UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ

***CAMPUS* LUIZ MENEGHEL**

ELIZABETE YANASE HIRABARA BENTO

IMPLEMENTAÇÃO DE UM SERVIÇO DE PRESERVAÇÃO DIGITAL UTILIZANDO *WEB SERVICES*

Bandeirantes

2011

ELIZABETE YANASE HIRABARA BENTO

IMPLEMENTAÇÃO DE UM SERVIÇO DE PRESERVAÇÃO DIGITAL UTILIZANDO *WEB SERVICES*

Trabalho de Conclusão de Curso submetido às Faculdades Luiz Meneghel da Universidade Estadual do Norte do Paraná, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientadora: Profª. Ma. Cristiane Yanase Hirabara de Castro

Bandeirantes

2011

ELIZABETE YANASE HIRABARA BENTO

IMPLEMENTAÇÃO DE UM SERVIÇO DE PRESERVAÇÃO DIGITAL UTILIZANDO *WEB SERVICES*

Projeto Final apresentado à Universidade Estadual do Norte do Paraná – C*ampus* Luiz Meneghel – como requisito para aprovação no curso de Sistemas de Informação.

COMISSÃO EXAMINADORA

_____

Profª. Ma. Cristiane Yanase Hirabara de Castro
UENP - Campus Luiz Meneghel

_____

Prof. Me. Bruno Miguel Nogueira de Souza
UENP – Campus Luiz Meneghel

_____

Prof. Me. Ricardo Gonçalves Coelho
UENP – Campus Luiz Meneghel

Bandeirantes, 9 de Dezembro de 2011

Ao meu querido esposo Eberton e minha linda Giulia, com todo o amor de minha vida.

AGRADECIMENTOS

Muitos agradecimentos só poderão acontecer daqui a alguns anos, quando os abraços forem possíveis, mas por enquanto, deixo em escrito toda minha gratidão.

Agradeço primeiramente a Deus, pelas chances e força concedidas.

Agradeço ao meu esposo Eberton e nossa Giulia por serem meu porto seguro, quando todos os dias eram tempestades.

Grata aos meus pais, Orlando e Celina, por acreditarem em mim, irmãs queridas Luciane e Cristiane por me apoiarem nas horas em que mais precisei, sem pedir nada em troca. E aos cunhados e sobrinhos, por todo o apoio.

Agradeço aos meus sogros Nilson e Ivanete, por serem meus pais por todos esses anos.

Grata às tias e tios de Santa de Amélia, pela companhia, preocupação e carinho.

À minha orientadora, Cristiane Y. H. de Castro, pela orientação no projeto e incentivo no desenvolvimento deste trabalho.

Agradeço aos mestres Bruno, Christian, Menolli e Ricardo pelas noites mal dormidas, mas que me fizeram entender a grande razão de querer aprender, sofrer e conseguir compreender uma transformação linear, uma conexão remota, um cubo dimensional e uma estrutura de dados bem definida.

Agradecimento ao ex-professor Roberto Vedoato, com quem aprendi o início de um caminho pelo qual tenho seguido desde então.

Grata aos professores Daniela e Merlin pelas oportunidades conferidas no projeto de extensão Empresarial e Tecnológica.

Agradeço a Nilcea e Alba, pelas lições que pude aplicar em casa, nas horas em que a psicologia e a didática da aluna deram lugar ao de mãe. E ao Glauco, pelas boas conversas no fim das aulas.

Agradeço ao Ederson Sgarbi, com quem descobri as maravilhas das árvores de estrutura e que até hoje me lembram de que uma mulher é capaz de balanceá-las melhor do que muitos homens. Nunca me esquecerei disso.

Agradeço ao professor Carlos Eduardo (Biluka), por meu papel de “Elisabeth” nos teatros das aulas e das boas risadas com toda a turma e à Mariana Monteiro (Mari) minha querida colega no passado e agora admirada professora.

Agradeço aos professores Dellamura, Fábio, Fernando, Lomba, Márcia e Viviane (Vivi) por todo o conhecimento adquirido e a Cidinha, que tanto me auxiliou nessa caminhada.

Agradeço aos amigos e companheiros de faculdade Alex (Cowboy), Artur(Tutu), Felipe(Felipe), Diego(da Sol), Diego(Ruivo), Hellen(Hell), Ricardo(Rico) e Thiago(Potinho) e demais colegas de classe pela longa jornada.

Agradeço aos queridos amigos do LAS (Gui, Jaime, Matsui, Saulo, João, Lélys, Josi, Alessandro, Jóia, Luis Fernando(Montanha), Ortoncelli e Marcel) e da Pacto & Byte´s (todos mesmo) .

Meus últimos agradecimentos, a todos que meu deram forças pra chegar ao fim desse trajeto e aos que não acreditaram que poderia chegar até aqui, mas que de maneira indireta me incentivaram.

"Estou convencido das minhas próprias limitações e esta convicção é minha força"
(Mahatma Gandhi)

RESUMO

Com o aumento das informações em formatos digitais, surgem problemas e desafios, pois essa informação necessita de cuidados que garantam sua preservação das inúmeras intervenções internas e externas como: perda, adulteração e destruição, degradação física, fatores que poderiam modificar o seu conteúdo, comprometendo sua qualidade e integridade. Tanto como questões ligadas à autenticidade e a acessibilidade dos documentos digitais podendo causar sua obsolescência tecnológica. Tratando-se disso, as várias propostas de atividades relacionadas à preservação de documentos têm como base os Formatos de Arquivo e sua preservação. Assim sendo, para compreender a implementação dos processos da Preservação Digital e mitigar os problemas citados anteriormente, necessita-se de um modelo arquitetural flexível, com características como interoperabilidade, independência de linguagens de programação e plataforma de desenvolvimento. Nesse contexto, utiliza-se em muitas soluções a aplicação de *Web Services*. Este trabalho se insere no contexto da PD, especificamente no processo de Ingestão dos dados, e tem como objetivo, propor a implementação de um serviço desse processo, utilizando *Web Services,* a fim de prover a identificação e atualização de formatos digitais, para auxiliar no desenvolvimento de Sistemas de Preservação Digital, a fim de garantir a obsolescência de formatos dos arquivos digitais.

Palavras-chave: Sistemas de Preservação Digital, Java, *Web Services*.

ABSTRACT

With the increase of information in digital formats, there are problems and challenges, as this information requires care to ensure its preservation of many internal and external interventions such as: loss, tampering and destruction, physical deterioration, factors that could modify its contents, committing quality and integrity .As far as issues of authenticity and accessibility of digital documents may cause their technological obsolescence. With regard to this, the various proposed activities related to the preservation of documents are based on the File Formats and preservation. Therefore, to understand the implementation processes of the Digital Preservation and mitigate the problems mentioned above, one needs a flexible architectural model, with features such as interoperability, independent of programming language and development platform. In this context, it is used in many application solutions for Web Services. This work is in the context of PD, specifically in the process of intake data, and aims to propose the implementation of a service of this process, using Web Services in order to provide the identification and update of digital formats to assist the development of digital Preservation Systems(SPD), to ensure the obsolescence of digital file formats.

Keywords*: Digital Preservation Systems, Java, Web Services.*

LISTA DE SIGLAS

AD	Arquivamento Digital
DROID	*Digital Record Object Identification(Identificação de Registro de Objeto Digital)*
ESB	*Enterprise Service BUS*
HTTP	*Hypertext Transfer Protocol(Protocolo de transferência de hipertexto)*
IANA	*Assigned Numbers Authority*
JISC	*Joint Information Systems Committee*
OAIS	*Open Archival Information System* (Sistema Aberto de Arquivos de Informação)
PD	Preservação Digital
PDO	Preservação Digital de Objetos
PI	Pacotes de Informação
PIA	Pacote de Informação para Arquivo
PIS	Pacote de Informação para Submissão
PREMIS	*PReservation Metadata Implementation Strategies*
PUID	*PRonom Unique IDentifier*
SIPreD	Serviço para a Ingestão na Preservação Digital
SOA	*Service Oriented Architecture (Arquitetura Orientada a Serviço)*
SOAP	*Simple Object Access Protocol (Protocolo Simples de Objeto de Acesso)*
SPD	Sistemas de Preservação Digital
UDDI	*Universal Description Discovery and Integration (Descrição Universal de Descoberta e Integração)*
UFPR	Universidade Federal do Paraná
WS	*Web Services*
WSC	*Web Services Composition (Composição de Serviços Web)*
WSDL	*Web Service Definition Language (Definição de Linguagem de Serviços Web)*
XML	*eXtensible Markup Language (Linguagem de Marcação Estendida)*

LISTA DE QUADROS

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

O advento das inovações computacionais tornou a informação uma ferramenta poderosa, pois além de ser valiosa para a descoberta e divulgação de conhecimento, possui características dinâmicas, facilitadas por tais recursos tecnológicos disponíveis. No entanto, com a evolução desses recursos, surgem problemas e desafios, pois a informação, anteriormente permanente em meios físicos, pode tornar-se digital e como tal deve ser preservada.

De acordo com CONWAY:

> "Preservação (*preservation*) é a aquisição, organização e distribuição de recursos a fim de que venham a impedir posterior deterioração ou renovar a possibilidade de utilização de um seleto grupo de materiais" (CONWAY, 2001, p. 14).

Sendo assim, a informação em formato digital necessita de cuidados que garantam sua preservação das inúmeras intervenções internas e externas como: perda, adulteração e destruição, degradação física, obsolescência tecnológica, entre fatores que poderiam modificar o seu conteúdo, comprometendo sua qualidade e integridade. Questões ligadas à autenticidade e a acessibilidade dos documentos digitais por longos períodos, somadas à volatilidade das mídias utilizadas para o registro dos dados. Há ainda uma grande dependência dos formatos de arquivos digitais que sofrem constantes modificações, tornando obrigatórias as migrações de formatos.

Segundo Bodê (2008), várias propostas de atividades relacionadas à preservação de documentos têm como base a forma e a estrutura na qual estão gravadas nos documentos digitais e constituem, de forma geral, os formatos de arquivo**.**

Dessa forma, para satisfazer as expectativas da Preservação Digital de Objetos (PDO), o Sistema de Preservação Digital (SPD) deve compreender alguns componentes funcionais, os quais são identificados pelo modelo de referência *OAIS* (*Open Archival Information System).* Segundo este modelo conceitual e abstrato, em um sistema de gerenciamento de preservação digital, deve estar presentes os processos de Ingestão (gerenciamento da inserção), Gestão dos Dados (gestão), Acesso (acesso aos dados), Repositório (repositório dos dados e metadados) e Administração (gestão entre as fases citadas anteriormente). Atenuando-se ao fato

de que todos os processos são realizados de forma independente, requerendo uma arquitetura que possibilite os serviços que a compõem, serem visualizadas independentes de plataforma de desenvolvimento, sistemas operacionais ou linguagens de programação.

Nesse contexto, os *Web Services* (WS) ou serviços web, tecnologia que segue o modelo arquitetural *SOA* (*Service Oriented Architecture* ou Arquitetura Orientada a Serviço), o qual busca solucionar desafios como os citados anteriormente, são utilizados por muitas bibliotecas e repositórios digitais com o intuito da Preservação Digital(PD).

1.1 JUSTIFICATIVA E FORMULAÇÃO DO PROBLEMA DE PESQUISA

O custo para implementar um SPD que garantam que a informação seja preservada corretamente, é muito alto, isso se deve ao alto valor do suporte de hardware e software (como sistemas operacionais), a utilização de linguagens de programação que podem se tornar obsoletas e os recursos humanos que além de caros não são encontrados com facilidade no mercado. Assim, a criação de novas abordagens para preservação digital deve contar com sistemas de baixo custo que permitam verificar a integridade dos dados, garantam a sua disponibilidade e acessibilidade através do tempo.

Tendo isso em vista, o uso de *WS* para o desenvolvimento de SPD, gera vantagens quanto à garantia de interoperabilidade, obsolescência tecnológica e utilização de linguagens de programação diferenciadas.

Dentre as iniciativas para preservação digital que utilizam WS algumas se destacam como os Projetos CAIRO, *DROID* e *PREMIS.*

Projeto CAIRO, financiado pelo *JISC (Joint Information Systems Committee* ou comitê conjunto de sistemas de informação), desenvolveu uma ferramenta que cria uma interface para o fluxo de trabalho de ingestão e reúne ferramentas de ingestão, principalmente ferramentas de criação de metadados (THOMAS, 2008).

O Projeto *DROID (Digital Record Object Identification*) desenvolvido pelo Departamento de Preservação Digital do Arquivo Nacional Britânico para determinar

os formatos de arquivos individuais, localizados em sistemas de arquivos local, na Web ou transmitidos diretamente pela interface(BROWN, 2006).

O Projeto *PREMIS (PReservation Metadata Implementation Strategies)*, um grupo de trabalho internacional e de manutenção, finalizado em 2005, cujo objetivo era definir os grupos “essenciais” de metadados de preservação utilizados pela comunidade de preservação digital(HITCHCOCK, et al., 2007).

Soluções baseados em *WS* também estão sendo pesquisadas na Universidade Federal do Paraná, buscando suprir a necessidade de instrumentos que proporcionem a criação de *SPD* com baixo custo e com alto grau de confiabilidade e disponibilidade. Pois, as ferramentas existentes, como o *PREMIS, DROID* e o CAIRO assumem que soluções de preservação digital são baseadas em reutilização de serviços existentes com características de interoperabilidade.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Este trabalho tem como objetivo geral testar a consumação de serviços web inseridos no processo de ingestão de dados encontrados na literatura e propor um serviço de Preservação Digital, utilizando WS consumidos de outros repositórios de serviços, que irá compor uma arquitetura de serviços de preservação digital, proposta pelo grupo de trabalho em PD da UFPR.

1.2.2 Objetivos Específicos

Os objetivos específicos deste projeto de pesquisa que permitiram cumprir com o objetivo geral foram:

- Pesquisar e estudar abordagens sobre arquiteturas de Preservação Digital;
- Pesquisar técnicas e ferramentas constantes na literatura para o processo de ingestão de dados;
- Estudar conceitos do paradigma computação orientados a serviços que envolvem *WS*;

- Pesquisar e testar técnicas e ferramentas que utilizem WS para o processo de ingestão;
- Definir um serviço web, que consuma outros serviços web que poderá a compor uma arquitetura complexa de serviços de PD;
- Testar o serviço proposto para verificar vantagens e desvantagens do uso de WS em um processo de PD.

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em quatro capítulos. O primeiro capítulo apresenta uma contextualização do assunto, o problema analisado, a relevância do trabalho e os objetivos a serem alcançados. O segundo capítulo envolve uma revisão bibliográfica de temas relevantes à pesquisa. No capítulo três é apresentado o desenvolvimento, descrevendo as etapas da implementação do serviço de PD, e por fim, no capítulo quatro a conclusão seguida das referências utilizadas.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo fundamentar teoricamente a utilização da Composição de *Web Services* e sua aplicação para o desenvolvimento de sistemas de Preservação Digital, assim como possíveis ferramentas a serem utilizadas na implementação do sistema proposto, que será avaliado para a validação da solução proposta por este trabalho. No Subcapítulo 2.1é abordada definição de Preservação Digital e o Subcapítulo 2.2 consta com fundamentações sobre *SOA* e sua utilização por *WS*.

2.1 Preservação Digital

A PD consiste na capacidade de garantir que a informação digital, seja ela em qualquer formato, permaneça acessível e com nível de qualidade e autenticidade que possibilite ser interpretada futuramente, mesmo que se recorra a uma plataforma tecnológica diferente da utilizada no momento da sua criação. Além disso, a PD é responsável por assegurar que a comunicação seja possível não só através do espaço como do tempo(FERREIRA, 2006)

Nesse contexto, para que a PD de um objeto digital seja possível, é necessário certificar que os níveis de abstração: nível físico, como hardware, nível lógico, tais como softwares, e nível conceitual, linguagem passível de comunicação humana como português ou inglês, estejam acessíveis e interpretáveis. Pois, caso a abstração de um nível para outro seja invalidada, pode tornar o objeto obsoleto (ARELLANO, 2004). Os níveis descritos anteriormente são ilustrados na Figura 1.

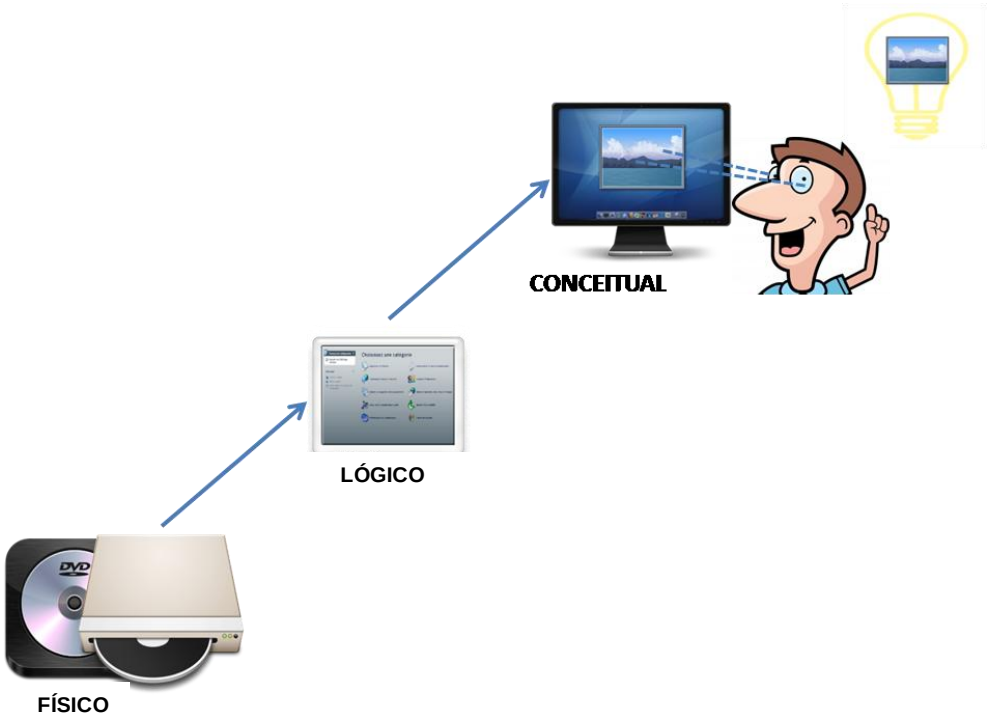

Figura 1 Níveis de abstração da Preservação Digital

Fonte: Adaptado de (FERREIRA, 2006)

Em geral, os SPDs foram criados para atender a necessidades na acessibilidade e segurança de documentos digitais no longo prazo, abordando três processos: a aquisição, a indexação e a distribuição do material a ser preservado. A aquisição de um material com a escolha da forma de armazenamento e a determinação do formato e local de armazenamento focando a preservação desse material no longo prazo. O processo de indexação automática permite a classificação do material obtido e prove o armazenamento de informações relevantes ao processo de recuperação e a distribuição com relação a disponibilidade e acessibilidade(ARELLANO, 2004)

No entanto, a integridade dos dados deve ser garantida, sendo o Arquivamento Digital (AD) o responsável por esta tarefa, sendo os principais objetivos: a obtenção de melhor desempenho no acesso aos documentos em relação ao arquivamento tradicional e o arquivamento no longo prazo utilizando dispositivos com vida útil, em torno de 5 anos . Dessa forma, aplicam-se estratégias de PD, podendo ser utilizadas de acordo o nível de abstração ou dividido em três classes:

- **Emulação:** utilização de emuladores, softwares capazes de reproduzir o comportamento de uma plataforma de hardware, numa outra que à partida seria incompatível(ARELLANO, 2004);
- **Migração:** transferência periódica de objeto digital de uma dada configuração de hardware/software, ou geração de tecnologia para outra subsequente(ARELLANO, 2004); e
- **Encapsulamento:** preservar juntamente com o material digital, a informação essencial e suficiente para permitir o futuro desenvolvimento de conversores, visualizadores e/ou emuladores(FERREIRA, 2006).

2.1.1 O modelo de referência OAIS

Com a crescente importância dos SPDs, surgiu a necessidade de um modelo de referência a fim de padronizar a criação dos sistemas em questão.Para tal, elaborou-se, através da comunidade *Open Archives Initiative,*o modelo OAIS, modelo conceitual que visa identificar os componentes funcionais que deverão fazer parte de um sistema de informação dedicado à preservação digital, como interfaces internas e externas do sistema e os objetos de informação que são manipulados no seu interior(FERREIRA, 2006). Sendo os principais itens avaliados e considerados pelo modelo:

- **Norma ISO/OAIS:** grau de aderência – parcial ou total –do software à norma OAIS e a sua capacidade de implementar os modelos de informação e funcional conforme especificada pelo modelo OAIS (MARCONDES, et al., 2009);
- **Outras metodologias:** nível de apoio ou aplicação do software com outras metodologias cumulativamente ou não com o OAIS (FERREIRA, 2006);
- **Migração:** disponibilidade de ferramentas de apoio à gestão do processo de migração (MARCONDES, et al., 2009);e
- **Outras estratégias de preservação digital:** disponibilidade de aplicação de alguma outra estratégia de preservação (MARCONDES, et al., 2009).

Dessa forma, é importante salientar que o modelo OAIS é composto por duas entidades funcionais, a externa e a interna, e juntas tornam o modelo

operacional. Na Figura 2, estão ilustrados as entidades externas de um modelo OAIS, onde o Produtor corresponde aos responsáveis por fornecer a informação a ser preservada ao longo do tempo, o Administrador é designado à política estrutural de alto nível das atividades do OAIS e o Consumidor inclui os indivíduos que podem aceder e utilizar a informação arquivada no OAIS.

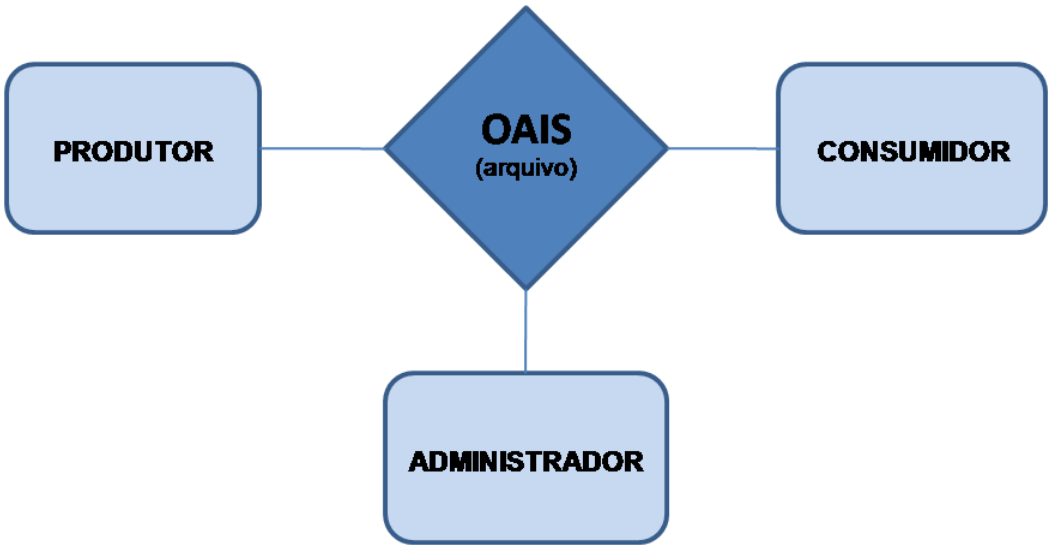

Figura 2 Entidades externas do modelo OAIS
Fonte: Adaptado de (FERREIRA, 2011)

Assim, com as entidades internas de um modelo OAIS é possível observar que a submissão da informação por parte do Produtor, o armazenamento, gestão e preservação dentro do Arquivo e a disseminação da informação por parte do Consumidor são efetuados pelos Pacotes de Informação (PI), os quais, dependendo do momento e do lugar em que se encontram dentro do arquivo, assumem diferentes tipologias: Pacote de Informação para Submissão (PIS), Pacote de Informação para Arquivo (PIA), Pacote de Informação para Disseminação (PID) (FERREIRA, 2011). E juntamente com as seis entidades internas, da entidade funcional externa, tornam o modelo operacional, sendo elas a Ingestão, o Repositório de Dados, a Gestão de Dados, a Administração, o Planejamento da Preservação e o Acesso, podem ser ilustradas na Figura 3.

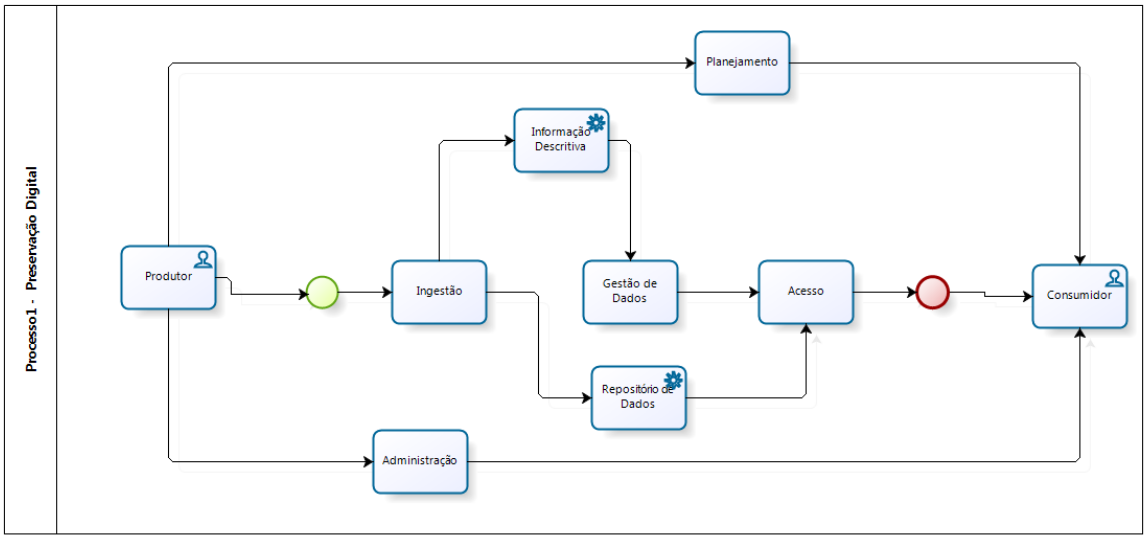

Figura 3 Entidades externas do modelo OAIS

Fonte: Adaptado de (FERREIRA, 2006)

Nesse contexto, é possível salientar que as entidades internas são subprocessos do processo de PD, em que os subprocessos de Ingestão, Informação Descritiva e Gestão de Dados são os responsáveis pelo processo de incorporação, integridade da informação recebida, interface entre o arquivo OAIS e os produtores da informação, produzir o metadado (informações obre os dados), descobrir e localizar o material preservado, armazenar e manter a informação descritiva. Já o Repositório de Dados, fica a cargo do depósito do material a ser preservado, enquanto cabe ao Planejamento definiras políticas de preservação, monitorar o ambiente externo, identificar formatos obsoletos e desencadear os eventos de preservação. Por fim, para o subprocesso de Administração fica o papel de efetuar as operações de manutenção.

2.1.2 Identificação de Formatos de Arquivos Digitais

No processo de Ingestão de Dados, é necessária anteriormente à inserção dos dados, a identificação dos formatos dos arquivos, uma consulta aonde as informações relevantes sobre os dados serão ressaltadas, podendo ainda haver uma identificação de possíveis formatos que em breve serão aposentados. Tal tarefa é necessária para garantir a não obsolescência em relação aos softwares e seus respectivos formatos(BROWN, 2006).

Esse serviço de identificação de formatos pode ser realizado através do *DROID,* sistema automático de identificação de formatos de arquivo, que utiliza os serviços disponibilizados pelo PRONOM (*WS* de registro técnicos sobre formatos e especificações de arquivos e seus formatos), que foi desenvolvido pelo Arquivo Nacional do Reino Unido, sendo a primeira e única forma, até à data, de funcionamento de registro de arquivo público do mundo (The National Archives). Entretanto, os serviços do *DROID,* se encontram em arquitetura desktop sendo necessário o estudo de seu código para efetuar a integração de seu serviços juntamente a um novo serviço ou processo de PD(BROWN, 2006).

Para prover identificadores únicos e inequívocos para o registro em repositórios digitais, foram consultados inúmeros esquemas, dentre eles o MIME *Media Type*, que especifica que tipos de conteúdo, subtipos de conteúdo, conjuntos de caracteres, tipos de acesso, e os valores de conversão, e foi elaborado pelo Internet *Assigned Numbers Authority* (IANA). No entanto, não fornece a granularidade suficiente ou cobertura para satisfazer os requisitos para identificadores únicos. O esquema PRONOM *Unifique Identifier* (PUID), disponibilizado pelo PRONOM, foi desenvolvido com a finalidade única de prestar tais identificadores (The National Archives).

O PUID foi confinado a uma determinada classe de representação de esquema de informação: o formato em que um objeto digital é codificado. Tal esquema é extensível, com mais de 130 dos formatos mais comuns identificadores já atribuído, e mais sendo adicionados em uma base regular, localizada no Reino Unido. Esse PUID é na realidade, um identificador resultante de uma assinatura dos formatos. Essa assinatura do formato é uma coleção de características (podem ser utilizados para indicar o formato do objeto digital em questão) e pode ser externa ou interna ao *bitstream* do objeto real (The National Archives).

Assim sendo, cada formato pode ter vários associados assinaturas internos e externos, com cada assinatura interno composto por uma ou mais sequências de bytes. Formatos também podem ser atribuídos *PUID*s, que pode fornecer uma ligação inequívoca e persistente entre a identificação do formato do arquivo, e a descrição do formato no PRONOM(BROWN, 2006).

Uma versão simplificada do modelo de dados brevemente explicado acima, usado pelo PRONOM para descreveras relações entre os formatos e suas assinaturas, é ilustrado na Figura 4 do diagrama de classe UML a seguir.

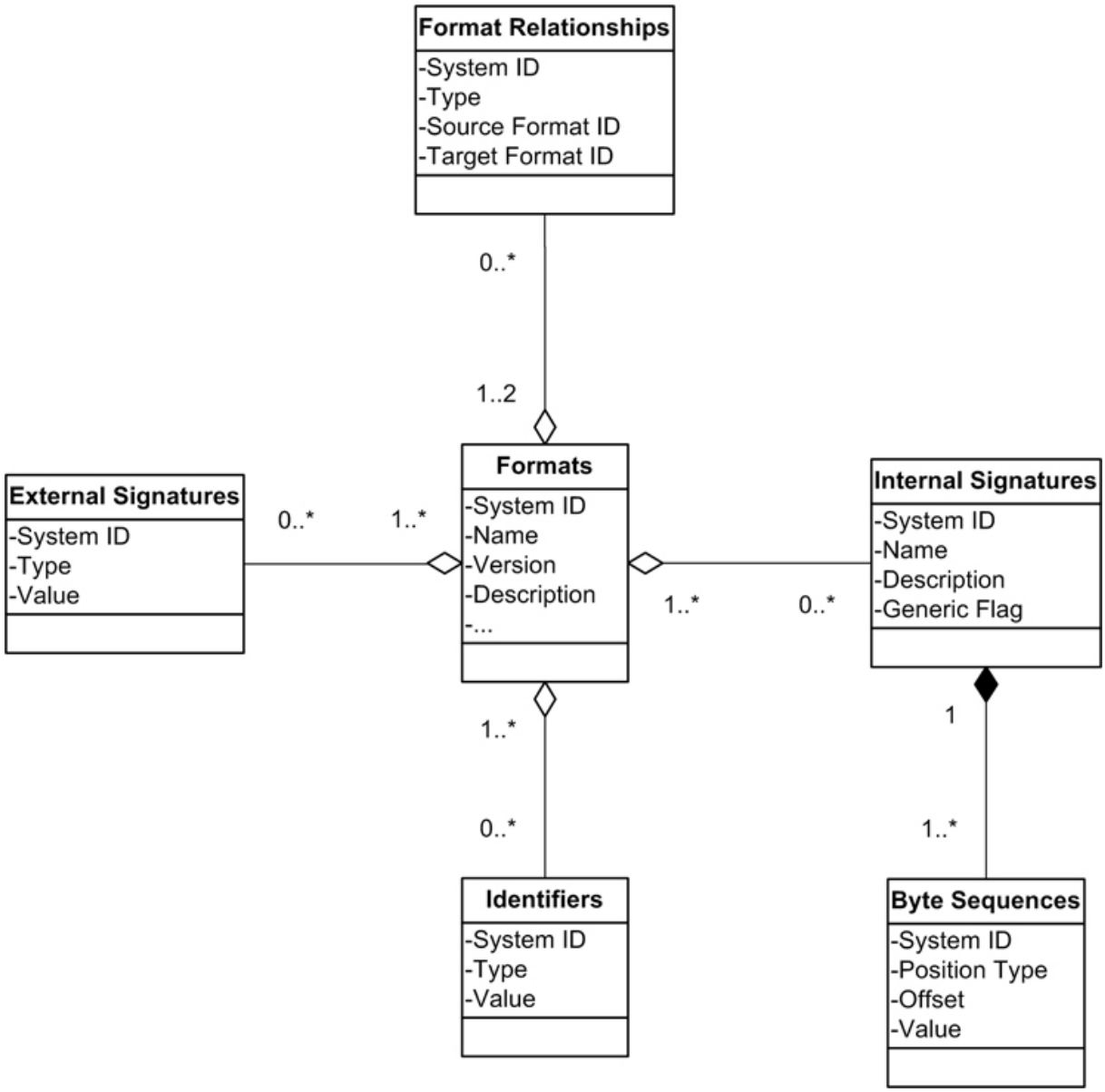

Figura 4 Diagrama de classes do serviço PRONOM
Fonte: (BROWN, 2006)

Nesse contexto, a classe “ExternalSignatures” classificada por este trabalho como Assinatura Externa e a classe “InternalSignatures” por Assinatura interna, são as responsáveis pela identificação do formato do arquivo, sendo estas abordadas a seguir.

2.1.3 Assinatura Externa

Assinaturas externas abrangem todos os indicadores de formato que são externos à sequência de bit do objeto, como dados Macintosh e extensões de arquivos do Windows(The National Archives).

Em alguns sistemas operacionais, uma assinatura externa é fornecida pela extensão do arquivo. Isso indica que o formato do objeto, por exemplo, MyDoc.doc para um documento do tipo DOC, e Mypic.tif para uma imagem TIFF e assim por diante (The National Archives).

No entanto, a principal função da extensão não é validar o formato, mas sim para indicar ao sistema operacional o pacote de software padrão que deverá ser usado para abrir o arquivo. A desvantagem da utilização da assinatura externa se deve aos seguintes critérios(BROWN, 2006):

- Extensões não são necessariamente exclusivos de um único formato;
- Eles não fornecem a granularidade suficiente para identificar adequadamente as versões de formato; e
- As extensões podem ser definidas ou alteradas pelos usuários.

2.1.4 Assinatura Interna

Assinaturas internas abrangem todos os indicadores de formato que estão contidas no *bitstream* de um objeto. Por definição, uma especificação de formato de arquivo impõe uma estrutura específica sobre o conteúdo do *bitstream*, o que é consistente entre todos os objetos digitais de cada formato. Essa característica pode, portanto, ser utilizada como uma assinatura para identificar o formato(BROWN, 2006).

No PRONOM, uma assinatura interna é composta de uma ou mais sequências de bytes, cada um compreendendo uma sequência contínua de valores de bytes hexadecimal e, opcionalmente, as expressões regulares. A sequência de bytes de assinatura é modelada por descrever sua posição inicial dentro de um *bitstream* e seu valor(BROWN, 2006).

A posição inicial pode ser um dos dois tipos básicos:

- Absoluta: a sequência de bytes começa em uma posição fixa dentro do *bitstream*. Esta posição é descrito como um deslocamento de início ou o fim do *bitstream*. A sequência de bytes pode ser localizada movendo para o deslocamento especificado, da esquerda para a direita (BOF) ou da direita para a esquerda (EOF).Levando em consideração o Offset(a partir da posição) que pode ir de 0 a n-1 do tamanho, sendo este o tamanho do *bitstream*; e
- Variável: a sequência de bytes pode começar em qualquer deslocamento dentro da *bitstream*. A sequência de bytes pode ser localizada através do exame dos *bitstream* inteiro.

A assinatura interna pode ainda ser classificada de acordo com sua especificidade, ou seja, se as extensões em questão são específicas para uma única versão ou não de determinado formato. Entretanto, em muitos casos, uma única assinatura pode ser comum a mais de uma versão de um formato Assinaturas que são únicas para um registro em formato PRONOM são chamadas de "específicos" assinaturas, enquanto aqueles que são comuns a vários registros de formato são chamadas de "genéricos" de assinaturas(The National Archives).

2.1.5 Abordagem para a PD

Entre os desenvolvimentos no contexto de Sistemas de Preservação Digital está a inclusão de Arquiteturas Orientadas a Serviço, o que permite esconder as especificidades dos sistemas que disponibilizam o serviço e de cada plataforma que o suporta; e a existência de múltiplos caminhos de migração permite que a solução resista ao desaparecimento gradual de parte dos arquivos digitais (BAPTISTA, et al., 2005)

Sobretudo, o que impulsiona este tipo de abordagem é a principal variável para a redução dos custos de preservação. Pois, o custo para desenvolver um SPD que conservem as características de AD e as demais desejáveis definidas anteriormente é muito alto, referindo-se tanto ao alto valor do suporte de hardware e a adaptação do ambiente físico quanto aos recursos humanos, que deve ser altamente qualificado(BAPTISTA, et al., 2005). Dessa forma, uma abordagem para preservação digital deve contar com sistemas de baixo custo, interoperabilidade caso seja necessário algum tipo de migração, baixo acoplamento e reusabilidade,

características que as arquiteturas de novas abordagens na área estão se aprofundando.

2.2 SERVICES ORIENTED ARCHITECTURE

A definição de *Services Oriented Architecture* (Arquitetura Orientada a Serviços)vai além das capacidades de implementação de um software. Pois, pra considerar que esteja aplicada com sucesso, deve englobar o projeto e desenvolvimento de aplicações e princípios de gestão do negócio (BEAN, 2010). Este trabalho considera a definição desta arquitetura de acordo com sua importância para a engenharia de software apenas. Nesse contexto, o princípio fundamental da SOA é que as aplicações devem ser vistas como serviços, devendo focar essencialmente: baixo acoplamento, interoperabilidade e reuso (KRAFZIG, et al., 2005), como será abordado e exemplificado no decorrer deste capítulo.

Desse modo, é importante salientar que a Arquitetura Orientada a Serviço é baseada em quatro abstrações principais: aplicação *frontend*, serviço, repositório de serviço e serviços de barramento (*Enterprise Service BUS* ou ESB). A aplicação *frontend* é a proprietária do processo do negócio, o serviço consiste de implementação, que fornecer a lógica de negócio e os dados, contrato, especificando funcionalidade, restrições e uso do cliente (podendo ser a aplicação *frontend* ou outro serviço), e interface, que disponibiliza fisicamente as funcionalidades. O repositório de serviço mantém os contratos de cada serviço da SOA individualmente. O serviço de barramento é o que relaciona as aplicações *frontends* e serviços(KRAFZIG, et al., 2005).

Na Figura 5é possível visualizar os artefatos citados acima.

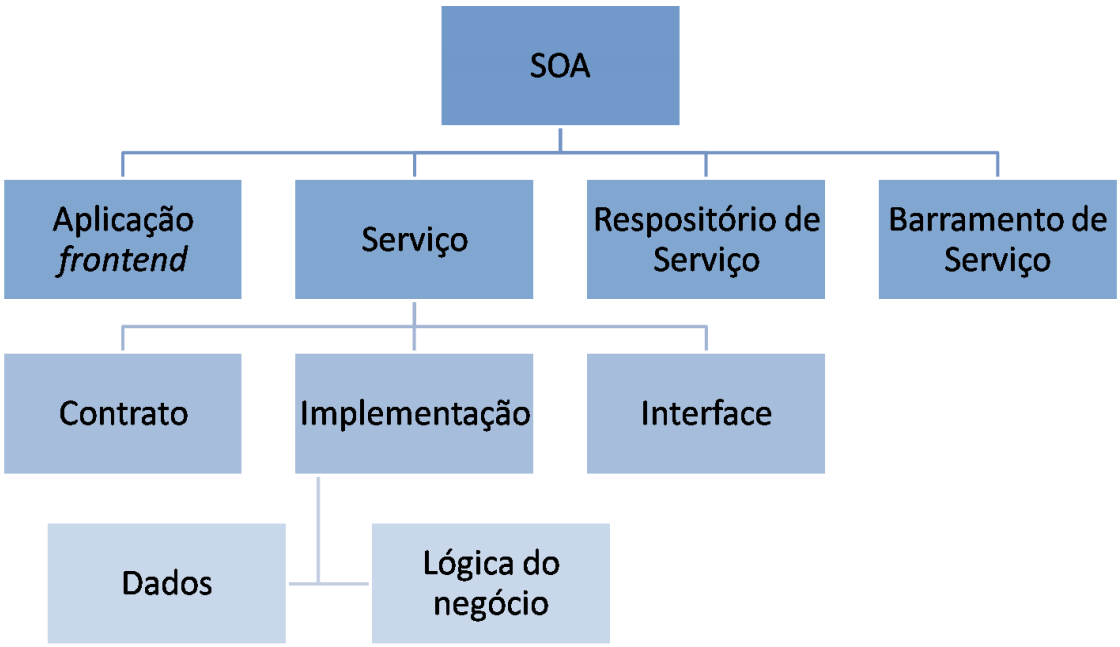

Figura 5 Hierarquia da Arquitetura Orientada a Serviços
Fonte: Adaptado de (BEAN, 2010) (KRAFZIG, et al., 2005)

Com a utilização dos barramentos de serviços (*Enterprise Service BUS*) no relacionamento entre a parte lógica (aplicação *frontend)* e o serviço, paralelamente, ocorre a comunicação através de mensagens para efetuar requisições ou respostas evitando qualquer conexão direta entre as partes em questão e isolá-las completamente, possibilitando que eventuais mudanças nos serviços interfiram no consumo deste, mitigando possíveis riscos de atualizações gerarem algum tipo de erro.Este isolamento caracteriza o baixo acoplamento em SOA.

Na Figura 6 é possível verificar as diferenças em uma comunicação sem a utilização do barramento de serviços e com a aplicação deste(KRAFZIG, et al., 2005).

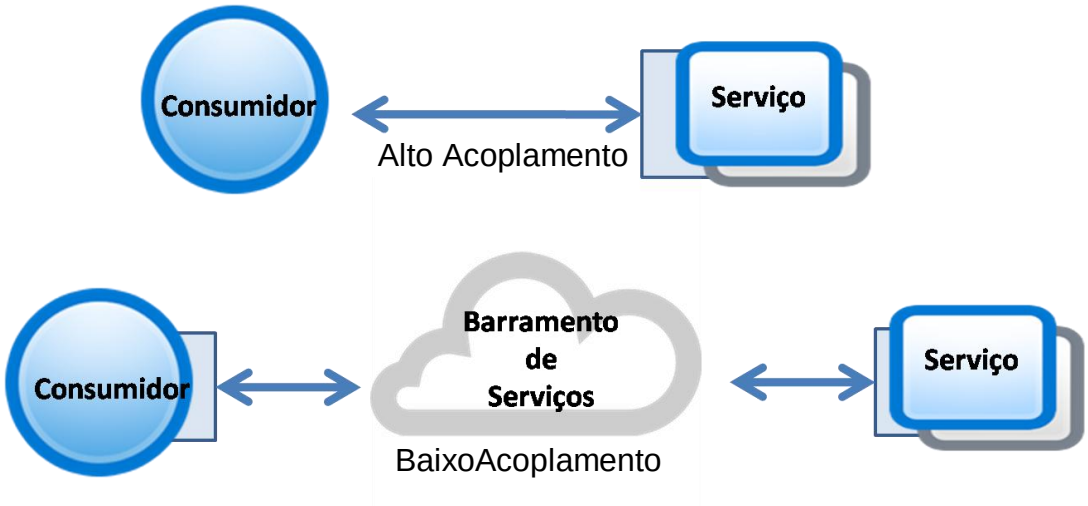

Figura 6Aplicações com alto acoplamento e baixo acoplamento
Fonte: Adaptado de (BEAN, 2010) (KRAFZIG, et al., 2005)

Desta forma, a comunicação ocorre devido ao fato de que mensagens e serviços devem ser autônomos, sem estado *(stateless)*, devendo ser providas de inteligência para controlar sua função na lógica do processo. (KRAFZIG, et al., 2005). Assim, para manter esta independência, os serviços devem encapsular sua lógica dentro de um contexto, podendo este ser um único serviço, uma entidade de negócios ou um agrupamento lógico.

Com isso, é possível verificar que a SOA é semelhante às arquiteturas distribuídas do passado, porque ambas apoiam-se na troca de mensagens e na separação da interface, do processo lógico (KRAFZIG, BANKLE, & SLAMA, 2005). No entanto, o que distingue as duas arquiteturas, é a maneira como os três componentes principais (serviços, descrições e mensagens) são projetados, o que é mostrado na Figura 7. Desta distinção em diante, começa a orientação ao serviço (BEAN, 2010).

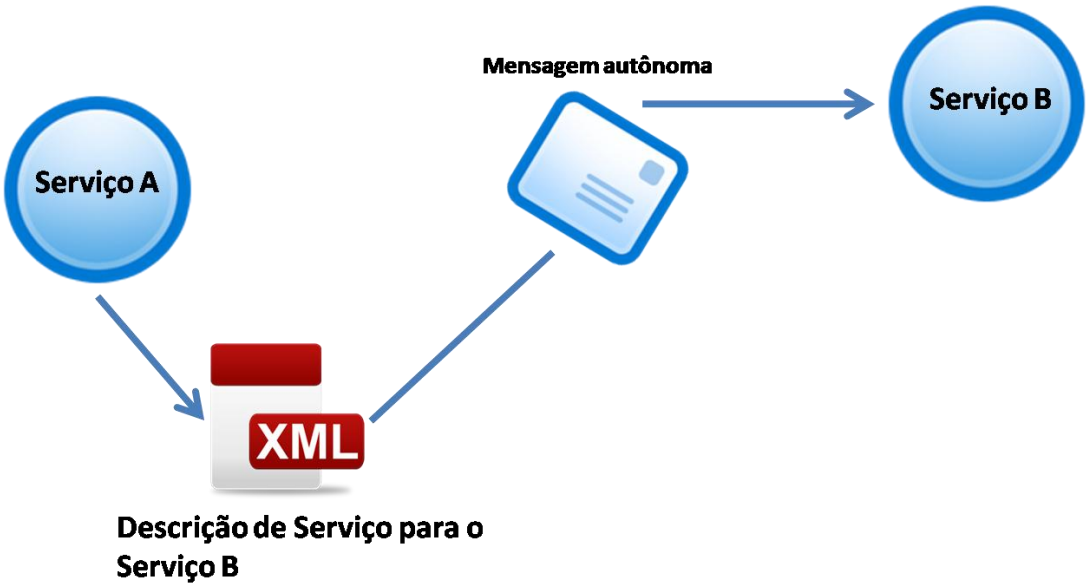

Figura 7 Troca de mensagens entre Serviço A Serviço B
Fonte: Adaptado de (FILAGRANA, 2008)apud (ERL, 2009)

Com o conceito de troca de mensagens citado anteriormente, para que a interoperabilidade ocorra em SOA, a troca de dados entre diferentes sistemas e linguagens de programação deve utilizar um de protocolo de comunicação padronizado, fornecendo base para a integração entre aplicações em diferentes plataformas. Em sua maioria, utilizam-se aqueles baseados em XML (eX*tensible Markup Language*) (BEAN, 2010)Este conceito está ilustrado na Figura 8

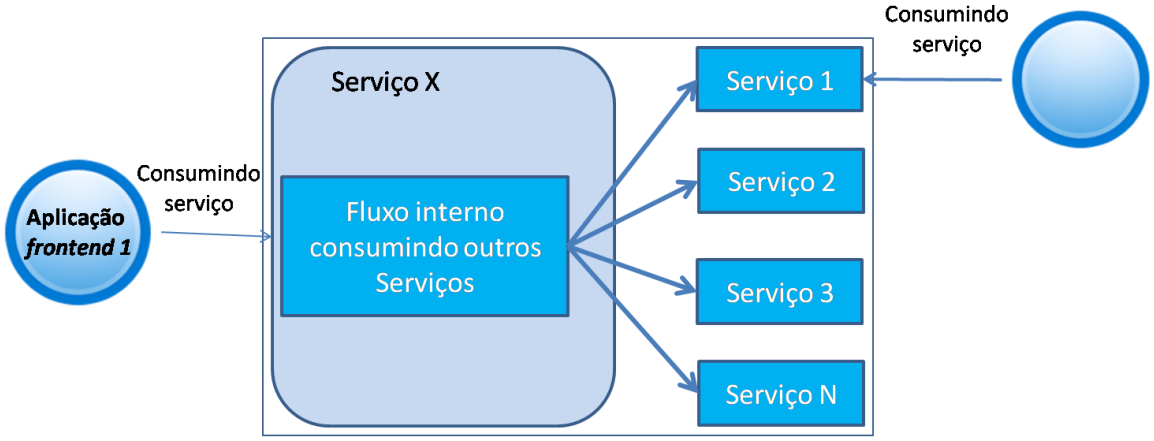

Figura 8 Consumo de Serviços independente de plataforma
Fonte: Adaptado de (FILAGRANA, 2008)apud (ERL, 2009)

Juntamente com o baixo acoplamento e interoperabilidade, os serviços devem ser implementados de forma genérica e encapsulados, não sendo específico para um problema, possibilitando a utilização por outros serviços distintos e

processos ou aplicações e, portanto, incorporando à arquitetura a condição de reusabilidade destas funcionalidades (ENDO, 2008)como ilustrado na Figura 9.

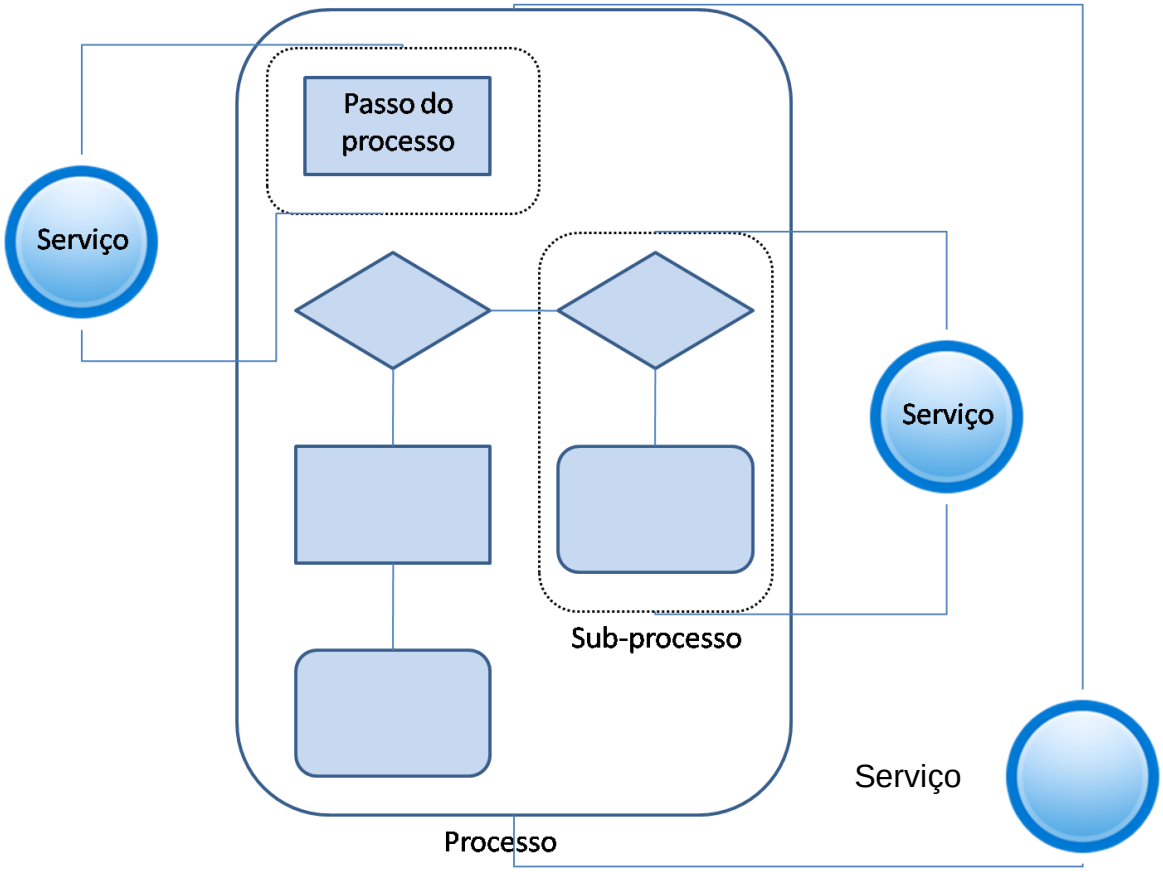

Figura 9 Reusabilidade de serviços
Fonte: Adaptado de (FILAGRANA, 2008) apud (ERL, 2009)

2.2.1 *Web Services* (Serviços Web)

WS podem implementar SOA, por se tornarem operáveis através de protocolos padrões em rede. Estes serviços podem representar tanto novas aplicações quanto apenas um acesso para sistemas legados existente em rede(ENDO, 2008). Em resumo, os WS são sistemas de software projetados para suportar interação máquina-máquina interoperáveis sobre uma rede (BEAN, 2010). Essa interoperabilidade é obtida através de um conjunto de XML baseado em padrões abertos, tais como WSDL, SOAP e UDDI. Estes padrões fornecem uma abordagem comum para a definição, publicação e utilização de *WS* (FARIA, et al., 2010). O funcionamento dos WS ocorre por três principais entidades:

- **Provedor de serviços**: é responsável por realizar a descrição do serviço web, podendo ser compreendido, publicado e permitir a disponibilidade deste serviço(FARIA, et al., 2010).
- **Consumidor de serviços**: é quem utiliza o serviço criado pelo provedor de serviços. Para estabelecer a relação com o serviço e utilizara aplicação é necessário realizar uma pesquisa pelo documento de descrição do serviço, interpretá-lo e enfim inicializar a interação com o serviço web(FEUERLICHT, et al., 2008)
- **Registro do serviço:** é o local centralizado onde são disponibilizadas as informações sobre um serviço web, através do documento de descrição do serviço(FEUERLICHT, et al., 2008)

Na Figura 10 é possível visualizar as entidades funcionais de *Web Services* simulando a comunicação através dos protocolos em rede.

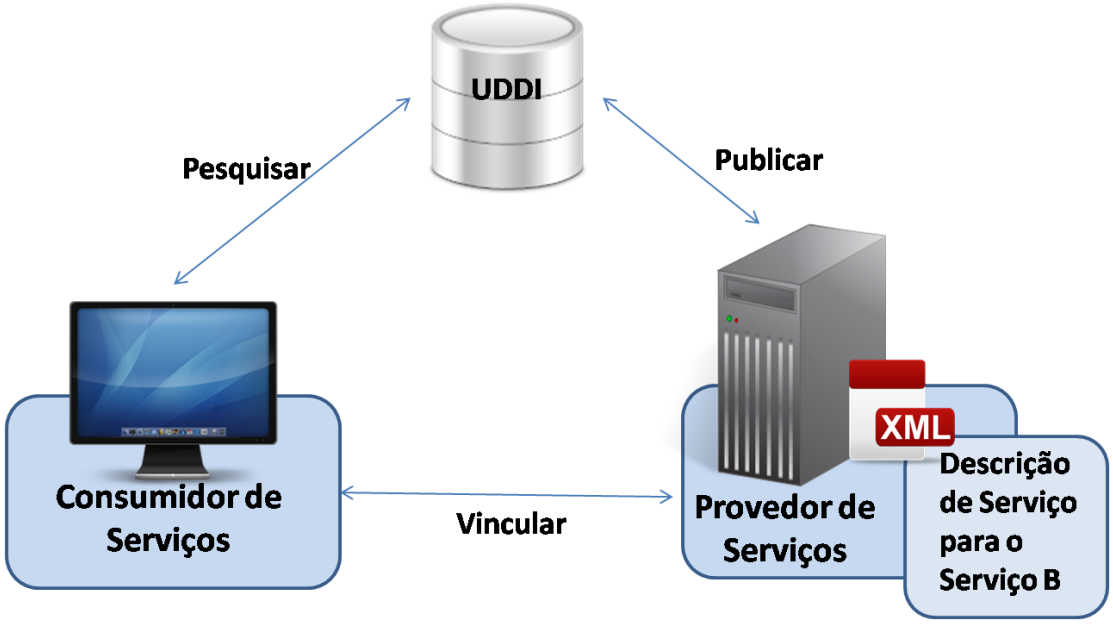

Figura 10 Entidades de funcionamento de Web Services
Fonte: Adaptado de (BEAN, 2010)

Para permitir o funcionamento dos serviços web descrito no modelo básico, há três operações fundamentais: a publicação, a pesquisa e a vinculação. Detalhadamente estas operações devem realizar as seguintes ações(KRAFZIG, et al., 2005):

- **Publicação:** disponibilização do documento de descrição do serviço web por parte do provedor de serviços para permitir a acessibilidade do serviço pelo consumidor de serviços(KRAFZIG, et al., 2005)

- **Pesquisa:** operação de busca pelo documento de descrição do serviço web em um registro dos serviços. Esta procura é realizada pelo consumidor de serviços para possibilitar a interpretação e o entendimento das funcionalidades do serviço(FILAGRANA, 2008)
- **Vinculação:** Após a análise do documento de descrição do serviço realiza-se o vínculo entre o consumidor de serviços e o provedor de serviços. Nesta operação inicia-se a integração entre estas duas entidades permitindo a utilização do serviço web(FILAGRANA, 2008).

Como em SOA, a interoperabilidade nas aplicações de serviços web ocorre através da utilização de padrões que garantam a execução das operações de publicar, pesquisar e vincular. Estas operações devem, portanto, ser realizadas independentemente da plataforma de desenvolvimento e tecnologia de desenvolvimento, para tal, é necessário alguns protocolos básicos citados anteriormente, os quais são comuns entre todos que pertencem ao ambiente de serviços web(ENDO, 2008)

Sendo assim, a pilha básica de protocolos, como na Figura 11, representa a arquitetura necessária para possibilitar a execução das operações em questão, permitindo a integração entre as entidades destacadas no modelo conceitual dos serviços web, lembrando que o fator fundamental é permitir a transparência na execução das operações. Os quais serão brevemente abordados nos subcapítulos a seguir(CERAMIS, 2002).

Figura 11 Pilha de protocolos de *Web Services*
Fonte: Adaptado de(ENDO, 2008)

2.2.1.1 Rede de Transporte

A primeira camada da pilha básica de serviços web é a camada de Rede de Transporte, sendo responsável pela distribuição de diversos protocolos de transporte que poderão ser utilizados, como o HTTP, SMTP, FTP e HTTPS, entre outros. Sendo que a escolha do protocolo envolve fatores como a segurança, desempenho, facilidade de uso e disponibilidade, contanto que o transporte do serviço será transparente para o desenvolvimento do aplicativo. O mais usual é o HTTP, por sua popularidade entre os browsers e ser vastamente utilizado na Internet(BEAN, 2010) (KRAFZIG, et al., 2005).

2.2.1.2 Troca de Mensagens XML

Nesta segunda camada da pilha básica de Serviços Web, define-se o formato para a realização da troca de mensagens na comunicação entre os aplicativos. O protocolo padronizado e utilizado frequentemente nos serviços web é o SOAP, baseado em *Extensible Markup Language* (XML) e *Hypertext Transfer Protocol* (HTTP), para assegurar a interoperabilidade e intercomunicação entre os diferentes sistemas, pois facilitam a expansibilidade através e na independência do protocolo de transporte dos dados, respectivamente (ENDO, 2008).

O protocolo SOAP consiste de três partes: o envelope, que define um *framework* para a descrição do conteúdo da mensagem e como ela deve ser processada, o cabeçalho, opcional e define dados adicionais necessários e o corpo que contém as chamadas e as respostas dos procedimentos(FILAGRANA, 2008).

Na Figura 12a seguir pode-se visualizar a estrutura do protocolo SOAP

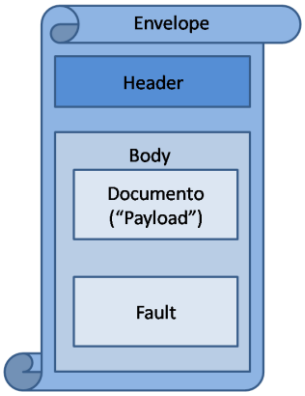

Figura 12 Protocolo SOAP

Fonte:Adaptado de (FREIRE, 2007)

2.2.1.3 Descrição do Serviço

A camada de descrição do serviço fornece, descreve as funcionalidades existentes, possibilitando ao consumidor entender o funcionamento da aplicação a ser chamada, assim como o que será necessário realizar para ser possível esta interação, facilitando o estudo das solicitações e a programação para integrar o provedor de serviços e o consumidor(FREIRE, 2007). Em *Web Services*, esta descrição e realizada através de um documento, o *Web Service Description Language (WSDL)* ou Linguagem para Descrição de Serviços Web, sendo dividido em duas partes:

- **Definição da interface do serviço:** parte reutilizável da definição do serviço que representa apenas uma definição abstrata. Composta por elementos que descrevem a interface oferecida por um serviço e como chamá-la, como tipos de porta, definição das operações de um serviço web, mensagens, contém a definição dos dados a serem transmitidos, tipos, os tipos de dados que estão presentes na mensagem, e vínculos, que captura o protocolo particular e os elementos tipos de porta, mensagem e tipo mencionados acima; e
- **Definição da implementação do serviço:** representa como uma interface de serviço específica está implementada por um provedor de serviços. Composta por serviços, coleção de elementos porta, e porta, informações necessárias para identificar o local para onde deve ser enviada a solicitação real.

2.2.1.4 Publicação e descoberta de serviço

A camada de publicação e descoberta de serviços e a responsável por promover a disponibilidade dos Serviços Web. Pelo documento de descrição do serviço interpretam-se os procedimentos para estabelecer o vinculo com o serviço web, mas não se conhece o acesso a este documento. Assim, quando o provedor de serviços publica informações referentes ao seu serviço web no um registro central, estes se tornarão disponíveis publicamente aos consumidores de serviços interessados(CERAMIS, 2002)

A especificação para a publicação e localização de informações comerciais é a *Universal Discovery, Description, and Integration* (UDDI) ou Integração e

descoberta da Descrição universal, que fornece uma estrutura comercial, usada para descrever um determinado negócio, na qual constam as seguintes informações (FILAGRANA, 2008) como:

- **Negócio:** informações comerciais (empresa ou o fornecedor do serviço), instâncias da estrutura do serviço, nome do negócio e uma descrição em várias línguas, informações para contatos e classificações comerciais;
- **Serviço**: Diferentes serviços web fornecidos por este negócio. Cada serviço contém uma ou mais estruturas de especificação técnica; e
- **Especificação técnica:** detalhes técnicos referentes ao serviço web. Uma dessas especificações técnicas é a especificação WSDL.

3 SERVIÇO PARA A INGESTÃO NA PRESERVAÇÃO DIGITAL – SIPRED

Este trabalho propõe a implementação de um serviço de identificação de formatos de arquivo para o processo de Ingestão de PD utilizando *WS,* sendo disponibilidade como um Serviço Web. Para tal, foram utilizados os Serviços Web disponibilizados pelo PRONOM.

3.1 MATERIAIS E MÉTODOS

3.1.1 Métodos

Este trabalho, por sua natureza, constitui-se em uma pesquisa aplicada, pois evidencia problemas típicos da área de PD. Do ponto de vista de procedimentos, divide-se em duas partes: pesquisa bibliográfica e exploratória, pois se fundamenta teoricamente em materiais anteriormente publicados de fontes variáveis e desenvolvimento de protótipo para avaliar e validar a solução proposta.

3.1.1.1 Métodos de validação

O método de validação deste trabalho é a implementação de clientes para efetuar a consumação do *WS* proposto, tendo como objetivo avaliar o desenvolvimento do protótipo de um serviço e resultados obtidos através deste, e reunir informações relevantes ao aperfeiçoamento do trabalho.

3.1.2 Materiais

3.1.2.1 Linguagem de Programação

Linguagem desenvolvida pela Sun Microsystems, contendo a característica de portabilidade, pois pode ser executada em diferentes plataformas.A versão do Java utilizada foi a contida no pacote de desenvolvimento J2SDK,Enterprise Edition (Versão: 1.6.0.20). Este pacote contém as APIs da linguagem Java e um conjunto de bibliotecas que facilitam no tratamento de documentos XML e SOAP utilizados no processo de desenvolvimento, juntamente com os recursos de interoperabilidade que as demais ferramentas utilizam.

3.1.2.2 Biblioteca para *WS*

API utilizada para desenvolver o *WS* proposto foi o Java API for XML Web Services (JAX-WS) devido sua fácil utilização e aplicação juntamente à linguagem escolhida.

3.1.2.3 Servidor de aplicação

Glassfish é um servidor de aplicação desenvolvido pela Sun Microsystems para a plataforma Java *Enterprise Edition* (Java EE), sendo selecionado por sua fácil integração com a API JAX-WS.

3.1.2.4 Ambiente de desenvolvimento

O *Netbeans* IDE é um ambiente de desenvolvimento de sistemas, completamente escrito em Java, mas pode suportar qualquer linguagem de programação orientada no paradigma procedural. Há também um grande número de módulos para estender o IDE *Netbeans.* Tais características tornam o processo de implementação mais fácil e por tais razões foi utilizado.

3.2 ARQUITETURA

Para o desenvolvimento do serviço proposto por este trabalho, foi considerado a arquitetura em desenvolvimento, elaborada pela Universidade Federal do Paraná(UFPR), que desenvolve atualmente um Sistema de Gerenciamento de Preservação Digital. Tal arquitetura de PD segue os padrões definidos pela OAIS e pode ser visualizada na Figura 10

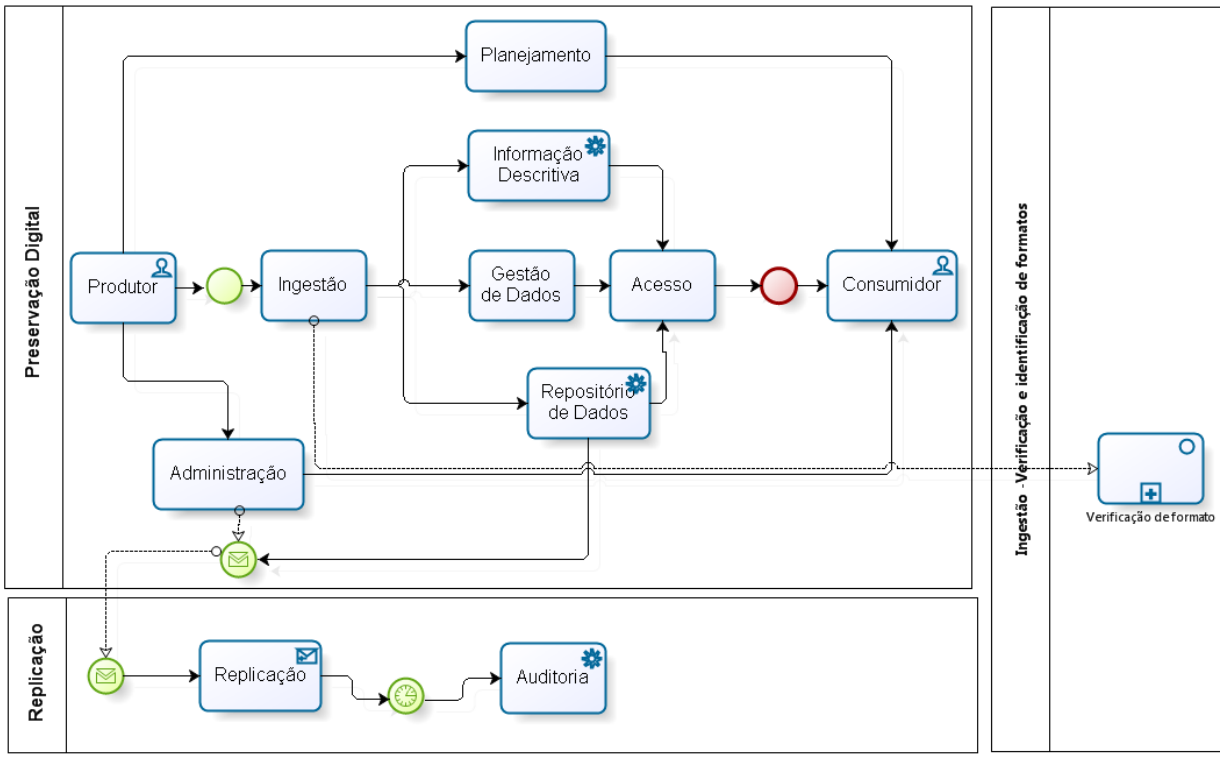

Figura 13 Arquitetura de PD proposta pela UFPR

Fonte: Adaptado de Grupo de trabalho em PD da UFPR

3.3 PROJETO

O serviço proposto por este trabalho se insere no âmbito de identificação do formato, pois enquanto a identificação verifica o formato no qual um objeto se propõe a ser codificada, a validação assegura que o objeto está plenamente em conformidade com a especificação do formato.

Desse modo, garantiu-se que essa identificação fosse realizada visando os requisitos de assinatura interna e externa. Para este trabalho, o processo é elaborado de duas maneiras distintas: processo de consulta por arquivo e processo de consulta por extensão do arquivo. E são disponibilizadas pelo sistema através dos métodos de extensão e operação respectivamente, os quais retornam uma lista de objetos do tipo Registro, que possui os atributos básicos de informação sobre o arquivo ou extensão deste. As classes essenciais do projeto podem ser visualizadas na do diagrama de classe do SIPreD.

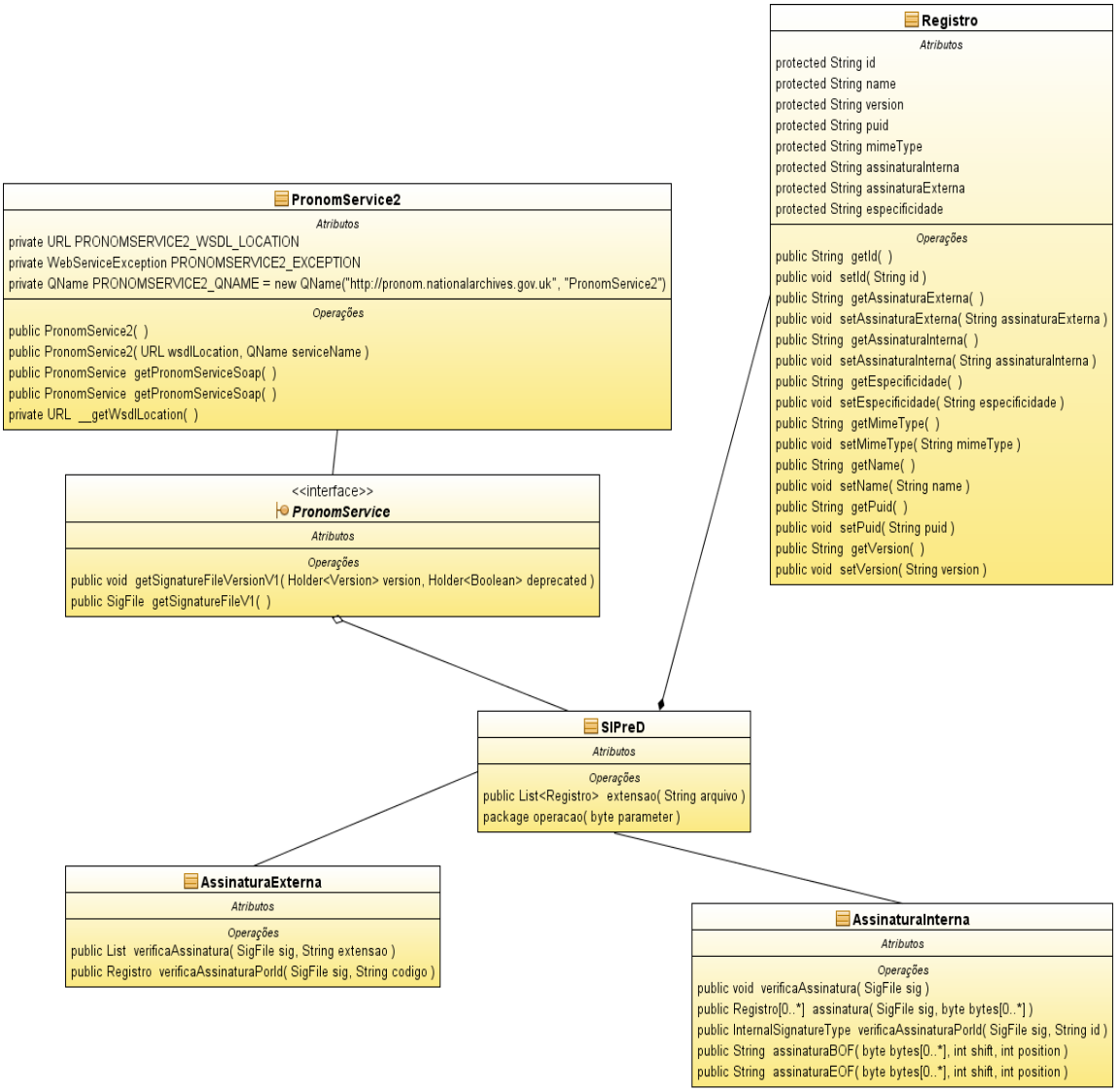

Figura 14 Diagrama de classe SIPreD

Desse modo, no primeiro processo uma procura pelas assinaturas interna e externa de acordo com os dados do arquivo recebido pelo servidor, enquanto o segundo processo identifica tais assinaturas de acordo com a extensão do arquivo enviado pelo cliente que consome o *WS* proposto.

Essas duas maneiras explicadas acima podem ser visualizadas nas Figura 15 Processo de identificação de assinaturas pelo arquivo e Figura 16 respectivamente.

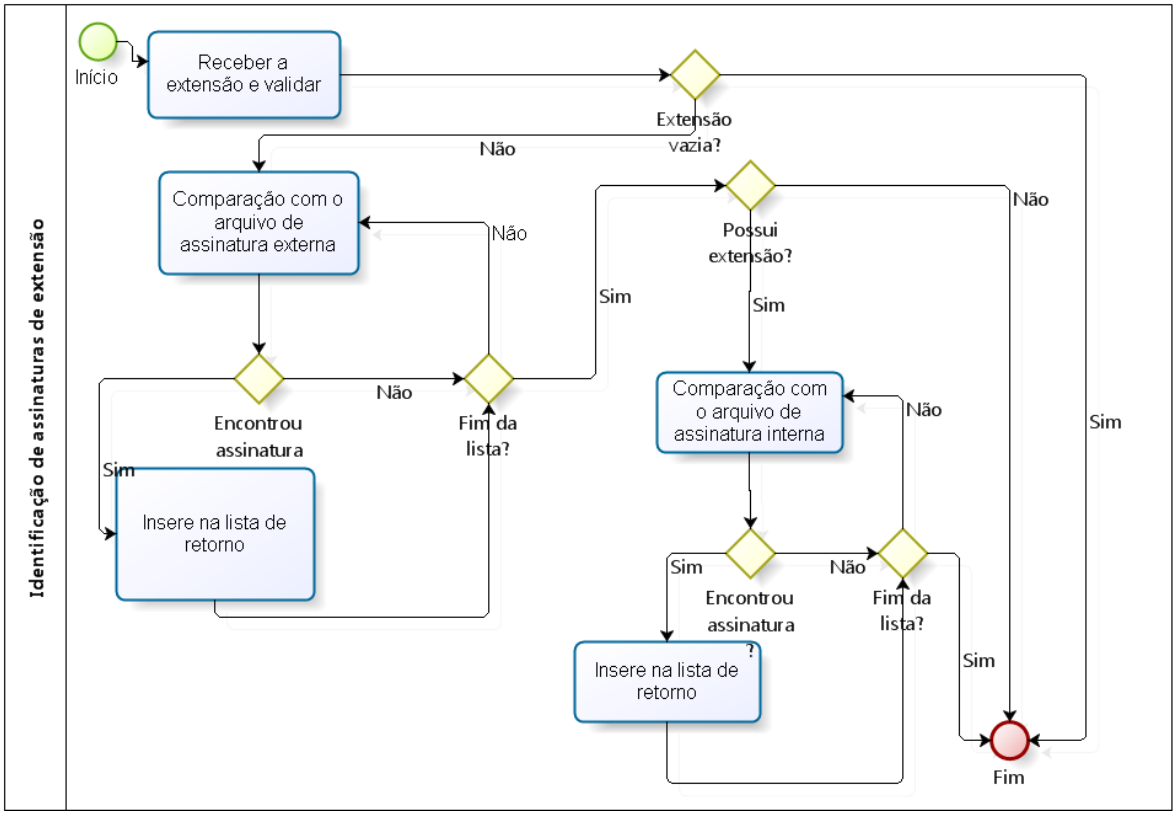

Figura 15 Processo de identificação de assinaturas pelo arquivo

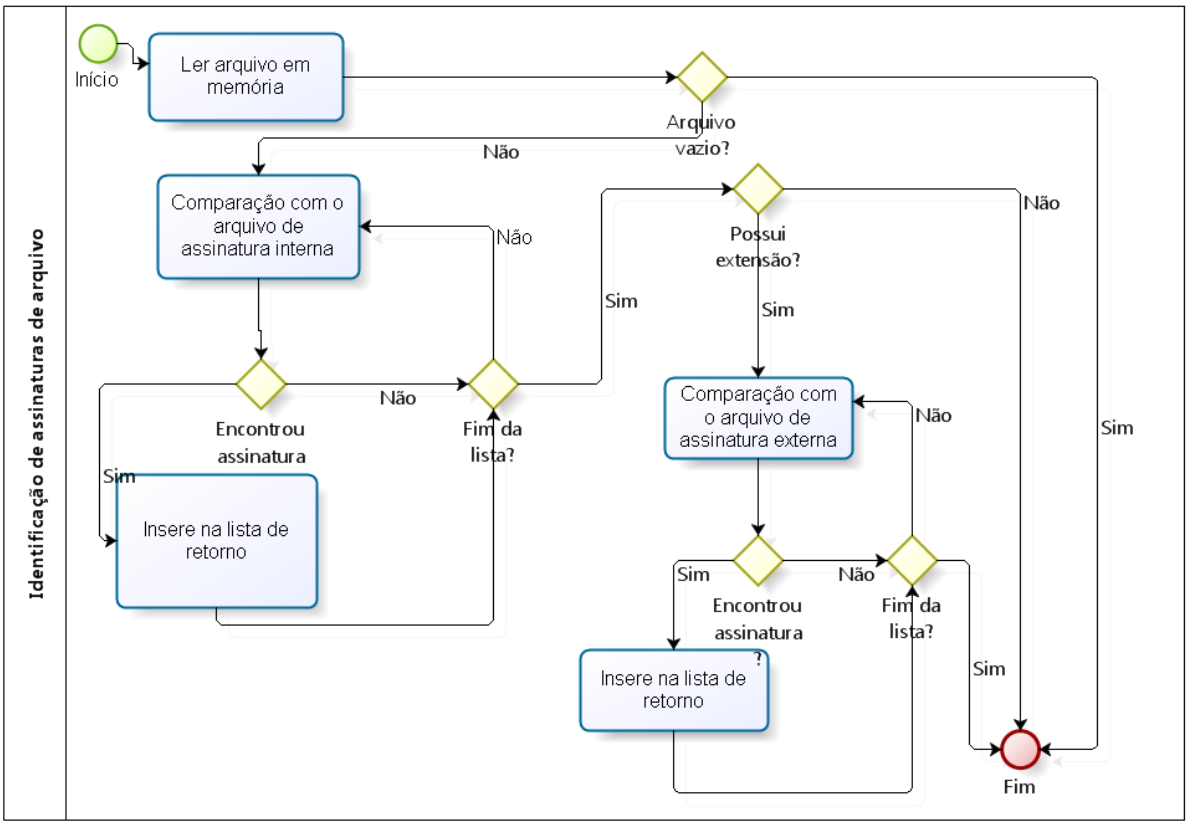

Figura 16 Processo de identificação de assinaturas pela extensão

Assim sendo, os processos de identificação de assinatura Interna e assinatura externa serão abordados mais detalhadamente a seguir.

3.3.1 Processo de Identificação de Assinatura Interna

Este processo ocorre por meio da verificação do tipo do arquivo de acordo com sua especificação, que consta no arquivo binário do mesmo. O foco inicial do serviço é a identificação de arquivos texto ou dos tipos “doc” ou “odf” mais utilizados para a elaboração de artigos e trabalhos acadêmicos. E por esta razão, o serviço limita-se ao tipo BOF de assinatura ou EOF, sendo, portanto necessário que seja identificado a subsequência de acordo com o definido na assinatura. As etapas podem ser definidas a seguir:

Etapa 1- Nesta etapa verifica se a subsequência é do tipo BOF ou EOF;

Etapa 2 – A segunda etapa consulta se a posição é absoluta ou variável, entretanto, para este trabalho, será considerada como etapa 2 a posição inicial, offset, a qual será realizada a pesquisa do hexadecimal;

Etapa 3 – Terceira etapa identifica a quantidade de casas hexadecimais, ou shift, serão consultadas para a montagem da assinatura; e

Etapa 4 – Na última etapa a assinatura está completa e deve ser comparada com a assinatura consultada.

As etapas acima podem ser melhor visualizadas na Figura 17 a seguir.

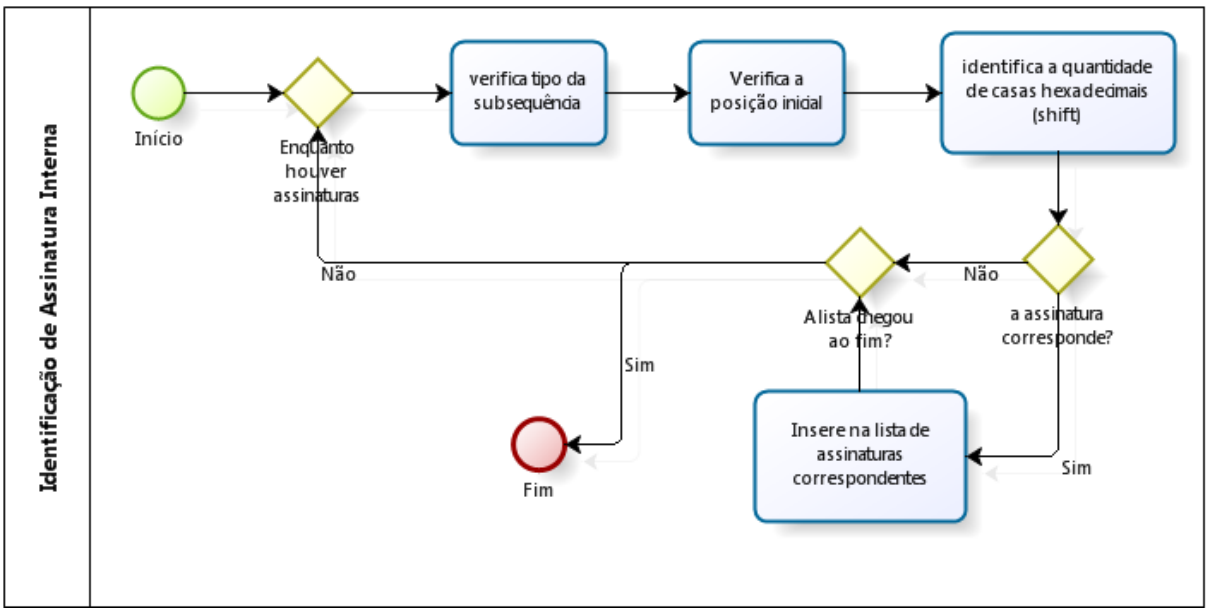

Figura 17 Identificação de Assinatura Interna

3.3.2 Processo de Identificação de Assinatura Externa

Já a identificação externa ocorre por meio de verificação de a extensão constar ou não na consulta de dados efetuada no PRONOM. É considerado um processo simples, mas essencial, pois os dados que retornam para a elaboração do metadado do serviço proposto.

3.3.3 Utilização do Serviço PRONOM

Para realizar o consumo dos serviços oferecidos pelo PRONOM, foi necessário a conexão com seu *WS* através do endereço “http://www.nationalarchives.gov.uk/pronom/Services/Contract/PRONOM.wsdl”, local em que se situa o arquivo *WSDL*, com toda a especificação do serviço em questão. Com isso, efetua-se uma conexão com o protocolo SOAP através do endereço “wsdlhttp://www.nationalarchives.gov.uk/pronom/service.asmx?WSDL”, utilizando a porta denominada “PronomServiceSoap”, todos de acordo com a especificação do *WSDL,* como é possível observar no Quadro 1 estão ilustrados as definições desse protocolo *WSDL* do serviço web disposto pelo PRONOM, o qual estava presente na classe da Figura 14 Diagrama de classe SIPreD.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions name="PronomService1"
                  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
                  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xmlns:tns="http://pronom.nationalarchives.gov.uk"
                                xmlns:sfns="http://www.nationalarchives.gov.uk/pronom/SignatureFile"
                  targetNamespace="http://pronom.nationalarchives.gov.uk"
                  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
                  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

        <wsdl:service name="PronomService2">
                <wsdl:port name="PronomServiceSoap" binding="tns:PronomServiceSoap">
                        <soap:address location=""/>
                </wsdl:port>
        </wsdl:service>
```

Quadro 1 Identificação do serviço PRONOM

Os serviços, ou melhor, operações disponíveis pelo PRONOM pode ser conferidos no Quadro 2 Identificação das operações do PRONOM, sendo os dois métodos nomeados “getSignatureFileVersionV1In” e “getSignatureFileV1In” respectivamente, aonde o primeiro retorna uma lista de assinaturas atualizadas

sem receber parâmetros para a consulta, e o outro recebendo a versão e uma variável booleana se esta está depreciada ou não.

```
<wsdl:binding name="PronomServiceSoap" type="tns:PronomService">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
        <wsdl:operation name="getSignatureFileVersionV1">
                <soap:operation soapAction="http://pronom.nationalarchives.gov.uk:getSignatureFileVersionV1In"
                style="document"/>
                <wsdl:input><soap:body use="literal"/></wsdl:input>
                <wsdl:output><soap:body use="literal"/></wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getSignatureFileV1">
                <soap:operation soapAction="http://pronom.nationalarchives.gov.uk:getSignatureFileV1In"
                style="document"/>
                <wsdl:input><soap:body use="literal"/></wsdl:input>
                <wsdl:output><soap:body use="literal"/></wsdl:output>
        </wsdl:operation>
</wsdl:binding>

<wsdl:message name="getSignatureFileVersionV1In">
        <wsdl:part name="messagePart" element="tns:getSignatureFileVersionV1"/>
</wsdl:message>
        <wsdl:message name="getSignatureFileVersionV1Out">
        <wsdl:part name="messagePart" element="tns:getSignatureFileVersionV1Response"/>
</wsdl:message>
<wsdl:message name="getSignatureFileV1In">
        <wsdl:part name="messagePart" element="tns:getSignatureFileV1"/>
</wsdl:message>
<wsdl:message name="getSignatureFileV1Out">
        <wsdl:part name="messagePart" element="tns:getSignatureFileV1Response"/>
</wsdl:message>
```

Quadro 2 Identificação das operações do PRONOM

Na sequência, é possível visualizar no Quadro 3 Especificação da classe "SigFile" a classe “SigFile” que é a classe de retorno da consulta pelo serviço do PRONOM, logo em seguida, no Quadro 4 Especificação das classes "InternalSignatureType" e “ByteSequenceType”, está o mapeamento da classe “InternalSignatureType” a qual corresponde à classe “InternalSignature” especificada no diagrama de classe abordado no Quadro 4

```
<xs:complexType name="SigFile"><xs:all>
                <xs:element name="FFSignatureFile" type="tns:SignatureFileType">
                        <xs:key name="FormatIdKey">
                                <xs:selector xpath="FileFormatCollection/FileFormat"/>
                                <xs:field xpath="@ID"/>
                        </xs:key>
                        <xs:key name="SignatureIdKey">
                                <xs:selector xpath="InternalSignatureCollection/InternalSignature"/>
                                <xs:field xpath="@ID"/>
                        </xs:key>
                        <xs:keyref name="fileformat-haspriorityover-formatid" refer="tns:FormatIdKey">
                                <xs:selector xpath="FileFormatCollection/FileFormat/HasPriorityOverFileFormatID"/>
                                <xs:field xpath="*"/>
                        </xs:keyref>
                        <xs:keyref name="fileformat-to-signatureid" refer="tns:SignatureIdKey">
                                <xs:selector xpath="FileFormatCollection/FileFormat/InternalSignatureID"/>
                                <xs:field xpath="*"/>
                        </xs:keyref>
                </xs:element>
        </xs:all>
</xs:complexType>
```

Quadro 3 Especificação da classe "SigFile"

```
<xs:complexType name="InternalSignatureType">
        <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element name="ByteSequence" type="tns:ByteSequenceType"/>
        </xs:choice>
        <xs:attribute name="ID" type="xs:nonNegativeInteger" use="required"/>
        <xs:attribute name="Specificity" use="required">
                <xs:simpleType>
                        <xs:restriction base="xs:string">
                                <xs:enumeration value="Generic"/>
                                <xs:enumeration value="Specific"/>
                        </xs:restriction>
                </xs:simpleType>
        </xs:attribute>
</xs:complexType>
<xs:complexType name="ByteSequenceType">
         <xs:choice minOccurs="0" maxOccurs="unbounded">
                 <xs:element name="SubSequence" type="tns:SubSequenceType"/>
         </xs:choice>
         <xs:attribute name="Reference" use="optional">
                 <xs:simpleType>
                         <xs:restriction base="xs:string">
                                 <xs:enumeration value="BOFoffset"/>
                                 <xs:enumeration value="EOFoffset"/>
                                 <xs:enumeration value="IndirectBOFoffset"/>
                                 <xs:enumeration value="IndirectEOFoffset"/>
                                 <xs:enumeration value="NOoffset"/>
                         </xs:restriction>
                 </xs:simpleType>
         </xs:attribute>
         <xs:attribute name="Endianness" use="optional">
                 <xs:simpleType>
                         <xs:restriction base="xs:string">
                                 <xs:enumeration value="Big-endian"/>
                                 <xs:enumeration value="Little-endian"/>
                         </xs:restriction>
                 </xs:simpleType>
         </xs:attribute>
         <xs:attribute name="IndirectOffsetLocation" use="optional"/>
         <xs:attribute name="IndirectOffsetLength" use="optional"/>
</xs:complexType>
```

Quadro 4 Especificação das classes "InternalSignatureType" e “ByteSequenceType”

Na sequência a classe “SubSequenceType”, responsável por prover as informações necessárias para o cálculo da assinatura Interna, pode ser visualizada no Quadro 5.

```
<xs:complexType name="SubSequenceType">
        <xs:sequence>
                <xs:element name="Sequence" type="tns:HexBytes"/>
                <xs:element name="DefaultShift" type="xs:integer"/>
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                        <xs:element name="Shift" type="tns:ShiftType"/>
                        <xs:element name="LeftFragment" type="tns:FragmentType"/>
                        <xs:element name="RightFragment" type="tns:FragmentType"/>
                </xs:choice>
        </xs:sequence>
        <xs:attribute name="Position" type="xs:integer" use="required"/>
        <xs:attribute name="SubSeqMinOffset" type="xs:integer" use="required"/>
        <xs:attribute name="SubSeqMaxOffset" type="xs:integer" use="optional"/>
        <xs:attribute name="MinFragLength" type="xs:integer" use="required"/>
</xs:complexType>
```

Quadro 5 Especificação da classe "SubSequenceType " *WSDL* do Servidor

A descrição das mensagens necessárias para o funcionamento do serviço descritas com a linguagem WSDL estão no Quadro 6, aonde são descritas as mensagens utilizadas pelo serviço, sendo que qualquer aplicação que queira utilizar as funcionalidades desse serviço tem que utilizar essa especificação.

```
- <definitions targetNamespace="http://pd.me.edu/" name="SIPreDService">
  - <types>
    - <xsd:schema>
        <xsd:import namespace="http://pd.me.edu/" schemaLocation="http://localhost:8080/SIPreD/SIPreDService?xsd=1"/>
      </xsd:schema>
    </types>
  - <message name="extensao">
      <part name="parameters" element="tns:extensao"/>
    </message>
  - <message name="extensaoResponse">
      <part name="parameters" element="tns:extensaoResponse"/>
    </message>
  - <message name="file">
      <part name="parameters" element="tns:file"/>
    </message>
  - <message name="fileResponse">
      <part name="parameters" element="tns:fileResponse"/>
    </message>
  - <portType name="SIPreD">
    - <operation name="extensao">
        <input wsam:Action="http://pd.me.edu/SIPreD/extensaoRequest" message="tns:extensao"/>
        <output wsam:Action="http://pd.me.edu/SIPreD/extensaoResponse" message="tns:extensaoResponse"/>
      </operation>
    - <operation name="file">
        <input wsam:Action="http://pd.me.edu/SIPreD/fileRequest" message="tns:file"/>
        <output wsam:Action="http://pd.me.edu/SIPreD/fileResponse" message="tns:fileResponse"/>
      </operation>
    </portType>
  - <binding name="SIPreDPortBinding" type="tns:SIPreD">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    - <operation name="extensao">
        <soap:operation soapAction=""/>
      - <input>
          <soap:body use="literal"/>
        </input>
      - <output>
          <soap:body use="literal"/>
        </output>
      </operation>
    - <operation name="file">
        <soap:operation soapAction=""/>
      - <input>
          <soap:body use="literal"/>
        </input>
      - <output>
          <soap:body use="literal"/>
        </output>
      </operation>
    </binding>
  - <service name="SIPreDService">
    - <port name="SIPreDPort" binding="tns:SIPreDPortBinding">
        <soap:address location="http://localhost:8080/SIPreD/SIPreDService"/>
      </port>
    </service>
</definitions>
```

Quadro 6 WSDL do SIPreD

Já no Quadro 7 estão descritos os atributos da classe “Registro”, que será retornado de acordo com a consulta escolhida pelo cliente, seja por arquivo ou por extensão, que também pode ser visto no diagrama de classe da Figura 14.

```
- <definitions targetNamespace="http://pd.me.edu/" name="SIPreDService">
  - <types>
    - <xsd:schema>
        <xsd:import namespace="http://pd.me.edu/" schemaLocation="http://localhost:8080/SIPreD/SIPreDService?xsd=1"/>
      </xsd:schema>
    </types>
  - <message name="extensao">
      <part name="parameters" element="tns:extensao"/>
    </message>
  - <message name="extensaoResponse">
      <part name="parameters" element="tns:extensaoResponse"/>
    </message>
  - <message name="file">
      <part name="parameters" element="tns:file"/>
    </message>
  - <message name="fileResponse">
      <part name="parameters" element="tns:fileResponse"/>
    </message>
  - <portType name="SIPreD">
    - <operation name="extensao">
        <input wsam:Action="http://pd.me.edu/SIPreD/extensaoRequest" message="tns:extensao"/>
        <output wsam:Action="http://pd.me.edu/SIPreD/extensaoResponse" message="tns:extensaoResponse"/>
      </operation>
    - <operation name="file">
        <input wsam:Action="http://pd.me.edu/SIPreD/fileRequest" message="tns:file"/>
        <output wsam:Action="http://pd.me.edu/SIPreD/fileResponse" message="tns:fileResponse"/>
      </operation>
    </portType>
  - <binding name="SIPreDPortBinding" type="tns:SIPreD">
      <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    - <operation name="extensao">
        <soap:operation soapAction=""/>
      - <input>
          <soap:body use="literal"/>
        </input>
      - <output>
          <soap:body use="literal"/>
        </output>
      </operation>
    - <operation name="file">
        <soap:operation soapAction=""/>
      - <input>
          <soap:body use="literal"/>
        </input>
      - <output>
          <soap:body use="literal"/>
        </output>
      </operation>
    </binding>
  - <service name="SIPreDService">
    - <port name="SIPreDPort" binding="tns:SIPreDPortBinding">
        <soap:address location="http://localhost:8080/SIPreD/SIPreDService"/>
      </port>
    </service>
  </definitions>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Quadro 7 Especificação da classes do SIPreD

3.4 VALIDAÇÃO DO SIPRED

Para validar e comprovar o funcionamento do serviço de busca por extensão do arquivo foi planejada a conexão de possíveis clientes para consumir o serviço de PD proposto. Esta validação composta pela implementação de um cliente em linguagem Java e outro em C#, para que fosse visualizada a ideia de interoperabilidade e reutilização de código.

Na Figura 18 é possível verificar a tela de consulta por extensão implementada em Java.

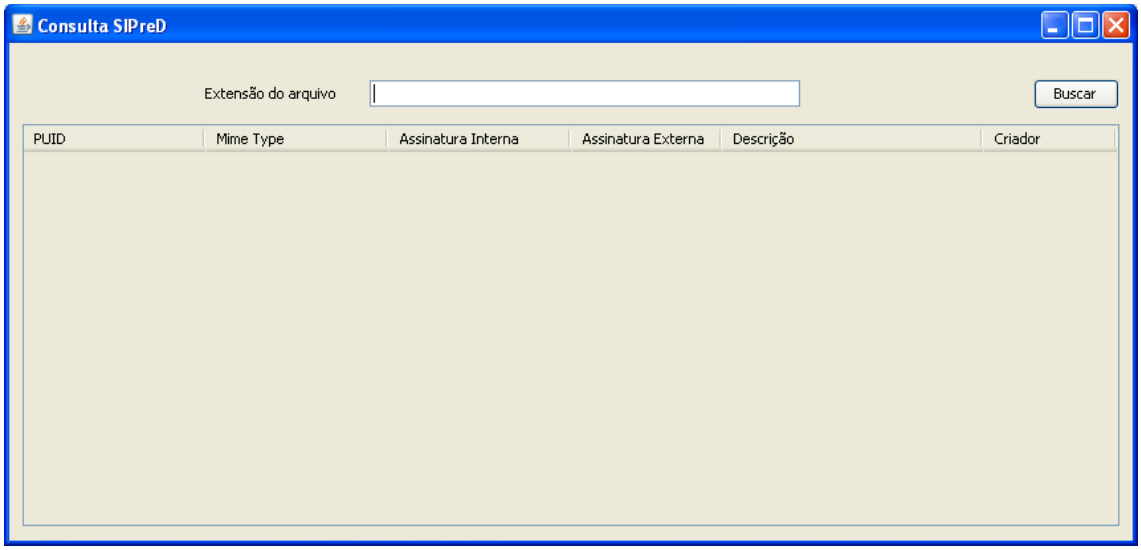

Figura 18 Tela de consulta por extensão em Java

Na Figura 19 é possível verificar a tela de consulta por extensão implementada em Java, com os dados retornados pelo SIPreD.

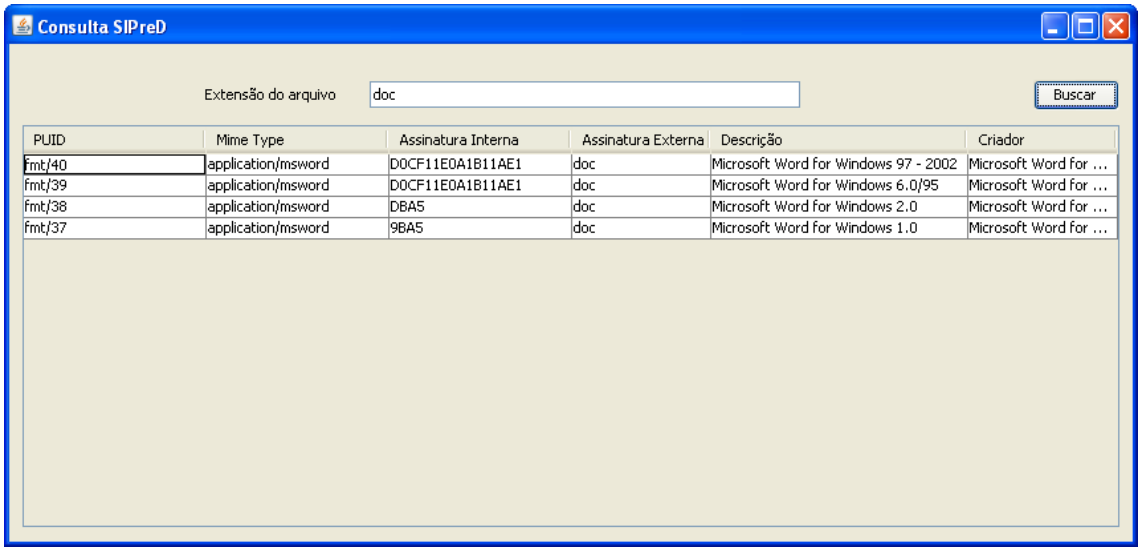

Figura 19 Dados da consulta por extensão em Java

Na Figura 20 é possível verificar a tela de consulta por extensão implementada em C#.

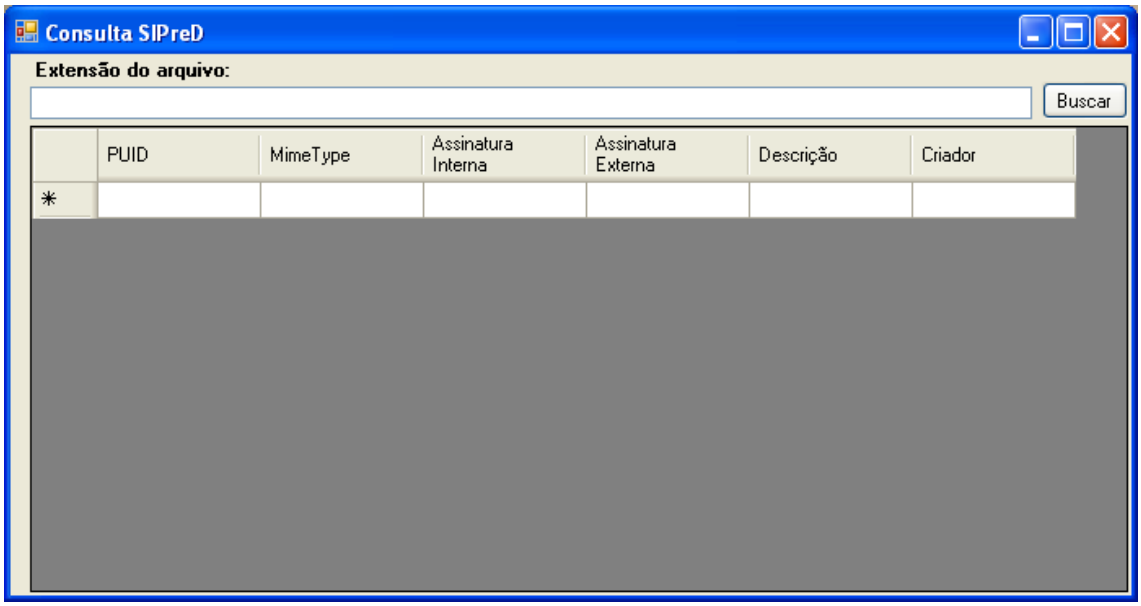

Figura 20 Tela de consulta por extensão em C#

Na Figura 21 é possível verificar a tela de consulta por extensão implementada em C#, com os dados retornados pelo SIPreD.

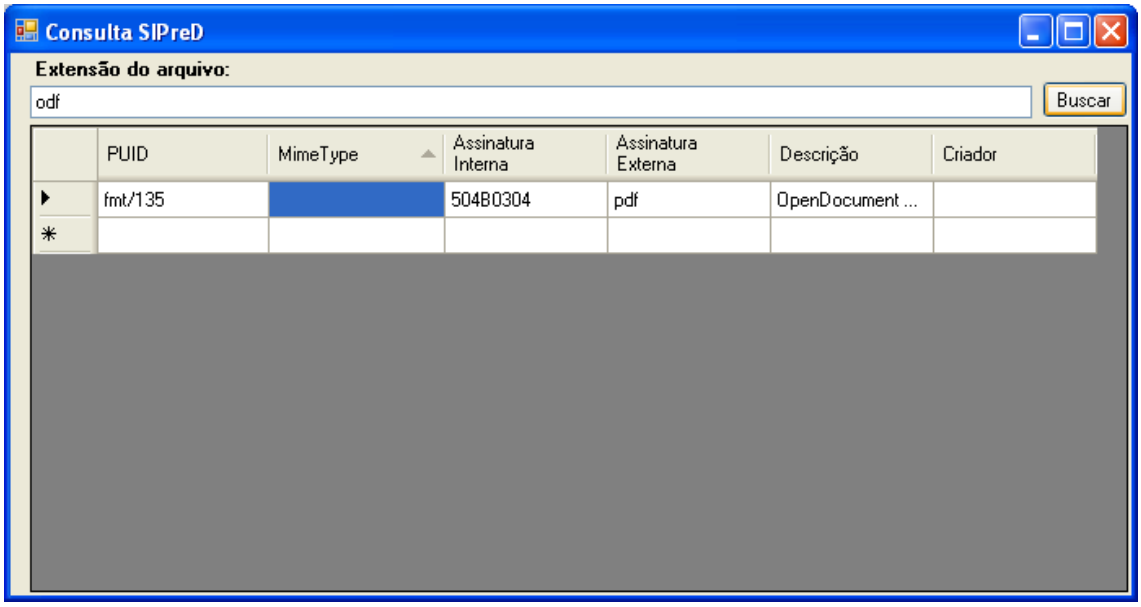

Figura 21 Dados da consulta por extensão em C#

Já a segunda parte da validação ocorreu com a implementação simples de um cliente em Java, cujo método (operação) a ser chamado foi o file, enviando para o servidor um arquivo convertido para um array de bytes, como é possível visualizar no Quadro 8 a seguir.

```
public static void main(String[] args) {

    try {
         SIPreDService service = new SIPreDService(new URL("http://localhost:8080/SIPreD/SIPreDService?WSDL"),
                 new QName("SIPreDService"));
        SIPreD port = service.getSIPreDPort();

    FileInputStream in = null;
        File file = new File("C:\\1.doc");
        in = new FileInputStream(file);
        byte[] bytes  = new byte[in.available()];
        in.read(bytes);
        List lista = port.file(bytes);
        while(!lista.isEmpty()){
            Registro reg = (Registro) lista.iterator().next();
          System.out.println("Mime type "+reg.getMimeType());
            System.out.println("assinatura externa "+reg.getAssinaturaExterna());
            System.out.println("Assinatura interna "+reg.getAssinaturaInterna());
            System.out.println("especificidade "+reg.getEspecificidade());
            System.out.println("nome "+reg.getName());
            System.out.println("PUID "+reg.getPuid());
            System.out.println("-------------------------");
            lista.remove(lista.iterator().next());
        }
```

Quadro 8 Cliente para consumir metodo file em Java

A saída do método especificado na Quadro 8 acima pode ser observada no Quadro 9

```
Mime type application/msword
assinatura externa doc
Assinatura interna D0CF11E0A1B11AE1
especificidade Specific
nome Microsoft Word for Windows 97-2002
PUID fmt/40
--------------------------
Mime type application/msword
assinatura externa doc
Assinatura interna D0CF11E0A1B11AE1
especificidade Specific
nome Microsoft Word for Windows 6.0/95
PUID fmt/39
--------------------------
CONSTRUÍDO COM SUCESSO (tempo total: 2 minutos 10 segundos)
```

Quadro 9 Saída do método file em Java

4 CONCLUSÃO

Este trabalho propôs testar a consumação de serviços web inserido no processo de ingestão de dados encontrados na literatura e propor um serviço de Preservação Digital, utilizando WS a fim de consumir tais serviços e compor uma arquitetura de serviços de preservação digital, proposta pelo grupo de trabalho em PD da UFPR. Tais objetivos foram alcançados integralmente e tais resultados foram comprovados por meio da utilização dos serviços oferecidos pelo PRONOM e por implementação do SIPreD A validação e testes realizados no SIPreD através de conexão com clientes (implementados em Java e C#) está disponível para ser incorporado na arquitetura proposta.

Observou-se que a utilização de *WS* por serviços de PD na literatura tornou-se abrangente, pois muitos projetos como CAIRO (THOMAS, 2008), *DROID*(The National Archives) (BROWN, 2006) e *PREMIS*(HITCHCOCK, et al., 2007) utilizam tal tecnologia, o que facilitou o desenvolvimento deste trabalho. O modelo de arquitetura OAIS de acordo com (ARELLANO, 2004), (FERREIRA, 2011), (FERREIRA, 2006) e (MARCONDES, et al., 2009) surgiu para padronizar os SPD, e baseado nesse, o grupo de trabalho em PD da UFPR propôs uma arquitetura de processo da PD, na qual o serviço deste trabalho se insere. Sendo esse serviço proposto elaborado a fim de demonstrar as possíveis utilizações de *WS* para a implementação de serviços para a PD, com todas as características requeridas por esta e ainda oferecendo vantagens adicionais como baixo custo na manutenção tecnológica, pois a tecnologia utilizada dispõe de possibilidade de interoperabilidade, portabilidade e diminuição nos riscos de obsolescência tecnológica por ser implementar a arquitetura *SOA* como corrobora (BEAN, 2010), (CERAMIS, 2002), (ERL, 2009), (FARIA, et al., 2010), (FEUERLICHT, et al., 2008), (FILAGRANA, 2008) e (KRAFZIG, et al., 2005).

Podem-se destacar como principais contribuições deste trabalho a utilização de *WS* por sua arquitetura SOA, em serviços de PD, além da sua possível reutilização processos de PD diferenciados por conter as características de portabilidade, interoperabilidade e possibilitar sua consumação por diferentes linguagens de programação.

Como trabalhos futuros sugere-se a implementação de subprocessos posteriores e complementares para possível incorporação de um subprocesso de ingestão de dados completo, e este por sua vez incorporando o processo total de PD.

REFERÊNCIAS

AALST, Wil van der. “**Don’t Go with the flow: Web Services Composition Standards Exposed.**” (IEEE Intelligent Systems) 18, n. 1 (2003).

ARELLANO, Miguel Angel. "**Preservação de documentos digitais**." *site da SciELO no Brasil.*2004. http://www.scielo.br/scielo.php?pid=S0100-19652004000200002&script=sci_arttext (accessed 06 01, 2011).

BAPTISTA, Ana Alice, Miguel FERREIRA, e José Carlos RAMALHO. **“Avaliação Automática de Migração em Redes Distribuídas de Conversores.”**Bragança, 2005.

BEAN, James. ***SOA and Web Services Interface Design: Principles, Techniques, and Standards.*** Burlington: Elsevier Inc., 2010.

BENNY, Mathew, Matjaz B. JURIC, e Sarang POORNACHANDRA. ***Business Process Execution Language for Web Service****.* Vol. II. Birmingham: PACKT, 2006.

BROWN, Adrian. ***Digital Preservation Technical Paper 1: Automatic Format Identification Using PRONOM and DROID.*** The National Archives, 2006.

CERAMIS, Ethan. ***Web services essentials****.* Vol. I. Sebastopol: O´Reilly, 2002.
CONWAY, P. **Preservação no universo digital**. 2 ed. Rio de Janeiro: Projeto Conservação Preventiva em Bibliotecas e Arquivos: Arquivo Nacional, 2001.

ENDO, André Takeshi. **“Teste de composicao de web services: Uma estrategia baseada em um modelo de teste de programas paralelos.”**São Carlos, 2008.

ERL, Thomas. ***Service-oriented architecture: concepts, technology, and design.***Boston: Prentice Hall Professional Technical Reference, 2009.

FARIA, Flávio de Paula, e Leonardo Guerreiro AZEVEDO. ***Um Estudo Sobre Mashup Para O Desenvolvimento De Aplicações Em Uma Abordagem Soa****.* Rio de Janeiro: Andréa Magalhães, 2010.

FERREIRA, Carla Alexandra Silva. **“Preservação da Informação Digital: uma perspectiva orientada para as bibliotecas.**” Coimbra, 2011.

FERREIRA, Miguel. ***Introdução à Preservação Digital****.* Guimarães , Portugal: Escola de Engenharia da Universidade do Minho, 2006.

FEUERLICHT, George, e Winfried LAMERSDORF. ***Service-Oriented Computing - ICSOC 2008 Workshops: ICSOC 2008, International Workshop.***Sidney: Springer, 2008.

FILAGRANA, Ivan Correia. **“Um Estudo Sobre Mashup Para O Desenvolvimento De Aplicações Em Uma Abordagem Soa.”** Itajai, Julho de 2008.

FREIRE, Cassio Jose Santos. **“Regente: Um Arcabouço Para Gerenciamento Eficiente De Orquestrações De Serviços Web.”** Rio de Janeiro, 2007.

HITCHCOCK STEVE [et al.] **Preservation Metadata for Institutional Repositories: applying PREMIS**. - Southampton : University of Southampton, 2007.

KRAFZIG, Dirk, Karl BANKLE, e dirk SLAMA. ***Enterprise SOA: service-oriented architecture best practices****.*Hagerstown: Pearson Education Inc., 2005.

MARCONDES, Carlos Henrique, Flavia Garcia ROSA, Luis SAYÃO, e Lídia Brandão TOUTAIN. ***Implantaçãoe gestão de repositórios institucionais***. Salvador: Editora da Universidade Federal da Bahia, 2009.The National Archives. ***The National Archives.***http://www.nationalarchives.gov.uk/aboutapps/pronom/ (acesso em 20 de 05 de 2011).

THOMAS, Susan,**Cairo Project** [Relatório]. - Oxford : Bodleian Library, 2008.