Universidade de Aveiro
2012
Departamento de Electrónica, Telecomunicações e Informática

Jorge André Fonseca Rocha

INTERFACE DE ACESSO A REPOSITÓRIOS DIGITAIS

Universidade de Aveiro
2012
Departamento de Electrónica, Telecomunicações e Informática

Jorge André Fonseca Rocha

INTERFACE DE ACESSO A REPOSITÓRIOS DIGITAIS

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Sistemas de Informação, realizada sob a orientação científica do Doutor Joaquim Arnaldo Carvalho Martins, Professor Catedrático do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri

presidente	Prof. Doutor Joaquim Manuel Henriques de Sousa Pinto Professor Auxiliar, Universidade de Aveiro
orientador	Prof. Doutor Joaquim Arnaldo Martins Professor catedrático do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro
arguente principal	Professora Doutora Ana Alice Rodrigues Pereira Baptista Professora auxiliar do Departamento de Sistemas de Informação da Escola de Engenharia da Universidade do Minho.

agradecimentos

Ao professor Joaquim Arnaldo Martins, meu orientador, pela confiança, apoio, motivação e empenho na concretização deste trabalho.

E a todos os que partilham conhecimento, pois é através deste que se consegue alcançar melhores resultados.

palavras-chave Repositórios digitais, Interoperabilidade.

resumo

Nos últimos anos o número de repositórios para gerir conteúdos digitais tem aumentado, tornando difícil a escolha para as necessidades dos gestores das bibliotecas digitais. Assim, é necessário construir um sistema que atue como uma ponte entre os diferentes repositórios.

Neste sentido, foram definidos serviços/ operações que facilitam a troca e agregação de informação entre vários repositórios digitais. Ao contrário de alguns protocolos e aplicações existentes no mercado, não é necessário alterar o repositório para que o sistema interaja com este.

Os resultados comprovam que este sistema facilita a gestão das bibliotecas digitais, uma vez que permite enriquecer a biblioteca, disponibilizar informação a outras bibliotecas e permite mudar os conteúdos de um sistema para outro sem perda de informação.

keywords Digital Repositories, Interoperability.

abstract

In the last years, the number of repositories to manage digital contents has been growing, making hard the choice for needs of digital libraries manager. So, it's necessary to build a system that will act as a bridge between different repositories.

In this direction, this work defines a set of services/ operations that facilitate the exchange and aggregation of information between many digital repositories. Unlike some protocols and applications on the market, it isn't necessary to change the repository for the system to interact with it.

The results show that this system helps in the digital libraries management, because it makes the library more complete, it provides information to other libraries and it allows change the contents from a system to another without information loss.

Índice

Índice ..... **i**

Índice de Figuras ..... **iii**

Acrónimos ..... **iv**

Introdução **1**

1.1. Das bibliotecas tradicionais às digitais 2
1.1.1. Enquadramento 2
1.1.2. Bibliotecas Tradicionais 2
1.1.3. Bibliotecas digitais 3
1.1.4. Tradicional vs. Digital 4

2. Repositórios digitais **6**

2.1. Requisitos 6

2.2. Repositórios Digitais 7
2.2.1. *DSpace* 8
2.2.2. *EPrints* 9
2.2.3. *FEDORA* 10
2.2.4. *Greenstone* 11
2.2.5. *Digital Commons* 12
2.2.6. *IntraLibrary* 12
2.2.7. *CONTENTdm* 13
2.2.8. *JeromeDL* 14
2.2.9. *DuraCloud* 15

2.3. Comparação entre repositórios 16
2.3.1. Resumo 16

2.4. Interoperabilidade 24

2.5. Metadados 26

2.6. Análise das Características 27

2.7. Casos de estudo 29

3. Proposta de API de Serviços **31**

3.1. Motivação 31

3.2. Requisitos 31
3.2.1. Casos de utilização 31
3.2.2. Arquitetura 33
3.2.3. Modelo de Dados 35

3.3. Desenvolvimento 35
3.3.1. Repositórios 36
3.3.2. Módulo de aquisição e inserção 36
3.3.3. Módulo de agregação 38

3.4. Testes 44
3.4.1. Procedimento 44
3.4.2. Resultados finais 46

4. Conclusões 47

Bibliografia 49

Anexos 53

Anexo A - Distribuição geográfica de alguns repositórios digitais 53

Anexo B - Identificadores 55

Índice de Figuras

Figura 1. Diagrama simplificado da possível comunicação entre os repositórios 31
Figura 2. Casos de utilização (comunicação entre 2 repositórios) 32
Figura 3. Casos de utilização (agregador) 33
Figura 4. Arquitetura 34
Figura 5. Modelo de dados 35
Figura 6. Comunicação entre os repositórios de teste 44
Figura 7. Repositórios no Mundo 53
Figura 8. Tipo de conteúdo e área cientifica mais utilizados nos repositórios 53
Figura 9. Repositórios em Portugal 54
Figura 10. Nomenclatura do ISBN 55
Figura 11. Nomenclatura do ISSN 55
Figura 12. Exemplo de um URL 56
Figura 13. Exemplo de um URN 56
Figura 14. Exemplo de um identificador PURL 57
Figura 15. Relação entre URI, URL e URN 57
Figura 16. Exemplo de um identificador *Handle System* 57

Acrónimos

ACL – Access Control List

ATOM – The Atom Publishing Protocol

Atom (feed) – Atom Sindication Format

CNRI – Corporation for National Research Initiatives

DC – Dublin Core

DERI – Digital Enterprise Research Institute

FEDORA – Flexible Extensible Digital Object Repository Architecture

HP – Hewlett-Packard

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

IBM – International Business Machines Corporation

ISO – International Organization for Standardization

JISC – Joint Information Systems Committee

LDAP – Lightweight Directory Access Protocol

LOM – Learning Object Metadata

METS – Metadata Encoding and Transmission Standard

MD5 – Message-Digest Algorithm

MIT – Massachusetts Institute of Technology

NGO – Non-Governmental Organization

ODRL – Open Digital Rights Language

OAI-ORE – Open Archives Initiative - Object Reuse and Exchange

OAI-PMH – Open Archives Initiative Protocol for Metadata Harvesting

OpenDOAR – The Directory of Open Access Repositories

PS – PostScript

PURL – Persistent Uniform Resource Locator

RDFa – Resource Description Framework – in – attributes

RDF – Resource Description Framework

REST – Representational State Transfer

RSS – Rich Site Summary

RTF – Rich Text Format

SGML – Standard Generalized Markup Language

SHA – Secure Hash Algorithm

SOAP – Simple Object Access Protocol

SWF – ShockWave Flash Format

SWORD – Simple Web-service Offering Repository Deposit

UNESCO – United Nations Educational, Scientific and Cultural Organization

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

URN – Uniform Resource Name

XACML – eXtensible Access Control Markup Language

XML – Extensible Markup Language

Introdução

Desde tábuas de argila, encontradas numa pequena região do sul do Iraque, algumas das quais com 5000 mil anos, passando pelos pergaminhos até ao papel, como hoje o conhecemos, as bibliotecas têm o objetivo de expandir e preservar o conhecimento. Se estas não existissem, era quase impossível compreendermos ou conhecermos a história do mundo.

Com o aparecimento do computador e sobretudo com a expansão da *Internet*, a quantidade de informação aumentou e, com esta, a necessidade de adaptar as bibliotecas às novas realidades. Estas novas bibliotecas, designadas por bibliotecas digitais, ao contrário das anteriores, permitem a redução de custos de manutenção, pois a informação é armazenada em formato digital, reduzindo o espaço físico utilizado, aumenta a disponibilidade da mesma, uma vez que é possível aceder à informação a qualquer momento e em qualquer local e, com as tecnologias de conversão de um formato analógico para um digital, é possível melhorar a qualidade da informação – remover manchas, tornar texto mais legível. No entanto, como o acesso à informação é simples, a possibilidade de manipulação dessa informação também é simples, colocando problemas de segurança, como a alteração de direitos de autor ou o acesso indevido. Outro fator que coloca algumas restrições na utilização de bibliotecas digitais está relacionado com a aquisição de informação, ou seja, um leitor, antes da aquisição de um livro, lê algumas partes do mesmo, o que nem sempre é possível nas bibliotecas digitais. A dificuldade de troca de informação entre bibliotecas digitais é outro problema.

Para colmatar este problema, no âmbito do projeto EDUCA, um projeto cujo objetivo é a agregação, pesquisa e disponibilização de conteúdos multimédia em diferentes áreas, surgiu a ideia de se explorar um conjunto de serviços/ operações que permitam a troca de informação entre diferentes bibliotecas, independentemente da tecnologia que estas utilizem.

Assim, numa primeira fase, serão analisadas algumas das principais características e diferenças entre os sistemas utilizados na gestão de conteúdos armazenados nas bibliotecas digitais (capítulo 2.2 e capítulo 2.3, respetivamente) e as limitações existentes nos protocolos e projetos utilizados na troca de informação entre bibliotecas digitais (capítulo 2.7). Após essa análise, no capítulo 3 serão apresentados os passos principais que envolvem este trabalho, ou seja, será proposto um conjunto de operações para a troca e agregação de conteúdos digitais. No final do capítulo serão descritos os procedimentos utilizados na realização dos testes a estas operações. Por fim, no último capítulo serão descritas algumas conclusões e melhorias a realizar em trabalhos futuros.

Antes da análise de alguns sistemas utilizados na gestão de conteúdos digitais será apresentada a evolução das bibliotecas, desde a sua origem, nomeadamente as necessidades das bibliotecas ao longo do tempo.

1.1. Das bibliotecas tradicionais às digitais

1.1.1. Enquadramento

Ao longo dos séculos as bibliotecas sofreram grandes alterações, adaptando-se às necessidades de cada época. Desde um simples arquivo até grandes bibliotecas com milhares de obras, e desde documentos escritos em argila até documentos armazenados digitalmente, as bibliotecas estiveram e estão quase sempre ligadas à vida das pessoas, pois sem elas o conhecimento perder-se-ia. Neste capítulo serão apresentadas dois grandes tipos de bibliotecas – tradicionais e digitais. As tradicionais armazenam as obras de forma física, como livros, pergaminhos, tábuas de argila, pinturas em diferentes materiais, enquanto as digitais utilizam ferramentas para disponibilizá-las de forma digital (livros digitais, áudio, vídeo, imagem).

1.1.2. Bibliotecas Tradicionais

Alguns dos primeiros documentos (escritos em tábuas de argila e em *Cuneiform*), foram descobertos num templo na *Sumária* (que corresponde a uma pequena região do atual Iraque), alguns dos quais escritos no século XXVI a.C. Por essa razão, os historiados dizem que, possivelmente, as bibliotecas surgiram entre os séculos XXVI e XXIV a.C. (Casson 2002) (Krasner-Khait 2001). Com o aparecimento de novas formas de escrita e com a expansão desta, as bibliotecas tiveram de se adaptar às novas realidades. Contudo, a mais significativa biblioteca do mundo e com milhares de pergaminhos (Biblioteca de Alexandria) (Britannica 2012), surgiu no Egito por volta do século VIII a.C. e funcionava como um centro de estudos (Whitehouse 2004). Durante um longo período outras bibliotecas foram surgindo, sobretudo na Grécia. No ocidente, as primeiras bibliotecas surgiram durante as conquistas romanas. Ao contrário das bibliotecas gregas, no ocidente os leitores tinham acesso direto aos pergaminhos, que estavam organizados e expostos em prateleiras e separados consoante a língua (grego e latim). Com a probabilidade dos pergaminhos se estragarem e com a exportação para outros países do mundo, foi necessário proceder à cópia destes, mantendo os originais em locais de acesso restrito.

Com o crescimento no número de obras disponíveis, no início do século VI, durante a dinastia *Han*, foi criado o primeiro sistema de classificação e notação de obras, um catálogo, que permitia a pesquisa de livros de forma mais rápida (Zurndorfer 1995).

Durante a Idade Média, e séculos posteriores, sobretudo nos séculos XVII e XVIII, houve apenas evoluções no número e na forma de organização das bibliotecas, uma vez que o número de obras a armazenar crescera. Contudo, foi no ocidente que essas alterações foram mais visíveis, sobretudo pelo facto dos grandes eventos acontecerem nesses locais – por exemplo, Império Romano, expansão do Cristianismo e revoluções Francesas. Os manuscritos produzidos no ocidente tinham, em grande parte, origem grega, latina e bíblica, muitos dos quais ainda hoje existem. Diferentes tipos de bibliotecas foram surgindo nesses locais, desde bibliotecas privadas, públicas, académicas ou infantis, com características, obras e público-alvo diferentes, mas com a mesma essência – possibilitar a expansão e preservação do conhecimento (Staikos 2012).

1.1.3. Bibliotecas digitais

Com o aparecimento dos computadores e, sobretudo, coma expansão da *Internet*, a quantidade de informação digital disponível para consulta levou à evolução das bibliotecas, possibilitando o armazenamento de informação na forma digital. Estas bibliotecas ficaram conhecidas por "bibliotecas digitais". Foram muitos os impulsionadores no crescimento das bibliotecas digitais e características destas.

Segundo *Mary Brown* (Brown 2005) o conceito de tratamento de informação digital/ eletrónica surgiu nos finais dos anos 30, pois *H.G. Wells* pensava em encontrar uma forma de mudar o mundo das bibliotecas tradicionais, complementando-as com funcionalidades eletrónicas. *Vannevar Bush*, em 1945, aproveitando a ideia de *Wells*, desenhou a máquina *memex*, com o intuito de armazenar e extrair informação (livros, comunicações) de forma simples e rápida, pois, para extrair informação, apenas seria necessário introduzir o código correspondente. Uma ideia visionária uma vez que, apenas alguns anos mais tarde, em 1965, *Ted Nelson* construiu um modelo que permitia a criação e utilização de conteúdos digitais, cujo nome conhecido é hipertexto (Lesk 2005) (Candela, Castelli e Pagano 2011). Este conceito foi crescendo à medida que a tecnologia e a *Internet* também cresciam. Contudo, os anos 90 foram os grandes impulsionadores do conceito e utilização das bibliotecas digitais, e segundo *Wright* (Wright 2002), *Dan Greenstein* caracterizava como sendo a 2ª geração das bibliotecas digitais, onde o foco e os conceitos de tecnologias de digitalização, metadados, técnicas para a gestão de dados e a preservação (digital) ganharam destaque nos projetos académicos.

Contudo, encontrar uma definição para biblioteca digital pode ser uma tarefa difícil, porque são vários os pontos de vista, desde pessoas a organizações que, de alguma forma, estão envolvidas direta ou indiretamente com as bibliotecas, tradicionais ou digitais.

Por um lado, a comunidade das bibliotecas, ao longo dos anos utilizou diferentes termos, como "biblioteca eletrónica", "biblioteca computorizada", "biblioteca *online*", "biblioteca digitalizada", "biblioteca virtual", "biblioteca sem paredes", até chegar à definição atual - "bibliotecas digitais" (Chisenga 2003) (Bordinha 2002) (Cleveland 1998). Por outro, são muitas as propostas para uma definição de biblioteca digital, como por exemplo alguns programadores consideram uma coleção de algoritmos ou programas como sendo uma biblioteca digital, no entanto um editor diz que uma biblioteca digital é um catálogo *online*. Embora exista um consenso no termo a utilizar, a definição de biblioteca digital é vista de diferentes maneiras e é uma definição evolutiva.

De acordo com *Chisenga* (Chisenga 2003), a *IBM* (em 1994), no início dos anos 90 defendia um modelo hibrido, onde estavam incluídas as características das bibliotecas tradicionais acrescentadas às características das digitais, como uma representação de obras lidas por máquinas, ou seja, um conjunto de informação digital, ferramentas de armazenamento e *software* necessário para reproduzir, emular e ampliar as funcionalidades das bibliotecas tradicionais. Anos mais tarde, *Oppenheim* (em 1997), *Smithson* (em 1997), *Leiner* (em 1998) e *Arms* (em 2000) consideravam as bibliotecas digitais como sendo uma coleção de serviços e de informação disponíveis na rede em formato digital/ eletrónico. *Waters* (em 1998), seguindo a mesma lógica, defendia que as bibliotecas digitais eram organizações com os recursos necessários para a gestão e disponibilização a todo o tempo dos conteúdos digitais.

Enquanto *Witten* (Witten, Bainbridge e Nicols 2003) descreve as bibliotecas como estando focadas nas coleções de objetos digitais, como texto, áudio, imagem, vídeo, *Lesk* (Lesk 2005) aponta uma definição de equilibro entre o foco nas coleções dos objetos digitais e o foco nos serviços para a gestão destas.

Em 2006, *Seadle* (Seadle e Greifeneder 2007), com pontos de vista semelhantes a algumas definições anteriores, afirma que as bibliotecas digitais não substituem as bibliotecas tradicionais, mas são apenas uma atualização destas, assim como aconteceu na transformação de manuscritos em documentos impressos.

Adam (Adam e Yesha 2007) preocupa-se mais com a segurança e concentra-se apenas nos serviços para a criação e gestão da informação digital e no movimento dessa informação através das redes globais.

Embora existam pontos de vista diferentes sobre os conceitos e objetivos das bibliotecas digitais, possivelmente, definições relacionadas com a área que cada um desempenha, existem dois pontos centrais – o conteúdo digital e os serviços para gerir esses conteúdos.

1.1.4. Tradicional vs. Digital

Porque surgiram as bibliotecas digitais? Como referido anteriormente, a quantidade de informação a armazenar e as necessidades das pessoas, levaram a que se atualizasse as bibliotecas. Por exemplo, é mais rápido e cómodo encontrar artigos e livros relacionados com "gastronomia típica portuguesa na América" na *Internet,* que encontrar o mesmo livro ou artigo numa biblioteca tradicional. Para além desta vantagem, muitas outras foram descritas por vários autores, desde as tarefas realizadas pelos utilizadores até às tarefas de quem gere a biblioteca.

Com o aparecimento das bibliotecas digitais, os utilizadores podem aceder à informação em qualquer parte do mundo e a qualquer hora (Lesk 2005) (Rajashekar 2006) (Wright 2002), de forma simples (Cleveland 1998) e com poucas burocracias (Chisenga 2003). Ao contrário das tradicionais, nas bibliotecas digitais o acesso a uma obra pode ser efetuado por vários utilizadores sem que esta se danifique (Lesk 2005). As obras podem ser pesquisadas por diversos filtros (palavra, frase, titulo, autor ou data de publicação) o que torna mais fácil a procura de uma obra desejada (Lesk 2005) (Rajashekar 2006). Para os gestores da biblioteca, muitas são os benefícios na utilização de bibliotecas digitais. Uma obra armazenada numa biblioteca tradicional ocupa mais espaço físico que uma obra numa biblioteca digital, uma vez que as bibliotecas digitais necessitam de pouco espaço físico para armazenar grandes espaços de informação. O acesso a obras relacionadas é mais fácil numa biblioteca digital, pois cada obra permite que existam ligações para diferentes relacionamentos. As cópias das obras em formato digital ocorrem sem erros e podem ser atualizadas para formatos digitais diferentes, sem perda da sua essência (Lesk 2005) (Chisenga 2003) (Arms 2001). Com a passagem do analógico ao digital, é possível melhorar a qualidade de algumas obras, como a remoção de manchas e nódoas impostas pelo tempo, fazer a descoloração, modificar as imagens para que a obra fique mais legível (Gertz 2000) (Chisenga 2003) ou mesmo fazer a tradução das obras.

Todos estes fatores permitem reduzir os custos de manutenção necessários nas bibliotecas tradicionais (Warr e Hangsing 2009) (Lesk 2005), nomeadamente os custos relacionados com a preservação das obras, equipa de gestão/manutenção ou custos inerentes à produção de cópias.

Contudo, existem desvantagens na utilização das bibliotecas digitais. Como curiosidade, segundo *Lesk* (Lesk 2005), *Arthur Samuel* em 1964 diz-se que em 1984 as bibliotecas tradicionais, exceto os museus, iriam desaparecer. Por muitos fatores, esta previsão de *Samuel* não aconteceu. O primeiro fator está relacionado com a passagem das obras do formato tradicional ao formato digital. Para a construção das máquinas que permitiriam a transformação de 100 milhões de livros, *Lesk* previa que fossem necessários cerca de 800 milhões de euros, acrescido das compensações dos direitos de autor (Lesk 2005), tornando-se num processo dispendioso e moroso (Rajashekar 2006). Outro fator defendido por *Lesk* está relacionado com as preferências dos utilizadores. Um leitor, normalmente, prefere um livro em formato papel a um digital, uma tendência que tem decrescido com o avanço da tecnologia digital. Isto, porque na compra do livro poderemos folhear e ler algumas partes para verificarmos se desejamos o livro, o que não acontece em muitas bibliotecas digitais (Visly 2001). A leitura de uma obra em formato digital – exemplo dos leitores de *ebooks* como o *Amazon Kindle* ou o *Sony Reader*, como é necessário a utilização de um monitor, é mais cansativa, embora já existam progressos ao nível da redução de cansaço provocado pelos monitores (Samsung 2012). Os controlos de acesso e gestão dos direitos de autor são outros problemas que colocam um impasse à utilização de obras digitais, pois é mais fácil copiar, replicar e distribuir de forma indevida no mundo digital que no tradicional, mesmo que existam ferramentas de controlo muitas vezes o preço não é acessível. Nestes casos é necessária uma avaliação do custo-benefício da utilização dessas ferramentas. Acrescentando a estes problemas, a compatibilidade entre ferramentas para a gestão das obras digitais é baixa, tornando difícil a troca de informação entre as diferentes ferramentas.

Para colmatar este problema, e depois de conhecida a evolução das bibliotecas digitais e, sobretudo, das suas principais características e diferenças em relação às bibliotecas anteriores – as bibliotecas tradicionais, serão analisadas algumas das ferramentas mais utilizadas para gerir o conteúdo das bibliotecas digitais.

2. Repositórios digitais

2.1. Requisitos

As ferramentas que permitem gerir os conteúdos digitais designam-se por repositórios digitais.

O significado do termo "repositório digital" é muitas vezes debatido pela comunidade científica. Por um lado, muitos autores e investigadores consideram um repositório digital como uma ferramenta/ conjunto de serviços para armazenamento, organização e gestão de acessos aos conteúdos digitais (Semple 2006) (JISC 2005)(Technology 2007) (Infokit 2012). Por outro, o termo é também frequentemente referido como "repositório institucional" ou "arquivo digital", ou seja, como uma coleção digital que armazena, preserva e gere informação produzida por comunidades académicas ou como um local para armazenamento de informação digital. (Marques e Maio 2007) (Green e Bowie 2010) (Lee, Libraries e Foundation 1997) (Peterson 2011).

Apesar das divergências, nesta dissertação optou-se por utilizar o termo "repositório digital" como uma ferramenta utilizada na gestão dos conteúdos digitais.

As funcionalidades de um repositório digital foram-se complementando ao longo dos tempos e de acordo com diferentes pontos de vista: ponto de vista dos utilizadores, da comunidade de investigação/científica, da economia de mercado e da comunidade das bibliotecas tradicionais. Os utilizadores das bibliotecas tradicionais defendem que, em primeiro lugar, é necessário resolver os problemas das bibliotecas tradicionais, como a informação desatualizada e deterioração das obras. Enquanto a comunidade científica propõe inovação e implementação de novas funcionalidades, aproveitando os recursos da *Internet* de forma a tornar as bibliotecas mais acessíveis, a economia de mercado defende uma posição mais de marketing, ou seja, defendem que a principal tarefa das bibliotecas digitais é a oferta de serviços atraentes aos setores privados ou públicos, à comunidade académica ou à indústria, de forma a extrair informação importante dos documentos. A comunidade das bibliotecas tradicionais defende uma melhor disponibilização da informação (facilidade de acesso, aluguer, informação com mais qualidade) e uma melhor estrutura da organização de forma a adaptá-la aos novos desafios (Shustitskiy 2004).

Com base nestes novos desafios, a comunidade de investigadores – foco nos conteúdos colecionados e organizados de forma a corresponder às necessidades dos utilizadores – e a comunidade de bibliotecas tradicionais – foco nas instituições que oferecem a informação e as estruturas existentes de forma a adaptá-las aos novos desafios – construíram um conjunto de objetivos e características necessárias ao desenvolvimento de um repositório digital. Em primeiro lugar é necessário analisar os recursos humanos – para desenvolvimento, gestão de *hardware*, da base de dados e da informação – os recursos financeiros – para sustentar o desenvolvimento e a manutenção da biblioteca digital e da informação que esta possui – e os recursos estruturais – relacionados com os equipamentos necessários para a disponibilidade da informação – tendo em vista um possível crescimento/expansão da biblioteca (Chisenga 2003). Para além destes recursos, necessários antes, durante e depois do desenvolvimento da biblioteca digital, são necessários garantir outros requisitos, que auxiliem/ facilitem os utilizadores e gestores da biblioteca. Um dos requisitos mais importantes é a preservação. As bibliotecas tradicionais tentavam de alguma forma preservar as obras, utilizando sistemas antifogo, anti-roubo, ambientes controlados ou materiais mais duradouros, no entanto, o tempo levava ao desgaste de muitas dessas obras. Com

o aparecimento das bibliotecas digitais, e sobretudo com a tecnologia de digitalização de livros, de artigos, de jornais, de revistas, de fotografias, de desenhos, de microfilmes, alguns dos problemas da preservação foram resolvidos. No entanto, como a tecnologia digital está em constante inovação é necessário proceder às atualizações das obras por forma a suportar os novos formatos de multimédia, novos sistemas operativos, *hardware*, aplicações, etc. Com a preservação é possível acrescentar valor às obras no presente e nas gerações futuras. Como curiosidade em relação à importância da preservação, em 1998, estimava-se que existiam 80 milhões de livros, dos quais cerca de 10 milhões eram únicos (Hedstrom 1998). Muitos desses livros poderão ter informação importante para a sociedade. Se não houver forma de os preservar, alguns dos segredos e história dos nossos antepassados serão perdidos. Outros fatores importantes e necessários antes da preservação são a aquisição e arquivo dos conteúdos. Sem estes fatores não existem bibliotecas, pois são eles que permitem enriquece-las. A aquisição de conteúdos pode ser efetuada pela digitalização das obras, pela interoperabilidade com outras bibliotecas, disponibilização por parte dos utilizadores, etc. Nas bibliotecas tradicionais, o responsável pela aquisição desses conteúdos era o dono da biblioteca, nas digitais, poder-se-á utilizar ferramentas que permitam efetuar a recolha de obras que estejam armazenadas noutras bibliotecas ou permitir aos utilizadores a inserção de obras. Para a disponibilidade da informação é necessário ter em consideração, alguns requisitos que melhoram o acesso aos conteúdos. O primeiro requisito é a pesquisa – uma biblioteca digital deverá permitir aos utilizadores aceder aos conteúdos através da pesquisa, que para obter melhores resultados é necessário que exista tecnologia para indexar e extrair informação com qualidade (utilizando, por exemplo, pesquisa "*full-text*", pesquisa semântica, multilíngua, indexação, catalogação). Outro fator importante é o controlo de acesso – quanto mais confiável uma biblioteca digital, mais procurada ela é. Existem alguns pontos que poderão contribuir para a proteção dos conteúdos, como as licenças de utilização, atribuição de regras e políticas de acessos, controlo dos direitos dos autores, segurança dos conteúdos mais vulneráveis ou confidenciais, políticas de integridade (Cleveland 1998) (Hedstrom 1998) (Reddy 1999) (Ager 1999) (Wright 2002) (Bordinha 2002) (Warr e Hangsing 2009).

Em suma, as bibliotecas digitais podem disponibilizar obras, obtidas no processo de digitalização, criadas pelos utilizadores ou provenientes de outras bibliotecas. Estas, para aumentar o número de utilizadores/acessos, devem possuir um conjunto de serviços para indexação e pesquisa, controlos de acesso, áreas para a gestão de conteúdos e de utilizadores e áreas personalizáveis. É necessário que estes serviços suportem os mais variados tipos, tamanhos e formatos para armazenamento da informação.

2.2. Repositórios Digitais

Criar um novo repositório digital, nem sempre é a melhor solução, por diversas razões: custos e tempo de desenvolvimento e manutenção – recursos humanos, *software* e *hardware* para garantirem disponibilidade, escalabilidade, eficiência, segurança, preservação da informação, recursos para interoperabilidade entre diferentes repositórios, recursos para a proteção dos direitos de autor (Reddy 1999) (Lesk 2005) (Rajashekar 2006); já existem vários repositórios no mercado – livres e pagos – que suportam diversos formatos (gráficos, áudio e vídeo dinâmicos)

(Shustitskiy 2004), acrescentam funcionalidades de partilha de conhecimento (redes sociais) (Kruk, et al. 2007), funcionalidades de importação e exportação de conteúdos e múltiplos acessos (Warr e Hangsing 2009) e por estas razões a avaliação custo-benefício entre um novo repositório e os existentes é complexa. Contudo, e como o objetivo desta dissertação não é a criação de um repositório, mas a construção de uma API de serviços para a interoperabilidade/ migração de informação entre diferentes repositórios, é necessário analisar algumas características técnicas (arquitetura e estrutura dos dados) e funcionais (metadados, formatos dos conteúdos, tecnologia de interoperabilidade suportados, pesquisa e indexação, suporte de versões, gestão e proteção de acessos aos conteúdos) dos repositórios existentes no mercado, de forma a aproximá-los. Desses repositórios, apenas alguns serão analisados com mais detalhe, pelo facto de serem os mais utilizados e os mais conhecidos (que contêm mais referências nos documentos que constam nesta bibliografia). Alguns desses exemplos são o *DSpace, EPrints, FEDORA, Greenstone, Digital Commons, IntraLibrary, CONTENTdm, JeromeDL* e *DuraCloud* (Krishnamurthy 2008) (Pirounakis e Nikolaidou 2009) (Ingram 2010) (Fojtu 2009) (Repositories Support Project 2011).

2.2.1. *DSpace*

O *DSpace* (DSpace 2012) é um repositório digital *open-source* desenvolvido pela *MIT Libraries* e *Hewlett-Packard Labs* (HP) com o intuito de gerir os recursos digitais de instituições académicas (escolas, universidades) e organizações sem fins lucrativos. Desenvolvido em Java e disponibilizado em 2002, o *DSpace* permite às organizações gerir e preservar informação digital da estrutura organizacional da entidade, como departamentos, laboratórios e equipas.

Características Técnicas

A arquitetura do *DSpace* é bastante simples, pois é uma arquitetura de apenas 3 camadas – Aplicação, Modelo de Negócio e Armazenamento. A camada aplicacional é responsável por transmitir informação aos utilizadores – como resultados de pesquisa - ou permitir que os administradores do repositório possam gerir os conteúdos digitais. Na camada seguinte encontram-se os serviços para a pesquisa, indexação, administração, controlo de acessos e interoperabilidade, entre outros. Por fim, a camada de armazenamento, como o nome indica, permite guardar a informação digital em base de dados – *PostgreSQL* ou *Oracle* , no sistema de ficheiros ou indexá-la utilizando, por exemplo, o *Jakarta Lucene API* (Lucene 2012).

A informação armazenada na base de dados é organizada por *Communities* e *Sub-Communities* – que representam, por exemplo, um departamento, um laboratório ou outra divisão da organização. O número e nível de profundidade das comunidades, dependerá da estrutura da organização, no entanto, uma Sub-Comunidade apenas pertencerá a uma Comunidade. Cada *community* ou *sub-community* contem coleções (*Collection*) de objetos (*Item*). Esta entidade (*Item*) contem os metadados e os conteúdos digitais que representam um recurso (obra) digital.

Características funcionais

Cada conteúdo guardado no *Item* possui um identificador único baseado no sistema *Handle* da *CNRI – Corporation for National Research Initiatives* (CNRI 2012), que permite apresentar o recurso digital numa página *web* única, tornando o sistema mais fácil de utilizar e mais intuitivo.

O *DSpace* suporta diferentes formatos de metadados, mas apenas os metadados *Dublin Core* (DublinCore 2012) são armazenados no repositório e consequentemente na base de dados, pois os restantes são armazenados como *bitstream* no sistema de ficheiros, assim como o conteúdo digital, que poderá estar em diversos formatos, como PDF, formatos da *MS Word*, *XML*, *HTML*, *SGML*, *TIFF*, *MPEG*, *JPEG* e *GIF*. Este repositório, como referido anteriormente, suporta relações entre os conteúdos, cuja informação (nome e descição) é armazenada na entidade *Collection* de cada *Community,* enquanto que os tipos de relações são guardados na entidade *Item*. Outras características do *DSpace* são o suporte multilíngua, tanto a nível de metadados, como do conteúdo digital e o suporte a diferentes versões dos conteúdos. Ao nível de segurança, o *DSpace* possui um conjunto de operações para controlar os acessos aos conteúdos, como a autenticação dos utilizadores – através do sistema de *passwords*, utilizando certificados *X509* ou através de *LDAP* – e as permissões de leitura/escrita e adição/remoção de conteúdos digitais atribuidas a utilizadores e grupos de utilizadores.

Para adicionar conteúdo ao repositório (incluindo metadados), os administradores poderão utilizar a interface *web* ou um ficheiro *batch*, cuja informação será indexada – os metadados DC serão indexados na base de dados, enquanto que os restantes serão processados pelo *Jakarta Lucene API* – e armazenada, como referido, na base de dados e no sistema de ficheiros, para posteriormente ser pesquisável, utilizando a interface *web* e partilhada com outros repositórios digitais, utilizando os protocolos de interoperabilidade *OAI-PMH, OAI-ORE*, *SWORD*. Outras formas de disponibilizar alguma informação a outros repositórios ou aplicações é a utilização dos *feeds Atom* e *RSS* (Jorum 2005) (Patil e Kanamadi 2008) (Pirounakis e Nikolaidou 2009) (Candela, Castelli e Pagano 2011) (Madalli, Barve e Amin 2012).

2.2.2. *EPrints*

Desenvolvido em *Perl* na Universidade de *Southampton*, em 2000, e considerado o repositório mais rápido e fácil na gestão de conteúdos, o *EPrints* (EPrints 2012), também *open-source*, tem como principal objetivo o armazenamento de informação cientifica, como teses, revistas, relatórios, documentos de investigação ou publicações de conferências.

Características técnicas

Os conteúdos digitais e os metadados relacionados com um recurso digital são armazenados numa entidade (*data object*), que possui um identificador único – *Uniform Resource Identifier* (URI), para cada documento, permitindo também que o utilizador introduza outros identificadores do recurso digital. Ao contrário do que acontece com outros repositórios, no *EPrints* não existe a noção de relações, exceto se a localização dos recursos digitais relacionados estiver implicito nos metadados do recurso.

Características funcionais

Os diferentes formatos de metadados são armazeados e indexados numa base de dados *MySQL*, enquanto os conteúdos digitais são armazenados no sistema de ficheiros, para posterior pesquisa e gestão utilizando a interface *web* e/ou linha de comandos e após a inserção de credenciais (*username+password*) válidas. O serviço de pesquisa do *EPrints*, suporta pesquisa *full-*

text, para os campos previamente indicados e independentemente da língua, uma vez que o *EPrints* suporta multilíngua, com a particularidade da informação da língua ficar armazenada como atributo, no ficheiro XML. À semelhança do *DSpace*, o *EPrints* suporte os mais variados formatos de ficheiros existentes, como por exemplo, *PDF*, *DOC*(X), *JPEG*, *GIF*, *SGML*.

Para além destes serviços – gestão e pesquisa – o *EPrints* permite importar conteúdos nos formatos *METS*, *MODS*, *EndNote* e permite a interoperabilidade com outros repositórios através de *OAI-PMH*, *OAI-ORE* ou *SWORD* ou disponibilizar alguma informação utilizando os *feeds RSS* e *Atom* (Jorum 2005) (Patil e Kanamadi 2008) (Pirounakis e Nikolaidou 2009) (Madalli, Barve e Amin 2012).

2.2.3. *FEDORA*

Em 1997, o grupo de investigação de bibliotecas digitais da Universidade de *Cornell* e mais tarde a biblioteca da Universidade de Virgínia, desenvolveram um repositório em *Java,* designado por *FEDORA* (*Flexible Extensible Digital Object Repository Architecture*) (Fedora 2012). À semelhança dos repositórios mencionados anteriormente, o *FEDORA* é um repositório *open-source* com o objetivo de permitir a gestão de conteúdos de diferentes áreas – educacionais, empresariais, económico-financeiras, etc.

Características técnicas

Com uma arquitetura modular e orientada para os princípios de interoperabilidade e extensibilidade, o *FEDORA* é adaptável às diferentes áreas científicas e facilmente adaptável às necessidades do mercado.

O *FEDORA* é constituído, por objetos digitais (*digital object*) que possui uma estrutura em *XML* específica (*FOXML – Fedora Object XML*) para identificar e organizar os conteúdos digitais – versões e respetivas datas, informação de *log*. Cada objeto digital é identificado por uma URI (por exemplo – *educa:1*) e contem um conjunto de outros objetos – *datastream* (com identificador único no contexto do objeto digital) – que representam a informação do recurso digital, como os metadados, as relações, miniaturas, etc. No caso das relações e dos metadados *Dublin Core*, existem *datastreams* específicos para armazenar essa informação, por exemplo, o *datastream* com o identificador "*DC*" armazena metadados *Dublin Core,* enquanto as relações externas (com outros objetos digitais) ou internas (entre *datastreams*) são armazenados, respetivamente, nos *datastreams RELS-EXT* e *RELS-INT*. Estes (*RELS-EXT*/*INT*) possuem uma ontologia simples baseada no RDF. Estas relações são indexadas utilizando um serviço do *FEDORA* designado por *FEDORA Resource Index*, que utiliza o *Lucene* para a indexação. A restante informação é armazenada em base de dados relacionais (*MySQL, Oracle, PostgreSQL*) e pesquisáveis utilizando um serviço designado por *gSearch* (*generic search*) que possui serviços de indexação e pesquisa utilizando o *Lucene, o Solr* ou o *Zebra Search*.

Características funcionais

O *FEDORA* possui diversas formas de acesso aos conteúdos digitais, como interface *web*, aplicação *desktop*, utilizando um ficheiro *batch*, ou através de *webservices* (acessíveis via *SOAP* ou *REST*). Apenas os administradores poderão criar, eliminar, alterar os conteúdos digitais, os

restantes utilizadores apenas podem pesquisar e visualizar os conteúdos digitais. Para isso, os utilizadores terão de autenticar-se utilizando uma *password,* através de *LDAP* ou autenticação com o endereço IP. Para além disso, o acesso a um objeto digital especifico é controlado por políticas *XACML*.

O conteúdo de um *datastream* pode ser armazenado de 3 formas: através de uma ligação externa (*URL*), no sistema de ficheiros ou no próprio *datastream* em formato *XML* ou equivalente. Para além disso o *FEDORA* poderá controlar a integridade dos conteúdos através dos algoritmos *MD5*, *SHA-1*, *SHA-256*, *SHA-384* e *SHA-512*, ao contrário dos restantes que apenas suportavam o *MD5*. Em relação à interoperabilidade, o *FEDORA* possui serviços que permitem disponibilizar os conteúdos digitais utilizando os protocolos *OAI-PMH*, *OAI-ORE*, *SWORD* ou exportá-los nos formatos *METS-XML e RDF*.

Por fim, à semelhança dos restantes, o *FEDORA* também suporta diversas línguas e diferentes formatos (*pdf*, *txt*, *jpg*, *gif*, *mpeg*, *avi*, *áudio*) (Jorum 2005) (Pirounakis e Nikolaidou 2009) (Madalli, Barve e Amin 2012).

2.2.4. *Greenstone*

O *Greenstone* (Greenstone 2012), repositório *open-source*, foi desenvolvido em 2000 pela universidade de *Waikato* em cooperação com a *UNESCO* e a *Human Info NGO* para organizar e publicar informação de universidades, bibliotecas ou outros serviços públicos.

Características técnicas

A entidade principal do *Greenstone* é *document*, que contem um identificador único e cujo conteúdo está no formato *XML*. Os documentos (*documents*) podem estar ligados a diversos recursos, que representam o conteúdo digital. Como nos repositórios anteriores, o *Greenstone* suporta relações, definidas como uma lista de características associadas a uma operação – indexação, pesquisa, importação de conteúdo digital. O *Greenstone* possui serviços para interoperabilidade utilizando os protocolos *OAI-PMH* e *Z39.50*.

Características funcionais

No *Greenstone* existem 3 grupos de utilizadores: administradores – têm permissões para adicionar ou remover outros utilizadores; criador de conteúdos – utilizador com permissões para criar e atualizar as relações entre os documentos; e utilizadores finais – com permissões de visualização dos documentos armazenados. O acesso a cada um dos componentes poderá ser efetuado através da linha de comandos ou aplicação *desktop*, para administradores e criadores de conteúdos e interface *web* para os utilizadores finais. O serviço de pesquisa permite pesquisar termos num documento ou apenas em partes do documento, pois todo o texto do documento é indexado por secções, definidas com *tags XML – XLinks*. Também, alguns campos dos metadados são indexados de forma a tornar a pesquisa mais simples. Tanto os metadados como os conteúdos são armazenados no sistema de ficheiros, os metadados são armazenados num ficheiro *XML*. Para além dos metadados *Dublin Core*, e como o *Greenstone* foi desenvolvido em ambiente académico e financiado por instituições da Nova Zelândia, o *Greenstone* suporta outros formatos de metadados, tais como, *New Zealand Government Locator Service Metadata*

Standard, Australian Government Locator Service. Existe um *plug-in* – *Greenstone's Metadata Set Editor* – que permite adicionar outros formatos de metadados, que estejam nos formatos *XML*, *METS*, *OAI*, *MARC*, etc. Como os restantes repositórios, o *Greenstone* suporta múltiplas línguas (Pirounakis e Nikolaidou 2009) (Madalli, Barve e Amin 2012).

2.2.5. *Digital Commons*

O *Digital Commons* (DigitalCommons 2012) é um repositório desenvolvido pela *BePress* e disponibilizado em 2002 para ambientes institucionais, de forma a permitir o armazenamento de artigos de conferências, teses, dissertações, trabalhos de investigação, anúncios de reuniões, etc. Ao contrário dos referidos anteriormente, o *Digital Commons* é um repositório com fins comerciais, desenvolvido em *Perl* e utilizado em ambientes *Linux*.

Características técnicas

Ao contrário dos repositórios mencionados anteriormente, o *Digital Commons* é um repositório *hosting service*, ou seja, o repositório digital não é instalado nas nossas máquinas, o que permite à organização poupar nas ferramentas necessárias ao alojamento dos recursos digitais dos utilizadores.

Características funcionais

A adição de novos conteúdos é possível utilizando uma interface *web* ou um ficheiro *batch*, controlados por diferentes permissões de acessos – editores, administradores ou utilizadores. Após a submissão, o sistema gera uma chave única – *URL*, que torna o seu acesso simples, via *web*. Para além do serviço de submissão, o *Digital Commons*, possui serviços para indexação, compatível com motores de pesquisa como o *Google* ou o *Bing*, que torna o serviço de pesquisa mais dinâmico. À semelhança dos restantes, o *Digital Commons* suporta multilíngua e múltiplos formatos de ficheiros – texto, imagem, áudio, vídeo, e possui serviços que permitem converter documentos *Word* ou *RTF* para *PDF* e aceder a vídeos no *youtube* ou no *vimeo*. Permite ainda a partilha dos conteúdos através do protocolo *OAI-PMH* ou através de *RSS*.

O *Digital Commons* apenas suporta metadados *Dublin Core* que são armazenados numa base de dados *PostgreSQL* (Ingram 2010) (Bepress 2012).

2.2.6. *IntraLibrary*

IntraLibrary (Intrallect, IntraLibrary 2012) é um repositório desenvolvido pela *Intrallect* (uma *spin-off* da Universidade de *Edinburgh*) e disponibilizado em 2002, com diferentes propósitos: educacionais – armazenamento e gestão de conteúdos relacionados com aulas, aprendizagem; académicos – armazenamento e gestão de testes, dissertações, teses, relatórios de investigação; e empresarial – gestão de recursos por departamentos, procedimentos internos, resultados de negócios.

Características técnicas

O *IntraLibrary* possui diferentes plataformas de acesso aos conteúdos com diferentes objetivos: *intraLibrary* – gestão, pesquisa e partilha de conteúdos digitais, *intraLibrary Connect* – conjunto de serviços *web* de suporte ao *intraLibrary* e *intraLibrary Host* – serviço de *hosting*. Desenvolvido em Java, o *intraLibrary* opera em diferentes sistemas operativos e plataformas (*cloud* e *hosted service*) e armazena a informação dos conteúdos – como os metadados, comentários dos utilizadores – numa base de dados *MySQL*.

Características funcionais

Como alguns dos repositórios analisados anteriormente, o *intraLibrary* possui um conjunto de serviços que permite gerir, pesquisar e partilhar os conteúdos digitais armazenados no repositório. Possui funcionalidades para a troca de informação entre os utilizadores, como comentários, *tags*, classificação, etc. Cada conteúdo armazenado possui um identificador único que é facilmente acessível via *web*, pois baseia-se em *URLs*.

A partilha das relações e dos conteúdos digitais com outros repositórios poderá ser efetuado utilizando o protocolo *OAI-PMH*, *SWORD*, *Z39.87* de metadados *Dublin Core, Learning Object Model* e *Open Digital Rights Language*. Este último grupo de metadados permite controlar os direitos de autor (Ingram 2010) (Intrallect, FAQS intraLibrary 2012)**.**

2.2.7. *CONTENTdm*

O *CONTENTdm* (ContentDM 2012) – *Content Digital Management* – surgiu na Universidade de *Washington* e após alguns anos de incubação, foi publicado, em 2001. Escrito em PHP e compatível com os sistemas operativos *Unix, Linux* e *MS Windows*, é o repositório comercial bastante simples, flexibilidade e segurança (Rosensweig 2008). Quando o *CONTENTdm* foi criado tinha como principais objetivos: a compatibilidade com ambientes *web*, uma pesquisa eficiente e usável, a interoperabilidade entre repositórios e a escalabilidade.

Características técnicas

O *CONTENTdm* é dividido em 3 módulos: *CONTENTdm server*, responsável pelo armazenamento dos conteúdos e imagens, com suporte a diferentes formatos; *CONTENTdm website*, permite aos administradores gerir os conteúdos e as coleções; e *CONTENTdm Project Client*, oferece um conjunto de serviços, como pesquisa, visualização, *download*, aos utilizadores finais. Além destes serviços base, os utilizadores poderão consultar apenas as suas coleções (assinaladas como favoritas, comentadas ou criadas pelo respetivo utilizador) e comentar e partilhar os conteúdos digitais através de redes sociais ou correio eletrónico. A entidade principal no *CONTENTdm* é a coleção – *Digital Collection*. Nesta entidade, é armazenada a informação relativa ao recurso digital – metadados (*Dublin Core, VRA Core*), relações e conteúdos (imagens, vídeos, texto, áudio). Toda esta informação é armazenada no sistema de ficheiros em XML (metadados e relações), no formato de origem do conteúdo ou em outro formato convertido, por exemplo o conteúdo poderá ser armazenado como texto, pois o *CONTENTdm* suporta ferramentas de *OCR* para essa conversão.

Características funcionais

Como referido anteriormente, os utilizadores do *CONTENTdm* poderão pesquisar, visualizar e partilhas os conteúdos digitais, no entanto os administradores poderão restringir essas ações. Estes poderão configurar o sistema para que o acesso a determinados conteúdos/ metadados seja limitado por autenticação/registo, *LDAP*, a utilizadores específicos, etc. Além disso, o *CONTENTdm* possui serviços que permitem identificar os autores e direitos de autores dos conteúdos digitais.

À semelhança dos restantes, o *CONTENTdm* suporta multilíngua e diferentes tipos de formatos. Para além disso, para cada conteúdo é criado uma URL única permitindo um acesso simples ao respetivo conteúdo. Na partilha dos conteúdos o *CONTENTdm* utiliza os protocolos *OAI-PMH*, *Z39.50*, o *feed RSS* ou exportar no formato *METS* (Ingram 2010) (CONTENTdm 2012).

2.2.8. *JeromeDL*

As principais funcionalidades dos primeiros repositórios digitais eram a criação e gestão de recursos digitais, incluindo o armazenamento e a pesquisa. Com o aparecimento das tecnologias relacionadas com a Web Semântica esse paradigma mudou, uma vez que com a pesquisa semântica podem-se obter melhores resultados – recursos relacionados, pesquisas por um determinado autor, pesquisa por termos semanticamente semelhantes. Por exemplo, se pesquisarmos por "melhores restaurantes em Aveiro", um serviço de pesquisa semântica "deduz" que estamos à procura dos melhores restaurantes em Aveiro e não à procura de documentos que contenham, simultaneamente, os termos "melhores", "restaurantes" e "Aveiro", como acontece com a pesquisa por palavras-chave. Um desses serviços de pesquisa é o Google (http://www.google.pt). Utilizando o termo de pesquisa "melhores restaurantes em Aveiro", o serviço devolve informação de restaurantes em Aveiro (como morada, localização no mapa, contactos e classificação dos utilizadores – introduzidas em páginas *web*, *blogs*, entre outros).

Contudo, as bibliotecas digitais, incluindo as bibliotecas com tecnologia semântica, foram construídas para melhorar algumas funcionalidades das bibliotecas tradicionais, como a organização dos conteúdos, a preservação ao longo do tempo, a facilidade de obter os conteúdos desejados e diminuir o espaço físico para o armazenamento dos conteúdos, ou seja, as bibliotecas digitais continuaram a ser orientadas para os gestores das bibliotecas, que partilham apenas informação e não o conhecimento. Com isto um novo paradigma surgiu – *Social Semantic Digital Libraries* – que junta as bibliotecas digitais, a *web semântica* e as redes sociais, onde é possível partilhar conhecimento. Uma das soluções possíveis para envolver os utilizadores – detentores do conhecimento – são as anotações. As anotações digitais, descritas por *Kruk et al.* (Kruk, et al. 2007), permitem aos utilizadores anexar ou visualizar notas (resumos ou outra informação) nos conteúdos. Outra forma de envolver os utilizadores é a partilha de conhecimento dentro de uma comunidade de utilizadores, permitindo que exista uma melhor comunicação entre os utilizadores de um repositório digital na e através das comunidades de utilizadores. Uma dessas aproximações é o JeromeDL.

O *JeromeDL* (JeromeDL 2012) é um repositório digital que resulta da parceria da *DERI International* e da Universidade de Tecnologia de *Gdansk*. Desenvolvido em 2003 em Java, é um repositório *open-source* destinado às instituições que pretendam publicar documentos via *web*.

Características Técnicas

O *JeromeDL*, à semelhança de outros repositórios, possui serviços para a gestão e pesquisa dos conteúdos digitais. No entanto, como é um repositório com tecnologia semântica, como o FEDORA, permite a pesquisa e armazenamento das relações semânticas dos conteúdos em base de dados de tripletos (*Sesame RDF*).Permite ainda o armazenamento e indexação (utilizado o *Lucene* (Lucene 2012)) de diferentes formatos de conteúdos (*PDF*, *RDF*, *SWF*, imagens, vídeos, *links* para conteúdos externos), diferentes tipos de informação (bibliografia, autor e outros metadados), a conversão entre formatos (*PDF* para *PS*, *TXT*, *RTF*, coleções de páginas para *PDF* ou *SWF* e vice-versa) e suporta conteúdos em diferentes línguas.

Características funcionais

O *JeromeDL* possui uma interface *web* para a gestão e pesquisa dos conteúdos armazenados e respetivos acessos, a utilizadores e grupos de utilizadores que são controlados através palavras-chave e de *ACLs – Access Control List* – permissões de impressão e cópia. Por sua vez, os administradores acedem ao sistema e submetem novos conteúdos, através de controlo remoto – protocolo *RMI* (*Remote Method Invocation*) – através de uma aplicação *desktop* designada por *JeromeAdmin*.

Para além das funcionalidades dos repositórios anteriores, o *JeromeDL* permite aos utilizadores introduzir comentários, anotações e classificações aos respetivos conteúdos e podem partilhar esses conteúdos na *Web* – redes sociais, comunidades de utilizadores. Também é possível a partilha e aquisição de conteúdos através de protocolos de interoperabilidade, nomeadamente, através do protocolo *OAI-PMH* (Krottmaier 2004) (Fox 2009) (JeromeDL 2012).

2.2.9. *DuraCloud*

O *DuraCloud* (DuraCloud, DuraCloud 2012) é uma plataforma *open-source*, desenvolvida em *Java,* para o armazenamento e preservação de conteúdos digitais em *Cloud Computing*. Embora não se enquadre na definição de repositório digital descrita anteriormente, contem um conjunto de funcionalidades semelhantes aos repositórios analisados e por essa razão será utilizado neste trabalho.

Com a utilização da tecnologia *Cloud*, o *DuraCloud* tem como objetivo aumentar a disponibilidade, a preservação e melhorar o acesso aos conteúdos digitais. Embora seja um serviço pago (DuraCloud, Pricing 2012), possui um conjunto alargado de destinatários, como pequenas e grandes instituições/ empresas, unidades de investigação, bibliotecas e organizações culturais, de um modo geral, organizações comerciais e não comerciais, de forma a acrescentar valor aos conteúdos partilhados.

A primeira versão disponibilizada ao público foi lançada em novembro de 2011.

Características técnicas

O *DuraCloud*, à semelhança dos restantes repositórios, possui um conjunto de serviços de armazenamento, pesquisa, indexação, controlo de acessos, partilha e transformação dos conteúdos digitais. No entanto, o principal foco é nos serviços de acesso e preservação, pois são estes os principais serviços que as tecnologias *Cloud* permitem melhorar. Em questões de

preservação, o *DuraCloud* permite cópias múltiplas em diferentes locais e permite que todos os conteúdos sejam exportados para uma máquina definida pelos administradores dos conteúdos.

Características funcionais

Neste serviço *hosted* é possível carregar diferentes formatos de dados, incluindo formatos de arquivo – *ZIP*, *TAR*, *AIP*, 7-*ZIP*, *RAR*, com um tamanho máximo de 5GB, no entanto, o *DuraCloud* recomenda o armazenamento de ficheiros com tamanho máximo de 1GB. Em relação aos metadados, o *DuraCloud* armazena qualquer tipo de metadados, uma vez que considera os metadados como uma lista de pares nome-valor, independentemente do *schema* utilizado. Em questões de segurança, no *DuraCloud* é possível enviar conteúdos protegidos ou através de canais de comunicação protegidos, e apenas é possível o envio de conteúdos utilizando as credenciais de acesso dos administradores dos conteúdos armazenados no repositório. Além disso, os conteúdos e os direitos de autor são da responsabilidade e controlados pelos administradores dos conteúdos.

O acesso aos conteúdos por parte dos utilizadores é possível utilizando plataforma *web* do repositório. No caso dos administradores, o acesso aos conteúdos ou o envio de novos conteúdos é possível de 3 formas diferentes: através da interface *web* destinada para esse fim, através de uma *API REST* ou através de uma ferramenta de sincronização instalada na máquina dos administradores (Waddington, et al. 2012) (DuraCloud, Welcome to the DuraCloud Wiki 2012).

2.3. Comparação entre repositórios

Como referido, existem muitos repositórios digitais para diferentes fins – comercial ou não comercial, académicos ou empresariais – no entanto, apenas uma pequena parte dos existentes foi analisada com algum detalhe. Para construir a *API* de serviços é necessário comparar as características dos repositórios digitais descritos anteriormente, a fim de obter um conjunto de características comuns e um mecanismo para adaptar as características diferentes.

Numa primeira fase, será efetuado um resumo das características de cada repositório. Posteriormente serão analisadas as características comuns e as características diferentes entre os diferentes repositórios. Por fim serão estudadas algumas formas de aproximar as características diferentes (Open Source Systems 2012).

2.3.1. Resumo

DSpace

Descrição

Ano	2002
Linguagem desenvolvimento	*Java*
Entidades envolvidas	*MIT Libraries e Hewlett-Packard Labs (HP)*
Tipo de Repositório	*Open-source*
Custo	-

Destinatários/ Objetivos	Instituições académicas e organizações sem fins lucrativos
Website	http://www.dspace.org/
Última versão	1.8

Armazenamento

Base de dados	*PostgreSQL* ou *Oracle*	Metadados *Dublin Core* Informação dos conteúdos
Sistema de Ficheiros		Restantes metadados Conteúdos digitais
Indexação	*Jakarta Lucene API* *Base de dados*	Restantes metadados Metadados *Dublin Core*

Outra informação

Tipo de Identificador	*CNRI Handle System*
Formatos de dados	Múltiplos
Formatos de metadados	Múltiplos
Relações	Sim
Multilíngua	Sim
Versões	Sim
Proteção de conteúdos	Palavra-passe, certificados *X509*, *LDAP*, restrições de leitura/escrita
Interoperabilidade/ partilha de informação	*OAI-PMH*, *OAI-ORE*, *Atom* (feed), *RSS*, *SWORD*
Acesso	Interface *web* (qualquer utilizador), *REST* ou ficheiro *batch* (administradores)

EPrints

Descrição

Ano	2000
Linguagem desenvolvimento	*Perl*
Entidades envolvidas	Universidade de *Southampton*
Tipo de Repositório	*Open-source*
Custo	-
Destinatários/ Objetivos	Armazenamento informação científica, teses, revistas, artigos de conferências
Website	http://www.eprints.org/
Última versão	3.3

Armazenamento

Base de dados	*MySQL*	Metadados
Sistema de Ficheiros		Conteúdos digitais
Indexação	Base de dados	Metadados

Outra informação

Tipo de Identificador	*Uniform Resource Identifier*, com possibilidade de acrescentar outros identificadores
Formatos de dados	Múltiplos
Formatos de metadados	Múltiplos
Relações	Não
Multilíngua	Sim (metadados)
Versões	-
Proteção de conteúdos	Palavra-passe
Interoperabilidade/ partilha de informação	*OAI-PMH, OAI-ORE, SWORD, RSS, Atom (feed),*
Acesso	Interface *web* e linha comandos

FEDORA

Descrição

Ano	1997
Linguagem desenvolvimento	*Java*
Entidades envolvidas	Universidade de *Cornell e* biblioteca da Universidade de Virgínia
Tipo de Repositório	*Open-source*
Custo	-
Destinatários/ Objetivos	Todos
Website	http://fedora-commons.org/
Última versão	3.6

Armazenamento

Base de dados	*MySQL, Oracle ou PostgreSQL*	Metadados Informação dos conteúdos
Sistema de Ficheiros		Conteúdos
Indexação	*Lucene ou Zebra Search*	Metadados

Outra informação

Tipo de Identificador	*Uniform Resource Identifier*
Formatos de dados	Múltiplos
Formatos de metadados	Múltiplos
Relações	Sim
Multilíngua	Sim (metadados)
Versões	Sim
Proteção de conteúdos	Politicas *XACML* Palavra-passe, através de LDAP ou autenticação com o endereço IP Integridade *MD5*, *SHA-1*, *SHA-256*, *SHA-384* e *SHA-512*
Interoperabilidade/ partilha de informação	*OAI-PMH, OAI-ORE, METS-XML, SWORD e RDF*
Acesso	Interface *web*, aplicação *desktop*, ficheiro *batch*, *webservices* (*SOAP* e *REST*)
Outros serviços	Pesquisa semântica

Greenstone

Descrição

Ano	2000
Linguagem desenvolvimento	-
Entidades envolvidas	Universidade de *Waikato*, *UNESCO* e *Human Info NGO*
Tipo de Repositório	*Open-source*
Custo	-
Destinatários/ Objetivos	Universidades, bibliotecas e outros serviços públicos
Website	http://www.greenstone.org/
Última versão	2.85

Armazenamento

Sistema de Ficheiros	Metadados Conteúdos digitais

Outra informação

Tipo de Identificador	-
Formatos de dados	Múltiplos
Formatos de metadados	Múltiplos
Relações	Sim
Multilíngua	Sim

Versões	Não
Proteção de conteúdos	Perfis com diferentes tipos de acesso
Interoperabilidade/ partilha de informação	*OAI-PMH e Z39.50*
Acesso	Linha de comandos, aplicação *desktop* (administradores e criadores dos conteúdos) e interface *web* (restantes utilizadores)

Digital Commons

Descrição

Ano	2002
Linguagem desenvolvimento	*Perl*
Entidades envolvidas	*BePress*
Tipo de Repositório	*Comercial e Hosting service*
Custo	Aprox. 1000 € / Ano
Destinatários/ Objetivos	Armazenamento teses, dissertações, investigação
Website	http://digitalcommons.bepress.com/
Última versão	-

Armazenamento

Base de dados	*PostgreSQL*	Metadados
Sistema de Ficheiros		Conteúdos digitais
Indexação		*Google, Bing*

Outra informação

Tipo de Identificador	*Uniform Resource Locator*
Formatos de dados	Múltiplos
Formatos de metadados	*Dublin Core*
Relações	-
Multilíngua	Sim
Versões	-
Proteção de conteúdos	*LDAP* e Palavra-passe
Interoperabilidade/ partilha de informação	OAI-PMH, RSS
Acesso	Interface web e ficheiro *batch*
Outros serviços	Conversão de conteúdos *Word* ou *RTF* para *PDF*, acesso a vídeos do *youtube* e *vimeo*

IntraLibrary

Descrição

Ano	2002
Linguagem desenvolvimento	*Java*
Entidades envolvidas	*Intrallect*
Tipo de Repositório	*Comercial*
Custo	-
Destinatários/ Objetivos	Escolas, universidades, empresas
Website	http://www.intrallect.com/
Última versão	3.3

Armazenamento

Base de dados	*MySQL*	Metadados, comentários
Sistema de Ficheiros		Conteúdos

Outra informação

Tipo de Identificador	Uniform Resource Locator
Formatos de dados	Múltiplos
Formatos de metadados	*Dublin Core, Learning Object Model* e *Open Digital Rights Language*
Relações	Sim
Multilíngua	Sim
Versões	-
Proteção de conteúdos	Acesso controlado Direitos de autor (metadados)
Interoperabilidade/ partilha de informação	*OAI-PMH, SWORD, Z39.87 e RSS*
Acesso	Interface *web*
Outros serviços	Partilha de conteúdos. Suporte de comentários, classificação, *tags*

CONTENTdm

Descrição

Ano	2001
Linguagem desenvolvimento	*PHP*
Entidades envolvidas	Universidade de *Washington*

Tipo de Repositório	*Comercial*
Custo	Depende do serviço a subscrever
Destinatários/ Objetivos	Melhorar a usabilidade e a interoperabilidade
Website	http://www.contentdm.org/
Última versão	6.1

Armazenamento

Servidores do *CONTENTdm*

Outra informação

Tipo de Identificador	-
Formatos de dados	Múltiplos
Formatos de metadados	*Dublin Core, VRA Core*
Relações	Sim
Multilíngua	Sim
Versões	-
Proteção de conteúdos	Palavra-passe *LDAP* Direitos de autor
Interoperabilidade/ partilha de informação	*OAI-PMH, Z39.50, METS, RSS*
Acesso	Interface *web*
Outros serviços	Serviço de comentários e partilha

JeromeDL

Descrição

Ano	2003
Linguagem desenvolvimento	*Java*
Entidades envolvidas	*DERI International* e da Universidade de Tecnologia de *Gdansk*
Tipo de Repositório	*Open-source*
Custo	-
Destinatários/ Objetivos	Instituições
Website	http://www.jeromedl.org/
Última versão	2.1

Armazenamento

Base de dados	*Sesame RDF*	Relações e informação pessoal
Indexação	*Lucene*	Conteúdos, metadados

Outra informação

Tipo de Identificador	-
Formatos de dados	Múltiplos
Formatos de metadados	Múltiplos
Relações	Sim
Multilíngua	Sim
Versões	-
Proteção de conteúdos	*ACL* (impressão e cópia) Acesso restrito consoante permissões dos utilizadores
Interoperabilidade/ partilha de informação	*OAI-PMH*
Acesso	Interface *web* (utilizadores) e *JeromeAdmin* (administradores)
Outros serviços	Conversão de formatos, anotações, comentários, classificação, pesquisa semântica, partilha de conteúdos nas redes sociais

DuraCloud

Descrição

Ano	2011
Linguagem desenvolvimento	*Java*
Entidades envolvidas	*DuraSpace*
Tipo de Repositório	*Open-source (hosting service)*
Custo	Depende do serviço a subscrever
Destinatários/ Objetivos	Todos
Website	http://www.duracloud.org/
Última versão	2.0

Armazenamento

Base de dados	*Cloud Computing*
Sistema de Ficheiros	
Indexação	

Outra informação

Tipo de Identificador	-
Formatos de dados	Múltiplos
Formatos de metadados	Múltiplos
Relações	Sim
Multilíngua	Sim

Versões	Sim
Proteção de conteúdos	Envio de conteúdos através de canais seguros Proteção com Palavra-passe Direitos de autor preservados
Interoperabilidade/ partilha de informação	-
Acesso	Interface *web* (utilizadores), *REST* e aplicação *desktop* (administradores)
Outros serviços	Aumento disponibilidade e tempo de preservação Exportação de conteúdos

2.4. Interoperabilidade

Antes da análise das características, é necessário compreender alguns conceitos utilizados anteriormente, nomeadamente os conceitos relacionados com a interoperabilidade entre os diferentes repositórios e os formatos de metadados (descritos no ponto 2.5).

Na interoperabilidade, existem diversos protocolos que permitem a troca de informação entre repositórios digitais, como o caso do *OAI-PMH, OAI-ORE, ATOM*, entre outros. Seguidamente serão apresentados alguns dos protocolos que permitem realizar essa tarefa.

O *OAI-PMH* (OIA-PMH 2012) – *Open Archives Initiative Protocol for Metadata Harvesting* – é um dos projetos da *OAI* – *Open Archives Initiative* – para a interoperabilidade entre repositórios digitais através da partilha de metadados. O protocolo *OAI-PMH* é composto por 2 entidades: *data providers* – entidade que contem a informação e implementam o protocolo *OAI-PMH* de forma a expor os seus metadados ao exterior – e os *service providers* – contem um conjunto de funcionalidades que permitem adquirir os metadados dos *data providers*. Contudo, é possível adicionar outras entidades intermédias – *aggregators* – que têm a responsabilidade de adquirir os metadados armazenados nos *data providers* e reenvia-los para os *service providers*. O *OAI-PMH* possui um conjunto de serviços que permitem expor os metadados dos objetos digitais, cuja informação é enviada por *HTTP* no formato *XML*. *Identify* retorna as principais informações do repositório; *ListMetadataFormats* lista os formatos de metadados implementados no repositório; *GetRecord* devolve os metadados de um determinado formato; *ListRecords* lista os registos do repositório; *ListIdentifiers* lista os identificadores de todos os registos do repositório; *ListSets* lista a estrutura do conjunto de um repositório. Todos eles têm parâmetros/ filtros de acordo com o que se pretende obter, por exemplo, para obter os registos com metadados *Dublin Core* que tenham sido inseridos numa determinada data, utiliza-se *http://www.educa.pt/oai/?verb=ListRecords&metadataPrefix=oai_dc&from=2008-03-20.*

OAI-ORE (OAI-ORE 2012) – *Open Archives Initiative - Object Reuse and Exchange* – também desenvolvido pela *OAI*, tem como objetivo o desenvolvimento de normas para a identificação, descrição e troca de agregações de recursos web, ou seja, o *OAI-ORE* pretende disponibilizar uma visualização/ um mapa dos conteúdos digitais existentes num repositório e respetivas agregações/ relações. Existem diversas formas de obter a informação das relações, mas primeiro

é necessário conhecer os objetos existentes – *Resource Map Discovery* – através dos *feeds* Atom, dos *Site Maps*, via *OAI-PMH* ou análise de determinadas páginas *web* (através de *links*). Depois de conhecer os objetos existentes é necessário analisar cada um, através da análise do conteúdo da página (que se encontra num dos formatos – *Atom, RDF, RDFa* ou *HTTP*), consegue-se obter as relações existentes entre os objetos digitais.

O *Z39.50* (NISO 2002) é um dos mais antigos protocolos de interoperabilidade entre repositórios digitais, pois surgiu nos anos 70. Tem como objetivo a pesquisa e aquisição de informação que existe numa base de dados remota, ou seja, possui uma filosofia cliente-servidor para a aquisição dessa informação. Este *standard* possui um conjunto de operações que permitem adquirir a informação remota, tais como: *Init* – inicializar a sessão com o servidor remoto*; Search* – após o envio de uma *query* que contem os termos a pesquisar, é devolvida uma resposta com os dados ou mensagens de erro*; Retrieval* – permite obter um conjunto de dados provenientes do servidor; *Delete* – utilizado para indicar ao servidor para eliminar um conjunto de resposta; *Access Control* – é um conjunto de procedimentos que permite dar acesso ou rejeitar o acesso de um utilizador aos dados armazenados; *Sort* e *Duplicate Detection* – permitem reorganizar os resultados provenientes do servidor, o primeiro ordena e o segundo serviço deteta e elimina registos duplicados; *Scan* – utilizado como paginador dos resultados; *Close* – termina uma ligação.

RSS – Rich Site Summary e *Atom – Atom Sindication Format* (Nottingham e Sayre 2005) – são documentos XML que pertencem ao grupo dos *feeds web*, e têm como objetivo o envio periódico de alterações recentes de uma página – notícias, publicações. Permite obter alguma informação dos conteúdos armazenados, como o título, descrição, data de publicação, autor, entre outros elementos. Assim, poderão ser utilizados em ambientes de interoperabilidade, no entanto têm mais restrições (metadados e respetivos valores enviados) em relação aos restantes, mas têm a vantagem, por vezes útil, de "avisar" da existência de novos conteúdos.

ATOM (Gregorio e hOra 2007) – *The Atom Publishing Protocol* – é um protocolo utilizado para publicar e editar recursos web via HTTP e XML. Possui um conjunto de serviços que permitem criar, editar e eliminar coleções e recurso *web*, através das operações *GET* – obter informação do repositório, obter objetos de uma coleção ou obter um objeto, *POST* – criar um novo objeto, *PUT* – atualizar/editar um objeto, *DELETE* – eliminar um objeto.

O *SWORD* (Lewis 2012) – *Simple Web-service Offering Repository Deposit* – surgiu em 2007 e foi desenvolvido pela *JISC – Joint Information Systems Committee*, com o objetivo de enviar/ depositar os conteúdos digitais de um repositório noutro repositório ou sistema, via *HTTP POST*.

METS (Federation 2010) – *Metadata Encoding and Transmission Standard* – criado em 2001, é um protocolo cujo objetivo é fornecer metadados para a gestão de conteúdos e para facilitar a troca desses conteúdos entre repositórios digitais. Os dados enviados pelo *METS* estão no formato *XML* e contem a seguinte informação: *header* – contem informação do documento (autor, data de criação); secção metadados descritivos; secção de metadados administrativos, que contêm informação relacionada com os conteúdos (direitos de autor, localização do ficheiro, metadados do conteúdo); lista de ficheiros (conteúdos); estrutura hierárquica dos conteúdos e relações entre os metadados e os conteúdos; relações entre a informação que se encontra na estrutura hierárquica; e conjunto de serviços que permitem associar ações automáticas e conteúdos digitais.

2.5. Metadados

Muitos são os formatos de metadados existentes e que permitem descrever os conteúdos digitais, de diferentes áreas – educacional, negócio, cientifica, cultural, multimédia. No entanto, o formato de metadados mais utilizado é o *Dublin Core,* porque descreve a maior parte dos recursos existentes na *web* – imagens, vídeos, áudio, páginas *web*, livros eletrónicos etc. Alguns dos metadados do *Dublin Core* são: *title* – armazena o título do recurso*, creator* – identifica o autor/criado do objeto*, subject* – pode ser utilizado como subtítulo, área científica, assunto*, format* – indica qual o formato do recurso ou dimensões*, identifier* – identificador do recurso*, relation* – indica quais os recursos relacionados*, language* – língua do recurso digital*, date* – pode ser usado como um período de tempo de um evento relacionado com o recurso.

Todos os repositórios digitais descritos neste trabalho utilizam o formato de metadados *Dublin Core*, no entanto existem outros formatos de metadados, utilizados por alguns repositórios que é necessário analisar, de forma a verificar a compatibilidade ou alternativas para o seu armazenamento. Um desses exemplos é o *LOM – Learning Object Metadata*.

O *LOM* tem como objetivo descrever recursos de aprendizagem, de forma a facilitar a reutilização, interoperabilidade e pesquisa desses recursos. Dividido em 9 grupos – *general* – metadados que descrevem o recurso*, lifecycle* – metadados de suporte aos registos históricos/ versões dos recursos*, meta-metadata* – descreve a informação dos metadados*, technical* – conjunto de requisitos técnicos e características do recurso digital*, educational* – metadados descrevem as características educacionais*, rights* – relacionados com os direitos de autor*, relation* – relação entre 2 recursos educacionais*, annotation* – armazena comentários e informação relacionada com estes*, classification* – sistemas de classificação dos recursos educacionais/de aprendizagem – possui alguns metadados que são facilmente adaptáveis aos metadados do *Dublin Core*, como por exemplo a língua, identificador, titulo, assunto, tipo de objeto, direitos de autor e relações. No entanto, existem outros que tornam a correspondência entre os metadados *LOM* e os metadados *Dublin Core* difícil ou impossível, como por exemplo, versão, estado do recurso (final, rascunho, indisponível), o contexto (escola primária, secundária, universidade) ou o nível de dificuldade (muito fácil, fácil, médio, difícil, muito difícil), pois são metadados que existem no ambiente educacional (IEEE 2002).

O *CONTENTdm*, para além do *Dublin Core*, utiliza o formato *VRA Core* – conjunto de metadados com o objetivo de descrever trabalhos relacionados com cultura visual – arquitetura, escultura, manuscritos, livros de arte, pinturas, etc. O *VRA Core* utiliza um conjunto de metadados (elementos e subelementos) para descrever essas obras, tais como: *agent* – descreve o autor, grupo de contribuidores para o desenvolvimento da obra*, culturalContext* – cultura, pais, língua, etnia associada à obra*, date* – representa a data ou datas de criação, publicação, apresentação, desenvolvimento da obra*, description* – comentários e outras notas que seja necessário associar à obra*, inscription* – informação adicional, como dedicatórias, assinaturas, datas importantes, história*, location* – localização geográfica ou do repositório onde se encontra a obra*, material* – material utilizado para o desenvolvimento da obra*, measurements* – tamanho, formato, peso da obra física ou digital*, relation* – composto por termos ou palavras que identifiquem obras relacionadas*, rights* – direitos de autor e *copyright, source* – referência à origem da obra*,*

stateEdition – número ou nome associado ao estado ou edição da obra*, stylePeriod* – define o período histórico ao qual a obra está inserida*, subject* – termos ou frases que descrevem ou identificam a obra*, technique* – descrevem os métodos utilizados no desenvolvimento da obra*, textref* – referências textuais*, title* – título atribuído à obra*, worktype* – identifica o tipo de coleção, trabalho ou imagem a descrever (Core 2007).

ODRL – Open Digital Rights Language – é um formato de metadados para a gestão de direitos de autor, em ambientes digitais. O *ODRL* consiste em 3 entidades principais *Assets* – contem o conteúdo digital/ físico; *Rights* – contem informação sobre as permissões de acesso – restrições, requisitos e outras condições; *Parties* – inclui informação dos utilizadores finais, titulares dos direitos e respetivos perfis de acesso. Embora com informação de direitos de autor mais completa, o *Dublin Core* possui alguns metadados que poderão representar parte dessa informação (Iannella 2002). São eles o *rights, licenses* e *rightsHolder (dcterms)*.

2.6. Análise das Características

Com base na informação dos repositórios descritos anteriormente, serão analisadas as características semelhantes e diferentes dos repositórios digitais, nomeadamente, os tipos de identificação, suporte a múltiplos formatos de dados e metadados, suporte de relações e as tecnologias utilizadas na interoperabilidade entre repositórios.

Identificadores

Foram encontrados vários tipos de identificadores entre os diferentes repositórios, nomeadamente, *CNRI Handle System, Uniform Resource Identifier e Uniform Resource Locator*.

Formato de dados

Todos os repositórios suportam múltiplos formatos de dados, no entanto, o mais completo é o *DuraCloud*.

Formatos de metadados

Todos os repositórios suportam múltiplos formatos de metadados, incluindo metadados *Dublin Core*. No entanto existem alguns repositórios que limitam a lista de outros formatos de metadados. São os casos do *Digital Commons*, apenas suporta metadados *Dublin Core*; do *IntraLibrary*, suporta apenas *Dublin Core, Learning Object Model* e *Open Digital Rights Language*; e do *CONTENTdm*, que além do *Dublin Core*, suporta apenas os metadados *VRA Core*.

Relações

Todos os repositórios mencionados anteriormente suportam relações, exceto o *EPrints* e o *Digital Commons*.

Multilíngua

Todos os repositórios suportam multilíngua.

Versões

Em relação às versões dos conteúdos digitais e/ou dos metadados, apenas os 3 repositórios da *DuraSpace – DSpace, FEDORA e DuraCloud* – possuem serviços para controlo das versões. Para os restantes, ou a informação não foi encontrada ou estes não suportam diretamente diferentes versões dos conteúdos, exceto se forem efetuadas cópias de segurança dos conteúdos digitais armazenados.

Proteção de conteúdos

Todos os repositórios oferecem um ou outro mecanismo para proteção dos conteúdos. Mas, os tipos de proteção que mais se destacam são a utilização de nome de utilizador e palavra passe, associados a um perfil de utilizador ou grupo de utilizadores e *LDAP*.

Em relação à integridade da informação, todos os repositórios suportam o algoritmo *MD5*, no entanto, o *FEDORA* oferece outros algoritmos – *SHA (1, 256, 384, 512).*

Interoperabilidade

Na partilha de conteúdos as tecnologias de interoperabilidade (apresentadas anteriormente, no ponto 2.4) têm um papel fulcral, pois permitem enriquecer os conteúdos de outros repositórios. Todos os repositórios mencionados anteriormente, exceto o *DuraCloud*, permitem a troca de metadados utilizando o protocolo *OAI-PMH*. A tecnologia *feed,* como o RSS, utilizada por alguns repositórios, também permite que alguns conteúdos de outros repositórios possam ser inseridos. Outros protocolos, tais como *OAI-ORE e o Z39.50,* também são utilizados por alguns dos repositórios mencionados.

Acesso

Perante o aparecimento e evolução da *web*, todos os repositórios possuem uma interface *web* para acesso e/ou gestão dos conteúdos digitais. Mas outras formas de acesso são possíveis, como as aplicações *desktop*, ficheiros *batch* e/ou linha de comandos. Outros repositórios oferecem serviços *Web*, via *SOAP* ou *REST*, para gestão e acesso aos conteúdos digitais, nomeadamente o *FEDORA* e o *DuraCloud*.

Perante as diferenças é necessário analisar a possibilidade de aproximá-las de forma a construir a API de serviços para a troca dos conteúdos entre os diferentes repositórios. As maiores diferenças, que poderão ser críticas na construção da API de serviços, foram encontradas nos identificadores, nos formatos de metadados (alguns dos quais descritos no ponto 2.5) utilizados, no suporte de relações entre objetos, no armazenamento e controlo de versões dos conteúdos digitais, na interoperabilidade e na interface de acesso aos conteúdos digitais.

A forma como os repositórios identificam os seus conteúdos não é de todo o ponto crítico das diferenças entre os repositórios, uma vez que, apesar de serem construídos de forma diferente e

com diferentes objetivos (Anexo B - Identificadores), cada repositório atribui de forma automática identificadores aos seus objetos, sem que estes tenham influência externa.

Alguns repositórios não suportam relações entre os conteúdos digitais, no entanto, como todos suportam metadados *Dublin Core*, pode-se guardar o identificador ou a ligação para o outro conteúdo digital no metadado "*relation*" do *Dublin Core*. Outra alternativa é armazenar todas as relações num único objeto, por exemplo, em formato XML, pois todos os repositórios suportam diferentes formatos de dados.

A mesma lógica pode ser aplicada às versões dos documentos, ou seja, caso se pretenda converter informação proveniente de um repositório que suporta versões para outro que não suporta versões de conteúdos, como por exemplo a passagem de informação do *DSpace* para o *EPrints*, pode-se armazenar cada versão de um objeto noutro objeto e adiciona-se uma relação entre estes objetos.

Como referido anteriormente, todos os repositórios suportam múltiplos formatos de metadados, incluindo o *DublinCore*. No entanto, existem algumas restrições, como são os casos do *CONTENTdm* – apenas suporta *Dublin Core* e *VRA Core;* do *Digital Commons* – apenas suporta *Dublin Core* e do *IntraLibrary* – suporta *Dublin Core, LOM e Open Digital Rights Language*. Embora alguns metadados são equiparados a metadados do *Dublin Core*, existem muitos que não são convertíveis ou dificilmente convertíveis em metadados *Dublin Core*. Contudo, a passagem da informação dos repositórios *CONTENTdm, IntraLibrary* e *Digital Commons* para outros repositórios, não é um problema porque esses repositórios suportam múltiplos metadados, o problema reside na passagem inversa – repositórios que suportam múltiplos formatos para os repositórios *CONTENTdm, IntraLibrary, Digital Commons* e outros repositórios que tenham restrições ao nível dos metadados. Uma solução possível para este problema é a utilização de objetos extra, uma vez que esses repositórios suportam múltiplos objetos e formatos de dados, que contenham esses metadados, à semelhança das relações e das versões dos conteúdos digitais.

A interoperabilidade não é um requisito importante, uma vez que os protocolos de interoperabilidade não influenciam os conteúdos internos do repositório, no entanto, em certos casos, poderão ser úteis para complementar a informação extraída de um repositório, por exemplo, o OAI-PMH permite obter formatos de metadados e respetivos metadados de cada objeto digital, auxiliando ou complementando as interfaces de acesso, sobretudo as interfaces *web* ou *desktop*, que tornam mais difícil a interação de programas automáticos na aquisição da informação. No caso das interfaces *web* pode-se utilizar *WebCrawlers*[1] – programa que permite obter, de forma automática, conteúdos que existam em determinadas páginas *web* ou páginas relacionadas (têm, por exemplo, uma ligação para outras páginas) ou protocolos de comunicação, como o OAI-ORE.

2.7. Casos de estudo

Na interoperabilidade entre repositórios com características diferentes foram efetuados diversos estudos, alguns dos quais com sucesso prático – como o *OAI-PMH*, utilizado pelos

[1] http://www.webcrawler.com/

repositórios para expor os metadados dos objetos armazenados; o *OAI-ORE*, utilizado para expor relações entre objetos. Foram também efetuados estudos mais específicos em relação aos metadados como, por exemplo, o estudo realizado por *Do Van Chau* (Chau 2011), cujo objetivo era a migração de metadados de uma base de dados *DUO – DigitaleutgivelservedUiO* (repositório digital da Universidade de *Oslo*) para *DSpace*. Esse estudo foi realizado na biblioteca da Universidade de *Oslo* e teve como principais desafios a redução de conflitos e riscos na migração dos respetivos dados entre os 2 formatos de metadados – metadados específicos do sistema *DUO* e metadados *Dublin Core* utilizados no *DSpace*. *Do Van Chau* identificou alguns conflitos nessa troca de informação, tais como, perda de dados ou dados incorretos, dados com informação vazia ou duplicados e perda da estrutura dos elementos. Apesar dessas dificuldades *Do Van Chau* encontrou duas formas para migrar a informação: mapear os metadados do DUO na lista de metadados *Dublin Core*, permitindo a interoperabilidade com outros repositórios (utilizando o *OAI-PMH,* por exemplo) ou construir um *schema* diferente do *Dublin Core* que armazenasse os metadados não suportados pelo *Dublin Core*. Em ambos os casos existem desvantagens. Na primeira abordagem, a probabilidade de perda de informação é grande, a segunda abordagem obriga a um esforço extra de recursos humanos e financeiros para o desenvolvimento e manutenção do repositório. Assim, a abordagem melhor depende dos recursos humanos, financeiros e estruturais disponíveis para a realização da tarefa.

Um segundo estudo é o projeto *SWORD – Simple Web-service Offering Repository Deposit*, desenvolvido pela *JISC – Joint Information Systems Committee*, com o objetivo de enviar/ depositar os conteúdos digitais de um repositório noutro repositório ou sistema, via *HTTP POST*. De acordo com *Julie Allinson* (Allinson 2009), o *SWORD* foi desenvolvido porque não existiam protocolos para carregar e transferir objetos entre repositórios. Assim, utilizando uma filosofia cliente-servidor, o *SWORD* permite o "depósito" de conteúdos provenientes de um repositório para outro ou de um repositório para múltiplos repositórios, de forma autónoma. No entanto, precisa de 2 instalações por comunicação – lado do cliente (contem conjunto de serviços para adquirir informação do repositório de origem) e do lado servidor (comunica com o repositório de destino). Para além deste problema, existe o facto de ser um protocolo restrito a alguns repositórios, *DSpace, EPrints, IntraLibrary* e *Fedora*.

Apesar de existirem alguns estudos, protocolos e serviços para a troca de informação entre repositórios digitais, esses serviços estão dependentes dos repositórios, ou seja, um repositório tem, de forma direta ou indireta, responder aos pedidos desses serviços. O SWORD ainda tem a limitação de enviar informação apenas de um repositório. Por estas razões foi pensado o desenvolvimento de um conjunto de serviços para permitir a troca e agregação de conteúdos digitais entre repositórios digitais, independentemente da tecnologia utilizada.

3. Proposta de API de Serviços

3.1. Motivação

Como referido no ponto anterior, são algumas as restrições impostas pelas tecnologias que permitem a troca de informação entre repositórios digitais. Foram essas as razões que nos levaram a pensar num conjunto de operações que melhorassem essa troca de informação, nomeadamente, permitissem a aquisição e inserção de conteúdos noutros repositórios, e permitissem a agregação de conteúdos provenientes de vários repositórios, para evitar dispersão da informação. Todas estas funcionalidades deverão ser externas e independentes aos repositórios digitais para não existir a necessidade de alterá-los.

3.2. Requisitos

Após o levantamento dos requisitos dos repositórios, as características de alguns repositórios e das funcionalidades que outros serviços realizam, neste capítulo serão apresentados alguns requisitos e características que a *API* de serviços deve ter em conta.

Em primeiro lugar, a *API* de serviços deve permitir a troca de dados (metadados e conteúdos) entre diferentes repositórios. Para isso deverão existir serviços para a gestão de metadados, de conteúdos, de versões dos conteúdos e dos metadados, gestão de relações entre objetos e para a gestão de acessos. Em segundo, e como ilustra a Figura 1, é necessário que existam serviços de exportação dos conteúdos para diferentes repositórios (1 ou mais). Por fim, é necessário garantir que os conteúdos provenientes de diferentes repositórios possam ser armazenados num ou vários repositórios.

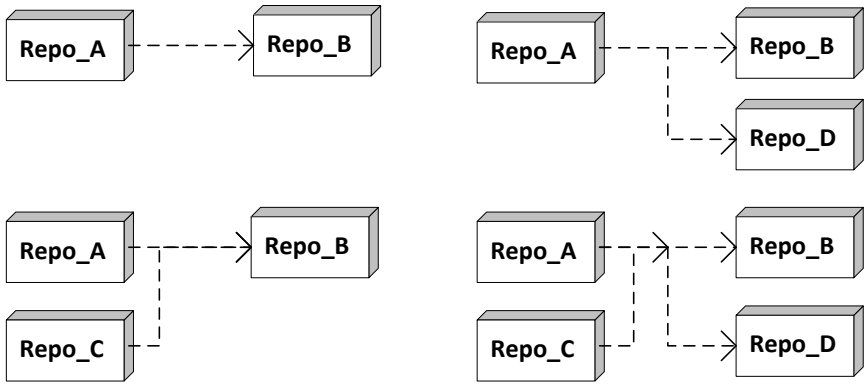

Figura 1. Diagrama simplificado da possível comunicação entre os repositórios

3.2.1. Casos de utilização

Por forma a construir uma aplicação mais simples, os requisitos mencionados no ponto anterior, foram subdivididos em 2 grupos com base nos objetivos de cada um. Assim, como ilustram a Figura 2 e a Figura 3, as operações cujo objetivo é a interação com os repositórios –

inserção e aquisição de conteúdos – serão agrupados no módulo de aquisição e inserção de conteúdos - Figura 2, enquanto os requisitos relacionados com a agregação dos conteúdos serão colocados no módulo agregador - Figura 3. De seguida serão descritos cada um dos módulos e respetivas funcionalidades.

Figura 2. Casos de utilização (comunicação entre 2 repositórios)

Na Figura 2 é apresentado um diagrama de casos de utilização que ilustra as operações existentes na comunicação entre 2 repositórios (origem e destino). Neste diagrama é possível observar as operações para a gestão de metadados (o sistema obtém os metadados de um repositório e as versões, se suportadas pelo repositório, para posteriormente inserir noutro repositório); gestão de conteúdos (funcionamento semelhante à gestão de metadados) e gestão de relações (o sistema obtém as relações, processa-as e envia-as para o (s) repositório (s) de destino).

Na Figura 3 é apresentado um modelo do sistema agregador, que permite analisar a informação (metadados, conteúdos) provenientes de vários repositórios e posteriormente, se a informação proveniente de um repositório for semelhante à informação de outro, agrupa-a e envia-a para o (s) repositório (s) de destino.

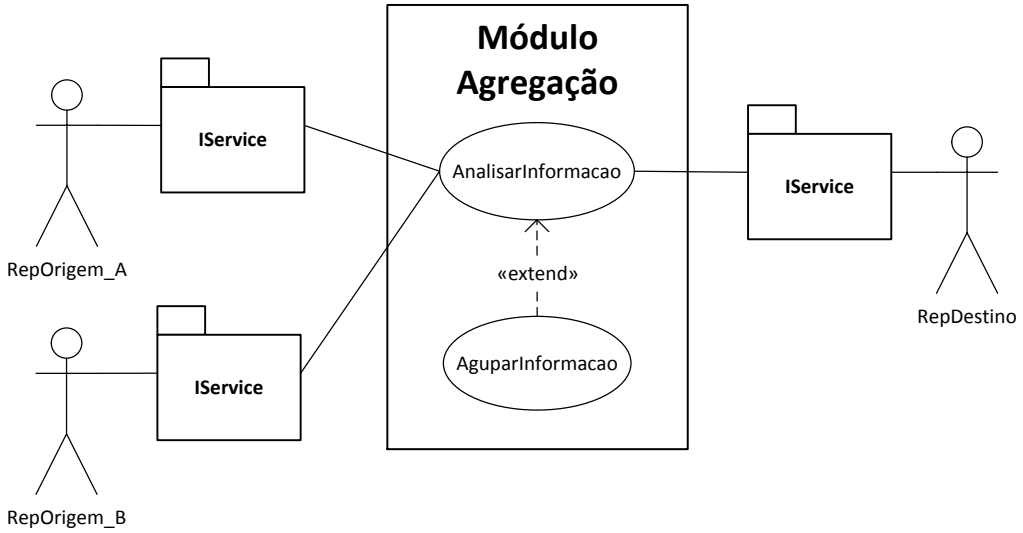

Figura 3. Casos de utilização (agregador)

3.2.2. Arquitetura

Para permitir a troca de informação entre repositórios com características diferentes, são necessários elos de ligação que os tornem compatíveis. Esta é a proposta deste trabalho, criar um ponto de união entre os diferentes repositórios, com o intuito de facilitar o envio de dados de um repositório para outro. A Figura 4 ilustra um conjunto de repositórios (*IntraLibrary, Fedora, DuraCloud, EPrints, DSpace*) que comunicam entre si através de um único ponto – *IService*. Nessa imagem é possível visualizar os quatro tipos de comunicação pretendidos com esta proposta de API de serviços para a interoperabilidade entre repositórios: envio de informação de um repositório para outro (*IntraLibrary* para *DSpace*), envio da informação para vários repositórios (*IntraLibrary* para *DSpace* e *EPrints*) e agregação de conteúdos provenientes de vários repositórios (componente *Aggregator*) e respetivo envio para 1 ou mais repositórios (do *Fedora* e *DuraCloud* para *DSpace* e/ou *EPrints*).

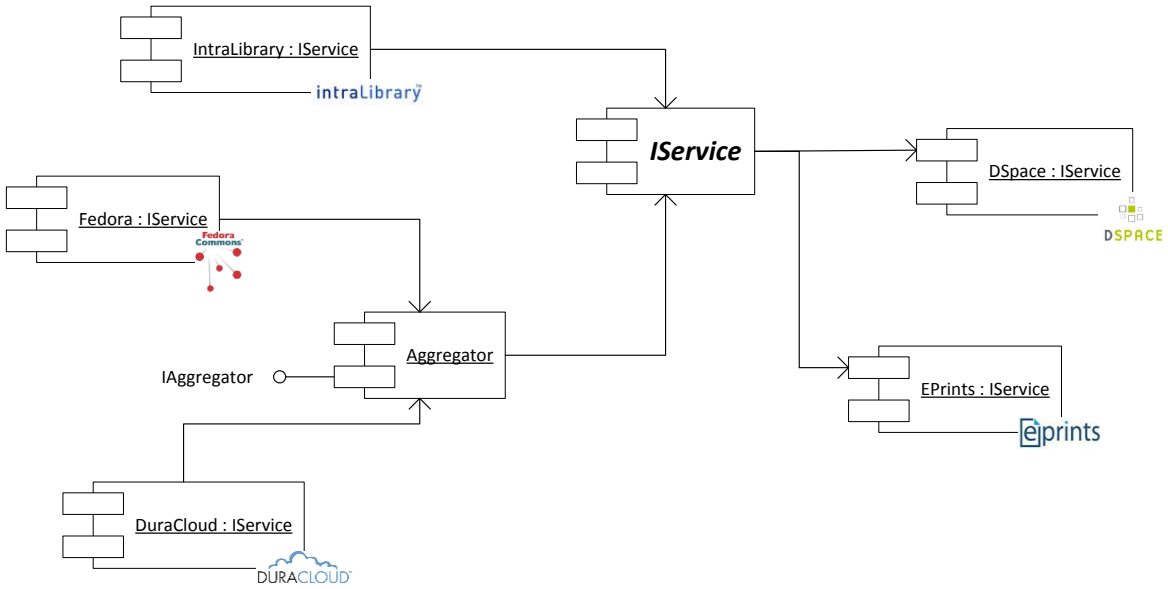

Figura 4. Arquitetura

O *IService* é um componente abstrato que contem informação sobre as operações necessárias para a troca de informação entre os repositórios, ou seja, é o componente que "faz a ponte" entre os repositórios digitais. Por sua vez, os componentes *IntraLibrary, Fedora, DuraCloud, DSpace* e EPrints implementam essas funcionalidades de acordo com as características dos repositórios associados. O componente *Aggregator* contem operações que permitem analisar a semelhança entre conteúdos digitais e operações para agregar esses conteúdos que, posteriormente serão submetidos para um ou mais repositórios.

Como analisado anteriormente, existem várias formas do *IService* aceder aos repositórios, utilizando ficheiros *batch*, interface *web* ou serviços *web* (REST, SOAP). Após o acesso à informação é necessário armazená-la nos objetos do *IService* para posteriormente ser enviada para o (s) repositório (s) de destino ou para o componente *Aggregator*.

3.2.3. Modelo de Dados

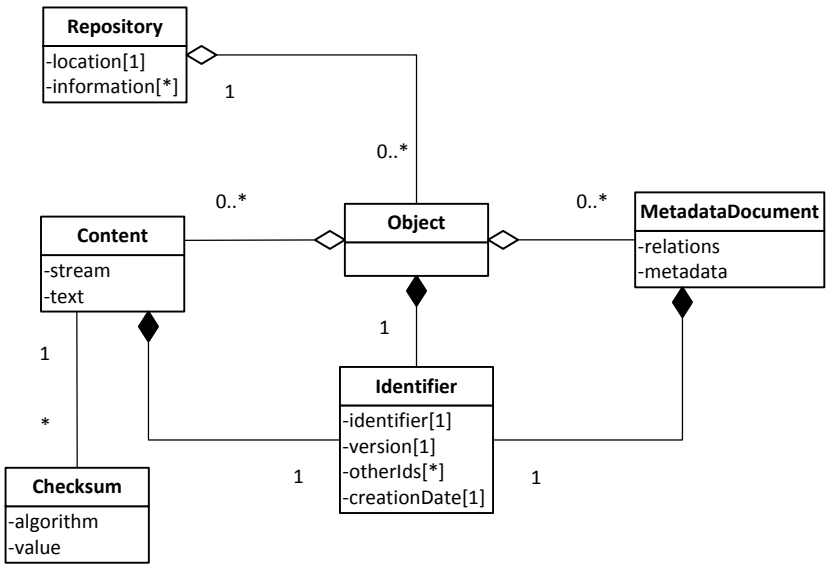

Figura 5. Modelo de dados

Na Figura 5 é ilustrada o modelo de dados de suporte aos conteúdos extraídos e aos conteúdos a inserir pela API. Um repositório digital, identificado pela localização deste e informação complementar, poderá ter uma lista de objetos digitais, onde serão armazenados os conteúdos, os metadados, as relações entre os diferentes objetos e respetivas versões. À semelhança dos repositórios analisados, todos os objetos possuem um identificador. No caso do *EPrints*, os utilizadores poderão inserir outros identificadores, assim os objetos da API deverão permitir o armazenamento desta informação, além de uma identificação da versão, e respetiva data, do objeto ou dos seus conteúdos. Em alguns casos, os conteúdos dos objetos possuem um algoritmo de controlo de erros – *checksum* – assim, é necessário que existam elementos que permitam armazenar esses valores – entidade *Checksum* apresentada no diagrama. Em cada objeto é possível existirem conteúdos, metadados e relações. Com isto, as entidades *Content* e *MetadataDocument* armazenam, respetivamente, essa informação. No caso da entidade *Content,* é possível armazenar um ficheiro, texto livre ou uma ligação (*URL*) externa. No caso da entidade *MetadataDocument*, existe uma *flag* que indica a existência de relações no documento. O atributo *metadata* desta entidade permite armazenar os metadados e relações (em formato texto ou *xml*) do documento.

3.3. Desenvolvimento

Após a análise dos requisitos do sistema e funcionalidades inerentes a estes e depois da descrição de uma arquitetura para a comunicação entre os repositórios e dos objetos que permitirão guardar temporariamente os conteúdos desses repositórios, a solução apresentada será implementada. Recorrendo à linguagem de desenvolvimento *Java* e, numa primeira fase, utilizando repositórios-modelo também desenvolvidos em *Java*, os módulos serão implementados

e testados de acordo com as necessidades inerentes à troca de informação entre os repositórios digitais, nomeadamente, serão desenvolvidos serviços que permitam adquirir os conteúdos, incluindo as suas versões, existentes num ou mais repositórios, serviços que permitam agregar esses conteúdos, caso a informação venha de vários repositórios, e serviços para inserir os conteúdos adquiridos diretamente dos repositórios ou os conteúdos agrupados, em um ou mais repositórios, independentemente das características, funcionalidades e ambientes de operação desses mesmos repositórios.

3.3.1. Repositórios

Nesta fase do trabalho, foram selecionados 3 repositórios com características díspares – o *Fedora*, o *DSpace* e o *EPrints*. O *Fedora* suporta diversos formatos de metadados e de conteúdos, contem informação para a construção de relações e permite o armazenamento de versões dos recursos. Da mesma forma o *DSpace* permite o armazenamento de diversos formatos de metadados, de conteúdos, de relações e respetivas versões. Difere do *Fedora* na forma de construção e armazenamento dos identificadores dos conteúdos, pois o *DSpace* utiliza o sistema *Handle* da CNRI (CNRI 2012), ao contrário do *Fedora* que utiliza o sistema de *URIs* (Anexo B - Identificadores). O *EPrints* foi escolhido por não ter um componente que permita armazenar, diretamente, as versões dos conteúdos e as relações entre recursos digitais e tem a particularidade de permitir a inserção de outros identificadores do recurso. Com estes repositórios, é possível testar mais funcionalidades, melhorando a abrangência da API de serviços.

Como referido no ponto anterior, foram criados objetos que simulam a utilização destes repositórios, no entanto, para melhorar os resultados dos testes será necessário, numa fase posterior, proceder a algumas alterações de código para que a API comunique diretamente com cada um dos repositórios.

Apesar das limitações da utilização de repositórios em ambientes de teste, os próximos passos descreverão a implementação do módulo de aquisição e inserção de conteúdos e do módulo de agregação, necessário na agregação de conteúdos provenientes de múltiplos repositórios digitais.

3.3.2. Módulo de aquisição e inserção

O módulo de aquisição e inserção é o módulo mais importante de todo o sistema, pois é o módulo responsável por obter os conteúdos dos diferentes repositórios, convertê-los em objetos que permitem a compatibilidade entre os repositórios e, posteriormente inserir esses mesmos conteúdos nos diferentes repositórios, consoante as necessidades.

Assim, antes da realização das operações é necessário estabelecer uma ligação com o repositório utilizando os meios de comunicação disponíveis, protocolos de interoperabilidade, interfaces de acesso – *web, SOAP, REST,* linha de comandos, entre outros. Após a validação e aceitação da ligação, as operações de acesso aos conteúdos podem ser realizadas. A operação utilizada pela API para a ligação com o repositório possui um parâmetro (*location*) que permite indicar qual o endereço *URL* ou caminho para o sistema que possui operações para a aquisição e inserção de conteúdos.

boolean connection(String location); *// permite definir um caminho para o serviço de inserção e aquisição de conteúdos*

String location(); *// devolve o caminho do serviço para a inserção e aquisição de conteúdos*

As restantes operações, utilizadas pela API, para a gestão dos repositórios, aquisição e inserção de conteúdos serão descritas de seguida.

De forma a verificar algumas das capacidades dos repositórios, foram construídas algumas operações, tais como *supportsRelations* e *supportsVersions*, que permitem indicar, respetivamente, o suporte de relações e de versões do repositório digital.

boolean supportsRelations();

boolean supportsVersions();

A operação *objects* foi construída para devolver a lista dos recursos (objetos) existentes no repositório. Após a utilização desta operação é possível obter os metadados, relações, conteúdos e respetivas versões associados a um objeto.

List<DigitalObject> objects(); *// obter os objetos*

List<MetadataDocument> metadata(boolean relations); *// obter todos os metadados (parametro* ***relations*** *com valor* ***falso***) *ou relações (parametro* ***relations*** *com valor* ***true***) *existentes no repositório*

List<MetadataDocument> metadata(String objectID, boolean relations); *// obter metadados ou relações (se parametro tiver o valor* ***true****) de um objeto, incluindo as versões*

List<Content> content(); *// obter todos os conteúdos do repositório*

List<Content> content(String objectID); *// obter conteúdos (texto ou conteúdo de um ficheiro) de um objeto, incluindo as versões dos conteúdos*

À semelhança da aquisição de conteúdos, existem operações para a gestão de objetos, metadados, relações, conteúdos e versões, nomeadamente a inserção e atualização destes recursos.

Para criar um objeto, é utilizado a operação *createObject*. Esta operação para além de criar um novo objeto no repositório de destino e devolver o identificador desse objeto, armazena temporariamente uma lista de mapeamentos dos identificadores dos objetos criados e dos identificadores utilizados no repositório de origem – razão pela qual existe o parâmetro *lastID*, que corresponde ao identificador no repositório de origem. Esta lista permite, posteriormente, atualizar todas as relações entre os objetos. Ou seja, se um objeto com identificador *educa:1* estiver relacionado com os objetos *educa:892* e *educa:1000* (informação armazenada em cada um dos objetos). Após a inserção de todos os objetos, sabe-se que o *educa:1* passou a designar-se por *abc:tese:45:9*, o objeto *educa:892* foi substituído por *abc:tese:29:1* e o *educa:1000* foi alterado

para *abc:tese:239:289*, no entanto os documentos que armazenam as relações não foram atualizados, porque essa tarefa só é possível após o conhecimento dos restantes identificadores. Assim, com a lista de mapeamentos, é possível atualizar os identificadores dos objetos existentes nos documentos das relações, após a criação dos mesmos no novo repositório.

String createObject(String lastID);

Para completar os objetos existe um conjunto de operações que permitem adicionar metadados, relações e conteúdos ao repositório digital, incluindo as versões dos mesmos. Se o repositório não suportar versões dos conteúdos, será criado um objeto e adicionada uma relação entre os dois objetos (o que contem o conteúdo mais recente e o objeto que contem a versão do conteúdo). Caso o repositório não suporte relações entre os objetos, será criado um objeto que conterá um documento (XML) com todas as relações (incluindo as relações resultantes das versões dos conteúdos) entre os objetos.

boolean addMetadata(String objectID, MetadataDocument metadata); *// permite adicionar a um objeto um documento de metadados*

boolean addRelation(String objectID, MetadataDocument relations); *// permite adicionar a um objeto as relações que este possui com outros objetos*

boolean addContent(String objectID, Content content); *// permite adicionar um conteúdo a um objeto*

Para a atualização das referências dos objetos, utilizadas nos documentos que contêm relações entre os objetos, é utilizado o método *updateRelations*. Este método, para cada documento relacional, altera os identificadores dos objetos antigos pelos novos identificadores, como referido anteriormente.

boolean updateRelations(MetadataDocument relations); *// permite atualizar todas as referências de um objeto*

3.3.3. Módulo de agregação

O módulo de agregação é responsável por juntar objetos semelhantes, evitando a replicação dos mesmos. A implementação e utilização deste módulo oferecem algumas vantagens, mas também algumas desvantagens. Em relação às vantagens, a utilização de um módulo agregador permite poupar espaço de armazenamento dos conteúdos e evitar que a informação do (s) repositório (s) de destino aparentem estar dispersas, ou seja, se um utilizador pesquisar uma obra com base no título, poderão surgir resultados que são, do ponto de vista do utilizador, iguais. Com esta funcionalidade, o número de resultados a apresentar ao utilizador diminuirá. Contudo, a implementação do conceito de objetos semelhantes é complexa e morosa, uma vez que é necessário comparar os conteúdos (ficheiros, ligações para páginas *web*, documentos sob a forma

de texto) e os metadados (incluindo os diferentes formatos e extensões destes) dos objetos, incluindo as versões dos mesmos. No momento de juntar os objetos semelhantes é necessário ordenar as versões dos conteúdos e dos metadados, no entanto, é difícil analisar qual ou quais as versões mais completas e corretas. Contudo, é necessário analisar algumas formas de diminuir os erros na análise e agregação dos objetos digitais, como será descrito no decorrer deste capítulo.

De um ponto de vista lógico, o módulo agregador é simples, pois contém apenas duas operações – *analyze* (utilizado para comparar 2 objetos) e *merge* (utilizado para agregar a informação de 2 objetos), como ilustra a figura seguinte.

```
boolean analyze(DigitalObject object1, DigitalObject object2); // permite verificar se 2 objetos são iguais

DigitalObject merge(DigitalObject object1, DigitalObject object2); // junta a informação de 2 objetos num único objeto
```

Operação de análise

Como referido, a análise dos objetos é uma tarefa difícil (porque a probabilidade de classificar erradamente 2 objetos como semelhantes é elevada) e morosa (porque é necessário comparar conteúdo a conteúdo, incluindo as versões dos mesmos). Mas, o número de erros na classificação de objetos semelhantes poderá ser diminuído.

Uma vez que todos os repositórios mencionados nesta dissertação suportam metadados *Dublin Core*, estes podem ser utilizados como um termo de comparação. Por exemplo, podem-se utilizar os campos *author/creator* (identifica o autor do recurso), *language* (indica a linguagem utilizada no recurso), *subject* (assunto do recurso), *title* (titulo e subtítulos), *description* (breve descrição do recurso), *type* (género do recurso), *format* (formato), entre outros. Como estes campos são textuais poderão não ser exatamente iguais, então é necessário construir uma função que calcule o desvio do texto (tamanho do texto, número de palavras iguais ou semanticamente iguais) de cada termo, e abaixo de um valor de desvio, considerar os conteúdos iguais. No final, se existirem 8 metadados, em 10, considerados semelhantes pode-se dizer que os objetos são semelhantes.

Contudo, a análise por via de metadados não é suficiente, uma vez que os documentos que contêm os metadados podem ser exatamente iguais, e o conteúdo (ficheiros, ligações externas, texto, etc.) ser diferente. Assim, também se deve aplicar outras técnicas para a análise dos objetos.

Na comparação de conteúdos que contenham texto, pode-se utilizar a mesma técnica referida anteriormente, no caso das ligações externas (URL) pode-se efetuar a comparação direta do conteúdo – por exemplo http://www.ua.pt é igual a http://www.ua.pt? No entanto, existem alternativas na comparação dos conteúdos textuais e dos conteúdos armazenados em ficheiros, como a utilização de funções *hash*. Alguns dos algoritmos mais utilizados para construir uma função *hash* são, entre outros, o MD5, SHA-1 e SHA-256. Contudo, existem algumas desvantagens na utilização de funções *hash* como o tempo de cálculo da *hash* de documentos grandes e a probabilidade de existirem colisões, ou seja, a probabilidade de documentos diferentes

produzirem o mesmo valor é grande, e aumenta com o número de documentos a analisar, embora a utilização de algoritmos SHA diminua esse problema, uma vez que o algoritmo, dependendo do tamanho da chave, é mais eficiente e eficaz (Mulvey 2007) (Ratzan 2004). Outra alternativa à comparação de ficheiros é a análise dos metadados dos mesmos. Apesar de ser mais rápido que a análise recorrendo à técnica de *hash*, esta análise pode não ser suficiente, porque, normalmente, os metadados existentes nos vídeos, nos ficheiros ou noutros formatos resumem-se ao autor, formato, tamanho, datas de criação, alteração, entre outros.

Apesar das desvantagens de cada uma das formas de comparar os objetos digitais, optou-se por criar um conjunto de funções cujos resultados indicam a semelhança dos objetos. Essas funções serão explicadas de seguida.

int classifyMetadata(MetadataDocument doc1, MetadataDocument doc2); *// devolve a percetagem de semelhança entre os documentos, 0% - documentos diferentes a 100% - documentos iguais*

A operação anterior permite obter um valor (de 0 a 100) que classifica (em percentagem) a semelhança entre documentos que possuam metadados, por exemplo, se o resultado for próximo de 100%, significa que os documentos são semelhantes. Como explicado anteriormente, serão selecionados alguns metadados desses documentos e será efetuada uma comparação (número de palavras semelhantes) dos valores desses metadados. No final, se existirem 8 metadados semelhantes de 10 possíveis, os documentos serão considerados semelhantes.

int classifyContent(Content doc1, Content doc2) *// devolve a percetagem de semelhança entre os conteúdos, 0% - documentos diferentes a 100% - documentos iguais*

No caso dos conteúdos, se estes forem apenas uma ligação externa, é efetuada a comparação direta, se for texto, o texto é analisado recorrendo à técnica utilizada nos metadados, pois a utilização de funções *hash* obriga a que os documentos sejam exatamente iguais. Poder-se-ia criar uma função *hash* de cálculo de semelhanças entre os documentos, mas este é um processo difícil e que requer muitos anos até encontrar uma função que seja eficiente e eficaz nessa análise. No caso dos restantes formatos dos conteúdos será aplicada, primeiramente, a análise de semelhança dos metadados dos conteúdos e depois a análise recorrendo à técnica de *hash*.

Depois destas análises, é efetuada uma soma dos valores e atribuído um peso superior na análise de texto, seguindo-se a análise de metadados e por fim a análise recorrendo à utilização de funções *hash*. Assim, como o texto e os metadados são analisados com detalhe, têm um maior peso que as funções *hash*, uma vez que o resultado destas não é tão fiável porque dependem da igualdade dos conteúdos – basta existir um caracter diferente, para o resultado da *hash* ser diferente – e dependem da capacidade do algoritmo evitar os conflitos. Com base no resultado obtido, os objetos são considerados semelhantes. Após a obtenção deste resultado, os objetos serão agregados, como explicado no ponto *"Operação de agregação"*. Mas, antes da análise da operação de agregação, serão explicadas algumas operações que auxiliam o processo de análise de semelhança de objetos digitais.

Comparação de texto

Na comparação de texto, foram utilizadas 3 operações: *words* que extrai todas as palavras de um documento, *countMatches* que permite contar o número de ocorrências de uma palavra num texto e *similarityIndexText*, que devolve o valor de similaridade (entre 0 – textos diferentes e 100 – textos iguais) entre dois textos, com base no número de palavras iguais.

double similarityIndexText(String txt1, String txt2); *// valor de similaridade entre 2 textos*

ArrayList<String> words(String txt); *// lista de palavras*

int countMatches(txt, word); *// número de ocorrências de uma palavra*

Comparação de metadados

No caso da comparação de metadados foram construídas as operações *metadata* – que permite extrair os metadados de um ficheiro, de um conjunto de *bytes* ou de um documento *XML* – e *similarityIndexMetadata*, que devolve o valor de similaridade (entre 0 e 100) de dois documentos de metadados.

double similarityIndexMetadata(MetadataDocument doc1, MetadataDocument doc2); *// valor de similaridade entre 2 documentos de metadados*

HashMap<String, String> metadata(T doc); *// permite extrair metadados de um documento ou ficheiro*

Comparação de conteúdos

Na comparação de conteúdos foram utilizadas algumas das operações descritas anteriormente, nomeadamente, a operação *metadata*, que permite extrair metadados de um ficheiro ou documento e a operação *similarityIndexText,* utilizada na comparação de texto. Contudo, além dessas operações, foram criadas a operação *similarityIndexContent,* que devolve o valor de similaridade entre 2 conteúdos, independentemente de serem documentos de texto, conteúdos de ficheiros ou ligações externas, e a operação *isSimilar*, utilizada para verificar se o conteúdo de dois ficheiros é semelhante, com base na extração de metadados (valor com peso no resultado final de 75%) e na comparação do resultado das funções *hash* (utilizando o algoritmo SHA-256 e com peso no resultado final de 25%) desses conteúdos. No caso das ligações externas, é utilizada a operação *equals* do java (*String.equals*).

double similarityIndexContent(Content doc1, Content doc2); *// devolve o valor de similaridade entre 2 conteúdos*

double isSimilar(File doc1, File doc2); *// retorna o valor de similaridade entre 2 conteúdos, com base nos metadados e resultado da hash* dos mesmos

Operação de agregação

Na agregação de objetos semelhantes, a questão principal centra-se na ordenação das versões dos conteúdos. Pode-se considerar que um documento é mais recente que o outro com base na análise de alguma informação que está junto do documento, como as datas de criação e de atualização. No entanto, é muito difícil considerar que a versão mais recente é aquela que está mais completa e correta. Uma hipótese seria a análise da informação, nomeadamente, os metadados ou os conteúdos textuais. Apesar dessa análise, uma versão pode ter mais informação, os conteúdos (ficheiros), os valores dos metadados e o texto podem ser maiores, e por consequência, assumir que a versão é a mais completa, mas não a mais correta. Esta última consequência é difícil ou mesmo impossível ser analisada recorrendo ao *software* existente atualmente, uma vez que este tipo de análise, até ao momento, é efetuado por pessoas.

Contudo, assume-se que uma versão é mais recente que outra, se as datas de atualização e criação forem superiores e, embora com menor peso na análise, se a quantidade de metadados preenchidos e o tamanho dos documentos e dos conteúdos for superior às restantes versões.

MetadataDocument compareMetadata(MetadataDocument doc1, MetadataDocument doc2)
// devolve o documento mais completo (em termos de metadados)

Content compareContent(Content doc1, Content doc2) *// devolve o conteúdo maior*

Resumidamente tem-se:

IService

Define um conjunto de serviços que permite a troca de informação entre repositórios digitais. As *DSpace, EPrints* e *Fedora* implementam este serviço e interagem com os respetivos repositórios digitais.

Estabelecer ligação com o repositório

boolean	connection(String location)
String	location()

Inserção de conteúdos

boolean	addContent(String objectID, Content content)
boolean	addMetadata(String objectID, MetadataDocument metadata)
boolean	addRelation(String objectID, MetadataDocument relations)

String	createObject(String lastID)
boolean	updateRelations()
boolean	updateRelations(MetadataDocument relations)

Aquisição de conteúdos

List<Content>	content()
List<Content>	content(String objectID)
List<MetadataDocument>	metadata(boolean relations)
List<MetadataDocument>	metadata(String objectID, boolean relations)
List<DigitalObject>	objects()
boolean	supportsRelations()
boolean	supportsVersions()

IAggregator

Define um conjunto de operações que permite analisar e agregar objetos, constituídos por metadados, relações e conteúdos digitais. *Aggregator* é uma classe que implementa essas operações.

boolean	analyze(DigitalObject object1, DigitalObject object2)
DigitalObject	merge(DigitalObject object1, DigitalObject object2)

Aggregator

Implementa as operações definidas no *IAggregator* e contem outras operações que obtêm a similaridade entre os diferentes objetos.

boolean	analyze(DigitalObject object1, DigitalObject object2)
int	classifyContent(Content doc1, Content doc2)
int	classifyMetadata(MetadataDocument doc1, MetadataDocument doc2)
DigitalObject	merge(DigitalObject object1, DigitalObject object2)

Após a análise e tomada de decisões em relação à implementação da API de serviços para a interoperabilidade entre os diferentes repositórios, é necessário realizar alguns testes às funcionalidades principais da aplicação. Estes testes serão descritos no ponto seguinte.

3.4. Testes

Neste subcapítulo serão apresentados alguns testes às funcionalidades implementadas anteriormente.

Como o desenvolvimento da API foi realizado em ambientes de simulação, apenas foram considerados 3 repositórios digitais - o *Fedora*, o *DSpace* e o *EPrints*, como referido no ponto 3.3.1 (Repositórios). Mas, antes da implementação dos testes, foram criados 5 instâncias dos repositórios – 2 repositórios *Fedora*, 1 repositório *DSpace* e 2 repositórios *EPrints*. A Figura 6 ilustra a comunicação entre os repositórios testados. Nesta figura pode-se observar que a informação armazenada no *EPrints.O* é enviada para o *Fedora* e a informação proveniente do *Fedora.O* e do *DSpace.O*, após o processo de agregação, é enviada para os repositórios *Fedora* e *EPrints*.

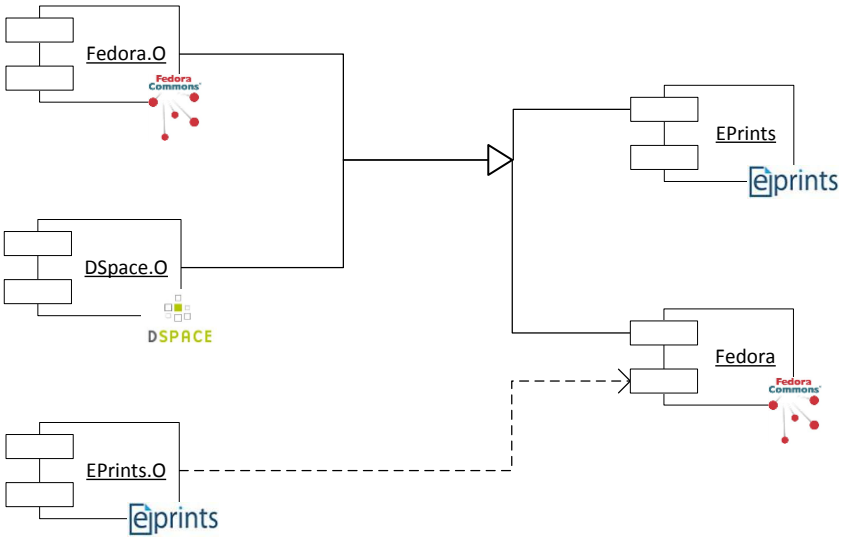

Figura 6. Comunicação entre os repositórios de teste

3.4.1. Procedimento

Nesta simulação, foram adicionados de forma aleatória conteúdos, metadados, versões e relações aos objetos de cada repositório, tendo em conta as características dos mesmos. Assim, foram criados 70 objetos no repositório *Fedora.O* e adicionados conteúdos aos mesmos, dos quais, 10 foram replicados no repositório *DSpace.O*, de forma a testar a semelhança entre objetos na realização da agregação. Aos 10 objetos do *Fedora.O* e replicados no repositório *DSpace.O*, foram adicionados, de forma aleatória, algumas versões dos mesmos. Assim, consegue-se testar a ordenação das versões no momento de agregação dos conteúdos. Desses 70 objetos inseridos no *Fedora.O,* 5 contêm relações. Ao *DSpace.O* foram adicionados mais 10 objetos, e no máximo 10 documentos relacionais. Em relação ao *EPrints*.O, foram criados 100 objetos, com conteúdo

aleatório. O repositório *EPrints* (lado direito da imagem) contem apenas 2 objetos (incluindo conteúdo). Por fim, ao repositório *Fedora* (lado direito) não foram adicionados quaisquer objetos.

Resumidamente tem-se:

Repositório	***Nº Objetos***	***Nº Conteúdos***	***Inclui versões***	***Inclui relações***
Fedora.O	70	>= 70	Sim	Sim
DSpace.O	20	>=20	Não	Sim
EPrints.O	100	100	Não	Não
Fedora	0	-	-	-
EPrints	2	2	Não	Não

Após a inserção e validação dos conteúdos inseridos nos diferentes repositórios, procedeu-se à realização dos testes. Os objetivos principais dos testes eram:

- A verificação do procedimento de cópia de conteúdos de um repositório para outro (*Eprints.O* para *Fedora*).
- A verificação do procedimento de cópia de conteúdos de um repositório para vários (*Fedora.O* para *Fedora* e para o *EPrints*)
- A verificação do procedimento de cópia de conteúdos de um repositório que suporta versões ou relações para um que não as suporte (*Fedora.O* para *EPrints*)
- A verificação do procedimento de agregação, incluindo a ordenação das versões e agregação dos recursos, e posterior inserção dos mesmos nos repositórios de destino (*Fedora.O* e *DSpace.O* para *Fedora* e *EPrints*)

Estima-se que todos os objetos sejam copiados corretamente para os respetivos repositórios.

Assim, o número de objetos resultantes da agregação dos objetos provenientes dos repositórios *Fedora.O* e *DSpace.O* pode variar entre 70 e 90, ou seja, como a geração de conteúdos (texto e metadados) é aleatória, no limite todos os objetos do *DSpace.O* têm um correspondente no *Fedora.O*, ou seja, existem 20 objetos comuns entre os 2 repositórios. Com isto, sobram 50 objetos do *Fedora.O*. Logo, 50 objetos diferentes, mais os 20 semelhantes dará um total de 70 objetos resultantes da agregação. No outro extremo encontram-se todos os objetos dos repositórios (70 do *Fedora.O* mais 20 do *DSpace.O*), ou seja, ou o módulo de agregação não conseguiu detetar semelhanças ou os objetos são díspares. Com isto, o número de objetos nos repositórios de destino podem variar bastante, em diferentes momentos. No final dos testes, no caso do repositório *Fedora*, o número de objetos pode variar entre 170 e 190, ou seja, o número de objetos resultantes da agregação mais o número de objetos provenientes do repositório *EPrints.O* (100 objetos). Se olharmos para o repositório *EPrints*, esse número pode aumentar bastante. No mínimo existirão 72 objetos (70 resultantes da agregação e 2 objetos já existentes no repositório). Contudo, o valor máximo é mais difícil de calcular pois o repositório *EPrints* não suporta, nem versões nem relações e a solução que se encontrou foi criar um objeto por versão e um objeto para armazenar as relações, o número de objetos no repositório final *EPrints* depende das versões e relações provenientes dos resultados da agregação.

Assim, tem-se:

Repositório	*Nº Objetos*	*Nº Conteúdos*	*Inclui versões*	*Inclui relações*
Fedora.O	70	>= 70	Sim	Sim
DSpace.O	20	>= 20	Não	Sim
EPrints.O	100	100	Não	Não
Fedora	170-190	>= 90	Sim	Sim
EPrints	>= 72	>= 92	Sim	Sim

3.4.2. Resultados finais

Para aumentar a fiabilidade dos resultados, os testes foram realizados diversas vezes, e os seus resultados armazenados para posterior análise. Após a análise dos resultados, conclui-se que todos os objetos e conteúdos foram inseridos e armazenados com sucesso nos repositórios de destino, no entanto, cerca de 5% dos objetos, 20% dos documentos de metadados e 30% dos conteúdos não foram bem classificados, ou seja, 5% dos objetos não foram inseridos corretamente nos repositórios de destino, após a análise e agregação dos mesmos. Em relação aos documentos de metadados e aos conteúdos, alguns não foram inseridos nos objetos corretos. Essas falhas estão relacionadas, sobretudo, com a comparação e a ordenação dos conteúdos dos objetos (metadados e conteúdos) resultantes da agregação.

Para melhorar estes resultados seria necessário melhorar as técnicas de análise e ordenação dos conteúdos, utilizadas no módulo agregador, como por exemplo, o aumento do número de metadados para a distinção dos objetos ou a utilização de ferramentas de análise de conteúdos que utilizem dicionários de sinónimos (por exemplo, *thesaurus*) ou comparação de ficheiros (*Matlab, WinMerge, javax.tools.StandardJavaFileManager*).

4. Conclusões

Como se verificou, as bibliotecas sofreram grandes e importantes evoluções desde a origem até aos nossos dias, desde um simples local para armazenamento de obras com diferentes origens até grandes bibliotecas, com grandes quantidades de obras armazenadas.

Contudo, o aparecimento da era digital trouxe novas funcionalidades, aumentando a facilidade de gestão das obras e a sua durabilidade, uma vez que a múltipla utilização das obras não as danifica e as tecnologias que permitem a preservação dos conteúdos, permitem adaptá-los às novas tecnologias que vão surgindo. Contudo, as bibliotecas digitais trouxeram problemas na segurança, sobretudo na facilidade de violação dos direitos de autor – um problema que tem vindo a ser debatido.

Apesar dessas dificuldades e problemas, o número de repositórios digitais que permitem a gestão dos conteúdos existentes nas bibliotecas digitais, tem aumentado, dificultando a escolha de um repositório para determinados fins, e as tecnologias para a interoperabilidade e migração dessa informação entre os diferentes repositórios é limitada. Alguns estudos foram realizados de forma a colmatar as dificuldades de troca de informação entre os diferentes repositórios, como o *OAI-PMH, OAI-ORE, SWORD* ou outros mais específicos, como o estudo para a migração de metadados de uma base de dados para um repositório digital. No entanto, foram encontradas algumas limitações nesses estudos, como por exemplo, o *OAI-PMH* apenas permite a troca de metadados, o *OAI-ORE* apenas permite extrair alguma informação textual dos objetos e as relações existentes entre estes e o *SWORD,* apesar de permitir a troca de conteúdos de um para vários repositórios, não permite agregar conteúdos e necessita que os repositórios suportem este tipo de comunicação.

Para resolver alguns desses problemas e após a análise de alguns dos repositórios existentes no mercado, foi pensado um conjunto de serviços, comuns, que permitam a troca de informação entre diferentes repositórios. Este novo conceito acrescentará valor na gestão das bibliotecas digitais, pois, os gestores das bibliotecas poderão replicar os conteúdos noutros repositórios, poderão agregar conteúdos provenientes de outros repositórios ou, alterar de repositório, de forma mais simples, com menor custo e com uma perda de informação baixa. Assim, foi necessário dividir a aplicação em 2 módulos – um módulo para a extração e inserção de conteúdos dos (nos) repositórios e um módulo para a análise e agregação de conteúdos. Apesar de algumas dificuldades na escolha de técnicas para a comparação de conteúdos e metadados, necessárias no módulo agregador, encontrou-se um conjunto de funções que permite diminuir o número de erros provocados pela agregação dos objetos. Exemplos dessas funções são a análise de semelhança de conteúdos textuais, incluindo metadados, e a análise de semelhança de conteúdos não textuais (vídeos, imagens, etc.) recorrendo à análise de texto, dos metadados desses conteúdos e recorrendo às funções *hash*. Contudo, e após alguns testes apresentados nesta dissertação, em artigos publicados na conferência *WWW/Internet 2012* (Rocha, et al. 2012) (Caixinha, et al. 2012) e realizados no âmbito do projeto EDUCA, uma plataforma *open-source*, em colaboração com a Universidade de Aveiro, que permite a agregação, pesquisa e publicação de conteúdos multimédia de áreas de negócio, científicas educacionais e culturais, a arquitetura e os serviços propostos ainda carecem de melhorias, sobretudo ao nível da agregação de metadados (nas técnicas de análise e agregação utilizadas), na atualização das referências entre os objetos

(utilizadas nas relações e versões), na obtenção dos conteúdos e respetiva informação proveniente das interfaces *web* e, sobretudo, ao nível do número de repositórios testados e da respetiva qualidade dos testes.

Para colmatar estas falhas é necessário construir mais testes, nomeadamente, testes de usabilidade, de carga (por forma a verificar a capacidade de processamento quando a informação flui entre os vários repositórios) e testes num contexto real. Como foram selecionados alguns repositórios, é necessário testar e analisar se a aplicação está preparada para importar e exportar conteúdos para outros repositórios não analisados neste trabalho. Para a obtenção de melhores resultados e reduzindo os erros na importação é necessário melhorar a utilização dos *webcrawlers* e estudar mais ferramentas e protocolos que permitam a extração ou inserção de conteúdos provenientes das interfaces *web* ou através de outros serviços *web,* REST, SOAP, etc.

Como as questões de segurança não foram o ponto mais importante nesta análise, é igualmente necessário verificar e analisar políticas e mecanismos de proteção de conteúdos, necessárias, sobretudo, na troca de informação entre os repositórios recorrendo a redes informáticas desprotegidas, como a *Internet*.

Contudo, fica a ideia de que é possível melhorar e facilitar a gestão das bibliotecas digitais e sobretudo melhorar e aumentar a informação existente nestas. Do resultado deste trabalho, feito em parte no âmbito do projeto EDUCA, foram publicados 2 artigos "*EDUCA Repository Service: API to support different digital repositories*" (Rocha, et al. 2012) e "*Description standards: crosswalk proposal for EDUCA*" (Caixinha, et al. 2012)

Bibliografia

Adam, Nabil R. , e Yelena Yesha. *Introduction. International Journal on Digital Libraries.* 2007.

Ager, Tryg. "Digital Information Organization in Japan." *Architecture and systems*, 1999: 23-25.

Allinson, Julie. *SWORD - Simple Web-service Offering Repository Deposit.* The University of York: British Library, 2009.

Arms, William Y. *Digital Libraries.* MIT Press, 2001.

Bepress. *Digital Commons FAQ.* 2012. http://digitalcommons.bepress.com/faq/ (acedido em 19 de 09 de 2012).

Berners-Lee, T., R. Fielding, U.C. Irvine, e L. Masinter. *Uniform Resource Identifiers (URI).* 1998. http://www.ietf.org/rfc/rfc2396.txt (acedido em 23 de 09 de 2012).

Bordinha, Jose Luis. "The Digital Library: Taking in Account also the Traditional Library." *elpub2002.* Berlin, 2002.

Britannica, Encyclopædia. *Library of Alexandria.* 2012. http://www.britannica.com/EBchecked/topic/14417/Library-of-Alexandria (acedido em 08 de 09 de 2012).

Brown, Mary E. *History and definition of digital libraries.* 2005. http://www.southernct.edu/~brownm/dl_history.html (acedido em 09 de 09 de 2012).

Caixinha, Ana, Joaquim Arnaldo Martins, Jorge Rocha, e Marco Fernandes. "Description standards: crosswalk proposal for EDUCA." *www/internet 2012.* Madrid: IADIS, 2012. 529-532.

Candela, Leonardo, Donatella Castelli, e Pasquale Pagano. "History, Evolution, and Impact of Digital Libraries." IGI Global, 2011.

Casson, Lionel. *Libraries in the Ancient World.* Yale University Press, 2002.

Chau, Do Van. *Challenges of metadata migration in digital repository: a case study of the migration of DUO to Dspace at the University of Oslo Library.* Oslo University College: Faculty of Journalism, Library and Information Science, 2011.

Chisenga, Justin. "Digital Libraries and Virtual Libraries: Definitions, concepts and goals." Addis Ababa, Ethiopia: Avlin, 2003.

Cleveland, Gary. "Digital libraries: Definitions, issues and challenges." Ottawa, Canada, 1998.

CNRI. *Handle System.* 2012. http://www.handle.net/ (acedido em 23 de 09 de 2012).

ContentDM. *CONTENTdm Digital Collection Management Software by OCLC.* 2012. http://www.contentdm.org/ (acedido em 30 de 04 de 2012).

CONTENTdm. *Overview: features, requirements and user experience.* 2012. http://www.oclc.org/contentdm/overview/ (acedido em 19 de 09 de 2012).

Core, VRA. "VRA Core 4.0." 04 de 05 de 2007. http://www.loc.gov/standards/vracore/VRA_Core4_Element_Description.pdf (acedido em 25 de 09 de 2012).

DigitalCommons. *Digital Commons - Open Access Institutional Repository Software.* 2012. http://digitalcommons.bepress.com/ (acedido em 30 de 04 de 2012).

DSpace. *DSpace.* 2012. http://www.dspace.org/ (acedido em 30 de 04 de 2012).

DublinCore. *Dublin Core Metadata Initiative - DCMI.* 2012. http://dublincore.org/ (acedido em 22 de 10 de 2012).

DuraCloud. *DuraCloud.* 2012. http://www.duracloud.org/ (acedido em 22 de 09 de 2012).

—. *Pricing.* 2012. http://www.duracloud.org/pricing (acedido em 22 de 09 de 2012).

—. *Welcome to the DuraCloud Wiki.* 2012. https://wiki.duraspace.org/display/duracloud/DuraCloud (acedido em 22 de 09 de 2012).

EPrints. *EPrints - Digital Repository Software.* 2012. http://www.eprints.org/ (acedido em 30 de 04 de 2012).

Federation, DL. *Metadata encoding and transmission standard: primer and reference manual.* 2010. http://www.loc.gov/standards/mets/METSPrimerRevised.pdf (acedido em 24 de 09 de 2012).

Fedora. *Fedora Commons Repository.* 2012. http://fedora-commons.org/ (acedido em 30 de 04 de 2012).

Fojtu, Andrea. "Open source vs commercial solutions for a long-term preservation in digital repositories." *Systémy pro zpřístupňování VŠKP.* 2009.

Fox, Robert. "Library in the clouds." *OCLC Systems & Services* 25, n.º 3 (2009): 156-161.

Gertz, Janet. "Selection for Preservation in the Digital Age." *Library Resources & Technical Services* 44, n.º 2 (2000): 97-104.

Green, Michelle A., e Mary Jo Bowie. *Essentials of Health Information Management: Principles and Practices.* Delmar, 2010.

Greenstone. *Greenstone.* 2012. http://www.greenstone.org/ (acedido em 17 de 09 de 2012).

Gregorio, J., e B. de hOra. *The Atom Publishing Protocol.* 2007. http://www.ietf.org/rfc/rfc5023.txt (acedido em 24 de 09 de 2012).

Hedstrom, Margaret. "Digital Preservation: A Time Bomb for Digital Libraries." *Computers and the Humanities* 31 (1998): 189–202.

Iannella, Renato. "Open Digital Rights Language (ODRL)." 08 de 08 de 2002. http://odrl.net/1.1/ODRL-11.pdf (acedido em 26 de 09 de 2012).

IEEE. "Draft Standard for Learning Object Metadata." 15 de 07 de 2002. http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf (acedido em 25 de 09 de 2012).

Infokit. "Digital Repositories." 21 de 11 de 2012. http://www.jiscinfonet.ac.uk/infokits/digital-repositories/ (acedido em 15 de 12 de 2012).

Ingram, Rob. *Repository software survey, November 2010.* Repositories Support Project. November de 2010. http://www.rsp.ac.uk/start/software-survey/results-2010/ (acedido em 23 de 04 de 2012).

Intrallect. *FAQS intraLibrary.* 2012. http://www.intrallect.com/index.php/intrallect/support/faqs (acedido em 20 de 09 de 2012).

—. *IntraLibrary.* 2012. http://www.intrallect.com/ (acedido em 20 de 09 de 2012).

JeromeDL. *JeromeDL - e-Library with Semantics.* 2012. http://www.jeromedl.org/ (acedido em 30 de 4 de 2012).

JISC. *Digital Repositories: Helping universities and colleges.* Agosto de 2005. http://www.jisc.ac.uk/uploaded_documents/JISC-BP-Repository(HE)-v1-final.pdf (acedido em 15 de Dezembro de 2012).

Jorum, Team. *Report on Open Source Learning Object Repository Systems.* Jorum, 2005.

Krasner-Khait, Barbara. *Survivor: The History of the Library.* 2001. http://www.history-magazine.com/libraries.html (acedido em 08 de 09 de 2012).

Krishnamurthy, M. “Open Access, open source and digital libraries.” *Program: electronic library and information systems* 42 (2008): 48-55.

Krottmaier, Harald. “The Future of Digital Libraries.” 2004.

Kruk, Sebastian Ryszard, Tomasz Woroniecki, Adam Gzella, Maciej Dabrowski, e Bill McDaniel. “The anatomy of a Social Semantic Digital Library.” *European Semantic Web Conference.* 2007.

Lee, Sul H., University of Oklahoma. Libraries, e University of Oklahoma Foundation. *Economics of Digital Information: Collection, Storage, and Delivery.* Routledge, 1997.

Lesk, Michael. *Understanding Digital Libraries (Second Edition).* San Francisco: Elsevier, Inc, 2005 .

Lewis, Stuart. *SWORD.* 2012. http://swordapp.org/about/ (acedido em 23 de 09 de 2012).

Lucene. *Apache Lucene.* Apache. 2012. http://lucene.apache.org/core/ (acedido em 05 de 01 de 2012).

Madalli, Devika P., Sunita Barve, e Saiful Amin. “Digital Preservation in Open-Source Digital Library Software.” *The Journal of Academic Librarianship* 38 - 3 (2012): 161-164.

Marques, Amélia Maria Nunes, e Sílvia Raquel da Silva Maio. “Repositórios Institucionais.” 2007. http://repositoriosdigitais.web.simplesnet.pt/PDF%27S/Artigo%20%20Repositorios%20Institucionais.pdf (acedido em 15 de Dezembro de 2012).

Mulvey, Bret. *Hash Functions.* 2007. http://home.comcast.net/~bretm/hash/ (acedido em 24 de 10 de 2012).

NISO. “Information Retrieval (Z39.50): Application Service Definition and Protocol Specification.” 27 de 11 de 2002. http://www.loc.gov/z3950/agency/Z39-50-2003.pdf (acedido em 24 de 09 de 2012).

Nottingham, M., e R. Sayre. *The Atom Syndication Format.* 2005. http://www.ietf.org/rfc/rfc4287.txt (acedido em 24 de 09 de 2012).

OAI-ORE. *OAI-ORE.* 2012. http://www.openarchives.org/ore/ (acedido em 24 de 09 de 2012).

OIA-PMH. *OIA-PMH.* 2012. http://www.openarchives.org/pmh/ (acedido em 24 de 09 de 2012).

Open Source Systems. “Open source systems.” 13 de 8 de 2012. https://wiki.albany.edu/download/attachments/39428999/IR_Comparison_KB9.xls (acedido em 22 de 09 de 2012).

Patil, M S., e Satish Kanamadi. “Digital Library Open Source Software: A Comparative Study.” 2008.

Peterson, Michael. *What is a Digital Archive?* Junho de 2011. http://www.ltdprm.org/reference-model/what-is-an-archive (acedido em 15 de Dezembro de 2012).

Pirounakis, George, e Mara Nikolaidou. “Comparing Open Source Digital Library Software.” In *Handbook of Research on Digital Libraries: Design, Development, and Impact*, 51-60. IGI Global, 2009.

Rajashekar, T.B. *Digital Library and Information Services in Enterprises: Their Development and Management.* 2006. http://www.ncsi.iisc.ernet.in/raja/is214/is214-2006-01-04/topic-1.htm (acedido em 09 de 09 de 2012).

Ratzan, Lee. *Understanding Information Systems.* ALA Editions, 2004.

Reddy, Raj. “Digital Information Organization in Japan.” *Global digital libraries: building the infrastructure*, 1999: 5-12.

Repositories Support Project. *Commercial Repository Solutions.* Repositories Support Project. 8 de 2011. http://www.rsp.ac.uk/documents/briefing-papers/2011/CommercialRepositorySolutions_RSP_0811.pdf (acedido em 26 de 4 de 2012).

Rocha, Jorge, Ana Caixinha, Joaquim Arnaldo Martins, e Marco Fernandes. "EDUCA Repository Service: API to support different digital repositories." *www/internet 2012.* Madrid, Spain: IADIS, 2012. 499-523.

Rosensweig, Aviva. "759 Portfolio Home Page." 2008. http://www.upstarts.net/759/11.htm (acedido em 20 de 04 de 2012).

Samsung. *23" 550 Series LED Monitor.* 2012. http://www.samsung.com/us/computer/monitors/LS23A550HS/ZA-features (acedido em 09 de 09 de 2012).

Seadle, Michael, e Elke Greifeneder. "Defining a digital library." Humboldt University, Berlin: Emerald, 2007. 169-173.

Semple, Najla. "Digital Repositories." 4 de 04 de 2006. http://www.dcc.ac.uk/resources/briefing-papers/introduction-curation/digital-repositories (acedido em 15 de 12 de 2012).

Shustitskiy, Maria. "Collaboration of digital libraries." Vienna University of Economics and Business Administration, Vienna, 2004.

Staikos, Konstantinos Sp. *The history of the library in western civilization: the Renaissance - from Petrarch to Michelangelo.* New Castle: Oak Knoll Press, 2012.

Technology, University Information. *Digital Libraries and Repositories - UIT Encyclopedia for Teaching with Technology.* 2007. https://wikis.uit.tufts.edu/confluence/display/UITKnowledgebase/Digital+Libraries+and+Repositories (acedido em 15 de 12 de 2012).

University of Nottingham, UK. *The Directory of Open Access Repositories.* 2012. http://www.opendoar.org/ (acedido em 27 de 09 de 2012).

Visly, Alexander. *Bibliotecas digitais.* 2001. http://www.elbib.ru/content/journal/2001/200101/vislii/vislii.ru.htm (acedido em 09 de 09 de 2012).

Vitiello, Giuseppe. "Identifiers and Identification Systems." *D-Lib Magazine* 10 (2004).

Waddington, Simon, Jun Zhang, Gareth Knight, Mark Hedges, Jens Jensen, e Roger Downing. "Kindura: Repository services for researchers based on hybrid clouds." *Journal of Digital Information* 13 (2012).

Warr, Hanadashisha, e Dr. P. Hangsing. "Open source digital library software: a literature review." Imphal, 2009.

Whitehouse, Dr David. *Library of Alexandria discovered.* 2004. http://news.bbc.co.uk/2/hi/3707641.stm (acedido em 08 de 09 de 2012).

Witten, Ian H., David Bainbridge, e David M. Nicols. *How to Build a Digital Library.* San Francisco: Morgan Kaufmann Publishers, 2003.

Wright, Cheryl D. "Digital Library Technology Trends." Sun Microsystems, Inc., 2002.

Zurndorfer, Harriet Thelma. *China Bibliography: A Research Guide to Reference Works about China Past & Present.* Leiden, Netherlands: Brill, 1995.

Anexos

Anexo A - Distribuição geográfica de alguns repositórios digitais

O *OpenDOAR* é uma entidade que regista a localização e outra informação de repositórios *open-source*. À data (University of Nottingham 2012) existem 2207 repositórios, em 1848 organizações académicas dos ”4 cantos do mundo”. A maior parte destes repositórios estão na Europa e na América do Norte, como ilustra a imagem seguinte (Figura 7 – lado esquerdo).

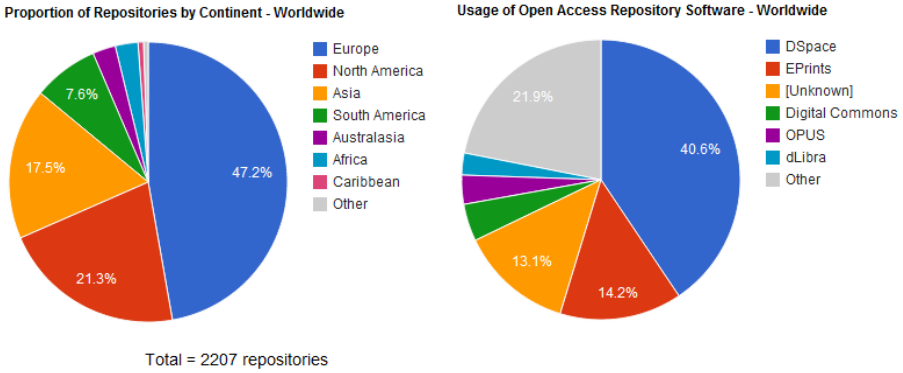

Figura 7. Repositórios no Mundo

Repositórios no Mundo

Dos repositórios registados na entidade *OpenDOAR*, 2207 repositórios, o *DSpace* é o repositório mais utilizado. No entanto, como existem organizações registas na *OpenDOAR* sobre as quais se desconhece o repositório utilizado (13,1%) ou organizações que utilizam outros tipos de repositórios (21,9%), poderão existir outros repositórios que se aproximem destes valores (Figura 7 – lado direito). Os repositórios, na maior parte das vezes, contêm artigos de revista, teses e dissertações, em diferentes áreas (multidisciplinares), seguindo-se Medicina e História, como se pode observar na imagem seguinte (Figura 8).

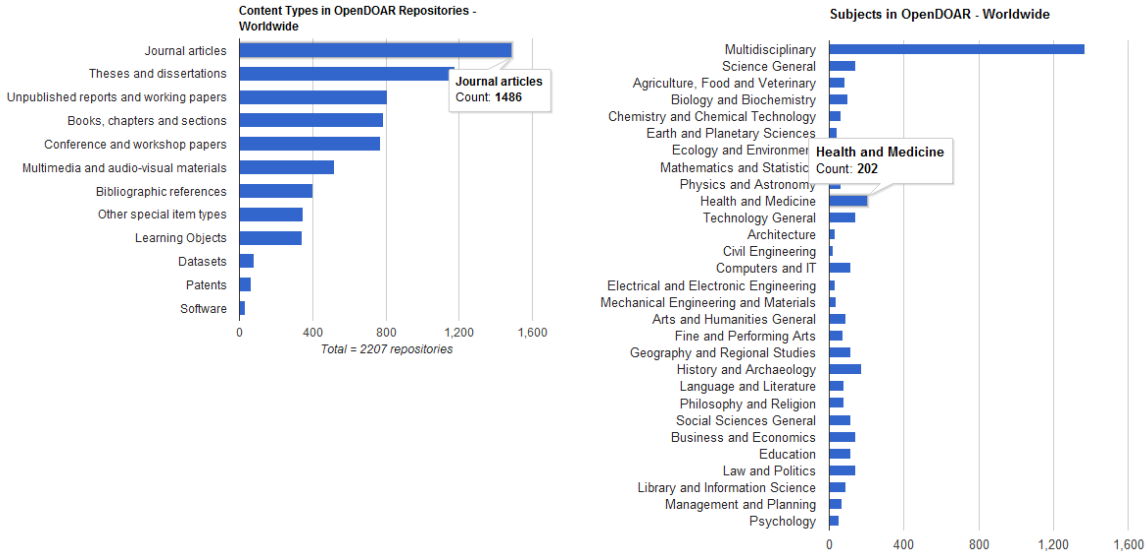

Figura 8. Tipo de conteúdo e área cientifica mais utilizados nos repositórios

Repositórios em Portugal

Em Portugal, de um total de 40 repositórios registados, o mais utilizado é o *DSpace*, com 85% dos repositórios a utilizar esta forma de armazenamento (Figura 9 - lado esquerdo). Os portugueses utilizam os repositórios digitais para armazenamento de artigos, teses, dissertações e documentos de conferências, como se observa na imagem seguinte (Figura 9 - lado direito).

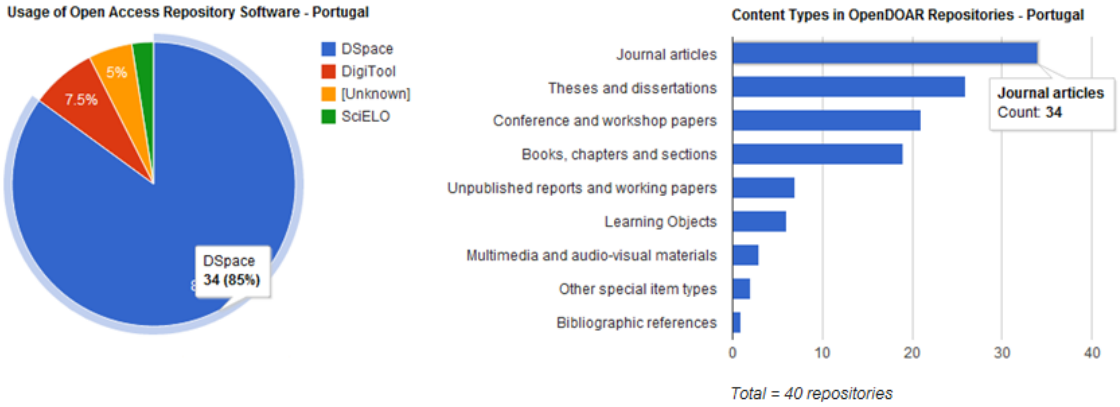

Figura 9. Repositórios em Portugal

Anexo B - Identificadores

Todos os livros, revistas, jornais e outra documentação armazenada nas bibliotecas tradicionais, começaram a ser identificadas no século VI, durante a dinastia *Han* (Zurndorfer 1995). No entanto, no século XX, foram introduzidas classificações normalizadas, como o *ISBN* e o ISSN.

O *ISBN* – *International Standard Book Number* – criado em 1970 pela ISO, é um identificador numérico para referenciar um livro a nível mundial. Este identificador, atualmente, é composto por 13 dígitos e subdividido em 3 partes – primeira parte corresponde a um identificador do livro a nível mundial, a segunda parte, é um identificador do livro no país e por fim, o terceiro nível é um identificador do livro no ambiente do editor.

Figura 10. Nomenclatura do ISBN[2]

A norma *ISSN* – *International Standard Serial Number* – criada em 1975 pela *ISO*, é um número único de 8 dígitos que identifica uma publicação (incluindo publicações eletrónica), ou seja, é o número de série de uma obra. Enquanto o *ISBN* identifica um livro, o *ISSN* identifica um exemplar desse livro.

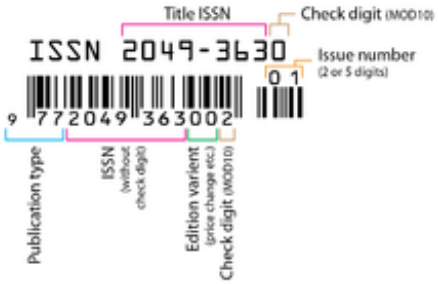

Figura 11. Nomenclatura do ISSN[3]

Existem muitos outros códigos que pretendem identificar de forma única uma obra. Exemplos desses códigos são o *ISRC* – *International Standard Recording Code* – identificar uma gravação de vídeo/áudio, o *ISMN* – *International Standard Music Number* – utilizado para identificar uma obra musical, ou outros identificadores não relacionados com bibliotecas, como *ISWN* – *International*

[2] http://en.wikipedia.org/wiki/International_Standard_Book_Number
[3] http://en.wikipedia.org/wiki/International_Standard_Serial_Number

Standard Wine Number – utilizado na indústria dos vinhos, com o mesmo propósito dos restantes identificadores. Existe uma variedade de identificadores dentro e fora do âmbito das bibliotecas, todos com o mesmo propósito, distinguir um produto e proporcionar às pessoas e máquinas uma identificação mais rápida do produto e possíveis fraudes (Vitiello 2004).

No ambiente das bibliotecas digitais os identificadores proporcionam aos utilizadores um acesso mais rápido a cada objeto. Os identificadores são uma referência para os objetos digitais da biblioteca e respetivas citações, de forma a melhorar a gestão de acessos e armazenamento por um longo período de tempo. Existem vários identificadores no âmbito das bibliotecas digitais, no entanto, apenas os indentificadores utilizados nos repositórios digitais descritos neste trabalho serão apresentados – *URL, URI, PURL e CNRI Handle System*.

O *URL* – *Uniform Resource Locator* – é o endereço típico utilizado na *web*, que pretende identificar uma localização de conteúdo digital na *Internet*, permitindo que exista um número variado de aplicações na *Internet*. Contudo, nas bibliotecas digitais, a preservação do identificador ao longo do tempo causa problemas – falta de persistência – ou seja, se a localização de um objeto digital for alterada o identificador – *URL* – fica indisponível. Por esta razão foram criados outros indentificadores persistentes, como o *URN* e o *PURL*.

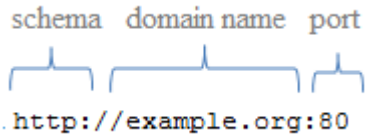

Figura 12. Exemplo de um URL

O *URN* – *Uniform Resource Name* – é um identificador complementar à *URL*, com o objetivo de resolver o problema de falta de persistência da *URL*. É um identificador global, único e persistente, que identifica uma cópia do objeto digital, definida por uma lista de *URLs*. Em comparação com a vida real, o URN representa o nome de uma pessoa, enquanto o *URL* diz onde essa pessoa mora, ou seja, "quem" – *URN* vs. "onde" – *URL*.

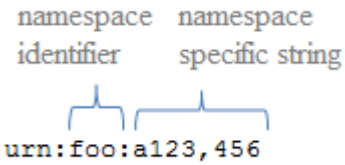

Figura 13. Exemplo de um URN

O *PURL* – *Persistent Uniform Resource Locator* – oferece um identificador globalmente único e, como contem *software* para resolver nomes de domínio e manipular *URLs*, permite que alterações à localização interna dos objetos digitais, não se propague para o exterior, ou seja, no seguinte exemplo (Figura 14) se a localização de um objeto for alterada, o endereço para esse objeto mantem-se, caso a gestão dos endereços seja bem efetuada (Arms 2001).

Figura 14. Exemplo de um identificador PURL

URI – Uniform Resource Identifier – é um conjunto de caracteres que identificam um nome e/ou recurso digital. Normalmente é composto, como ilustra a imagem seguinte (Figura 15), pelo *URL* – contem a localização do objeto digital na rede; pelo *URN* – contem a identificação global, única e persistente do objeto digital; ou por ambos (Berners-Lee, et al. 1998).

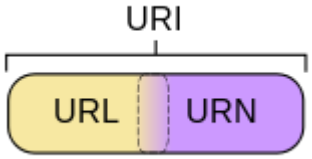

Figura 15. Relação entre URI, URL e URN

O *CNRI Handle System* (CNRI 2012) é outra tecnologia utilizada para identificar de forma única um objeto digital. Desenvolvido pela *CNRI – Corporation for National Research Initiatives* – é uma tecnologia para a gestão e resolução de identificadores persistentes de objetos digitais existentes nos repositórios digitais e na *Internet*. Possui um conjunto de identificadores, designados por *Handles*, armazenados numa máquina e que permitem fazer a tradução entre a localização de um objeto digital e o respetivo objeto. Evita que as alterações à informação do objeto digital sejam propagadas para o exterior.

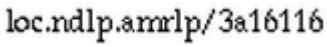

Figura 16. Exemplo de um identificador *Handle System*