

GLAUBER LUIZ DA SILVA COSTA

**AVALIAÇÃO DO *FRAMEWORK* CONCEITUAL FEDORA COMO  
REPOSITÓRIO DE PADRÕES DE ANÁLISE PARA UMA  
INFRAESTRUTURA DE REUSO DE ARTEFATOS  
COMPUTACIONAIS**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

VIÇOSA  
MINAS GERAIS - BRASIL  
2013

Ficha catalográfica preparada pela Seção de Catalogação e  
Classificação da Biblioteca Central da UFV

T

C837a  
2013

Costa, Glauber Luiz da Silva, 1974-

Avaliação do *framework* conceitual fedora como repositório de padrões de análise para uma infraestrutura de reuso de artefatos computacionais / Glauber Luiz da Silva Costa. – Viçosa, MG, 2013.

xiii, 60f. : il. (algumas color.) ; 29 cm.

Orientador: Jugurta Lisboa Filho.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f.57-60.

1. Framework (Programa de computador). 2. FEDORA (Programa de computador). 3. Metadados. I. Universidade Federal de Viçosa. Departamento de Informática. Programa de Pós-Graduação em Ciência da Computação. II. Título.

CDD 22.ed. 005.3

**GLAUBER LUIS DA SILVA COSTA**

**AVALIAÇÃO DO FRAMEWORK CONCEITUAL FEDORA COMO  
REPOSITÓRIO DE PADRÕES DE ANÁLISE PARA  
UMA INFRAESTRUTURA DE REUSO DE  
ARTEFATOS COMPUTACIONAIS**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 30 de julho de 2013.

  
Flavio Vieira Pontes

  
José Luís Braga  
(Coorientador)

  
Juguia Lisboa Filho  
(Orientador)

*Dedico este trabalho a todos que acreditam que os limites só existem para deixar a vida mais emocionante.*

## AGRADECIMENTOS

Ao longo desses anos e mais de 25 mil km rodados como estudante de mestrado foram vários os obstáculos que só foram superados com o apoio das muitas pessoas que torceram por mim e me incentivaram. Citar a todos e as circunstâncias em que foram atuantes nessa caminhada me levaria a escrever um texto tão grande quanto esta dissertação. Porém, ainda que corra o grande risco de cometer injustiça com alguma dessas pessoas, me arrisco a agradecer nominalmente a estes que se tornaram mestres comigo.

A Deus, todo poderoso, por ter me concedido tudo o que precisei para cumprir mais esta etapa em minha vida.

Sr. Ozeias e D. Maria. Meus primeiros professores. Meus mais sábios professores. Meus maiores incentivadores. Meus pais. Aqueles que primeiramente me mostraram que meu único limite são as barreiras físicas às quais todos estão sujeitos e que essas barreiras podem ser intransponíveis a quem as enxerga como tal ou podem ser encaradas apenas como uma forma de fazer com que outro ser humano se sinta bem por poder ajudar a superar. Dizer obrigado é pouco.

Cleyder, meu irmão. Meu primeiro amigo. Pai do João Victor. Nunca mediu esforços para que eu chegasse onde cheguei. Não raras foram as vezes que abriu mão de seu próprio conforto em detrimento de minhas necessidades. A começar do colo da mãe, ao qual teve que abdicar muito cedo, ainda que fosse o caçula. Dizer obrigado é pouco.

Aos amigos de quem me ausentei. Os verdadeiros continuam sempre do meu lado e sei que a distância não alterou em nada nossa amizade. Dizer obrigado é pouco.

Aos professores e colegas do Departamento de Informática da Universidade Federal de Viçosa. A experiência que obtive ao compartilhar esses anos com todos é inestimável. Aos amigos que conheci durante o curso Ronney, Alexandre, Dangelo, Lucas e Wellington Ribeiro e às antigas amigas que reencontrei na UFV Luciana e Simone. Dizer obrigado é pouco.

Ao meu orientador, Jugurta. O mestre por excelência. Profissionalmente um espelho do professor que um dia pretendo ser. Agradeço às inúmeras vezes que esteve ao meu lado, direcionando, sendo paciente e sendo paciente de novo. Dizer obrigado é pouco.

Ao grande amor da minha vida, Eliséia. Meu sustentáculo. Minha inspiração. Há

alguns anos atrás uma conversa que tivemos mudou os rumos da minha vida, me fez ter atitudes diferentes, me fez crer muito mais em mim e vai sempre ser fonte de inspiração. Foram muitas as minhas ausências, foram muitas as noites e fins de semana que abriu mão de sua vida pessoal e apesar de tudo isso sua torcida por mim sempre aumentou. Dizer obrigado é pouco.

# SUMÁRIO

<i>ÍNDICE DE TABELAS</i> .....	<i>vii</i>
<i>ÍNDICE DE FIGURAS</i> .....	<i>viii</i>
<i>LISTA DE ABREVIATURAS</i> .....	<i>ix</i>
<i>RESUMO</i> .....	<i>xi</i>
<i>ABSTRACT</i> .....	<i>xiii</i>
<b>1. INTRODUÇÃO</b> .....	<b>1</b>
<b>1.1. O problema e sua importância</b> .....	<b>1</b>
<b>1.2. Objetivos</b> .....	<b>3</b>
<b>1.3. Metodologia</b> .....	<b>4</b>
<b>2. REVISÃO BIBLIOGRÁFICA</b> .....	<b>6</b>
<b>2.1. Padrões e suas aplicações na computação</b> .....	<b>6</b>
2.1.1 Contextualização histórica de padrões .....	6
2.1.2 O uso de padrões na computação .....	8
2.1.3 Padrões de análise .....	9
<b>2.2. APRI</b> .....	<b>11</b>
<b>2.3. Padrões de Metadados</b> .....	<b>14</b>
<b>2.4. Repositórios de dados</b> .....	<b>20</b>
<b>2.5. Serviços Web</b> .....	<b>21</b>
<b>3. ANÁLISE DE CANDIDATOS A COMPONENTES DO REPOSITÓRIO DE PADRÕES DE ANÁLISE</b> .....	<b>23</b>
<b>3.1. O Modelo Open Archive Information Systems (OAIS)</b> .....	<b>24</b>
3.1.1 O ambiente .....	24
3.1.2 A informação .....	25
3.1.3 Aspectos estruturais e de interoperabilidade .....	28
<b>3.2. Ambientes de desenvolvimento de repositórios de dados</b> .....	<b>29</b>
<b>3.3. Conclusão</b> .....	<b>32</b>
<b>4. FEDORA</b> .....	<b>33</b>

<b>4.1.</b>	<b>Contextualização histórica do Fedora.....</b>	<b>34</b>
<b>4.2.</b>	<b>Estrutura do Fedora .....</b>	<b>35</b>
4.2.1	Objetos Digitais .....	35
4.2.2	O Digital Object Identifier .....	37
4.2.3	Datastreams.....	37
<b>4.3.</b>	<b>Relacionamentos entre objetos no Fedora .....</b>	<b>38</b>
<b>4.4.</b>	<b>A Arquitetura de Modelo de Conteúdo.....</b>	<b>39</b>
<b>4.5.</b>	<b>Servidor de repositório do Fedora.....</b>	<b>40</b>
<b>4.6.</b>	<b>Conclusão.....</b>	<b>41</b>
<b>5.</b>	<b><i>ARQUITETURA PARA UM PORTAL DE ACESSO A APRI.....</i></b>	<b>42</b>
<b>5.1.</b>	<b>Descrição da arquitetura .....</b>	<b>43</b>
<b>5.2.</b>	<b>Componentes da arquitetura .....</b>	<b>44</b>
5.2.1	Islandora.....	45
5.2.2	Drupal .....	46
5.2.3	Componentes de busca.....	47
<b>5.3.</b>	<b>Conclusão.....</b>	<b>49</b>
<b>6.</b>	<b><i>VIABILIZAÇÃO TECNOLÓGICA DA APRI: UMA VISÃO GERAL.</i></b>	<b>50</b>
<b>6.1.</b>	<b>Modelo de Conteúdo para o DC2AP .....</b>	<b>50</b>
<b>6.2.</b>	<b>Interação com o sistema .....</b>	<b>51</b>
<b>6.3.</b>	<b>Conclusão.....</b>	<b>54</b>
<b>7.</b>	<b><i>CONCLUSÕES .....</i></b>	<b>55</b>
<b>8.</b>	<b><i>REFERÊNCIAS BIBLIOGRÁFICAS.....</i></b>	<b>57</b>

## ÍNDICE DE TABELAS

TABELA 1 - ELEMENTOS SIMPLES DO DUBLIN CORE.....	17
TABELA 2 - ELEMENTOS QUALIFICADOS DO DUBLIN CORE.....	18

## ÍNDICE DE FIGURAS

FIGURA 1 - APLICAÇÃO DOS PADRÕES ÀS FASES DE DESENVOLVIMENTO DE SOFTWARE.....	9
FIGURA 2 - ESCOPO DE APLICAÇÃO DE PADRÃO DE ANÁLISE.....	9
FIGURA 3 - ARQUITETURA DA APRI .....	13
FIGURA 4 - MODELO DE AMBIENTE DE UM OAIS .....	25
FIGURA 5 - OBTENÇÃO DE INFORMAÇÃO SIGNIFICATIVA DE DADOS .....	26
FIGURA 6 - RELACIONAMENTO ENTRE OS COMPONENTES DO PACOTE DE INFORMAÇÃO.....	27
FIGURA 7 - ENTIDADES FUNCIONAIS DOS OAIS.....	28
FIGURA 8 - MODELO UML DE UM OBJETO DE INFORMAÇÃO.....	29
FIGURA 9 MODELO DE DADOS DE OBJETOS DIGITAIS DO FEDORA .....	36
FIGURA 10 - PERSPECTIVA DE ACESSO DE UM OBJETO DIGITAL .....	38
FIGURA 11 - RELACIONAMENTO ENTRE OS TIPOS FUNDAMENTAIS DO FEDORA NA CMA.....	39
FIGURA 12 INTERAÇÃO CLIENTE/WEB SERVICES .....	40
FIGURA 13 - ARQUITETURA DA SOLUÇÃO DE INTEGRAÇÃO ENTRE FEDORA, DRUPAL PARA REALIZAÇÃO DA APRI .....	44
FIGURA 14 - A ARQUITETURA DA SOLUÇÃO .....	45
FIGURA 15 - GRAFO RDF .....	48
FIGURA 16 - CONSTRUTOR DE FORMULÁRIO .....	52
FIGURA 17 - XML DE DESCRIÇÃO DO FORMULÁRIO .....	53
FIGURA 18 - TELA DE PREVISÃO DO FORMULÁRIO .....	53

## LISTA DE ABREVIATURAS

AIP	<i>Archival Information Package</i>
API	<i>Application Programming Interface</i>
APRI	<i>Analisis Patterns Reuse Infrastructure</i>
BPEL4WS	<i>Business Process Execution Language for Web Services</i>
CCDS	<i>The Consultative Commite for Space Data Systems</i>
CMA	<i>Content Model Architecture</i>
CModel	<i>Content Model</i>
DARPA	<i>Defense Advanced Research Project Agency</i>
DC2AP	<i>Dublin Core to Analisis Patterns</i>
DCMI	<i>Dublin Core Metadata Initiative</i>
DIP	<i>Dissemination Information Package</i>
FAO	<i>Food and Agriculture Organization</i>
Fedora	<i>Flexible Extensible Digital Object Architecture</i>
FOXML	<i>Fedora Object XML</i>
GPL	<i>General Public License</i>
HTML	<i>Hypertext Markup Language</i>
IDE	<i>Infraestrutura de Dados Espaciais</i>
NFS	<i>National Science Foundation</i>
OAI-PMH	<i>Open Archives Initiative/Protocol for Metadata Harvesting</i>
OAIS	<i>Open Archival Information Systems</i>
PDI	<i>Preservation Description Information</i>
PID	<i>Persistent Identifier</i>
PURL	<i>Persistent-URL</i>
RDF	<i>Resource Description Framework</i>
SDef	<i>Service Definition Object</i>
SDep	<i>Service Deployment Object</i>
SDI	<i>Spatial Data Infrastructure</i>
SIG	<i>Sistema de Informações geográficas</i>
SIP	<i>Submission Information Package</i>
SOA	<i>Service Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
SOC	<i>Service Oriented Computing</i>

UML	<i>Uniform Modeling Language</i>
URI	<i>Universal Resource Identifier</i>
URL	<i>Universal Resource Locator</i>
W3C	<i>World Wide Web Consortium</i>
WSDL	<i>Web Service Description Language</i>
XMI	<i>XML Metadata Interchange</i>
XML	<i>eXtensible Markup Language</i>

## RESUMO

COSTA, Glauber Luiz da Silva, M.Sc., Universidade Federal de Viçosa, julho de 2013. **Avaliação do framework conceitual Fedora como repositório de padrões de análise para uma infraestrutura de reuso de artefatos computacionais.** Orientador: Jugurta Lisboa Filho. Coorientador: José Luís Braga.

Padrão de análise é um importante instrumento no processo de desenvolvimento de software, contribuindo fortemente para o aumento da qualidade do produto final e na redução dos custos de produção. O principal benefício que se pode extrair deste tipo de padrão é o conhecimento estruturado que ele disponibiliza com relação à solução de problemas recorrentes em diversos contextos. A baixa disseminação dos padrões de análise se deve a fatores como a grande variedade de formatos nos quais estão disponibilizados. Em geral estes padrões estão documentados em livros, artigos e outros artefatos físicos ou eletrônicos que tornam difícil o seu compartilhamento. Uma abordagem utilizada para superar este problema é o uso de infraestruturas informacionais que garantam a implementação de ambientes de preservação e disseminação de conteúdo digital que utilizam estruturas de organização da informação. As infraestruturas informacionais são arcabouços compostos por diversos elementos como metadados, protocolos de comunicação e ambientes computacionais orientados a serviços. Estes elementos atuam para que se garanta a coleta, preservação, documentação, acesso, disseminação e intercâmbio dos dados da infraestrutura. A Infraestrutura de Reuso de Padrões de Análise (APRI) propõe uma abordagem que possibilita a documentação, armazenamento, intercâmbio e disseminação dos padrões de análise por meio de um repositório de padrões de análise, os quais são documentados por meio de um perfil de aplicação do padrão de metadados Dublin Core, denominado DC2AP, que pode ser acessado via serviços Web. Este trabalho analisou diferentes alternativas para viabilizar tecnologicamente a APRI e apontou uma arquitetura em três camadas para fornecer acesso aos padrões de análise tanto para usuários humanos como para consumidores de serviços Web. Esta infraestrutura tem como camada base o *framework* de gerenciamento de repositório de metadados Fedora. Na camada intermediária está o *framework* de integração chamado Islandora, que se interliga ao Sistema de Gerenciamento de Conteúdo Drupal, o qual fornece a interface gráfica de

acesso aos padrões de análise. Para testar a viabilidade da arquitetura foi implementado um protótipo que possibilita o registro e a recuperação de padrões de análise utilizando a arquitetura proposta.

## ABSTRACT

COSTA, Glauber Luiz da Silva, M.Sc., Universidade Federal de Viçosa, July 2013.  
**Evaluation of the Fedora conceptual framework as an analysis patterns repository for a computing artifacts reuse infrastructure.** Adviser: Jugurta Lisboa Filho.  
Coadviser: José Luís Braga.

Analysis pattern is a very important instrument in the software development process, strongly contributing for the final product quality and in the production costs reduction. The main benefit that can be extracted from this kind of pattern is that it provides structured knowledge regarding the solution of recurring problems in different contexts. The low dissemination of analysis pattern is due to factors such as the variety of formats in which they are available. In general these standards are documented in books, papers and other physical or electronic artifacts that make the sharing process very difficult. An approach to overcome this problem is the use of informational infrastructure that can ensure the implementation of environment preservation and dissemination of digital content that use organizational structures of information. The informational infrastructure frameworks are composed of various elements such as metadata, communication protocols and service oriented computing environments. These elements act to ensure that the harvest, preservation, documentation, access, dissemination and infrastructure's data exchange. The Analysis Pattern Reuse Infrastructure (APRI) proposes an approach which enables the documentation, storage, exchange and dissemination of the analysis patterns by means of an analysis patterns repository, which are documented by an application profile of the Dublin Core metadata standard called DC2AP, which can be accessed via Web services. This study analyzed different alternatives to allow APRI technologically and pointed to a three-tier architecture to provide access to analysis patterns for both human users and Web services consumers. This infrastructure has the management framework Fedora metadata repository as the base layer. In the intermediate layer is the Islandora integration framework, which connects to the Drupal Content Management System, which provides a graphical interface for access to analysis patterns. To test the feasibility of the architecture, a prototype that enables the recording and retrieval of analysis patterns using the proposed architecture was implemented.

# 1. INTRODUÇÃO

## 1.1.O problema e sua importância

Uma prática adotada em diversas áreas do conhecimento humano, entre elas a área de tecnologia da informação, é a possibilidade de se reutilizar ideias ou soluções que obtiveram resultados positivos em situações semelhantes em diferentes contextos. Pesquisas na área de engenharia de software sugerem que esta prática, denominada reuso, tem se mostrado como uma abordagem bastante efetiva na obtenção de resultados positivos no desenvolvimento de software (PURAO et al., 2003).

O reuso pode ocorrer tanto pelo reaproveitamento de componentes de software já utilizados previamente em outros projetos quanto pela aplicação de padrões, que são a sistematização e documentação de conhecimentos sobre determinado problema. Exemplos da aplicação de padrões podem ser encontrados nas diversas fases do processo de desenvolvimento de sistemas: padrões de análise, na fase de análise, padrões de projeto na fase de projeto e padrões de código na fase de desenvolvimento (BLAIMER et al., 2010).

Padrões, de uma forma geral, apresentam soluções para problemas recorrentes na engenharia de software (PANTOQUILHO et al., 2003) e oferecem como característica mais importante a estruturação de um conhecimento especializado. Esta estruturação possibilita atingir resultados como maior agilidade no desenvolvimento de sistemas, maior flexibilidade destes sistemas e tornar os projetos que utilizam este conhecimento menos suscetíveis a erros (BLAIMER et al., 2010).

Os padrões de análise, especificamente, têm um grande potencial para redução dos custos de um projeto por contribuir para a diminuição do número de falhas do software (BLAIMER et al., 2010) uma vez que erros detectados nas fases iniciais do projeto são menos onerosos que aqueles encontrados em suas fases finais (GLASS, 2002). Isto ocorre porque o foco deste padrão está na construção de estruturas conceituais de processos de negócios ao invés de ater-se apenas à implementação do sistema (PANTOQUILHO et al., 2003).

A utilização de padrões de análise é, portanto, uma abordagem que traz contribuições significativas no sentido de se obter os benefícios da reutilização em seus vários aspectos. No entanto, a utilização de padrões de análise de forma efetiva é limitada por questões como a grande variedade de formatos da sua documentação e a consequente dificuldade em disponibilizá-la adequadamente à comunidade de usuários. Esta barreira é recorrente e áreas como a biblioteconomia e Sistemas de Informações Geográficas (SIG) fazem uso de abordagens que contribuem significativamente para contornar o problema de limitação do acesso a informações sobre suas respectivas áreas de atuação.

No campo de estudo de Sistemas de Informação Geográfica, as Infraestruturas de Dados Espaciais (IDE) fornecem um arcabouço para a criação, manutenção e distribuição de dados espaciais em diferentes níveis de organização. Uma das partes fundamentais das IDE é justamente a forma de documentação de dados espaciais, utilizando metadados como forma de tornar as informações disponíveis e intercambiáveis entre sistemas que possuem o mesmo objetivo final (NOGUERAS-ISO et al., 2005). Metadados são dados sobre dados e é através deles que são feitas as documentações tanto dos dados geográficos em si quanto dos serviços que a infraestrutura disponibiliza (NEBERT, 2008).

De forma análoga ao que ocorre com os dados espaciais, os padrões de análise podem ser documentados usando-se um padrão de metadados que consiga atender às demandas específicas deste contexto, oferecendo a possibilidade de compartilhamento dos padrões de análise de forma estruturada a grupos de usuários interessados no assunto.

Além da documentação consistente e estruturada, o compartilhamento das informações acerca dos padrões de análise depende de como estas ficarão acessíveis a seus potenciais usuários. Uma característica inerente a sistemas de computação que utilizam o paradigma de computação orientada a serviços é a possibilidade de compartilhar o produto de determinado ambiente independente de plataforma ou linguagem de programação por meio de *Web Services*. Estes são aplicações modulares, auto contidas e auto descritivas que podem ser publicadas, localizadas e invocadas através da Web (BLAIMER et al., 2010). Estes componentes de software permitem

interações diretas entre agentes de softwares usando trocas de mensagens via protocolos de Internet (FERRIS; FARRELL, 2003).

Com o intuito de contribuir para a construção de um ambiente que possibilite a disseminação e consequente evolução dos padrões de análise usando uma abordagem voltada a *Web Services*, o grupo de Sistemas de Informação do Departamento de Informática da Universidade Federal de Viçosa propôs a criação da APRI (*Analysis Patterns Reuse Infrastructure*) – Infraestrutura de Reuso de Padrões de Análise (VEGI; PEIXOTO; et al., 2012). Esta infraestrutura propõe a criação de um repositório de padrões de análise documentados a partir de um perfil de metadados baseado no padrão Dublin Core, chamado– *Dublin Core Application Profile to Analysis Patterns* (VEGI; LISBOA-FILHO; et al., 2012). Cada elemento proposto no DC2AP é associado a um *Universal Resource Identifier* – (URI) descrito semanticamente como um *Resource Description Framework* – (RDF). Esta abordagem de descrição dos padrões de análise permite a utilização do conceito de *Linked Data* fornecendo meios de disponibilização de padrões de análise em um ambiente onde dados podem ser processados direta ou indiretamente por máquinas (VEGI et al., 2013).

## 1.2. Objetivos

A partir da proposta de uma infraestrutura de reuso de padrões de análise que permite documentar, compartilhar e recuperar estes padrões segundo necessidades específicas de cada projeto de desenvolvimento de software, o objetivo geral deste trabalho é disponibilizar um ambiente de gestão de padrões de análise baseado em serviços. Este ambiente deve possibilitar a interação de agentes externos, sejam eles humanos ou outros sistemas computacionais, com os padrões armazenados no repositório de acordo com o perfil de metadados DC2AP por meio de *Web Services*, tornando possível seu uso para documentação de padrões de análise. A viabilização tecnológica da APRI visa, portanto, fornecer um ambiente onde:

- O perfil de metadados DC2AP esteja disponível através de *Web Services*, tornando possível seu uso para documentação de padrões de análise;

- Os padrões de análise documentados estejam disponíveis à comunidade de interessados na área, possibilitando seu uso, disseminação e evolução;
- A relação entre os padrões de análise e os serviços que permitam sua disseminação, uso e evolução sejam documentadas por meio de RDF tornando possível a recuperação destas informações a partir de *Linked Data*;
- A gestão de dados e metadados possa ser efetuada a partir da utilização de *Web Services* tornando estes repositórios acessíveis aos potenciais usuários por um meio que lhes forneça recursos para fazer uso dos padrões de análise em ambientes diversificados.

Os objetivos específicos deste trabalho são:

- Analisar as ferramentas computacionais capazes de dar suporte tecnológico ao desenvolvimento de repositórios de fins gerais;
- Mapear os metadados propostos no DC2AP no modelo fornecido pela ferramenta escolhida para viabilizar o repositório de dados;
- Disponibilizar ambiente de registro dos padrões de análise de acordo com os metadados do DC2AP.

### **1.3. Metodologia**

O método de pesquisa utilizado para o desenvolvimento deste trabalho consistiu no levantamento de soluções tecnológicas pré-existentes; na análise destas soluções levando em conta quesitos determinados por normas internacionais e outros critérios que indicam a viabilidade da aplicação da solução escolhida; no mapeamento do perfil de padrão de análise DC2AP de modo a torna-lo aplicável em um ambiente computacional e na criação de um protótipo denominado PANDORA, que viabiliza tecnologicamente as premissas determinadas para a criação da APRI.

A análise de soluções tecnológicas visava elencar e definir ferramentas que poderiam contribuir para a criação de um ambiente que possibilitasse o armazenamento, preservação e disseminação de informações sobre padrões de análise. A proposta da APRI de definição de um ambiente de preservação de objetos digitais para

documentação de padrões de análise que fosse orientado a serviços e fosse capaz de disponibilizar facilidades de acesso a usuários a partir de um portal web. Partindo destas necessidades e de soluções propostas para problemas análogos foram estudados padrões internacionais de modelos como o *Open Archives* e protocolos como o OAI-PMH que visam, respectivamente, contribuir para o estabelecimento de normas técnicas e organizacionais de estruturas de publicação aberta e o estabelecimento de uma interface de acesso aos metadados fornecendo interoperabilidade e extensibilidade de objetos digitais.

Com base nos padrões e protocolos analisados, a escolha da ferramenta mais adequada às necessidades da APRI levou ainda em consideração critérios como modelo de licença sob o qual estão liberadas as soluções, aderência aos protocolos internacionais e adequação às necessidades do próprio projeto da APRI. A ferramenta escolhida foi o Fedora, um *framework* conceitual capaz de prover um servidor de repositório apto a armazenar qualquer tipo de informação digital.

Definida a ferramenta foi criado um protótipo que permitisse mostrar ser possível alcançar as metas de disponibilizar um ambiente para disponibilização de padrões de análise que propiciasse acesso tanto a usuários humanos quanto a outros sistemas computacionais via serviços web. Para a construção de um protótipo que atendesse a estas características foi necessário definir uma arquitetura que permitisse a introdução de um Sistema de Gestão de Conteúdo que se valeu dos *web services* do Fedora para oferecer uma interface gráfica com o usuário final, uma vez que a interface padrão do Fedora requer compreensão de conceitos específicos do domínio de repositório de dados digitais.

## 2. REVISÃO BIBLIOGRÁFICA

A disseminação dos padrões de análise e conseqüente evolução destes padrões requer o uso de técnicas e ferramentas que propiciem a entrega da informação a seu público alvo de maneira ampla, de fácil recuperação e com poucas limitações no que diz respeito ao formato do dado a ser disponibilizado. Este capítulo faz uma introdução aos aspectos que precisam ser considerados para um bom entendimento da solução proposta.

Na seção 1 é feita uma definição sobre padrões de análise, traçando um breve histórico e delineando as principais características e objetivos destes no ciclo de desenvolvimento do software e dos seus modelos de representação. A seção 2 apresenta a APRI, uma infraestrutura que propõe o armazenamento dos padrões de análise de maneira que estes se tornem acessíveis tanto para usuários humanos quanto para máquinas. A seção 3 trata de padrões de metadados utilizados para descrever qualquer tipo de objeto e apresenta também o padrão de metadados Dublin Core, usado pela APRI como base para a descrição do seu perfil para descrição de padrões de análise, o DC2AP. A seção 4 apresenta o conceito de repositório de objetos digitais como um arcabouço de métodos, protocolos e tecnologias que propiciam a gestão de dados e metadados acerca de um assunto específico ou de cunho mais particular. A seção 5 apresenta o conceito de Serviços Web que trata de um aspecto importante na promoção da interoperabilidade entre sistemas de repositório de dados e seus clientes, sejam humanos ou máquinas.

### 2.1. Padrões e suas aplicações na computação

#### 2.1.1 Contextualização histórica de padrões

É comum, em diversas áreas do conhecimento humano, a ideia da reutilização de conhecimentos adquiridos em experiências anteriores e que tenham surtido efeitos positivos quando aplicados repetidamente. Este conceito denominado “padrão”, tem como objetivo servir como referência ou modelo para auxiliar na determinação de soluções de problemas análogos em determinado contexto. Coad em seu artigo seminal *Object-Oriented Patterns* (COAD, 1992) definiu padrão como sendo: “Uma forma original plenamente realizada; modelo proposto ou aceito por imitação. Com padrões,

pequenas partes são padronizadas em grandes partes ou unidades. Padrões se tornam blocos de construção para projetos e construções”.

Junto a esta definição Coad acrescentou exemplos de aplicação de padrões em áreas distintas como psicologia, arqueologia, linguística, numismática, decoração, construção civil, entre outros. A partir desta premissa afirmou que o uso e aplicação de padrões indica o nível de maturidade de determinado campo do conhecimento humano.

Uma das primeiras ações no sentido de formalizar o conhecimento de determinada área como sendo um padrão que pudesse ser reaplicado em vários outros projetos foi desenvolvida por um grupo de arquitetos liderado por Christopher Alexander (ROBERTSON, 2000). No trabalho de Alexander foram catalogados 253 padrões de projeto relacionados com situações comuns no campo da arquitetura.

A adaptação da abordagem de padrões elaborada por Alexander para a área da computação aplicou-se inicialmente à reutilização de código, como na analogia proposta por Peter Coad em (COAD, 1992) onde classes e objetos corresponderiam a elementos básicos, através dos quais seria possível criar “blocos de construção”, que tornariam mais efetivo o trabalho de análise e de projeto de sistema usando padrões.

Martin Fowler apresenta sua definição de padrão de análise como sendo “uma ideia que foi útil em um contexto prático e provavelmente será útil em outros” (FOWLER, 1996). Com este trabalho Fowler tornou-se referência quanto ao uso dos padrões de análise ao apresentar um catálogo com 63 padrões e interligar a aplicação destes com o reuso.

Fowler questionou a influência do trabalho de Alexander na computação (FOWLER, 1996) onde propõe que o livro da “*Gang of Four*” (Padrões de Projeto de Erich Gamma, Richard, Ralph Johnson e John Vlissides) como sendo responsável por uma influência muito maior em padrões de análise do que os trabalhos de Alexander. Fowler justifica sua afirmativa expondo que três dos quatro autores nem mesmo teriam lido os estudos de Alexander e que estes não seriam amplamente aceitos mesmo na área de arquitetura. Por outro lado, autores como Coad (COAD, 1992) e Robertson (ROBERTSON, 2000) citam o trabalho de Alexander como sendo de vital importância para desenvolvimento de padrões em outras áreas do conhecimento bem como na computação.

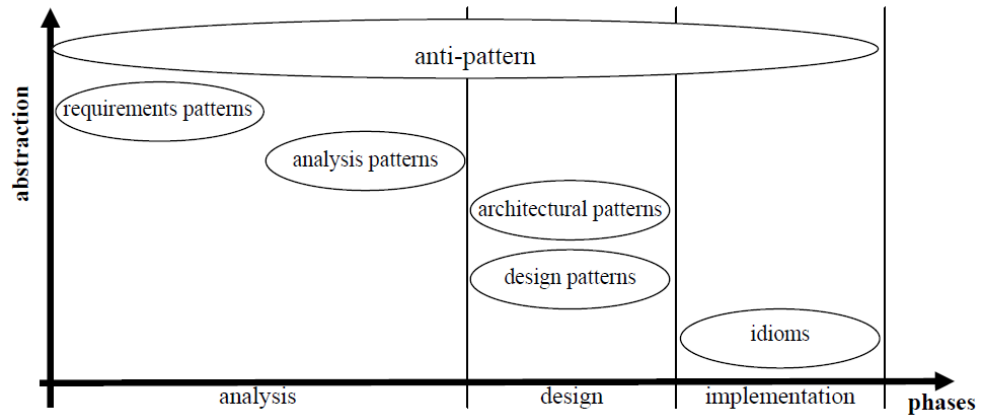
### 2.1.2 O uso de padrões na computação

No ciclo de desenvolvimento de sistemas utilizando o paradigma da orientação a objetos, os padrões são utilizados em suas diversas fases como análise, projeto e implementação. A cada uma destas fases pode-se associar diferentes padrões, que recebem diferentes nomenclaturas como padrões de análise, padrões de projeto, padrões de arquitetura, padrões de código e outros (LISBOA FILHO et al., 2002).

A Figura 1 mostra os diferentes padrões associados às respectivas fases do projeto de sistemas onde são aplicáveis bem como os diferentes níveis de abstração relacionados a cada padrão.

No processo de desenvolvimento de software, como visto na Figura 1, as fases de análise e levantamento de requisitos consistem em obter uma abstração do problema a ser tratado pelo do software. A fase subsequente é o projeto (*design*) que, por sua vez é uma abstração do sistema a ser implementado. Nesta fase são definidos aspectos mais concretos como a plataforma para a qual o sistema será projetado, linguagem de programação, armazenamento, etc. É ainda nesta fase que se trata da construção do modelo do software a ser codificado. Na fase final, o modelo é convertido em um programa executável a partir de uma linguagem de programação.

Em todas as fases do desenvolvimento de software pode-se encontrar ainda os anti-padrões, conforme ilustra a Figura 1, que são contraexemplos que descrevem comportamentos observados com frequência na prática cujos efeitos negativos podem ser evitados. (BLAIMER et al., 2010).

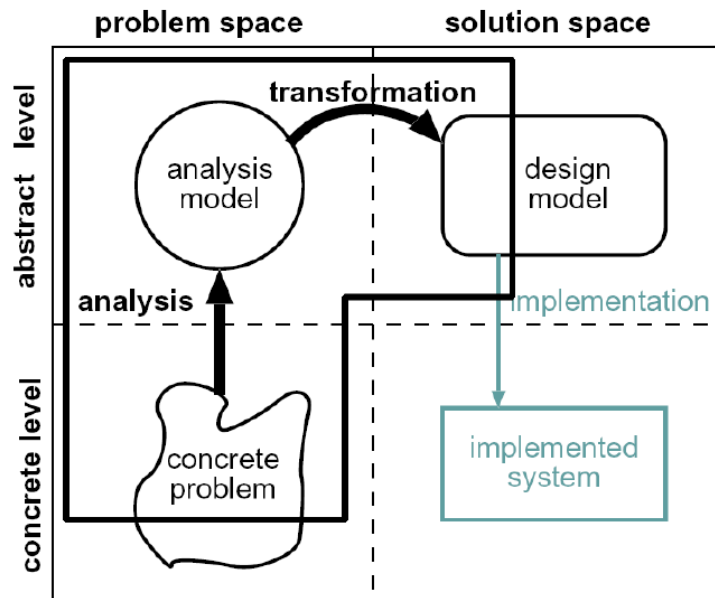


Fonte: (BLAIMER et al., 2010)

Figura 1 - Aplicação dos padrões às fases de desenvolvimento de software

### 2.1.3 Padrões de análise

Padrões de análise têm por finalidade ajudar a entender o problema em si ao invés de apenas ilustrar como são efetivamente modelados (HAMZA, 2002). O escopo de sua aplicação pode ser mostrado através da Figura 2:



Fonte: (BLAIMER et al., 2010)

Figura 2 - Escopo de aplicação de padrão de análise

Os padrões de análise são um mecanismo que permitem o reuso de ideias comprovadamente eficazes no desenvolvimento de softwares, à medida que apresentam soluções anteriormente aplicadas para tratar de problemas recorrentes na engenharia de software (PANTOQUILHO et al., 2003). Eles oferecem como característica mais importante, a estruturação de um conhecimento especializado, possibilitando atingir resultados com maior agilidade, flexibilidade e com menor suscetibilidade a erros no desenvolvimento de sistemas (BLAIMER et al., 2010).

Os padrões de análise têm um grande potencial para redução dos custos de um projeto por contribuir para a redução do número de falhas do software (BLAIMER et al., 2010) uma vez que erros detectados nas fases iniciais do projeto são menos onerosos que aqueles encontrados em suas fases finais (GLASS, 2002). Isto ocorre porque o foco deste padrão está na construção de estruturas conceituais de processos de negócios ao invés de ater-se apenas à implementação do sistema (PANTOQUILHO et al., 2003).

O desenvolvimento de padrões de análise leva em consideração a existência de similaridades entre diversos componentes presentes em projetos com diferentes objetivos finais. A identificação destas similaridades e a formalização de suas principais ideias permite a criação de modelos reaplicáveis em outros contextos. A representação destes modelos foi apresentada em (RIEHLE et al., 1996) como sendo feita em três formas:

- A forma Alexandriana, baseada na forma de catalogação proposta por Alexander, apresenta-se dividida em 3 partes: problema, contexto e solução. Esta forma de representação de padrão tem como intenção guiar o usuário a encontrar soluções para os problemas descritos;
- O catálogo de padrões, que propõe a descrição do padrão através de seções, cada qual descrevendo uma característica do padrão. O foco desta forma de representação está mais na descrição do padrão em si e menos na situação em que se deve usá-lo;
- A forma geral que sugere a descrição do padrão em *Contexto* e *Padrão*. A Seção *Contexto* descreve restrições, o contexto e forças que deram origem ao padrão. A Seção *Padrão* trata da maneira de se aplicar o padrão em dado contexto. O principal objetivo desta forma de

apresentação é discutir a estrutura e a dinâmica recorrentes sem se preocupar em especificar uma forma de aplicação específica do padrão.

O formato de catálogo de padrões é o mais apropriado para a documentação de padrões voltados para projetos de software usando o paradigma de programação orientada a objetos por permitir uma boa descrição do projeto em si. Porém, características como a descrição da interação padrão/contexto preconizada pela forma geral ou a descrição de como determinado problema pode ser resolvido pela aplicação de um padrão, prevista pela forma Alexandriana, são igualmente importantes para a aplicação adequada dos padrões.

## **2.2.APRI**

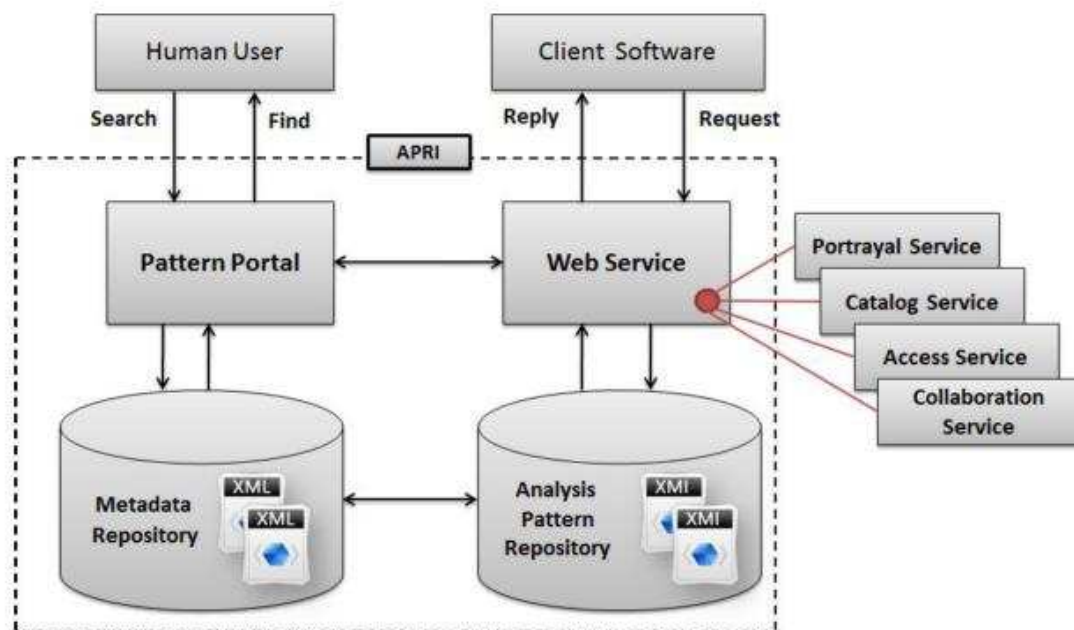
Apesar dos ganhos com a qualidade do software e redução de custos evitando erros futuros, os padrões de análise parecem ser muito menos adotados pela comunidade de desenvolvedores que padrões de projeto, quando comparadas apenas as quantidades de publicações sobre respectivos assuntos (BLAIMER et al., 2010). Uma das razões que explicaria este baixo interesse por padrões de análise está em sua documentação inadequada. Pouca atenção parece ser dada à forma de representação dos padrões de análise (RIEHLE et al., 1996) e diversos autores apresentam suas próprias soluções para representá-los, por exemplo, com descrições textuais, diagramas de entidade relacionamento, diagrama de fluxo de dados, entre outros (BLAIMER et al., 2010). Este problema impacta negativamente nos resultados que se espera obter com o uso dos padrões de análise: reusabilidade e efetividade (HAMZA; FAYAD, 2002), uma vez que dificulta a sua disseminação para a comunidade de usuários o que, por sua vez, não contribui para a evolução dos atuais padrões de análise existentes e nem para a descoberta de novos padrões.

Com o objetivo de tornar os padrões de análise disponíveis a seus potenciais usuários de forma padronizada, legível não apenas por humanos mas também por máquinas, o grupo de Sistemas de Informação da Universidade Federal de Viçosa propôs a criação de uma Infraestrutura de Reuso de Padrões de Análise – (*Analysis Patterns Reuse Infrastructure*) – APRI, na sigla em inglês (VEGI; PEIXOTO; et al.,

2012). A ideia da APRI deriva de soluções semelhantes aplicadas em outras áreas de estudo como a de Sistemas de Informação Geográfica denominadas *Spatial Data Infrastructure* (SDI).

O repositório de padrões proposto como sendo um dos componentes da APRI tem por objetivo permitir que os usuários consigam definir qual padrão melhor se adequa ao projeto que pretendem desenvolver, a partir da busca por padrões que atendam às características do projeto. Além disso, favorece o formato de catálogo de padrões um aspecto importante para a contribuição na criação e disseminação de uma cultura do uso de padrões de análise: como possibilita o armazenamento da descrição do padrão de maneira estruturada, este formato facilita a descoberta e compartilhamento de informações a seu respeito usando meios como a Internet. Este papel de comunicação é destacado por Blaimer (BLAIMER et al., 2010) como fator importante no reuso dos padrões: “não posso usar o que não encontrei ou aplicar o que encontrei, mas não compreendo”.

Baseado nos trabalhos de Pantoquilha (PANTOQUILHO et al., 2003) e de Béjar (BÉJAR et al., 2009) que propõem, respectivamente, um *template* que possibilita a captura de elementos importantes na documentação de um padrão de análise e um estilo arquitetural para SDI que fornece ferramentas e vocabulários para auxiliar arquitetos de sistemas a projetar estas infraestruturas, (VEGI; PEIXOTO; et al., 2012) propõem uma infraestrutura de padrões de análise inspirado em soluções aplicadas na área de SIG. A infraestrutura denominada APRI (*Analysis Patterns Reuse Infrastructure*), conforme mostrada na Figura 3, preconiza que os padrões de análise devem ser documentados por meio de metadados e podem ser recuperados via *Web Services*.



Fonte: (VEGI; PEIXOTO; et al., 2012)

Figura 3 - Arquitetura da APRI

A documentação dos padrões de análise é feita a partir da adaptação do *template* de Pantoquilha ao padrão *Dublin Core*. A partir desta adaptação foi gerado um perfil de documentação de padrões de análise chamado DC2AP (*Dublin Core Application Profile to Analysis Patterns*). Um perfil descreve como especificações de padrão podem ser implantadas para dar suporte a requisitos de uma comunidade ou aplicação em particular. Na comunidade de metadados o termo perfil de aplicação é utilizado para descrever a geração de padrões para aplicações específicas (NILSSON et al., 2008).

Com o objetivo de fazer com que os padrões de análise descritos na APRI sejam utilizados não apenas por usuários humanos, mas também por outros softwares ou serviços, cada elemento proposto pelo DC2AP foi associado com um identificador *Universal Resource Identifier* - URI e descrito semanticamente através de *Resource Description Framework* – RDF (VEGI et al., 2013). Ao utilizar esta metodologia para descrever os padrões de análise, a APRI disponibiliza seus dados utilizando o conceito de *Linked Data*.

O conceito de *Linked Data* contrapõe-se à forma de disponibilização de informações da Web utilizada nos dias de hoje ao propor que os recursos e dados

contenham também significado próprio e possam ser, desta forma, interpretados por outros recursos. Tecnicamente é a disponibilização de dados na Internet de forma tal que seu significado seja explicitamente definido num formato legível por máquina e interligado a conjuntos de dados externos (BIZER et al., 2009).

Enquanto na Web atual os documentos são descritos através de HTML, onde as páginas são interligadas através de links, na *Linked Data* as informações são descritas através de RDF que fornece um modelo de dados genérico baseado em grafos para estruturar e interligar dados que descrevem informações do mundo real. Estes dados são endereçados através de URIs que fornecem uma forma também genérica para identificá-los. (BIZER et al., 2009).

### **2.3. Padrões de Metadados**

Existem diferentes formatos de documentação de padrões de análise, muitos deles baseados nos modelos utilizados para a catalogação de padrões de projeto, moldados de acordo com as necessidades e estilos de seus autores (RAMINHOS et al., 2006). A grande variedade de formatos de documentação de padrões de análise não contribui necessariamente para a disseminação de seu uso por estarem disponíveis muitas vezes em formatos legíveis apenas por humanos como artigos científicos e livros. Este fato faz com que padrões assim documentados sejam pouco suscetíveis à detecção por meios eletrônicos como máquinas de busca, o que dificulta sua reutilização.

Em diversas áreas do conhecimento humano existe uma situação semelhante que é contornada pela aplicação de metadados como, por exemplo, a de (SIG). Os metadados são usados para documentar os recursos visando não apenas a documentação em si, mas também a possibilidade de se utilizar estes recursos em diversos ambientes e com outros propósitos que não sejam apenas aqueles para os quais foram inicialmente gerados.

Metadados são dados sobre dados e fornecem meios de documentação de informações sobre diferentes recursos (um padrão de análise, um item de um mapa, o

próprio mapa, etc.). Tal documentação possibilita a descrição de conteúdo, qualidade, autoria entre outros aspectos do recurso. Esta caracterização do recurso que está sendo documentado constitui o mecanismo que possibilita usuários e aplicações utilizarem e compartilharem tais recursos ou serviços (NOGUERAS-ISO et al., 2005).

O prefixo grego *meta* significa algo que transcende ao que é natural ou algo que vai além do convencional. A palavra *dados* deriva do Latim, do verbo “dar”, e indica algo que se deu sobre o qual se pode deduzir fatos. Dados, no contexto da área de banco de dados pode ser definido como uma informação acerca de algo que tenha um significado intrínseco (ELMASRI & NAVATHE, 2011). A ideia por trás de metadados pode ser tratada como sendo a descrição sistematizada de dados de forma estruturada. Esta descrição é feita por meio da definição de um conjunto de elementos descritivos capazes de representar informações acerca de determinado recurso, como: título, assunto, autor, formato, etc. O fato de as informações acerca dos dados serem estruturadas torna possível, entre outras situações, a disponibilização *online* da descrição de recursos de forma legível por máquinas. Por recursos pode-se entender desde a descrição de um livro, objetos pessoais e músicas até a forma como estes recursos tornam-se disponíveis na rede (RUHLE et al., 2011).

A estruturação dos metadados, conhecido como esquemas ou formatos de metadados, tem como objetivo permitir a descrição de diferentes informações em diferentes contextos (SAYÃO, 2010). Alguns exemplos da diversidade de esquemas de metadados:

- MARC 21 (*Machine-readable cataloguing*): esquema de metadados voltados para a área de catalogação bibliográfica;
- MODS (*Metadata Object Description Schema*): esquema derivado do MARC 21 mas que pode atender a propósitos gerais;
- EAD (*Encoded Archival Description*): voltado para a área de Arquivologia;
- LOM (*Learning Object Metadata*): para gerenciar, avaliar e localizar objetos de aprendizagem.

Estes esquemas de metadados desempenham um importante papel em seus respectivos contextos de aplicação, possibilitando resultados positivos no que diz respeito a interoperabilidade e comunicação das informações por eles abordadas. Foi com base neste sucesso em áreas distintas que surgiu a necessidade de se alcançar resultados semelhantes, porém de forma mais ampla, permitindo a documentação de elementos de contextos diversos usando-se um mesmo esquema. Esta iniciativa deu origem ao esquema, ou padrão Dublin Core.

O Dublin Core foi proposto em meados da década de 90 a partir das ideias de um corpo de pesquisadores de diferentes áreas como biblioteconomia, cientistas da informação, museólogos e outros estudiosos da área de gestão do conhecimento, reunidos na cidade de Dublin, no estado americano de Ohio. O foco principal do encontro foi a obtenção do consenso sobre os elementos básicos (o *Core*) que fossem capazes de descrever recursos de diferentes áreas do conhecimento humano, além da proposição de padrões e protocolos para facilitar a descrição, organização, descoberta e acesso a uma rede de informações sobre recursos (DCMI, 1995). A continuação do desenvolvimento e suas especificações correlatas ficam a cargo da DCMI – *Dublin Core Metadata Initiative*.

Diferente dos outros padrões de metadados existentes, o Dublin Core utiliza uma abordagem alternativa uma vez que seu objetivo não é voltado para a descrição de recursos de um domínio específico. O seu conjunto de descritores de dados básicos, formado por apenas 15 elementos, e sua estrutura flexível o tornam adequado para a descrição e descoberta de diferentes tipos de recursos em níveis de complexidade variados (NEBERT, 2008). O Dublin Core utiliza a estratégia de criação de perfis de aplicação para criar padrões de metadados para fins específicos, acrescentando outros elementos aos 15 originais ou mesmo retirando alguns destes elementos básicos quando não contribuem diretamente para a documentação do recurso descrito.

Constantes discussões entre duas vertentes de usuários do Dublin Core, os minimalistas e os estruturantes levaram a distinção os elementos do padrão em Simple e Qualificado. Os minimalistas defendem a manutenção fiel ao *Core* de elementos reduzido e simplificado, enquanto os estruturantes, que apostam em adicionar características que propiciem uma descrição semântica mais refinada (NISO.ORG, 2004). O padrão Dublin Core Simple aborda os 15 elementos originais e o Qualificado

inclui três novos elementos (*Audience*, *Provenance* e *RightsHolder*) e 4 qualificadores (*InstructionalMethod*, *AccuralMethod*, *AccuralPeriodicity* e *AccuralPolicy*) com o objetivo de refinar as qualidades semânticas da descrição do recurso, contribuindo para a descoberta do recurso (HILLMANN, 2011).

As tabelas a 1 e 2 descrevem, respectivamente os elementos simples e qualificados fornecendo exemplos de como podem ser preenchidos. Alguns dos exemplos de preenchimento do elemento fornecidos podem ter mais de um valor. Fatores como a ordem de apresentação dos elementos, sua obrigatoriedade e repetição são previstos na estruturação do Dublin Core (NISO.ORG, 2004).

*Tabela 1 - Elementos simples do Dublin Core*

<b>Elemento</b>	<b>Descrição</b>	<b>Exemplo</b>
<i>Title</i>	Guarda o nome dado ao recurso	Title= "Analysis patterns: reusable object models"
<i>Subject</i>	Expressa o conteúdo do assunto tratado por meio de palavras chave ou códigos de classificação que retratem o recurso.	Subject = "Analysis Patterns"
<i>Description</i>	Explica o conteúdo do assunto através de um resumo, sumário, referências gráficas, entre outros dados que auxiliem na compreensão do recurso disponibilizado.	Description = "Analysis patterns are groups of concepts that represent a common construction in business modeling. It may be relevant to only one domain, or it may span many domains."
<i>Type</i>	Define o tipo de informação que está sendo disponibilizado. Recomenda-se utilizar vocabulários controlados que permitam denotar claramente informações como funções, gêneros, ou níveis de agregação do conteúdo.	Type = "Physical Object" Type = "Book"
<i>Source</i>	Uma referência a fonte da qual o recurso se origina	Source = "RC607.A26W574 1996"
<i>Relation</i>	Referências ao recurso que está sendo descrito. Estas referências podem ser descritas de forma unidirecional ou bidirecional. Para uma descrição das referências com maior acurácia é aconselhável o uso de elementos qualificados do Dublin Core.	Title = "Analysis patterns: reusable object models" Relation: "An Infrastructure Oriented for Cataloging Services and Reuse of Analysis Patterns" O relacionamento utilizado é: IsReferencedBy

<i>Coverage</i>	A extensão do conteúdo do recurso. Este elemento situa o recurso no tempo e/ou espaço.	Coverage = “1996” Coverage = “Indianapolis”
<i>Creator</i>	Identifica a pessoa ou entidade responsável pela criação do conteúdo.	Creator = “Fowler, Martin”
<i>Publisher</i>	Identifica a pessoa ou entidade responsável pela disponibilização do conteúdo.	Publisher = “Costa, Glauber”
<i>Contributors</i>	Identifica a pessoa ou entidade responsável por contribuir com o conteúdo que está sendo disponibilizado	Contributor = “Lisboa Filho, Jugurta”
<i>Rights</i>	Indica as informações legais acerca do acesso ao conteúdo.	Rights = “Acesso limitado a associados”
<i>Date</i>	Indica uma data associada a data de criação ou a determinado evento ligado ao ciclo de vida do elemento	Date = “1996” Date = “1996-03” Date = “1996-03-28”
<i>Format</i>	Indica a manifestação física do recurso. Pode incluir o meio pelo qual o recurso é disponibilizado ou suas dimensões.	Type = “Physical Object” Format = “Brochure”
<i>Identifier</i>	Uma referência formal, única em relação ao conteúdo.	Identifier = “ISBN-10: 0201895420”
<i>Language</i>	Linguagem na qual foi descrito o conteúdo intelectual do recurso	Language = “en-US”

*Tabela 2 - Elementos Qualificados do Dublin Core*

Elemento	Descrição	Exemplo
<i>Audience</i>	Indica entidades às quais o recurso é direcionado ou a quem pode ser útil	Audience = “Software Developers”
<i>Provenance</i>	Indica mudanças de custódia ou propriedade do recurso. Pode ser utilizado para indicar sucessivas mudanças relativas a propriedade do recurso	Provenance = “This copy is owned by Glauber Costa”
<i>RightsHolder</i>	Indica a pessoa ou organização que detém os direitos sobre o recurso.	RightsHolder = “Martin Fowler”

A partir do ano 2000, a comunidade Dublin Core voltou seus esforços para a definição de perfis de aplicação cuja meta era que registros de metadados poderiam ser utilizados juntamente com outros vocabulários especializados para ir ao encontro a requisitos particulares de implementação. Nesta mesma época ocorreu a consolidação de um trabalho da W3C em um modelo genérico de metadados, o Framework de Descrição de Recursos (RDF – *Resource Description Framework*). Como parte de um conjunto estendido de termos de metadados do DCMI, o *Dublin Core* se tornou um dos vocabulários mais populares para uso do RDF.

A consolidação do RDF motivou um esforço de traduzir o estilo de metadados da comunidade *Dublin Core* em um modelo abstrato DCMI compatível com o RDF. Este modelo foi projetado para interligar o paradigma de *Linked Data* com um paradigma mais familiar de registros de metadados validáveis como os utilizados no OAI-PMH – *Open Archives Initiative/Protocol for Metadata Harvesting*. O OAI-PMH é um protocolo composto de 6 serviços que podem ser invocados via HTTP que visa incrementar a interoperabilidade no intercâmbio de recursos documentados via metadados.

A especificação do Perfil Conjunto de Descrição (DSP – *Description Set Profile*) define uma linguagem para expressar as restrições de forma genérica, independente de aplicação. O DSP permite a expressão de restrições de um perfil de aplicação Dublin Core. O Framework Singapura para perfis de aplicação Dublin Core define o conjunto de componentes úteis para documentação de um perfil de aplicação com o máximo de reutilização.

A simplicidade dos elementos que compõem o padrão de metadados Dublin Core aliado à capacidade de descrever estes elementos através de RDF e a definição de restrições estruturais destes elementos fornece um arcabouço importante para a descrição semântica de recursos catalográficos de uso geral. O conjunto destas ferramentas possibilita o surgimento de ambientes favoráveis à disseminação de informações de âmbito irrestrito.

## 2.4.Repositórios de dados

Os repositórios de dados têm sido reconhecidos como ferramentas críticas para diversas áreas do conhecimento humano como ciência, tecnologia, economia, entre outras, por fornecer meios para armazenar informações e compartilhá-las com a comunidade de interessados em determinado assunto. O mapeamento do genoma humano é um exemplo de como o compartilhamento de informações vem ganhando adeptos no meio acadêmico através da disponibilização da troca de dados (MARCIAL; HEMMINGER, 2010).

Outro exemplo da aplicação dos repositórios de dados se dá na área de informação geográfica. As SDI fornecem um arcabouço para a criação, manutenção e distribuição de dados espaciais em diferentes níveis de organização.(NOGUERAS-ISO et al., 2005)

Uma iniciativa pioneira no sentido da criação de um ambiente que propicia a criação e compartilhamento de informações científicas ocorreu no início da década de 90, no Laboratório Nacional de Los Alamos, nos Estados Unidos, onde se construiu e implantou o repositório digital *arXiv*<sup>1</sup> (OLIVEIRA; CARVALHO, 2011). Este repositório tem como objetivo agilizar o acesso a resultados de pesquisa na forma de artigos ou de relatórios técnicos.

Outros exemplos de repositórios semelhantes surgiram e sua diversidade e o crescimento de sua disponibilidade gerou a necessidade da criação de organismos como a *Open Archives Initiative* (OAI) que contribuiu para o estabelecimento de normas técnicas e organizacionais de estruturas de publicação aberta. Esta iniciativa originou o modelo chamado *Open Archives* cujo principal objetivo reside em propiciar alto nível de interoperabilidade entre os repositórios. Esta interoperabilidade ocorre através do uso de dois pilares principais: o padrão de metadados Dublin Core e o protocolo de coleta de metadados, denominado OAI-PMH que fornece uma interface que possibilita o acesso às informações. O protocolo OAI-PMH é uma interface de acesso aos metadados

---

<sup>1</sup> Leia-se “archive”, como se a letra X representasse a letra grega *Chi*

fornecendo características como interoperabilidade e extensibilidade, obtidas através do uso do padrão de metadados Dublin Core (OLIVEIRA; CARVALHO, 2011).

## 2.5.Serviços Web

A computação orientada a serviços (*Service Oriented Computing - SOC*) é um paradigma de desenvolvimento cujos conceitos, princípios e métodos na qual a computação é realizada por meio de uma arquitetura orientada a serviços (*Service-oriented Architecture – SOA*). Nessa arquitetura as aplicações são construídas baseadas em componentes de serviços independentes com interfaces padronizadas (TSAI et al., 2006).

Os conceitos que compõem o arcabouço da computação orientada a serviços fornecem meios que possibilitam a comunicação entre diferentes aplicações de software, independente da linguagem em que são desenvolvidas, rodando em diversas plataformas ou *frameworks* através do uso de serviços como elementos fundamentais para o desenvolvimento de aplicações (PAPAZOGLU; GEORGAKOPOULOS, 2003). Desta forma torna-se possível separar explicitamente a engenharia de software da programação propriamente dita, enfatizando o processo de desenvolvimento mais na primeira em detrimento da segunda (TSAI et al., 2006).

O paradigma de Serviços Web oferece meios que possibilitam a interoperabilidade de sistemas heterogêneos acrescentando-lhes flexibilidade e dinamicidade através de aplicações baseadas na Internet (HULL, 2005). Serviços Web são entidades de software autônomas, fracamente acopladas, auto descritivas, independentes de plataforma cujas interfaces e ligações são passíveis de serem definidas, descritas e descobertas como artefatos (FERRIS; FARRELL, 2003). São capazes de efetuar funções que vão desde a resposta a uma requisição simples, como verificar a validade de um determinado documento ou buscar o endereço através do CEP, até a construção de sistemas sofisticados que envolvem vários processos de negócios entre empresas diferentes usando múltiplas camadas de consumidores e usuários de serviços.

Por utilizar a Web como meio de intercâmbio de mensagens, os Serviços Web transformam as fontes de informação em componentes reutilizáveis para a construção de aplicações mais complexas. A forma de realizar esta troca de mensagens, de maneira que diferentes participantes possam compreendê-las e utilizá-las, é feita através de padrões baseados na Internet, que incluem o *Simple Object Access Protocol* (SOAP) – Protocolo de Acesso a Objetos Simples – responsável pela transmissão das informações, a *Web Services Description Language* (WSDL) – Linguagem de Descrição de Serviços Web – usada para a definição dos serviços e a *Business Process Execution Language for Web Services* (BPEL4WS) para a orquestração dos serviços, ou seja, a combinação de dois ou mais serviços com o objetivo de criar outros serviços Web mais sofisticados (PAPAZOGLU; DUBRAY, 2004).

### **3. ANÁLISE DE CANDIDATOS A COMPONENTES DO REPOSITÓRIO DE PADRÕES DE ANÁLISE**

A reutilização de artefatos computacionais nas fases do desenvolvimento de sistemas computacionais contribuem positivamente para a obtenção de um software de qualidade. Os padrões de análise são um exemplo deste fato, uma vez que sua utilização ocasiona a redução de erros de projeto por fazer uso de soluções comprovadamente eficazes em situações análogas, entre outros benefícios.

O desafio a ser contornado é a oferta de uma grande quantidade de formas de documentação destes padrões, o que gera uma dificuldade de fazer com que sejam expostos ao seu público alvo e aplicados de maneira mais consistente. Visando superar este desafio, a APRI propõe uma abordagem do uso de uma infraestrutura que possibilite a documentação, armazenamento, intercâmbio e disseminação dos padrões de análise por meio de um repositório de padrões de análise, documentados a partir de um perfil de metadados, gerenciáveis e acessíveis através de serviços Web.

As experiências na superação de desafios semelhantes em áreas de estudo como Sistemas de Informação Geográfica e Biblioteconomia serviram como ponto de partida para a realização deste trabalho. Nestas áreas é constante a aplicação de infraestruturas computacionais que viabilizem o intercâmbio de objetos digitais, tendo repositórios de dados como um de seus componentes chave.

Este capítulo descreve os componentes necessários para a realização da APRI. Na subseção 3.1 é descrito um modelo de sistemas de compartilhamento de dados abertos, seus conceitos e modelos de intercâmbio de informação. A subseção 3.2 analisa algumas soluções tecnológicas existentes, sob a ótica dos quesitos que estas soluções devem atender para viabilizar o portal de padrão de análise. Ainda nessa subseção justifica-se a escolha do Fedora como componente da APRI que servirá como disseminador deste artefato computacional para a comunidade interessada na área. A subseção 3.3 faz uma conclusão deste capítulo.

### **3.1. O Modelo Open Archive Information Systems (OAIS)**

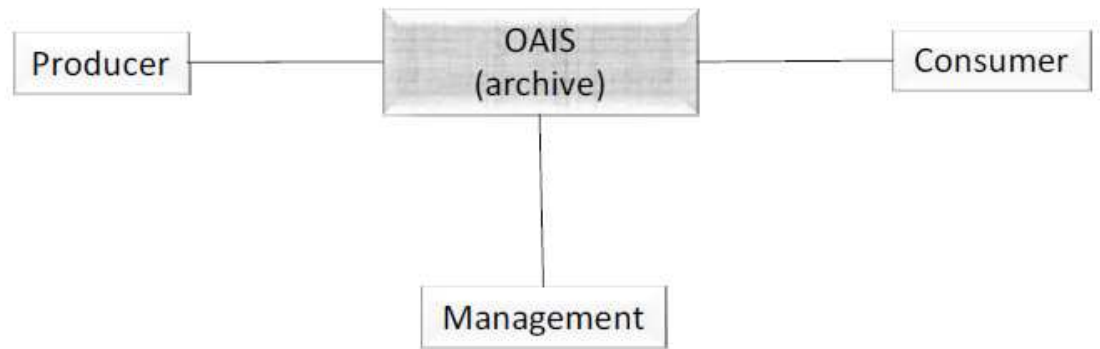
Em sua essência, um repositório de dados é um ambiente computacional, em geral um site Web, capaz de armazenar dados e metadados sobre determinado assunto, fornecendo à comunidade de interessados recursos que lhe permitam descrever e gerenciar o objeto nele armazenado, preservar as informações deste objeto por um período indeterminado e torna-las acessíveis a esta comunidade. O padrão de referência do CCSDS (*The Consultative Committee for Space Data Systems*) define estas características em seu modelo de referência para um sistema de informação de arquivos abertos (CCSDS, 2012).

O principal objetivo deste modelo é dar conhecimento dos requisitos necessários para a preservação e disponibilização de objetos de informação com os mais variados fins. Entenda-se como objeto de informação os objetos de dados em conjunto com sua representação de informação (CCSDS, 2012).

O conhecimento do modelo OAIS, no contexto deste trabalho, fornece as informações necessárias para a especificação de uma arquitetura que permita disponibilizar os componentes de base tecnológica previstos na APRI. Além disso, possibilita uma análise de algumas soluções para construção de um ambiente de preservação e disponibilização de padrões de análise baseada em parâmetros fornecidos por uma Prática Recomendada pela CSDS denominada CSDS 650.0-M-2.

#### **3.1.1 O ambiente**

O ambiente no qual se insere um OAIS é ilustrado através da *Figura 4*.



Fonte: (CCSDS, 2012)

Figura 4 - Modelo de ambiente de um OAIS

A interação com o ambiente do OAIS é efetuada por três participantes básicos:

- O Produtor: papel executado por um usuário ou um outro OAIS cuja responsabilidade seja a de fornecer a informação a ser preservada;
- O Gestor: membro de uma organização que cumpre um papel de responsável por coordenar as políticas internas de um OAIS, que fazem parte de um conjunto mais amplo de políticas que regem o domínio da informação como um todo;
- O Consumidor: pessoa ou sistema cliente integrante da comunidade de interessados que interage com o OAIS com a finalidade de adquirir informação de seu interesse.

Os papéis de Consumidor e Produtor podem ser efetuados por outros OAIS. Um Produtor pode, por exemplo, ser responsável por um determinado objeto de informação que é necessário para compor as informações de um terceiro OAIS. Um Consumidor pode utilizar de um objeto de informação sob a responsabilidade de outro OAIS. A escolha entre coletar toda a informação necessária à representação de uma informação ou delegar parte dela a outro OAIS cabe a cada implementação do ambiente.

### 3.1.2 A informação

A informação pode ser definida como qualquer tipo de conhecimento compartilhável que é expresso a partir de uma informação estruturada durante o

compartilhamento (CCSDS, 2012). Para se interpretar esta informação é necessário que o consumidor (usuário humano ou software cliente de serviços) tenha um conhecimento base para interpretar o dado ou que lhe seja fornecido a representação da informação afim de transforma-lo em informação significativa.

Conhecimento base é a capacidade de interpretar e inferir valor à informação recebida pelo usuário e a representação da informação são as regras necessárias para compreender a informação que está sendo representada. Assim, um artigo escrito no idioma Português é representado pelas letras dispostas de forma a compor palavras pertencentes ao dicionário deste idioma. Estas palavras devem obedecer a um conjunto de regras gramaticais. Para que conhecimento apresentado pelo artigo seja compreendido adequadamente, o leitor deve ter como conhecimento base a capacidade de compreender o idioma Português, ou seja, deve conhecer o dicionário e a gramática deste idioma.

A definição da comunidade de usuários interessados nas informações disponibilizadas pelo OAIS exerce um papel fundamental no processo de definição do ambiente. Se o nível de conhecimento base dos usuários for menos abrangente, o detalhamento de informação da representação deverá ser maior, para que seja possível descrever o dado num nível de abstração compatível com a base de conhecimento do usuário. O contrário também pode ser aplicado, ou seja, numa comunidade de usuários com base de conhecimento elevada exige um nível de detalhamento de representação da informação menos abrangente. Em geral pode ser dito que o dado interpretado usando sua informação da representação gera conhecimento, como ilustra a *Figura 5*.



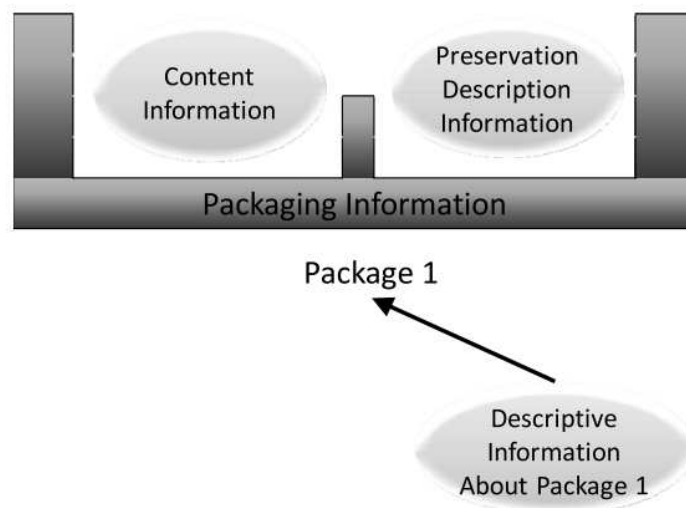
*Fonte:* (CCSDS, 2012)

*Figura 5 - Obtenção de informação significativa de dados*

Para que um objeto de informação seja preservado adequadamente é de fundamental importância para um OAIS identificar claramente a diferença entre o

objeto de dados e sua representação informacional. Entretanto, cada submissão de informação efetuada por um Produtor ou cada informação disseminada por um Consumidor ocorre como uma transmissão discreta onde o dado e sua representação são enviados. Desta forma é conveniente a definição de um conceito de pacote de informação (CCSDS, 2012).

O pacote de informação é um container conceitual composto de dois tipos de dados, chamados de informação de conteúdo e descrição da informação de preservação (PDI – *Preservation Description Information*). Semanticamente a relação entre os componentes do pacote podem ser interpretados de acordo com o representado na *Figura 6*.



Fonte: (CCSDS, 2012)

*Figura 6 - Relacionamento entre os componentes do Pacote de Informação*

A informação de conteúdo são os dados sobre o objeto da informação e a sua descrição, enquanto que o PDI é utilizado para contextualizar e compreender a forma como o objeto de dados foi adquirido, a quem pertence originalmente (caso esta informação seja pertinente), como se relaciona com outros objetos, como referenciá-lo globalmente e seu formato original.

### 3.1.3 Aspectos estruturais e de interoperabilidade

A especificação do formato do dado e sua descrição são fatores primordiais dentro de um OAIS, porém outros quesitos contribuem fortemente para assegurar que outras características também importantes estejam presentes em sistemas que visam garantir a durabilidade e usabilidade da informação. Assim é preciso garantir ainda que a estruturação da informação forneça meios para inter-relacionar os dados, para lhes inculcir significado semântico e para possibilitar o intercâmbio desses dados intra-OAIS e extra-OAIS. Estes são os objetivos dos modelos funcional e modelo informacional.

O modelo funcional descreve de forma detalhada as relações previstas na seção 3.1.1 que trata sobre o ambiente dos OAIS, acrescentando as entidades funcionais e suas interfaces de relacionamento conforme ilustra a *Figura 7*.

A *Figura 7* mostra ainda os pacotes de informação relativas a cada tipo de usuário, denotado pelas siglas SIP (*Submission Information Package*), AIP (*Archival Information Package*) e DIP (*Dissemination Information Package*).

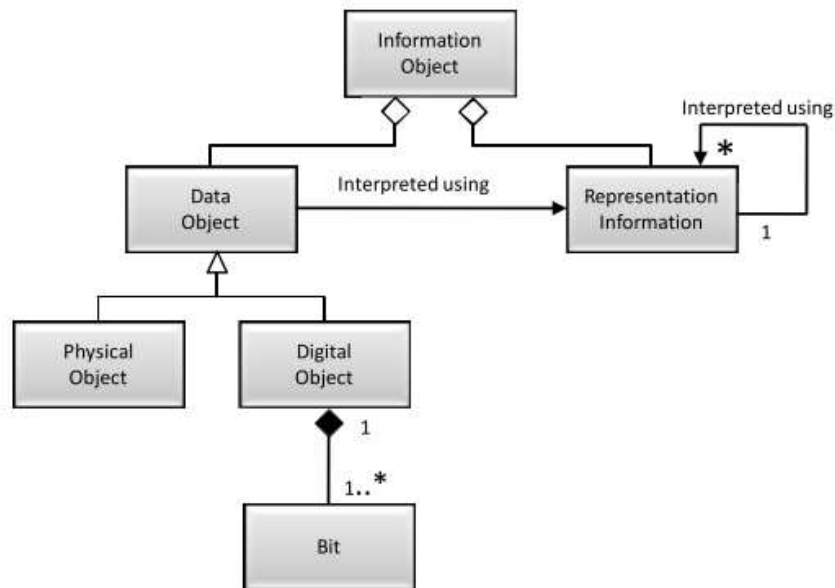


Fonte: (CCSDS, 2012)

*Figura 7 - Entidades Funcionais dos OAIS*

O modelo informacional ajuda a compreender os componentes de dados a serem armazenados e gerenciados pelos OAIS. Este modelo abrange o modelo de dados a ser

seguido para obter a preservação da informação e os objetos e *containers* conceituais para a representação do conteúdo, de acordo com a *Figura 8*.



Fonte: (CCSDS, 2012)

*Figura 8 - Modelo UML de um objeto de informação*

O diagrama UML mostrado na Figura 8 ilustra o conceito básico do modelo de referência OAIS onde a informação é um conjunto formado pelo dado em si e pela representação da informação. O objeto de informação é formado pela composição de um objeto de dados, físico ou digital e da representação da informação. A representação da informação possui um relacionamento recursivo, indicando que pode ser composta por outras representações de informação, sendo a representação da informação responsável por definir estruturas de informação ou a semântica de um objeto.

### **3.2. Ambientes de desenvolvimento de repositórios de dados**

O cenário mundial oferece muitas soluções consolidadas, no que diz respeito a ambientes computacionais que atendam às práticas recomendadas, descritas na seção anterior, pela CSDS 650.0-M-2. O nível de aderência às especificações estipuladas neste documento varia de solução para solução bem como o nível de flexibilidade na criação dos modelos de dados a serem tratados por cada uma delas. Nesta seção são

apresentados alguns ambientes que propiciam a implementação da arquitetura proposta para a APRI e define-se a mais adequada segundo os objetivos globais da infraestrutura.

O *Open Society Institute* elaborou um estudo (CROW, 2004) onde é elencado um grupo de sistemas computacionais utilizados na criação de repositórios abertos. A escolha dos sistemas analisados se baseou em 3 critérios fundamentais:

1. Os sistemas devem estar disponíveis sob licença de *Open Source*;
2. Os sistemas devem ser compatíveis com o protocolo OAI-MPH;
3. Os sistemas devem ter sido usados na implementação de repositórios abertos disponíveis ao público.

Os 3 critérios mencionados na pesquisa são compatíveis com a necessidade existente para a implementação do repositório de padrões de análise proposto na APRI. O fato de os sistemas estarem disponíveis sob a GPL (*General Public License*) contribui, além de outros fatores, para que outras pesquisas possam ser realizadas utilizando o ambiente, podendo alterá-lo e distribuí-lo sem custos referentes à licença de software. A compatibilidade com o protocolo OAI-MPH permite o intercâmbio de dados com outras infraestruturas com o mesmo fim garantindo assim a interoperabilidade com outros ambientes e com a disseminação das informações sobre os padrões de análise. O terceiro critério que trata do uso do ambiente na implementação de outros repositórios contribui positivamente ao mostrar a utilização dos mecanismos de implantação de repositório de dados em situações reais que possam servir de fonte de pesquisa para a solução em implementação.

Dos sistemas analisados na pesquisa – Archimede, ARNO, CDSware, DSpace, Eprints, Fedora, i- Tor, MyCoRe, e OPUS – nem todos se mostraram como alternativas viáveis ao trabalho em questão por terem seu foco voltado especificamente para ambientes de documentos eletrônicos, fato que seria um limitador importante na criação do ambiente. Por este motivo as plataformas Archimede, Eprints, CDSware, i-Tor, MyCoRe e OPUS foram desconsideradas no restante da análise.

A plataforma ARNO, um projeto sob responsabilidade de instituições holandesas como a universidade de Amsterdam, a universidade Tilburg e a universidade de Twente possui características importantes para a implantação do repositório de padrão de análise como, por exemplo, a grande flexibilidade no que diz respeito aos

objetivos dos repositórios e a adequação aos protocolos da OAI. Porém alguns fatores fizeram com que seu uso fosse considerado pouco adequado ao projeto, a saber:

- O baixo número de repositórios registrados usando a plataforma, segundo entidades especializadas em registros de repositórios de acesso aberto (<http://roar.eprints.org/> e <http://www.opendoar.org/>);
- O fato de todos os repositórios criados serem voltados para instituições de pesquisa holandesas;
- O fato de todos eles serem focados na área de documentação de recursos bibliográficos.

As duas plataformas restantes, Fedora e DSpace, são compatíveis com o *framework* proposto no modelo de referência para um sistema de informações de arquivos abertos – OAIS. Ambas as plataformas têm seu desenvolvimento sob a responsabilidade da Duraspace.org, uma instituição sem fins lucrativos voltada para o desenvolvimento de tecnologias direcionadas à preservação de acervos digitais.

Tanto o DSpace quanto o Fedora atendem ao quesito de serem compatíveis aos conceitos propostos pela CSDS 650.0-M-2. O Dspace é fortemente comprometido com os conceitos definidos para a implementação de OAIS e, sempre que possível, adota o modelo e os vocabulários articulados no modelo OAIS para seu projeto e terminologias utilizadas no ambiente (BASS et al., 2002). O Fedora, ao utilizar um repositório com funcionalidades orientadas a serviços e modelo de armazenamento digital, viabiliza o desenvolvimento de ambientes que podem coexistir confortavelmente com o modelo proposto para os OAIS (LAGOZE et al., 2005).

Um estudo realizado pela FAO (*Food and Agriculture Organization*), organismo da Organização das Nações Unidas responsável por políticas mundiais relacionadas a alimentação, foi utilizado como fonte de informação importante para a tomada de decisão com relação ao software a ser utilizado. Neste estudo foram analisados 32 quesitos dos quais 20 eram critérios que deviam ser atendidos obrigatoriamente. Dos 32 quesitos gerais o Fedora obteve melhores resultados em 21 e com relação aos quesitos obrigatórios ficou melhor posicionado em 12 dos 20 (BAGDANOV et al., 2009).

### **3.3. Conclusão**

Para este trabalho o uso do Fedora se mostrou mais pertinente por se mostrar um ambiente mais flexível com relação ao uso de perfis de metadados definidos pelos usuários, além de possuir uma arquitetura que favorece a criação de novas funcionalidades a partir do uso de serviços Web e possuir mecanismos de interoperabilidade compatíveis com protocolos internacionalmente utilizados como o OAI-MPH. O aspecto relativo aos perfis de metadados foi relevante na decisão pela escolha do Fedora uma vez que o uso do perfil de metadados da APRI, o DC2AP, é um importante pilar na infraestrutura por documentar de forma detalhada os padrões de análise.

## 4. FEDORA

O foco principal da APRI como uma estrutura de armazenamento e disseminação de padrões de análise é oferecer um conjunto de ferramentas que possibilitem aos seus usuários encontrar formas de interagir com os padrões. Esta interação pode ocorrer de duas formas, primeiro por meio do acesso direto às informações básicas dos padrões (objetivos e soluções documentadas no DC2AP), ou por meio de um navegador Web, tendo acesso a estas informações utilizando serviços Web, possibilitando que outros agentes de software façam uso dos padrões para fins diversos.

O Fedora (*Flexible Extensible Digital Object Architecture*) é um *framework* conceitual que usa um conjunto de abstrações sobre informações digitais para fornecer as bases para sistemas de softwares que podem gerenciar informação digital (STAPLES; SHIN, 2012). O Fedora é capaz de prover um servidor de repositório apto a armazenar qualquer tipo de informação digital como documentos, vídeos, conjunto de dados, imagens e metadados que fornecem a descrição de cada um desses tipos de informações (DAVIS, 2008).

Os componentes do Fedora, suas abstrações e modelos de tratamento de informações o tornaram uma escolha viável para a implementação do repositório de dados previsto pela arquitetura da APRI por fornecer características como:

- Compatibilidade com a terminologia, modelagem da informação e interoperabilidade dos dados determinados pelas normas técnicas estabelecidas para sistemas OAIS;
- Permitir a auto descrição das informações a partir de metadados, incluindo-se relacionamentos semânticos entre estes dados;
- Persistência tanto dos metadados quanto dos dados em si através de um repositório;
- Acesso às informações e suas características por meio de Web Services;

- Integração dos objetos digitais documentados com outros sistemas ou objetos por meio de *middlewares* que usam estes serviços;
- É ofertado sob a GPL (*Gnu Public License*), que garante a possibilidade de utilizar, alterar e disponibilizar o código fonte, garantindo assim que as contribuições feitas ao software sejam disponibilizadas à comunidade.

O restante deste capítulo está estruturado da seguinte forma: a subseção 4.1 retrata historicamente a trajetória do Fedora; a subseção 4.2 descreve a estrutura do Fedora introduzindo os principais conceitos do *framework* como objetos digitais e seus componentes; a subseção 4.3 trata dos relacionamentos entre os objetos digitais dentro do repositório de metadados; a subseção 4.4 fornece os conceitos necessários para o entendimento da arquitetura de modelo de conteúdo visando a criação de objetos que sirvam de molde para a criação de outros objetos; a subseção 4.5 detalha o ambiente do Fedora que permite compreender como ocorre a comunicação entre o Fedora e o ambiente externo; a subseção 4.6 apresenta algumas conclusões do capítulo.

#### **4.1.Contextualização histórica do Fedora**

O Fedora surgiu como um projeto na Cornell University no ano de 1997 patrocinado pelo DARPA (*Defense Advanced Research Project Agency*) e pela NFS (*National Science Foundation*) com o objetivo de implementar repositórios de objetos digitais de diversos setores como bibliotecas, repositórios institucionais e publicação de conteúdo (STAPLES et al., 2003; DAVIS, 2008). Antes disso, um projeto do próprio DARPA lançou as bases para sistemas como o Fedora ao definir as noções iniciais de objetos digitais e implementar o Dienst, uma arquitetura de biblioteca de objetos digitais em múltiplos formatos com disseminação de conteúdo baseada em protocolos (LAGOZE et al., 2005).

O primeiro projeto de repositório de objetos digitais desenvolvido usando o Fedora foi na Universidade de Virginia em 2001. A primeira versão do sistema que visava a construção de um protótipo de uma biblioteca digital foi liberado em maio de 2003 sob a licença GPL (STAPLES et al., 2003).

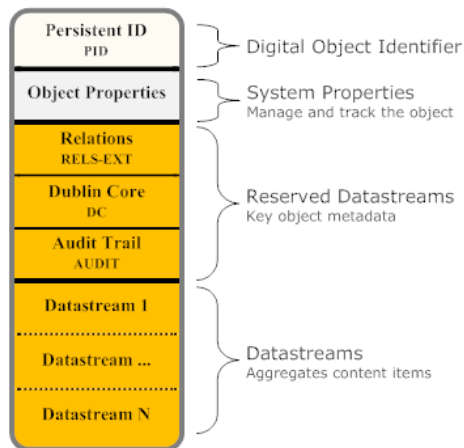
Em 2008 foi criada uma corporação sem fins lucrativos para atuar como provedora do desenvolvimento sustentável da plataforma e manter sua missão de construir tecnologias para possibilitar o acesso a informações digitais de forma permanente (DAVIS, 2008). Atualmente o desenvolvimento do Fedora está a cargo da DuraSpace, entidade sem fins lucrativos também responsável por outra plataforma de disponibilização de objetos digitais, o DSPACE.

## **4.2. Estrutura do Fedora**

As informações armazenadas no repositório do Fedora, tanto local quanto remotamente, são chamadas de Objetos Digitais e são definidas como instâncias de um modelo de dados que possui 2 componentes básicos: dados e metadados principais. Esses componentes possuem, entre outras informações, um identificador universal e um conjunto de outros metadados. Os objetos se inter-relacionam dentro da infraestrutura e esse relacionamento é expresso por meio de arquivos RDF. Os componentes de um objeto de dados e seus relacionamentos são estruturados a partir da Arquitetura de Modelo de Conteúdo – *Content Model Architecture* (CMA) – do Fedora, que descreve uma estrutura integrada para persistência e entrega de características essenciais dos objetos digitais do Fedora.

### **4.2.1 Objetos Digitais**

O modelo de dados de Objetos digitais do Fedora, ilustrado na *Figura 9* mostra os componentes padrão de um Objeto Digital. Um objeto digital no repositório descreve o conteúdo (dados e metadados) e serviços a ele aplicáveis. O modelo de objetos digitais é o construtor básico da arquitetura de modelo de conteúdo e outras funcionalidades do Fedora (DAVIS, 2012).



Fonte: (DAVIS, 2012)

*Figura 9 Modelo de dados de Objetos Digitais do Fedora*

Existem quatro tipos distintos de objetos digitais no Fedora que ajudam a definir como o sistema de repositório funciona (DAVIS, 2012):

- Objeto de dados: usado para representar uma entidade de conteúdo digital. Representa a informação de interesse principal da comunidade de interessados: imagens, livros, conjuntos de dados, textos eletrônicos, entre outros;
- Objeto de definição de serviços: um *Service Definition Object* – SDef – é um objeto de controle que armazena a definição de um serviço Web. Permite definir a interface que informa as operações existentes para determinado objeto digital sem especificar como a operação será executada;
- Objeto de *deployment* de serviço: um *Service Deployment Object* – SDep – é um objeto que controla como o repositório executará a operação descrita em um SDef;
- Objeto de modelo de conteúdo: um *Content Model* – CModel – é um *container* para um modelo de conteúdo para caracterizar uma classe de objetos digitais. Fornece um modelo de relacionamentos que são permitidos, excluídos ou obrigatórios entre grupos de objetos digitais.

Cada componente dentro do Fedora é manipulado por objetos digitais, cada qual contendo seu conjunto de *datastreams* (STAPLES; SHIN, 2012).

## 4.2.2 O Digital Object Identifier

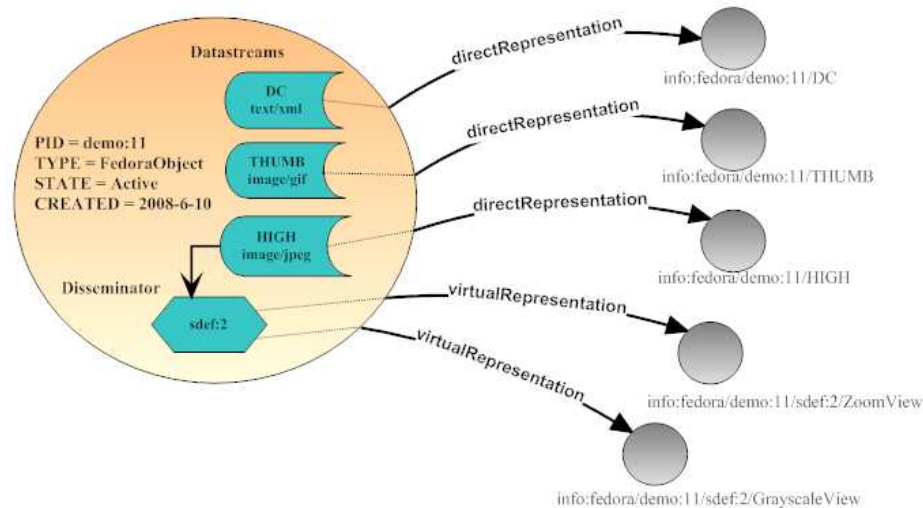
O *Persistent Identifier* – PID é o identificador único do objeto dentro do repositório. Para que a disseminação dos objetos digitais ocorra, uma função importante provida pelo repositório é o fornecimento de URL (*Universal Resource Locator*) persistentes, que visam garantir que seja possível recuperar um objeto ainda que seu endereço Web original seja alterado. O fornecimento de URL persistentes ocorre em duas camadas: a primeira camada é o uso de URI (*Universal Resource Identifier*), objetos que identificam de forma exclusiva um único objeto digital. Uma segunda camada opcional usada por muitos repositórios é o de um serviço de identificador persistente de terceiros.

No caso do Fedora utiliza-se, por padrão, o ‘info scheme’ (<http://info-uri.info/>), o que deixa a arquitetura livre para utilizar qualquer sistema de resolução de identificador (LAGOZE et al., 2005) embora existam alternativas, como o sistema Persistent-URL (PURL) que também pode ser utilizado no Fedora. Esses serviços funcionam empregando um nível extra de uma via indireta. A PURL aponta para um domínio de terceiros, que, por sua vez, resolve a URL para a URL do repositório.

## 4.2.3 Datastreams

Os *datastreams* são elementos que representam um item de conteúdo em um objeto digital do Fedora. Estes elementos garantem que um mesmo objeto digital pode ser descrito a partir de múltiplas visões da mesma informação (KAHN; WILENSKY, 2006). Desta forma, um objeto criado para descrever, por exemplo, um determinado padrão de análise pode conter tanto seu diagrama de classes quanto seu diagrama de casos de uso, além das outras informações definidas pelo DC2AP.

A *Figura 10* mostra um esquema de relacionamento de um objeto de dados com as propriedades do objeto e seus respectivos *datastreams*. O relacionamento entre o objeto e seus *datastreams* é feito por meio de pontos de acesso capazes de disseminar uma representação do objeto digital. Cada ponto de acesso é identificado por uma URI obedecendo o esquema de URI info do Fedora.



Fonte: (DAVIS, 2012)

*Figura 10 - Perspectiva de acesso de um objeto digital*

Cada *datastream* possui um identificador dentro do contexto do objeto digital e o Fedora reserva 4 identificadores específicos para *datastreams* específicos que desempenham papéis de descrição básica do objeto (DC), auditoria (AUDIT) e relacionamentos externos (RELS-EXT) e internos (RELS-INT). O conjunto destes *datastreams* é chamado de *Reserved Datastreams* (DAVIS, 2012).

### 4.3. Relacionamentos entre objetos no Fedora

Os objetos dentro do repositório podem relacionar-se uns com os outros através de metadados de relacionamento de objetos. Existe um conjunto de relacionamentos comuns definidos na ontologia de relacionamentos do Fedora, que fornece um meio genérico de interligar objetos semanticamente por meio de esquemas RDF. É possível ainda que usuários criem suas próprias ontologias entre seus objetos digitais. Utilizando os *datastreams* reservados RELS-INT e RELS-EXT pode-se criar, respectivamente, relacionamentos entre informações de um mesmo objeto e com outros objetos dentro de um repositório (DAVIS; WILPER, 2011a).

Os relacionamentos internos entre os objetos internos e externos são indexados pelo serviço de repositório do Fedora o que permite a criação de um grafo unificado

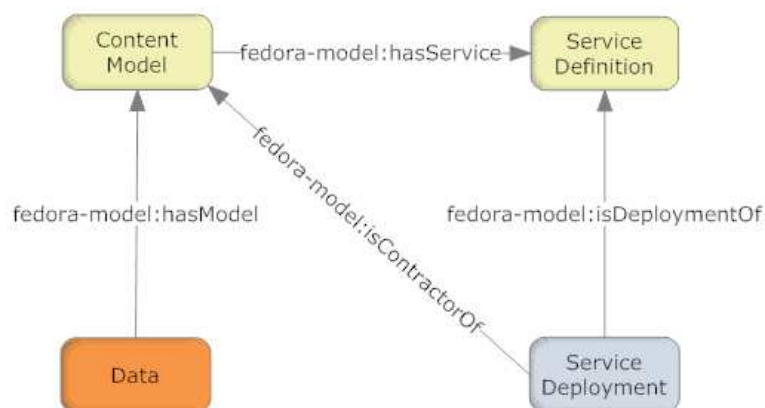
entre todos os objetos do repositório, que leva a possibilidade de recuperação desta informação através de consultas usando SPARQL e iTQL. Estas linguagens de consulta permitem que sejam efetuadas consultas usando sintaxes semelhantes às realizadas por meio da linguagem SQL aplicadas a banco de dados.

#### 4.4.A Arquitetura de Modelo de Conteúdo

A arquitetura de modelo de conteúdo – *Content Model Architecture (CMA)* – descreve uma estrutura integrada para armazenar e disponibilizar as características essenciais de um objeto digital (DAVIS; WILPER, 2011b). As características do modelo podem incluir informações estruturais, comportamentais e semânticas dos objetos digitais.

Como qualquer outra informação armazenada pelo Fedora, um modelo de conteúdo também é um objeto para o qual existe um identificador específico, o CModel, que é codificado em uma URI. Esta abordagem oferece duas vantagens principais: atende às necessidades internas do Fedora e é compatível com a arquitetura Web no sentido de ser representada usando RDF.

A CMA trata dos relacionamentos entre os quatro tipos de objetos fundamentais do Fedora. Estes relacionamentos, mostrados na *Figura 11*, são descritos em RDF por meio de *datastreams* do tipo RELS-EXT que também são coletados e indexados, favorecendo a recuperação de sua semântica.



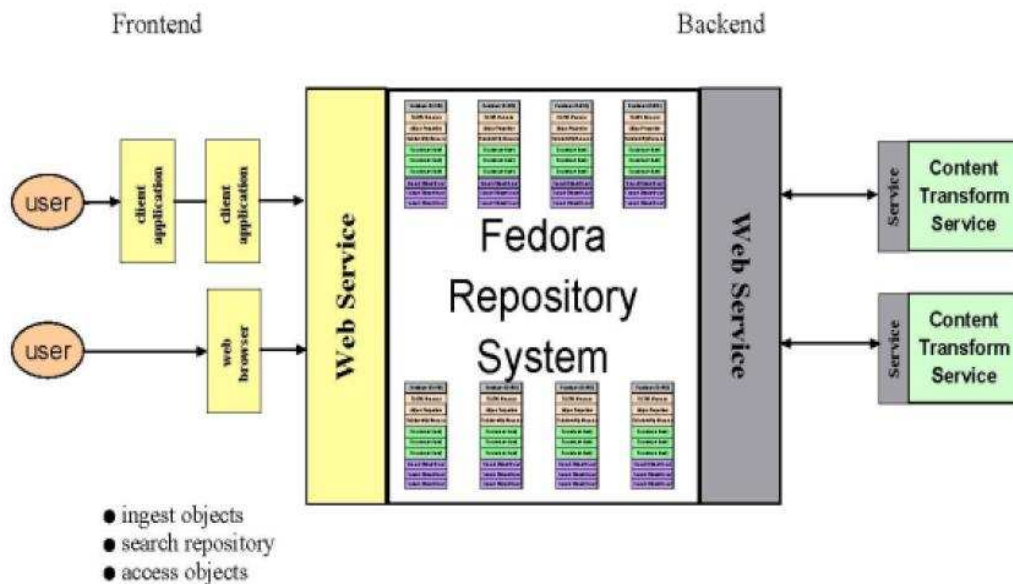
Fonte: (DAVIS; WILPER, 2011b)

*Figura 11 - Relacionamento entre os tipos fundamentais do Fedora na CMA*

## 4.5. Servidor de repositório do Fedora

Os aspectos descritos nas seções anteriores deste capítulo dão uma visão interna do Fedora enquanto ambiente de descrição de objetos digitais e de seus relacionamentos. Entretanto, para melhor entendimento do propósito de seu uso no desenvolvimento da solução proposta neste trabalho, se faz necessária uma visualização de como ocorre a interação entre o Fedora e outros agentes que fazem uso de seus serviços.

Usuários, sejam eles humanos ou outras aplicações, têm acesso aos objetos digitais exclusivamente por meio de chamadas a serviços Web. Internamente as transformações requeridas pelo usuário também são realizadas através de *Web Services*. A Figura 12 mostra um panorama destas interações: usuários, no *front end* do Fedora tendo acesso ao repositório via aplicações ou via navegador de internet e as requisições internas, no *Back end*. Em ambos os casos o acesso aos dados é intermediado por *Web Services*.



Fonte: (DAVIS, 2012)

Figura 12 Interação Cliente/Web Services

## 4.6. Conclusão

O repositório do Fedora é um sistema de armazenamento acessível via rede que possui mecanismos para gerenciamento dos dados que possibilita adicionar objetos, pesquisar informações relativas a ele e torná-las disponíveis ao usuário final. Sua arquitetura, conforme ilustra a *Figura 9*, possibilita tanto a definição da informação (metadados), quanto a própria informação em si (dados). A manipulação de dados e metadados é feita por meio de Web Services, expostos por meio de APIs de gerenciamento, acesso e busca, que permite que o Fedora seja utilizado como um componente integrável a sistemas que fornecem funções adicionais de organizações ou usuários finais.

Como a arquitetura do Fedora é projetada para gerenciar diferentes tipos de informação e este gerenciamento se dá através de serviços, é natural que a recuperação de informações acerca dos objetos ocorra da mesma forma. Serviços de busca internos ou de terceiros ficam a cargo da recuperação de informações dentro do sistema. Esses serviços recolhem conteúdo e metadados de várias formas para diversos propósitos (STAPLES; SHIN, 2012). A busca semântica também é prevista através do módulo de *Resource Index* que possibilita o uso da ontologia interna do próprio arcabouço ou de ontologia criada pelo usuário. Todos os relacionamentos são expressos usando RDF e podem ser recuperadas usando uma linguagem de consulta de RDF, por exemplo, SparQL ou iTQL.

## 5. ARQUITETURA PARA UM PORTAL DE ACESSO A APRI

Os capítulos anteriores contextualizaram os problemas relativos à subutilização de padrões de análise devido à ausência de recursos para documentá-los e ofereceu uma solução pautada no uso de uma infraestrutura para reuso de padrões de análise. Esta infraestrutura prevê a aplicação de um perfil de metadados baseado no padrão Dublin Core para documentar os padrões de análise – o DC2AP – e um repositório que pudesse armazená-los e disponibilizá-los. Os usuários dos padrões de análise podem ser tanto seres humanos quanto outras aplicações que irão interagir com o ambiente a partir de *Web Services*.

O *framework* Fedora foi escolhido como a solução de armazenamento dos metadados e dos padrões de análise por eles descritos. Esta escolha se deu a partir de critérios como forma de disseminação das informações, flexibilidade da solução no que diz respeito à criação de perfis de metadados e adequação a normas internacionais de documentação e acesso a estes metadados e aos dados por ele descritos.

Com os recursos definidos até este ponto é possível tratar os componentes da APRI que se referem aos repositórios de metadados e dos padrões de análise por eles documentados, além do acesso a eles por meio de *Web Services*. Outro componente importante da APRI diz respeito ao portal de padrões de análise, que faria a interface entre o usuário humano e os padrões de análise armazenados no repositório.

A interface gráfica com o usuário final oferecida pelo *framework* Fedora é bastante simplificada e exige conhecimentos mais aprofundados sobre sua terminologia, conceitos e arquitetura para que se possa tirar algum proveito das informações por ela fornecida. Tais fatores tornam pouco aconselhável seu uso para fins de disseminação das informações armazenadas no repositório a comunidades que não estejam familiarizadas com a abordagem por ele oferecida. Entretanto, como sua arquitetura favorece o acesso às informações catalogadas em seu repositório por meio de *Web Services*, é possível agregar outros serviços que forneçam interfaces gráficas que tratem as informações num nível de abstração mais apropriado à comunidade a que se destinam os dados do repositório de padrão de análise.

Este capítulo propõe uma arquitetura que permite a criação de um portal de padrões de análise utilizando um *framework* chamado Islandora que faz a comunicação do repositório de padrão de análises criado no Fedora com um sistema de gerenciamento de conteúdo – *Content Manager System* (CMS) – bastante conhecido e de fácil customização, o Drupal. Este CMS é a interface com o usuário final e fica responsável por formatar, exibir e permitir a busca dos padrões de análise armazenados no repositório. O restante do capítulo está estruturado da seguinte forma: a seção 5.1 descreve a arquitetura proposta para a definição do portal de padrões de análise. A seção 5.2 descreve os componentes desta arquitetura que ainda não foram abordados anteriormente e a forma como eles interagem e a seção 5.3 faz a conclusão do capítulo apresentando os resultados na integração dos componentes da arquitetura.

## 5.1. Descrição da arquitetura

O projeto de desenvolvimento do Fedora e sua arquitetura privilegiam o surgimento de vários sistemas que utilizam seu *core* para criar novos sistemas com finalidades específicas como acesso a dados bibliográficos, conteúdos institucionais e artigos científicos. Estes sistemas fazem uso dos *Web Services* fornecidos pelo Fedora e por meio de sua interface de programação de aplicações – *Application Programming Interface* (API) – são capazes de fornecer acesso direto aos objetos digitais e aos metadados que os descrevem.

Os aplicativos que usam os serviços de repositório de dados do Fedora em geral objetivam fornecer uma interface que propicie aos seus usuários finais um ambiente onde se possa obter informação de valor sobre os objetos digitais. A instalação básica do sistema também oferece um *front end* de acesso padrão que permite a manipulação e busca de objetos digitais. Entretanto, o uso desta interface requer o domínio de conceitos que podem estar fora do contexto ao qual os usuários pertencem. Não se pode exigir, por exemplo, que a comunidade de interessados em padrões de análise detenha conhecimentos acerca de padrões de metadados ou documentos RDF.

Existem vários projetos voltados para o desenvolvimento de interfaces que oferecem níveis de abstração da informação condizentes com o contexto do usuário, que são construídos com base no Fedora (STAPLES; SHIN, 2012). Alguns deles são

patrocinados pela própria Fedora-commons e outros desenvolvidos paralelamente por outras instituições interessadas na disponibilização da informação digital.

Um destes projetos, chamado Islandora, oferece uma abordagem que o torna adequado para este trabalho por permitir que os padrões de análise sejam disponibilizados por meio de um Web Site desenvolvido utilizando o CMS Drupal (STAPELFELDT; MOSES, 2013). O Islandora age como um módulo que torna possível a comunicação entre o repositório do Fedora e o Drupal que juntos criam um sistema de gestão de acervo digital robusto que pode ser utilizado para manutenção de dados de curta ou longa duração em um ambiente colaborativo. A Figura 13 fornece uma abstração da arquitetura proposta para desenvolver o ambiente completo da APRI, no qual o Fedora se encarrega de fornecer os repositórios de dados e metadados bem como os serviços necessários a sua gestão. O Drupal se encarrega de fornecer os serviços voltados para a interface de acesso aos padrões de análise com usuários humanos ao se integrar com o Fedora por intermédio do Islandora.

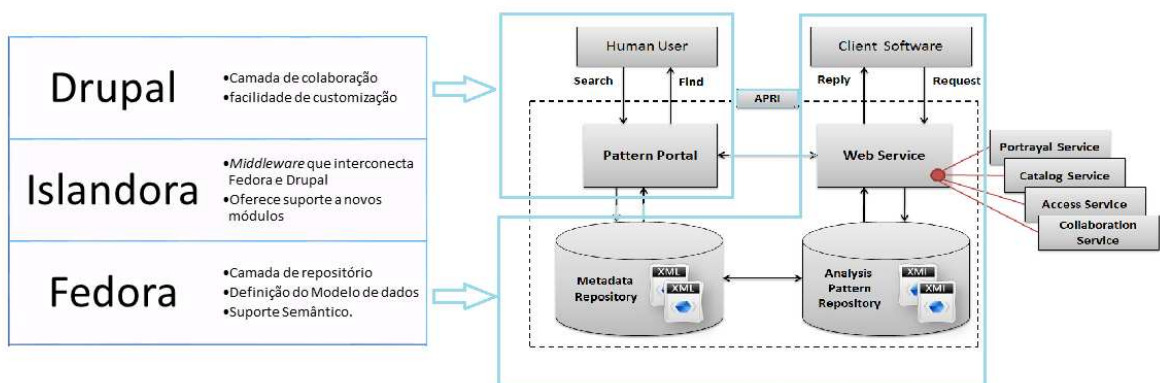


Figura 13 - Arquitetura da solução de integração entre Fedora, Drupal para realização da APRI

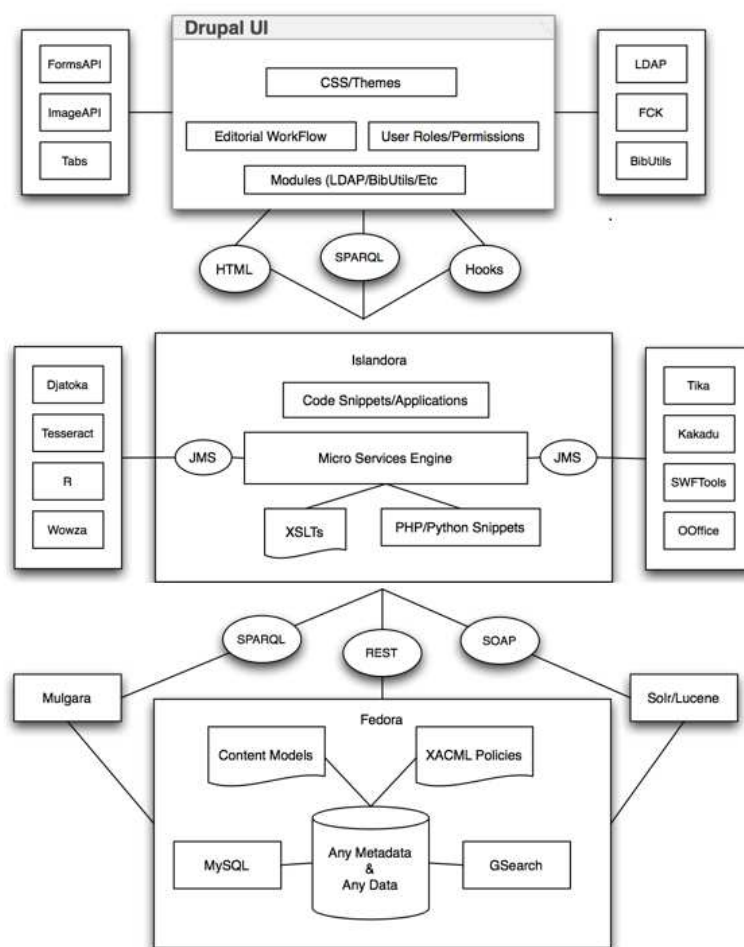
## 5.2. Componentes da arquitetura

Esta subsecção descreve os componentes da arquitetura que ainda não foram abordados nos capítulos anteriores e a forma como eles interagem.

## 5.2.1 Islandora

O Islandora é um sistema de gestão de acervo digital usado para integrar o servidor de repositórios digitais do Fedora e o sistema de gestão de conteúdo Drupal (STAPELFELDT; MOSES, 2013). O sistema é um *framework* que interliga os dois sistemas e com isso torna possível um ambiente que oferece uma interface de comunicação via Web de fácil gestão enquanto se comunica com um ambiente complexo de documentação e criação de objetos digitais.

A Figura 14 ilustra em detalhes a arquitetura do ambiente para suporte à APRI. Na base encontra-se o Fedora oferecendo os modelos de conteúdo e o repositório de dados e metadados. O Islandora faz a inter-relação entre as duas aplicações e o Drupal, no topo da arquitetura se responsabiliza pela disponibilização das informações ao usuário final.



Fonte: (STAPELFELDT; MOSES, 2013)

Figura 14 - A arquitetura da Solução

O Islandora é um projeto de código fonte aberto desenvolvido pela universidade *Prince Eduard Island*, situada na província canadense que leva o mesmo nome. O *framework* abrange uma série de componentes internos que lhe permitem a integração com o Drupal, Fedora e outros sistemas. Entre esses componentes cabe destacar alguns devido a sua relevância para os objetivos deste trabalho:

- Um módulo customizado para o Drupal que possibilita a integração com o Fedora;
- Um gestor de regras escrito em Java que o capacita a integrar diversos outros sistemas ao Fedora;
- Um componente editor de metadados exposto por meio do editor de formulários do Drupal;
- Um componente que utiliza o Apache *Solr* como provedor de funcionalidades de indexação e busca de objetos digitais.

Outras soluções de integração, *workflow* de consumo de objetos digitais, gerenciamento de metadados, perfis de busca, visualização de objetos entre outros podem ser desenvolvidas e integradas ao Islandora utilizando os *Solution Packs* (STAPELFELDT; MOSES, 2013). Estes pacotes são módulos customizados do Drupal e apresentam-se como um ponto de entrada para usuários que necessitam de tratar objetos de dados específicos.

O uso combinado destes módulos permite a gestão dos metadados e dados acerca dos padrões de análise, além de possibilitar a recuperação de informações sobre os objetos digitais armazenados no repositório.

### **5.2.2 Drupal**

No final da década de 90 as páginas web eram estáticas. Para que houvesse alteração no conteúdo exibido por site, era necessário que alguém, com conhecimento em HTML codificasse a informação no formato exigido pela linguagem. Os sistemas de gestão de conteúdo ou CMS surgiram nessa época em resposta a essa questão,

emergente na época, que era a necessidade de tornar o conteúdo mais dinâmico o exposto via Web. Esta abordagem, inovadora para aquele tempo, advogava a separação na gestão de dados (conteúdo) da estrutura (*layout*). A ideia central é a de que o conteúdo pode ser reutilizado de diversas formas e não apenas representado estaticamente via documento HTML. Várias soluções surgiram naquela época viabilizando esta ideia como, por exemplo, o Plone e o Typo3 (DELBRU et al., 2009).

O Drupal está entre os três principais CMSs em número de instalações e *pageviews* mundiais (OPENSOURCE-CMS, 2013). Os motivos para esta grande adesão por parte de usuários finais e desenvolvedores incluem a facilidade no gerenciamento do conteúdo, a existência de funcionalidades que permitem o gerenciamento de *workflow*, o controle de níveis de acesso dos usuários e a possibilidade de expansão através da criação de módulos para gerenciar conteúdos específicos.

O papel exercido pelo Drupal na arquitetura proposta é o de apresentar os conteúdos dos objetos digitais armazenados no Fedora. Desta forma supre-se a necessidade da existência de um componente da APRI denominado Portal de Padrões de Análise. O portal desenvolvido usando o Drupal contém funcionalidades que possibilitam buscar, expor e manipular os conteúdos documentados pelos *datastreams* dos objetos digitais dos padrões de análise. Todas estas tarefas já são fornecidas pelo Drupal para informações extraídas de bancos de dados relacionais e são aplicáveis também aos objetos digitais por meio de serviços de transformação oferecidos pelo Islandora. Estas funcionalidades podem ser estendidas para tratamento de conteúdo específico criando novos *Solutions Packs*, que são módulos que podem ser desenvolvidos para o Islandora e que fornecem novas funcionalidades tais como modelos de conteúdo, processamento de arquivos, gerenciadores de metadados, perfis de busca, visões customizadas, entre outros (STAPELFELDT; MOSES, 2013).

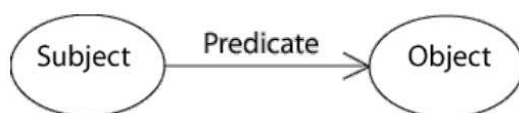
### **5.2.3 Componentes de busca**

A estruturação dos dados em ambientes altamente modularizados, como o exposto na Figura 13, permite grande flexibilidade na elaboração das informações e de suas inter-relações. Esta flexibilidade pode ser notada na maneira como vários mecanismos são utilizados para indexar e recuperar informações com objetivos distintos como dados e metadados. Os metadados além de estruturar a informação têm como

função indicar a semântica de suas informações e os dados podem conter informações de naturezas bem diversas como textos e vídeos. Indexar estas informações exige abordagens diferentes.

No perfil de padrão de análise criado para a APRI, o DC2AP, identificadores universais (URI) são associados a cada elemento do padrão e as relações semânticas entre estes elementos são descritas a partir de arquivos RDF (VEGI et al., 2013). O modelo utilizado pelo RDF permite interligar informações no formato da tripla: sujeito-predicado-objeto. Sujeito e objeto podem ser ambos URIs que identificam determinado recurso ou podem respectivamente ser uma URI e uma informação literal. O predicado indica como estas duas informações se relacionam (BIZER et al., 2009).

Um arquivo RDF é composto por um conjunto de triplas e o conjunto destes é chamado de Grafo RDF e pode ser ilustrado como um diagrama direcionado como mostra a *Figura 15* (KLYNE et al., 2004).



Fonte: (KLYNE et al., 2004)

*Figura 15 - Grafo RDF*

Para armazenar informações no formato descrito por arquivos RDF, o Fedora em seu módulo *Resource Index*, utiliza o *triplestore* Mulgara. O *Resource Index* é um módulo do Fedora que fornece a infraestrutura de indexação entre os objetos e seus componentes. O Fedora expressa relacionamentos com base em uma ontologia própria e é capaz de incluir declarações vindas de outras ontologias (DAVIS; PRATER, 2012).

Um *triplestore* é um banco de dados de finalidade específica para armazenar e recuperar as triplas sujeito-predicado-objeto. Diferente dos bancos de dados relacionais que são preparados para lidar com esquemas de dados estáveis e bem definidos, o Mulgara é capaz de lidar com informações em constante mudança devido a própria natureza dos metadados (MULGARA, 2012).

Os objetos no Fedora são descritos utilizando o FOXML – *Fedora Objects XML* – um arquivo formato XML que expressa o modelo de objetos digitais do Fedora. Cada objeto do Fedora ao ser ingerido pelo sistema tem seu arquivo FOXML transformado

em um formato que seja legível pelo serviço de busca e indexação do Fedora, chamado GSearch. A transformação efetuada pelo GSearch permite que as informações indexadas sejam usadas pela máquina de busca desenvolvida pela Apache Foundation chamada Solr (pronuncia-se “Solar”).

O Solr é uma máquina de busca construída a partir do projeto Apache Lucene. O Lucene fornece uma API para manipulação de consultas em arquivos indexados e o Solr foi desenvolvido utilizando os seus serviços como base. A principal característica do Solr é a busca textual, busca em arquivos enriquecidos (PDF, DOC, etc.) e buscas espaciais (TARGETT, 2013). O Islandora permite configurar a busca fornecida pelo Solr para atender às demandas específicas do modelo de conteúdo que está sendo utilizado pelo usuário.

### **5.3. Conclusão**

Para garantir a flexibilidade do ambiente onde os padrões de análise serão catalogados e expostos, a APRI precisa disponibilizar um ambiente que garanta que sejam obedecidas as complexas estruturas definidas para armazenar as informações sobre os padrões de análise e ao mesmo tempo forneça à comunidade de interessados ferramentas que lhes permita ter um contato produtivo com a informação buscada. Para garantir que estas duas metas sejam atendidas, foi utilizada uma arquitetura composta de 3 camadas. A camada inferior, o Fedora, fica a cargo de armazenar a informação e indexa-la para que a camada intermediária, o Islandora, possa atuar transportando a informação para a camada superior, o CMS Drupal, que é usado como interface com o usuário humano.

A comunicação entre as camadas ocorre sempre via Web Services expostos a partir de APIs fornecidas por cada nível, propiciando assim a comunicação com agentes não humanos e possibilitando a criação de novos serviços.

## **6. VIABILIZAÇÃO TECNOLÓGICA DA APRI: UMA VISÃO GERAL**

A proposta do projeto APRI é criar um ambiente onde padrões de análise estejam acessíveis aos usuários em diversos meios e para diversos fins. Um desses meios é o portal de padrão de análise que permite que o usuário tenha contato direto com os padrões armazenados no repositório. A arquitetura proposta viabiliza este aspecto da APRI ao disponibilizar o CMS Drupal como interface com o usuário.

Neste capítulo é apresentado o protótipo criado para o portal de padrões de análise da APRI chamado PANDORA, uma sigla para “Portal de Padrões de Análise usando o Fedora”. A subseção 6.1 trata da construção do modelo de conteúdo que faz o mapeamento dos elementos do DC2AP para um modelo de objetos do Fedora. A seção 6.2 descreve a criação de um formulário, usando recursos do Drupal, para o que seja possível o cadastro de dados sobre os padrões de análise e a subseção 6.3 apresenta a conclusão do capítulo.

### **6.1. Modelo de Conteúdo para o DC2AP**

O objetivo principal do DC2AP é a melhoria nas formas de descrição e recuperação de padrões de análise e, por conseguinte, aumento de seu reuso por meio de recursos computacionais. Por se tratar de um perfil de metadados baseado no padrão Dublin Core, ele possui outros elementos além dos elementos básicos pertencentes ao padrão base.

Os elementos que compõem o DC2AP permitem documentar os padrões de análise sob diferentes aspectos, como características descritivas, históricas, relacionais, de modelagem, entre outros. Uma descrição completa destes elementos mostrando um exemplo prático de sua aplicação pode ser vista em (VEGI, 2012).

Com base nos elementos do DC2AP foi desenvolvido um modelo de conteúdo para armazenar as informações sobre os padrões de análise. Os modelos de conteúdo podem ser entendidos de maneira análoga a uma classe no contexto da programação

orientada a objetos. Todas as informações criadas a partir desta classe são consideradas objetos desta classe.

Para criar o modelo de conteúdo foi utilizada uma extensão do modelo de conteúdo do Fedora chamado *Enhanced Content Model* (ECM). Na abordagem original do Fedora o modelo de conteúdo descreve como devem ser os objetos digitais criados a partir do CModel e indica quais os *datastreams* que eles devem conter. Os CModels criados a partir da abordagem ECM mantém a compatibilidade com o modelo antigo e acrescenta a possibilidade de definir a cardinalidade na relação entre os objetos e entre os *datastreams* de um objeto.

## 6.2. Interação com o sistema

A partir da definição do modelo de conteúdo para o DC2AP é possível a documentação dos padrões de análise usando a interface padrão do Fedora ou pela chamada dos *Web Services* oferecidos pelo Fedora e pelo Islandora. Porém, tanto o acesso via *Web Services* quanto pela interface do padrão do Fedora tornam muito complicada a tarefa de usar os padrões de análise. Esta situação pode ser contornada utilizando uma interface interativa por meio de um formulário criado usando uma ferramenta do Islandora para este fim.

Para que o usuário possa criar objetos que representam os padrões de análise dentro da estrutura do Fedora foi criado um formulário com o apoio do construtor de formulários do Islandora. O Islandora *Form Builder* torna possível para os usuários criar, copiar, editar, utilizar os formulários e associá-los a modelos de conteúdo no repositório (WILCOX; ANEZ, 2013). A Figura 16 exibe a interface de criação de formulários. No quadro a esquerda fica o editor de elementos, que permite determinar quais serão os campos presentes no formulário e no quadro da direita fica o descritor de elementos que permite descrever o elemento e impor restrições de preenchimento.

As guias *Advanced Form Controls* e *More Advanced Controls* oferecem funcionalidades mais elaboradas no que diz respeito à inserção de informações no repositório. Seu objetivo é cuidar das restrições de preenchimento como termos válidos para preenchimento da informação, formatos especiais, como data e hora, se o campo

receberá informações provenientes de arquivos (texto, PDF, imagens, etc) e funções de validação criadas pelo usuário.

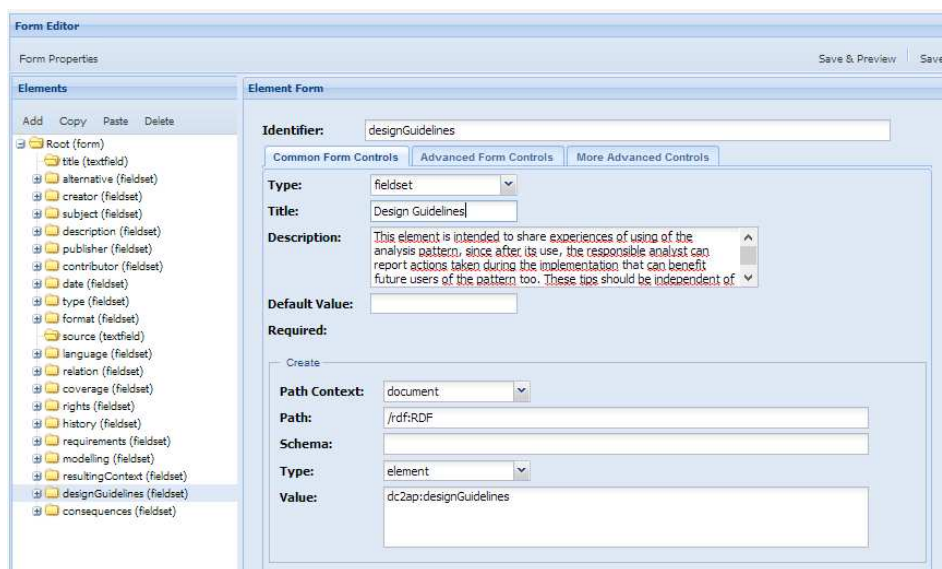


Figura 16 - Construtor de Formulário

A descrição do formulário é feita usando arquivos XML, conforme ilustra a Figura 17 e o sistema permite que este seja importado ou exportado do sistema, podendo ainda ser associado a um modelo de conteúdo. O arquivo está organizado desta forma apenas para fins de demonstração, pois para que se possa extrair maior eficiência de indexação das máquinas de busca do sistema, os arquivos originais são armazenados sem qualquer formatação, com todo seu conteúdo guardado em uma única linha (DAVIS, 2008).

```

17     <namespace prefix="dc">http://purl.org/dc/elements/1.1/</namespace>
18   </namespaces>
19 </properties>
20 <form>
21   <properties>
22     <type>form</type>
23     <access>TRUE</access>
24     <required>FALSE</required>
25     <tree>TRUE</tree>
26   </properties>
27   <children>
28     <element name="title">
29       <properties>
30         <type>textfield</type>
31         <access>TRUE</access>
32         <required>FALSE</required>
33         <title>Título</title>
34         <tree>TRUE</tree>
35         <actions>
36           <create>
37             <path>/rdf:RDF</path>
38             <context>document</context>
39             <schema />
40             <type>element</type>
41             <prefix>dc</prefix>
42             <value>dc:title</value>

```

Figura 17 - XML de descrição do formulário

A Figura 18 exibe uma previsão da tela de registro do objeto digital criada a partir do *Form Builder*.

The screenshot shows the 'Form Editor' interface. On the left, the 'Elements' tree lists various fieldsets and textfields, including 'title (textfield)', 'creator (fieldset)', and 'date (fieldset)'. The 'Preview' area on the right shows a header for 'APRI - PANDORA' with navigation links like 'Digital Repository' and 'Islandora Administration Map'. Below the header, the 'Preview Form' section contains three input fields: 'Título:', 'Título Alternativo:', and 'Criador:'. The 'Título Alternativo:' field has the value 'alternative' entered.

Figura 18 - Tela de Previsão do Formulário

### **6.3. Conclusão**

O acesso aos padrões de análise por usuários humanos é um importante componente proposto para contribuir na disseminação dos padrões de análise. Neste capítulo foram apresentados elementos que viabilizam este portal: o modelo de conteúdo baseado no DC2AP e o formulário criado a partir deste modelo para inserção de dados no repositório de dados.

## 7. CONCLUSÕES

Este trabalho propôs um ambiente para viabilizar tecnologicamente uma infraestrutura para reuso de padrões de análise denominada APRI. O objetivo da APRI é fomentar a disseminação do uso de padrões de análise ao fornecer recursos de documentação e recuperação destes padrões de forma legível por máquina ou para acesso direto de pessoas interessadas neste tipo de recurso.

Para a realização deste objetivo foi sugerida uma arquitetura em 3 camadas onde a primeira se responsabiliza pelo repositório de padrões de análise e a segunda se encarrega de fazer a interligação da primeira com a terceira, que por sua vez se encarrega de fazer a exibição das informações sobre os padrões de análise.

A primeira camada é concretizada pelo Fedora, um *framework* conceitual que usa um conjunto de abstrações sobre informações digitais para fornecer as bases para sistemas de softwares que podem gerenciar informação digital. A segunda camada é realizada pelo Islandora, um *framework* que integra-se ao Fedora para fornecer-lhe uma interface gráfica. Essa interface gráfica fica a cargo do CMS Drupal.

As duas primeiras camadas contribuem para tornar possível o acesso às informações via *Web Services*. Ambas oferecem APIs que possibilitam a interação direta entre os padrões de análise armazenados no repositório e sistemas computacionais capazes de consumir *Web Services*. Este fato abre possibilidades de se estender a arquitetura criando novos serviços de integração, buscas específicas, interação com outras infraestruturas de fins semelhantes.

O Fedora oferece recursos para documentar os elementos descritivos dos padrões de análise como *Linked Data*, fazendo com que seja possível recuperar a semântica das relações entre os elementos que descrevem um padrão e também a semântica da relação entre os próprios padrões. Mecanismos de indexação dentro do próprio Fedora são capazes de recolher as informações das triplas e disponibiliza-las a máquinas de busca genéricas utilizadas pelo *framework*, permitindo realizar a busca semântica.

O Drupal contribui para realizar o objetivo de criação de um ambiente de comunicação direta com usuários humanos ao fornecer uma interface de gerenciamento de conteúdo bastante flexível e expansível. Este ambiente de interação entre usuário final e padrões de análise recebeu o nome de Pandora, para indicar que se trata de um Portal de Padrão de Análise usando o Fedora.

A interação do Drupal com o Fedora por meio do Islandora propiciou a disponibilização de um formulário que permite ao usuário registrar padrões de análise sem que este tenha que dominar conceitos diretos relacionados aos metadados que documentam os padrões de análise.

O modelo de desenvolvimento orientado a serviços adotado pelas 3 soluções e a flexibilidade inerente ao próprio Fedora para descrever os mais diferentes tipos de informação, abre espaço para o desenvolvimento e integração de diversas outras ferramentas.

Alguns trabalhos futuros podem ser realizados visando abranger outros tipos de padrões usados no desenvolvimento de projetos de software, como padrões de projeto, padrões de requisitos, padrões de código. Pode-se também pensar em documentar projetos de software e criar relações semânticas entre os padrões que contribuíram para sua realização. Por fim, um experimento utilizando a APRI, a partir do PANDORA, com uma volume de padrões de análise carregados na infraestrutura é fundamental.

## 8. REFERÊNCIAS BIBLIOGRÁFICAS

- BAGDANOV, A.; KATZ, S.; NICOLAI, C.; SUBIRATS, I. Fedora Commons 3.0 versus DSpace 1.5: Selecting an enterprise-grade repository system for FAO of the United Nations. 4th International Conference on Open Repositories. **Proceedings...**, 2009. Atlanta: Georgia Institute of Technology.
- BASS, M. J.; STUVE, D.; TANSLEY, R. et al. **DSpace – Institutional Digital Asset Services – Technology & Architecture**. Disponível em: <http://libraries.mit.edu/dspace-mit/technology/architecture.pdf>. Acesso em 30/06/2013
- BÉJAR, R.; LATRE, M. Á.; NOGUERAS-ISO, J.; MURO-MEDRANO, P. R.; ZARAZAGA-SORIA, F. J. An architectural style for spatial data infrastructures. **International Journal of Geographical Information Science**, v. 23, n. 3, p. 271-294, mar. 2009.
- BIZER, C.; HEATH, T.; BERNERS-LEE, T. Linked Data - the story so far. **International Journal on Semantic Web and Information Systems**, v. 5, n. 3, p. 01-22, 2009.
- BLAIMER, N.; BORTFELDT, A.; PANKRATZ, G. **Patterns in object-oriented analysis**, Working Paper No. 451, Faculty of Business Administration and Economics, University of Hagen (Germany), 2010.
- CCSDS. **Reference Model for an Open Archival Information System (OAIS)**. 2nd ed. Washington: CCSDS Secretariat, 2012.
- COAD, P. Object-oriented patterns. **Communications of the ACM**, v. 35, n. 9, p. 152–159, 1992.
- CROW, R. **A guide to institutional repository software**. Disponível em: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:A+Guide+to+Institutional+Repository+Software#0>. Acesso em: 26/06/2013.
- DAVIS, D. **Introduction to Fedora**. Disponível em: <https://wiki.duraspace.org/display/FEDORACREATE/Tutorial+1+-+Introduction+to+Fedora>. Acesso em: 24/05/2013.
- DAVIS, D. **Fedora Digital Object Model**. Disponível em: <https://wiki.duraspace.org/display/FEDORA36/Fedora+Digital+Object+Model>. Acesso em: 28/05/2013.
- DAVIS, D.; PRATER, S. **Resource Index**. Disponível em: <https://wiki.duraspace.org/display/FEDORA36/Resource+Index>. Acesso em: 07/07/2013.
- DAVIS, D.; WILPER, C. **Digital Object Relationships**. Disponível em: <https://wiki.duraspace.org/display/FEDORA36/Digital+Object+Relationships>. Acesso em: 01/07/2013a.
- DAVIS, D.; WILPER, C. **Content Model Architecture**. Disponível em: <https://wiki.duraspace.org/display/FEDORA36/Content+Model+Architecture>. Acesso em: 01/07/2013b.

- DELBRU, R.; CLARK, T.; POLLERES, A.; DECKER, S. Produce and Consume Linked Data with Drupal. International Semantic Web Conference - ISWC. **Proceedings...** v. 1380, p.763–778, 2009. Washington.
- DCMI - DUBLIN CORE METADATA INITIATIVE. **The Essential Elements of Network Object Description.** Disponível em: <http://dublincore.org/workshops/dc1/general.shtml>. Acesso em: 18/6/2013.
- ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados.** 6a ed. São Paulo: Pearson-Addison-Wesley, 2011.
- FERRIS, C.; FARRELL, J. What Are Web Services ? **Communications of the ACM**, v. 46, n. 6, p. 31–31, 2003.
- FOWLER, M. **Analysis patterns: reusable object models.** 1st ed. Indianapolis: Pearson Education, 1996.
- GLASS, R. L. **Facts & Fallacies of Software Engineering.** 3rd ed. Boston: Addison Wesley, 2002.
- HAMZA, H. **A foundation for building stable analysis patterns.** 2002. 146 f. Tese (doutorado em computação) – University of Nebraska. Lincoln.
- HAMZA, H.; FAYAD, M. E. Model-based Software Reuse Using Stable Analysis Patterns. ECOOP02 - Workshop on Model-based Software Reuse. **Proceedings...** v. 2548, 2002. Málaga, Spain: Springer.
- HILLMANN, D. **Using Dublin Core - The elements.** Disponível em: <<http://dublincore.org/documents/usageguide/elements.shtml>>. Acesso em: 01/06/2013.
- HULL, R. Web services composition: a story of models, automata, and logics. IEEE International Conference on Web Services (ICWS'05). **Proceedings...** p.xx–xxi vol.1, 2005. Orlando: Florida.
- KAHN, R.; WILENSKY, R. A framework for distributed digital object services. **International Journal on Digital Libraries**, v. 6, n. 2, p. 115–123, 2006. Disponível em: <<http://www.springerlink.com/index/10.1007/s00799-005-0128-x>>. Acesso em: 15/3/2013.
- KLYNE, G.; CARROLL, J. J. **Resource description framework (RDF): concepts and abstract syntax.** 2004. Disponível em: <<http://www.w3.org/TR/rdf-concepts/>>. Acesso em: 10/7/2013.
- LAGOZE, C.; PAYETTE, SANDY; SHIN, E.; WILPER, C. **Fedora: an architecture for complex objects and their relationships.** **International Journal on Digital Libraries**, v. 6, n. 2, p. 124–138, 2005. Disponível em: <<http://www.springerlink.com/index/10.1007/s00799-005-0130-3>>. Acesso em: 15/3/2013.
- LISBOA FILHO, J.; IOCHPE, C.; BORGES, K. A. V. Analysis patterns for GIS data schema reuse on urban management applications. **CLEI Electronic Journal**, v. 5, n. 2, p. 1–15, 2002.
- MARCIAL, L. H.; HEMMINGER, B. M. Scientific data repositories on the Web: An initial survey. **Journal of the American Society for Information Science and Technology**, v. 61, n. 10, p. 2029–2048, 2010.

- MULGARA. **Mulgara - Semantic Store**. Disponível em: <<http://docs.mulgara.org/overview>>. Acesso em: 9/7/2013.
- NEBERT, D. D. Developing spatial data infrastructures: the SDI cookbook, version 2.0. [S.l.]: GSDI-Technical Working Group. 2004. Disponível em: <<http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf>>. Acesso em: 21/09/2012
- NILSSON, M.; BAKER, T.; JOHNSTON, P. **The Singapore framework for Dublin Core application profiles**. 2008. Disponível em: <<http://dublincore.org/documents/singapore-framework/>>. Acesso em: 21 janeiro 2013.
- NISO.ORG. **Understanding Metadata**. ,2004. Bethesda,: NISO Press. Disponível em: <<http://www.niso.org/publications/press/UnderstandingMetadata.pdf>>. 19/06/2013.
- NOGUERAS-ISO, J.; ZARAZAGA-SORIA, F. J.; MURO-MEDRANO, P. R. **Geographic Information Metadata for Spatial Data Infrastructures: Resources, Interoperability and Information Retrieval**. Information Retrieval, 2005. Secaucus, New Jersey, USA: Springer-Verlag New York.
- OLIVEIRA, R. R. DE; CARVALHO, C. L. DE. **Bibliotecas Digitais e o Repositório Fedora**. Goiânia, 2011.
- OPENSOURCE-CMS. **CMS Market Share**. Disponível em: <<http://www.opensourcecms.com/general/cms-marketshare.php>>. Acesso em: 08/07/2013.
- PANTOQUILHO, M.; RAMINHOS, R.; ARAÚJO, J. Analysis Patterns Specifications: Filling the Gaps. In: VIKING PLOP CONFERENCE. **Proceedings...** v. 18, p.1–13, 2003. Bergen, Norway.
- PAPAZOGLU, M.; DUBRAY, J. JACQUES. **A Survey of Web Services Technologies**. Computer, 2004. Trento - Italia.
- PAPAZOGLU, M. P.; GEORGAKOPOULOS, D. Introduction: Service-oriented computing. **Communications of the ACM**, v. 46, n. 10, p. 24, 2003.
- PURAO, S.; STOREY, V. C.; HAN, T. Improving Analysis Pattern Reuse in Conceptual Design: Augmenting Automated Processes with Supervised Learning. **Information Systems Research**, v. 14, n. 3, p. 269–290, 2003.
- RAMINHOS, R.; PANTOQUILHO, M.; ARAÚJO, J.; MOREIRA, A. A systematic analysis patterns specification. In: INTERNATIONAL CONFERENCE ON ENTREPRISE INFORMATION SYSTEMS (ICEIS), 8, 2006, Paphos, Cyprus. **Proceedings...** [S.l.]: [s.n.], 2006, p. 453-456
- RIEHLE, D.; ZÜLLIGHOVEN, H.; BANK, U. Understanding and Using Patterns in Software Development. **Theory and Practice of Object Systems**, v. 1, p. 1–14, 1996.
- ROBERTSON, S. **Reusing Requirements: Taking Advantage of What you Know**. Atlantic, p. 1–36, 2000.
- RUHLE, S.; BAKER, T.; JOHNSTON, P. **DCMI User Guide**. Disponível em: <[http://wiki.dublincore.org/index.php/User\\_Guide](http://wiki.dublincore.org/index.php/User_Guide)>. Acesso em: 1/5/2012.

- SAYÃO, L. F. Uma outra face dos metadados: informações para a gestão da preservação digital. **Encontros Bibli: Revista Eletrônica de Biblioteconomia e Ciência da Informação**, p. 1–31, 2010.
- STAPELFELDT, K.; MOSES, D. Islandora and TEI: Current and Emerging Applications/Approaches. **Journal of the Text Encoding Initiative**, v. 5, n. Issue 5, 2013.
- STAPLES, THORNTON; WAYLAND, R.; PAYETTE, SANDRA. The Fedora Project. **D-Lib Magazine**, v. 9, n. 4, p. 1–12, 2003.
- STAPLES, THORNY; SHIN, E. **Getting Started with Fedora**. Disponível em: <<https://wiki.duraspace.org/display/FEDORA36/Getting+Started+with+Fedora>>. Acesso em: 20/1/2013.
- TARGETT, C. **Solr: A Quick Overview**. Disponível em: <<https://cwiki.apache.org/confluence/display/solr/A+Quick+Overview>>. Acesso em: 7/7/2013.
- TSAI, W.; CHEN, Y.; BITTER, G. **Introduction to Service-Oriented Computing**. Arizona State, 2006. Disponível em: <<http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Introduction+to+Service-Oriented+Computing#0>>. Acesso em: 11/12/2011.
- VEGI, L. F. M. **Technical description of Dublin Core application profile to analysis patterns (DC2AP)**. 2012. Disponível em: <<http://purl.org/dc2ap/TechnicalDescription>>. Acesso em: 13 abril 2013.
- VEGI, L. F. M.; LISBOA FILHO, J.; CROMPVOETS, J. A machine-processable Dublin Core application profile for analysis patterns to provide linked data. In: INTERNATIONAL CONFERENCE ON DUBLIN CORE AND METADATA APPLICATIONS, 2012, Kuching, Sarawak, Malaysia. **Proceedings...** Kuching: DCMI-Dublin Core Metadata Initiative, 2012. pp. 23-32.
- VEGI, L. F. M.; LISBOA FILHO, J. COSTA, G. L. S.; OLIVEIRA, A. P.; BRAGA, J. L. DC2AP: A Dublin Core application profile to analysis patterns. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING (SEKE), 24., 2012, Redwood City, San Francisco Bay, USA. **Proceedings...** Skokie, IL, USA: Knowledge Systems Institute Graduate School, 2012. pp. 511-516.
- VEGI, L. F. M.; PEIXOTO, D. A.; SOARES, L. S.; LISBOA FILHO, J.; OLIVEIRA, A. P. An infrastructure oriented for cataloging services and reuse of analysis patterns. In: INT. WORKSHOP ON REUSE IN BUSINESS PROCESS MANAGEMENT (rBPM in conjunction with The 9th Int. Conf. on Business Process Management), 2011, Clermont-Ferrand, France. **Proceedings...** Berlin Heidelberg: Springer LNBIP N.100, Parte 4 - Lecture Notes in Business Information Processing, 2012. pp.338-343.
- WILCOX, D.; ANEZ, M. **XML Form Builder**. Disponível em: <<https://wiki.duraspace.org/display/ISLANDORA6131/XML+Form+Builder>>. Acesso em: 26/06/2013.