



IMPLEMENTAÇÃO DE UM PROTÓTIPO DE BIBLIOTECA DIGITAL COM ECM E SOA

Andrey Prezotto do Prado

Recall do Brasil Ltda., Jundiaí, SP, Brasil,
andreypp@gmail.com

Peter Jandl Junior

Centro Universitário Padre Anchieta
pjandl@anchieta.br

RESUMO

Este trabalho analisa os resultados da implementação de um protótipo de biblioteca digital empregando conceitos de *Enterprise Content Management* (ECM) e *Service Oriented Architecture* (SOA), os quais permitiram obter um sistema mais flexível e consistente do ponto de vista de armazenamento, classificação e recuperação de documentos.

Palavras-Chave: biblioteca digital; arquitetura orientada a serviços; gestão corporativa de conteúdo; serviços web; desenvolvimento de software.

ABSTRACT

This work analyzes the results of the implementation of a digital library prototype that uses Enterprise Content Management (ECM) and Service Oriented Architecture (SOA) concepts, which allowed to obtain a more flexible and consistent system in terms of storage, classification and document retrieval.

Keywords: digital library; service oriented architecture (SOA); enterprise content management (ECM); web services; software development.



1. Introdução

Historicamente, as bibliotecas possuem um papel indiscutivelmente importante para a sociedade. Segundo Teixeira, Silva & Marinho (2011, p.2), elas atuam como organismo responsável pela democratização do acesso ao conhecimento e à informação. Nesta perspectiva, as bibliotecas digitais são igualmente importantes para divulgação de trabalhos científicos e acadêmicos de qualquer instituição educacional ou de pesquisa, pois são fontes confiáveis de conteúdo que contribuem no processo de formação, educação e suporte de um grande número de usuários geograficamente dispersos.

Entretanto, muitos dos sistemas existentes apresentam uma série de deficiências em função de projetos centrados nos aspectos tradicionais de arquivamento de bibliotecas físicas, ou seja, limitações relacionadas à forma como os documentos digitais ficam armazenados no sistema associadas ao uso de métodos ineficientes de classificação que tornam a recuperação de documentos mais difícil ou, algumas vezes, ineficaz.

A implementação de um protótipo de biblioteca digital empregando conceitos de *Enterprise Content Management (ECM)* e *Service Oriented Architecture (SOA)* pretendeu obter um sistema mais flexível e mais consistente do ponto de vista de armazenamento, classificação e recuperação de documentos.

2. Bibliotecas Digitais

As *bibliotecas digitais* nasceram da evolução natural do sistema bibliotecário, que passou a usar a computação não apenas como ferramenta de apoio e aperfeiçoamento dos seus mecanismos de catalogação e organização de acervo, mas como uma tecnologia para disseminar de conhecimento. Como um serviço de informação, as bibliotecas digitais permitem superar muitas das limitações relacionadas ao uso de acervos físicos tradicionais e, ao mesmo tempo, possuem os mesmos objetivos: organizar, manter e disponibilizar informação aos seus usuários. Segundo Leiner:

Uma biblioteca digital é a coleção de serviços e a coleção de objetos de informação, sua organização, estrutura e apresentação, que suporta o relacionamento dos utilizadores com os objetos de informação, disponíveis direta ou indiretamente via meio eletrônico/digital (1988, apud, ASSOREIRA & MOURÃO, 2001).



Uma outra definição para bibliotecas digitais é:

(...) coleções focadas de objetos digitais que incluem texto, vídeo e áudio, junto com métodos para acesso e recuperação, seleção, organização e manutenção. [tradução livre] (WITTEN, BAINBRIDGE & NICHOLS, 2009).

Em geral, as bibliotecas apresentam características como: acesso remoto ao acervo digital; ausência das restrições ligadas aos documentos físicos; capacidades para armazenar mídias diferentes e bases de dados completas; além da disponibilização de métodos de pesquisa e recuperação eficientes.

Exemplos dos principais sistemas de bibliotecas digitais *open-source* existentes são: Invenio, DSpace, Greenstone e EPrints.

2.1 CERN Invenio

É um pacote de software de código aberto que permite o gerenciamento de documentos digitais em um repositório de natureza institucional. Foi desenvolvido pelo *Conseil Européen pour la Recherche Nucléaire* (CERN), daí sua primeira denominação *CERN Document Server* (CDSware), para gerenciar sua enorme biblioteca digital, uma das maiores do mundo. A partir de 2006, esse pacote de software recebeu o nome de CDS Invenio, simplificado mais recentemente para Invenio.

A maior parte de seu código é escrita em Python, mas sua arquitetura é baseada numa combinação do Apache HTTP server, *Web Server Gateway Interface* (WSGI) e o banco de dados MySQL. É capaz de operar apenas em sistemas *Unix-like*, predominantemente distribuições derivadas do Red Hat Enterprise Linux 5 (RHEL5) ou do Debian (INVENIO, 2013).

As principais características do Invenio são: organização dos documentos em coleções navegáveis normais ou virtuais; portais customizáveis para cada coleção; mecanismo próprio de pesquisa configurável de alto desempenho; inclusão de metadados no formato MARC (padrão XML da biblioteca do congresso americano), que possibilita grande adaptabilidade das informações armazenadas; manipulação de artigos, livros, teses, fotos, vídeos e outros elementos; personalização da interface de usuário; além de ferramentas para inclusão, submissão, classificação, indexação e disseminação de conteúdos (INVENIO, 2013).

A Figura 1 mostra a aplicação do Invenio no próprio CERN e também no InfoScience, um projeto da Escola Politécnica Federal de Lausanne (EPFL).

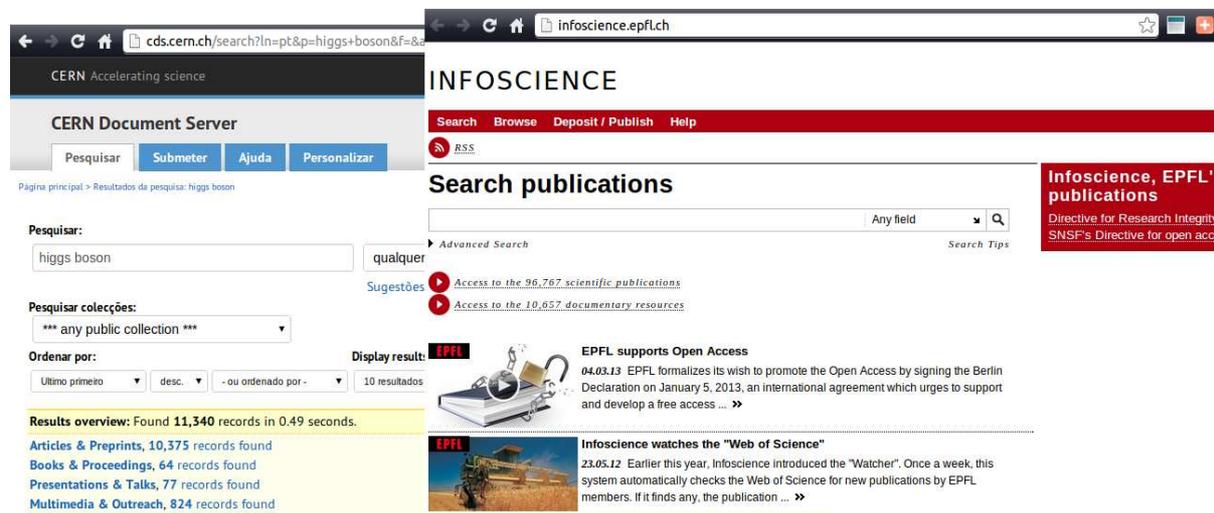


Figura 1 – Aplicação do Invenio no CERN e no projeto InfoScience (EPFL).

O conjunto de características do Invenio são suas vantagens mais significativas, além do fato de suportar o *Open Archives Initiative Protocol for Metadata Harvesting* (OAI-PMH) (OAI, 2013). Mas também apresenta várias desvantagens: sua operação efetiva se restringe a poucas plataformas operacionais e possui dependência com muitos módulos, o que torna a instalação complexa em servidores não dedicados. Além disso, sua arquitetura sofisticada dificulta o uso de sua infraestrutura em outros serviços de gerenciamento de documentos e sua interoperabilidade é restrita.

2.2 DuraSpace DSpace

É outro pacote de software *open-source*, desenvolvido inicialmente por meio de parceria do *Massachusetts Institute of Technology* (MIT) e os laboratórios da Hewlett Packard. Os resultados da iniciativa originaram a fundação DuraSpace que atualmente mantém o projeto DSpace.

Construído sobre a plataforma Java e com uso do Apache Tomcat, o DSpace é um conjunto de aplicações web e programas utilitários capazes de organizar, manter e publicar um acervo digital de documentos de várias naturezas, bem como seus metadados. Sua arquitetura modular possibilita a criação de grandes repositórios multidisciplinares, utilizando banco de dados Oracle ou PostgreSQL, que podem ser expandidos além dos limites



institucionais. Seu projeto permite a manipulação de qualquer formato eletrônico, bem como o agrupamento de arquivos e suas descrições em itens que podem ser organizados em coleções. Uma das preocupações de seu projeto é a preservação de funcionalidades associadas aos formatos usados, mesmo com a evolução destes (DSpace, 2013).

A Figura 2 ilustra o DSpace em funcionamento na *National Air and Space Agency* (NASA) e na *Brasiliiana*, uma biblioteca digital da Universidade de São Paulo (USP).

The image shows two side-by-side screenshots of the DSpace digital library interface. The left screenshot is from the NASA website (search.nasa.gov) and shows search results for 'mario de andrade'. The right screenshot is from the Brasiliiana USP website (www.brasiliiana.usp.br) and shows search results for 'mario de andrade' with a list of books, including 'Como e porque sou romancista' and 'Ao Imperador: cartas de Erasmo'.

Figura 2 – Aplicação do DSpace na NASA e Brasiliiana (USP).

Constituem suas maiores vantagens a sua operação em múltiplos ambientes operacionais; o modelo de objetos e sua gerência; o controle de acesso; e sua interoperabilidade. Também suporta o OAI-PMH. Suas desvantagens são o limitado suporte a idiomas e menor facilidade de customização (TRAMBOO, HUMMA, SHAFI & GUL, 2012; PYROUNAKIS & NIKOLAIDOU, 2009; PATIL & KANAMADI, 2008).

No geral, é considerada a melhor alternativa de software para bibliotecas digitais. Apenas no Brasil existem mais de 60 instituições registradas como usuárias do DSpace, entre elas: Embrapa, FGV, Inmetro, MEC, Senado Federal, UFRGS e USP.

2.3 Greenstone

Desenvolvida pela Universidade de Waikato (Nova Zelândia) em parceria com a UNESCO e a ONG belga Human Info, o projeto Greenstone é um sistema de software de código aberto destinado a construção de bibliotecas digitais compostas de um grande número de documentos digitais pesquisáveis e organizados em coleções.

É implementado em C++; faz uso extensivo da *Standard Template Library* (STL); mas também utiliza Perl e Java (para a interface gráfica do bibliotecário). A sua interface web é baseada no *Common Gateway Interface* (CGI) e pode utilizar como infraestrutura o Apache HTTP server ou Microsoft *Internet Information Server* (IIS), mas seu banco de dados constitui uma solução própria. Seu projeto é modular e inclui ferramentas, relativamente simples, para inclusão, catalogação e publicação de documentos e seus metadados em diversos formatos, como MARC, DSpace ou BibTex (GREENSTONE, 2013).

A Figura 3 mostra os sites da Biblioteca Nacional do Uruguai e da Bolsa de Cereais de Buenos Aires, que operam com o Greenstone.



Figura 3 – Aplicação do Greenstone na Biblioteca Nacional do Uruguai e da Bolsa de Cereais de Buenos Aires.

Outras características interessantes do Greenstone são seu suporte para OAI-PMH e a possibilidade de uso para compor uma biblioteca digital local e autônoma, que pode ser distribuída em um CD-ROM ou DVD-ROM (GREENSTONE, 2013).

As maiores vantagens do Greenstone são sua simplicidade; facilidade relativa de sua instalação; operação em ambientes Windows, Unix/Linux ou Mac OS-X; suporte às coleções; suporte a múltiplos idiomas; e possibilidades de customização. Por outro lado, exibe várias desvantagens: usa uma tecnologia web considerada obsoleta (CGI); seu banco de dados não é uma solução de mercado; o modelo de objetos e metadados é limitado; e o controle de acesso é deficiente (TRAMBOO, HUMMA, SHAFI & GUL, 2012; PYROUNAKIS & NIKOLAIDOU, 2009; PATIL & KANAMADI, 2008).

2.4 EPrints

Resultado do esforço da Universidade de Southampton (Inglaterra), é uma plataforma de software livre que constitui um repositório genérico para preservação e divulgação de documentos digitais. É usada frequentemente para catalogação e divulgação de materiais de pesquisa científica.

Foi construída em Perl, usando o Apache HTTP server como infraestrutura web, pode operar em ambientes Windows ou Unix/Linux com uso de banco de dados Oracle, MySQL, ou PostgreSQL. Sua interface web é bastante adequada para as tarefas de inclusão, organização e recuperação de documentos dos formatos mais comuns, incluindo seus metadados; ou seja, é um sistema de fácil utilização (EPRINTS, 2013).

A Figura 4 mostra aplicações típicas do EPrints como bibliotecas digitais de teses na Universidade de Glasgow e Caltech.

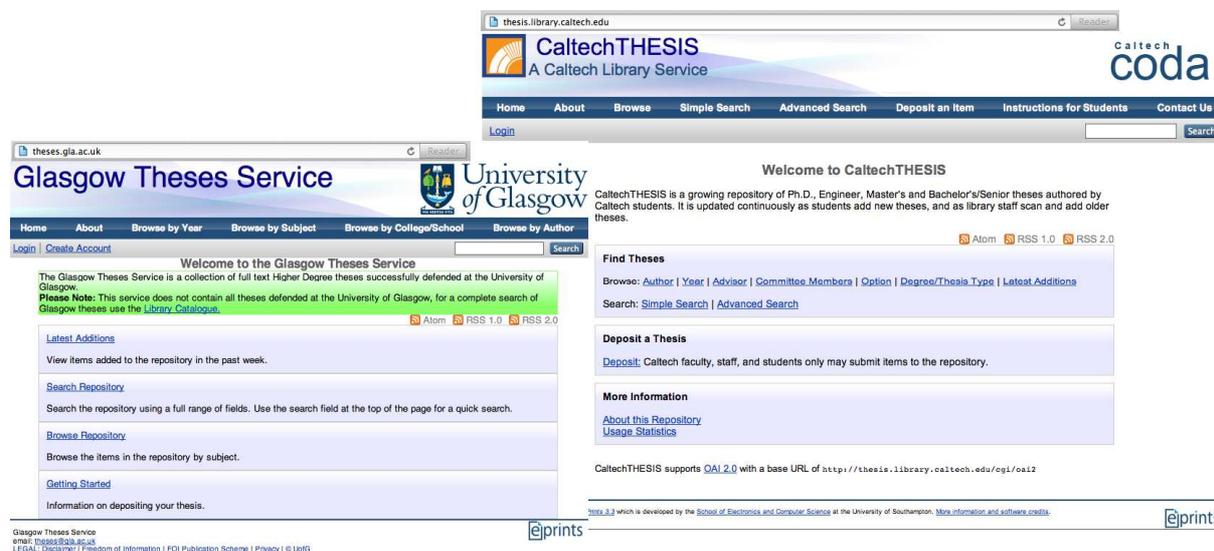


Figura 4 – Aplicação do EPrints na Universidade de Glasgow e Caltech.

As vantagens exibidas pelo EPrints são grande interoperabilidade; suporte ao OAI-PMH; usabilidade de suas interfaces; métodos de pesquisa de documentos; suporte a múltiplos idiomas; e gerenciamento de seus objetos. Em contrapartida, suas desvantagens são o controle de acesso deficiente, limitações no armazenamento de metadados e seu modelo de objetos (TRAMBOO, HUMMA, SHAFI & GUL, 2012; PYROUNAKIS & NIKOLAIDOU, 2009; PATIL & KANAMADI, 2008).



2.5 Considerações Gerais

Um dos principais aspectos na análise e seleção de um sistema de software para constituir uma biblioteca digital é sua interoperabilidade, quesito onde algumas das soluções analisadas deixam a desejar. Muito das restrições de interoperabilidade e também das possibilidades (ou dificuldades) de customização são originadas da plataforma de programação escolhida.

Cada um dos sistemas vistos usa uma base de programação distinta: Python no Invenio; Java no DSpace; C++ no Greenstone; e Perl no EPrints. Com isso, sua manutenção, expansão ou integração com outros sistemas acabam por depender, em maior ou menor grau, da plataforma inicialmente escolhida.

O uso de *web services* como uma plataforma de programação ainda mais neutra e simultaneamente, adequada a operação na web constituiu uma das motivações deste trabalho na construção de um protótipo de biblioteca digital que aliasse grande modularidade (obtida com uma arquitetura orientada a serviços), neutralidade de plataforma e características de sistemas de gestão de conteúdo.

3. Enterprise Content Management (ECM)

A *Association for Information and Image Management* (AIIM) define o termo *Enterprise Content Management* (ECM) como:

As estratégias, métodos e ferramentas utilizadas para capturar, gerenciar, armazenar, preservar e fornecer conteúdo e documentos relacionados aos processos organizacionais. ECM abrange a gestão da informação dentro de todo o escopo de uma empresa não se importando se a informação está na forma de um documento em papel, um arquivo eletrônico, banco de dados, de um fluxo de impressão, ou até mesmo um e-mail (AIIM, 2011).

Estas estratégias visam facilitar o gerenciamento do ciclo de vida da informação corporativa, ou seja, o controle desde sua criação, incluindo seu arquivamento, utilização, publicação e eventual descarte. As soluções ECM são compostas por conceitos e tecnologias descritos pela AIIM e também por outros autores como Kampffmeyer (2006), onde se destacam o uso de componentes para gerência do ciclo de vida dos documentos corporativos.



3.1 Componente de Captura

Formado por tecnologias especializadas em realizar o tratamento das entradas de informação, independente de sua origem física ou eletrônica. Os documentos de origem física (impressos) são digitalizados para que seu conteúdo seja capturado e armazenado digitalmente no sistema, enquanto os de origem eletrônica são simplesmente armazenados. A partir dos documentos eletrônicos, é possível aplicar algoritmos de análise e processamento capazes executar o reconhecimento de códigos de barras, o *reconhecimento óptico de caracteres* (OCR) ou mesmo o processamento de imagens que possibilitam a extração de dados do documento original e a obtenção de textos editáveis, metadados ou mesmo bases de dados.

3.2 Componente de Retenção

Corresponde aos processos e meios de armazenamento, segurança, *backup* e acesso dos documentos digitais de uma solução ECM. A implementação geral é na forma de repositório de documentos, cuja estruturação lógica é capaz de fornecer um canal de comunicação único com os demais componentes e também suportar grandes volumes de informação, sem que o desempenho do sistema seja prejudicado pela quantidade de arquivos armazenados ou pelo número de acessos.

3.3 Componente de Gerenciamento

Este componente interliga os demais por meio de uma interface padronizada que possibilita ao usuário interagir com todos os elementos do sistema. Geralmente, é composto por: *controle de acesso*, que restringe o acesso aos conteúdos armazenados; *controle de versão*, que administra a criação e utilização de uma ou mais versões de um mesmo documento; *mecanismos de pesquisa*, para a busca de informações sobre os documentos armazenados no sistema; *ferramentas de visualização*, que auxiliam o uso estruturado dos documentos; e *manutenção de metadados*, que suportam a entrada, consulta e atualização de dados referentes aos documentos, como título, autores, resumo, versão e data de criação e modificação.



3.4 Componente de Workflow

Este componente, definido como pela *Workflow Management Coalition* (WfMC, 1999), deve suportar a automação, total ou parcial, de um processo de negócios, no qual documentos são passados de um participante para outro através de ações validadas dentro do conjunto de regras estabelecido para tal processo.

3.5 Componente de Distribuição

Oferece métodos e ferramentas para exportação de dados (p.e., como arquivos XML), transformação de formatos de arquivo (como geração de PDF) e publicação de conteúdo (em mídias como CD e DVDs ou na Internet), provendo também a integração de sistemas.

4. Arquitetura Orientada a Serviços

A *arquitetura orientada a serviços* (SOA) pode ser descrita como um modelo arquitetural que objetiva aumentar a eficiência, a agilidade e a produtividade de uma empresa, disponibilizando sua lógica de negócios através de serviços (ERL, 2007).

Uma tarefa ou função bem definida, que pode ser executada isoladamente de outras tarefas associadas, pode ser classificada como um *serviço*, ou seja, “*cada serviço é atribuído ao seu próprio contexto funcional distinto e é composto por um conjunto de capacidades relacionadas a este contexto*” (ERL, 2007, p.39); onde capacidades são funcionalidades fornecidas por um serviço para um consumidor.

Dentro de seu contexto, um serviço deve ser desenvolvido para representar precisamente a lógica de negócios como um recurso reutilizável, independente do processo que o consome. Para atingir esse objetivo, esta lógica deve ser projetada conforme o paradigma de orientação a serviços, semelhante ao conceito de engenharia de software chamado *separação de interesses*, o qual afirma que um problema é melhor resolvido se dividido em problemas menores.

Isso permite dividir a lógica de um processo em pequenos serviços encarregados de resolver um único problema, sem preocupação com o problema maior onde se inserem. Como consequência esse serviço se torna independente e oportuniza a reutilização de sua lógica na resolução de outros problemas.



Assim como um sistema de biblioteca digital pode se beneficiar dos conceitos de ECM, ambos podem ser desenvolvidos baseados no paradigma SOA, no qual um componente pode ser representado por um serviço responsável por disponibilizar suas capacidades como um recurso reusável aos demais elementos do sistema e capaz de executar seu papel de forma autônoma.

5. Análise e Levantamento de Requisitos

Foi realizada uma análise para identificar os requisitos essenciais para a implementação de uma biblioteca digital comuns as características de sistemas de ECM. A análise dos requisitos é um processo importante da engenharia de software que define as funcionalidades e o escopo do sistema, pois, como afirma Sommerville, “*os requisitos para um sistema são as descrições dos serviços prestados pelo sistema e suas restrições operacionais*” (2006, p.118).

Tabela 1 – Características ECM essenciais numa biblioteca digital.

Característica	Descrição
Repositório de documentos	Estrutura para armazenamento e recuperação eficientes dos documentos digitais (arquivos). Composto por (1) um diretório raiz, previamente configurado, que serve como base para (2) uma hierarquia lógica de subpastas criadas dinamicamente a partir do uso do sistema.
Métodos para classificação e recuperação de documentos	Classificação de documentos baseado na associação de uma ou mais palavras-chave (<i>tags</i>), previamente cadastradas pelo administrador do sistema, o que possibilita ao usuário recuperar os documentos associados de acordo com sua necessidade.
Armazenamento de metadados de documentos	Todo documento gerenciado é composto pelo (1) arquivo de conteúdo em formato digital; e (2) seus metadados (informações complementares). De acordo com natureza do documento, um conjunto distinto de metadados pode ser aplicado, permitindo a configuração de diferentes modelos que deverão ser escolhidos e preenchidos pelo usuário no momento de inclusão de um novo documento na biblioteca digital.
Segurança	Controle de acesso realizado em dois níveis: (1) controle de usuários baseado no perfil de acesso; e (2) controle de conteúdo baseado por <i>tags</i> . A todo usuário cadastrado é atribuído um perfil de acesso (Tabela 2) que determina as funcionalidades disponíveis. O acesso e visualização de documentos é regido por <i>tags</i> de classificação específicas que associam tipos de conteúdos distintos



	a cada perfil de usuário.
Publicação de documentos	Permite disponibilizar documentos para consulta de usuários externos à instituição, mediante cadastro no sistema. Usuários externos são capazes de acessar todos os documentos públicos e seus metadados, de qualquer <i>tag</i> de classificação, sendo restringido o <i>download</i> dos arquivos eletrônicos.
Métodos de pesquisa	São os mecanismo de pesquisa de documentos, filtros de <i>tags</i> de conteúdo e metadados para os usuários que possuem permissão de consulta. Os resultados obtidos com estes métodos possibilitam selecionar documentos para visualizar de conteúdo e metadados.

Na etapa de descoberta de requisitos, foram discutidas as funcionalidades necessárias para uma biblioteca digital, avaliadas sua importância e comportamento, bem como identificados os conceitos de ECM aplicáveis. Muitos destes requisitos foram determinados com base na avaliação dos sistemas de biblioteca digital Nou-Rau (UNICAMP, 2011), DSpace (DSPACE, 2013), Greenstone (GREENSTONE, 2013) e Invenio (INVENIO, 2013); feita em paralelo com a análise do funcionamento da solução de ECM OnBase (HYLAND, 2011), resultando num conjunto essencial de funcionalidades desejadas para o protótipo em desenvolvimento (Tabela 1).

Os requisitos determinados foram classificados, de acordo com o seu grau de necessidade e relevância, como essenciais, importantes ou desejáveis. Nesta etapa foram descartados os requisitos de ECM não aplicáveis ou indesejados em bibliotecas digitais. A classificação dos requisitos permitiu priorizar a sequência de implementação para obtenção de um protótipo com funcionalidades mínimas.

Tabela 2 – Perfis de acesso do protótipo de biblioteca digital SOA-ECM.

Perfil de acesso	Descrição do perfil (permissão)
Submissão de documentos	Permissão para envio de novos documentos para o sistema.
Aprovação de documentos	Permissão para aprovação ou reprovação de documentos submetidos para o sistema.

Consulta interna	Permissão para consulta de informações completas de documentos armazenados no sistema (conteúdo, metadados e <i>tags</i>), de acordo com seu segundo nível de acesso. Utilizado para usuários pertencentes à instituição.
Consulta pública	Permissão para consulta de informações limitadas dos documentos armazenados no sistema (metadados e <i>tags</i>) e que estão publicados para visualização externa.
Administração	Permissão para execução todas as atividades supracitadas, com acréscimo de permissões administrativas para configuração da aplicação (locais do repositório de documentos, cadastro de usuários, cadastro de <i>tags</i> , etc.).

A partir da seleção dos requisitos desejados para o sistema, foi escrita uma especificação em linguagem natural para descrever seu funcionamento e elaborado um diagrama de casos de uso (Figura 5) para representar as funcionalidades disponíveis e os usuários envolvidos em cada processo.

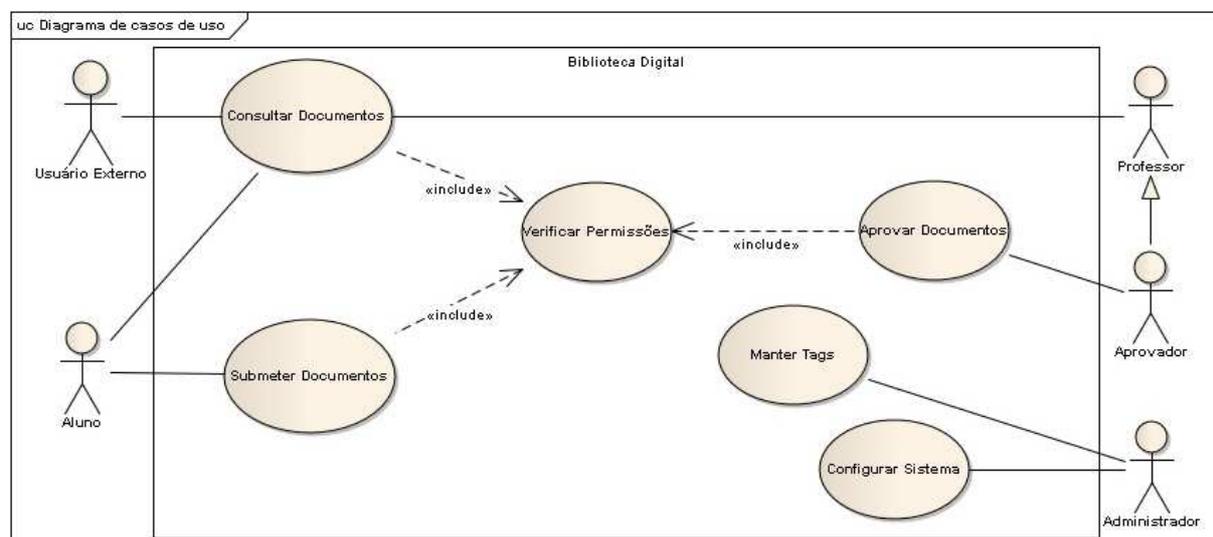


Figura 5 – Diagrama de casos de uso do protótipo de biblioteca digital SOA-ECM.

6. Modelagem de Processos

Conforme apresentado na seção 5 e ilustrado na Figura 5, o protótipo fornece ferramentas para gerenciar os processos de negócio descritos a seguir.



6.1 Submissão de Documentos

É executada toda vez que um usuário, associado ao perfil de *submissão de documentos*, envia um novo documento para armazenamento na biblioteca digital. Este processo requer que o usuário informe um título para o documento; selecione o arquivo desejado para *upload*; indique e preencha o modelo de metadados desejado; e, por último, escolha as *tags* de classificação usadas para indexação e posterior recuperação do documento que está sendo criado.

Todos os novos documentos são criado com *status pendente* (de aprovação), sendo visíveis apenas para os usuários que realizaram sua submissão e para aqueles com perfil de *aprovação de documentos* nas *tags* de classificação do documento. Estes últimos são denominados usuários aprovadores.

6.2 Aprovação de Documentos

Neste processo, os documentos submetidos são analisados pelos usuários aprovadores que, de acordo com critérios administrativos, podem aprovar ou reprovar um documento. A *aprovação do documento* torna-o disponível para todos os usuários com permissão de consulta às *tags* associadas, conferindo o status *aprovado*. A partir deste ponto também é possível publicar o documento pela associação da *tag* apropriada. Por sua vez, a *reprovação do documento* torna-o inacessível aos demais usuários, atualizando seu *status* para *reprovado* e criando um registro dessa operação para visualização do usuário que efetuou a submissão.

6.3 Pesquisa de Documentos

Este processo envolve a consulta de todos os documentos aprovados, i.e., com *status aprovado*, por parte dos usuários autorizados e inclui os métodos de pesquisa por *tag* de classificação ou por busca de metadados.

A pesquisa por *tags* (ou filtragem) só é possível para aquelas habilitadas no perfil de acesso do usuário e retorna uma lista de documentos associados a todas as *tags* informadas como critério. Já a busca por metadados consiste na seleção do tipo do documento a ser pesquisado e consequente determinação da lista de metadados associada, o que resulta numa lista de documentos que possuem os valores informados em seus metadados, que também podem filtrados por *tag*.



7. Modelagem Orientada a Serviços (SOA)

A modelagem dos serviços conforme o paradigma de orientação a serviços, ou seja, da *separação de interesses*, resulta de um inventário de serviços.

7.1 Inventário de Serviços

Numa arquitetura orientada a serviços, *“a maioria dos serviços disponíveis são posicionados como recursos reusáveis e agnósticos aos processos de negócios”* (ERL, 2007, p.42). Tais serviços são desenvolvidos com uma lógica genérica, não é específica em relação ao uso de seus resultados por parte da tarefa que os solicitou, ou seja, esses serviços *“não pertencem a nenhuma aplicação”* em específico.

Ao invés de aplicações, um projeto SOA possui um inventário de serviços agnósticos agrupáveis em composições de serviços responsáveis por coordenar tarefas mais abrangentes. Erl resume o resultado da modelagem orientada a serviços: *“O que antes era uma aplicação agora é uma composição de serviços. E é uma composição criada a partir de serviços que provavelmente participam em outra composição”* (2007, p. 91).

7.2 Análise Orientada a Serviços

A obtenção do inventário de serviços padronizado é feita por meio dos processos de análise e modelagem orientada a serviços. *“Esses processos são centrados em torno da expressão precisa da lógica de negócio através da tecnologia”* (ERL, 2007, p.52). Já a representação de negócios como serviços é conceituada pela IBM (2011) como *“um estilo de arquitetura que suporta a transformação de negócios em serviços interligados, ou tarefas de negócio repetíveis”*.

Posto isso, analistas de negócio e arquitetos de sistema trabalham em conjunto no projeto com o intuito de garantir um grau elevado de alinhamento entre os modelos de negócio documentados e sua real implementação. É desejável que sejam aplicados os princípios de projetos orientados a serviços que, resumidamente, são padronização de contratos, baixo acoplamento, abstração, reusabilidade, autonomia, independência de estado do serviço, capacidades de descoberta e de composição (interoperabilidade). Ou seja:



“...uma diretriz altamente recomendada para moldar a lógica da solução de certa maneira tendo alguns objetivos em mente. Essas metas são geralmente associadas com o estabelecimento de uma ou mais características do projeto (como resultado da aplicação de um princípio)” (ERL, 2007, p.28)

O resultado da aplicação correta destes princípios e do esforço de trabalho realizado pelos projetistas é a garantia de sua *orientação a serviços*, pois são criadas definições conceituais denominadas *candidatos de serviços* que comporão o inventário inicial de serviços do projeto. Uma sequência de interações executadas durante a análise, à luz de tais princípios, refina a especificação e o escopo dos *candidatos de serviço*.

7.3 Modelos de Serviços

A modelagem de serviços pode ser entendida com auxílio de Erl:

“...na construção de vários tipos de serviços, torna-se evidente que eles podem ser categorizados em função: do tipo de lógica que eles encapsulam; do nível do potencial de reuso que essa lógica possui; como essa lógica se relaciona com os domínios existentes na empresa (2007, p.43)”.

O processo de análise orientada a serviços se concentra em projetar e categorizar os serviços de modo a criar três camadas distintas de abstração lógica de serviços (ERL, 2007), ilustradas na Figura 7.

- ***Camada de serviços de entidade***

Disponibiliza as operações básicas para a manutenção de uma entidade de negócio: criação, alteração, exclusão e consulta. Outras operações podem ser desenvolvidas e acrescentadas a tais serviços, desde que intimamente ligadas ao seu funcionamento. Os serviços de entidade são *“considerados serviços altamente reusáveis porque são agnósticos aos processos de negócio que o referenciam”*, de acordo com Erl (2007, p.44).

A principal entidade de negócio do protótipo de biblioteca digital é documento, representada pelo serviço `DocumentService`, o qual possibilita criar um novo documento ou recuperar informações de documentos existentes. Outra entidade importante é *tag*, utilizada na classificação e pesquisa dos documentos. O administrador do sistema utiliza o serviço



`TagService` para criar ou excluir *tags*, enquanto usuários autorizados farão uso desse serviço para acessar a lista de *tags* disponíveis para submissão de documentos.

- ***Camada de serviços de tarefa***

Contém serviços responsáveis pela representação de processos de negócio mais complexos que podem envolver a manipulação de outras entidades de negócio, tal como `SubmitDocumentService`, que possui apenas uma operação para enviar um novo documento ao processo de aprovação da biblioteca digital, mas se preocupa com outros aspectos, além da criação do documento propriamente dito, pois precisa interagir com outros serviços antes de efetivar a criação do documento: verificação de permissões, *upload* do arquivo selecionado e registro do novo documento no sistema. `DocumentService` possui uma operação para criação de documentos e outras para manutenção de registros, mas se concentra exclusivamente no registro dos documento no banco de dados.

Diante das características apresentadas pode-se concluir que “*esse tipo de serviço tende a ter menor potencial de reuso*” (ERL, 2007, p.45), uma vez que eles reusam lógicas agnósticas de outros recursos, para gerar uma composição de serviços mais intimamente ligada com um processo de negócio específico.

- ***Camada de serviços de utilidade***

Diferentemente das camadas apresentadas anteriormente, esta não é orientada ao negócio, pois é “*dedicada a fornecer funcionalidades utilitárias reusáveis, tais como registro de eventos, notificação e tratamento de exceção*” (ERL, 2007, p.46). O serviço de aprovação de documentos `ApprovalService` acessa `LogService` um serviço de utilidade do protótipo que registra os eventos de aprovação de documentos, i.e., data e hora da aprovação, o usuário aprovador e o comentário informado.

7.4 SOA e Web Services

Erl (2007) afirma que SOA é um modelo independente de plataforma de desenvolvimento, ou seja, é um conjunto de diretrizes e práticas utilizadas para que uma organização atinja seus objetivos estratégicos, permitindo o contínuo crescimento tecnológico e de negócios com o menor impacto possível.

A tecnologia mais comumente associada com a implementação de SOA são os *web services*, definidos pelo *World Wide Web Consortium* (W3C) como:



Um web service é um sistema de software projetado para suportar uma interação interoperável máquina a máquina sobre uma rede. Possui uma interface descrita em um formato processável por máquina (especificamente o WSDL). Outros sistemas interagem com o web service de uma maneira prescrita em sua descrição usando mensagens SOAP, tipicamente transmitidas usando HTTP com uma serialização XML em conjunto com outros padrões relacionados à web. (W3C, 2004).

É típico que os *web services* sejam descritos por meio da *Web Service Description Language* (WSDL) e que utilizem o *Simple Object Access Protocol* (SOAP). Segundo Burke & Monson-Haefel (2006), tanto SOAP como WSDL são gramáticas XML, mantidas pelo W3C. A primeira define um protocolo de aplicação flexível e extensível, usado tanto em comunicação assíncrona como em *Remote Procedure Call* (RPC), situação onde se encaixam os *web services*. A segunda especifica uma linguagem para descrição das interfaces envolvidas nos *web services*, o que inclui o tipo de formato esperado para a mensagem, o protocolo Internet usado (como SOAP) e o endereço do serviço.

8. Implementação do Protótipo

8.1 Ambiente de Desenvolvimento

A criação dos componentes de software e dos *web services* do protótipo foi feita com Java (ORACLE, 2011B), é uma linguagem de programação de alto nível, orientada a objetos e cujas muitas características e extensa API possibilitam o desenvolvimento de aplicações *desktop* ou distribuídas (JANDL, 2007). Sua plataforma também inclui bibliotecas adequadas ao desenvolvimento de aplicações *web* e de *web services*, além de permitir o uso de *frameworks* de terceiros para suportar diferentes modelos de desenvolvimento.

O *Java API for XML-Based Web Services* (JAX-WS) é a implementação de referência para “*construção de web services e clientes que se comunicam utilizando XML*” (ORACLE, 2011A). O *framework* Apache CXF foi utilizado no desenvolvimento de *web services* para simplificar a geração dos contratos de serviço em WSDL, o processamento de mensagens SOAP e a padronização dos modelos XML transmitidos pelos *web services* (ASF, 2011A; ASF, 2011B). A IDE utilizada foi o Eclipse Platform (ECLIPSE, 2011; D’ANJOU, 2004), na qual foi integrado o *framework* Apache CXF.

8.2 Modelagem do Banco de Dados

O banco de dados foi modelado para suportar o armazenamento dos dados referentes aos documentos, arquivamento, *tags*, metadados, usuários e permissões de acesso. As tabelas necessárias estão descritas na Tabela 3, seu relacionamento pode ser visto na Figura 6.

Tabela 3 – Tabelas do banco de dados e suas descrições.

Nome da tabela	Descrição da tabela
Document	Principal tabela do protótipo que armazena os dados de cada documento.
DocumentStatus	Contém o cadastro dos possíveis <i>status</i> do documento, usados no controle do fluxo de trabalho da biblioteca.
User	Armazena o cadastro de usuários.
ApprovalLog	Contém os registros de aprovação/reprovação de documentos.
SearchTag	Armazena o cadastro das <i>tags</i> de classificação e pesquisa.
DocumentSearchTag	Relaciona os documentos e suas <i>tags</i> de classificação.
UserSearchTag	Relaciona usuários e as <i>tags</i> de pesquisa de seu perfil de acesso.
PermissionTag	Armazena o cadastro de <i>tags</i> de permissão de acesso às funcionalidades da biblioteca digital.
UserPermissionTag	Relaciona usuários e as <i>tags</i> de permissão de seu perfil de acesso.
File	Armazena os dados referentes ao arquivo eletrônico de cada documento.
Package	Estrutura utilizada para agrupar os arquivos facilitando sua gerência.
Repository	Armazena o registro único que aponta o caminho (<i>path</i>) onde os arquivos deverão ser salvos/consultados.
DocumentType	Armazena o cadastro de modelos de metadados que se relacionam com cada documento.
FieldType	Armazena o cadastro individual de metadados.
FieldDataType	Armazena o cadastro de tipos de dados dos metadados.
DocumentTypeField	Relaciona os modelos de metadados e os próprios metadados.
FieldConfig	Contém os registros de padronização de nomenclaturas dos metadados.
FieldTableX	Para cada metadado cadastrado na tabela <code>FieldType</code> é criada uma tabela <code>FieldTable<id_metadado></code> que armazena em separado os dados informados no momento da submissão de documentos. As colunas são nomeadas de acordo com a tabela <code>FieldConfig</code> .

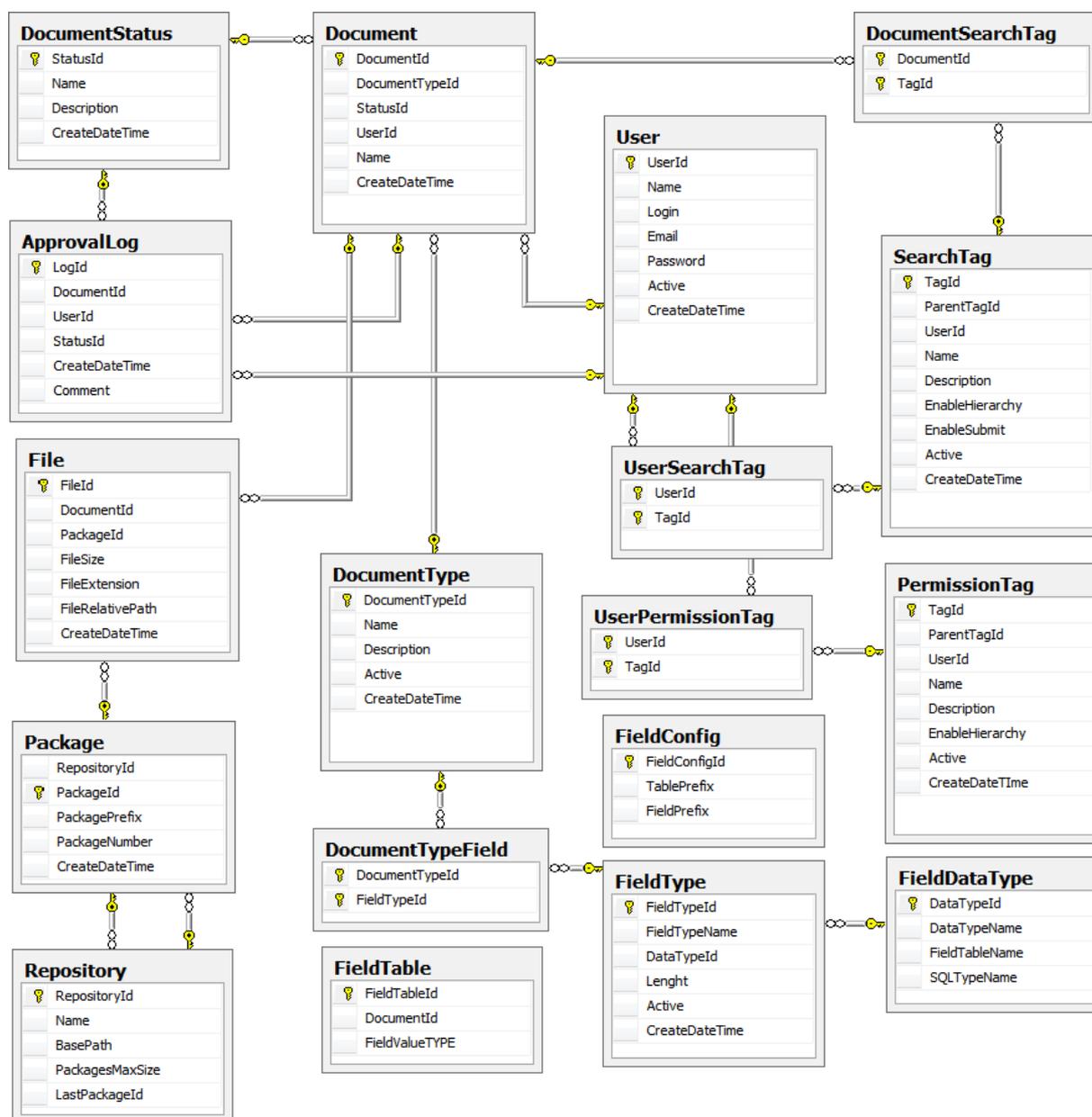


Figura 6 – Diagrama de banco de dados do protótipo de biblioteca digital SOA-ECM.

8.3 Organização dos Web Services

Os *web services* foram desenvolvidos e organizados conforme as diretrizes descritas na seção 7. As camadas de serviços criadas para contemplar a modelagem orientada a serviços, assim como a relação de cada serviço disponível no protótipo de biblioteca digital podem ser visualizadas na Tabela 4 e também na Figura 7.

Tabela 4 – Organização dos *web services* em camadas de serviços.

Camada	Web services.	Descrição
Serviços de Entidade	DocumentService	Gerenciamento de documentos.
	DocumentTypeService	Gerenciamento de modelos de metadados.
	FieldTypeService	Gerenciamento de metadados.
	RepositoryService	Gerenciamento do repositório de arquivos eletrônicos correspondentes aos documentos.
	TagService	Gerenciamento de <i>tags</i> de classificação e pesquisa.
	UserService	Controle de usuários e permissões de acesso.
Serviços de Tarefa	SubmitDocumentService	Composição de serviços para submissão de documentos para a biblioteca digital.
	ApprovalService	Composição de serviços para aprovação de documentos submetidos para a biblioteca digital.
	SearchService	Composição de serviços para pesquisa realizada por usuários internos e externos.
Serviços de Utilidade	LogService	Serviço responsável por registrar as atividades realizadas na operação da biblioteca digital.

Após a organização dos serviços, primeiramente foram desenvolvidos os serviços de entidade, de acordo com o seu grau de relacionamento com outros serviços, pois possuem maior inter-relacionamento com os demais serviços, constituindo sua infraestrutura. Posteriormente, foram criados os serviços de utilidade e finalmente os de tarefa. Após o desenvolvimento de todos os serviços, foi consolidado um inventário de serviços que representa a lógica de negócios do protótipo de biblioteca digital SOA-ECM.

A Figura 7 exibe as três camadas de serviços implementadas (entidade, tarefa e utilidade) e seu relacionamento. Observa-se que todos os serviços da camada de tarefa são composições de outros existentes na camada de entidade, mostrando a aplicação dos princípios do SOA principalmente no tocante a reusabilidade da infraestrutura construída.

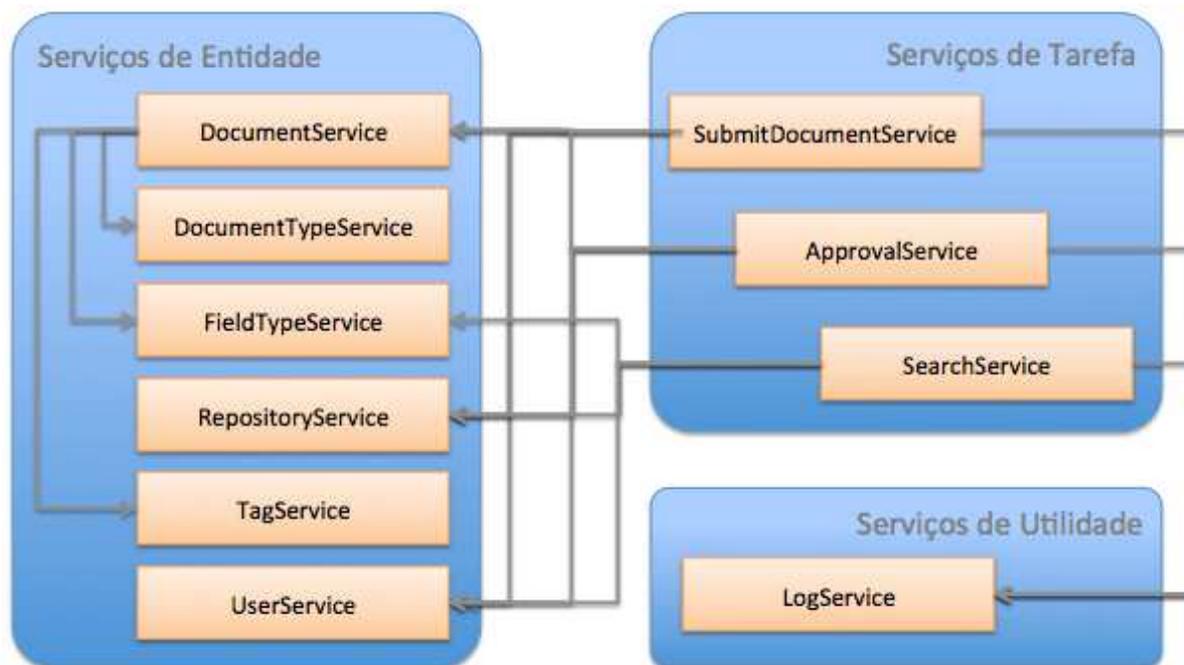


Figura 7 – Diagrama de serviços do protótipo de biblioteca digital SOA-ECM.

8.4 Testes dos Web Services

Para testar os *web services* desenvolvidos foi utilizada a ferramenta SoapUI (SOAPUI, 2011) em sua versão *plugin* para o Eclipse Platform, que fornece uma interface gráfica para automatizar testes funcionais em *web services* utilizando o protocolo SOAP. A partir de um contrato WSDL é possível obter os pacotes SOAP necessários para testar qualquer operação implementada por um serviço e, também, visualizar os pacotes de resposta retornados pelos serviços após a execução do teste.

8.5 Aplicação Cliente

Foi construída uma aplicação *web* para utilizar inventário de serviços implementado que fornece as operações da biblioteca digital ECM-SOA. É importante ressaltar que a modalidade de aplicação escolhida poderia ter sido outra, já que nada impede que *web services* sejam consumidos por aplicativos *desktop* ou mesmo móveis.

O *site* da biblioteca digital foi desenvolvido utilizando a linguagem C# e o *framework* ASP.NET da plataforma Microsoft .NET que, segundo Shepherd (2008), oferecem serviços de execução e uma biblioteca de classes bem projetada para melhorar o desenvolvimento *web*.

A aplicação cliente possui seis páginas *web*: submissão de documentos, aprovação de documentos, pesquisa de documentos, resultados da pesquisa, visualização do documento e administração.

- ***Página de submissão de documentos***

É exibida apenas para os usuários que possuem a *tag* de permissão para submissão de documentos em seu perfil de acesso. Como mostra a Figura 8, nela o usuário informa um título para o documento, o arquivo eletrônico para *upload*, o conjunto de *tags* de classificação (recuperadas pelo serviço *TagService*), o modelo de metadados (obtidos com *DocumentTypeService*) e o valor destes campos (recuperados por *FieldTypeService*). O serviço *SubmitDocumentService* efetua o processamento dessa requisição.

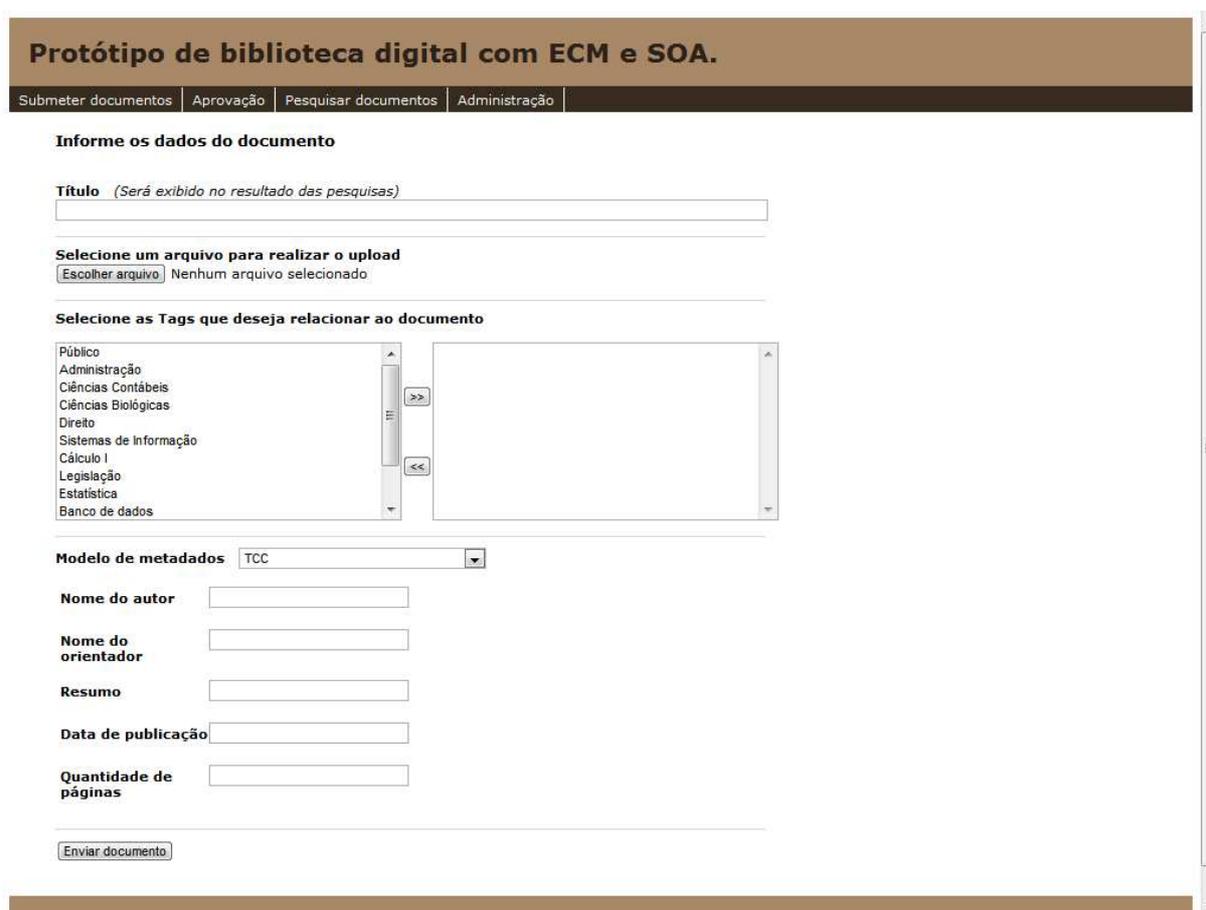


Figura 8 – Página de submissão de documentos do protótipo de biblioteca digital ECM-SOA.

- ***Página de aprovação de documentos***

Esta página é exibida apenas aos usuários autorizados a realizar a aprovação de documentos em seu perfil de acesso. Nela, estão listados os documentos que foram



submetidos para a biblioteca digital, mas ainda não foram aprovados. Um filtro (*ApprovalService*) baseado nas *tags* de pesquisa associadas ao usuário aprovador é aplicado a fim de exibir os documentos de acordo com seu perfil. A lista de documentos permite visualizar o documento, aprovar ou reprovar o mesmo, o que emprega os serviços *ApprovalService* e *LogService*.

- ***Página de pesquisa de documentos***

Exibida apenas aos usuários que possuem permissão para pesquisa, contém duas abas: uma para efetuar a pesquisa por *tags* e outra para efetuar a pesquisa por modelo de metadados. A lista de *tags* para seleção é carregada e obtida por *SearchTagService*, que verifica se o usuário é interno ou externo. A pesquisa é realizada pelo serviço *SearchService* que acessa *DocumentService* para efetivamente buscar os documentos existentes que atendem aos critérios de pesquisa. Vale ressaltar que apenas documentos aprovados são retornados.

Página de resultados da pesquisa

Após a execução de uma pesquisa pelo serviço *SearchService* é retornada uma lista de documentos que pode ser visualizada por esta página, que contém o título dos documentos encontrados, seus modelos de metadados, datas de submissão e *tags* associadas. A partir dessa página é possível acessar a página de visualização de detalhes do documento.

- ***Página de visualização de documentos***

O serviço *DocumentService*, consumido por essa página, é o responsável por retornar os metadados e *tags* do documento selecionado. Usuários internos visualizam os metadados, *tags* e o conteúdo do documento, bem como podem efetuar seu *download* por meio de *link* existente no final da página. Usuários externos somente visualizam os metadados e as *tags* do documento. O *download* do arquivo eletrônico é contemplado pelo serviço *RepositoryService*, o qual transmite os arquivos tanto na submissão quanto na consulta.

- ***Página de administração***

Página exibida apenas aos usuários com perfil administrativo e que concentra funcionalidades de configuração da biblioteca digital que são os cadastros de *tags* de classificação, de campos de metadados e de modelos de metadados, implementados respectivamente pelos serviços *TagService*, *FieldTypeService* e *DocumentTypeService*.



8.6 Análise dos Resultados

A aplicação dos conceitos de ECM beneficiou o protótipo, pois permite a inclusão simplificada de conteúdo e ao oferece um fluxo de aprovação que pode efetivar seu controle de qualidade. O mecanismo de classificação de documentos por *tag* torna a recuperação de informação flexível e possibilita realizar pesquisas abrangentes ou específicas. Outra vantagem da classificação por *tags* é que também operacionaliza os mecanismo de restrição de acesso utilizados na aprovação e permite criar diversos níveis de acesso. A estrutura flexível de modelos de metadados permite o cadastro de quaisquer conjuntos de dados considerados pertinentes a um tipo específico de documento.

A implementação também foi beneficiada pela aplicação dos princípios de SOA, pois a lógica de negócio ficou centralizada no inventário de serviços e pôde ser reutilizada em diversos lugares pela aplicação cliente. A implementação dos serviços como *web services*, além de componentizar a aplicação, tornou a lógica de negócios independente de plataforma, fato evidenciado pelo uso da plataforma Java na codificação dos *web services* e a criação da aplicação cliente com a plataforma .NET.

Em suma, pode-se dizer que o protótipo de biblioteca digital ECM-SOA oferece características e funcionalidades desejáveis a qualquer sistema de biblioteca digital, além de possibilitar o reuso dessas funcionalidades por uma ou mais aplicações cliente, independente da plataforma escolhida.

9. Conclusões

Os resultados obtidos com o protótipo de biblioteca digital ECM-SOA e sua análise permitem concluir que a aplicação dos conceitos de gerenciamento de documentos propostos pela ECM beneficiou o sistema criado, tornando-o consistente no armazenamento de arquivos eletrônicos, flexível na classificação da informação e eficaz na recuperação de documentos.

Observa-se que o esforço investido na etapas de análise de requisitos e modelagem do sistema orientada a serviços trouxe um retorno substancial ao projeto SOA, particularmente na reutilização de serviços, na possibilidades de extensão de funcionalidades, além da independência de plataforma dos consumidores dos serviços criados, o que distingue o protótipo dos sistemas de biblioteca digital comentados.



Entende-se assim que o protótipo de biblioteca digital ECM-SOA atingiu os objetivos propostos de construção de um sistema mais flexível e consistente do ponto de vista de armazenamento, classificação e recuperação de documentos.

Em trabalhos futuros, seria interessante a implementação dos serviços necessários para compatibilização deste protótipo com o OAI-PMH; a extensão dos serviços de submissão para possibilitar o processamento de lotes de documentos; o aperfeiçoamento do modelo de metadados para maior conformidade com padrões internacionais; além de melhorias nas interfaces de usuário.

10. Referências Bibliográficas

AIIM (*Association for Information and Image Management*). *What is Enterprise Content Management?* Disponível em: <<http://www.aiim.org/What-is-ECM-Enterprise-Content-Management.aspx>>. Acesso em 29/set/2011.

ASF (*Apache Software Foundation*). *Apache CXF - Why CXF?* Disponível em <<http://cxf.apache.org/docs/why-cxf.html>>. Acesso em 25/nov/2011A.

_____. *Apache CXF – Download*. Disponível em <<http://cxf.apache.org/download.html>>. acesso em 25/nov/2011B.

ASSOREIRA, P.; MOURÃO, C.; *Bibliotecas Digitais*. Disponível em: <http://www2.ufp.pt/~lmbg/formacao/trabs/biblioteca_digital2.pdf>. Acesso em 03/out/2011.

BURKE, B. & MONSON-HAEFEL, R.; *Enterprise Java Beans 3.0*. 5th Ed. Sebastopol: O'Reilly, 2006.

D'ANJOU, J. et. al.; *The Java(TM) Developer's Guide to Eclipse*. 2nd. Ed. Boston: Addison-Wesley Professional: 2004.

DSPACE; *Dspace.org*. Disponível em <<http://www.dspace.org/>>. Acesso em 14/mar/2013.



ECLIPSE; *Eclipse Download* Disponível em <<http://eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/indigosr1>>. Acesso em 25/nov/2011.

EPRINTS; *EPrints: The original institutional repositon solution.* Disponível em <<http://www.eprints.org/>>. Acesso em 17/mar/2013.

ERL, T.; *SOA: Principles of Service Design.* 1a. Ed. Upper Saddle River: Prentice Hall, 2007.

GREENSTONE. *Greenstone Digital Library Software.* Disponível em <<http://www.greenstone.org/>>. Acesso em 14/mar/2013.

HYLAND SOFTWARE; *Enterprise Content Management Software.* Disponível em: <<http://www.hyland.com/>>. Acesso em 25/mar/2011.

IBM.; *New to SOA and web services.* Disponível em: <<http://www.ibm.com/developerworks/webservices/newto/index.html>>. Acesso em 11/nov/2011.

INVENIO. *About Invenio.* Disponível em <<http://invenio-software.org/>>. Acesso em 14/mar/2013.

JANDL, P. Jr.; *Java – Guia do Programador.* São Paulo: Novatec, 2007.

KAMPFFMEYER, U.; *ECM Enterprise Content Management.* Disponível em: <http://www.project-consult.net/Files/ECM_White%20Paper_kff_2006.pdf>. Acesso em 03/out/2011.

OAI (Open Archives Initiative). *Open Archives Initiative Protocol for Metadata Harvesting.* Disponível em <<http://www.openarchives.org/pmh/>>. Acesso em 16/mar/2013.

ORACLE; *Building web services with JAX-WS* Disponível em <<http://docs.oracle.com/javase/5/tutorial/doc/bnayl.html>>. Acesso em 25/nov/2011A.

_____ *Java SE Downloads.* Disponível em: <<http://www.oracle.com/technetwork/java/javase/downloads/index.html>>. Acesso em 25/nov/2011B.



SOAPUI; *SoapUI – The home of functional testing*. Disponível em <<http://www.soapui.org>>. Acesso em 25/nov/2011.

PATIL, M. S.; KANAMADI, S.; *Digital Library Open Source Software: A Comparative Study*. 2008. Disponível em <http://www.academia.edu/250425/Digital_Library_Open_Source_Software_A_Comparative_Study>. Acesso em 16/mar/2013.

PYROUNAKIS, G., & NIKOLAIDOU, M.; *Comparing Open Source Digital Library Software*. In Y. THENG, S. FOO, D. GOH, & J. Na (Eds.), *Handbook of Research on Digital Libraries: Design, Development, and Impact* (pp. 51-60). Hershey: Information Science Reference, 2009.

SHEPHERD, G.; *Microsoft ASP.NET Step by Step*. Redmond: Microsoft Press, 2008.

SOMMERVILLE, I.; *Software Engineering*. 8th Ed. New York: Pearson Education, 2006.

TEIXEIRA, C. M. S; SILVA, R. L & MARINHO, R. R.; *Tecnologia Open Source na Criação de Bibliotecas Digitais*. Disponível em: <<http://bibliotecascruesp.usp.br/3sibd/docs/teixeira385.pdf>>. Acesso em 10/mar/2011.

TRAMBOO, S.; HUMMA; SHAFI, S. M. & GUL, S.; *A Study on the Open Source Digital Library Software's: Special Reference to DSpace, EPrints and Greenstone* In *International Journal of Computer Applications*. 59 (16)1-9, December/2012.

UNICAMP; *Biblioteca digital da UNICAMP*. Disponível em: <<http://cutter.unicamp.br/>>. Recuperado em 10/mar/2011

.

W3C (WWW Consortium).; *Web Services Architecture*. Disponível em <<http://www.w3.org/TR/ws-arch/>>. Acesso em 25/nov/2011.

WFMC (Workflow Management Coalition).; *Terminology and Glossary*. Disponível em: <<http://www.wfmc.org/Download-document/WFMC-TC-1011-Ver-3-Terminology-and-Glossary-English.html>>. Acesso em 03/out/2011.



WITTEN, I. H.; BAINBRIDGE, D. & NICHOLS, D. M.; *How to Build a Digital Library*. 2nd Ed.
Burlington: Morgan Kaufmann, 2009.