de um domínio em um modelo, para permitir a comunicação e colaboração entre especialistas de domínio e os desenvolvedores, portanto o modelo de conhecimento dos especialistas será objeto da pesquisa.

A seguir será apresentada a arquitetura dos SAD e explicada a importância dos modelos de conhecimento existentes dentro um SAD.

2.2 Arquitetura para Sistemas de Apoio à Decisão

A arquitetura de um *software* define a organização dele em termos de componentes, de interconexões e das interações com sistemas externos (JONG, 1997). A arquitetura fornece as informações de como os componentes dela relacionam-se, explicando a parte externa das ligações entre seus componentes, sendo que as implementações internas não são consideradas parte da arquitetura.

Figura 1 – Componentes de um SAD.



Fonte: Adaptada de Tweedale, Phillips-Wren e Jain (2016).

A arquitetura de um SAD pode ser representada pelos componentes ilustrados na Figura 1, representando o processo realizado pelos SADs, no qual recebem uma entrada, fazem o processamento dela e retorna resultados que são analisadas pelo tomador(es) de decisão por meio da tecnologia computacional (TWEEDALE; PHILLIPS-WREN; JAIN, 2016).

A Figura 1 mostra os componentes de um SAD, que são generalizados em:

Entradas (*Inputs*) corresponde às entradas do sistema, composta dos dados que serão processados e dos modelos de conhecimento dos especialistas. Os dados estão armazenados em bancos de dados e os modelos em geral estão implícitos no SAD ou podem estar em uma base de conhecimento. Esses dois componentes devem ser o mais precisos e completos possíveis para garantir respostas confiáveis do sistema.

Processamento (*Processing*) composto pelos modelos e métodos de organização e processamento dos dados, que têm restrições para avaliar as alternativas de resposta. Os métodos podem ser de tipo matemáticos, que processam os dados e geram os resultados do sistema.

Saídas (*Outputs*) são os resultados do processamento dos inputs e permitem comparar as alternativas de decisão. As saídas comuns são relatórios, previsões e recomendações, apresentados por meio de uma interface gráfica para facilitar o entendimento e interação por parte dos usuários.

Durante a evolução dos SAD, várias melhorias aconteceram, entre elas o desenvolvimento da *web* permitiu integrar novas técnicas no processamento dos dados, tecnologias na representação visual de resultados e no uso colaborativo por parte dos usuários (SHIM et al., 2002). Também existe a tendência da integração com métodos de inteligência artificial, para estender a aplicabilidade dos SAD a problemas complexos.

2.3 Taxonomia dos Sistemas de Apoio à Decisão

Power (2002) classificou os SADs em 5 categorias principais cujo critério de classificação foi a funcionalidade dominante de cada SAD. As categorias propostas foram:

- *Data-Driven DSS*: SADs especializados no processamento e gerenciamento de grandes quantidades de dados estruturados (principalmente dados históricos). Esses SADs podem estar *on-line* fornecendo funcionalidades de agregação de dados e cálculos simples de maneira remota.
- *Model-Driven DSS*: SADs especializados em gerenciar modelos que representam aspectos de uma realidade, exemplos deles são ferramentas estatísticas e analíticas onde é possível editar os modelos. Esses SADs usam dados e parâmetros fornecidos pelos tomadores de decisão para suportar o processo de tomada de decisão, mas não são sistemas de dados intensivos.

- *Knowledge-Driven DSS*: SADs especializados em resolução de problemas por meio de regras, fatos, procedimentos ou conhecimento do domínio que permite recomendar soluções aos tomadores de decisão. Esses sistemas são caracterizados por conter conhecimento de um domínio em particular.
- *Document-Driven DSS*: SADs especializados em ajudar os tomadores de decisão a obter, recuperar, classificar e gerenciar documentos não estruturados. Esses sistemas integram uma variedade de tecnologias de armazenamento e de processamento para fornecer a recuperação e análise de informações.
- *Communications-Driven*: SADs especializados suportar a comunicação, colaboração e decisão dentro de um grupo de pessoas. Esses sistemas usam tecnologias de comunicação e modelos de processos de decisão para resolver problemas por duas ou mais pessoas.

Os *Knowledge-Driven DSS*, também denominados *Rule Based Systems* (TWEEDALE; PHILLIPS-WREN; JAIN, 2016), usam bases de conhecimento que geralmente integram regras para suportar a representação, organização e inferência de conhecimento. Esses tipos de sistemas compõem o escopo da presente pesquisa.

Um tipo de *Knowledge-Driven DSS*, são os SADs baseados em ontologias que representam o conhecimento dos especialistas, permitindo definir, classificar, relacionar e inferir conhecimento.

2.4 Sistema de Apoio à Decisão SustenAgro

Um domínio de conhecimento caracterizado pela complexidade são os sistemas produtivos agrícolas. Eles envolvem fenômenos de natureza diversa (SIMON, 1991), integrando aspectos ambientais, sociais e econômicos.

Particularmente, a produção da cana-de-açúcar e os subprodutos dela, são extremamente importante para a economia do estado de São Paulo e do Brasil, devido ao fato de ser uma das principais culturas produzidas no país (TORQUATO, 2015). Atualmente a cana-de-açúcar é a mais importante fonte de energia renovável no Brasil (SEABRA et al., 2011), permitindo a produção de etanol e de bioeletricidade, além de ter mais de 20 subprodutos, entre eles açúcar, bioplásticos e hidrocarbonetos ¹.

A produção da cana-de-açúcar e dos subprodutos dela, influem em aspectos ambientais consumindo os recursos naturais, em aspectos sociais envolvendo pessoas na produção e em aspectos econômicos na comercialização. Esses aspectos fazem complexo manter a produtividade sem afetar a sustentabilidade. Por essas razões, a Embrapa Meio Ambiente escolheu especificamente o sistema produtivo da cultura de cana-de-açúcar na região

¹ <<http://sugarcane.org/sugarcane-products>>

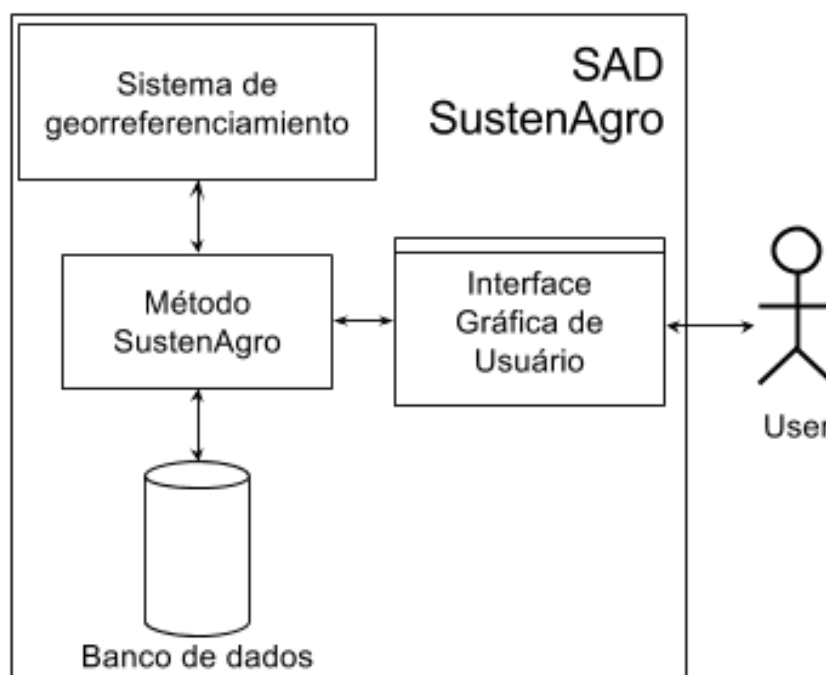
centro-sul do Brasil, como sistema piloto para desenvolver um método e SAD de avaliação da sustentabilidade (Apêndice A).

Dada a complexidade da análise da sustentabilidade em sistemas de produção agrícola, os pesquisadores da Embrapa Meio Ambiente trabalharam na definição de métodos que permitissem avaliar a sustentabilidade de maneira integral (SINGH et al., 2012). Por essa razão, desenvolveram o método SustenAgro que aborda a avaliação em termos de indicadores, simplificando a complexidade deste sistema agrícola. Cada indicador mede um determinado aspecto crítico no sistema produtivo, para determinar o quão sustentável ele é. A partir da análise de cada indicador, é possível gerar recomendações de medidas corretivas para as unidades produtivas ou para o embasamento de políticas públicas que incentivem a sustentabilidade. A definição conceitual do processo de avaliação da sustentabilidade em cana-de-açúcar está detalhada no apêndice A.

A partir do método SustenAgro, foi desenvolvido o SAD Sustenagro que suporta a avaliação de sustentabilidade, ele usa o método SustenAgro e que consegue adaptar-se às mudanças do domínio.

O SAD SustenAgro suporta a avaliação da sustentabilidade em cana-de-açúcar no centro-sul do Brasil. A Figura 2 apresenta a arquitetura inicial do SAD SustenAgro, definida a partir dos requisitos dos especialistas. Esta arquitetura corresponde a um sistema de informação tradicional, que requer a intervenção de desenvolvedores de *software*, para definir ou atualizar o conhecimento dos especialistas implícito no SAD.

Figura 2 – Arquitetura inicial do SAD SustenAgro .



Fonte: Elaborada pelo autor.

Os especialistas em sustentabilidade definiram o SAD SustenAgro com as seguin-

tes características:

- Sistema *web* com banco de dados para armazenar e recuperar as informações do sistema.
- Integração e implementação do método SustenAgro de avaliação de sustentabilidade (Apêndice A)
- Flexibilidade para adaptar o método SustenAgro a outras culturas.
- Integração com sistemas de georreferenciamento.
- Desenvolvimento de *widgets* (componentes visuais dos sistemas web) para mostrar resultados obtidos, especificamente a implementação da matriz de sustentabilidade e do semáforo da sustentabilidade (Capítulo 5).
- Geração de relatórios e de recomendações de sustentabilidade.

Um dos problemas identificados foi que os especialistas não tinham uma definição clara do SAD SustenAgro. Portanto foi necessário realizar um levantamento de requisitos (Capítulo 5), para definir os requisitos funcionais e não funcionais. Além disso, foi necessário reestruturar o desenvolvimento do SAD SustenAgro para que fosse integrado no processo de pesquisa.

O SAD SustenAgro faz parte de um conjunto de ferramentas de avaliação definidas pela Embrapa Meio Ambiente. A partir da análise das ferramentas similares ao SAD SustenAgro, foi evidenciada a necessidade de fornecer métodos e ferramentas computacionais que organizem a informação. Para apoiar aos especialistas a tomar decisões baseadas em conhecimento, permitindo simplificar a resolução de problemas que de outra maneira não seriam triviais.

As seguintes ferramentas *software* foram analisadas e concluiu-se que têm em comum um método de avaliação que processa de maneira matemática um conjunto de dados e gera relatórios com resultados da avaliação, gráficos e recomendações.

1. Sistema Innova-Tec: avaliação do impacto da inovação tecnológica².
2. Sistema Nano-Tec: avaliação do impacto das nanotecnologias³.
3. Sistema GMP-RAM v.1.1: avaliação de Risco de Plantas Geneticamente Modificadas (GMP)⁴.
4. Software para avaliação de segurança e impactos de plantas geneticamente modificadas⁵.

² <http://www.cnpma.embrapa.br/forms/inova_tec.php3>

³ <<https://www.embrapa.br/en/busca-de-publicacoes/-/publicacao/951543/metodologia-para-avaliacao-de-impactos-das-nano>>

⁴ <http://www.cnpma.embrapa.br/forms/gmp_ram.php3>

⁵ <<http://www.cnpma.embrapa.br/nova/mostra2.php3?id=857>>

5. Sistema Atlantis: Sistema para levantamento e sistematização da informação técnica em temas de pesquisa, tecnologias e inovação.⁶

Uma característica importante nesses sistemas foi a existência de conceitos de domínio específico na organização dos dados de entrada dos SAD e no método de avaliação. Esta característica permitiu identificar que cada um dos sistemas utiliza um processo de modelagem dos conceitos do domínio por parte dos desenvolvedores.

Identificou-se que a implementação desse conhecimento gerava dificuldades de compreensão entre os especialistas do domínio e os desenvolvedores de *software* por serem de áreas diferentes. Evans (2003) propõe que este conhecimento deve ser representado em um modelo independente.

Por conseguinte, afirmou-se a hipótese de usar um KOS para representar dito conhecimento, e baseando-se analisando o problema de pesquisa, foram selecionadas as ontologias como o modelo mais completo para representar o conhecimento do domínio dos especialistas na definição de SAD. Desta maneira foi evitado que o conhecimento ficara implícito no *software* como aconteceu no desenvolvimento dos SAD listados.

O uso de uma ontologia permite representar e estruturar o conhecimento de avaliação da sustentabilidade em agricultura, por meio da definição e atualização de conceitos por parte dos especialistas, permitindo que eles mesmos descrevam o domínio sem precisar dos desenvolvedores de *software*. Os especialistas do domínio têm familiaridade com os termos da ontologia e poderão especificar grande parte do conhecimento envolvido no SAD. Idealmente, essa definição deve ser detalhada o suficiente para que os desenvolvedores possam desenvolver a parte computacional do SAD sem necessidade de *feedback* dos especialistas.

Essa representação de conhecimento permite representar relaciones semânticas, integrar sistemas de inferência que complementem o modelo e gerem informações para suportar a decisão. A partir desse modelo computável definido pelos especialistas foi gerado o SAD SustenAgro.

Neste contexto, o SAD SustenAgro foi escolhido como projeto piloto para desenvolver a presente pesquisa, porque permite avaliar alternativas na definição e geração de SAD baseados em conhecimento.

2.5 Trabalhos relacionados

Com a finalidade de relacionar pesquisas sobre o tema que forneçam ideias e exemplos para abordar o problema, realizou-se uma consulta na literatura por SADs que usassem ontologias do domínio dos especialistas, e SADs semelhantes ao SustenAgro. Foi feita uma pesquisa bibliográfica utilizando fontes de informação acadêmica.

⁶ <<https://www.embrapa.br/en/busca-de-produtos-processos-e-servicos/-/produto-servico/2102/atlantis---atlantis>>

Sobre o uso de ontologias em domínios similares ao SustenAgro, tem-se:

O vocabulário *Agricultural Vocabulary (AGROVOC)*⁷ que é um *thesaurus* (sistema de referência de termos) fornece termos padronizados sobre alimentação, nutrição, agricultura, pesca, floresta e meio ambiente criados de maneira colaborativa e coordenados pela *Food and Agricultural Organization*⁸ (FAO).

Esses termos podem ser reutilizados em ontologias (LIANG et al., 2006), dando um exemplo de padronização com os identificadores dos conceitos, reutilizando informações e integrando os conceitos com outros dados da *Linked Open Data* (LOD)

Kraines e Guo (2011) desenvolveram uma ferramenta com o objetivo de criar um sistema de compartilhamento de conhecimento (*Knowledge Sharing System*), para pesquisa em sustentabilidade, por meio de um processo de modelagem semântica. Uma ontologia, fundamentada em lógica descritiva, foi desenvolvida por meio do modelo de dados ISO 15926 para descrever três tipos de conceitualizações relacionadas à sustentabilidade: conhecimento situacional, métodos analíticos e *frameworks* de cenários. Os conhecimentos dos especialistas podem ser descritos por meio de afirmações semânticas (*semantic statements*).

Abt, Vigier e Schneider (2009) apresenta uma proposta para modelar conceitualmente fazendas para desenvolver sistemas de gerenciamento que abrangem maiores complexidade do domínio. O objetivo é melhorar o sistema produtivo, principalmente em aspectos de sustentabilidade.

Bonacin, Nabuco e Junior (2013) apresentam uma ontologia sobre recursos hídricos na agricultura que tem o objetivo de fornecer um meio de recuperação e compartilhamento do conhecimento de especialistas. Esse trabalho tem o objetivo de fornecer um meio de representação de conhecimento em agricultura, mas não aborda maneiras de facilitar seu uso por especialistas sem conhecimento de modelagem.

Cada uma dessas pesquisas fornece um exemplo do uso de ontologias na criação de soluções baseadas em conhecimento. Isto foi confirmado por (ROUSSEY et al., 2010) que afirma que o uso ontologias têm sido realizado em várias aplicações relacionadas a agricultura. Dadas as afirmações dessas pesquisas, pode-se concluir que uma ontologia pode proporcionar o suporte na representação e organização de conhecimento necessário para cumprir os requisitos do sistema SustenAgro.

Sobre SADs semelhantes ao SustenAgro, encontrou-se que uma estratégia para abordar a complexidade em SADs é a utilização de métodos e metodologias de avaliação que utilizam indicadores. Um exemplo desse enfoque é a pesquisa de Olsson et al. (2009). Nela foi desenvolvido um *framework* de indicadores que relaciona, de uma maneira consistente, as dimensões ambiental, econômica e social do desenvolvimento sustentável. Seu principal benefício é uma relativa simplicidade na apresentação da informação e a possibilidade de

⁷ Definição do Agrovoc <<http://aims.fao.org/agrovoc>>

⁸ Site da FAO <<http://www.fao.org/home/en/>>

vincular novos indicadores.

Ewert et al. (2009) apresentam várias estratégias para abordar a complexidade nos sistemas agrícolas. Eles começam relacionando a agricultura com os sistemas socioeconômicos e naturais e enfrentam o problema de gerir suas múltiplas funções, de uma maneira sustentável.

Existem pesquisas que abordam a sustentabilidade por meio de ferramentas tecnológicas, as quais podem servir de referência ao sistema SustenAgro. Uma delas foi desenvolvida por (BRILHANTE et al., 2006) e consiste em um *framework* (MOeMA-IS) para análise de aspectos de sustentabilidade do estado do Amazonas. Ele usa uma ontologia para descrição de indicadores de sustentabilidade (*ISD-Economics Ontology*).

Soullignac et al. (2012) apresenta um sistema de gestão de conhecimento que fornece serviços computacionais para selecionar e formalizar o conhecimento em agricultura orgânica. Esse trabalho abrange técnicas para formalizar o conhecimento agrícola com a visão de fornecer informações que ajudem a tomar decisões e a adotar práticas sustentáveis.

Kumazawa et al. (2009) advoga que, na sustentabilidade, é necessário criar meios que permitam definir conhecimento estruturado e demonstra que as ontologias fornecem o suporte para isso. Kumazawa et al. (2009) confirma que as ontologias podem ser o meio de representação de conhecimento que é requerido em sistemas como o SustenAgro, já que a proposta é focada em sustentabilidade.

SADs têm usado ontologias para suportar várias fases do processo de decisão como: a organização dos dados, seu processamento e a representação da informação produzida (ROSPOCHER; SERAFINI, 2012).

2.6 Considerações finais

A partir da análise dos SADs desenvolvidos pela Embrapa Meio Ambiente e dos requisitos iniciais do SAD SustenAgro, foi evidenciado é necessário um meio computável de definição de conhecimento por parte dos especialistas para organizar dito conhecimento e integrá-lo com os SADs.

Segundo o estado da arte dos SADs relacionados com ontologias, permite identificar que as ontologias suportam vários tipos de SADs e domínios de conhecimento. Porém, será avaliada se as ontologias suportam a representação de conhecimento de sustentabilidade e as características do SAD SustenAgro. O uso de ontologias permite também testar novas possibilidades na definição e geração de SAD baseados em conhecimento.

Desta forma será abordada uma solução ao problema da inexistência de uma representação de conhecimento para definir SADs, que tenha um formato computável, entendível e acessível aos especialistas do domínio e desenvolvedores de *software*.

Web Semântica e DSLs

Ontologias da *web* semântica e DSLs têm um papel fundamental na criação de um meio de descrição de conhecimento por parte dos especialistas e, portanto, suportar o *design* de SADs baseados em conhecimento.

Ontologias servem para representar o conhecimento de especialistas do domínio e DSLs servem para customizar o comportamento dos SADs. As ontologias apareceram originalmente no contexto da filosofia, onde se referem ao estudo da natureza, existência e realidade dos entes. Elas são usadas em vários campos do conhecimento. Neste projeto, ontologias referem-se a representações de conhecimento, que precisam ser implementadas em código. As implementações de software podem trabalhar com ontologias da Web Semântica que fornecem a criação, armazenamento, busca e modificação de ontologias, seguindo padrões de formatos abertos.

Neste capítulo, vamos apresentar e discutir as ontologias da *web* semântica e as DSLs. Descrevendo a teoria da Web Semântica: fundamentos, Ontologias, o *Resource Description Framework* (RDF) e a *Web Ontology Language* (OWL). Finalmente serão abordadas as *Domain Specific Languages* (DSLs) que são linguagens que permitem definir um meio de comunicação entre os especialistas e o sistema desenvolvido.

3.1 Web Semântica

A web foi criada para possibilitar o acesso, intercâmbio e recuperação de informações de maneira rápida e simples, seu crescimento exponencial e caótico fez com que a mesma se tornasse hoje um gigantesco repositório de documentos, o que dificulta a recuperação de informações. Até o momento, não existe nenhuma estratégia abrangente e satisfatória para a organização de documentos por meio de “motores de busca” que seja coerente com uma estrutura linguística (SOUZA; ALVARENGA, 2004).

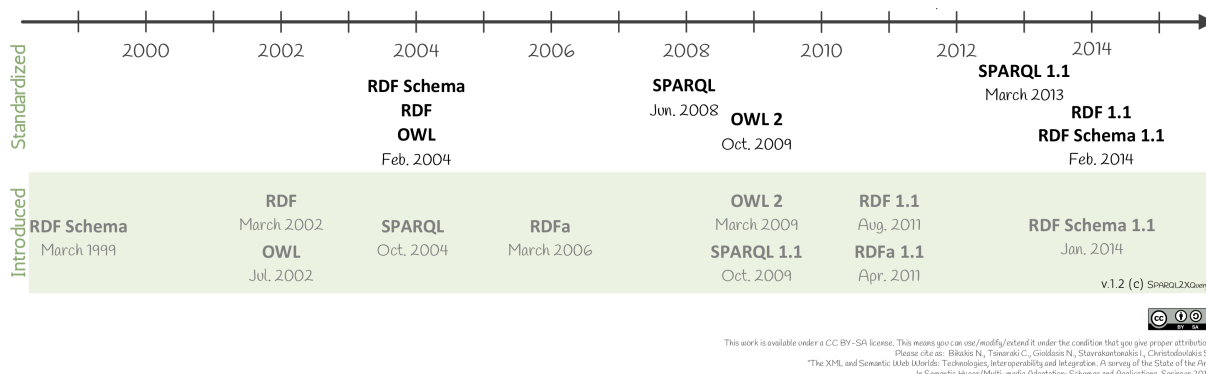
Um exemplo da deficiência da *web* atual pode ser identificada na busca realizada

pelos sistemas de recuperação de informação, que usam palavras-chave nas buscas, onde apenas a similaridade e o número de ocorrências de certas palavras no conteúdo de documentos são levados em consideração e não a semântica presente naquela informação. (SOUZA; ALVARENGA, 2004).

A *Web Semântica* aparece como uma proposta para organizar o conhecimento da internet semanticamente em formatos entendíveis pelos humanos e máquinas (BERNERS-LEE; HENDLER; LASSILA, 2001). Procurando métodos para que as máquinas consigam realizar a interpretação do significado, que é uma habilidade inata dos seres humanos, por meio da associação dos conceitos que estão no cérebro por meio de estruturas neurais e que não é suportado pelas máquinas tradicionais.

A *Web Semântica* tem como finalidade estruturar os dados e informações disponíveis na *Web*, para que tenham significado e sejam computáveis por máquinas. Gerando um ambiente onde agentes de *software* e usuários possam trabalhar de maneira cooperativa. A *Web Semântica* é definida por um conjunto de padrões propostos pelo *World Wide Web Consortium* (W3C). A Figura 3 apresenta alguns dos padrões que constituem a *Web Semântica* de maneira cronológica.

Figura 3 – História da *Web Semântica*

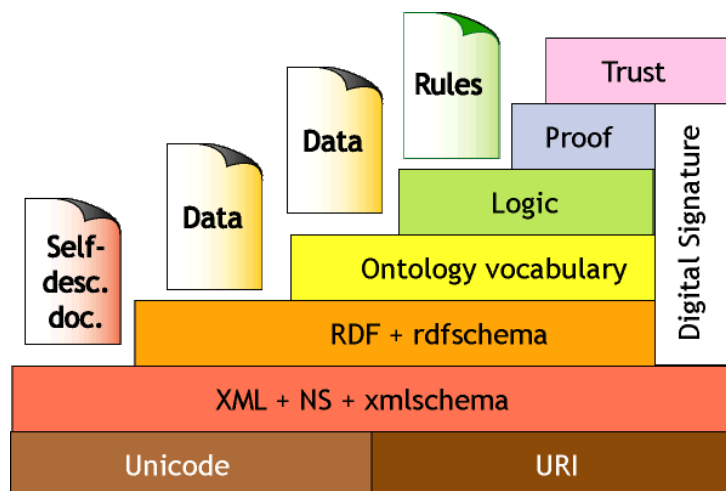


Fonte: Bikakis et al. (2013).

Berners-Lee, Hendler e Lassila (2001) propuseram a *Web Semântica*, em 2001, como uma extensão da *Web* atual, na qual é possível vincular conceitos de maneira estruturada e padronizada. Permitindo a criação de conhecimento estruturado, computável por máquinas, que pode ser compartilhado entre humanos e máquinas. A finalidade é criar uma *web* universal dos conhecimentos da humanidade.

A partir dessa visão conceitual sobre a *web*, Berners-Lee, Hendler e Lassila (2001) propuseram uma arquitetura que organiza as representações do conhecimento por meio de camadas, conhecida como *Semantic Web Cake* (FENSEL et al., 2011), que é ilustrada na Figura 4.

Figura 4 – Arquitetura em camadas da Web Semântica



Fonte: Fensel et al. (2011).

A base dessa arquitetura é estabelecida pelos padrões *Unicode* e *Uniform Resource Identifier* (URI), que padronizam a representação dos dados por meio das seguintes camadas:

Unicode é um padrão que codifica os caracteres na maioria dos sistemas de escrita para representação de texto com fins de processamento computacional.

URI permite identificar os recursos disponíveis na Web por meio de uma 'String' única.

XML representa os dados de maneira sintática, por meio da definição de *markups* que codificam documentos em formatos preestabelecidos. Ela permite que informações sejam legíveis tanto por humanos como por computadores, suportando as camadas superiores na arquitetura.

RDF é um modelo padrão para intercambiar dados na web. *RDF* tem características que permitem a integração de dados inclusive de esquemas diferentes, e suporta especialmente a evolução dos esquemas por meio do tempo sem requerer que mudanças nos consumidores de dados. *RDF* é uma recomendação do W3C¹.

Ontology estende a camada de descrição, fornecendo mais expressividade na definição de conceitos, de classificações, de relações e de inferência.

Logic permite definir regras lógicas para deduzir e inferir novas informações que conseguem mudar a estrutura da ontologia de maneira dinâmica.

Proof fornece mecanismos para avaliar o nível de confiabilidade das fontes de recursos e informações.

¹ <<https://www.w3.org/RDF/>>

Trust representa o conhecimento validado e confiável.

Digital-Signature permite integrar métodos de segurança que garantam a segurança da informação.

Uma das contribuições importantes da *Web Semântica* foi a formalização da representação de ontologias (próxima sessão). No desenvolvimento desta pesquisa, foram usadas desde as camadas inferiores até o *OWL*, permitindo definir ontologias que representam os domínios de conhecimento.

Ontologias

Existem várias interpretações do conceito ontologia, dependendo da finalidade para qual elas sejam usadas. Smith et al. (2007) descrevem a ontologia como uma área da filosofia, que estuda a natureza, existência e realidade dos entes, assim como as categorias do ser e das relações semânticas.

Na ciências da computação e informação, a palavra “ontologia” é definida como uma especificação formal e explícita de uma conceitualização compartilhada de um domínio de conhecimento. Allemang e Hendler (2011) definem as ontologias, no contexto da *Web Semântica*, como um esquema de representação que permite conceitualizar e estruturar conhecimento, permitindo a interpretação por computadores, com o objetivo de compartilhar conhecimento entre humanos e computadores.

Uma ontologia é um sistema de organização e representação do conhecimento, do inglês *Knowledge Organization System* (KOS), que é uma estrutura conceitual e computacional que permite representar o conhecimento, de qualquer domínio, por meio de entidades, classificações, relações semânticas, regras e axiomas. Uma ontologia é especificada por meio de componentes básicos que são as classes, relações, axiomas e instâncias.

Classes são o foco da maioria das ontologias. Elas são utilizadas para descrever os conceitos de um domínio, possibilitando a organização e classificação dos indivíduos em um sistema lógico e hierárquico, contendo subclasses que representam conceitos específicos (NOY; MCGUINNESS et al., 2001).

Relações representam o tipo de interação entre os conceitos de um domínio e as propriedades presentes nas classes e indivíduos. Elas podem ter características próprias, como serem transitivas, simétricas, ou terem uma cardinalidade definida.

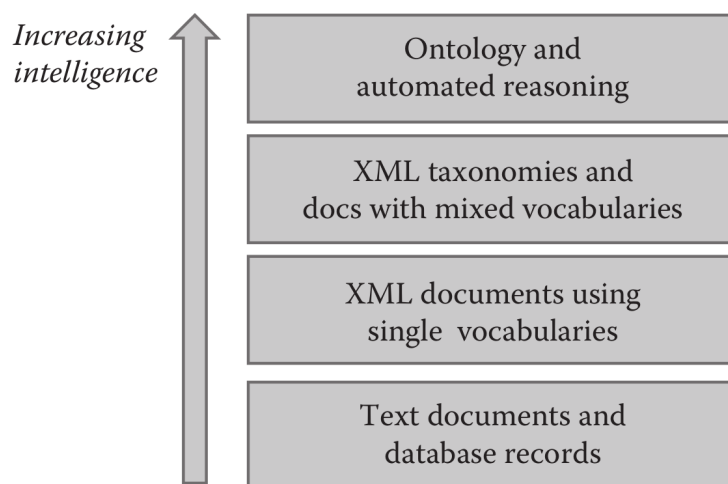
Axiomas são utilizados para modelar regras assumidas como verdadeiras no domínio em questão, de modo que seja possível associar relacionamentos entre os indivíduos, além de fornecer características descritivas e lógicas para os conceitos.

Indivíduos ou instâncias das classes, são utilizados para representar elementos específicos, ou seja, os próprios dados, que juntamente com a definição de uma ontologia,

constituem a base de conhecimento (NOY; MCGUINNESS et al., 2001). Indivíduos representam objetos do domínio de interesse (HORRIDGE; BECHHOFFER, 2011).

Segundo Patel-schneider (2005), a representação de uma ontologia é feita por meio de lógica de predicados e lógica descritiva, usando padrões adotados pela comunidade, como *RDF* e *OWL*. A Figura 5 mostra os níveis de representação de dados na forma de conhecimento processável por máquinas (ALLEMANG; HENDLER, 2011).

Figura 5 – *Smart data continuum*: níveis de representação de dados na forma de conhecimento processável por máquinas.



Fonte: Allemang e Hendler (2011).

O nível mais baixo de representação começa com os dados sem nenhum significado semântico, dependentes do contexto da aplicação. O segundo nível envolve a definição de esquemas *XML* para conseguir independência dos dados da aplicação, os dados fluem entre aplicações em um único domínio mas não podem ser compartilhados fora do domínio. No terceiro nível, os dados podem ser combinados a partir de diferentes domínios, sendo suficientemente independentes para serem recuperados e combinados com outras fontes de dados. Finalmente no quarto nível, é possível inferir novos dados a partir dos existentes e compartilhá-los entre aplicações sem requerer interferência humana (SUGUMARAN; GULLA, 2011).

Resource Description Framework (RDF)

O *Resource Description Framework* (RDF) é uma família de especificações da W3C, que foi disponibilizada em 1999 como parte do W3C's *Semantic Web Effort*. Elas fornecem um estrutura comum que permite que dados sejam compartilhados e reusados por meio das fronteiras das aplicações, empresas e comunidades ². O *RDF* foi originalmente projetado

² <<http://www.w3.org/2001/sw/>>

como um modelo de metadados e também chegou a ser usado como um método de descrições conceituais, principalmente para descrever recursos web e formalmente é um formato de dados de tipo grafo direcionado e rotulado para representar informação na *web*³.

O *RDF* é usado em várias áreas de aplicação, como *resource discovery*, para melhorar as capacidades dos motores de busca, cataloging, para descrever conteúdo e as relações de conteúdo disponibilizados em um sistema web particular, e descrição de *intellectual property rights* de páginas web. Seu modelo básico de dados consiste em um padrão de três tipos de objetos, conhecido como triplas:

- **Sujeito:** representa os recursos e são identificados por meio de URIs. Por exemplo, uma página web ou um elemento *HyperText Markup Language* (HTML) podem ser recursos.
- **Predicado:** são aspectos, características, atributos ou relações específicas que descrevem o sujeito, cada predicado têm um significado específico e relaciona um sujeito com um objeto.
- **Objeto:** um recurso específico ou valor de propriedade que representa uma característica do sujeito⁴

Com *RDF* é possível explicitar relações entre dois objetos (usando-se uma Tripla *RDF*), mas não é possível fazer modelagens específicas nem inferência. Para descrever detalhadamente o que um objeto representa e suas relações com outros objetos, são necessárias ontologias descritas no padrão *OWL*.

SPARQL Protocol and RDF Query Language (SPARQL)

SPARQL é uma linguagem de consulta semântica usada por bancos de armazenamento e recuperação de dados de dados compatíveis com o formato RDF ou que sejam fornecidos como RDF via *middle-ware*. Atualmente é um padrão especificado pela W3C e uma das tecnologias principais da web semântica.

A versão de SPARQL 1.1, veio com novas características que permitem a atualização de dados em formato RDF, permitindo atualizar, criar e remover dados em formato RDF em um *graph store*⁵.

Web Ontology Language (OWL)

A *Web Ontology Language* (OWL) foi recomendada pelo W3C em 2004 para representar e compartilhar ontologias na Web. Essa linguagem foi projetada para aplicações que necessitam processar o conteúdo da informação, em vez de apenas organizar informações

³ <<https://www.w3.org/TR/rdf-sparql-query/>>

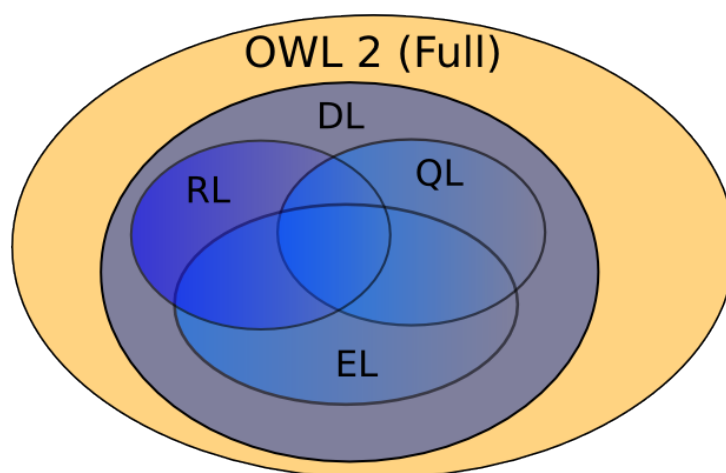
⁴ <http://www.w3.org/TR/PR-rdf-syntax/>

⁵ <<https://www.w3.org/TR/sparql11-update/>>

em nós (MCGUINNESS; HARMELEN et al., 2004). *OWL* é uma linguagem que permite que a semântica seja explicitamente associada ao conteúdo dos dados na *web* e formalmente especificada por meio de ontologias, compartilhadas na *internet*.

A versão *OWL 2* é a versão mais recente da linguagem. De acordo com as especificações do W3C⁶, a *OWL 2* adicionou três novos perfis (sub-linguagens) aos perfis *DL* e *Full* já existentes: *OWL 2 EL*, *OWL 2 QL* e *OWL RL* (Figura 6)⁷. Cada um desses perfis fornece características de expressividade diferente para diversos cenários de aplicação:

Figura 6 – OWL2 Profiles.



Fonte: W3C <<https://www.w3.org/People/Sandro/owl2-profiles-doc>>

Full O perfil *OWL Full* é direcionado para usuários que querem a máxima expressividade e a liberdade sintática do *OWL* sem garantia computacional. É improvável que qualquer motor de raciocínio seja capaz de suportar completamente cada recurso da *OWL Full* (MCGUINNESS; HARMELEN et al., 2004).

DL O perfil *OWL DL* (*Description Logic*) é para aplicações que necessitam de máxima expressividade, enquanto mantém a computabilidade (todas as conclusões são garantidas de ser computáveis) e decidibilidade (todas as computações terminarão em tempo finito) (MCGUINNESS; HARMELEN et al., 2004). *OWL DL* inclui as construções da linguagem *OWL*, mas elas podem ser usadas somente sob certas restrições.

EL O perfil *OWL 2 EL* é baseado na família *EL++* de lógica descritiva (*Description Logic*). Esse perfil é particularmente útil em aplicações utilizando ontologias que contêm um grande número de propriedades e/ou classes. Além disso, o *OWL 2 EL* utiliza um padrão comum, utilizado em ontologias, para conceitos e planejamento, ou seja, a combinação de conjunção e qualidades existenciais.

⁶ <http://www.w3.org/TR/owl2-overview/>

⁷ Figura original do W3C <<https://www.w3.org/People/Sandro/owl2-profiles-doc>>

- QL** O perfil OWL 2 QL é baseado na família DL-Lite de lógica descritiva. Esse perfil foi criado para permitir o raciocínio (*reasoning*) eficiente com grandes quantidades de dados estruturados de acordo com esquemas relativamente simples. Ele fornece a maioria dos recursos necessários para capturar modelos conceituais, tais como diagramas de classe UML, diagramas de entidade/relacionamento, e esquemas de banco de dados.
- RL** O perfil OWL 2 RL é voltado para aplicações que exigem raciocínio escalável em troca de alguma restrição de poder expressivo. Ele define um subconjunto sintático de OWL 2 que favorece a implementação utilizando tecnologias baseadas em regras. Esse perfil pode ser utilizado na maioria das construções OWL 2. Porém, para permitir implementações baseadas em regras de raciocínio, a forma como essas construções podem ser usadas em axiomas foi restringida.

Protégé

O editor de ontologias Protégé (MUSEN, 2015), é a ferramenta recomendada pela comunidade para criar ontologias em formato OWL, fornecendo:

- *GUI Framework*: para suportar múltiplas vistas da ontologia e layouts configuráveis de componentes.
- *API*: para suportar o desenvolvimento de sistemas baseados em conhecimento.
- *Modularization*: permite suportar a edição de múltiplas ontologias em um mesmo entorno.
- *Navigation*: fornece buscas globais e locais e hipervínculos nos editores.
- *Refactoring tools*: verificação de coerência das ontologias.
- *Reasoning*: compatibilidade com vários *reasoner* para suportar a inferência.
- *Plug-ins*: arquitetura extensível que suporta diferentes tipos de *plug-ins*.

Triplestores

Uma *triplestore* é um tipo de banco de dados, baseado em grafos, para armazenar e recuperar fatos (*assertions*). Esses fatos são representados na forma de triplas no padrão RDF (Seção 3.1). Dados são armazenados em forma de redes de objetos com vínculos rotulados entre eles (RUSHER, 2003). Esse tipo bancos de dados é recomendável quando os dados tem uma estrutura flexível, cujas relações não tem um padrão definido.

A *triplestore Blazegraph*⁸ é um dos mais completos bancos de dados baseados em grafos e com suporte SPARQL. Ela integra as características de uma *triplestore*, fornece suporte nativo à SPARQL e implementa o SPARQL Protocol Endpoint. Esse último, padroniza a comunicação com os clientes e a compatibilidade com os sistemas web, por meio de um *endpoint* (um endereço onde requisições em SPARQL podem ser feitas).

O *Blazegraph* foi escolhido por ter código aberto (Licença GPL) e ser compatível com os padrões da Web Semântica. Mas qualquer *triplestore* que seja compatível com os mesmos padrões (SPARQL 1.1 e RDF) pode ser usada. As principais características dela são:

- Banco de dado baseado em grafo de alta performance
- Suporte *Blueprints API* e RDF/SPARQL
- *Clusters* de replicação altamente disponíveis (HAJournalServer)
- Armazenamento de dados de uma única máquina até ~50B triples/quads (RWStore)
- O armazenamento de dados em *cluster* é essencialmente ilimitado (BigdataFederation)
- REST API Com *deployment* embutida e / ou *webapp* (NanoSparqlServer)
- SPARQL 1.1 nativo
- RDFS+ Inferência e manutenção da verdade
- *Triples, quads*, ou Reificação feita corretamente (RDR) *support*
- Gerenciador de memória Java aproveita o JVM nativo *heap* (no GC)
- API centrada em vértices (RDF_GAS_API)
- Licença dupla: GPLv2 ou comercial
- Assinaturas de suporte ao desenvolvedor e produção

Para desenvolver os sistemas de software, presentes neste trabalho, foram usadas tecnologias da *web* semântica. Primeiramente foram desenhadas as ontologias em formato OWL, na ferramenta Protégé, depois elas foram exportadas em RDF e, finalmente, integradas no *framework Decisioner*, que foi implementado usando a *triplestore Blazegraph*.

O *Blazegraph* permitiu instanciar as ontologias, em formato RDF, suportar inferência e permitir consultas por meio da linguagem SPARQL. Essas funcionalidades foram complementadas com ferramentas para edição da ontologia via *web* (Capítulo 4).

Além das ontologias, foi necessário fornecer um meio de definição de conhecimento mais próximo à linguagem dos especialistas do domínio, para suportar a definição

⁸ <<https://www.blazegraph.com/>>

dos conceitos do domínio nos SADs. O melhor meio de definição de conhecimento identificado foi uma a definição de uma DSL.

3.2 Linguagem de domínio específico

Em desenvolvimento de *software* e engenharia de domínio, uma linguagem de domínio específico, em inglês *Domain-Specific Language (DSL)*, é um tipo de linguagem de programação, ou linguagem de especificação, dedicada a um domínio particular de problema que usa expressões próprias dos especialistas daquele domínio. Um usuário, relacionado com um domínio específico, pode usar uma DSL sem ter experiência em desenvolvimento de *software*, pois a DSL está relacionada com seu domínio de trabalho. Fowler (2010) afirma que programadores instruem o computador no que ele deve fazer, pois já entendem a maneira dele trabalhar, mas, com DSLs, é feito o inverso: o computador começa a entender o que o usuário do domínio escreve.

Segundo Mernik, Heering e Sloane (2005), as vantagens das DSL, em comparação com as linguagens de propósito geral, são a expressividade, facilidade de uso e a integração com o domínio da aplicação. O conceito não é novo, linguagens de programação de propósito específico existem desde o começo das linguagens de programação, mas o termo tornou-se padrão devido à ascensão da modelagem de domínio específico. DSLs são classificadas da seguinte forma (GHOSH, 2010):

- *Domain-Specific Markup Languages*: são linguagens de um domínio particular com a particularidade de anotar os dados com etiquetas para que eles sejam sintaticamente distinguíveis. Um exemplo delas é a *Hypertext Markup Language (HTML)*, que permite anotar dados no domínio das páginas *web*.
- *Domain-Specific Modeling Languages (specification languages)*: são linguagens que permitem especificar sistemas com o propósito de modelá-los. São compostas de uma estrutura consistente e de um conjunto de regras que permitem interpretar o significado dos componentes modelados. Uma linguagem desse tipo é a *Unified Modeling Language (UML)*, que permite especificar sistemas de software.
- *Domain-Specific Programming Languages*: são linguagens que permitem a programação em alto nível aplicada a um domínio específico de conhecimento. Uma linguagem desse tipo é a linguagem R que permite a programação de conceitos estatísticos e geração de gráficos.

Segundo o tipo de implementação, as DSLs podem ser divididas em *external DSL* e *internal DSL* (FOWLER, 2010):

External DSL: é uma linguagem de domínio específico que é definida com uma sintaxe independente de outras linguagens de programação, tendo como principal vantagem a

flexibilidade. A desvantagem é que requer um desenvolvimento de um *full parser* para processá-la.

Internal DSL: é uma linguagem de domínio específico escrita dentro de uma linguagem *host* existente. A vantagem desse enfoque de definição de DSL é que o tempo e custo de desenvolvimento é menor, em relação a uma DSL externa. Linguagens desse tipo são escritas sobre uma linguagem de propósito geral e, por isso, apresentam a desvantagem de depender das instruções e características da linguagem *host*, o que, em alguns casos, afeta a flexibilidade e expressividade da DSL que se quer definir.

Linguagem Groovy

Groovy é uma linguagem dinâmica para a máquina virtual Java (JVM) (KOENIG et al., 2007). Ela tem uma sintaxe parecida com *Java*, suporte para programação funcional, produz *JVM bytecodes* e interopera bem com código e bibliotecas *Java*. Traz características de linguagens como *Python*, *Ruby* e *Smalltalk* para uma linguagem similar a *Java* (KOENIG et al., 2007).

O grande benefício que *Groovy* traz para esta pesquisa é o suporte que a sua natureza dinâmica dá ao desenvolvimento de DSLs. Essas DSLs podem rodar diretamente na JVM e usar bibliotecas Java já existentes. DSLs em *Groovy* se integram facilmente à própria linguagem *Groovy*, de modo que não é aparente onde o código em *Groovy* termina e a DSL começa (DEARLE, 2015). Isso permite que a DSL seja implementada como uma DSL interna, estendendo a linguagem *Groovy* (e simplificando a sua criação), mas mantenha uma sintaxe próxima à linguagem usada pelos especialistas de domínio. Como uma DSL interna, ela pode usar ferramentas já existentes para auxiliar a escrita de código em *Groovy*, como editores com *syntax highlighting* e *code completion*, para a sua edição.

Outra vantagem de *Groovy* é que ela é uma linguagem para a JVM e existem muitas bibliotecas, em Java, que dão suporte às tecnologias da Web Semântica. Isso inclui bibliotecas como a OWL API, para trabalhar com ontologias em OWL, Apache Jena, para acesso a *triplestores*, entre outras. Finalmente, os SADs a serem criados serão aplicativos *web*, *Groovy* tem um *web framework* completo e amadurecido, o *Grails*⁹.

Grails usa uma abordagem de convenção sobre configuração que usa opções *default* razoáveis. Ele se integra bem com a JVM e tem características como ORM integrado, DSLs, meta programação durante *runtime* e *compile-time*, e programação assíncrona (SMITH; LEDBROOK, 2009).

Uma DSL pode suportar a definição do comportamento de um SAD, fornecendo uma solução compatível com os termos usados pelos especialistas. Isso facilita que eles possam especificar o comportamento de um SAD com um alto grau de detalhamento, sufi-

⁹

<<https://grails.org/>>

ciente para diminuir ou evitar a necessidade de intervenção de desenvolvedores de *software*. Especialistas podem se tornar, na prática, programadores de seus próprios SADs.

Neste projeto, DSLs servem para customizar o comportamento dos SADs. Por isso, uma *Domain-Specific Programming Language* em *Groovy* foi desenvolvida. Ela faz uso da ontologia para organizar o conhecimento do domínio e definir o comportamento dos SADs.

3.3 Considerações finais

A partir do problema identificado e da revisão da literatura, foi concluído que o desenvolvimento de ontologias é um área de pesquisa (abrangida pela *Web Semântica*) que permite desenvolver sistemas *web* baseados em conhecimento, satisfazendo os requisitos de desenvolvimento do SAD SustenAgro (Capítulo 2).

Porém, a definição de uma ontologia não é uma tarefa trivial, existem dificuldades, por parte dos especialistas do domínio, para formalizar ontologias. Diante deste cenário, foram analisadas várias soluções e encontrou-se que, fornecendo ferramentas simplificadas para edição e complementando ontologias com uma DSL, é possível facilitar a definição dos conceitos de um SAD e seu comportamento por especialistas.

No próximo capítulo, será apresentado o *Framework Decisioner*. Ele faz uso das tecnologias, abordadas neste capítulo, para definir um *framework* que facilita a definição de conhecimento dos especialistas com a finalidade de gerar SADs.

Framework Decisioner

A partir da descrição das características do SAD SustenAgro (Seção 2.4), do requisito de modelar o conhecimento por meio de ontologias (usando tecnologias da *Web Semântica*) e de definir uma DSL para facilitar a definição de comportamentos por parte dos especialistas, foi modelado e desenvolvido o protótipo *Framework Decisioner* que permite definir e gerar SADs baseados em conhecimento. O *Framework Decisioner* é um sistema *web* formado por ontologias, que representam conhecimento do domínio, e por uma DSL que permite definir comportamentos e estabelecer configurações gerais de um SAD.

Satisfazer os requisitos do SAD SustenAgro requeridos pelos pesquisadores da Embrapa Meio Ambiente guiou o desenvolvimento desta pesquisa. Para chegar à arquitetura aqui apresentada, houve um trabalho colaborativo com os especialistas em sustentabilidade para chegar aos requisitos que eram gerais, que deveriam estar no framework, e os que eram específicos, que deveriam estar definidos na ontologia ou na DSL do SustenAgro.

A seguir será apresentada a arquitetura do protótipo Framework Decisioner.

4.1 Arquitetura do Framework Decisioner

Os SADs, segundo a descrição feita no Capítulo 2, são compostos por banco de dados, base de conhecimento, módulo de processamento da informação e módulo gerador de resultados. Em cada um desses módulos, está implícito o conhecimento dos especialistas, portanto a primeira característica definida do *Framework Decisioner* foi a integração de ontologias.

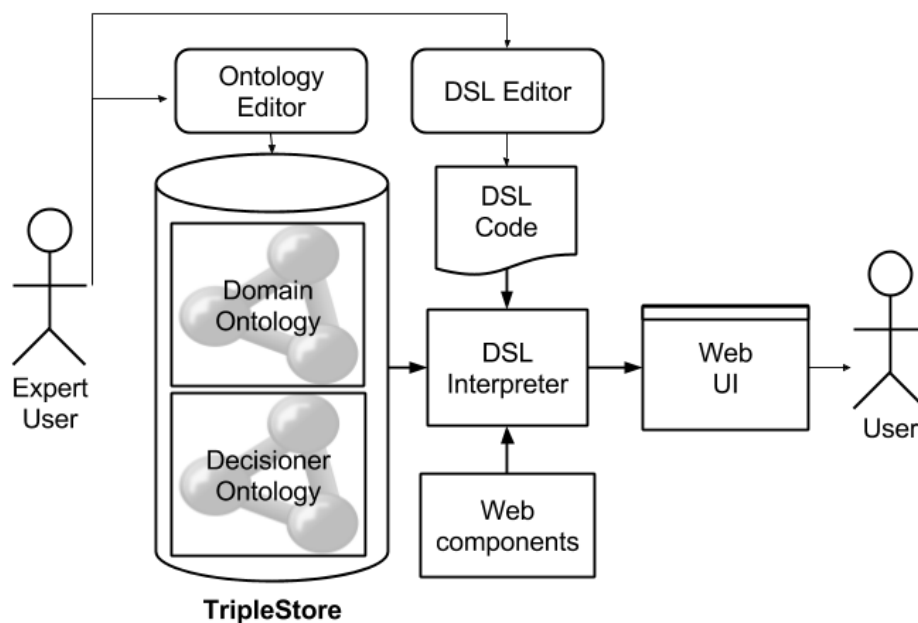
As ontologias permitem a definição do conhecimento dos especialistas em um componente independente. Ele é complementado pela DSL, que permite definir comportamento e características do SAD. O diagrama para a arquitetura desenvolvida é apresentado na Figura 7. Com base nesse diagrama, os componentes dos SADs foram generalizados e, por meio do desenvolvimento de experimentos, foi definida uma arquitetura que pode ser

reusada em diferentes SADs (do mesmo tipo do SustenAgro).

O *Framework Decisioner* pode ser classificado como um *framework*. Já que é uma plataforma de *software* com *design* reutilizável e implementações reutilizáveis pelos clientes, que o especializam em um domínio particular (RIEHLE, 2000).

A arquitetura do *Framework Decisioner* é composta pelos componentes gerais para a definição de SADs, apresentados na Figura 7. Usuários especialistas podem usar o *framework* por meio dos editores de ontologias e DSL. Usuários finais usam os SADs por meio da sua interface *Web*.

Figura 7 – Arquitetura do Decisioner



Fonte: Elaborada pelo autor.

Os componentes dessa arquitetura são:

1. **Domain ontology:** representa os conceitos específicos dos especialistas que serão utilizados no SAD.
2. **Decisioner ontology:** faz uma ligação entre os tipos de dados e as interfaces gráficas capazes de mostrar ou editar esses dados, fazendo um mapeamento entre os dois.
3. **Ontology Editor:** componente que permite editar as ontologias em um formato mais fácil para o uso pelos especialistas.
4. **Triplestore:** sistema de armazenamento e recuperação da informação em formato de triplas RDF, (explicado em detalhe na Seção 3.1). Ele permite o gerenciamento de dados e ontologias em formato RDF e a consulta por meio da linguagem padrão *SPARQL Protocol and RDF Query Language* (SPARQL) (PRUD; SEABORNE et al., 2006).

5. **DSL code**: representa uma instancia da *DSL* com as definições particulares para um SAD específico.
6. **DSL Editor**: editor visual web da *DSL* com recursos como *code completion* e *syntax coloring*. Permite a edição do código da *DSL* por parte dos especialistas do domínio.
7. **Web components**: conjunto de elementos visuais que podem ser reusados nos SADs com a finalidade de modularizar e simplificar a geração de *Web User Interfaces* (UI).
8. **DSL Interpreter**: interpretador da *DSL* para processar as definições do SAD e criar um SAD específico. Ele usa a *DSL*, ontologias e componentes web para gerar automaticamente a interface e comportamento de um SAD específico.
9. **Web UI**: interfaces visuais geradas pelo *DSL Interpreter*. Estão compostas dos *Web Components* que são usados pelos usuários finais na interação com o SAD.

Essa arquitetura foi implementada em um protótipo do *Framework Decisioner*. Sempre que possível, na implementação do *Decisioner*, foram reutilizados componentes ou bibliotecas disponíveis publicamente na *Internet*.

4.2 Metodologia

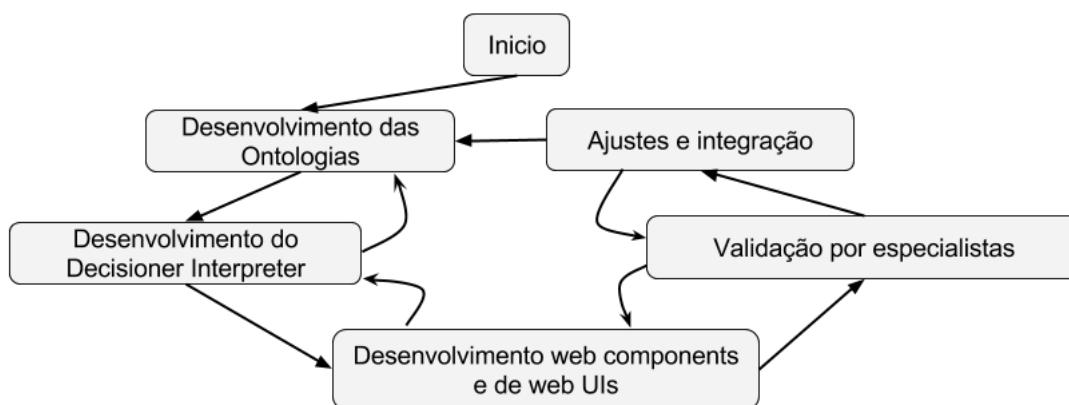
Com a finalidade de desenvolver o protótipo do *Framework Decisioner*, escolheu-se o SAD SustenAgro como um estudo de caso. Ele permitiu a definição de uma metodologia de desenvolvimento para os componentes. Teria sido interessante instanciar o *framework* para um segundo SAD, para demonstrar melhor a generalidade. Contudo, dada as limitações de tempo impostas a um trabalho de mestrado e ao tempo necessário ao desenvolvimento do *framework*, isso não foi possível. No decorrer deste texto, foram feitas algumas considerações sobre a generalidade do *framework*. A metodologia adotada durante este processo, incluiu as seguintes etapas:

1. Seleção da *Triplestore*: foram avaliadas as *triplestores* existentes com a finalidade de definir uma que se adaptasse aos requisitos do *framework*.
2. Seleção da linguagem de programação e *framework web*: foi realizada uma verificação das tecnologias de desenvolvimento de sistemas *web* compatíveis com as tecnologias da *web* semântica e com a *DSL*.
3. *Design* da *DSL*: durante o processo de desenvolvimento do SAD SustenAgro, foram generalizados seus componentes, permitindo definir cada uma das características da *DSL*. O processo foi iterativo, permitindo refinar a expressividade da linguagem com a ajuda de especialistas da Embrapa.

4. Desenvolvimento do *DSL editor*: implementação de uma *web UI* que permite editar a DSL em formato textual, fornecendo aos especialistas um editor moderno. Modificações na DSL podem ser vistas no SAD imediatamente.
5. Desenvolvimento do *Ontology Editor*: implementação de uma *web UI* com componentes específicos para suportar a edição das ontologias, por parte dos especialistas, em um formato textual.
6. Desenvolvimento do *DSL Interpreter*: o interpretador da DSL do Decisioner. Esse componente tinha que ser atualizado a medida que o *design* da DSL mudava.
7. Integração com *web components* e *web UIs*: *widgets* na forma de *Web components* suportam a geração das *Web UIs* que compõem os SADs. *Widgets* foram adicionadas para atender as necessidades do SustenAgro.

A metodologia de desenvolvimento do *Framework Decisioner* foi guiada pelo desenvolvimento do SAD SustenAgro. Primeiramente foram desenvolvidas as ontologias, continuando com o desenvolvimento do *DSL Interpreter* e depois com a integração dos *web components* e *UIs*. Foram realizados vários ciclos de desenvolvimento para refinar as funcionalidades. A Figura 8 representa a metodologia realizada.

Figura 8 – Metodologia de desenvolvimento do Decisioner



Fonte: Elaborada pelo autor.

A seguir, cada componente da arquitetura do *Framework Decisioner* é discutido, começando pela ontologia Decisioner.

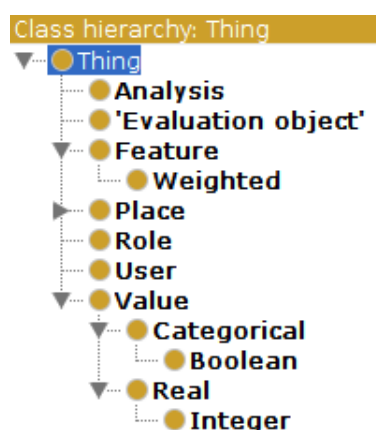
4.3 Ontologia Decisioner

Para desenvolver um protótipo de uma ontologia geral, que abstraísse os componentes dos SADs, foram analisados os SADs desenvolvidos pela Embrapa Meio Ambiente, descritos na Seção 2.4.

A versão inicial da ontologia foi modelada na ferramenta Protégé, no formato OWL. Depois foi criado um novo formato, usando o *YAML Ain't Markup Language* (YAML), mais simples que OWL, para ser usado pelos especialistas no *Ontology Editor*, explicado na Seção 4.4.

A ontologia Decisioner contém os elementos comuns identificados e abstraídos dos SADs da Embrapa, e tem o propósito de fornecer uma ontologia geral que dê suporte a SADs diferentes.

Figura 9 – Modelagem abstrata do SAD



Fonte: Elaborada pelo autor.

Na Figura 9, é apresentada a hierarquia de classes resultante. Ela contém as classes:

Evaluation Object: classe que representa os objetos que serão analisados em cada processo de avaliação. Eles serão indivíduos dessa classe ou de alguma subclasse dela.

Feature: classe que representa as características a avaliar em um *Evaluation Object*. Elas serão quantificadas, analisadas e usadas no processo de geração de relatórios no processo de avaliação. As *Features* têm associado um *Value* que as quantifica. Existe a subclasse *Weighted* que representa uma *Feature* vinculada a um peso. Esse peso pode ser usado nas fórmulas para o cálculo dos modelos codificados pelos especialistas.

Place: classe que representa a localização física dos objetos modelados, permitindo referenciar geograficamente um *Evaluation Object*.

Analysis: classe que representa uma avaliação associada a um *Evaluation Object*. Suas instâncias têm propriedades, como nome e data da avaliação, e correspondem a uma avaliação cadastrada.

Value: classe que representa os valores que são atribuídos a cada instância de *Feature*. As subclasses *Real* e *Categorical* representam tipos de valores.

User: classe que representa os usuários do sistema.

Role: classe que representa os papéis de usuário do sistema e suas permissões. Por padrão, estão instanciados os perfis *User* e *Admin*.

A partir dessas classes é possível organizar os conceitos específicos de cada SAD como sub-classes delas. Um aspecto importante das ontologias é que suportam a inferência de novos conhecimentos, permitindo classificar e relacionar dados novos do sistema, ajudando desta maneira na automatização da geração dos SADs. Isso permite ao *DSL Interpreter* mapear qualquer conceito novo, específico de um SAD em particular, a um desses conceitos conhecidos e saber o que deve ser feito com ele.

4.4 Editor de Ontologias

Para suportar a edição das ontologias do domínio, por parte dos especialistas, foi implementado um editor *web*. Ele permite editar a ontologia específica do SAD SustenA-gro em um formato baseado em YAML, a descrição da ontologia neste formato passa a um conversor para ficar no formato OWL que é processado pela API de *protégé* para instanciar a ontologia com as restrições definidas e finalmente exportado como RDF para ser substituído na *triplestore Blazegraph*, este processo é realizado a cada vez que o usuário salva a ontologia.

A Figura 10 mostra uma imagem do editor. Ela mostra o código da ontologia em formato YAML, um *side panel*, que apresenta as classes, propriedades e indivíduos hierarquicamente (permitindo a referência dos elementos no código), um *button restore* para restituir a ultima versão da ontologia e um *button save* para salvar e carregar no *Framework Decisioner* a ontologia em edição.

Idealmente, especialistas de domínio deveriam poder criar a ontologia usando um editor gráfico. Mas devido as restrições de escopo de um projeto de mestrado, não haveria tempo para criar um. A criação de uma ferramenta, como um editor gráfico, teria o escopo de um novo mestrado.

Usar um editor para OWL, como o *Protégé*, exigiria que os especialistas de domínio aprendessem OWL e lógica descritiva, o que não é uma opção viável. Este editor de ontologias é uma solução intermediária. Ele adota um formato em YAML, menos complexo que OWL, mas expressivo o suficiente para as necessidades das ontologias. Ele ajuda os usuários a encontrar os elementos da ontologia (*side panel*) e permite a inserção da ontologia no *Framework Decisioner*. Apesar de ser possível aos especialistas desenvolver uma ontologia totalmente nova, usando o editor, seu objetivo é permitir que eles possam fazer modificações localizadas nas ontologias.

A ontologia definida com este editor será usada para definir a estrutura dos SADs, as funcionalidades dos SADs são descritas na DSL apresentada a seguir.

Figura 10 – Editor da ontologia SustenAgro

RESTORE
SALVAR

Classes

+ Categorical

+ Boolean

+ EvaluationObject

+ Feature

Properties

+ AnnotationProperty

+ ObjectProperty

+ DataProperty

Individuals

+ Existe/Não existe

+ Sim/Não

+ Categoria do origem da cana-de-acucar

+ Categoria de sustentabilidade

+ Categoria de sistema de produção agrícola

+ Categoria da disponibilização dos resultados da avaliação

```

1 #
2 # Esse arquivo está no formato yaml, nele você tem chaves com valores
3 # associados:
4 # label: description
5 # a string description fica associada a chave label
6 # Quando se quer uma lista de coisas, o - é usado, exemplo:
7 # cidades:
8 #   - New York
9 #   - San Francisco
10 #   - São Paulo
11
12
13 # Nome (URL) que define a sua ontologia
14 ontology: http://semantic.icmc.usp.br/sustenagro#
15
16 # As ontologias que você inclui. No nosso caso vai ser sempre uma
17 #includes:
18 #   - SemanticUI.yaml
19
20 imports:
21   - prefix: ui
22     file: SemanticUI.yaml #http://purl.org/biodiv/semanticUI#
23
24 #OWLImports:
25 #   - prefix: ui
26 #     uri: http://purl.org/biodiv/semanticUI#
27 #     file: SemanticUI.rdf
28
29 prefix:
30   dbp: http://dbpedia.org/ontology/
31
32 map:
33   Provider: http://dbpedia.org/ontology/Provider
34   PhysicalPlant: http://dbpedia.org/resource/PhysicalPlant
35   state: dbp:state
36   State: dbp:State
37   Microregion: http://dbpedia.org/page/Microregion_(Brazil)
38
39 # rdf:type
40
41 #
42 # Properties:
43 #   Relacionam as coisas na ontologia
44 #   Dica:
45 #     type: Functional - quer dizer que algo só pode ter a propriedade
46 #     uma vez: Katia hasCPF 8999999999999 (hasCPF é funcional porque
47 #     cada pessoa só pode ter um CPF
48 #
49
50 relevance:
51   subPropertyOf: AnnotationProperty
52   domain: Indicator
53   label: [relevance @en, relevância @pt]
54
55 hasSugarcaneSource:
56   subPropertyOf: ObjectProperty
57

```

Fonte: Elaborada pelo autor.

4.5 DSL do Framework Decisioner

Para permitir que os especialistas definam o comportamento do *Framework Decisioner*, foi definida uma DSL que permite definir as principais características de um SAD. Tal DSL foi baseada na modelagem geral da arquitetura do *Framework Decisioner* e permite relacionar conceitos específicos dos especialistas, criar equações para os modelos usados, gerar a interface *web* para o usuário final, servindo de interface entre a ontologia do Decisioner e a ontologia de SustenAgro.

A DSL foi desenvolvida na linguagem *Groovy*, porque ela suporta o desenvolvimento de DSLs que se comportam como extensões da linguagem *Groovy*.

O uso da DSL, por parte especialistas, diminui o esforço necessário no desenvolvimento de um SAD. Ela permite que os próprios especialistas sejam capazes de fazer parte do desenvolvimento e validação do SAD. Especialmente na parte de refinamento e atualização do SAD, especialistas podem fazer modificações no sistema sem a ajuda de programadores e ver o resultado dessas mudanças imediatamente.

As instruções definidas na DSL são:

Evaluation Object

Nos SAD focados na avaliação, existe um objeto de avaliação que representa as entidades a serem avaliadas. Esse objeto é constituído por propriedades que especificam o que está sendo avaliado. A instrução *evaluationObject* permite definir as propriedades desse objeto. Por exemplo, caso uma fazenda esteja sendo avaliada, é possível criar propriedades como nome, tipo de produção, localização, etc.

A instrução tem como argumentos a URI (ou *label*) da classe da ontologia do domínio, que será objeto de avaliação, e cada uma das propriedades relacionadas. O código-fonte 4.1 apresenta um exemplo que usa a classe *ProductionUnit*, como classe dos objetos a serem avaliados e define as propriedades *hasName*, para nome, e *hasAgriculturalProductionSystem*, para tipo de produção. Podem ser usados os *labels* das propriedades, ao invés de suas URIs.

Código-fonte 4.1 – Definição do Evaluation Object

```
evaluationObject ":ProductionUnit", {
  instance "ui:hasName", label: ["en": "Name", "pt": "Nome"]
  instance ":hasAgriculturalProductionSystem"
  type label: ["en": "Type", "pt": "Tipo"]
}
```

O comando *instance* vincula uma propriedade definida na ontologia por meio da URI. Ela pode ser complementada por parâmetros que customizam a representação visual da propriedade. O comando *type* faz com que os *EvaluationObject* tenham que ser de subclasses da classe principal. Por exemplo, uma unidade produtiva pode ser uma plantação greenfield (mecanizada e uniforme), fazenda familiar, etc. Os parâmetros que podem complementar as instruções anteriores são:

1. *required*: define uma propriedade obrigatória
2. *label*: define um texto associado
3. *placeholder*: define um texto de ajuda

4. *widget*: define um controle gráfico de usuário

Feature

A instrução *Feature* define as características, do Evaluation Object, que serão usadas na sua avaliação. O *DSL Interpreter* vai gerar uma interface gráfica, onde o usuário final terá que preencher os dados sobre cada característica. Cada característica tem um tipo associado a ela (na ontologia de domínio). A partir dele, é possível associar uma *widget* específica para edição. Por exemplo, a Figura 11 mostra a *widget* para uma característica que tem um tipo categórico (Controle biológico de pragas) com 2 possíveis valores (Sim e Não). Os textos mostrados vêm da ontologia e fazem parte da descrição de cada elemento. É possível criar descrições em mais de um idioma.

Figura 11 – Widget gerada a partir da definição da DSL

Fonte: Elaborada pelo autor.

Quando o usuário final usa a *widget*, o valor escolhido é anotado, no *Evaluation Object*, usando a propriedade *has value*. Nas fórmulas, usadas para os cálculos do modelo usado, é possível acessar esses valores. Os usuários não precisam preencher todas as características.

Código-fonte 4.2 – Definição de Features

```
feature ':EnvironmentalIndicator', 'extraFeatures': true
```

O comando tem como argumento uma URI (ou *label*) que vincula todas as subclasses da classe referenciada (Código-fonte 4.2). O parâmetro opcional *extraFeatures* permite ativar a inserção de novas *features*, por parte do usuário do SAD.

Report

O comando *Report* permite definir as fórmulas e procedimentos matemáticas necessários para o cálculo do modelo usado pelos especialistas de domínio, e como os resultados serão apresentados (Código-fonte 4.3). Fórmulas e procedimentos para modelagem

matemática são de responsabilidade dos especialistas de domínio. A DSL permite desde fórmulas simples, de uma linha, até o uso de bibliotecas complexas, chamadas usando a JVM. Como a DSL é uma extensão da linguagem *Groovy*, qualquer comando da linguagem pode ser usado nela, incluindo operações lógicas e aritméticas. Para facilitar o trabalho dos especialistas de domínio, recomenda-se encapsular qualquer algoritmo ou chamada de função mais complicados em um comando simples.

As fórmulas usadas têm acesso a todos os dados associados às *Features*, pelos usuários finais. Esses dados são usados no modelo adotado e podem gerar múltiplos resultados de avaliação. O Código-fonte 4.3, o comando *weightedSum(data.':EnvironmentalIndicator')* calcula a média ponderada de todos os indicadores ambientais fornecidos pelo usuário final. Esse comando foi criado para simplificar o trabalho dos especialistas. Biblioteca de comandos, como essa, podem ser adicionadas ao *Framework Decisioner* ou criados a pedido dos especialistas.

Código-fonte 4.3 – Definição da lógica de avaliação.

```
report {
  environment = weightedSum(data.':EnvironmentalIndicator')
  economic = weightedSum(data.':EconomicIndicator')
  social = weightedSum(data.':SocialIndicator')
  sustainability = (environment + social + economic)/3
  ...
}
```

Os resultados do processo de avaliação podem ser apresentados por meio de várias *widgets* que facilitam a representação e compreensão dos resultados da avaliação. No algoritmo 4.4 o comando *sustainabilityMatrix x: sustainability, y: efficiency* apresenta os valores das variáveis *sustainability* e *efficiency* em um gráfico de matriz de sustentabilidade (Figura 33). Para executar esse comando, o *DSL Interpreter* simplesmente coloca a *widget sustainabilityMatrix* na UI e passa os valores das variáveis, como atributos. A *widget* vai ser responsável por criar o gráfico. Essas *widgets* gráficas podem ser criadas como *web components* padrão (HTML 5) ou componentes do *Framework Grails* (Seção 4.7). O *Framework Decisioner* vem com um conjunto de *widgets* predefinidos, mas novas podem ser adicionadas.

Código-fonte 4.4 – Definição dos componentes visuais do relatório.

```
report {  
  ...  
  sustainabilityMatrix x: sustainability, y: efficiency  
  text 'en': 'Microregion map', 'pt': 'Mapa da microregião'  
  map data.'Microregion'  
}
```

Por meio dos comandos da DSL, é possível definir o comportamento e as características gerais dos SAD.

Os elementos gráficos (*widgets*), seja os que representam as Features ou os usados nos relatórios, são implementados como *Web Components* HTML 5 ou Grails. Isso dá muita flexibilidade ao *Framework Decisioner*. A qualquer tempo é possível se acrescentar novas *widgets*, para novos tipos de dados (*features*) ou gráficos de relatórios.

4.6 Editor da DSL

Para suportar a edição da *Decisioner DSL*, foi implementado um editor textual web que permite aos especialistas de sustentabilidade editar e rodar o código da DSL. Ele é composto por um editor de código para a DSL, um *button restore* para carregar novamente a última versão válida da DSL e um *button save* para salvar e carregar a DSL no *DSL Interpreter*.

O Editor de código foi baseado no *Ace Editor*¹, que fornece *syntax highlighting* para várias linguagens de programação, entre elas *Groovy*. Como a DSL foi baseada em *Groovy*, o editor foi configurado para reconhecer a sintaxe de *Groovy*. Também foi ativado o suporte para *code completion* para fornecer uma experiência de uso mais amigável.

A principal vantagem desse editor é a funcionalidade de salvar a ontologia e reconfigurar o SAD imediatamente, permitindo uma experiência em tempo real de redefinição do SAD. Se o usuário errar algo, ele pode simplesmente restaurar as configurações *default*.

O editor da DSL tem compatibilidade com *web components*, tanto os fornecidos pelas bibliotecas integradas no *Framework Decisioner*, como com os *Web Components* do *Framework Grails* e, adicionalmente, podem ser definidos Web Components especializados.

A Figura 12 mostra o editor em ação.

¹ <<https://ace.c9.io/>>

Figura 12 – Editor da DSL



Fonte: Elaborada pelo autor.

4.7 Web Components

*Web Components*² é um conjunto de APIs padrão para definir novas tags HTML personalizadas, reutilizáveis e encapsuladas para o uso em páginas ou aplicações *web*. O *Framework Grails* também disponibiliza o uso de *layout templates* para implementar partes reusáveis de uma *view* (página HTML). Com essas duas tecnologias, é possível a criação de componentes (ou *widgets*) reusáveis da UI.

Para suportar a geração de SADs de tipo avaliação, foi necessário disponibilizar vários tipos de *widgets* que permitissem visualizar e editar diferentes tipos de dados. Para relacionar os *web components* com o conhecimento dos especialistas, modelou-se na ontologia Decisioner, os *data-types* que permitem relacionar os tipos de dados com *widgets* específicas (capazes de editá-los), usadas nas *web UIs* dos SADs gerados. Os dados das *features* dos SADs podem ser de vários tipos e, para cada tipo, existe uma *widget* apropriada

² <<https://www.webcomponents.org/introduction>>

para visualiza-o. Por exemplo, para representar uma propriedade de tipo numérico discreto é possível usar uma widget visual tipo *spinner*.

Nos relatórios, os especialistas devem contar com *widgets* para apresentar seus resultados em vários formatos, como tabelas, mapas, matriz de sustentabilidade, etc. Essas *widgets* podem ser específicas para um tipo de SAD. Por isso, além do *Framework Decisioner* contar com uma biblioteca de *widgets* prontas, deve ser possível adicionar novas *widgets* facilmente. Ao usar padrões, como *web components*, o *Framework Decisioner* permite a fácil inclusão de *widgets* novas. Não se espera que os especialistas de domínio criem essas *widgets*, mas sim que seja fácil para eles consegui-las de desenvolvedores independentes (que só precisam conhecer o padrão para *Web Components* e os requisitos da *widget*).

O *Framework Decisioner* tem uma biblioteca de *web components* construída usando o *web Framework Bootstrap*³, que conta com diversos componentes básicos para a geração das *Web UI*. A maioria deles foi definida usando as *layout templates* do *Framework Grails*, por ser mais fácil de programar. Atualmente, apenas dois componentes usam *Web Components*.

4.8 Web UI

A *Web UI* é uma interface *web* de usuário responsável por toda a interação com o usuário final. Ela permite apresentar e editar as informações do SAD, gerar as análises e visualizar os resultados na *web* ou em relatórios impressos. O mais importante é que ela é gerada automaticamente pelo *DSL Interpreter*, usando a DSL e ontologias particulares a cada SAD.

A Figura 13 apresenta uma *Web UI* que foi gerada automaticamente, a partir da ontologia e das definições na DSL do SAD SustenAgro. Nela são apresentados vários tipos de *Web Components* que demonstram o suporte a diferentes tipos de dados.

Parte da *Web UI* é igual para todos os SADs, como formulários de *login*. O resto dela é gerado com a ajuda dos *Web Components*. Eles são relacionadas aos tipos de dados existentes no sistema, por meio da ontologia *Decisioner*. Por exemplo, dados podem ser dos tipos numérico contínuo, numérico discreto, percentagem, booleano, categóricos ou alfanumérico. Dada essa diversidade, os tipos de dados foram modelados em ontologias com a finalidade de permitir a adaptação automática (ou semiautomática) da interface às mudanças dos conceitos do domínio.

³ <<http://getbootstrap.com/>>

Figura 13 – Web UI com web components

Register new production unit / farm:
[Select production unit / farm](#)

Production unit or farm name*

Name

Agricultural production system*

	Options
<input checked="" type="radio"/>	Sugarcane production system

Production unit type*

	Options
<input type="checkbox"/>	Provider
<input type="checkbox"/>	Plant

Sugarcane source*

	Options
<input type="radio"/>	Lease origin
<input type="radio"/>	Own origin
<input type="radio"/>	Supplier origin

Availability of evaluation results*

	Options
<input type="radio"/>	Private
<input type="radio"/>	Public

Register

Fonte: Elaborada pelo autor.

Mudanças no layout geral podem ser feitas também por meio da edição das *Cascading Style Sheets* (CSS) do *Framework Decisioner*. É esperado que, quando da instalação do sistema, técnicos de informática façam uma customização do sistema para adequá-lo aos padrões de apresentação da instituição. Eles também devem incluir qualquer *Web Component* (*widget*) necessário mas não disponível por *default* no *Framework Decisioner*.

4.9 DSL Interpreter

O *DSL Interpreter* é o principal módulo do *Framework Decisioner*. A sua principal funcionalidade é a interpretação das DSLs. Ele executa cada uma das instruções da DSL, vinculando dados e informação, das ontologias ou fornecidos pelos usuários finais, aos *web components* com a finalidade de gerar as *Web UIs*.

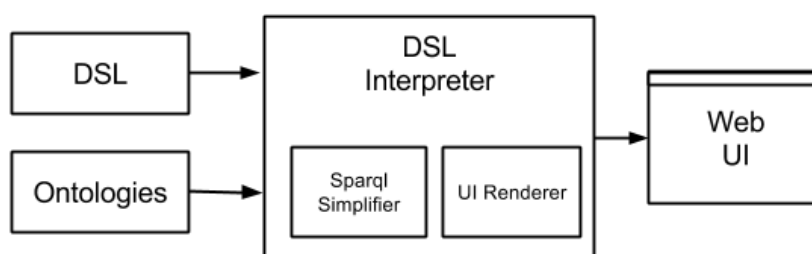
Para interagir com os dados da ontologia e apresentá-los na linguagem dos especialistas, foi necessário criar, no *DSL Interpreter*, uma camada de consulta de dados na *triple-store* para simplificar as consultas SPARQL (Figura 14). Esta camada, *Sparql Simplifier*, foi

desenvolvida como uma solução eficiente para a recuperação de informações semânticas de um domínio de conhecimento usando um formato simplificado.

O componente *DSL Interpreter* processa as instruções dos especialistas na linguagem Decisioner DSL. Foram utilizadas técnicas padrões para a criação de DSLs usando a linguagem *Groovy* (DEARLE, 2015), o que facilitou a definição das DSLs.

Finalmente, o *DSL Interpreter* usa o *UI Renderer* para renderizar dinamicamente as *Web UIs* usando os *Web Components*. Esse processo é repetido toda vez que usuários solicitam uma *View* de um SAD. A Figura 14 apresenta o *DSL Interpreter* e os outros módulos com os quais ele se conecta.

Figura 14 – Arquitetura do *DSL Interpreter*.



Fonte: Elaborada pelo autor.

A Figura 14 também representa a parte computacional do método proposto nesta pesquisa para definir SADs. O método consiste em representar o conhecimento dos especialistas em ontologias da *web* semântica e complementar com uma DSL que descreve os conceitos de *evaluation object* com *features*, método de avaliação e relatório de resultados, permitindo desta maneira definir o SAD por parte dos especialistas.

4.10 Considerações finais

Neste capítulo foram apresentados os principais componentes do *Framework Decisioner*, existem características dele como o gerenciamento de usuários e segurança que foram desenvolvidos, mas não foram especificados porque não contribuíram de maneira relevante ao desenvolvimento da pesquisa.

O desenvolvimento do *Framework Decisioner* foi realizado simultaneamente com o SAD SustenAgro, devido a que era necessário uma instancia para validar se as funcionalidades foram implementadas corretamente. O processo teve dificuldades para separar os dois desenvolvimentos pois tinha componentes em comum, que a partir de um processo iterativo, foram organizando-se e permitindo separar os dois sistemas.

Não foi possível instanciar outro sistema no *Framework Decisioner* durante o desenvolvimento do mestrado, mas atualmente encontra-se em andamento a implementação do SAD Nano-Tec como segunda instanciação do *Framework Decisioner*. O que permitirá

generalizar ainda mais e servir como um caso de uso adicional para validar as hipóteses em outros domínios. Esta arquitetura foi validada por meio da instanciação do SAD SustenAgro, o que permite validar se realmente funcionam cada uma das características aqui descritas.

No capítulo seguinte será apresentado o SAD SustenAgro, gerado como primeiro caso de uso deste *Framework*, e depois será apresentado o processo de avaliação que formaliza a avaliação deste sistema.

SAD SustenAgro

O *Framework Decisioner* organiza e gerencia os componentes gerais dos SADs de avaliação. Cada SAD tem particularidades que precisam ser definidas e ajustadas para configurar as funcionalidades. As ontologias específicas do domínio e as DSL, explicadas nos capítulos anteriores fornecem o meio de definição dessas particularidades.

O SAD SustenAgro foi usado como a primeira instanciação do *Framework Decisioner*. Ele é composto por uma ontologia de domínio, DSL e elementos gráficos de *design* (ícones, imagens de fundo, etc.) que permitem instanciá-lo no *Decisioner*. Sendo seu principal componente a ontologia do domínio de avaliação da sustentabilidade da produção de cana-de-açúcar na região centro-sul do Brasil.

Neste capítulo, serão explicadas a arquitetura, a metodologia e os componentes mais importantes do SAD SustenAgro. Serão feitas também algumas considerações sobre o processo de definição do SAD.

5.1 Arquitetura do SustenAgro

O SAD SustenAgro serviu de base para modelar e desenvolver os componentes que fazem parte do *Framework Decisioner*. Por isso, o processo real de desenvolvimento dele foi muito mais complicado que uma simples instanciação de um *framework*. Ele envolveu diversas iterações para determinar o que deveria ser implementado como parte do *Framework Decisioner* e como parte do SAD. Para simplificar o texto e facilitar o entendimento de como o *Framework Decisioner* é usado para instanciar um SAD, as diversas versões do *framework* não serão discutidas.

A arquitetura de um SAD implementado usando o *Framework Decisioner* tem os seguintes componentes:

1. Ontologia do domínio: ontologia que representa os conceitos do domínio. No caso do SustenAgro, o domínio é a avaliação da sustentabilidade do sistema produtivo de

- cana-de-açúcar, na região centro-sul. Essa ontologia é a base para o SAD pois permite estabelecer os conceitos fundamentais, que são utilizados pelo sistema. No caso do SustenAgro, eles são: indicadores, componentes de indicadores, índices, dimensões da sustentabilidade, recomendações e o método de avaliação.
2. DSL: programa descrevendo a configuração e comportamento do SAD. Ele especifica as *features* do domínio a serem usadas, as fórmulas do modelo e o aspecto e estrutura do relatório a ser gerado. No caso do SustenAgro, as *features* são os indicadores, as formulas calculam os índices de sustentabilidade e produtividade, e o relatório usa a Matriz e o Semáforo de Sustentabilidade.
 3. *Web components*: o SustenAgro usa dois *Web Components* específicos (além das fornecidas por padrão): a Matriz de Sustentabilidade e o Semáforo de Sustentabilidade. Ambas são implementadas usando a biblioteca *Polymer* da *Google*.
 4. *Imagens e layout*: Um conjunto de imagens e arquivos de *layout* (css) compõem o *look-and-feel* específico do SAD, incluindo o logo de cada SAD.

5.2 Metodologia

O conhecimento do domínio abrangido no sistema SustenAgro está em contínua evolução. Por isso, foi necessário usar uma metodologia que suporte mudanças na estrutura e nos dados do sistema, durante cada uma das fases do desenvolvimento. O desenvolvimento da ontologia de domínio SustenAgro foi realizada de forma ágil e modular, por meio de técnicas de prototipação rápida, abrangendo grupos de conceitos relacionados entre si.

O desenvolvimento da ontologia depende essencialmente da comunicação entre os especialistas de domínio e os modeladores. Dessa forma, foram definidos meios de comunicação (reuniões presenciais e virtuais) e de representação do conhecimento (modelos conceituais), que permitiram explorar o domínio.

Um dos meios, que permitiu uma melhor comunicação, foi o desenvolvimento de um mapa conceitual, por meio da ferramenta *Cmap Tools*¹, com a participação de um grupo de especialistas em modelagem de conhecimento. Esse processo começou em uma reunião da equipe na Embrapa Informática Agropecuária (situada na Universidade Estadual de Campinas, UNICAMP). Nessa reunião, um especialista em desenvolvimento de ontologias da Embrapa forneceu treinamento sobre a metodologia para definir ontologias, desenvolvimento de mapas conceituais, com os principais conceitos, e desenvolvimento de modelos em OWL, para tornar esse conhecimento computável.

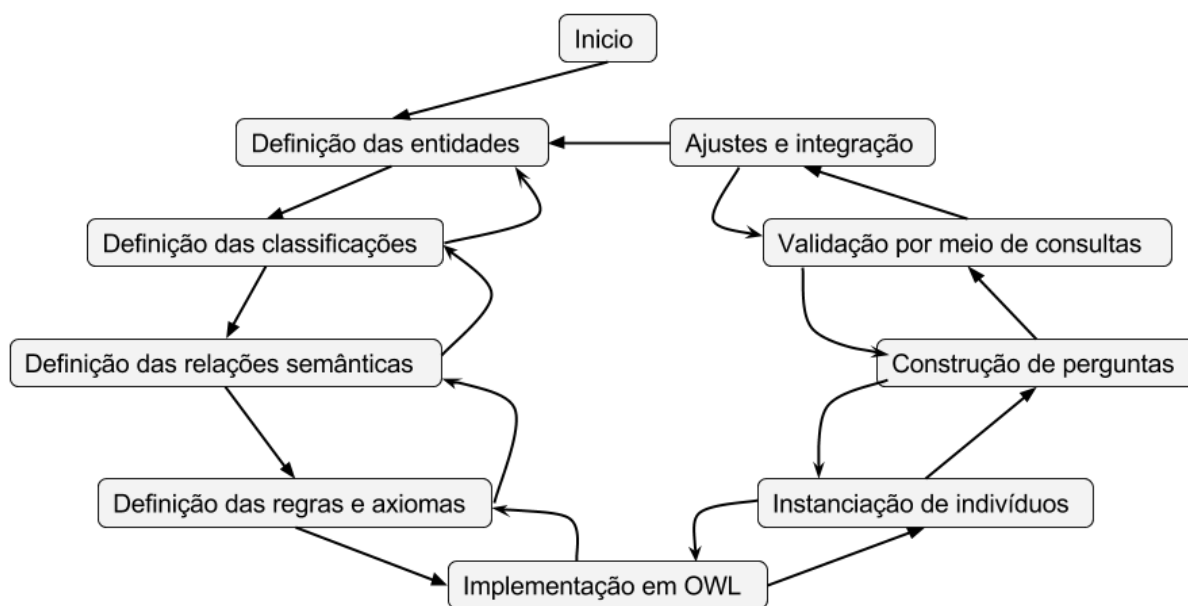
Após realizada a modelagem, o especialista do domínio definiu perguntas de interesse, com as quais os modeladores (o autor e um colega de mestrado) definiram consultas

¹ <http://cmap.ihmc.us/>

que o sistema deveria responder, a partir dos resultados resultantes foi validado e ajustado o modelo até ter um protótipo confiável.

Na Figura 15 é apresentada a metodologia para desenvolver a ontologia SustenA-gro, a qual teve vários ciclos de desenvolvimento nos quais foram integradas novas características.

Figura 15 – Metodologia de definição da ontologia SustenAgro.



Fonte: Elaborada pelo autor.

Um aspecto importante das ontologias é que elas fornecem um formato que adapta-se às mudanças do domínio e permite separar o conhecimento dos especialistas dos outros componentes do sistema.

A metodologia, que direcionou o desenvolvimento do SAD SustenAgro, foi a SCRUM (SCHWABER; BEEDLE, 2002), que permitiu integrar práticas ágeis no desenvolvimento do sistema. Nesse contexto, o termo ágil refere-se ao desenvolvimento em tempos curtos e geração de protótipos facilmente adaptáveis às mudanças. Cada uma das etapas da metodologia foi realizada várias vezes e, por isso, foi necessário redesenhar os componentes. As metodologias ágeis são cíclicas e os protótipos mudam em cada ciclo para cumprir os novos requisitos.

A metodologia de desenvolvimento dos *Web Components* e das *Web UI* tiveram um enfoque baseado em *User Centered Design*. A avaliação foi realizada integradamente para validar os requisitos.

O processo de design das *web UI* incluiu as seguintes etapas de levantamento de requisitos:

1. Descrição de *User Stories*: técnica de desenvolvimento ágil que permite descrever características do *software* desde a perspectiva do usuário. Ela fornece uma identificação dos usuários, das funcionalidades e explica o porque cada funcionalidade é necessária.
2. Descrição de *Scenarios*: técnica de desenvolvimento ágil que permite descrever detalhadamente as características das *user stories*.
3. Descrição de *Storyboards*: descreve cada uma das interações do usuário com o sistema em uma determinada tarefa, visualizando a interação como uma história em quadros.
4. Descrição de *Mockups*: *design* do esboço da interface gráfica do sistema. Eles foram analisados pelos especialistas da Embrapa, para avaliar se atendiam às funcionalidades básicas descritas no levantamento dos requisitos.
5. Desenvolvimento de protótipo visual: A partir da validação dos *Mockups*, foi desenvolvido um protótipo da interface gráfica, com a finalidade de que os especialistas do domínio avaliassem se as interfaces cumprem os requisitos.

Cada uma dessas etapas de desenvolvimento, foram realizadas sempre em parceria com os especialistas do domínio. Isso foi importante para realizar o levantamento correto de requisitos tanto das *web UI* como das funcionalidades do SAD, identificadas a partir destas técnicas. Todo esse processo foi necessário pois não existia uma definição específica do que os especialistas precisavam.

5.3 Ontologia de domínio: SustenAgro

A ontologia SustenAgro representa o conhecimento necessário para suportar avaliação de sustentabilidade no sistema produtivo de cana-de-açúcar na região centro-sul do Brasil. Ela representa conceitos por meio de entidades, classes, relações semânticas e axiomas. Esses elementos organizam e representam a realidade modelada.

Para definir a ontologia SustenAgro, realizou-se uma pesquisa das fontes de dados relacionadas com ontologias do domínio de avaliação de sustentabilidade em sistemas produtivos de cana-de-açúcar. Concluiu-se que não existem ontologias que suportem esse domínio. Por isso, propôs-se desenvolver uma ontologia que utilizasse conceitos sobre avaliação de sustentabilidade e sistemas agrícolas. Essa ontologia representa conceitos gerais sobre sustentabilidade e os conceitos particulares do SAD SustenAgro, identificados por Cardoso (2013).

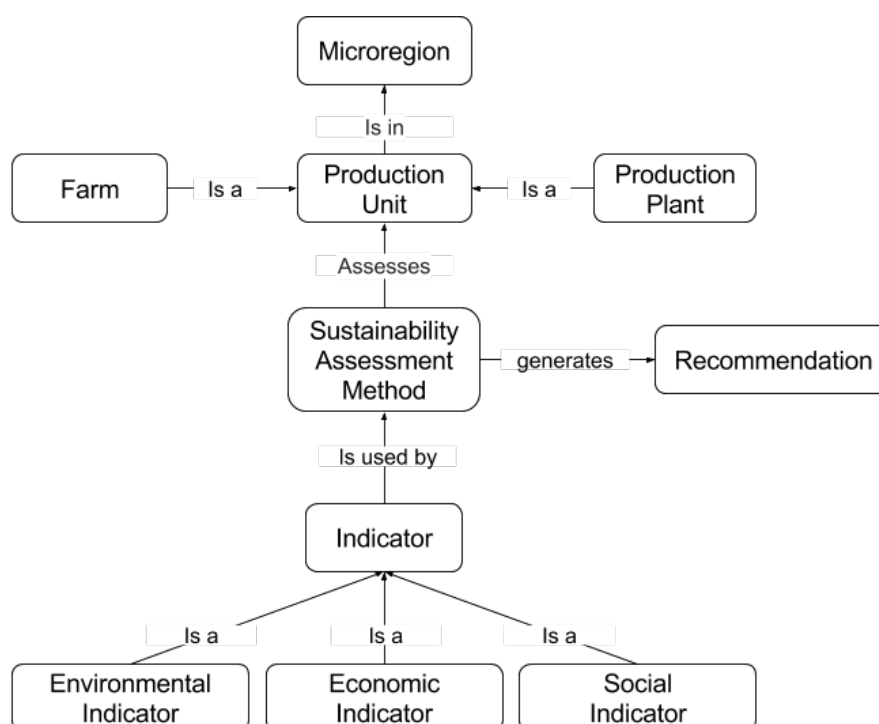
Deve-se destacar que a ontologia SustenAgro abrange um domínio bem específico; sustentabilidade de sistemas produtivos de cana-de-açúcar na região centro-sul do Brasil. Acreditamos que essa é uma característica deste tipo de SAD. Modelagens desse tipo tendem

a ser específicos. No caso do SustenAgro, ele abrange apenas um só sistema produtivo em uma região específica.

A Figura 16 representa um mapa conceitual com os principais conceitos modelados no SAD SustenAgro e como eles estão relacionados entre si. As etiquetas, em cada relação dos conceitos, permitem identificar a relação entre os dois conceitos.

As ontologias da web semântica permitem separar o conhecimento do domínio da lógica da computação, independizando o desenvolvimento destes aspectos do SAD.

Figura 16 – Mapa conceitual da ontologia SustenAgro.



Fonte: Elaborada pelo autor.

Para conseguir que a ontologia SustenAgro fosse computável, foi necessário defini-la na linguagem OWL (fornecendo uma representação compreensível por humanos e computadores). O editor de ontologias Protégé² foi usado, permitindo modelar o conhecimento e exportar ele ao formato RDF, para ser compatível com sistemas *triple-stores* (Seção 3.1) (ALLEMANG; HENDLER, 2011).

Triplestores suportam a realização de consultas complexas para permitir a resposta a perguntas de interesse aos usuários do sistema. Essa característica permite a recuperação da informação com significado semântico, permitindo que os sistemas deem respostas às consultas complexas de interesse. Também o RDF suporta a integração de conhecimento externo existente em formatos da web semântica, como são vários sistemas de representação

² <<http://protege.stanford.edu/>>

do conhecimento como dicionários, *thesaurus* e redes semânticas, o que permite aumentar as possibilidades de desenvolvimento de novas funcionalidades.

A ontologia do SustenAgro modela o conhecimento dos especialistas baseando-se na ontologia do Decisioner (Figura 9). Ela inclui a ontologia Decisioner e define conceitos gerais dos SADs. Isso é obrigatório, pois o código do *Framework Decisioner* entende os conceitos da ontologia SustenAgro apenas porque eles também são conceitos válidos derivados da ontologia Decisioner. Exemplos de conceitos/classes modelados são: *Production Unit*, *Microregion*, *Indicator*, *Categorical*. A partir dessas classes foi possível desenvolver o modelo de dados em OWL.

As classes da ontologia SustenAgro são relacionados por meio de *Object Properties* e *Data Properties* que permitem vincular semanticamente as instâncias das classes. A principal contribuição da ontologia do domínio é ser uma representação semântica do conhecimento de domínio, tanto para os usuários como para o sistema computacional, tornando-se um meio de comunicação entre os especialistas de domínio e os programadores.

Nas próximas seções, serão apresentadas as principais classes modeladas na ontologia do domínio de avaliação da sustentabilidade (SustenAgro).

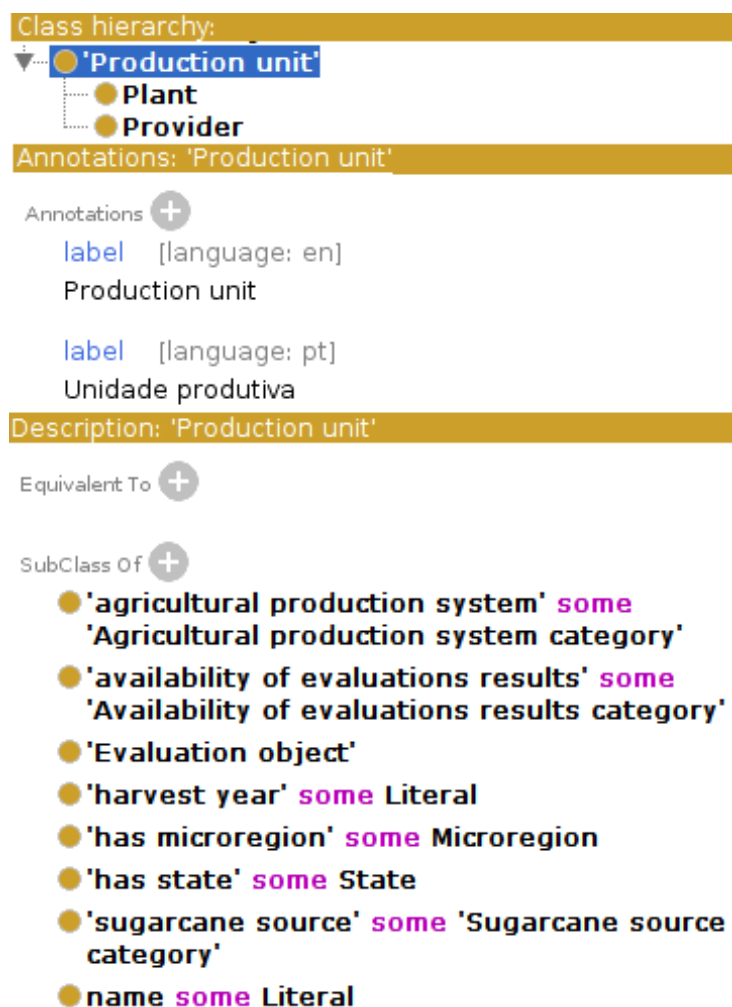
Classe *Production Unit*

Representa as organizações que podem ser avaliadas pelo sistema SustenAgro. Atualmente elas podem ser *Fornecedores de cana-de-açúcar* e / ou *Usinas processadoras de cana-de-açúcar*. Cada processo de avaliação requer dados que identifiquem as unidades produtivas por meio de propriedades que as definam.

Esta classe tem propriedades obrigatórias como:

- *hasAgriculturalProductionSystem*: relaciona o sistema de produção agrícola em avaliação.
- *hasAvailabilityOfEvaluationResults*: relaciona o tipo de disponibilização dos resultados.
- *hasSugarcaneSource*: relaciona a origem da cana.
- *harvestYear*: define o ano da safra.
- *canavialLongevity*: define a longevidade do canavial.

A Figura 17 apresenta a modelagem da classe *Production Unit*, feita na ferramenta Protégé.

Figura 17 – Modelagem da classe de unidade produtiva (*ProductionUnit*).

Fonte: Elaborada pelo autor.

Subclasse *Categorical*

A classe *Categorical* representa os possíveis *Value* discretos que um *Indicator* pode ter na forma de categorias (por exemplo, Existe e Não Existe). Um *Value* também pode ser *Real* ou *Integer*. A classe *Categorical* é definida na ontologia Decisioner, mas a SustenAgro cria diversas classes filhas para definir uma série de valores categóricos.

Cada subclasse de *Categorical* é composta por um conjunto finito de elementos ou valores. Cada valor é modelado como indivíduo da classe, permitindo assim, restringir as opções de instanciação de cada indicador.

Na Figura 18, é apresentada a classe *Value* e suas subclasses, tanto *Categorical* para conjunto finito de valores e *Real* para valores numéricos. Um exemplo de classe categórica seria a *Yes/No* que representa os valores de sim e não e é composta pela lista de instâncias *Yes* e *No*.

Cada indivíduo da classe *Value* tem a propriedade *as number* que relaciona a ele

um valor numérico. Esse valor define um critério de comparação entre os indivíduos da mesma classe. Ele é usado nas fórmulas do método de avaliação.

Figura 18 – Modelagem de Value



Fonte: Elaborada pelo autor.

Classe *Microregion*

Representam os locais onde são localizadas as unidades produtivas. É permitido definir a microrregião onde as fazendas e usinas do sistema produtivo de cana-de-açúcar se localizam. Atualmente, a ontologia tem os 7 estados pertencentes ao centro-sul do Brasil e as 243 microrregiões dentro desses estados. Esses dados foram originalmente obtidos por consulta SPARQL à *DBpedia* e integrados à ontologia.

A Figura 19 mostra a modelagem das localizações geográficas usadas no sistema SustenAgro, com algumas instâncias de *Microregion*.

Figura 19 – Modelagem de microrregiões.



Fonte: Elaborada pelo autor.

Classe *Indicator*

Os indicadores são o principal componente da ontologia. Eles foram propostos por um grupo de especialistas de diversas áreas da produção agrícola e sustentabilidade (CARDOSO, 2013).

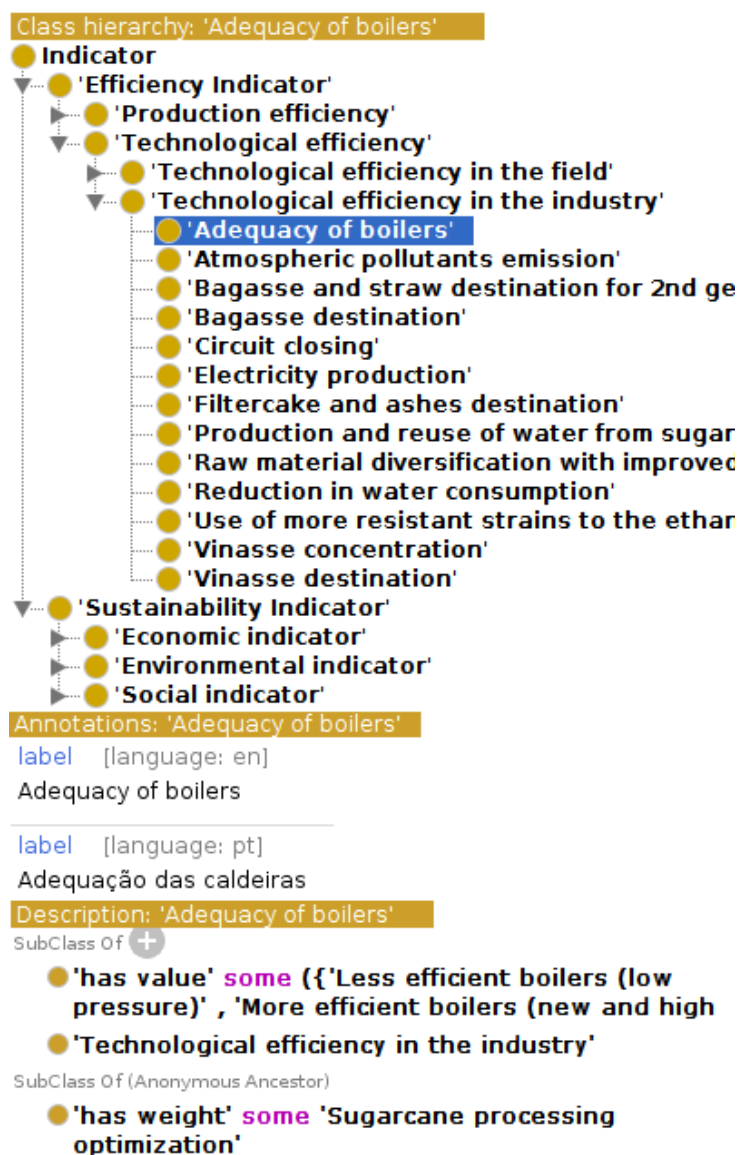
Eles representam as características das unidades produtivas que serão identificadas e quantificadas em cada processo de avaliação. Eles têm uma propriedade *has value* que quantifica a sustentabilidade do indicador. Também permitem a integração de conceitos, inclusive quando pertencem a domínios sem relação aparente.

A Figura 20 apresenta a hierarquia dos indicadores, que está subdividida em *Efficiency Indicator* e *Sustainability Indicator*. A propriedade *has value* usa instâncias da classe *Value* para quantificar a sustentabilidade. Instâncias de *Value* representam as opções de resposta que os usuários têm para cada indicador. Existe outra propriedade, *has weight*, que é opcional e estabelece um peso para o indicador.

A Figura 20 mostra o indicador intitulado *Adequacy of boilers*. Na propriedade *has value* ele tem uma restrição que limita os valores dessa propriedade a valores de uma lista

de valores categóricos. A propriedade *has weight* também tem uma restrição que limita seus valores a instâncias da classe *Sugarcane process Optimization*.

Figura 20 – Modelagem de indicador



Fonte: Elaborada pelo autor.

Segundo a descrição do método de avaliação (Apêndice A), os indicadores de sustentabilidade são classificados em três dimensões de sustentabilidade: dimensão ambiental, dimensão social e dimensão econômica. Tendo as três uma participação equitativa no método de avaliação (KRAINES; GUO, 2011).

A seguir serão apresentados cada uma das três dimensões.

Dimensão ambiental

A dimensão de indicadores ambientais, é composta dos seguintes conceitos:

- Atributo solo (*Soil Attribute*): indicadores que avaliam os aspectos referentes às características do solo.
- Atributo hídrico (*Hydric Attribute*): indicadores que avaliam os aspectos referentes à disponibilidade e qualidade das fontes hídricas.
- Atributo clima (*Weather Attribute*): indicadores que avaliam os aspectos climáticos.

Dimensão social

Nesta dimensão é importante reconhecer que as unidades produtivas, sejam do tipo fazendas ou usinas, têm vínculos com pessoas, tanto internamente como externamente. Por isso, é importante refinar os indicadores para incluir a população externa à unidade produtiva que é afetada pelas práticas produtivas.

A Agência Paulista de Tecnologia dos Agronegócios (APTA) forneceu dados econômicos das principais usinas do estado de São Paulo, que permitiram definir a dimensão econômica das unidades produtivas na ontologia de domínio.

A dimensão social está composta dos seguintes conceitos:

- Atributo emprego e renda (*Employment and Income Attribute*): indicadores que avaliam os aspectos referentes à mão de obra.
- Atributo saúde (*Health Attribute*): indicadores que avaliam os aspectos de segurança dos trabalhadores.
- Atributo treinamento (*Training Attribute*): indicadores que avaliam os aspectos da capacitação dos trabalhadores.

Dimensão econômica

Esta dimensão está composta pelos seguintes conceitos:

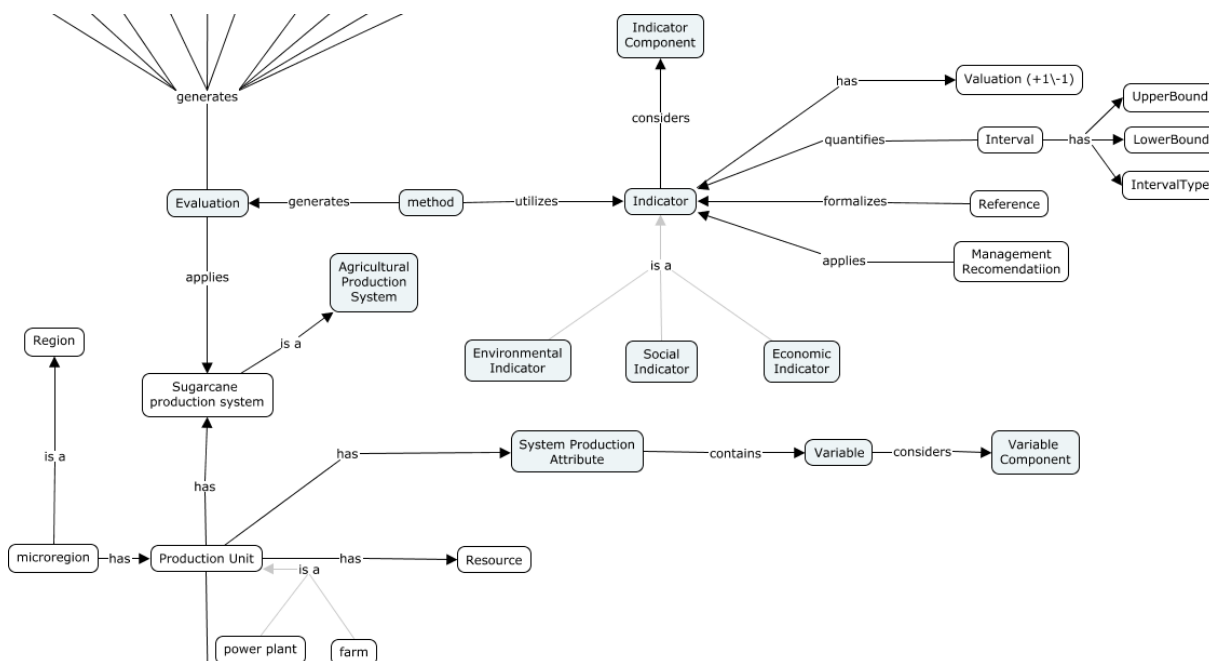
- Atributo industrial (*Industrial Attribute*): indicadores que avaliam os aspectos industriais.
- Atributo área recuperada (*Recovered Area Attribute*): indicadores que avaliam os aspectos da área produtiva e das técnicas produtivas.
- Atributo produtividade (*Productivity Attribute*): indicadores que avaliam os aspectos dos produtos e dos processos produtivos.
- Atributo custo (*Cost Attribute*): indicadores que avaliam os aspectos dos custos da produção.

Cada uma das três dimensões deve ser avaliada equitativamente para gerar um resultado coerente com a teoria da sustentabilidade agrícola (TILMAN et al., 2002).

Modelagem do método de avaliação SustenAgro

A Figura 21 mostra um mapa conceitual dos conceitos envolvidos na avaliação da sustentabilidade, que recebe como entrada os indicadores selecionados pelos usuários e aplica as fórmulas de avaliação, para gerar os índices de sustentabilidade e eficiência. Usando esses índices, é possível quantificar a sustentabilidade.

Figura 21 – Mapa conceitual - Método de Avaliação.



Fonte: Elaborada pelo autor.

As dimensões da sustentabilidade permitiram organizar os indicadores, levando essa organização desde os modelos de mapas conceituais, às ontologias, método de avaliação e, finalmente, até a representação dos resultados nas *web UI* dos SADs.

5.4 SustenAgro Web UI

O *design* das interfaces gráficas e desenvolvimento dos *web components* foram realizados por meio de várias técnicas de levantamento de requisitos para especificar as funcionalidades que os especialistas precisavam do sistema SustenAgro. Foram usadas as seguintes técnicas: *User Stories*, *Scenarios*, *Storyboard*, *Mockups* e protótipo de interface gráfica.

Na fase inicial, foram definidos os perfis de usuários do SAD SustenAgro, inicialmente definiram-se os perfis descritos a seguir:

- **Administrador:** usuário com permissões para editar Ontologias, DSL e *web UI*. Ele é o responsável pela administração do SAD e tem todas as permissões do sistema.

- Especialista de domínio: especialista em sustentabilidade ou afins, com permissões para recuperar e gerenciar informações das avaliações, gerar reportes e mudar os conceitos relacionados com os indicadores e método de avaliação.
- Usuário final: usuário padrão do sistema que tem permissões para realizar avaliações de sustentabilidade em cana-de-açúcar e de gerenciar os dados cadastrados por ele.

No SAD SustenAgro v1.0, foi removido o perfil especialista, ficando apenas os perfis administrador e usuário final. As funções desse perfil foram incluídas no perfil administrador.

As técnicas realizadas para desenvolver as *web UI*, são descritas a seguir.

User Stories

Histórias de usuário são uma técnica para descrever, de uma forma curta e simples, as características do sistema a partir da perspectiva do usuário ou cliente do sistema, gerando uma definição de alto nível de um requisito. O padrão é: como um “tipo de usuário”, quero atingir “algum objetivo” para “alguma finalidade”.

Na aplicação dessa técnica foram obtidas as seguintes histórias:

1. O usuário poderá identificar e cadastrar a localização geográfica e a área da sua lavoura (definir região geográfica do IBGE, latitude e longitude - a partir do *Google Maps*).
2. O usuário poderá identificar e cadastrar a microrregião a que pertence a sua lavoura. O sistema fará uma sugestão de cadastro a partir dos dados da localização geográfica.
3. O usuário deverá preencher o estado de cada indicador específico nas dimensões ambiental, econômica e social, devendo adaptar-se às condições das regiões e microrregiões do Brasil.
4. O usuário poderá obter o resultado dos índices, segundo a informação preenchida e a fórmula de agregação dos indicadores.
5. O usuário poderá armazenar a informação dos indicadores para futuras consultas.
6. O usuário poderá acrescentar indicadores que considere importantes para a análise. Deve-se estabelecer regras para essa funcionalidade de tal modo que os novos indicadores (criados pelos usuários) sejam recuperáveis de um modo separado dos indicadores cadastrados no sistema.
7. O sistema deve fornecer um cronograma de avaliação, sendo recomendado realizar a avaliação depois de cada safra.

Scenários

É uma técnica que permite a descrição das funcionalidades do sistema desde a perspectiva do usuário ou cliente, realizando uma descrição detalhada de cada um dos passos dos usuários no sistema para completar uma tarefa. A seguir serão apresentadas as 8 histórias de usuários do SAD SustenAgro com os cenários associados:

História de usuário #1: “O usuário poderá identificar e cadastrar a localização geográfica e a área da sua lavoura (definir região geográfica do IBGE, latitude e longitude - a partir do *Google Maps*).”

1. O usuário ingressa na conta dele por meio do sistema *web* SustenAgro em <http://sustenagro.embrapa.br>, e o sistema apresenta a tela “*Home*”
2. O usuário seleciona a aba “unidades produtivas” e dá um *click* em “cadastrar unidade produtiva”, o sistema apresenta a tela de cadastro de unidades produtivas, onde tem um mapa do *Google Maps*.
3. O usuário seleciona no mapa um ponto que identificará a localização da unidade produtiva, se ele quiser, também é possível marcar a área da lavoura para que o sistema possa ter dados mais específicos para o processo de avaliação de sustentabilidade. Uma vez terminado, o usuário dá um *click* no botão “próximo” e o sistema cadastra a informação preenchida.

História de usuário #2: “O usuário poderá identificar e cadastrar a microrregião a que pertence a unidade produtiva dele, por meio de uma sugestão que o sistema faz com os dados da localização geográfica.”

1. O usuário poderá fazer a “História de usuário #1” ou entrar no sistema e continuar com o cadastro da unidade produtiva de onde ele tenha parado. O sistema apresentará uma tela com sugestões de microrregiões.
2. O usuário poderá escolher a microrregião, onde esteja localizada a unidade produtiva, e salvá-la no sistema por meio do botão “próximo”.

História de usuário #3: “O usuário deverá preencher o estado de cada indicador específico nas dimensões ambiental, econômica e social. Esses indicadores devem adaptar-se às condições das regiões e microrregiões do Brasil, da mesma forma as faixas de limiares de sustentabilidade foram definidas.”

1. O usuário poderá fazer a “História de usuário #2” ou entrar no sistema e continuar com o cadastro dos indicadores de onde ele tenha parado. O sistema apresentará uma tela com três abas que contém os controles que permitirão fazer o cadastro dos indicadores nas dimensões ambiental, econômica e social.

2. O usuário dá um *click* na primeira aba e começa a preencher os dados dos indicadores ambientais, principalmente os limiares que identificam o estado do indicador. A interface também permite eliminar ou acrescentar indicadores específicos, por parte dos usuários (funcionalidade que é explicada na “História de usuário #4”).
3. O usuário preenche os dados das outras duas dimensões e o sistema salva as mudanças.

História de usuário #4: “Permitir o emprego da metodologia para avaliação caso a caso: possibilitar que o usuário selecione quais indicadores vai utilizar. Dentro dos indicadores, ele pode recomendar limiares mais adequados para a sua realidade, também pode inserir novos indicadores/limiares.”

1. O usuário poderá fazer a “História de usuário #3” ou entrar no sistema e continuar na tela de cadastro de indicadores e, quando acontecer que o usuário precise de um indicador que não seja oferecido pelo sistema, o usuário poderá acrescentá-lo por meio do botão “acrescentar indicador”
2. O usuário dá um *click* no botão “acrescentar indicador” e lhe é apresentada uma interface de entrada, onde ele deverá cadastrar o título, a descrição, os limiares, a medida do manejo e a justificativa desse indicador. Em seguida preencher o estado do indicador. O sistema salva esses dados inseridos.
3. O usuário também poderá eliminar alguns indicadores segundo seu critério.

História de usuário #5: “O usuário poderá obter o resultado dos índices segundo a informação preenchida e a formula de agregação dos indicadores.”

1. Depois de terminada a “História de usuário #4”, o sistema fará a avaliação, que foi definida no sistema pelos especialistas.
2. O resultado da avaliação será cadastrado no sistema com informações sobre a metodologia utilizada.
3. A metodologia de avaliação pode ser atualizada pelos administradores para uso em avaliações futuras.

História de usuário #6: “O usuário poderá armazenar a informação dos indicadores para futuras consultas.”

1. O usuário preenche alguns indicadores nos formulários do SustenAgro.
2. Esses dados serão salvos quando o usuário mudar de formulário ou quando der um *click* no botão “próximo”.

História de usuário #7: “O usuário poderá acrescentar indicadores que considere importantes para sua análise, devem-se estabelecer regras para essa funcionalidade de tal modo que os novos indicadores (criados pelos usuários) sejam recuperáveis de um modo separado dos indicadores cadastrados no sistema.”

1. Quando o usuário estiver preenchendo os indicadores gerados pelo sistema, o sistema fornecerá um conjunto de controles que permitam a inclusão de um novo indicador. Esse novo indicador será definido pelo próprio usuário baseado na sua experiência na área.
2. O sistema armazenará esse novo indicador com uma classificação especial que permita sua identificação e separação dos outros indicadores.
3. O usuário poderá preencher os dados do novo indicador, para que sejam inclusos na avaliação de sustentabilidade.

História de usuário #8: “Cronograma de avaliação, depois de cada safra.”

1. Depois de fazer o cadastro da fazenda e das culturas que são plantadas nela, o sistema poderá identificar quando termina cada safra, gerando um alerta para que o usuário faça o processo de avaliação nessa data.
2. O usuário lerá o alerta e poderá fazer o processo de avaliação de sustentabilidade.

Storyboard

Storyboards são similares aos cenários. Elas ilustram a interação necessária para atingir um objetivo sem utilizar uma lista de passos. A interação é visualizada por meio de uma história em quadrinhos.

Essa representação permite uma visão holística da interação do usuário, com ênfase nos aspectos funcionais da interação e não nos aspectos da interface de usuário. A seguir, são apresentados os textos das *storyboard* dos processos identificados:

A Figura 22 apresenta o processo de cadastro da localização da unidade produtiva, para conseguir vincular dados a partir da localização geográfica.

Figura 22 – *Storyboard* definição da localização.

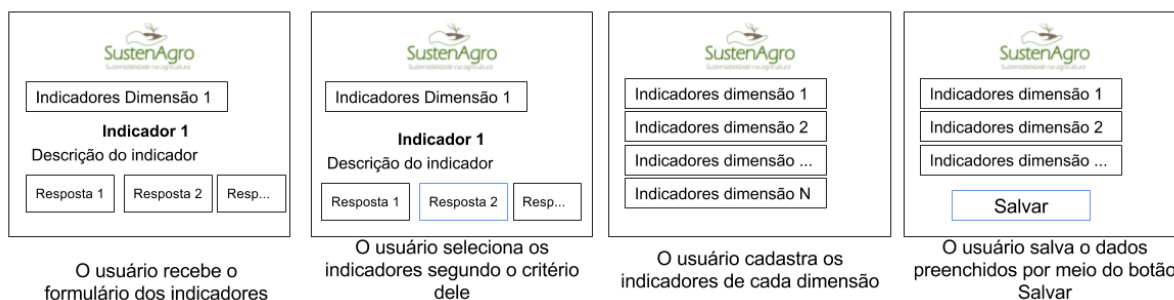
Fonte: Elaborada pelo autor.

A Figura 23 apresenta o formulário de seleção da microrregião que faz parte da localização descrita no *storyboard* anterior. Essa informação é importante para caracterizar a unidade produtiva.

Figura 23 – *Storyboard* seleção da unidade produtiva.

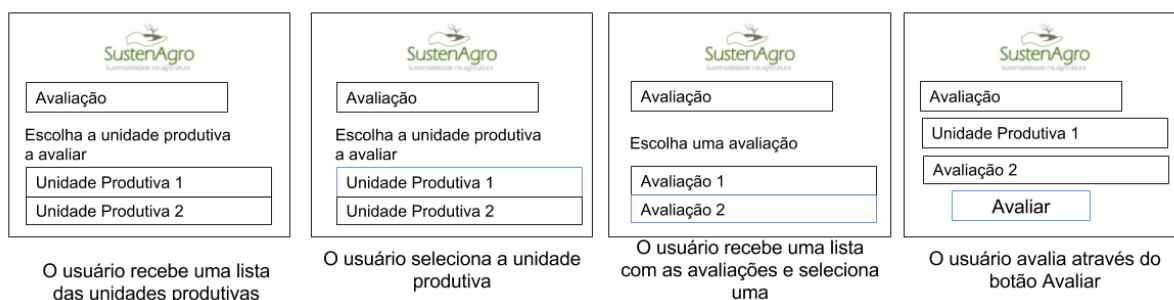
Fonte: Elaborada pelo autor.

A Figura 24 apresenta o esquema do formulário de preenchimento dos indicadores que permite cadastrar uma avaliação, dito formulário é adaptável a vários tipos de dados dos indicadores, permitindo construir interfaces amigáveis para os usuários.

Figura 24 – *Storyboard* mostrando o preenchimento dos indicadores.

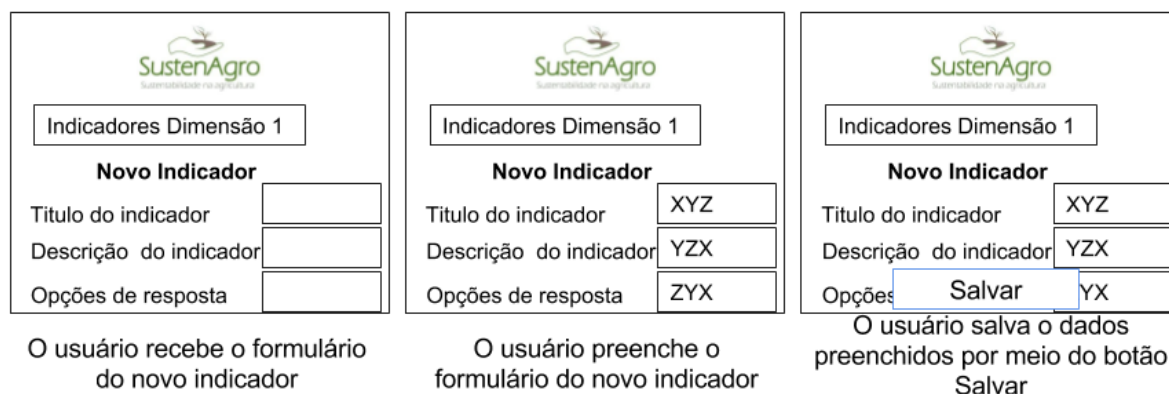
Fonte: Elaborada pelo autor.

A Figura 25 apresenta o processo de avaliação para uma unidade produtiva, segundo o método Sustenagro. Ele vai processar os indicadores preenchidos para gerar uma análise.

Figura 25 – *Storyboard* sobre a avaliação de unidade produtiva

Fonte: Elaborada pelo autor.

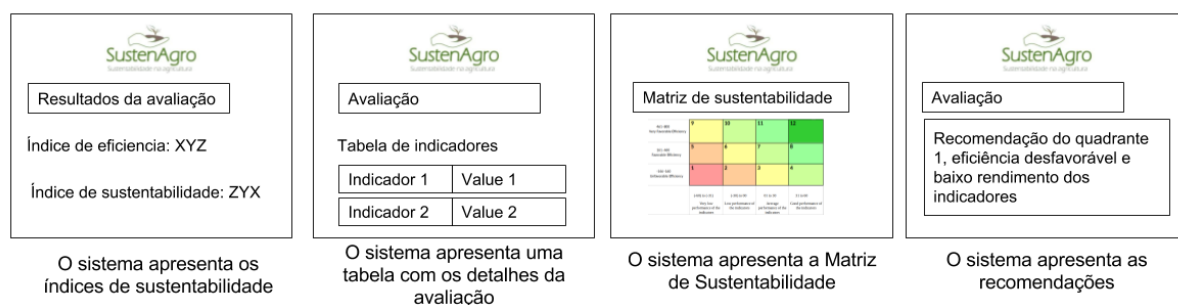
A Figura 26 apresenta o formulário de definição de novos indicadores, por parte dos usuários do sistema. Eles permitem a integração de novos conceitos ao sistema.

Figura 26 – *Storyboard* para cadastro de novo indicador.

Fonte: Elaborada pelo autor.

A Figura 27 apresenta o relatório resultante do processo de avaliação. Ele é composto pelos índices, uma tabela dos dados cadastrados, a matriz de sustentabilidade e as recomendações

Figura 27 – *Storyboard* mostrando a apresentação de resultados



Fonte: Elaborada pelo autor.

Mockups das Interfaces do SustenAgro

A partir das técnicas anteriores, foi possível identificar as tarefas que os usuários do sistema SustenAgro realizarão. O fluxo das tarefas e os dados de cada uma foram mudando ao longo do tempo, como resultado na aplicação de cada técnica, gerando varias definições das *Web UI*. Uma vez identificadas as caracterizaras essenciais da interface gráfica, procedeu-se a definição dos *Mockups*, que permitiram criar uma representação visual das interfaces do sistema.

O desenvolvimento dos *Mockups* foi feito com a ferramenta *Moqups*³.

A Figura 28 mostra uma interface gráfica da tela inicial do SAD SustenAgro. A interface mostra uma descrição do sistema e as principais abas, dentre elas, a aba de Ferramenta que permite iniciar o processo de avaliação de sustentabilidade.

³ Moqups <<https://moqups.com/>>

Figura 28 – Mockup da tela inicial do SustenAgro.



Fonte: Elaborada pelo autor.

Os *Mockups* representam as *widgets*, o que permitem avaliar a interface gráfica de uma maneira mais próxima à interface final e assim suportar a sua avaliação por parte dos especialistas do domínio.

A Figura 29 apresenta os passos do processo de avaliação, na ordem representada pela numeração das abas. A tela apresentada corresponde ao formulário de cadastro dos indicadores, que permite cadastrar o valor correspondente a cada indicador.

Figura 29 – Mockup da tela de indicadores do SustenAgro.

SustenAgro - WebTool

http://sustenagro.embrapa.br

SustenAgro
Sustentabilidade na agricultura

Início Ferramenta Contacto

Sign In Login

1. Localização: Microregião 342 2. Cultura: Cana de açúcar 3. Aderência às tecnologias 4. Caracterização dos sistemas produtivos no Centro-Sul 5. Indicadores 6. Recomendações

Ambientais Econômicos Sociais

Indicadores

▼ Indicador	▼ Descrição	▼ Valor	▼ Unidades
Indicador 1	...	10	...
Indicador 2	...	-10	...
Indicador 3	...	10	...
Indicador 4	...	10	...
Indicador 7	...	10	...
Indicador 8	...	0	...
Indicador 9	...	0	...
Indicador 10	...	-10	...
Novo Indicador	Nova Descrição	-10	...

Embrapa ICMC USP UFST/UFPA

Fonte: Elaborada pelo autor.

Na apresentação dos *Mockups* aos especialistas, foram identificadas várias mudanças para facilitar a navegabilidade da *Web UI*. Essas mudanças corresponderam à integração de tarefas e a melhora na apresentação dos formulários e dos resultados de avaliação. Essas mudanças reestruturaram o *look-and-feel* do SAD.

Protótipo da Interface Gráfica do SustenAgro

A partir das melhoras identificadas nas avaliações dos *Mockups*, foram identificadas as características finais da *Web UI*, com as quais foi desenvolvido um protótipo funcional da *Web UI* do SustenAgro. Inicialmente, esse protótipo só permitia interagir com dados simulados.

A Web UI do SAD SustenAgro foi integrada com cada uma das funcionalidades do SAD em cada ciclo de desenvolvimento, e, atualmente, está disponível nos servidores do laboratório Intermídia do ICMC-USP ⁴.

Na Figura 30 é apresentada a página inicial do protótipo.

Figura 30 – Protótipo do SustenAgro – Tela inicial.



Fonte: Elaborada pelo autor.

Nessa tela pode-se observar o texto explicativo da ferramenta e as abas de “Início”, “Ferramenta” e “Contato”. A opção “Ferramenta” permite iniciar o processo de avaliação de sustentabilidade.

Uma vez cadastrada uma unidade produtiva, disponibiliza-se a opção de criar nova avaliação. Essa, ação vai gerar a tela da Figura 31, que permite visualizar os indicadores para que os usuários preencham cada um, segundo a realidade da unidade produtiva em avaliação. Cada indicador tem várias opções de resposta que estão ligadas a valores que quanti-

⁴ <http://biomac.icmc.usp.br:8080/sustenagro/>

ficam a sustentabilidade. Esses valores estão definidos na ontologia de sustentabilidade (a ontologia SustenAgro) e são usados nas fórmulas para gerar os índices de sustentabilidade.

Na Figura 31, é apresentado o formulário dos indicadores de eficiência. Eles são subdivididos em eficiência de produção e tecnológica. Na Figura 31, é mostrado o indicador Manejo, como exemplo. Um tipo de manejo foi escolhido e o peso desse indicador foi declarado como direto. Esses valores serão usados nas fórmulas da avaliação da sustentabilidade.

Figura 31 – Cadastro de indicadores

Fonte: Elaborada pelo autor.

A partir dos dados cadastrados, são gerados os resultados do sistema. Eles consistem na planilha de eficiência e custo, na planilha da sustentabilidade e o relatório do sistema. As planilhas permitem visualizar os atributos dos indicadores e a tela de relatório apresenta a matriz de sustentabilidade, onde são relacionados os índices de eficiência e de sustentabilidade. O relatório é apresentado na Figura 32.

Figura 32 – Planilhas do resultado da avaliação

Assessment
Results

Efficiency and cost spreadsheets
Sustainability spreadsheets
Assessment report

Sustainability assessment

Environmental Dimension

Indicator	Relevance	Registered value	Value	Total value
Absolute emissions of greenhouse gases (GEE) or associated gases	3.0	Meets determined values	1.0	3.0

Justification

Indicator	Justification
Absolute emissions of greenhouse gases (GEE) or associated gases	

Index	Total value
Environmental index	3.0

Economic Dimension

Indicator	Relevance	Registered value	Value	Total value
Have fuel stocks in order to regulate supply / demand and prices of ethanol (buffer stocks)	2.0	Yes	1.0	2.0

Justification

Indicator	Justification
Have fuel stocks in order to regulate supply / demand and prices of ethanol (buffer stocks)	

Fonte: Elaborada pelo autor.

5.5 Web Components

Foram desenvolvidos dois *Web Components* específicos para o SustenAgro. Eles geram gráficos específicos do relatório solicitado pelos especialistas da Embrapa Meio Ambiente

Matriz de Sustentabilidade

Com a finalidade de suportar a geração de relatórios, no formato definido pelos especialistas do domínio, foi necessário implementar dois *Web Components* específicos. Um deles foi a *widget* intitulada Matriz de Sustentabilidade. Ela é composta por dois eixos que

correspondem ao índice de eficiência, eixo Y, e ao índice de sustentabilidade, eixo X. Os índices tem magnitudes que são divididas em segmentos que permitem dividir a área em doze quadrantes da sustentabilidade. Cada avaliação realizada com o método SustenAgro gerará dois índices que são localizados em um quadrante da matriz de sustentabilidade. Cada quadrante está relacionado com uma recomendação específica.

A Figura 33 mostra a implementação desse *Web Component*, mostrando resultados reais de uma avaliação.

Figura 33 – Matriz de sustentabilidade mostrando um índice de sustentabilidade de 2,3 e um índice de eficiência de 2,4.



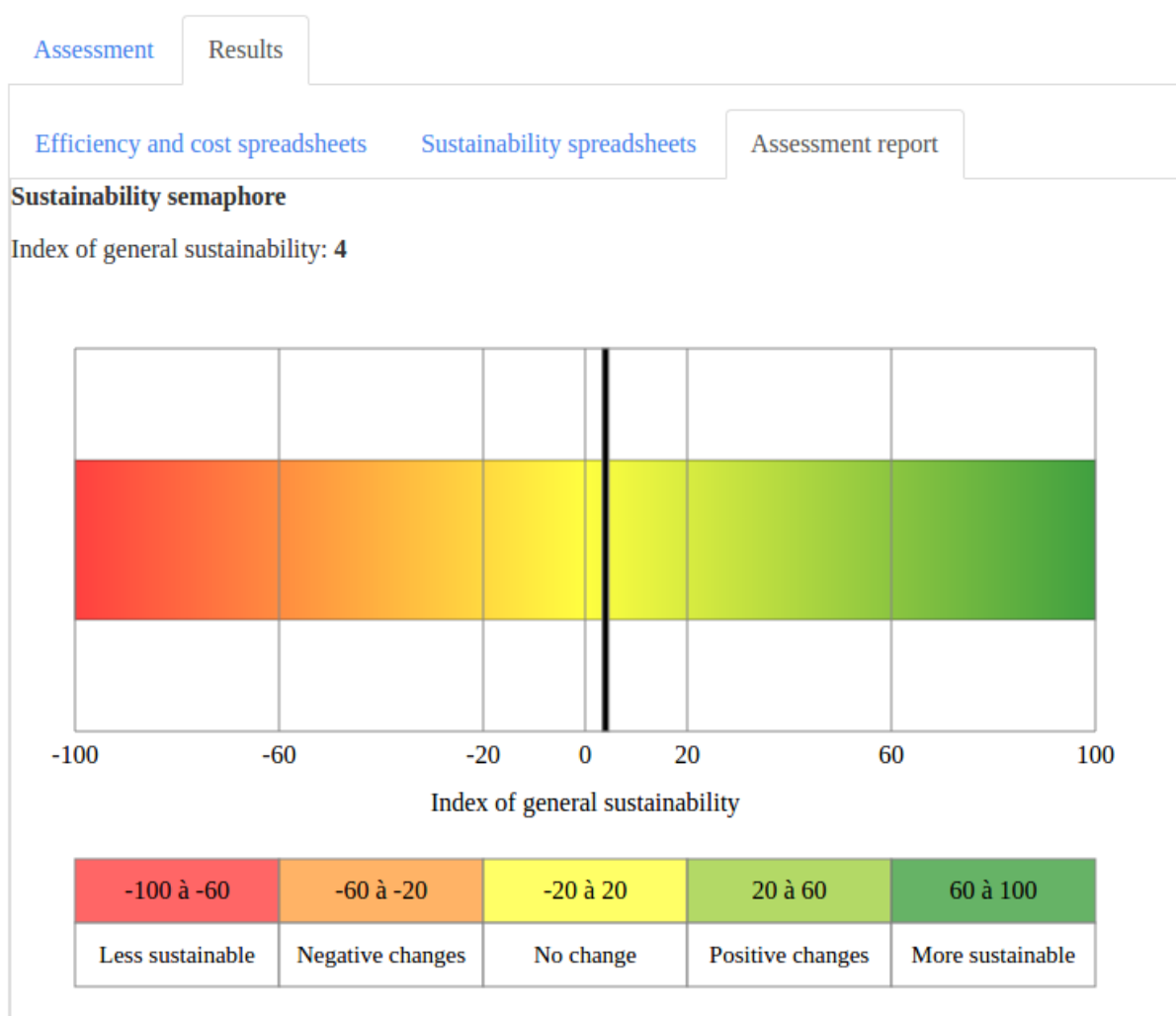
Fonte: Elaborada pelo autor.

Semáforo da Sustentabilidade

O *Web Component* do Semáforo da Sustentabilidade foi o segundo componente para geração de relatórios, no formato definido pelos especialistas do domínio, desenvolvido para o método SustenAgro. Ele tem um eixo que quantifica o valor da sustentabilidade normalizado entre -100 até +100, dividindo o intervalo em 5 segmentos que correspondem às categorias de sustentabilidade.

A Figura 34 mostra esse componente no sistema SustenAgro com os valores de uma avaliação de sustentabilidade.

Figura 34 – Semáforo de Sustentabilidade mostrando um índice de sustentabilidade geral de 4 numa escala de -100 até +100.



Fonte: Elaborada pelo autor.

5.6 DSL: code

A implementação de uma DSL permite aos especialistas do domínio definir como são usados e apresentados os conceitos da ontologia, por meio de elementos da interface gráfica, como os índices de sustentabilidade serão calculados e quais os elementos presentes no relatório final. A DSL permite a criação de SADs facilmente adaptáveis às mudanças do domínio. Os próprios especialistas podem modificá-la sem o auxílio de programadores.

Para definir o comportamento do SustenAgro, especialistas tiveram que:

Definir o Objeto da Avaliação

No comando, a seguir 5.1, o Objeto de Avaliação é definido como uma instância da classe *ProductionUnit*. Essa classe foi definida pelos próprios especialistas na ontologia e tem como filhos as classes *Farm* e *Power Plant*. Também são declaradas todas as propriedades que os usuários terão que preencher quando criarem uma avaliação. Por exemplo, na propriedade *hasName* os usuários devem preencher o nome da unidade de produção.

Código-fonte 5.1 – DSL que define o Evaluation Object

```
evaluationObject ':ProductionUnit', {
  instance 'ui:hasName', label: ['en': 'Production unit or farm name',
    'pt': 'Nome da unidade produtiva ou fazenda']
  instance ':hasAgriculturalProductionSystem', label: ['en': '
    Agricultural production system', 'pt': 'Sistema de produção
    agrícola']
  type label: ['en': 'Production unit type', 'pt': 'Tipo da unidade
    produtiva']
  instance ':hasSugarcaneSource', label: ['en': 'Sugarcane source', '
    pt': 'Origem da cana'], multipleSelection: true, required: true
  instance 'dbp:state', label: ['en': 'State', 'pt': 'Estado']
  instance 'ui:hasMicroregion', label: ['en': 'Production unit
    microregion', 'pt': 'Microrregião da unidade produtiva']
  instance ':hasAvailabilityOfEvaluationResults', label: ['en': '
    Availability of evaluation results', 'pt': 'Disponibilização dos
    resultados da avaliação']
}
```

Definir as características a serem avaliadas

Agora os especialistas têm que escolher quais características (*Features*) dos Objetos de Avaliação serão usadas. No comando *feature*, são indicadas as classes das *Features* a serem usadas. Serão mostradas todas as *features* das classes indicadas e de suas descendentes. Por exemplo, o comando *feature ':ProductionEfficiencyFeature'* vai mostrar todas as

Código-fonte 5.2 – Definição de Features

```

feature ':EnvironmentalIndicator', 'extraFeatures': true
feature ':EconomicIndicator', 'extraFeatures': true
feature ':SocialIndicator', 'extraFeatures': true
feature ':ProductionEfficiencyFeature'
feature ':TechnologicalEfficiencyFeature', {
  conditional ":ProductionUnit", 'http://dbpedia.org/ontology/Provider', {
    include ':TechnologicalEfficiencyInTheField'
  }
  conditional ":ProductionUnit", 'http://dbpedia.org/resource/PhysicalPlant', {
    include ':TechnologicalEfficiencyInTheField', ':TechnologicalEfficiencyInTheIndustrial'
  }
}

```

Features relacionadas com eficiência da produção. Na ontologia SustenAgro foram estabelecidas as famílias de *Features*: *EnvironmentalIndicator*, *EconomicIndicator*, *SocialIndicator*, *ProductionEfficiencyFeature* e *TechnologicalEfficiencyFeature*.

O comando *feature* também permite dizer se existirão novas *Features* criadas pelos usuários ('*extraFeatures*': *true*) e a apresentação condicional de grupos de *Features*. No exemplo do algoritmo 5.2, são mostradas *Features* industriais apenas à usuários de usinas.

Definir a modelagem e a forma de apresentação dos resultados

Finalmente, no comando *data*, os especialistas definem o nome da variável que contém as respostas dos usuários e, no comando *report*, fazem o cálculo da modelagem e definem o que vai ser apresentado. No algoritmo 5.6, é possível ver o cálculo dos índices de sustentabilidade e eficiência por meio das formulas do modelo usado pelo SustenAgro. É possível usar qualquer comando da linguagem *Groovy* ou biblioteca externa. As variáveis guardam os valores de interesse que serão mostrados no relatório de avaliação. No caso do SustenAgro, a variável *sustainability* guarda o índice de sustentabilidade e a variável *efficiency* guarda o índice de eficiência do sistema.

No relatório aparecerão a Matriz de Sustentabilidade (*SustainabilityMatrix*), o Semáforo de Sustentabilidade (*SustainabilitySemaphore*), o texto "Mapa da Microrregião" e o mapa da microrregião (onde a unidade produtiva se encontra). Cada uma dessas *widget* é instanciada na DSL e os valores de interesse são passados para os *Web Components* encarregados da apresentação. O *Framework Decisioner* delega aos *Web Components* a tarefa de apresentação.

Usuários podem pedir a geração de relatórios em *pdf*, uma ferramenta de conversão de HTML para *pdf* é usada.

```

data 'data'
report {
  environment = weightedSum(data.':EnvironmentalIndicator')
  economic    = weightedSum(data.':EconomicIndicator')
  social      = weightedSum(data.':SocialIndicator')

  sustainability = (environment + social + economic)/3

  cost_production_efficiency = sum(data.':ProductionEfficiencyFeature')
  technologicalEfficiencyInTheField = 0.8*weightedSum(data.':
    TechnologicalEfficiencyInTheField')
  technologicalEfficiencyInTheIndustrial = 0.2*weightedSum(data.':
    TechnologicalEfficiencyInTheIndustrial')

  efficiency = Math.abs(cost_production_efficiency) * (
    technologicalEfficiencyInTheField+
    technologicalEfficiencyInTheIndustrial)

  sustainabilityMatrix x: sustainability, y: efficiency,
    label_x: ['en': 'Sustainability Index', 'pt': '
      Índice de Sustentabilidade'],
    label_y: ['en': 'Efficiency index', 'pt': '
      Índice de Eficiência'],
    range_x: [-43,43],
    range_y: [-160,800],
    quadrants: [4,3]

  sustainabilitySemaphore value: sustainability,
    label: ['en': 'Sustainability Level', 'pt': '
      Índice da sustentabilidade geral'],
    range: [-60,60]

  text 'en': 'Microregion map', 'pt': '**Mapa da microregião**'

  map data.'Microregion'
}

```

5.7 Considerações Finais

O desenvolvimento do sistema Sustenagro satisfaz uma necessidade presente na unidade da Embrapa Meio Ambiente: um sistema de avaliação de sustentabilidade em cana-de-açúcar no centro-sul do Brasil. O SAD SustenAgro representa e permite complementar informações e dados do estado atual de sustentabilidade nas fazendas e usinas. Ele produz relatórios com a finalidade de embasar e formalizar políticas para promover práticas produtivas mais sustentáveis, de acordo com critérios ambientais, sociais e econômicos.

Além de satisfazer uma necessidade institucional, o SustenAgro é uma proposta de SAD baseado em conhecimento e vinculado às tecnologias da web semântica. O SustenAgro não foi apenas uma instanciação do *Framework Decisioner*, ele definiu o próprio *framework*. Por meio do desenvolvimento do SustenAgro, foi possível determinar as características gerais desse tipo de SAD, e implementá-las no *framework*, e as características específicas do Sustenagro, que foram implementadas usando a ontologia SustenAgro, os dois *Web Components* e a DSL.

Tendo o sistema SustenAgro instanciado e rodando, foi possível validar as funcionalidades por meio de experimentos com integrantes do projeto SustenAgro da Embrapa. Os resultados obtidos nas avaliações dos experimentos serão detalhados no próximo capítulo.

Avaliação

O SAD SustenAgro (Capítulo 5) foi instanciado usando o *Framework Decisioner* (Capítulo 4) e avaliado, durante diferentes estágios de desenvolvimento, por meio de experimentos realizados com especialistas de domínio e usuários da Embrapa. Esses dois sistemas foram desenvolvidos com metodologias iterativas, onde foram realizadas várias inspeções e testes durante a implementação. Uma avaliação final também foi realizada para analisar se as funcionalidades foram implementadas corretamente.

A seguir, são apresentadas as avaliações realizadas no SAD SustenAgro e ao *Framework Decisioner*. Elas foram independentes e geraram resultados que levaram ao redesenho das arquiteturas de ambos sistemas em diferentes etapas do seu desenvolvimento.

6.1 Avaliação das Web UI

Uma vez finalizado o processo de *design* das Web UI, explicado na Seção 5.4, foi realizada uma avaliação da usabilidade das Web UI do SAD SustenAgro. Os detalhes dessa avaliação foram:

Data Junho de 2015

Participantes Usuários da ferramenta: especialista em sustentabilidade e especialista em economia agrícola

Local Instituto de Ciências Matemáticas e de Computação (ICMC-USP)

Técnica Avaliação de usabilidade

Nesta avaliação foi apresentado aos usuários especialistas, o processo de design da UI e o protótipo da interface gráfica de usuário do SAD SustenAgro sem dados reais. Nessas interfaces, eles interagiram com as telas, por meio de um navegador *web*, fazendo uso das funcionalidades do SAD que simulava os dados durante o processo de avaliação.

Durante esta avaliação foram verificados, os três aspectos da usabilidade:

- **Eficácia:** as interfaces permitiram realizar as tarefas segundo as funcionalidades definidas e permitiam interagir de maneira intuitiva para realizar as tarefas.
- **Eficiência:** o acesso à ferramenta foi realizado em tempos esperados e foi possível realizar as tarefas com recursos típicos de um laptop e um navegador *web*.
- **Satisfação:** os usuários conseguiram realizar as tarefas sem problemas e com uma experiência de fácil uso, onde a ferramenta fornecia informação de ajuda para realizar as interações.

A avaliação demonstrou que as interfaces gráficas satisfizeram os requisitos, cumprindo os três requisitos de usabilidade. Esta avaliação gerou várias recomendações para melhorar a interface gráfica de usuário, entre elas as mais relevantes estão:

- Integrar as tarefas de definir a localização, as principais características e a disponibilização da informação das unidades produtivas em um processo unificado inicial dentro do processo de avaliação, agrupando essas tarefas no processo denominado “caracterização da unidade produtiva”.
- Agrupar os resultados das avaliações em uma seção “Resultados”, integrando os *Web Components* específicos do SustenAgro com os resultados da avaliação.
- Organizar os indicadores em uma hierarquia simplificada que facilita o preenchimento dos mesmos.

Depois de implementar as recomendações anteriores, que foram as principais em relação a UI, foi continuado o desenvolvimento com a integração da ontologia SustenAgro e da DSL. Nessa fase, foi necessário realizar ajustes da interface ao integrar cada funcionalidade.

6.2 Avaliação da ontologia de domínio de avaliação da sustentabilidade

A ontologia SustenAgro foi o resultado da modelagem do conhecimento dos especialistas, que passou por várias etapas, desde ser definida em texto, modelada em mapas conceituais e especificada no padrão OWL (esse processo foi detalhado na Seção 5.2). Cada uma dessas etapas requereram inspeções por parte dos especialistas para verificar a correta modelagem, por isso uma avaliação formal foi realizada e os detalhes dela estão descritos a seguir.

Participantes Especialista em sustentabilidade e especialista em modelagem de conhecimento

Local Embrapa Informática Agropecuária - Campinas

Técnica Visualização da ontologia e recuperação de conhecimento

O especialista em modelagem de conhecimento da Embrapa Informática Agropecuária e o especialista em sustentabilidade reuniram-se para realizar a revisão da ontologia SustenAgro por meio de ferramentas de engenharia de conhecimento que permitiram visualizar a estrutura das ontologias.

As ferramentas usadas na avaliação foram yWorks ¹ e Gephi ². Elas permitiram representar aspectos da ontologia usando visualizações de grafos que ajudam a avaliar a coerência das conexões entre os nós daqueles grafos.

Também foi analisada a ontologia SustenAgro integrada no SAD SustenAgro, por meio do uso das funcionalidades do SAD que requerem dados da ontologia, verificando que a informação recuperada seja coerente em relação à existente na ontologia.

A reunião de avaliação chegou no consenso: a ontologia OWL está representando a avaliação da sustentabilidade do sistema de produção de cana no centro-sul do Brasil, destacando a especificidade dela para suportar o SAD Sustenagro e que seu foco em conceitos concretos.

Foi recomendado integrar outros tipos de visualizações e editores de ontologia no *Framework Decisioner*. Entre eles foi recomendado integrar um editor visual de ontologias, baseado na ferramenta WebVOWL, definida e desenvolvida por Lohmann et al. (2014). Devido a limitações de tempo, não foi possível a implementação dessa funcionalidade.

6.3 Avaliação do protótipo funcional com dados

Com a finalidade de avaliar a implementação do método SustenAgro, foram realizados testes de vários tipos de avaliações, para comparar com resultados de outras fontes, e assim validar a correta implementação. Os detalhes da avaliação foram:

Data 6 de junho do 2016

Participantes Especialista em sustentabilidade e especialista em economia agrícola

Local Agência Paulista de Tecnologia dos Agronegócios (APTA)

Técnica Testes numéricos dos resultados do software

¹ <<http://www.yworks.com/>>

² <<https://gephi.org/>>

A revisão dos resultados numéricos foi realizada a partir de vários cenários de indicadores, sendo realizada uma comparação dos resultados processados manualmente com os resultados de avaliação do SAD SustenAgro. Os resultados manuais foram iguais aos calculados pelo sistema, o que permitiu validar a correta definição do método SustenAgro e sua implementação no SAD SustenAgro.

Os cenários avaliados foram casos extremos dos indicadores, processo denominado teste de mínimos e máximos, gerando resultados em forma de relatório para cada um dos cenários.

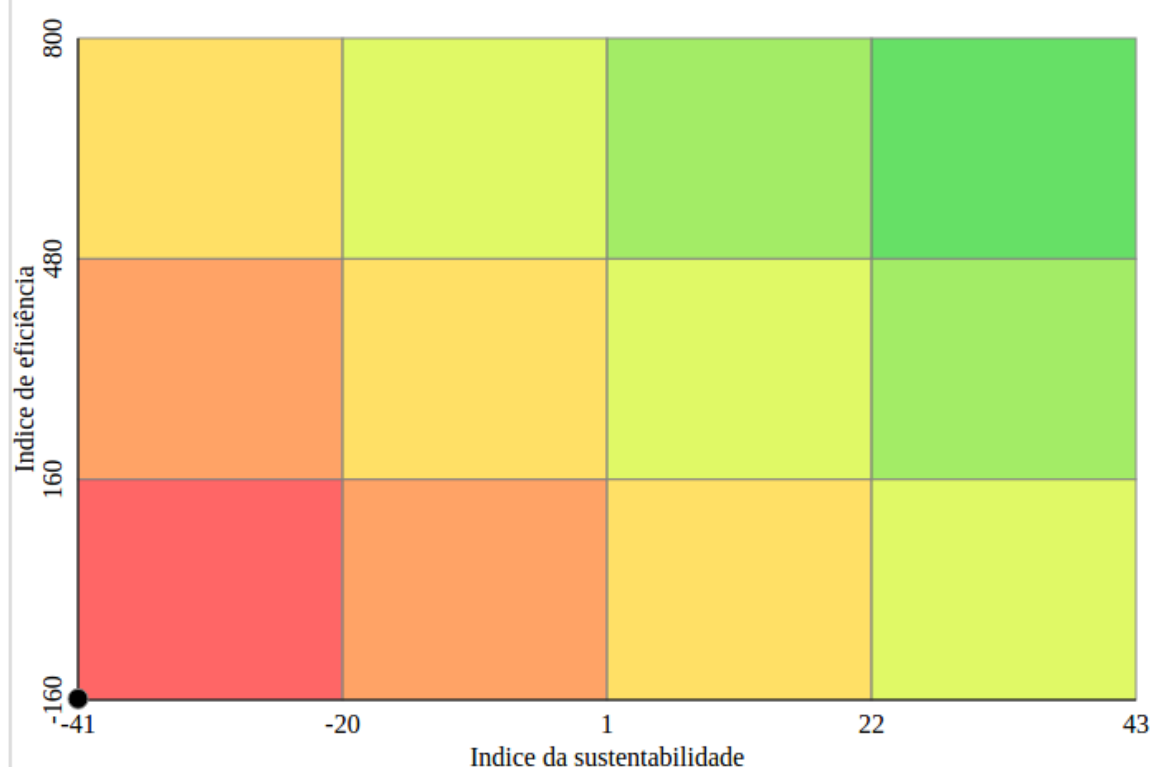
A Figura 35 apresenta a Matriz de Sustentabilidade com o cenário de mínimos. Na matriz, o ponto preto identifica o resultado da avaliação. Ele ficou no primeiro quadrante tal como se esperava. Os resultados para testes de máximos também se comportaram como o esperado.

Um dos aspectos mais importantes da ferramenta, é que ela permite que os próprios especialistas mudem, em tempo real, as fórmulas do método de avaliação, permitindo fazer ajustes finos no método. Por exemplo, durante a realização dos testes, foi descoberto um problema na fórmula do método SustenAgro original. Usando a DSL, algumas opções de correção foram tentadas e, rapidamente, foi possível redefinir a fórmula de avaliação. A possibilidade da interação direta e fácil dos especialistas, em tempo real, com o sistema permitiu demonstrar agilmente possíveis cenários de resolução desse problema e aceitar, rapidamente, uma solução (proposta pelo autor).

Figura 35 – Matriz de Sustentabilidade com valores mínimos.

Matriz de AvaliaçãoÍndice da sustentabilidade: **-41**Índice de eficiência: **-158.4**Quadrante: **1**

Recomendação: **Avaliação da eficiência: balanço da eficiência ‘tecnológica – produção – custo’ desfavorável ao sistema de produção de cana / sistema de produção de cana na fase inicial de implementação (avaliação de sustentabilidade comprometida) ou com muito baixa sustentabilidade – sistema de produção de cana não recomendado.**



Fonte: Elaborada pelo autor.

6.4 Avaliação do SAD SustenAgro e do *Framework Decisioner*

A partir da avaliação da ontologia, da implementação do método SustenAgro e do desenvolvimento das *Web UI* realizou-se uma avaliação do SAD com a finalidade de avaliar a integração desses componentes.

Data 18 de maio do 2016 até o dia 22 de junho do 2016

Participantes Usuários especialistas em sustentabilidade e usuários finais.

Local Instituto de Ciências Matemáticas e de Computação (ICMC-USP)

Técnica Teste de usabilidade

Para realizar uma avaliação integral do SAD SustenAgro e do Framework Decisioner, foi necessário fazer testes de usabilidade com usuários de ambos os perfis do sistema (especialistas de domínio e usuários finais). Essa avaliação foi realizada com a maioria dos membros da equipe SustenAgro e com usuários finais, de maneira remota e independente, totalizando 8 avaliações.

A avaliação consistiu em realizar um conjunto de tarefas com o SAD SustenAgro v1.0 e responder se foi possível terminar a tarefa e as sugestões. As tarefas e perguntas solicitadas aos usuários estão listadas no Apêndice D.1 e permitiram gerar os seguintes resultados das avaliações:

Tabela 1 – Avaliação dos usuários finais do SAD SustenAgro

Perfil de Usuário	Avaliação	Sugestões
Usuário final 1	Satisfatória, realizou as 5 tarefas de usuário final com sucesso	<ul style="list-style-type: none"> • Aumentar a ajuda para cada funcionalidade • Barra de progresso durante a avaliação • Resultados de avaliação mais detalhados
Usuário final 2	Satisfatória, realizou as 5 tarefas de usuário final com sucesso	<ul style="list-style-type: none"> • Melhorar a explicação do processo de avaliação • Resultados numéricos com formatação
Usuário final 3	Satisfatória, realizou as 5 tarefas de usuário final com sucesso	<ul style="list-style-type: none"> • Segurança no cadastro da senha • Melhorar a localização do botão avaliar • Remover scroll externo • Salvar automaticamente os dados

Tabela 2 – Avaliação dos especialistas do SAD SustenAgro

Perfil de Usuário	Avaliação	Sugestões
Especialista em sustentabilidade	Satisfatória, realizou as 5 tarefas de usuário e as 5 tarefas de especialista de domínio com sucesso	<ul style="list-style-type: none"> • Definir e acrescentar os termos de uso • Melhorar a sequencia de telas durante o cadastro de novo usuário • Balão explicativo dos campos do formulário de nova unidade produtiva, especificamente a propriedade de publicação dos dados • Definir o limite de caracteres para o campo de justificativa • Salvar avaliação ao mudar de aba • Justificativas sempre visíveis na tela de resultados. • Acrescentar bandeira em inglês e termos de uso
Especialista economia	Satisfatória, realizou as 5 tarefas de usuário e as 5 tarefas de especialista de domínio com sucesso	<ul style="list-style-type: none"> • Salvar dados da seção automaticamente • Mudanças em alguns labels • Mudanças nos indicadores • Componentes gráficos para representar os dados • Editor visual de ontologia e internacionalização
Especialista em ontologias	Satisfatória, realizou as 5 tarefas de usuário final com sucesso	<ul style="list-style-type: none"> • Melhorar a apresentação do botão salvar • Indicadores em forma de pergunta com verbo e simbolo de pergunta • Informar a possibilidade de edição de indicadores
Especialista agricultura	Satisfatória, realizou as 5 tarefas de usuário final com sucesso	<ul style="list-style-type: none"> • Cadastrar mais de uma microrregião • Remover scroll externo, mudanças em vários indicadores • Melhorar a localização do botão salvar e avaliar • Salvar a seção para não perder os dados • Melhorar a apresentação do pdf
Especialista em computação	Satisfatória, realizou as 5 tarefas de usuário final com sucesso	<ul style="list-style-type: none"> • Salvar a seção e os dados dela em tempo real • Integrar sistemas externos para poupar informação (caracterização da unidade produtiva)

A partir dessa avaliação integral, feita por usuários finais e administradores, foram definidas várias melhorias a serem realizadas no *Framework Decisioner* e o SAD SustenAgro. Devido às limitações de tempo e de desenvolvedores, foram implementados apenas os ajustes visuais nas web UI, melhoras na apresentação dos resultados, na geração do PDF e, principalmente, a inclusão da linguagem inglês. Essa última mudança foi selecionada como de especial importância, por parte dos especialistas. Neste documento, são apresentadas várias telas, tanto em português como em inglês, que foram geradas a partir da implementação dessa funcionalidade.

6.5 *Workshop*: validação do software SustenAgro v1.0 com equipe de especialistas

A partir das melhoras realizadas na avaliação interna pelos dois tipos de usuários do SAD SustenAgro, o sistema foi disponibilizado em servidores *web* do ICMC. Esta publicação permitiu dar suporte a uma avaliação no formato de *workshop* com especialistas de diversos perfis que tinham interesse no SAD SustenAgro. O *workshop* foi intitulado “Validação do software SustenAgro”, os detalhes do *workshop* são apresentados a seguir:

Data 14 de julho do 2016

Participantes Equipe do projeto SustenAgro de várias unidades da Embrapa

Local Embrapa Informática Agropecuária - Campinas

Técnica Delphi

O *workshop* teve o objetivo de avaliar a qualidade e acuidade do SAD SustenAgro em termos da clareza da informação técnica apresentada nas interfaces, com vistas a garantir o entendimento do usuário e possibilitar que a avaliação da sustentabilidade do sistema de produção de cana-de-açúcar seja realizada da melhor maneira possível.

No *workshop* foi apresentado o formulário de avaliação (Seção D.3) usando a técnica *Delphi* (WRIGHT, 1985) (recomendada pelos especialistas em sustentabilidade), a um grupo de especialistas com perfis de várias áreas do conhecimento, entre eles destacam-se:

- Especialista em sustentabilidade
- Especialista em ciências agrícolas
- Especialista em modelagem de conhecimento
- Especialista em ciência e tecnologia do bioetanol
- Especialista em ciências da computação
- Especialista em economia agrícola

- Especialista em biotecnologia
- Mestrando em ciências da computação (sistemas web e multimídia)
- Mestrando em Planejamento de Sistemas Energéticos

As interações com o sistema foram filmadas enquanto os usuários executavam uma lista de tarefas. Monitores do ICMC ficavam estimulando os usuários a falar o que estavam pensando (técnica Think Aloud (DAVEY, 1983)) e os questionavam, quando tinham alguma dificuldade de interação. Ao final, houve um *debriefing* e foram também colhidas mais sugestões de mudança.

Os especialistas validaram que ferramenta satisfizes os requisitos, tanto na interação como no conteúdo dela. Finalmente foram geradas as seguintes sugestões:

- As perguntas dos indicadores não são de fácil interpretação.
- Colocar mais informações na interface para facilitar o uso dela, por exemplo o significado de alinhamento dos indicadores.
- Alguns indicadores estão repetidos.
- Os resultados da avaliação deveriam estar por dimensão.
- Demora para salvar os dados inseridos.
- A informação dos site tem inconsistências em relação à informação do *pdf* gerado.
- A recomendações do relatório precisam mais detalhamento.

É interessante que muitas das sugestões não tem haver com aspectos computacionais ou de interface, mas sim com o processo de avaliação de sustentabilidade. Esse processo é de inteira responsabilidade dos especialistas de domínio (e foge do escopo deste trabalho). Isso foi um ponto que não afeta as contribuições realizadas. É natural que especialistas em sustentabilidade estejam muito mais interessados na sua área do que nos aspectos computacionais do SAD SustenAgro. Uma boa interface é aquela que desaparece da mente do usuário e permite que este se foque na sua tarefa. Neste caso, a avaliação de sustentabilidade. Acreditamos que o SAD SustenAgro atendeu bem a esse requisito.

6.6 Avaliação de SAD SustenAgro nos servidores da Embrapa

A partir da aprovação do SAD SustenAgro, por parte dos especialistas no *workshop*, foi autorizada a instalação da ferramenta nos servidores da Embrapa Meio Ambiente. Essa instalação foi um esforço coordenado entre o desenvolvedor do framework Decisioner (o autor deste trabalho) e técnicos de informática da Embrapa.

Data 18 de agosto do 2016

Participantes Especialista em Sustentabilidade e especialista em TI

Local Embrapa Meio Ambiente - Campinas

Técnica Teste de integração

A instalação, coordenada pelo desenvolvedor do *Framework Decisioner* e técnicos da Embrapa, foi problemática. A Embrapa não autoriza o acesso físico ou via *ssh* ao servidor por parte de profissionais externos. Por esta razão, foi criado um documento de instalação, descrito no Apêndice C. Nele estão as instruções para instalar o *Framework Decisioner* e instanciar o SAD SustenAgro.

A instalação foi exitosa e o sistema está funcionando no endereço <<https://sustenagro.embrapa.br/>>. Atualmente o endereço não está no ar, pois o SAD SustenAgro está em processo de registro no Instituto Nacional de Propriedade Intelectual (INPI), em nome da Embrapa e USP, e a Embrapa ter uma política de exigir esse registro para liberação para uso externo.

6.7 Conclusões

Essas avaliações permitiram verificar que os requisitos do *Framework Decisioner* e do SAD SustenAgro foram implementados corretamente e que atenderam as necessidades dos especialistas de domínio e usuários finais, identificadas nos levantamentos de requisitos.

As avaliações ocorreram em diferentes fases do projeto e cada uma trouxe correções e melhoras que foram implementadas, na medida do possível, segundo a relevância de cada correção e do tempo disponível de desenvolvimento.

Por ter sido desenvolvido em um processo cíclico, cada iteração acrescentou novas funcionalidades que fizeram mudar a arquitetura dos sistemas, gerando bugs e inconsistências. Na versão 1.0, os sistemas contam com funcionalidades estáveis que permitem fornecer os serviços implementados. Correções e sugestões recebidas durante a avaliação e não implementadas por problema de tempo, foram deixadas como trabalhos futuros, que serão apresentados no próximo capítulo juntamente com as conclusões.

Conclusões

As principais contribuições desta pesquisa foram o método e ferramenta de geração de SADs baseados em conhecimento usando ontologias e *DSLs*. Esse método permite complementar o conhecimento descrito em uma ontologia com uma *DSL*, que define o comportamento e formato de um SAD.

Ontologia e *DSL* estão abertas a edição pelos especialistas de domínio. Eles podem editá-las por meio de editores online (na própria ferramenta), e modificar aspectos fundamentais do SAD em tempo real. Essa característica permite um ciclo de desenvolvimento mais curto e uma participação ativa dos especialistas de domínio. Como o *Framework Decisioner* fornece uma aplicação semiacabada (como todo o *framework*) que os próprios especialistas de domínio podem modificar, isso tende a diminuir os custos associados a criação de SADs.

O *Framework Decisioner*, o SAD SustenAgro e o método de definição de SADs proposto tiveram uma avaliação positiva por parte dos especialistas do domínio sustentabilidade e usuários finais da Embrapa e do projeto SustenAgro. Essas avaliações demonstraram que as ferramentas e o método proposto solucionaram, para o caso do SustenAgro, os problemas identificados na pesquisa e que os objetivos específicos desta pesquisa foram satisfeitos.

Deve-se ressaltar que, apesar dos bons resultados das avaliações e do entusiasmo dos especialistas da Embrapa (que nunca tiveram tantas facilidades para modificar o código de um projeto de SAD), ainda não é possível, para especialistas de domínio, desenvolver um SAD, usando o *Decisioner*, a partir do zero. Porém, os resultados mostraram que eles podem modificar muitos aspectos do sistema sem a ajuda de programadores. O que é um avanço muito grande, se comparado com o que é possível fazer em métodos de desenvolvimento tradicionais. Outra importante contribuição foi a geração de uma versão funcional do SAD SustenAgro. Ela permitiu cumprir os objetivos do projeto SustenAgro da Embrapa, fornecendo uma ferramenta de avaliação da sustentabilidade em cana-de-açúcar que pode

efetivamente ser usada por empregados de fazendas e usinas.

Para continuar a pesquisa, foram sugeridos trabalhos futuros. Dentre os mais relevantes estão a metodologia de criação de ontologias e as ferramentas web para editá-las.

7.1 Resultados

Os resultados obtidos foram:

1. Ontologia SustenAgro sobre avaliação de sustentabilidade em sistemas produtivos de cana-de-açúcar na região centro-sul do Brasil. Ela representa os principais conceitos desse domínio, necessários para o suporte da avaliação de sustentabilidade. Dentre eles, estão os indicadores, unidades produtivas, microrregiões, índices e métodos de avaliação. A ontologia padroniza o conhecimento dos especialistas em um formato computável. Ela foi desenvolvida em parceria com os especialistas de domínio de sustentabilidade da Embrapa.
2. Ontologia sobre tipos de dados e controles visuais para suportar a geração automática das interfaces gráficas dos SADs. Ela permite associar tipos de dados às *widgets* (*Web Components*) necessárias para a geração automática das *UI* para os SAD.
3. DSL que permite definir o comportamento de um SAD, usando os conceitos das ontologias, e integrar *Web Components* para gerar as *Web UIs*. Dentro dos elementos suportados, estão a definição dos objetos de avaliação, de indicadores, das fórmulas do método de avaliação e da definição do formato do relatório de cada análise.
4. Método e *Framework Decisioner* para definir SADs, baseados em ontologias e DSLs. O *Framework Decisioner* possui uma arquitetura escalável para permitir sua aplicabilidade a outros tipos de SADs (além do SustenAgro). O código fonte deste *framework* está publicado em: <<https://github.com/DilvanLab/Decisioner>>
5. O SAD SustenAgro: sistema para avaliação da sustentabilidade em cana-de-açúcar, que foi implementado usando o *Framework Decisioner*. O SAD SustenAgro permite a reconfiguração, em tempo de execução, dos conceitos do domínio, métodos de avaliação e componentes das interfaces gráficas, por meio do *Framework Decisioner*.
6. Resultados de avaliação do processo de *design* do SAD SustenAgro e do seu protótipo final. Validando que é possível gerar um SAD, baseado no *Framework Decisioner*, a partir de uma ontologia e DSL, e ter sua *Web UI* gerada automaticamente. Eles também demonstram que especialistas de domínio são capazes de alterar a ontologia e DSL, sem a ajuda de programadores, e ver os resultados dessas modificações em tempo real. Essa flexibilidade torna o desenvolvimento de SADs mais ágil com potencial de redução de seus custos finais.

7. Artigo (co-autoria) “Sustainability assessment of sugarcane production systems: SustenAgro Method” submetido ao periódico da Elsevier “Energy for sustainable Development”. ISSN: 0973-0826. Submissão realizada no dia 23 de dezembro de 2016 (Artigo não foi aceito).
8. Artigo (primeiro autor) “SustenAgro Sistema de Apoio à Decisão baseado em Ontologias e definido por uma Linguagem de Domínio Específico” submetido ao periódico “Revista Brasileira de Sistemas de Informação”. ISSN Eletrônico: 1984-2902. Submissão realizada no dia 24 de março de 2017.

7.2 Dificuldades

Durante a realização desta pesquisa foram identificadas as seguintes dificuldades:

A primeira delas refere-se à falta de dados e informações para modelar o domínio do conhecimento de avaliação em sustentabilidade. Depois de estabelecer meios de representação para modelar o conhecimento dos especialistas, descobriu-se que não existiam dados na Embrapa Meio Ambiente que representassem instâncias dos conceitos definidos. Portanto foi necessário fazer a coleta de dados por parte dos desenvolvedores, desenvolvendo formulários *web* e analisando os dados enviados, este processo dificultou o desenvolvimento do projeto.

Outra dificuldade foi a falta de trabalhos de referência sobre a integração de ontologias e DSL, o que levou a realização de alguns desenvolvimentos de maneira errada que consumiram tempo da pesquisa.

A dificuldade mais importante enfrentada foi de carácter organizacional, entre a equipe do projeto do ICMC e da Embrapa. Esse problema deve-se a uma diferença entre culturas de áreas diferentes, computação e sustentabilidade, e da existência de interesses heterogêneos, por parte da USP e da Embrapa. A Embrapa não desenvolve apenas pesquisas, mas usa essas pesquisas em programas de extensão agrícola. Assim, aspectos como, por exemplo, registro de software, que no mundo acadêmico tem importância periférica, são muito mais importantes para eles. A Embrapa não permite que programas, desenvolvidos por ela ou em colaboração com ela, sejam disponibilizados para o público em geral antes do registro no Instituto Nacional da Propriedade Industrial (INPI). Como o SAD SustenAgro foi uma colaboração entre Embrapa e USP, a USP deve compartilhar esse registro com a Embrapa. Infelizmente, a burocracia da USP é muito demorada e esse registro demora mais do necessário. A Embrapa tem uma comunidade de produtores de cana, usinas e cooperativas do setor interessados em sustentabilidade. Apesar do SustenAgro ter sido avaliado por um número expressivo de profissionais, seria muito interessante uma avaliação por usuários finais de fazendas e usinas.

7.3 Trabalhos futuros

A partir da análise dos resultados e das dificuldades, foram identificadas várias ideias que permitirão complementar e melhorar os resultados desta pesquisa. Também, a partir das seguintes ideias de trabalhos futuros, fica a possibilidade de definir projetos de pesquisa e desenvolvimento tecnológico.

Metodologia de modelagem de conhecimento dos especialistas

O método de definição de conhecimento, explicado na Seção 5.3, permite propor uma metodologia geral de modelagem de conhecimento de especialistas de domínio. Ela seria uma nova proposta que poderia estar composta das seguintes etapas:

1. Definição textual do conhecimento do domínio do especialista e identificação dos elementos principais.
2. Modelagem de um mapa conceitual, a partir dos elementos principais, que permita classificar os conceitos e estabelecer relacionamentos entre eles.
3. Modelagem da ontologia de maneira gráfica, por meio de um editor visual de ontologias, que faça uso dos conceitos, classificações e relacionamentos, identificados nos mapas conceituais, e permita definir ontologias em formato OWL
4. Geração do formato computável da ontologia e carregá-la automaticamente em um sistema *web*, para permitir o gerenciamento dela.
5. Instanciação da ontologia, por meio de serviços *web*, com instâncias que representem dados reais do domínio para permitir o uso da ontologia e seus dados em outros sistemas.

Essa metodologia facilitará a definição de conhecimento, por parte dos especialistas, por fornecer uma abordagem mais gradual da modelagem.

Editor web de ontologias em OWL

O *Framework Decisioner* integra um editor de ontologias, baseado no formato YAML. Ele dá suporte a edição de ontologias, em tempo de execução, por meio de uma *Web UI*. Ele pode ser complementado com um editor visual de ontologias que permita a visualização e edição em forma de grafos. Um exemplo desse tipo de abordagem é o visualizador de ontologias WebVOWL ¹.

O editor de ontologias em OWL integrará a metodologia de modelagem de conhecimento, proposto anteriormente, e poderá ter um repositório online de conhecimento que permita a reuso.

¹ <<http://vowl.visualdataweb.org/webvowl.html>>

Linguagem simplificada de consultas Sparql

O *DSL Interpreter* tem um módulo que simplifica as consultas SPARQL nas *triple-stores*. Esse módulo é um protótipo que pode ser melhorado e generalizado para facilitar o armazenamento e recuperação de informações semânticas, por meio de uma linguagem mais simples que SPARQL.

Esse módulo pode ser disponibilizado como uma biblioteca que dê suporte ao desenvolvimento de programas que façam uso de Sparql.

Linguagem de edição de Web UIs

As *web UI* para os SADs possuem vários tipos de *web components*. É possível modificar a aparência dessas *web UIs*, usando *Cascading Style Sheets* (CSSs) e uma DSL interna de uso dos programadores. Mas esse tipo de modificação fica além do que especialistas podem fazer.

Para suportar a definição e modificação da organização e apresentação das UI, por parte dos especialistas, é necessária uma linguagem de gerenciamento das *widgets* (elementos de uma *UI*) com suporte a edição gráfica.

Instanciação de outros SAD usando o *Framework Decisioner*

O *Framework Decisioner* foi criado para permitir a instanciação de uma classe de SADs que criam análises e recomendações para um cenário específico. Mas ele foi desenvolvido a partir do caso de uso de apenas um SAD desse tipo, o SustenAgro. Com a instanciação de outros SADs, para áreas diferentes de sustentabilidade, a generalidade do *framework* pode ser testada e, se necessário, mudanças implementadas. Instruções e funcionalidades, que façam parte de outros SADs diferentes do SustenAgro, podem ser integradas ao *framework* para generalizar seu escopo.

Georreferenciamento

Um aspecto importante, para vários tipos de SADs, são a associação de termos e tecnologias de georreferenciamento. Eles são importantes para relacionar conhecimentos associados a locais específicos. No caso dos SADs associados a agricultura, como SustenAgro, isso permitirá a recuperação de dados existentes em fontes geográficas externas, como Geonames², Wikimapia³ e DBpedia⁴.

² <<http://www.geonames.org/>>

³ <<http://wikimapia.org/>>

⁴ <<http://wiki.dbpedia.org/>>

Biblioteca de *Web Components*

O *Framework Decisioner* conta com aproximadamente 60 *web components*. Eles dão suporte na geração de *UIs*, as quais podem ser agrupadas em uma biblioteca, para permitir seu reuso e a definição de *Web Components* especializados, como os desenvolvidos para o SAD SustenAgro.

A biblioteca *Polymer*⁵ suporta a criação de *Web Components* reusáveis para compor *Web UIs*. Dada as suas características recomenda-se a implementação dessa biblioteca com essa tecnologia.

7.4 Agradecimentos

O presente trabalho foi realizado com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) no âmbito do Ministério da Educação do Brasil, quem financiou-o por meio de uma bolsa de estudo de mestrado.

Um especial agradecimento à Dra. Katia Regina Evaristo de Jesus, pesquisadora da Embrapa Meio Ambiente, coordenadora do Projeto 'Avaliação da Sustentabilidade de Sistemas de Produção da Cana-de-açúcar e Soja na Região Centro-Sul do Brasil: Uma proposta metodológica e de modelo conceitual SustenAgro', financiado pela Embrapa. Os dados técnicos empregados no desenvolvimento do Software foram desenvolvidos no escopo do projeto supra citado.

Também um especial agradecimento ao pesquisador Sérgio Alves Torquato da Agência Paulista de Tecnologia do Agronegócio – APTA/ UPD Tietê/SP.

⁵ <<http://wiki.dbpedia.org/>>

Referências

ABT, V.; VIGIER, F.; SCHNEIDER, M. *Enterprise Business Modelling Languages Applied to Farm Enterprise: A Case Study for IDEF0, GRAI Grid, and AMS Languages*. Boston, MA: Springer US, 2009. 167–191 p. ISBN 978-0-387-75181-8. Disponível em: <http://dx.doi.org/10.1007/978-0-387-75181-8_9>. (Citado na página 42.)

ALLEMANG, D.; HENDLER, J. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Elsevier Science, 2011. ISBN 9780123859662. Disponível em: <https://books.google.com.br/books?id=_qGKPOlB1DgC>. (Citado nas páginas 48, 49 e 77.)

BELLEN, H. M. V. *Indicadores de sustentabilidade: uma análise comparativa*. FGV editora, 2005. 256 p. ISBN 85-225-0506-3. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1679-39512005000300012&nrm=iso>. (Citado nas páginas 125 e 129.)

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web. *Scientific American*, v. 284, n. 5, p. 34–43, maio 2001. Disponível em: <<http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>>. (Citado na página 46.)

BIKAKIS, N. et al. The xml and semantic web worlds: technologies, interoperability and integration: a survey of the state of the art. In: *Semantic Hyper/Multimedia Adaptation*. [S.l.]: Springer, 2013. p. 319–360. (Citado na página 46.)

BONACIN, R.; NABUCO, O. F.; JUNIOR, I. P. Conceptualizing the impacts of agriculture on water resources: Experiences and ontology engineering challenges. In: *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems*. New York, NY, USA: ACM, 2013. (MEDES '13), p. 262–269. ISBN 978-1-4503-2004-7. Disponível em: <<http://doi.acm.org/10.1145/2536146.2536185>>. (Citado na página 42.)

BOSSEL, H. *Indicators for sustainable development: theory, method, applications*. International Institute for Sustainable Development Winnipeg, 1999. ISBN 1-895536-13-8. Disponível em: <<http://www.iisd.org/sites/default/files/publications/balatonreport.pdf>>. (Citado na página 129.)

BRILHANTE, V. et al. Information integration through ontology and metadata for sustainability analysis. In: *Proc. of the 3rd biennial meeting of the International Environmental Modelling and Software Society*. [s.n.], 2006. Disponível em: <<http://scholarsarchive.byu.edu/cgi/viewcontent.cgi?article=3579&context=iemssconference>>. (Citado na página 43.)

BRUNDTLAND, G. et al. Our common future: Report of the 1987 world commission on environment and development. *United Nations, Oslo*, p. 1–59, 1987. Disponível em: <<http://www.un-documents.net/our-common-future.pdf>>. (Citado nas páginas 125 e 126.)

BURTON, I. Report on reports: Our common future. *Environment: Science and Policy for Sustainable Development*, v. 29, n. 5, p. 25–29, 1987. Disponível em: <<http://dx.doi.org/10.1080/00139157.1987.9928891>>. (Citado na página 125.)

- CARDOSO, B. O. *Avaliação da sustentabilidade de sistemas de produção da cana-de-açúcar no estado de São Paulo: uma proposta metodológica e de modelo conceitual*. [s.n.], 2013. Disponível em: <http://bdtd.ibict.br/vufind/Record/SCAR_8cc1bc58f68c9fa5cb516bbce3a07483>. (Citado nas páginas 29, 30, 76, 81, 126 e 130.)
- DAVEY, B. Think aloud: Modeling the cognitive processes of reading comprehension. *Journal of Reading*, JSTOR, v. 27, n. 1, p. 44–47, 1983. Disponível em: <https://www.jstor.org/stable/40029295?seq=1#page_scan_tab_contents>. (Citado na página 111.)
- DEARLE, F. *Groovy for Domain-Specific Languages*. 1st. ed. Packt Publishing Ltd, 2015. ISBN 184719690X, 9781847196903. Disponível em: <<https://www.packtpub.com/application-development/groovy-domain-specific-languages-second-edition>>. (Citado nas páginas 55 e 71.)
- EHLERS, E. *Agricultura sustentável: origens e perspectivas de um novo paradigma*. Livros da terra, 1996. ISBN 9788585347376. Disponível em: <https://books.google.com.br/books?id=oaD_PgAACAAJ>. (Citado na página 125.)
- EVANS. *Domain-Driven Design: Tackling Complexity In the Heart of Software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003. ISBN 0321125215. Disponível em: <<https://books.google.com.br/books?id=hHBf4YxMnWMC>>. (Citado nas páginas 36 e 41.)
- EWERT, F. et al. A methodology for enhanced flexibility of integrated assessment in agriculture. *Environmental Science & Policy*, v. 12, n. 5, p. 546 – 561, 2009. ISSN 1462-9011. Integrated Assessment of Agricultural and Environmental Policies – concepts and tools. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1462901109000409>>. (Citado na página 43.)
- FENSEL, D. et al. *Semantic web services*. Springer Science & Business Media, 2011. ISBN 978-3-642-19192-3. Disponível em: <<http://www.springer.com/br/book/9783642191923#aboutBook>>. (Citado nas páginas 46 e 47.)
- FISCHER, R. et al. Can technology deliver on the yield challenge to 2050. In: *Expert Meeting on How to feed the World in*. [s.n.], 2009. v. 2050, p. 1–48. Disponível em: <<http://www.fao.org/3/a-ak542e/ak542e12a.pdf>>. (Citado na página 126.)
- FOWLER, M. *Domain-specific languages*. Pearson Education, 2010. 640 p. (Addison-Wesley Signature Series (Fowler)). ISBN 9780131392809. Disponível em: <https://books.google.com.br/books?id=ri1muolw_YwC>. (Citado nas páginas 31 e 54.)
- GAVRILOVA, T.; ANDREEVA, T. Knowledge elicitation techniques in a knowledge management context. *Journal of Knowledge Management*, v. 16, n. 4, p. 523–537, 2012. Disponível em: <<http://dx.doi.org/10.1108/13673271211246112>>. (Citado na página 29.)
- GHOSH, D. *DSLs in Action*. 1st. ed. Greenwich, CT, USA: Manning Publications Co., 2010. ISBN 9781935182450. Disponível em: <<https://www.manning.com/books/dsls-in-action>>. (Citado na página 54.)
- GLIESSMAN, S. R. *Agroecologia: processos ecológicos em agricultura sustentável*. Ed. da Univ. Federal do Rio Grande do Sul, UFRGS, 2001. Disponível

em: <<http://www.ufrgs.br/pgdr/publicacoes/livros/serie-estudos-rurais-pgdr/agroecologia-processos-ecologicos-em-agricultura-sustentavel>>. (Citado na página 125.)

HEINZLE, R.; GAUTHIER, F. A. O.; FIALHO, F. A. P. Semântica nos sistemas de apoio a decisão: o estado da arte. *Revista Da Unifebe Vers\ ao Eletrônica*, 2010. Disponível em: <<http://periodicos.unifebe.edu.br/index.php/revistaeletronicadaunifebe/article/view/551>>. (Citado na página 35.)

HORRIDGE, M.; BECHHOFFER, S. The owl api: A java api for owl ontologies. *Semantic Web*, IOS Press, v. 2, n. 1, p. 11–21, 2011. ISSN 1570-0844. Disponível em: <<http://dl.acm.org/citation.cfm?id=2019470.2019471>>. (Citado na página 49.)

ITTERSUM, M. K. V. et al. Integrated assessment of agricultural systems—a component-based framework for the european union (seamless). *Agricultural systems*, Elsevier, v. 96, n. 1, p. 150–165, 2008. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0308521X07000893>>. (Citado na página 126.)

JONG, E. de. Software architecture for large control systems: A case study description. In: *Coordination Languages and Models*. Springer, 1997. p. 150–156. Disponível em: <http://dx.doi.org/10.1007/3-540-63383-9_78>. (Citado na página 36.)

KOENIG, D. et al. *Groovy in action*. Manning Publications Co., 2007. v. 91. ISBN 1935182447, 9781935182443. Disponível em: <<https://www.manning.com/books/groovy-in-action-second-edition>>. (Citado na página 55.)

KRAINES, S.; GUO, W. A system for ontology-based sharing of expert knowledge in sustainability science. *Data Science Journal*, CODATA, v. 9, p. 107–123, 2011. Disponível em: <<http://doi.org/10.2481/dsj.Kraines>>. (Citado nas páginas 42 e 82.)

KUMAZAWA, T. et al. Toward knowledge structuring of sustainability science based on ontology engineering. *Sustainability Science*, Springer, v. 4, n. 1, p. 99, 2009. ISSN 1862-4057. Disponível em: <<http://dx.doi.org/10.1007/s11625-008-0063-z>>. (Citado na página 43.)

LEE, Z.; WAGNER, C.; SHIN, H. K. The effect of decision support system expertise on system use behavior and performance. *Information & Management*, v. 45, n. 6, p. 349 – 358, 2008. ISSN 0378-7206. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0378720608000669>>. (Citado na página 30.)

LIANG, A. et al. From agrovoc to the agricultural ontology service / concept server: An owl model for creating ontologies in the agricultural domain. *International Conference on Dublin Core and Metadata Applications*, v. 0, n. 0, 2006. ISSN 1939-1366. Disponível em: <<http://dcpapers.dublincore.org/pubs/article/view/841>>. (Citado na página 42.)

LOHMANN, S. et al. Webvowl: Web-based visualization of ontologies. In: SPRINGER. *International Conference on Knowledge Engineering and Knowledge Management*. 2014. p. 154–158. ISBN 978-3-319-17966-7. Disponível em: <http://dx.doi.org/10.1007/978-3-319-17966-7_21>. (Citado na página 105.)

MATTHEWS, K. et al. Wither agricultural dss? *Computers and Electronics in Agriculture*, v. 61, n. 2, p. 149 – 159, 2008. ISSN 0168-1699. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0168169907002207>>. (Citado na página 30.)

- MCGUINNESS, D. L.; HARMELEN, F. V. et al. Owl web ontology language overview. *W3C recommendation*, v. 10, n. 10, p. 2004, 2004. Disponível em: <<https://www.w3.org/TR/owl-features/>>. (Citado na página 51.)
- MERNIK, M.; HEERING, J.; SLOANE, A. M. When and how to develop domain-specific languages. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 37, n. 4, p. 316–344, dez. 2005. ISSN 0360-0300. Disponível em: <<http://doi.acm.org/10.1145/1118890.1118892>>. (Citado na página 54.)
- MOURA, L. G. V. Indicadores para a avaliação da sustentabilidade em sistemas de produção da agricultura familiar: o caso dos fumicultores de agudo-rs. 2002. Disponível em: <<http://www.lume.ufrgs.br/handle/10183/2624>>. (Citado na página 128.)
- MUSEN, M. A. The protégé project: A look back and a look forward. *AI matters*, ACM, v. 1, n. 4, p. 4–12, 2015. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4883684/>>. (Citado na página 52.)
- NOY, N. F.; MCGUINNESS, D. L. et al. *Ontology development 101: A guide to creating your first ontology*. [S.l.]: Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880, 2001. (Citado nas páginas 48 e 49.)
- OLSSON, J. A. et al. A goal oriented indicator framework to support integrated assessment of new policies for agri-environmental systems. *Environmental Science & Policy*, v. 12, n. 5, p. 562 – 572, 2009. ISSN 1462-9011. Integrated Assessment of Agricultural and Environmental Policies – concepts and tools. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1462901109000446>>. (Citado nas páginas 42 e 126.)
- ONU. The future we want. *Technical Report of United Nations Conference on Sustainable Development, Brazil*, 2012. (Citado na página 125.)
- PATEL-SCHNEIDER, P. F. Building the semantic web tower from rdf straw. In: *In Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*. [S.l.: s.n.], 2005. (Citado na página 49.)
- POWER, D. J. *Decision support systems: concepts and resources for managers*. Greenwood Publishing Group, 2002. ISBN 9781567204971. Disponível em: <<https://books.google.com.br/books?id=9NA6QMcte3cC>>. (Citado nas páginas 29 e 37.)
- PRUD, E.; SEABORNE, A. et al. Sparql query language for rdf. 2006. Disponível em: <<https://www.w3.org/TR/rdf-sparql-query/>>. (Citado na página 58.)
- RIEHLE, D. *Framework Design: A Role Modeling Approach*. Tese (Doutorado) — ETH Zürich, Switzerland, 2000. Ph.D. Thesis, No. 13509. Zürich. (Citado na página 58.)
- ROSPACHER, M.; SERAFINI, L. Ontology-centric decision support. In: *Proceedings of the 2012 International Conference on Semantic Technologies Meet Recommender Systems & Big Data - Volume 919*. Aachen, Germany, Germany: CEUR-WS.org, 2012. (SeRSy'12), p. 61–72. Disponível em: <<http://dl.acm.org/citation.cfm?id=2887638.2887644>>. (Citado na página 43.)

- ROUSSEY, C. et al. Ontologies in agriculture. In: *AgEng 2010, International Conference on Agricultural Engineering*. [S.l.: s.n.], 2010. (Citado nas páginas 29 e 42.)
- RUSHER, J. Triple store. In: *Workshop on Semantic Web Storage and Retrieval-Position Paper*. [S.l.: s.n.], 2003. (Citado na página 52.)
- SAXENA, K. B. C. Decision support engineering: a dss development methodology. In: *Proceedings of the Twenty-Fourth Annual Hawaii International Conference on System Sciences*. [S.l.: s.n.], 1991. iii, p. 98–107 vol.3. (Citado na página 30.)
- SCHWABER, K.; BEEDLE, M. Agile software development with scrum. Prentice Hall, 2002. Disponível em: <<https://books.google.com.br/books?id=BpFYAAAAYAAJ>>. (Citado na página 75.)
- SEABRA, J. E. et al. Life cycle assessment of brazilian sugarcane products: Ghg emissions and energy use. *Biofuels, Bioproducts and Biorefining*, Wiley Online Library, v. 5, n. 5, p. 519–532, 2011. (Citado na página 38.)
- SHIM, J. et al. Past, present, and future of decision support technology. *Decision Support Systems*, v. 33, n. 2, p. 111 – 126, 2002. ISSN 0167-9236. Decision Support System: Directions for the Nest Decade. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167923601001397>>. (Citado na página 37.)
- SIMON, H. A. *The architecture of complexity*. Springer, 1991. v. 106. 467–482 p. Disponível em: <<http://nicoz.net/images/ArchitectureOfComplexity.HSimon1962.pdf>>. (Citado nas páginas 38 e 126.)
- SINGH, R. K. et al. An overview of sustainability assessment methodologies. *Ecological Indicators*, v. 15, n. 1, p. 281 – 299, 2012. ISSN 1470-160X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1470160X11000240>>. (Citado na página 39.)
- SMITH, B. et al. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotech*, Nature Publishing Group, v. 25, n. 11, p. 1251–1255, nov 2007. ISSN 1087-0156. Disponível em: <<http://dx.doi.org/10.1038/nbt1346>>. (Citado na página 48.)
- SMITH, G.; LEDBROOK, P. *Grails in action*. Manning Publications, 2009. ISBN 978-1617290961. Disponível em: <<https://www.manning.com/books/grails-in-action-second-edition>>. (Citado na página 55.)
- SOULIGNAC, V. et al. A knowledge management system for exchanging and creating knowledge in organic farming. *EJKM/The Electronic Journal of Knowledge Management*, v. 10, n. 2, p. p–163, 2012. (Citado na página 43.)
- SOUZA, R. R.; ALVARENGA, L. A web semântica e suas contribuições para a ciência da informação. *Ciência da Informação, Brasília*, SciELO Brasil, v. 33, n. 1, p. 132–141, 04 2004. ISSN 0100-1965. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652004000100016&nrm=iso>. (Citado nas páginas 45 e 46.)
- SUGUMARAN, V.; GULLA, J. A. *Applied semantic web technologies*. CRC Press, 2011. ISBN 978-1-4398-0156-7. Disponível em: <<https://www.crcpress.com/Applied-Semantic-Web-Technologies/Sugumaran-Gulla/p/book/9781439801567>>. (Citado na página 49.)

- TILMAN, D. et al. Agricultural sustainability and intensive production practices. *Nature*, Nature Publishing Group, v. 418, n. 6898, p. 671–677, 2002. Disponível em: <<http://dx.doi.org/10.1038/nature01014>>. (Citado na página 83.)
- TORQUATO, K. R. E. d. J. & C. R. B. Z. S. A. Inovações no sistema de produção de cana-de-açúcar: uma contribuição do protocolo agroambiental para a região de piracicaba, estado de são paulo. *Informações Econômicas*, v. 45, n. 2, p. 10, 2015. (Citado na página 38.)
- TUDHOPE, D.; NIELSEN, M. L. Introduction to knowledge organization systems and services. *New Review of Hypermedia and Multimedia*, v. 12, n. 1, p. 3–9, 2006. Disponível em: <<http://dx.doi.org/10.1080/13614560600856433>>. (Citado na página 30.)
- TURBAN, E.; ARONSON, J. E.; LIANG, T.-P. *Decision Support Systems and Intelligent Systems (7th Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2004. ISBN 9780130461063. Disponível em: <<https://books.google.com.br/books?id=NfMJAQAAMAAJ>>. (Citado na página 29.)
- TWEEDALE, J. W.; PHILLIPS-WREN, G.; JAIN, L. C. *Advances in Intelligent Decision-Making Technology Support*. Cham: Springer International Publishing, 2016. 1–15 p. ISBN 978-3-319-21209-8. Disponível em: <http://dx.doi.org/10.1007/978-3-319-21209-8_1>. (Citado nas páginas 35, 36, 37 e 38.)
- WRIGHT, J. T. A técnica delphi: Uma ferramenta útil para o planejamento do brasil. *Encontro Brasileiro de Planejamento Empresarial- "Como Planejar"*, v. 86, p. 199–207, 1985. (Citado na página 110.)
- WRIGHT JAMES TC, S. O pas no futuro: aspectos metodológicos e cenários. *Estud. av.*, v. 20, p. 13–28, 2006. ISSN 0103-4014. (Citado na página 128.)
- ZORZO, C. R. B. *Caracterização e desenvolvimento de indicadores de sustentabilidade para os sistemas de produção de cana-de-açúcar na região Centro-Sul do Brasil*. [s.n.], 2015. Disponível em: <<https://repositorio.ufscar.br/handle/ufscar/7701>>. (Citado na página 130.)

Método SustenAgro de Avaliação de Sustentabilidade

Este anexo apresenta os principais conceitos relacionados com a avaliação da sustentabilidade, de acordo com o que foi desenvolvido pela equipe do Projeto SustenAgro, e como foram usados no processo de avaliação de sustentabilidade.

A.1 Sustentabilidade

Não existe um consenso sobre a definição de sustentabilidade, mas uma definição orientadora para os fins do presente projeto é a seguinte:

“O desenvolvimento sustentável prevê o atendimento das necessidades do presente sem comprometer a capacidade das gerações futuras de suprir suas próprias necessidades, Brundtland Commission” (BURTON, 1987; BRUNDTLAND et al., 1987)

Este conceito foi ratificado pela Conferência das Nações Unidas sobre o Meio Ambiente e Desenvolvimento, a Rio-92 (EHLERS, 1996) a Rio+20 (ONU, 2012), após do relatório *Brundtland* a ênfase do conceito desloca-se da integridade ambiental para o elemento humano, gerando um equilíbrio entre as dimensões econômica, social e ambiental (BELLEN, 2005).

Gliessman (2001) teoriza que não há como alcançar a sustentabilidade e, portanto, o seu conceito mais representativo, pois a mesma permanece sempre no futuro, dado o compromisso que os sistemas têm de garantir as necessidades das gerações futuras. Assim, a sustentabilidade é algo relativo ao tempo, ou seja, um sistema pode ser mais ou menos sustentável que outro dependendo do tempo em que for avaliado e do entendimento da sustentabilidade neste contexto.

A sustentabilidade está vinculada a vários domínios de conhecimento, um deles é a sustentabilidade em agricultura, que é de especial interesse na segurança alimentar. segundo a FAO em 2050 a população mundial atingirá 9.1 bilhões de pessoas (FISCHER et al., 2009), o qual imporá enormes desafios para garantir a sustentabilidade em meio do aumento de alimentos. Por isso são necessários incentivos e políticas para garantir a sustentabilidade na agricultura, por meio da geração de estratégias que permitam conhecer o estado dos sistemas produtivos e melhorar segundo as necessidades identificadas.

Segundo Ittersum et al. (2008) os sistemas agrícolas evoluem continuamente e são afetados por uma gama de forças globais e locais, sendo os tecnológicos e políticos que mais influenciam na sustentabilidade da agricultura, permitindo identificar e melhorar diversos aspectos da produção agrícola.

Uma estratégia para quantificar a sustentabilidade é a definição métodos e metodologias de avaliação, as quais utilizam indicadores. Um exemplo deste enfoque é exposto por Olsson et al. (2009) que desenvolveu um *framework* de indicadores que relaciona de uma maneira consistente as dimensões ambiental, econômica e social do desenvolvimento sustentável, seu principal benefício é uma relativa simplicidade na apresentação da informação e a possibilidade de vincular os indicadores com objetivos políticos de cada dimensão da sustentabilidade e assim facilitar a comparação dos impactos das novas políticas em cada dimensão.

A.2 Dimensões da sustentabilidade

As dimensões da sustentabilidade são classificações que permitem identificar e agrupar conceitos de sustentabilidade (OLSSON et al., 2009), dependendo da teoria de sustentabilidade escolhida. Existem diversas propostas de dimensões que podem ser usadas segundo a finalidade da pesquisa. Um exemplo desta classificação é a assumida na pesquisa de Cardoso (2013) onde são definidas seis dimensões da sustentabilidade: Ambiental, Social, Agrícola/Industrial, Produtos/Subprodutos, Tecnológica e Política.

No caso do sistema SustenAgro determinou-se pela equipe de especialistas em sustentabilidade fazer uma divisão segundo a proposta do relatório *Brundtland* (BRUNDTLAND et al., 1987), onde foram identificadas as três dimensões da sustentabilidade: ambiental, social e econômica, as quais têm a mesma importância gerando equilíbrio.

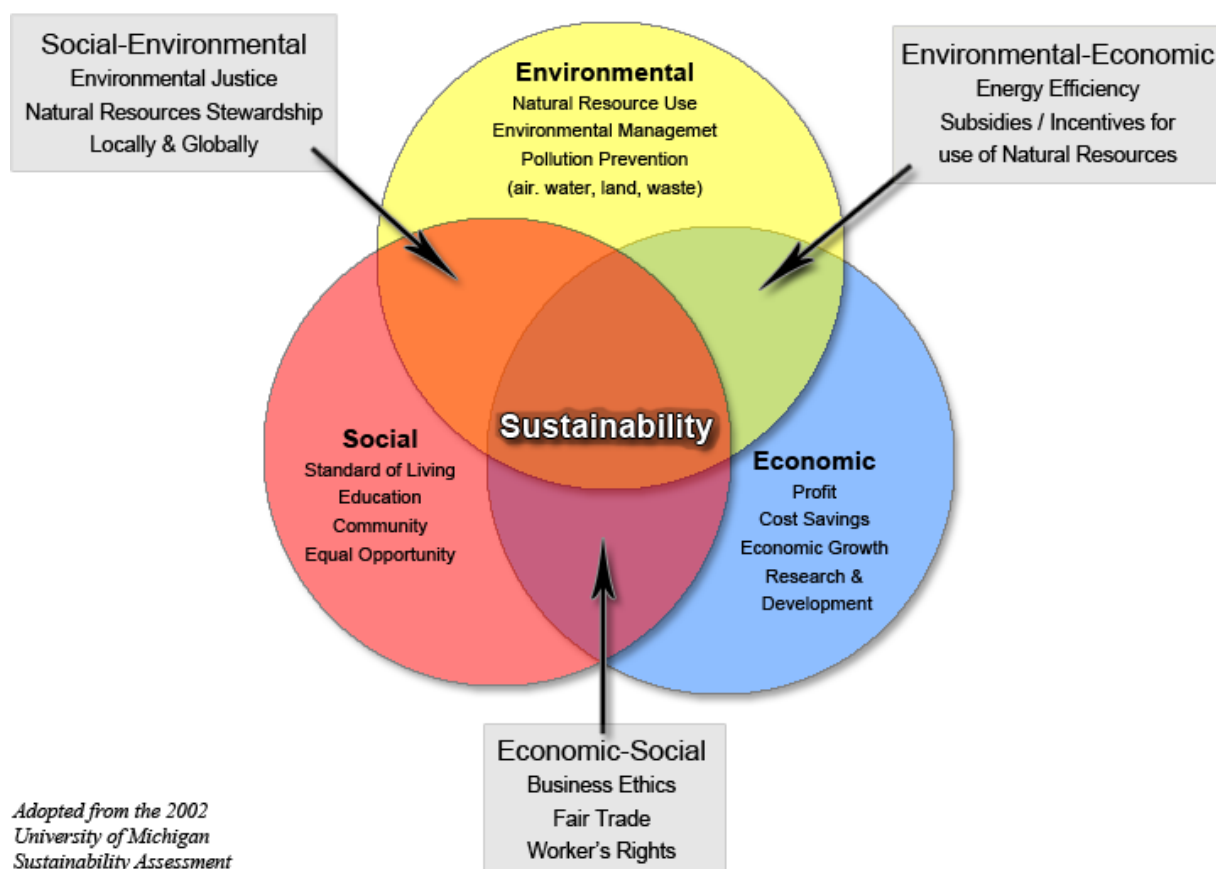
Ditas dimensões são sistemas complexos que integram fenômenos de natureza diversa (SIMON, 1991), integrando três subsistemas: (i) o subsistema ambiental que fornece as condições físicas, químicas e biológicas que suportam o desenvolvimento das culturas, (ii) o subsistema social que integra organizações e pessoas que realizam a produção, relacionando-se internamente e externamente com os sistemas produtivos e (iii) o subsistema econômico que estabelece as condições de oferta e demanda dos produtos e subprodutos do sistema de produção agrícola. Das interações entre estes subsistemas, emerge um

comportamento complexo que requer uma abordagem holística e inter-relacionada para suportar a tomada de decisões que garantam a sustentabilidade do sistema em análise.

A Figura 36 representa as três dimensões com a sustentabilidade como a interseção entre elas.

Figura 36 – Dimensões da sustentabilidade

The Three Spheres of Sustainability



Fonte: <http://www.vanderbilt.edu/sustainvu/cms/files/sustainability_spheres.png>

Essas dimensões serão usadas como contenedores gerais dos conceitos de sustentabilidade em agricultura permitindo agrupar conceitos relacionados.

A.3 Critérios de sustentabilidade

São variáveis transversais quantitativas e qualitativas, que são monitoradas regularmente para determinar os efeitos das atividades de intervenção ou não-intervenção do sistema em avaliação, que estabelecem os preceitos de orientação para que os indicadores sejam representativos para a sustentabilidade.

A formulação de cada indicador de sustentabilidade pela equipe do Projeto SustenAgro teve como orientação básica atender pelo menos um dos critérios de sustentabili-

dade. Os critérios de sustentabilidade definidos pela equipe de especialistas são (MOURA, 2002):

- Produtividade: Relacionado a eficiência e custos.
- Estabilidade: Capacidade do ecossistema de absorver perturbações e permanecer inalterado (Comissão Econômica para a América Latina e o Caribe/Programa das Nações Unidas para o Meio Ambiente, CEPAL/PNUMA, 1994)
- Equidade: Distribuição dos produtos do agroecossistema entre produtores e consumidores (Dias Junior, 2000)
- Resiliência: Capacidade do ecossistema de retornar ao estado original após de uma perturbação (CEPAL/PNUMA, 1994)
- Autonomia: Grau de integração do agroecossistema no fluxo de materiais, energia e informação entre as partes constituintes e entre o agroecossistema e o ambiente externo (Fernández, 1995)

Esses critérios guiam o desenvolvimento dos conceitos mais relevantes das metodologias de avaliação de sustentabilidade, os indicadores, e assim determinar instrumentos de medição que representem os aspectos críticos do sistema em termos de sustentabilidade.

A.4 Método SustenAgro

O método SustenAgro foi construído a partir dos indicadores de sustentabilidade identificados pela equipe do projeto SustenAgro e consiste em uma formula que quantifica os indicadores com a finalidade de suportar o processo de avaliação da sustentabilidade.

Ditos indicadores foram validados por meio da técnica Delphi de consultas aos especialistas (WRIGHT JAMES TC, 2006), esta técnica realiza uma consulta remota e compreende rodada de consultas ao público com perfil amplo para definir conceitos gerais entre os especialistas.

O desenvolvimento do método foi liderado pela pesquisadora Dra. Katia Regina Evaristo de Jesus da Embrapa Meio Ambiente, e a especificação dele é produção intelectual restrita da Embrapa Meio Ambiente.

A.5 Conclusões

O método de avaliação de sustentabilidade foi validado pelos especialistas e depois de várias iterações definiu-se uma versão estável, que foi usada no desenvolvimento do Sistema SustenAgro, dito método é mantido e atualizado pela Embrapa Meio Ambiente e os desenvolvedores de *software* garantem que ele seja aplicado corretamente mas não têm responsabilidade nenhuma pela eficácia da aplicação dele.

Indicadores de Sustentabilidade

B.1 Índice de Sustentabilidade

O índice de sustentabilidade revela o estado de um sistema ou fenômeno, sendo uma síntese das características ou variáveis analisadas. Um índice pode ser construído para analisar dados por meio da junção de um jogo de elementos com relacionamentos estabelecidos.

B.2 Indicadores de Sustentabilidade

Os indicadores são instrumentos usados para avaliar uma determinada realidade levando em conta variáveis pertinentes para sua composição. Além da avaliação, o uso de indicadores permite medir e monitorar aspectos da realidade. Ele agrega, quantifica e simplifica informações sobre fenômenos complexos de modo que as tendências ficam mais significativas e aparentes, a fim de melhorar o processo de entendimento e comunicação (BOSSEL, 1999; BELLEN, 2005).

A OECD 1 *Organisation for Economic Co-operation and Development*¹ estabelece três requisitos para selecionar indicadores: relevância política e utilidade para usuários, solidez analítica e mensurabilidade. Alguns exemplos de indicadores levantados no desenvolvimento do método SustenAgro são:

1. Risco climático;
2. Diversidade de culturas anuais;
3. Tipo de solo;
4. Risco de deficit hídrico;

¹ <http://www.oecd.org/brazil/>

5. Produtividade da terra;
6. Renovabilidade energética nos sistemas de produção;
7. Balanço de nutrientes (nitrogênio e fósforo);
8. Área de cultivo/áreas preservadas.

Os indicadores do presente projeto são uma representação dos fatores críticos que existem no sistema de produção de cana-de-açúcar no centro-sul do Brasil em cada dimensão da sustentabilidade, portanto a metodologia e o sistema SustenAgro é aplicável nesse contexto.

B.3 Dados fornecidos pela Embrapa Meio Ambiente

A principal fonte de dados para este projeto foi fornecida pela pesquisa de Cardoso (2013), onde inicialmente foram identificados 62 indicadores de sustentabilidade no sistema de cana-de-açúcar do centro-sul do Brasil. Estes indicadores foram analisados e caracterizados na pesquisa da ZORZO (2015), gerando 39 indicadores como os mais relevantes por meio de uma validação com porcentagem maior ou igual a 60% feita por uma comunidade de especialistas em sustentabilidade vinculados com o projeto SustenAgro da Embrapa Meio Ambiente.

B.4 Considerações finais

Os dados e especificações fornecidos pela Embrapa Meio Ambiente e pela APTA conseguiram explicar o conceito de avaliação de sustentabilidade segundo a visão da Embrapa Meio Ambiente. A complexidade envolvida requereu identificar um tipo de KOS que permita representar cada um dos conceitos necessários que compõem o processo de avaliação da sustentabilidade. O KOS precisa ser flexível e de fácil uso para conseguir se adaptar às mudanças do domínio, devido a que durante o processo de modelagem avalia a coerência dos dados, permitindo assim melhorar as especificações de dito domínio.

Instalação

A instalação dos Sistemas Decisioner e SustenAgro divide-se em dois processos, a configuração do servidor web e o deploy do arquivo *Web Application Archive* (WAR). A continuação são descritos ambos processos:

C.1 Configuração do servidor

Esta fase do processo consiste em instalar as tecnologias *Java*, *Apache Tomcat*, *WkHtmltoPdf* e a *triplestore Blazegraph*, em um servidor baseado em *linux*. Atualmente o sistema foi configurado e testado em uma máquina virtual com Ubuntu 14.04, *Java OpenJDK 8*, *Apache Tomcat*, *WkHtmltoPdf 0.12.3* e a *triplestore Blazegraph 2.1.0*. A instalação destas tecnologias segue uma orientação padrão que será descrita a seguir:

Instalação do Java

Segundo a documentação de *Java OpenJDK*, a instalação é realizada pelo comando:

Código-fonte C.1 – Instalação do Java

```
sudo apt-get update  
sudo apt-get install openjdk-8-jre
```

Instalação do *Apache Tomcat*

A instalação do *Apache Tomcat* depende da instalação do *Java 8*, e o *Apache Tomcat* versão 7 para suportar a compatibilidade do WAR gerado, isto é documentado no site¹ do

¹ <https://grails.org/wiki/Deployment>

Framework Grails, que exige uma versão 7 de *Apache Tomcat* para suportar o *deploy* dos arquivos *WAR*.

O processo de instalação consiste em fazer *download* dos arquivos binários, extrair eles em `/opt/tomcat/`, exportar as variáveis de entorno e executar o *Web Server*, com código:

Código-fonte C.2 – Instalação do Apache Tomcat

```
wget http://www-eu.apache.org/dist/tomcat/tomcat-7/v7.0.70/bin/
apache-tomcat-7.0.70.tar.gz
```

```
tar xvzf apache-tomcat-7.0.70.tar.gz -C /opt/tomcat
```

```
sudo /opt/tomcat/bin/startup.sh
```

Configurar *users* de *Tomcat* em: `/opt/tomcat/conf/tomcat-users.xml` e acrescentar os *Role* e *User*

Código-fonte C.3 – Configuração dos Users

```
<role rolename="manager-gui"/>
<user username="admin" password="s3cr3t" roles="manager-gui"/>
```

Depois disso, é registrado no final do arquivo `~/.bashrc` os próximos dois comandos que definem as variáveis de entorno

Código-fonte C.4 – Exportar variáveis de entorno

```
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
export CATALINA_HOME=/opt/tomcat
```

Finalmente executar o comando, e verificar a execução do programa na URL `/manager` do domínio do servidor

Código-fonte C.5 – Iniciar o servidor

```
sudo /opt/tomcat/bin/startup.sh
```

Instalação do WkHtmltoPdf

A tecnologia *WkHtmltoPdf* permite converter páginas web em formato HTML a *Portable Document Format* (PDF), suportando a funcionalidade de gerar os reportes em formato PDF. A instalação consiste em fazer *download* dos arquivos binários e a configuração

de um *X Server Virtual* para suportar o renderizado, os comandos são mostrados a continuação:

Código-fonte C.6 – Instalação do wkhtmltopdf

```
wget http://download.gna.org/wkhtmltopdf/0.12/0.12.3/
wkhtmltox-0.12.3_linux-generic-amd64.tar.xz

tar xf wkhtmltox-0.12.3_linux-generic-amd64.tar.xz

cp wkhtmltox/bin/wkhtmltopdf /usr/local/bin/wkhtmltopdf

sudo chmod a+x /usr/local/bin/wkhtmltopdf

sudo apt-get install openssl build-essential xorg libssl-dev
```

Depois disto é criado um *script* wkhtmltopdf.sh em /usr/local/bin/ e contém o seguinte comando:

Código-fonte C.7 – Execução do wkhtmltopdf

```
xvfb-run -a -s "-screen 0 640x480x16" wkhtmltopdf "$@"
sudo chmod a+x /usr/local/bin/wkhtmltopdf.sh
```

O comando wkhtmltopdf.sh é possível converter a PDF desde um sistema sem X11

Instalação da *Triplestore Blazegraph*

A instalação do *Blazegraph* consiste em fazer download do arquivo binário e executar com o arquivo de configuração RWStore2.properties o serviço de *triplestore*.

Código-fonte C.8 – Instalação do Blazegraph

```
wget https://sourceforge.net/projects/bigdata/files/bigdata/
2.1.1/blazegraph.jar

wget https://dl.dropboxusercontent.com/u/24827919/
SustenAgro/RWStore2.properties

java -server -Xmx4g -Dbigdata.propertyFile=RWStore2.properties
-jar blazegraph.jar
```

C.2 *Deploy* do arquivo WAR

O *deploy* do sistema consiste em executar os serviços do *Tomcat 7* e *Triplestore*, e fazer *upload* do arquivo *SustenAgro-1.0.war* ao servidor *Apache Tomcat* para fazer o *deploy* no *path* “/”, e finalmente reiniciar o *Apache Tomcat*.

Código-fonte C.9 – Execução do *sustenagro*

```
wget https://dl.dropboxusercontent.com/u/24827919/  
SustenAgro/sustenagro-1.0.war
```

Formulários de avaliação

D.1 Formulário de avaliação do SAD SustenAgro

O processo consiste em realizar cinco tarefas descritas a seguir, e responder as seguintes três perguntas para cada tarefa:

- Conseguiu realizar a tarefa?
- Teve algum problema/dúvida durante a realização da tarefa?
- Tem sugestões que permitam melhorar/facilitar a realização da tarefa?

Tarefa 1

Cadastrar usuário:

1. Ingressar no software SustenAgro em <http://java.icmc.usp.br:1300/>
2. Dar *click* na opção de 'Inscrever-se' e ingressar no formulário de cadastro de novo usuário
3. Preencher o formulário e enviar
4. Ingressar ao sistema com seus dados cadastrados.

Tarefa 2

Cadastrar uma unidade produtiva:

1. Dar *click* no link de 'Avaliação'
2. Cadastrar no formulário de uma nova unidade produtiva.
3. Preencher o formulário e enviar

Tarefa 3

Preencher dados da avaliação:

Se a anterior tarefa foi realizada com sucesso, o sistema vai encaminhá-lo ao formulário de nova avaliação, em caso negativo pode dar *click* no botão 'Análises' e 'Nova Análise', para realizar:

1. Preencher alguns dados de avaliação de eficiência e custo
2. Preencher alguns dados de avaliação da sustentabilidade
3. Solicitar avaliação por meio do botão 'Avaliar'

Tarefa 4

Visualizar os resultados:

Se a anterior tarefa foi realizada com sucesso, o sistema vai encaminhá-lo a tela de 'Resultados da avaliação', em caso negativo pode dar *click* no botão 'Análises' e selecionar a análise cadastrada, para realizar:

1. Visualizar a planilha de eficiência e custo
2. Visualizar a planilha de sustentabilidade
3. Visualizar o relatório da avaliação
4. Visualizar os dados da avaliação na aba 'Avaliação'

Tarefa 5

Editar avaliação:

1. Na tela de 'Resultados da avaliação' selecionar a aba 'Avaliação'
2. Acrescentar alguns dados de avaliação de eficiência e custo
3. Acrescentar alguns dados de avaliação da sustentabilidade
4. Solicitar a atualização por meio do botão 'Atualizar'

D.2 Formulário de avaliação do *Framework Decisioner*

O processo consiste em realizar cinco tarefas descritas a seguir, e responder as seguintes três perguntas para cada tarefa:

- Conseguiu realizar a tarefa?

- Teve algum problema/dúvida durante a realização da tarefa?
- Tem sugestões para melhorar/facilitar a realização da tarefa?

As seguintes tarefas devem ser realizadas na interface de administração do sistema:

Tarefa 1

Cadastrar novo indicador

1. Ingressar com *login* de administrador no *software* SustenAgro em <http://java.icmc.usp.br:1300/>
2. Dar *click* no link '*Ontology*'
3. Clicar em *Feature*, Indicador, Indicador de Sustentabilidade e Dimensão Social

Cadastrar uma nova dimensão, atributo e indicador a partir da linha 1020 no seguinte formato:

Código-fonte D.1 – Código de novo indicador

AgriculturalIndicator:

```
is_a: SustainabilityIndicator
relevance: 1.0
label:
  - Agricultural dimension @en
  - Dimensão agrícola @pt
```

DeforestationAttribute:

```
is_a: AgriculturalIndicator
label:
  - Atributo desmatamento @pt
  - Deforestation attribute @en
```

DeforestationIndicator:

```
is_a: DeforestationAttribute
label:
  - Indicador desmatamento @pt
  - Deforestation indicator @en
relevance: 2.0
value: YesNo
```

1. Dar *click* na opção "*Save*"
2. Acrescentar a nova dimensão na DSL, clicando na Aba DSL e no *link Main* adicionar na linha 106 o seguinte código

Código-fonte D.2 – Adição de feature na DSL

3. feature 'AgriculturalIndicator'
4. Dar *click* na opção “Save” - Procurar o indicador cadastrado na interface de Usuário, na tela de Usuário na aba Avaliação selecionar uma Unidade Produtiva e ir em Nova Avaliação

Tarefa 2

Editar a fórmula de avaliação

1. Dar *click* no link 'DSL'
2. Procurar o comando *Report* e editar algum valor da fórmula
3. Dar *click* na opção “Save”
4. Visualizar os índices de uma avaliação cadastrada no sistema por meio da aba *Report*, na tela de Usuário na aba Avaliação selecionar uma Unidade Produtiva, selecionar uma Avaliação e um *Report* ligado a ela.

Tarefa 3

Editar os limiares das *widgets* matriz e semáforo de avaliação

1. Dar *click* no link 'DSL'
2. Procurar o comando *sustainabilityMatrix* e editar o valor de *range_x*
3. Procurar o comando *sustainabilitySemaphore* e editar o valor de *range*
4. Dar *click* na opção “Save”
5. Visualizar os índices de uma avaliação cadastrada no sistema, na tela de Usuário na aba Avaliação selecionar uma Unidade Produtiva, selecionar uma Avaliação e um *Report* ligado a ela

Tarefa 4

Editar uma *View*

1. Dar *click* no link 'Views'
2. Selecionar uma *view* de *contact*
3. Editar o título da *view* por meio do comando *pageHeader*

4. Dar *click* na opção “*Save*”
5. Visualizar a *view* na interface do Usuário por meio do link *Contact*

Tarefa 5

Editar *labels* das *internationalizations*

1. Dar *click* no link *internationalization*
2. Selecionar inglês
3. Editar o texto de um *label*
4. Dar *click* na opção “*Save*”
5. Visualizar uma *view* que faça uso do *label* editado na interface do Usuário

D.3 Formulário Delphi para Workshop SustenAgro

Workshop: Validação do Software SustenAgro (v 1.0)

Local: Embrapa Informática Agropecuária

Data: 14/7/2016

Público-alvo: Equipe do Projeto SustenAgro

Coordenação: Katia R. Evaristo de Jesus (Embrapa Meio Ambiente) e Ivo Pierozzi Junior (Embrapa Informática Agropecuária)

Objetivo deste Workshop: Avaliar a qualidade e acuidade do Software SustenAgro em termos da clareza da informação técnica apresentada nas interfaces, com vistas a garantir o entendimento do usuário e possibilitar que a avaliação da sustentabilidade do sistema de produção de cana-de-açúcar seja realizada da melhor forma possível. As informações fornecidas no questionário Mini-Delphi nesse evento serão utilizadas apenas pela equipe do projeto.

Identificação do respondente:

1. Nome: _____

2. Instituição: _____

3. Cargo/Função: _____

4. _____ Telefones _____ para _____ contato:

5. E-mail: _____

Estou de acordo com as informações transcritas neste documento e autorizo que elas sejam utilizadas para as finalidades descritas acima:

Local e Data: _____

Assinatura: _____

MINI-DELPHI¹

Avaliação do Software *SustenAgro* – Avaliação da Sustentabilidade dos Sistemas de Produção de cana-de-açúcar

Regras para análise: assinale abaixo o grau de concordância com a relevância sobre o assunto abordado neste questionário:

- (1) Pouco adequado
- (2) Medianamente adequado
- (3) Plenamente adequado

1. Login e Cadastro do Sistema de Produção

- Cadastrar usuário e efetuar login no sistema;
- Cadastrar Sistema de produção e selecioná-las para avaliação;
- Cadastrar detalhes da avaliação;
- Instruções de uso do Software;

() AVALIAÇÃO DA INTERFACE

Comentários/Melhorias:

2. Avaliação

- Efetuar avaliação por indicadores, definindo Índice de Sustentabilidade do Sistema de produção de cana;
- Efetuar avaliação variáveis de eficiência, definindo Índice de Eficiência do Sistema de produção de cana;
- Preenchimento das justificativas da ponderação dos indicadores;
- Inserção de novos indicadores específicos.

() AVALIAÇÃO DA INTERFACE

Comentários / Melhorias:

3. Apresentação dos Resultados

- Planilhas para apresentação dos resultados das avaliações de Eficiência e de Indicadores;

¹ O Mini-Delphi pode ser conceituado como um processo estruturado de consulta a especialistas, que mantendo as principais características do método Delphi permite a realização de um estudo com grande agilidade em uma única sessão de trabalho ou *workshop* (WRIGHT, 1985; 1994).

- Acesso à visualização da Matriz de Sustentabilidade;
- Acesso à visualização do Semáforo de Sustentabilidade,
- Relatórios para visualização do Gerenciamento da Sustentabilidade;
- Relatórios de detalhes das justificativas da ponderação dos indicadores;

() AVALIAÇÃO DA INTERFACE

Comentários / Melhorias:

4. Gestão da Avaliação / Informação

- Gestão e recuperação da informação dentro do Software;
- Lista de Indicadores, lista de variáveis, criação de indicadores específicos
- Perfil do usuário e do sistema de produção avaliado.

() AVALIAÇÃO DA INTERFACE

Comentários / Melhorias:

5. Contribua com alguma(s) sugestão(ões) para melhorar a Metodologia e o Software SustenAgro:

6. De modo geral, qual foi sua impressão sobre o Software SustenAgro? O empregaria para avaliar algum sistema de produção de cana-de-açúcar?

7. INFORMAÇÕES ADICIONAIS

Existe alguma informação adicional que não foi citada neste questionário e que merece ser relatada sobre o Software SustenAgro?

Sim

Não

Comentários/Sugestões do respondente:

Existe alguma contribuição adicional em relação à avaliação da sustentabilidade na agricultura que merece ser relatada?

Sim

Não

Comentários/Sugestões do respondente:

Comentários adicionais do respondente:

Referência citada:

WRIGHT, J. T. C. A técnica Delphi: programa de estudos do futuro. São Paulo: FEA, USP, 1994, p.31

WRIGHT, J. T. C. A técnica Delphi: Uma ferramenta útil para o planejamento do Brasil? In: Encontro Brasileiro de Planejamento Empresarial – “Como Planejar 86”, 3, 28-29 nov.1985, Anais. São Paulo: SPE – Sociedade Brasileira de Planejamento Empresarial, 1986, p. 199-207.