

Proposta de uma arquitetura e modelo para criação de um Web Service Semântico para automatização de processo de negócio

Carolina dos Santos Oliveira Viana

carolina.sov@gmail.com

IFF

Simone Vasconcelos Silva

simonevsinfo@gmail.com

IFF

Mark Douglas de Azevedo Jacyntho

markjacyntho@gmail.com

IFF

Resumo: Com a crescente evolução da tecnologia da informação, as empresas nos dias de hoje estão cada vez mais recorrendo a tecnologia para automatizar os seus processos de negócio que podem ser assistidos por máquina. Mas nem todo o potencial das máquinas são explorados pois normalmente os dados de negócio são salvos de forma que a máquina não consegue entender o seu significado. Com a web semântica é possível mudar isso acrescentando significado semântico aos dados. Este trabalho tem como objetivo propor uma arquitetura de um web service semântico RESTful para execução automática de processos de negócio e armazenamento dos dados deste negócio em um banco de dados semântico nativo. Foi feita uma revisão bibliográfica para levantamento de dados, e a aplicação da arquitetura em um estudo de caso realista para criação de um protótipo. A implementação do protótipo foi feita a partir de uma modelagem do processo utilizando a notação BPMN (Business Process Management Notation), ontologias, ferramentas e tecnologias consagradas da web semântica.

Palavras Chave: web service - modelagem - processo - semântico -

1. INTRODUÇÃO

Uma empresa é composta por processos de negócio interativos que formam um sistema delicado, que devem ser equilibrados (ABPMP, 2013). As empresas nos dias atuais vêm cada vez mais definindo processos de negócio para governar a execução de processos tanto manuais quanto automatizados.

Com o passar dos anos, é crescente o aumento da utilização de sistemas computacionais para auxílio nas tarefas que podem ser automatizadas ou assistidas por computadores. No contexto de processos de negócios, a evolução da tecnologia da informação tornou possível a automação de alguns desses processos. Esta interação é extremamente importante para lidar de forma eficiente com processos que geram grandes quantidades de dados, e ao mesmo tempo, proporcionar uma forma mais rápida de avaliação e melhoramento desses processos (JACYNTHO; SCHWABE, 2010).

Todo processo de negócio gera dados de negócio, no entanto essas informações não são compreensíveis por máquina. Com o advento da *web* semântica é possível adicionar significado inteligível por máquina a essas informações por meio de modelos formais de representação do conhecimento conhecido como ontologias. As tecnologias da *web* semântica (ontologias, banco de dados semânticos, raciocinadores, linguagens de consulta, etc.) contam com ferramentas e métodos escaláveis para fazer representações de dados de forma que as máquinas consigam fazer a manipulação de conhecimento (BERNERS-LEE; HENDLER; LASSILA, 2001).

Dentro dos sistemas computacionais o *web service* vem se destacando pela sua flexibilidade de reuso desta camada de serviço por diferentes tipos de *front-end* (*mobile*, *desktop*, *web*, etc.) assim como pela simplicidade trazida pelo protocolo HTTP. No entanto, os *web services* trocam informações entre si que não são compreensíveis por máquina. Os *Semantic Web Services* (SWS) são *web services* que são capazes de utilizar conhecimento para realizar diversos tipos de ações como por exemplo automatizar a descoberta de outros servidores, realizar a substituição se necessário por um semelhante em tempo de execução, composição e execução de componentes de *software* entre outros (HEPP et al., 2005).

Desta forma, unindo todos estes conhecimentos apresentados, é possível alimentar uma base de dados semântica com dados de negócio produzidos e manipulados por processos de negócio, adicionando significado explícito e formal por meio das tecnologias e padrões da *web* semântica a estes dados, contidos em um *web service* semântico.

Este trabalho tem como objetivo principal propor uma arquitetura baseada no estilo arquitetural *RESTful* para execução de processos de negócios modelados com a notação BPMN, adicionando significado explícito as informações processadas ao longo do mesmo, criando uma base de conhecimento semântica totalmente inteligível por máquina e que pode ser utilizada no processo de tomada de decisão. Como objetivo secundário, definir um processo de negócio e desenvolver um protótipo corroborando a eficácia da arquitetura por meio de um estudo de caso realista utilizando ferramentas e ontologias.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. PROCESSOS DE NEGÓCIOS

O termo “negócio” abrange todos os tipos de organizações, públicas ou privadas, com ou sem fins lucrativos, de qualquer porte ou segmento, e representa pessoas interagindo na execução de um conjunto de atividades que geram valor para os clientes (um produto ou

serviço). Já a palavra “processo” remete a união de atividades e comportamento que podem ser executados por máquinas ou humanos para alcançar um resultado. Desta forma, processos de negócio são processos dentro de uma organização compostos por atividades governadas por regras de negócio, relacionadas entre si para solução de uma questão específica, que entrega valor para os clientes ou apoia outros processos fornecendo uma visão de sequência e fluxo. Estas atividades podem ser interfuncional ou até mesmo interorganizacional. Processos de negócio podem ser classificados em três tipos (ABPMP, 2013):

- Processo primário – Atividades essenciais da organização.
- Processo de suporte – Atividades que agregam valor a outros processos da organização.
- Processo de gerenciamento – Atividades de gerenciamento, como por exemplo atividades de mediação, monitoramento, controle, administração entre outros.

A tendência atual das organizações públicas e/ou privadas é de concentrar suas atividades em processos organizacionais e multifuncionais que agregam valor aos clientes. O gerenciamento de processos de negócio (BPM - *Business Process Management*) é uma disciplina gerencial que trata processos de negócio como ativos da organização e integra os objetivos e estratégias desta organização, com as necessidades e expectativas dos clientes, focando em processos de ponta a ponta. BPM se caracteriza como um ciclo de vida contínuo de atividades integradas. Engloba estratégias, objetivos, cultura, estruturas organizacionais, papéis, políticas, métodos e tecnologias para analisar, desenhar, implementar, gerenciar desempenho, transformar e estabelecer a governança de processos (SILVA; ROSA, 2017). Os objetivos organizacionais podem ser alcançados por meio da definição, desenho, controle e transformação contínua de processos de negócio.

A modelagem de processos de negócio é uma etapa muito importante no gerenciamento de processos de negócio, BPM é composto por uma coleção de métodos e ferramentas voltadas para a tarefa de modelagem, gestão e análise de processos. Na etapa de modelagem são feitos desenhos dos processos de negócios em uma visão gráfica de um fluxo de trabalho (*workflow*), com o objetivo de representar os processos da organização da forma atual (*As Is*) para analisar e aperfeiçoar, formando novas versões dos processos (*To Be*), que posteriormente podem ser implementados e monitorados (STEIN; FREITAS; THOM, 2019).

Business Process Management Notation (BPMN) é a notação padrão escolhida para esta etapa. Lançada em 2004 pela *Business Process Management Initiative* (BPMI), e mantida pela *Object Management Group* (OMG) desde 2006. Em 2011 foi lançada a versão BPMN 2.0, padrão ISO (STEIN; FREITAS; THOM, 2019). Um modelo de processo de negócio implementado em BPMN é uma representação gráfica de um determinado processo, um *workflow*, composto de tarefas a serem completadas com objetivo de produzir um produto ou executar um serviço (KLUZA et al., 2015). Atualmente é amplamente utilizada tanto em meios acadêmicos como profissionais, por proporcionar um nível aprofundado de detalhamento (SILVA; ROSA, 2017).

2.2. WEB SEMÂNTICA

O conteúdo produzido para *web* tem como objetivo o entendimento de seu significado para humanos, os dados não são estruturados o que torna complexo o entendimento semântico para máquinas. O objetivo da *web* semântica é criar uma estrutura através de tecnologias e padrões que adicione significado formal e explícito que suportem o entendimento de dados pelas máquinas, permitindo assim que agentes de *software* possam executar tarefas mais sofisticadas para usuários (BERNERS-LEE; HENDLER; LASSILA, 2001).

A *web* semântica não é uma *web* separada, ela é uma extensão da *web* atual com o adicional de incluir significados bem definidos aos dados, tornando-se a *Web* de Dados. A ideia é utilizar o poder da *web* como conhecemos para não somente publicar dados na *web*, como também para interligar esses dados (ou metadados) através de URIs (*Universal Resource Identifier*) permitindo a navegação entre dados (BERNERS-LEE, 2009).

A *web* semântica não busca somente a publicação de dados semânticos na *web*, um dos seus propósitos é a vinculação de dados através da criação de links entre eles, de forma que uma pessoa ou máquina consiga explorar uma grande *Web* de Dados pela navegação entre dados relacionados, assim definindo a *Linked Data*. A publicação de dados estruturados *Linked Data* na *web* devem seguir quatro princípios básicos (BERNERS-LEE, 2009):

- Use URIs como nome para coisas.
- Use URIs HTTP de forma que as pessoas possam procurar por esses nomes;
- Fornecer informações úteis usando os padrões (RDF, SPARQL, etc.) quando alguém acessar essas URIs.
- Incluir *links* para outras URIs, para que eles descubram mais coisas.

A representação do conhecimento para máquinas é feita a partir de coleções estruturadas de dados, e conjuntos de regras de inferências onde elas podem chegar a conclusões através de raciocínio automatizado feito pelos agentes de *software*. O modelo de dados padrão para a *web* semântica é o RDF (*Resource Description Framework*), uma linguagem de propósito geral para processamento de máquinas criado pelo *World Wide Web Consortium* (W3C), com o objetivo de descrever de forma semiestruturada a codificação de metadados de recursos disponibilizados na *Web* possibilitando uma comunicação transparente entre aplicações (JACYNTHO, 2012).

Os dados representados em RDF são estruturados em grafos, no qual a menor unidade de informação é conhecida como tripla: recurso, propriedade e valor. A tripla também pode ser interpretada como sujeito, predicado e objeto de um axioma, e são identificadas por URIs garantindo que a informação esteja conectada não só a palavras em um documento, mas a uma definição única passível de ser encontrada na *web* (BERNERS-LEE; HENDLER; LASSILA, 2001).

Uma ontologia é um modelo formal de uma representação abstrata. É um documento que define formalmente os termos que irão representar e descrever uma determinada área do conhecimento, formando assim, um domínio composto por conceitos como classes, relacionamentos entre termos, regras de inferências e propriedades ou atributos (HEFLIN, 2020). Com auxílio das ontologias os computadores conseguem manipular os termos de forma que faça sentido para o usuário, apesar de ainda não “entender” as informações manipuladas.

A arquitetura REST (*Representational State Transfer*) é um conjunto de princípios de engenharia de *software* composto por seis restrições arquiteturais baseado no protocolo HTTP, com o objetivo de minimizar a latência e a comunicação na rede enquanto maximiza a independência e escalabilidade da implementação dos componentes (FIELDING; TAYLOR, 2000). Dessas seis restrições cinco são obrigatórias e uma opcional, e o serviço que as implementa é chamado de RESTful, tendo como características: desempenho, escalabilidade, simplicidade, modificabilidade, visibilidade, portabilidade e confiabilidade. As seis restrições são:

- A) Interface Uniforme (FREDRICH, s.d.) – Define a interface entre cliente e servidor. Simplifica a arquitetura ao desacoplar, permitindo que as partes evoluam independente uma da outra. Os quatro princípios que guiam esta restrição são:

- Baseado em recursos – Recursos individuais são representados por URIs;
 - Manipulação de recursos através de representações – Se o cliente possuir uma representação do recurso e permissão de modificação, é possível modificar ou excluir o recurso no servidor;
 - Mensagens auto descritivas – Cada mensagem inclui as informações necessárias para ser processada;
 - Hipermídia como motor do estado da aplicação – Quando necessário, URI para recuperação do próprio objeto ou objetos relacionados são fornecidas na requisição.
- B) Sem Estado (FIELDING; TAYLOR, 2000) – Todas as requisições devem conter todas as informações necessárias para o conector entender a requisição, independente de requisições anteriores, desta forma:
- Suspende a necessidade da aplicação salvar o estado entre as requisições, reduzindo o consumo de recursos físicos e aumentando a escalabilidade;
 - Permite interações simultâneas em paralelo;
 - Permite atender requisições isoladas;
 - Força toda a informação que pode contribuir para a reutilização de uma resposta em cache em cada solicitação.
- C) *Cacheable* (FIELDING; TAYLOR, 2000) – Armazenar em cache respostas da interação atual, para serem utilizadas em futuras requisições de interação, com o intuito da diminuição de comunicação em rede ou evitar que o servidor repita o processo de gerar a resposta, em ambos os casos a meta é a redução da latência.
- D) Cliente-Servidor (FREDRICH, s.d.) – Separação de clientes e servidores permite que clientes não se preocupem com armazenamento de dados e servidores não se preocupem com interface do usuário, fazendo com que servidores e clientes possam ser modificados ou substituídos de forma independente desde que a interface não seja alterada.
- E) Sistema em camadas (FREDRICH, s.d.) – O cliente não consegue identificar se está conectado ao servidor final, ou a um intermediário. Servidores intermediários tem o objetivo de melhorar a escalabilidade do sistema como um todo, fazendo caches compartilhados e um balanceamento de carga, permitindo também a imposição de políticas de segurança.
- F) Código sob demanda (opcional)(FREDRICH, s.d.) – Transferência temporária da lógica do servidor para o cliente para que ele possa executar, exemplos: componentes compilados, como miniaPLICATIVOS *Java* e *scripts* do lado do cliente, como *JavaScript*.

3. PROCEDIMENTOS METODOLÓGICOS

Esta pesquisa é classificada em sua natureza como aplicada, pois tem como objetivo a criação de um modelo prático com base em uma pesquisa sistemática. Sua abordagem é classificada como qualitativa. Quanto ao seu objetivo a pesquisa pode ser classificada como exploratória e como procedimento técnico ela se classifica como uma pesquisa bibliográfica. Este trabalho é composto por três diferentes etapas, como pode ser visto a seguir.

3.1. TRABALHOS RELACIONADOS

A primeira etapa deste trabalho consiste em uma revisão sistemática baseada na recomendação PRISMA de revisões sistemáticas, com o objetivo de adquirir uma base inicial de referencial teórico para desenvolvimento das etapas seguintes do trabalho. A recomendação PRISMA tem como base a utilização de métodos sistemáticos e explícitos de buscas em bases de dados para identificar, selecionar e avaliar pesquisas relevantes (MOHER et al., 2015). As buscas foram realizadas em duas bases de dados, *Scopus* e *Web of Science*, conforme mostra o Quadro 1.

Quadro 1 – Estratégia de busca nas bases de dados.

Base	Busca utilizada	Resultado
<i>Web of Science</i>	TÓPICO: (("Semantic Web Services" OR "SWS") AND ("BPM" OR "Business Process Management"))	24
<i>Scopus</i>	Seção I.1 (TITLE-ABS-KEY("Semantic Web Services" OR "SWS") AND TITLE-ABS-KEY("BPM" OR "Business Process Management"))	84

Fonte: Elaboração própria.

A Figura 1 mostra o fluxograma adaptado do método PRISMA para este trabalho.

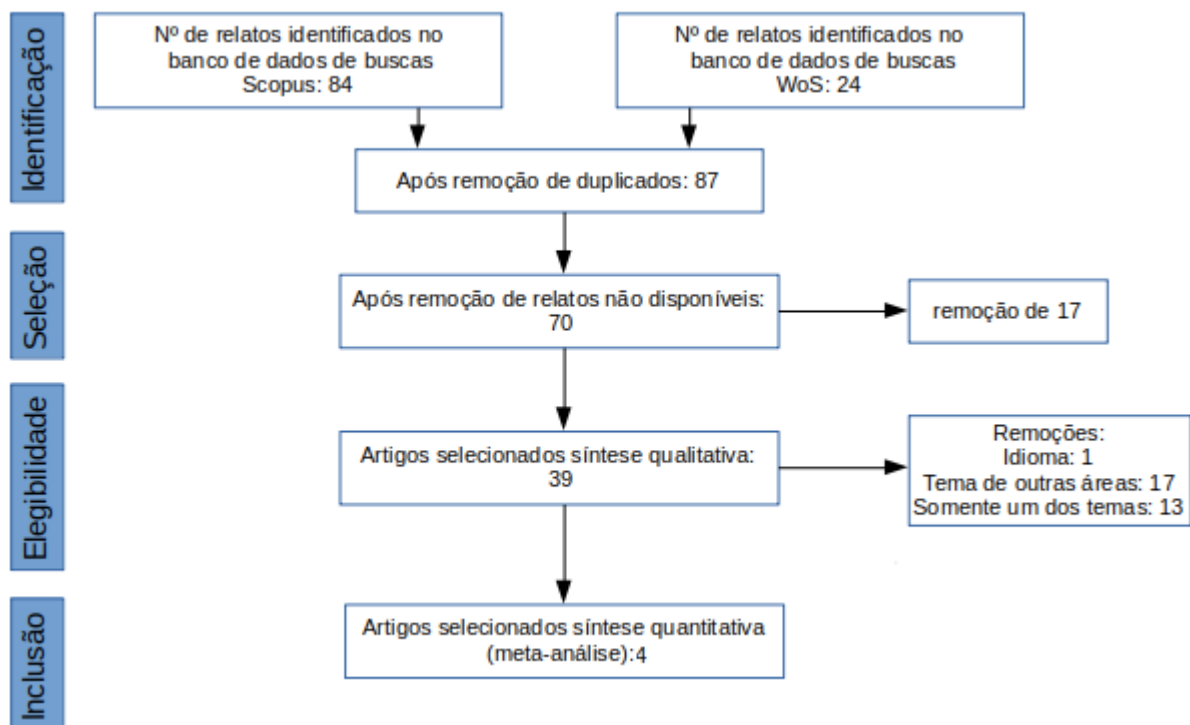


Figura 1 – Fluxo adaptado do método PRISMA.

Fonte: (MOHER et al., 2015) e elaboração própria.

Utilizando as estratégias de busca detalhadas no Quadro 1, foram encontradas um total de 108 publicações nas duas bases de dados, que em sequência passaram pelo fluxo de seleção detalhado na Figura 1. No Quadro 2 a apresentação das publicações selecionadas na etapa de inclusão do modelo, a etapa de síntese quantitativa.

Quadro 2 – Publicações selecionadas.

Autores	Título	Modelagem de processos	Ontologias utilizadas
Dang J; Hedayati A; Hampel K; Toklu C	<i>Personalized Medical Workflow Through Semantic Business Process Management</i>	BPEL	<i>Healthcare ontology</i>
Fonou-Dombeu Jv; Huisman M	<i>Engineering Semantic Web Services For Government Business Processes Automation</i>	BPMN	WSMO
Hepp M; Leymann F; Domingue J; Wahler A; Fensel D	<i>Semantic Business Process Management: A Vision Towards Using Semantic Web Services For Business Process Management</i>	BPEL	WSMO
Zeiner H; Halb W; Lernbei H; Jandl B; Derler C	<i>Making Business Processes Adaptive Through Semantically Enhanced Workflow Descriptions</i>	BPEL	Não citado.

Fonte: Elaboração própria.

O termo *Semantic Business Process Management* (SBPM) surgiu em (HEPP et al., 2005) com a ideia de abordar processos de negócios de um ponto de vista semântico, criando uma ponte entre as duas tecnologia que permitisse criar processos mais flexíveis.

Em (DANG et al., 2009) os autores utilizam um estudo de caso aplicado a um processo de negócio de admissão de pacientes em um hospital para apresentar uma arquitetura orientada para serviços (SOA – *Service-Oriented Architecture*). Na parte de BPM, os autores utilizam uma *engine* de BPEL (*Business Process Execution Language*) com recursos que permitem realizar o acompanhamento de KPIs para mensurar e monitorar o processo. Para adicionar anotação semântica ao processo é definido um modelo de processo baseado em OWL (*Web Ontology Language*) e em seguida é feito um mapeamento entre o processo BPEL e o modelo OWL, o armazenamento de dados não é feito em rdf nativo.

Já em (FONU-DOMBEU; HUISMAN, 2015) os autores propõem a re-engenharia de processos já existentes através de BPMN com anotação semântica dos mesmos, eles propõem também uma infraestrutura para aplicação de *web services* semânticos em ambientes distribuídos tendo como estudo de caso uma estrutura de *e-government* e utilizando a ontologia WSMO (*Web Service Modeling Ontology*).

O trabalho de (ZEINER et al., 2010) tem como propósito apresentar uma metodologia para criar processos de negócio flexíveis e adaptáveis que utilizam descrições de *workflow* melhorados semanticamente. Para um *workflow* ser adaptativo e flexível são necessários vários pequenos passos: inicia-se pela etapa de correspondência, responsável por identificar serviços apropriados entre todos os disponíveis; em seguida passa para etapa de seleção, responsável por escolher entre os serviços identificados na etapa anterior qual melhor se encaixa as necessidades e otimiza a qualidade do *workflow*; e finaliza com a substituição, que irá em tempo de execução substituir o serviço.

3.2. CRIAÇÃO DO MODELO E ARQUITETURA

A segunda etapa deste trabalho consiste na criação da arquitetura (Figura 2) baseada em *web service* semântico RESTful para automatização de processos de negócios.

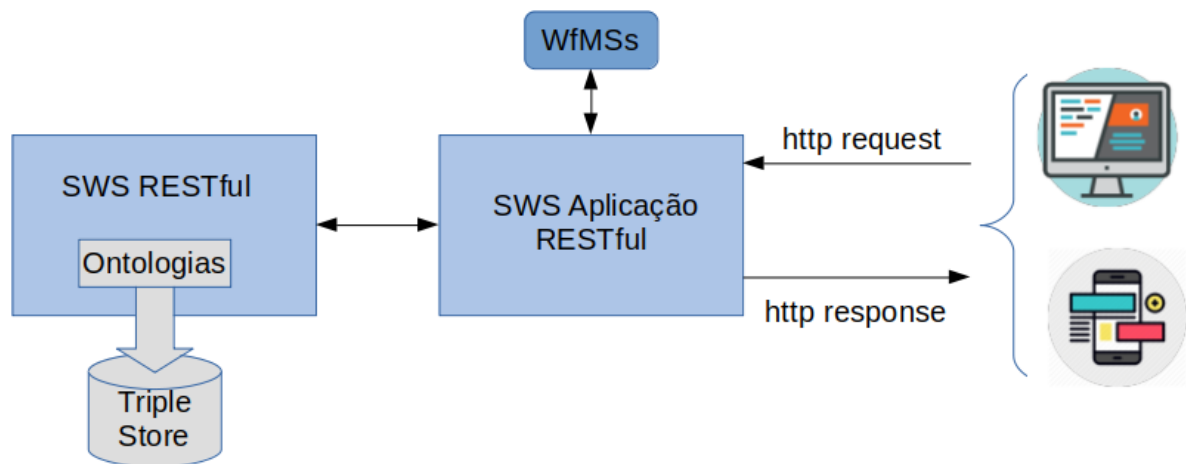


Figura 2: Arquitetura proposta.

Fonte: Elaboração própria.

Na Figura 2 é ilustrada a arquitetura proposta, ela é dividida em três camadas: camada de dados, camada de serviços e camada de aplicação.

A camada de dados é composta por um *web service* semântico RESTful, um banco de dados RDF nativo (*tripe store*) e ontologias ligadas ao negócio que está sendo executado. Todos os dados provenientes do negócio executado na camada de serviços são armazenados de forma enriquecida com significado explícito neste banco de dados.

A camada de serviços é composta por um *web service* semântico e um gerenciador de *workflow* de BPMN 2.0 (WfMSs). O *web service* é anotado semanticamente por uma ontologia de descrição de *web service*, o que permite que ele seja reconhecido em tempo de execução por clientes que estejam precisando deste tipo de processo de negócio para compor o seu próprio processo de negócio flexível. Este *web service* é responsável por fornecer as funcionalidades e a lógica do sistema, para isto conta com um gerenciador de *workflow*, uma *engine* que gerencia o fluxo de um processo modelado em BPMN 2.0.

A camada de aplicação representa o *front-end* da aplicação, pode ser um aplicativo *mobile*, uma aplicação *desktop*, uma aplicação *web*, outros *web services*, etc. Esta camada se comunica através de requisições feitas pelo protocolo HTTP, seguindo os padrões REST.

4. ESTUDO DE CASO REALISTA - ARTIFACTANALYSIS

A implementação da arquitetura proposta em um estudo de caso realista segue a seguinte estrutura:

- A definição do caso;
- Modelagem BPMN;
- Seleção e análise de ontologias;
- Implementação de um protótipo.

4.1. A DEFINIÇÃO DO CASO

Neste exemplo o processo de negócio é chamado de *ArtifactAnalysis* (JACYNTHO; SCHWABE, 2010), ele representa o processo de submissão, revisão e seleção de publicações para uma conferência. Uma conferência tem várias edições, em geral, uma por ano.

Quando o autor submete a publicação o sistema verifica o formato de arquivo enviado, se for o formato incorreto automaticamente o sistema rejeita a publicação. Após passar por essa primeira avaliação, o autor pode a qualquer momento desistir de participar da conferência e retirar a publicação do processo de avaliação.

As publicações possuem tipo (*paper*, *workshop*, e *talk*), autores (*firstAuthor*, *authors*) e *status* (*submitted*, *waitingReviewers*, *inReview*, *reviewed*, *acceptedWaitingLastVersion*, *withoutLastVersion*, *accepted*, *rejected*, e *withdrawn*).

Cada publicação é associada a uma edição de uma conferência. Todas as publicações devem ser designadas a três revisores diferentes que irão emitir a publicação revisada e parecer de recomendação de aprovação ou não. Após revisões, a publicação passa pela aprovação do editor-chefe da conferência. Se aprovada deve ser submetida uma versão final para publicação.

Algumas definições do processo:

- Submeter publicação – A submissão da publicação deve ser feita por um dos autores.
- Designar revisores – O editor-chefe responsável pela conferência deve escolher três revisores para cada publicação recebida.
- Receber publicação – Cada revisor recebe publicações para revisar.
- Submeter publicação revisada – Cada revisor emite sua avaliação sobre a publicação.
- Emitir parecer – O editor-chefe aceita ou rejeita a publicação de acordo com a avaliação dos revisores.
- Submeter versão final – Se a publicação for aprovada o autor pode aceitar as revisões sugeridas e submeter a versão final da publicação.
- Retirar publicação – O autor pode a qualquer momento desistir de participar da conferência e retirar a sua publicação.

4.2. MODELAGEM BPMN

Na Figura 3 é possível ver a modelagem em BPMN do caso proposto. A modelagem foi feita no *software Camunda Modeler* (CAMUNDA, 2020).

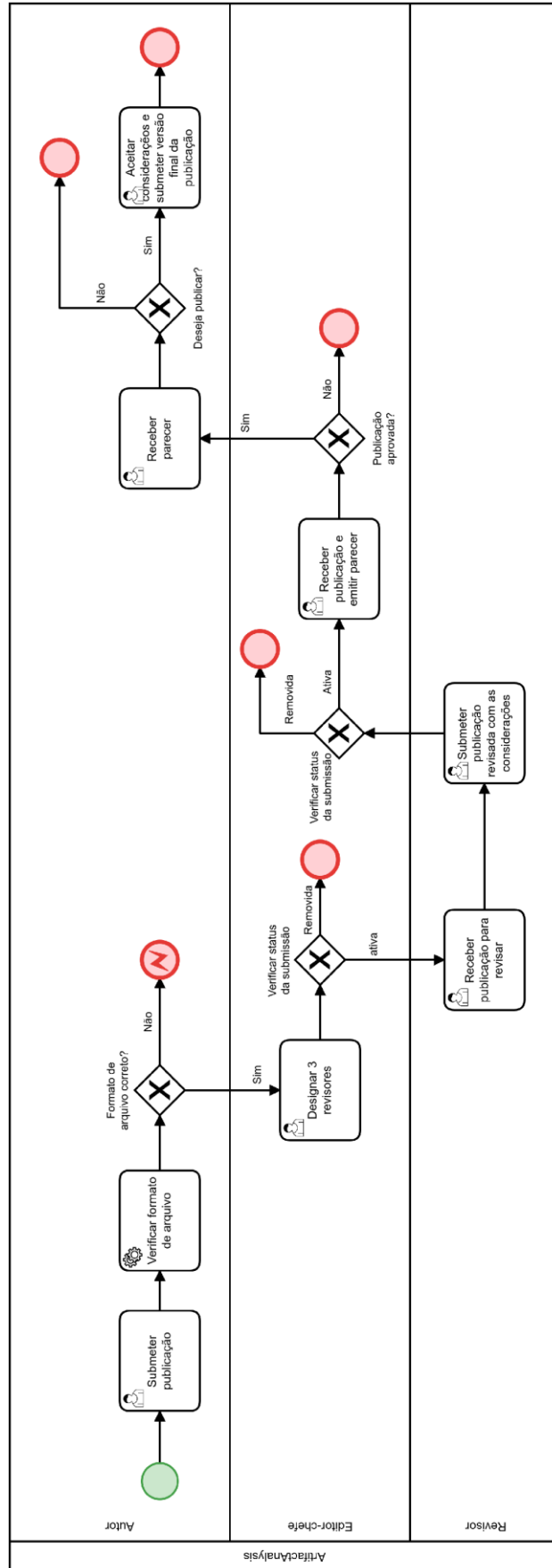


Figura 3: Modelagem BPMN

Fonte: Elaboração própria.

4.3. MODELO ONTOLÓGICO

A ontologia escolhida para enriquecer semanticamente a base de dados de negócio foi a *Schema.org*. Desenvolvida em 2011 pelos gigantes das ferramentas de busca (*Google*, *Microsoft* e *Yahoo*), o *schema.org* representa um conjunto de vocabulários baseados em padrões de sintaxes existentes. Foi criada com o objetivo de unir em um único local que seria amplamente utilizado na *web* por uma variedade de tópicos que inclui temas como: pessoas, lugares, eventos, produtos, ofertas entre outros (GUHA; BRICKLEY; MACBETH, 2015). Na Figura 4 uma adaptação do modelo UML mapeando a ontologia no modelo de classe do negócio.

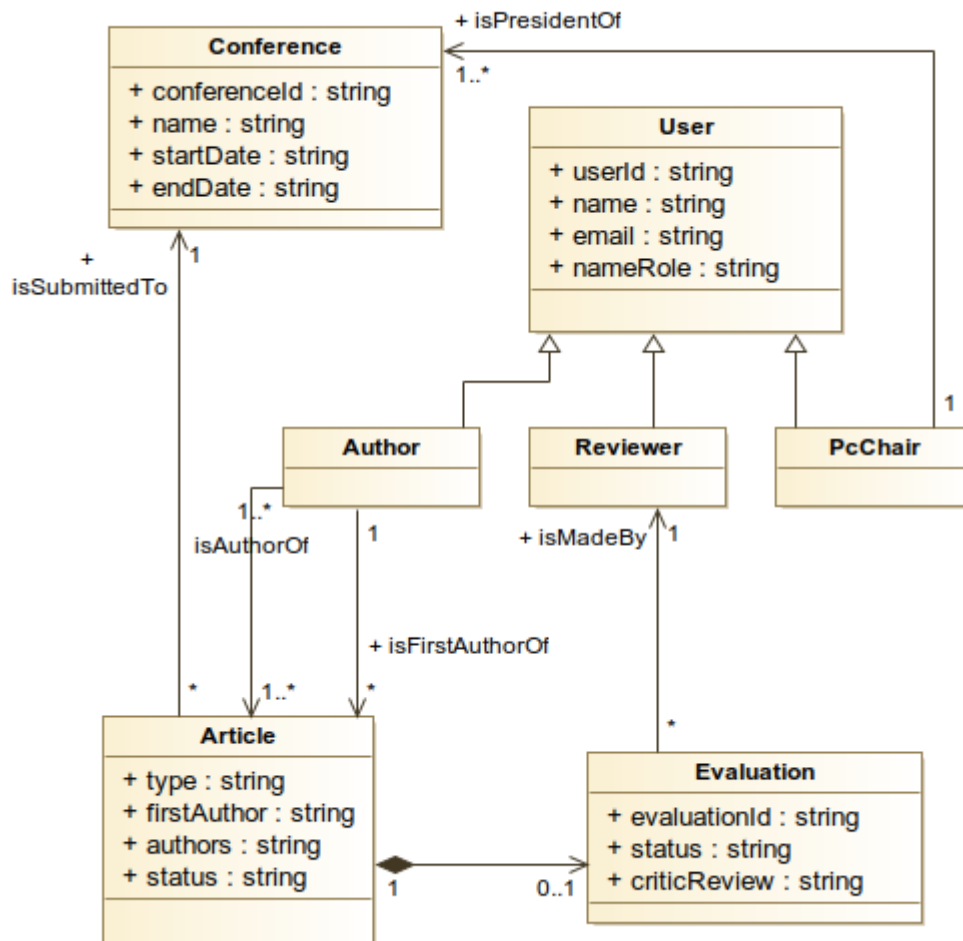


Figura 4: Modelo de classes mapeado em ontologias

Fonte: Elaboração própria.

A ontologia escolhida para identificar semanticamente o *web service* da camada de aplicação foi a *Web Service Modeling Ontology* (WSMO). A WSMO tem como objetivo enriquecer os serviços da *Web* com semântica processável por máquina através de uma linguagem formal para descrever semanticamente todos os aspectos importantes do serviço ofertado para facilitar a descoberta automática desses servidores na *web* (LAUSEN; POLLERES; ROMAN, 2005).

4.4. DETALHES DE IMPLEMENTAÇÃO

Para fins acadêmicos, todas as ferramentas selecionadas possuem licença de utilização do tipo *open-source*. Para implantação do protótipo foram utilizadas as ferramentas descritas no Quadro 3.

Quadro 3: Tecnologias utilizadas.

Tecnologia	Ferramenta utilizada
Linguagem de programação	JAVA
IDE de desenvolvimento	Eclipse
<i>Framework</i> RESTful	Jersey (JAX-RS)
Servidor <i>Web Service</i>	Tomcat
Banco de dados RDF	GraphDB
Framework para manipulação de dados em RDF	RDF4J
<i>Workflow Management</i>	Camunda
Requisição HTTP	Postman

Fonte: Elaboração própria.

Com base no padrão arquitetural REST, foram criados métodos para realização de operações CRUD básicas: *create* (POST), *read* (GET), *update* (PUT) e *delete* (DELETE). Com base nesses métodos e o padrão de projeto orientado a objeto *Command*, foram mapeadas as requisições HTTP com base nas regras de negócios estabelecidas na etapa de modelagem BPMN como descrito no Quadro 4. As operações de POST e PUT consomem dados no formato json, e a operação GET produz dados no formato rdf/xml.

Quadro 4: Requisições de negócio.

	Papel	Método	URI
Submeter publicação / Verificar formato de arquivo	Autor	POST	http://www.exemplo.com/articles
Designar 3 revisores	PcChair	POST	http://www.exemplo.com/articles/{articleId}/allocateReviewers
Receber publicação para revisar	Revisor	GET	http://www.exemplo.com/articles/recieveArticles/{userId}
Submeter considerações	Revisor	PUT	http://www.exemplo.com/articles/{articleId}/evaluations/{evaluationsId}/
Receber publicação	PcChair	GET	http://www.exemplo.com/articles/{articleId}/evaluations/evaluationsCompleted
Emitir parecer	PcChair	PUT	http://www.exemplo.com/articles/{articleId}/
Receber parecer	Autor	GET	http://www.exemplo.com/articles/{articleId}/evaluations/{evaluationsId}/
Aceitar considerações e submeter	Autor	PUT	http://www.exemplo.com/articles/{articleId}/

versão final da publicação			
Remover submissão	Autor	PUT	http://www.exemplo.com/articles/{articleId}/

Fonte: Elaboração própria.

O *Framework* Jersey foi responsável por montar a estrutura RESTful, e o cabeçalho com os métodos responsáveis por executar as requisições HTTP podem ser vistos na Figura 5. Cada entidade possui a sua própria classe *model* com métodos de acesso a dados, que são chamadas de acordo com os parâmetros de requisição.

```
@POST
@Path("/{type}/")
@Consumes("application/json")
public Response post(@PathParam("type") String type, InputStream data)

@GET
@Path("/{type}/{id}/")
@Produces("application/rdf+xml")
public Response getSingle(@PathParam("type") String type, @PathParam("id") int id)

@PUT
@Consumes("application/json")
@Path("/{type}/{id}/")
public Response putPublication(@PathParam("type") String type, @PathParam("id") String id, InputStream data)

@DELETE
@Path("/{type}/{id}/")
public Response deletePublication(@PathParam("type") String type, @PathParam("id")String id)
```

Figura 5: Código Jersey.

Fonte: Elaboração própria.

Através do *framework* Rdf4j foi possível manipular os dados em RDF no banco de dados GraphDB. Na Figura 6 um trecho do código responsável por fazer uma *query* para encontrar todos os artigos salvos no banco de dados.

```
public static String getArticles() {
    StringBuilder resultString = new StringBuilder();

    try (RepositoryConnection conn = repository.getConnection()) {

        String queryString = "PREFIX schema: <http://schema.org/>\n"+
            "select * where {\n"+
            "?s a schema:Article .\n"+
            "}";

        TupleQuery query = conn.prepareTupleQuery(queryString);
        TupleQueryResult result = query.evaluate();

        while(result.hasNext()){
            resultString.append(result.next().toString() + "\n");
        }

    } finally {
        repository.shutdown();
    }
    return resultString.toString();
}
```

Figura 6: Código do método GET artigos.

Fonte: Elaboração própria.

Os métodos desenvolvidos foram testados através do *software* de requisição HTTP *Postman*, na Figura 7 um exemplo de requisição POST na URI `http://localhost:8090/dataSWS/articles` para criação de um artigo. A operação teve como retorno o código 201 – indicando que o artigo foi criado com sucesso.

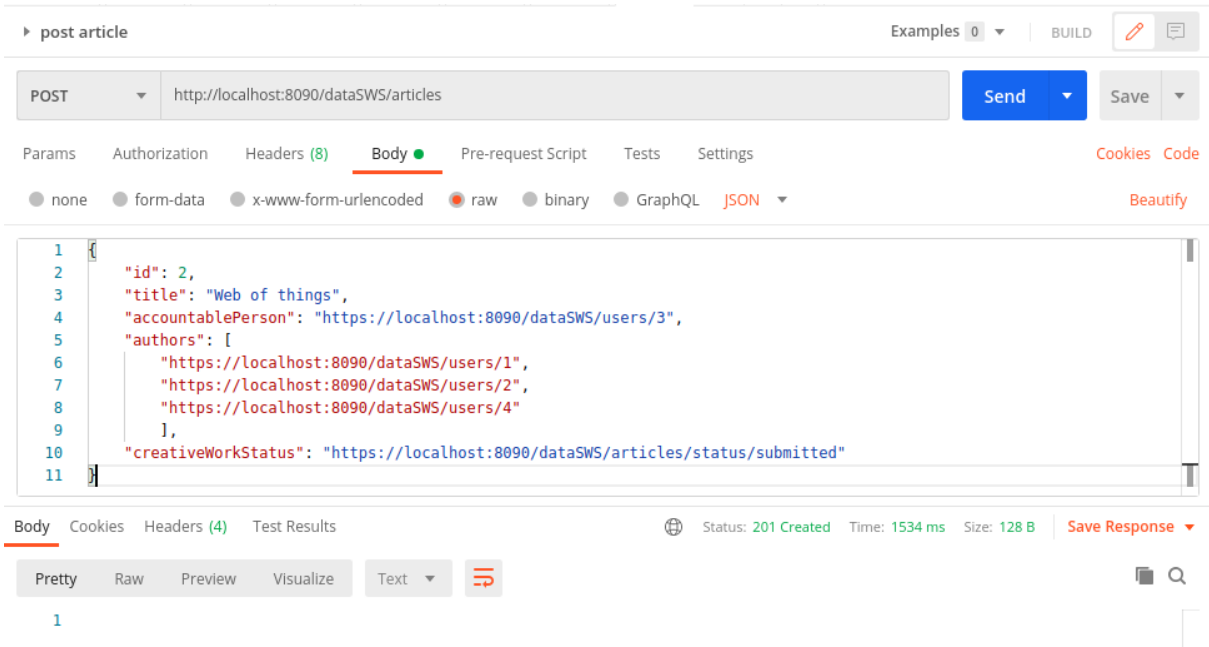


Figura 7: Requisição POST utilizando o *software* *Postman*.

Fonte: Elaboração própria.

Todo o código desenvolvido para este trabalho pode ser encontrado em: <https://github.com/carolviana/datasws>

5. CONCLUSÃO

Este trabalho teve como objetivo a proposição de uma arquitetura para criação de *web service* semântico para execução automática de processos de negócios assistidos por computadores, com dados de negócios armazenados em um banco de dados semântico nativo. O objetivo foi alcançado inicialmente fazendo uma revisão sistemática sobre o tema como forma de adquirir uma base referencial inicial, em seguida a proposição da arquitetura.

Foi feito um estudo de caso realista como forma de corroborar a ideia, o processo foi modelado na notação BPMN 2.0, foram utilizados padrões e tecnologias da *web* semântica, assim como ferramentas e ontologias conceituadas.

A utilização de bancos de dados semânticos nativos contribui para a criação de um banco de dados de negócios totalmente inteligível por máquinas, que acaba contribuindo para posteriores tomadas de decisão.

Como trabalho futuro é proposto o mapeamento da base interna do gerenciador de *workflow*, para dar sentido semântico para os dados de execução das instâncias do processo de negócio, com a intenção de enriquecer os dados e auxiliar o processo de tomada de decisão para aperfeiçoamento do processo de negócio.

REFERÊNCIAS

- ABPMP.** “BPM CBOK: Guia para o Gerenciamento de Processos de Negócio”. Corpo Comum do Conhecimento – ABPMP BPM CBOK V3.0, Association of Business Process Management Professionals, 2013.
- BERNERS-LEE, T.** World Wide Web Consortium, 2009. “Linked Data - Design Issues” Disponível em: <<https://www.w3.org/DesignIssues/LinkedData.html>>. Acessado em: 30 de outubro de 2020.
- BERNERS-LEE, T; HENDLER, J. & LASSILA, E.** “The Semantic Web”, *Scientific American*, vol. 284, nº 5, p. 34–43, 2001.
- CAMUNDA,** 2020. “Modeler”. Disponível em: <<https://camunda.com/download/modeler>>. Acessado em: 31 de Outubro de 2020.
- DANG, J; HEDAYATI, A; HAMPEL, K & TOKLU, C.** “Personalized medical workflow through semantic business process management”, 2009, vol. SAIC, p. 122–127, doi: 10.5220/0002006401220127.
- FIELDING, R & TAYLOR, R.** “Principled design of the modern web architecture”, *Proceedings - International Conference on Software Engineering*, p. 407–416, 2000, doi: 10.1145/337180.337228.
- FONOU-DOMBEU, J & HUISMAN, M.** “Engineering semantic web services for government business processes automation”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9265, p. 40–54, 2015, doi: 10.1007/978-3-319-22389-6_4.
- FREDRICH, T.** RestApiTutorial, [s.d.]. “Learn REST: A RESTful Tutorial”. Disponível em: <<https://www.restapitutorial.com/lessons/whatisrest.html>>. Acessado em: 31 de outubro de 2020.
- GUHA, R; BRICKLEY, D & MACBETH, S.** “Schema.org: Evolution of Structured Data on the Web” ACM Queue, 2015. Volume 13, Issue 9. Disponível em: <<https://queue.acm.org/detail.cfm?id=2857276>>. Acessado em: 31 de Outubro de 2020.
- HEFLIN, J.** World Wide Web Consortium, 2004. “OWL Web Ontology Language Use Cases and Requirements”. Disponível em: <<https://www.w3.org/TR/webont-req/#onto-def>>. Acessado em: 30 de outubro de 2020.
- HEPP, M.; LEYMANN, F.; DOMINGUE, J.; WAHLER, A & FENSEL, D.** “Semantic business process management: A vision towards using semantic web services for business process management”, 2005, vol. 2005, p. 535–540, doi: 10.1109/ICEBE.2005.110.
- JACYNTHO, M & SCHWABE, D.** *Models and Meta Models for Transactions in Web Applications*, vol. 6385. 2010, p. 48.
- JACYNTHO, M.** “Um Modelo de Bloqueio Multigranular Para RDF”, Doutor em Ciências em Informática, Pontifícia Universidade Católica do Rio De Janeiro, Rio de Janeiro, Brazil, 2012.
- KLUZA, K.; KACZOR, K.; NALEPA, G. & ŚLAŻYŃSKI, M.** “Opportunities for Business Process Semantization in Open-source Process Execution Environments”, out. 2015, p. 1307–1314, doi: 10.15439/2015F250.
- LAUSEN, H; POLLERES, A & ROMAN, D.** “Web Service Modeling Ontology (WSMO)”. World Wide Web Consortium, 2005. Disponível em: <<http://www.w3.org/Submission/WSMO>>. Acessado em: 31 de Outubro de 2020.
- MOHER, D; LIBERATI, A; TETZLAFF, J & ALTMAN, D.** “Principais itens para relatar Revisões sistemáticas e Meta-análises: A recomendação PRISMA”, *Epidemiol. Serv. Saúde*, vol. 24, nº 2, p. 335–342, jun. 2015, doi: 10.5123/S1679-49742015000200017.
- SILVA, S. & ROSA, L.** Modelagem dos Processos como Ferramenta de Comunicação - Um Estudo de Caso em Instituição Pública de Ensino. 2017.
- STEIN DANI, V; FREITAS, C & THOM, L.** “Ten Years of Visualization of Business Process Models: A Systematic Literature Review”, *Computer Standards & Interfaces*, abr. 2019, doi: 10.1016/j.csi.2019.04.006.

ZEINER, H; HALB, W; LERNBEISS, H; JANDL, B & DERLER, C. “Making business processes adaptive through semantically enhanced workflow descriptions”, apresentado em ACM International Conference Proceeding Series, 2010, doi: 10.1145/1839707.1839741.