

UNIVERSIDADE ESTADUAL PAULISTA  
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Uso de *Business Process Model* como apoio ao planejamento  
de experimentos em Engenharia de Software

**RELATÓRIO FINAL**

*Programa Institucional de Bolsas de Iniciação Científica  
Conselho Nacional de Desenvolvimento Científico e Tecnológico (PIBIC - CNPq)*

**Discente:** Darlan Murilo Nakamura de Araújo

**Orientação-Discente:** Leandro Ungari Cayres

**Orientador:** Rogério Eduardo Garcia

PRESIDENTE PRUDENTE

2017

## RESUMO

Processos de experimentação são realizados cotidianamente dentro da Engenharia de Software Experimental e seus resultados são armazenados em pacotes de laboratório. Pesquisadores, ao estudarem os experimentos, fazem uso dos pacotes de laboratório para realizar replicações e deparam-se com pacotes deficientes de informações importantes sobre o experimento. Diante de tal cenário, foi elaborada uma ontologia chamada *ExperOntology*, a qual visa definir os principais conceitos para a construção de pacotes de laboratórios. De forma a corroborar com esta questão foi desenvolvida a ferramenta *OntoExpTool*, a qual visa prover um ambiente de auxílio para pesquisadores, desde a fase de definição do experimento até o processo de criar o pacote de laboratório.

O presente projeto visa fornecer uma ferramenta que seja capaz de construir diagramas em Business Process Model apoiando o planejamento de experimentos em Engenharia de Software e que seja integrada com a ferramenta *OntoExpTool*, fornecendo assim um ambiente para criação e compartilhamento de pacotes de laboratório mais completos.

Após um estudo experimental, concluiu-se que o uso da notação *Business Process Model* apresentou-se de fácil elaboração e compreensão e que a ferramenta pode colaborar com a construção de pacotes de laboratório, apoiando assim o planejamento de experimentos em Engenharia de Software.

**Palavras-chave:** *ExperOntology*, Business Process Model, pacote de laboratório.

## ABSTRACT

Experimental processes are performed daily within Experimental Software Engineering and their results are stored in laboratory packages. Researchers, when studying the experiments, make use of the laboratory packages to perform replications and are faced with deficient packages of important information about the experiment. In view of this scenario, an ontology called *ExperOntology* was developed, which aims to define the main concepts for the construction of laboratory packages. In order to corroborate with this question the *OntoExpTool* tool was developed, which aims to provide an aid environment for researchers, from the definition phase of the experiment to the process of creating the laboratory package.

The present project aims to provide a tool that is able to build diagrams in *Business Process Model* supporting the planning of experiments in Software Engineering and that is integrated with the tool *OntoExpTool*, providing an environment for creation and sharing of more complete laboratory packages.

After an experimental study, was concluded that the use of the *Business Process Model* notation was easily developed and understood and that the tool can collaborate with the construction of laboratory packages, supporting the planning of experiments in Software Engineering.

**Keywords:** ExperOntology, Business Process Model, laboratory package.

## SUMÁRIO

1. INTRODUÇÃO.....	5
2. ESTUDO TEÓRICO.....	6
2.1. Engenharia de Software Experimental.....	6
2.2. Pacotes de laboratório.....	8
2.3. ExperOntology.....	9
2.4. Bases de dados não-relacionais.....	11
2.5. XML.....	14
2.6. UML.....	15
2.7. Business Process Modeling Notation.....	16
3. IMPLEMENTAÇÃO.....	19
4. RESULTADOS.....	26
5. CONCLUSÃO.....	28
4. REFERÊNCIAS BIBLIOGRÁFICAS.....	29

## 1. INTRODUÇÃO

Neste projeto de Iniciação Científica tem-se como objetivo geral a avaliação do uso da notação de modelagem de processo de negócio (*BPMN – Business Process Modeling Notation*) para a representação visual de um protocolo de um experimento, possibilitando o planejamento deste experimento através da construção de uma interface capaz de representar a notação. De modo específico, objetiva-se a modificação da camada de apresentação da *OntoExpTool*, tal ferramenta já utilizada no primeiro ano deste projeto de Iniciação Científica.

Pacotes de laboratório são produzidos cotidianamente por meio de processos de experimentação, no entanto, quando outros pesquisadores fazem uso das informações presentes no pacote para realizarem replicações, se deparam com informações ausentes, informações estas que deveriam descrever como o experimento foi realizado e conduzido.

O objetivo da ferramenta é fornecer um ambiente de criação e compartilhamento de pacotes de laboratórios de modo a corrigir o problema em que os pacotes de laboratório são produzidos com deficiências de informações.

Para apresentar as atividades realizadas, o texto está dividido nas seguintes seções: na Seção 2, referente as etapas do estudo teórico, foi composto pela revisão bibliográfica e mapeamento de conceitos em dados não-relacionais, assim como questões relevantes ao empacotamento na Engenharia de Software Experimental, o uso de meta-dados e da utilização da notação BPM; na Seção 3, são apresentados detalhes sobre a implementação da camada de persistência e instanciação dos pacotes de laboratório. Por fim na Seção 4, são apresentados os resultados deste estudo, juntamente com a discussão deste e por fim, na Seção 5 é apresentada a conclusão do trabalho.

## 2. ESTUDO TEÓRICO

### 2.1. Engenharia de Software Experimental

A Engenharia de Software Experimental descreve métodos e técnicas para efetuar estudos empíricos. Segundo Shull et al., a execução de estudos experimentais envolve a necessidade de um grande corpo de conhecimento, resultado do Processo de Experimentação. Portanto, é necessário que estudos sejam armazenados e compartilhados para que o pesquisador possa recuperar esse conhecimento em certo momento para ajudá-lo a tomar decisões baseado em outros estudos.

A experimentação verifica as teorias e métodos que foram propostos e formalizados, de forma a corrigir ou confirmar as teorias e métodos propostas (Basili, Shull e Lanubile, 1999).

A execução de um experimento requer uma sequência de atividades estabelecidas previamente, para que seja possível conduzir um processo experimental de forma precisa e sistemática. Desse modo, Wohlin et al. (2012) propõem a seguinte organização em fases:

- **Definição** — estabelece quais são os problemas tratados;
- **Planejamento** — determina-se o projeto, instrumentação e os possíveis riscos para validação dos resultados;
- **Operação** — coleta-se os dados do experimento;
- **Análise e Interpretação** — os dados são analisados e avaliados;
- **Apresentação e Empacotamento** — os resultados são apresentados e suas informações são armazenadas para uso futuro.

Na Figura 1 é apresentado um gráfico com o processo de experimentação adaptado de Wohlin et al. (2012): nela é possível observar as fases, assim como tarefas inerentes a cada uma.

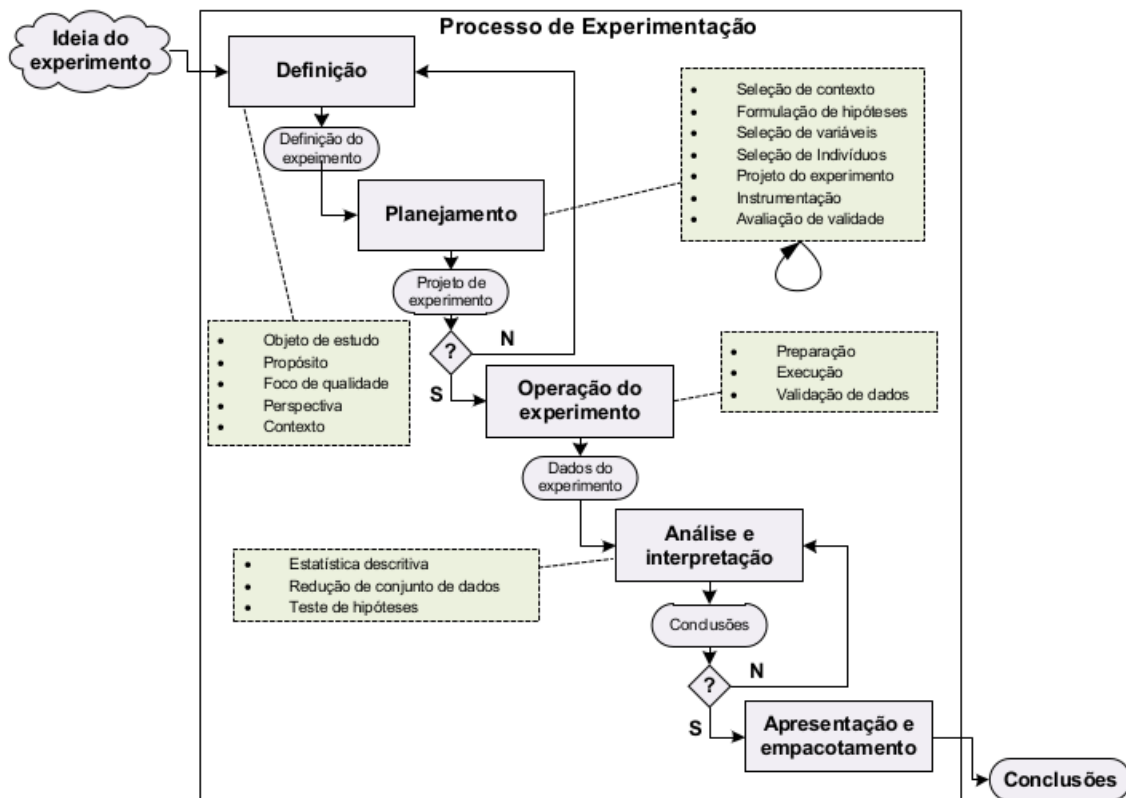


Figura 1 – Processo de Experimentação – adaptado de (WOHLIN et al., 2012).

Como apoio, os estudos experimentais atuam como ferramentas para obtenção dos dados necessários através de todo o processo de desenvolvimento de software, buscando resultados objetivos de forma a alcançar melhorias no processo. Também segundo Wohlin et al. (2012), tais estudos podem ser expressos nas seguintes categorias:

- **Pesquisa de opinião:** neste primeiro tipo de estudo experimental, busca-se obter dados a respeito de uma técnica ou ferramenta específica, seja uma análise qualitativa ou quantitativa, por meio de entrevistas ou questionários aplicados a uma amostra capaz de representar uma determinada população.

- **Estudos de caso:** esta categoria é voltada ao monitoramento de atividades, projetos ou tarefas visando observar um atributo específico, ou estabelecer relações entre alguns destes, a qual se prolonga por um determinado período de tempo. Por se tratar uma atividade observacional, o nível de controle é mais elevado do que o anterior.

- **Experimentos controlados:** por fim, nesta categoria, o estudo experimental tem sua execução manipulada de forma direta e sistemática, a fim de se ter controle sob todos os elementos que o compõem.

O processo experimental proporciona de modo sistemático, disciplinado e controlado a avaliação de processos e atividades humanas (TRAVASSOS et al., 2002).

Para que os pesquisadores adquiram conhecimento adicional a respeito dos conceitos estudados, é necessário que se faça a replicação dos experimentos, de preferência que ocorra em contextos diferentes, pois fatores humanos, socioeconômicos, ambientais e culturais impactam diretamente no resultado do experimento. Caso o experimento tenha sido replicado e sujeito a diferentes fatores e obtenha resultados parecidos, obtém-se uma maior precisão na verificação e na validade de hipóteses (SHULL et al., 2003). Os registros das atividades de um experimento controlado são mantidas no chamado Pacote de Laboratório.

## 2.2. Pacotes de laboratório

Os pacotes de laboratório são estruturas que armazenam os registros das atividades de um experimento controlado. A união desses pacotes de laboratório forma um corpo de conhecimento, necessário para que pesquisadores possam fazer uso de informações de outros experimentos para tomar decisões.

Diversos autores na literatura descrevem dificuldades na revisão dos pacotes de laboratório, bem como na integração do mesmo ao corpo de conhecimento devido a ausência de padronização de pacotes de laboratório.

Desta forma, é imprescindível uma definição de um pacote de laboratório com o uso de uma estrutura específica de simples compreensão, possibilitando inclusive o uso de ontologias para seu desenvolvimento.

Uma ontologia é uma especificação formal explícita de uma conceitualização compartilhada, definindo parte de um domínio por meio de termos relevantes e seus respectivos relacionamentos, cuja estruturação é baseada por determinadas regras.

Diante deste cenário, Garcia et al. propõem o uso de uma ontologia para a Engenharia de Software Experimental, que descreve padrões e conceitos que compõem um pacote de laboratório para experimentos controlados, chamada *ExperOntology*, apresentada na Figura 2.



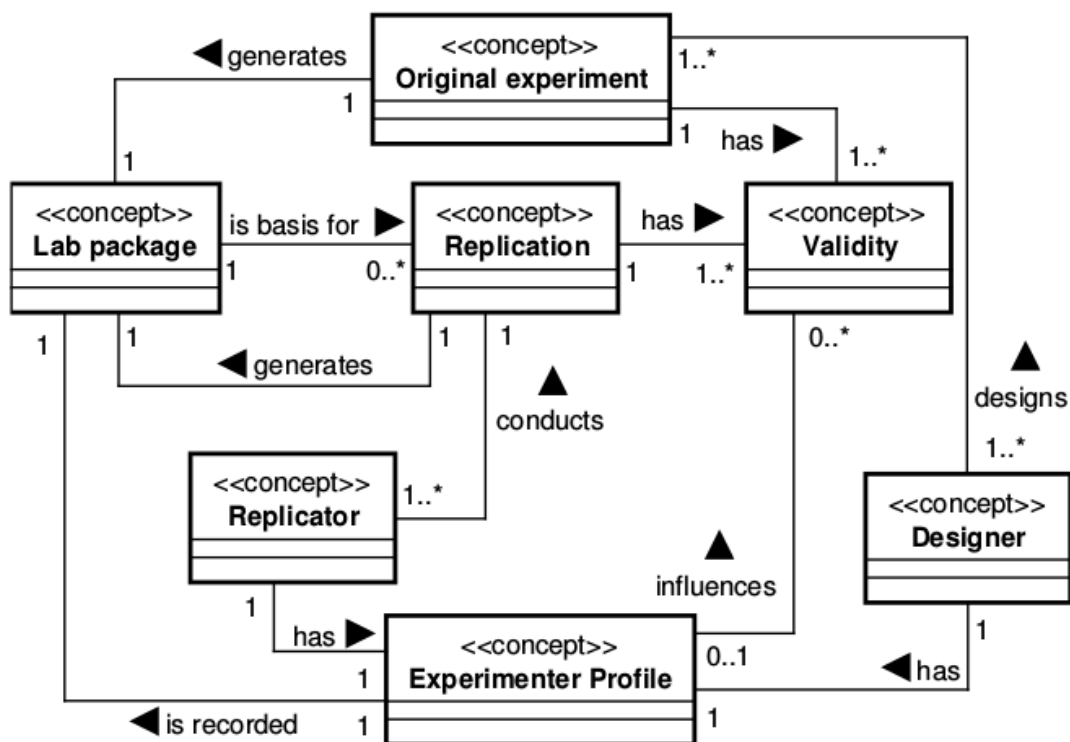


Figura 2 – Ontologia para Experimentos Controlados (GARCIA et al., 2008).

### 2.3. ExperOntology

A *ExperOntology* visa definir os principais conceitos de experimentos controlados desde a fase de definição até a análise de resultados, considerando a evolução do experimento e o uso de pacotes de laboratório para armazenamento dos registros das atividades dos experimentos controlados (GARCIA et al., 2008).

Segundo a *ExperOntology* os conceitos que devem fazer parte de um pacote de laboratório (figura 3). Inicialmente é definido um propósito, um contexto, um objeto de estudo e o foco a serem considerados no estudo controlado; em seguida é elaborada uma hipótese inicial, a qual, juntamente com uma objeto de estudo e um contexto, formam a formalização da hipótese. Em continuação, são definidas as hipóteses nulas e alternativas, assim como as variáveis dependentes e independentes. Durante a fase de planejamento são determinados os objetos de experimentação como as tecnologias e artefatos sob investigação.



Os conceitos apresentados devem pertencer ao Pacote de Laboratório, portanto devem ser armazenados utilizando algum meio para a persistência dos dados.

## 2.4. Bases de dados não-relacionais

Atualmente existem inúmeras maneiras de persistir os dados em um computador, seja utilizando um sistema gerenciador de banco de dados (SGBD), seja armazenando em planilhas e em diferentes estruturas de arquivos. Após a criação dos sistemas gerenciadores de banco de dados relacionais em 1970, estes tem dominado as aplicações, por conferir maneiras simples de realizar consultas aos dados, pela confiabilidade e por possuir restrições de integridade em relação ao armazenamento dos dados.

Porém com o aumento exponencial do volume de dados, os SGBD's relacionais apresentaram uma limitação em relação ao crescimento no volume de dados: as consultas tornaram-se demoradas e a replicação de base de dados inviáveis, uma limitação em relação a escalabilidade, sendo necessário um novo modelo que seja capaz de atender o armazenamento de grande volume de dados.

Assim surgiu o modelo não-relacional como solução, que, diferente do modelo relacional, sugere maior flexibilidade quanto ao armazenamento. Esta é a solução mais adequada para ambientes de Big Data, onde há um número muito grande de dados. Alguns representantes de base de dados não-relacionais são: *Cassandra*, *Dynamo*, *MongoDB* e *BigTable*.

Como banco de dados não-relacionais não possuem regras como no modelo relacional, diversas categorias de sistemas de banco de dados não-relacionais tem sido desenvolvidas, os principais são detalhados a seguir.

- **Orientados a Colunas (*Column*):** é a categoria que mais se aproxima do modelo relacional, porém todas as suas operações são voltadas para as colunas, ao invés das tuplas como no modelo relacional. O seu grande diferencial está na facilidade de inserção de novas colunas, ou seja, atributos com o sistema já em operação, conforme se tornarem necessários, sem apresentar problemas de esquema ou redundância de dados (VAISH, 2013).

- **Armazenamento em Documentos (*Document*):** nesta categoria cada item é armazenado em um novo arquivo, em geral sua organização é feita através de estruturas chamadas coleções as quais são equivalentes as tabelas no modelo relacional, não possuem qualquer esquema, dois objetos de uma mesma coleção podem ter atributos diferentes, o que permite grande flexibilidade, sendo que cada registro é um arquivo. Os principais formatos de arquivos são JSON, XML, BSON e YAML (VAISH, 2013).

- **Armazenamento Chave/Valor (*Key/Value*):** também muito semelhante à categoria anterior (Document), porém seu armazenamento é feito com uso de uma chave, assemelhando-se a uma tabela Hash, ou um array associativo. Sua grande vantagem é busca por chaves (VAISH, 2013).

- **Armazenamento em Grafos (*Graph*):** esta categoria tem foco nos relacionamentos entre as entidades, que no caso são os nós do grafo, permitindo o uso de múltiplas ligações entre os nós para demonstrar características em comum. Este modelo pode ser ideal para redes sociais. Uma prática usual é a mescla entre o banco de dados orientado a documentos e o orientado a grafos, tornando não obrigatória a presença de relacionamentos (VAISH, 2013).

Tabela 1 – Banco de Dados Não-Relacional e sua tecnologia de armazenamento.

Document	Key-Value	XML	Column	Graph
MongoDB	Redis	BaseX	BigTable	Neo4J
CouchDB	Membase	eXist	Hadoop/HBase	FlockDB
RavenDB	Voldemort	-	Cassandra	InfiniteGraph

Na Tabela 1 há uma breve lista de tecnologias de banco de dados não-relacionais e a respectiva classificação, segundo as classes de armazenamento apresentadas. Antes de se estabelecer uma comparação entre os modelos, é importante definir conceitos como que deem compor essa comparação, são eles:

- **Escalonamento:** este conceito, no contexto de banco de dados, consiste na capacidade em que uma base de dados tem de destruição tanto horizontal (scale out), neste caso a estruturação do sistema é dividida em várias máquinas tanto por parte do banco de dados; quanto vertical (scale up), no qual são realizadas melhorias de hardware em relação à processamento e armazenamento (TOTH, 2011).
- **Consistência:** refere-se à capacidade de manter os dados de forma íntegra, de modo que evite quaisquer problemas no banco de dados possam modificar ou corromper os dados armazenados (BRITO, 2010).
- **Disponibilidade:** este quesito se refere a capacidade de acesso do usuário a referida informação, tanto em quesito de velocidade quanto de solicitação (BRITO, 2010).

Tabela 2 – Análise Comparativa do Modelo Relacional e Não-Relacional (BRITO, 2010).

	Relacional	Não-Relacional
<b>Escalonamento</b>	Possível, porém complexo devido à natureza estruturada do modelo, a adição de forma dinâmica e transparente de novos nós no modelo não é realizada de modo natural.	Uma das principais vantagens desse modelo, por não possuir nenhum tipo de esquema predeterminado, o modelo possui maior flexibilidade o que favorece a inclusão transparente de outros elementos.
<b>Consistência</b>	Ponto mais forte do modelo relacional. As regras de consistência presentes propiciam uma maior grau de rigor quanto à consistência das informações.	Realizada de modo eventual no modelo: só garante que, se nenhuma atualização for realizada sobre o item de dados, todos os acessos a esse item devolvem o último valor atualizado.
<b>Disponibilidade</b>	Dada a dificuldade de se conseguir trabalhar de forma eficiente com a distribuição dos dados, esse modelo pode não suportar a demanda muito grande de informações do banco.	Outro fator fundamental do sucesso desse modelo. O alto grau de distribuição dos dados propicia que um maior número de solicitações aos dados seja atendida por parte do sistema e que o sistema fique menos tempo não-disponível.

Na Tabela 2 é apresentado um comparativo com o modelo relacional e o modelo não-relacional, analisando os quesitos de consistência, escalonamento e disponibilidade (BRITO, 2010).

Portanto, para a persistência da camada de interface será utilizada um modelo não-relacional orientado a documentos, utilizando documentos no formato XML (eXtensive Markup Language), de modo a facilitar transferência de protocolos experimentais expressos em BPMN entre experimentador e replicador.

## **2.5. XML**

O XML (exTensive Markup Language) é uma arquitetura que não possui elementos e marcas predefinidas. Não especifica como os autores vão utilizar metadados, sendo que existe total liberdade para utilizar qualquer método disponível, desde simples atributos, até a implementação de padrões mais complexos. Esta característica confere à linguagem XML “habilidades” semânticas, que possibilitam melhorias significativas em processos de recuperação e disseminação da informação (ALMEIDA, Maurício Barcellos, 2002).

De acordo com o W3 Consortium, entre os objetivos estabelecidos na especificação da linguagem XML, estão as seguintes características: ser diretamente utilizável na Internet; ser legível por humanos; facilitar às pessoas o processamento de dados pelo uso de softwares de baixo custo; facilitar a utilização de metadados que auxiliam na busca de informações; aproximar “produtores” e “consumidores” de informação (ALMEIDA, Maurício Barcellos, 2002).

A sintaxe do XML é adequada para descrever dados semi-estruturados, apesar da possibilidade de existência de ambigüidades introduzidas pela presença de atributos.

Na Figura 4 será apresentado abaixo a estrutura de um documento XML em que estão representado os dados de uma pessoa com o nome João, com nascida em 05/10/1993 com o e-mail joao@hotmail.com:

```
<peessoa>
  <nome>João</nome>
  <datanascimento>1993-10-05</datanascimento>
  <email>joao@hotmail.com</email>
</peessoa>
```

Figura 4 – Exemplo de documento XML.

## 2.6. UML

A UML (Unified Modeling Language) surgiu na metade da década de 1990, quando Grady Booch, Ivar Jacobson e James Rumbaugh, resolveram juntar três métodos que eram complementares na época, cada um com seus pontos fortes e fracos. Os métodos eram: Booch era expressivo principalmente durante as fases de projeto e construção de sistemas; o OOSE(Object-Oriented Software Engineering) fornecia excelente suporte para os casos como uma maneira de controlar a captura de requisitos, a análise e o projeto em alto nível; o OMT(Object Modeling Technique) era mais útil para análise e sistemas de informações com uso intensivo de dados. A unificação das linguagens teve como intuito fortalecer o ambiente estável para o mercado orientado a objetos, permitindo que os projetos tivessem como base uma linguagem madura de modelagem e que os produtores de ferramentas fornecessem recursos mais úteis (BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar., 2005).

A notação UML provê regras para visualizar, especificar, construir e documentar os sistemas de software ou não-software, de modo a facilitar a visualização e o entendimento entre todos os envolvidos com o projeto.

Atualmente a notação UML está na versão 2.5 e provê elementos de forma a facilitar a compreensão pré-implementação, e é adequada para a modelagem de sistemas, permitindo que os desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados. Junto com uma notação gráfica, a UML especifica significados.

## 2.7 Business Process Modeling Notation

A notação BPMN, pertencente a UML, é utilizada para a elaboração e representação de modelos de processos de negócio, útil para descrever a lógica de um processo por meio de diagramas. Assim, é possível visualizar melhor todas as suas etapas e analisá-las sem qualquer dificuldade, fornecendo uma notação intuitiva, de forma que usuários técnicos e usuários de negócios possam entender e representar semânticas de processos complexos.

A especificação da notação BPMN foi elaborada e lançada em 2004 pelo Instituto de Gerenciamento de Processos de Negócio ou BPMI (Business Process Management Institute), que realizou fusão ao Consórcio OMG. Posteriormente, a notação BPMN foi adotada pelo Consórcio OMG como o padrão para modelagem de processos.

Em sua primeira versão, a notação BPM discrimina seu conjunto de elementos nas seguintes categorias: objetos de fluxo, dados, conectores, *swimlanes* (raias de piscina) e artefatos. Especificamente, os objetos de fluxo definem o comportamento do processo, definindo as ações e as decisões, cujos principais elementos são os eventos, atividades e *gateways* (portões). Em relação aos objetos de dados, estes se assemelham muito a ideia de dados utilizadas em software, sob uma perspectiva de elementos de entrada, saída e armazenamento, desta forma, seus principais representantes são os objetos de dados, entrada de dados, saída de dados e armazenamento de dados. Por sua vez, os conectores definem como as informações transitam entre os elementos da diagramação, temos exemplos de fluxo de sequência, mensagens de fluxo, associações e associações de dados. Em relação aos *swimlanes*, estes elementos definem agrupamentos ou conjuntos dos próprios elementos, permitindo a ideia de sub-processos, tendo como representantes as piscinas e as raia. Por fim, em relação aos artefatos, tais elementos são usados para prover informações adicionais de forma a facilitar o entendimento do processo.

A Figura 5 apresenta, como exemplo, um simples diagrama de processo de negócio, através do qual é possível identificar a interação entre dois atores através da execução de atividades e trocas de mensagens, assim como, o uso de recursos como artefatos e repositórios de dados.



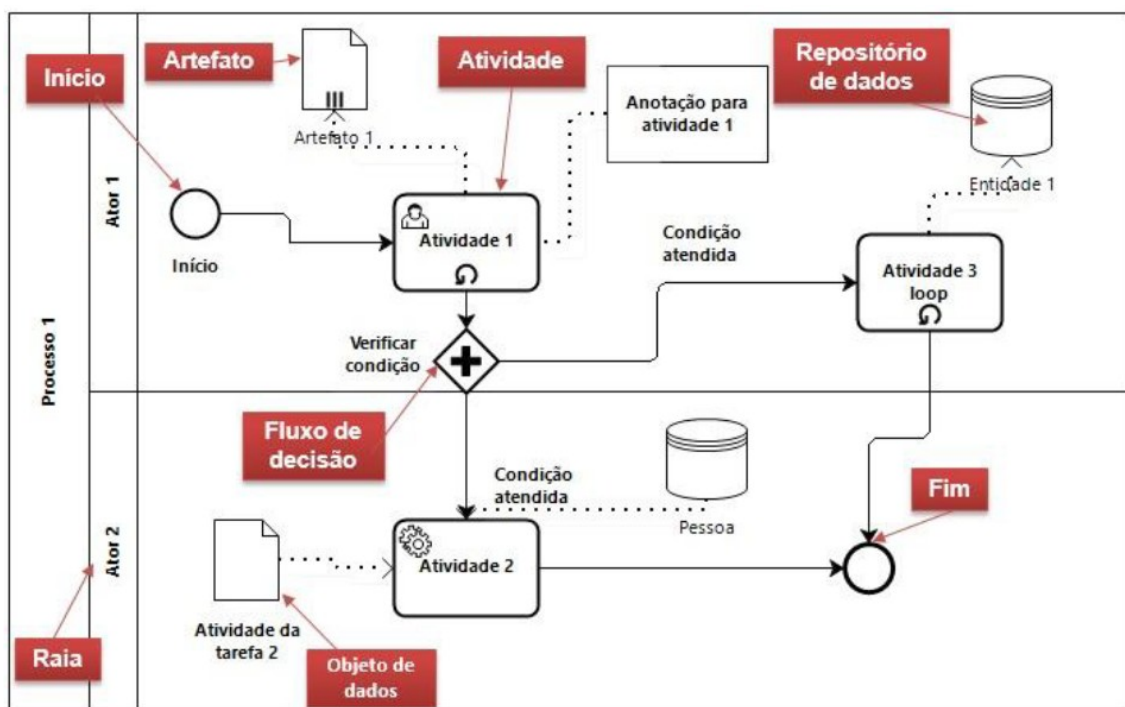


Figura 5 – Exemplo de diagrama BPMN – adaptado de (WESKE, 2012).

A especificação completa da atual versão define atributos que são agrupados em quatro categorias básicas de elementos: objetos de fluxo (Flow Objects), objetos de conexão (Connecting Objects), vias (Swimlanes) e artefatos (Artifacts).

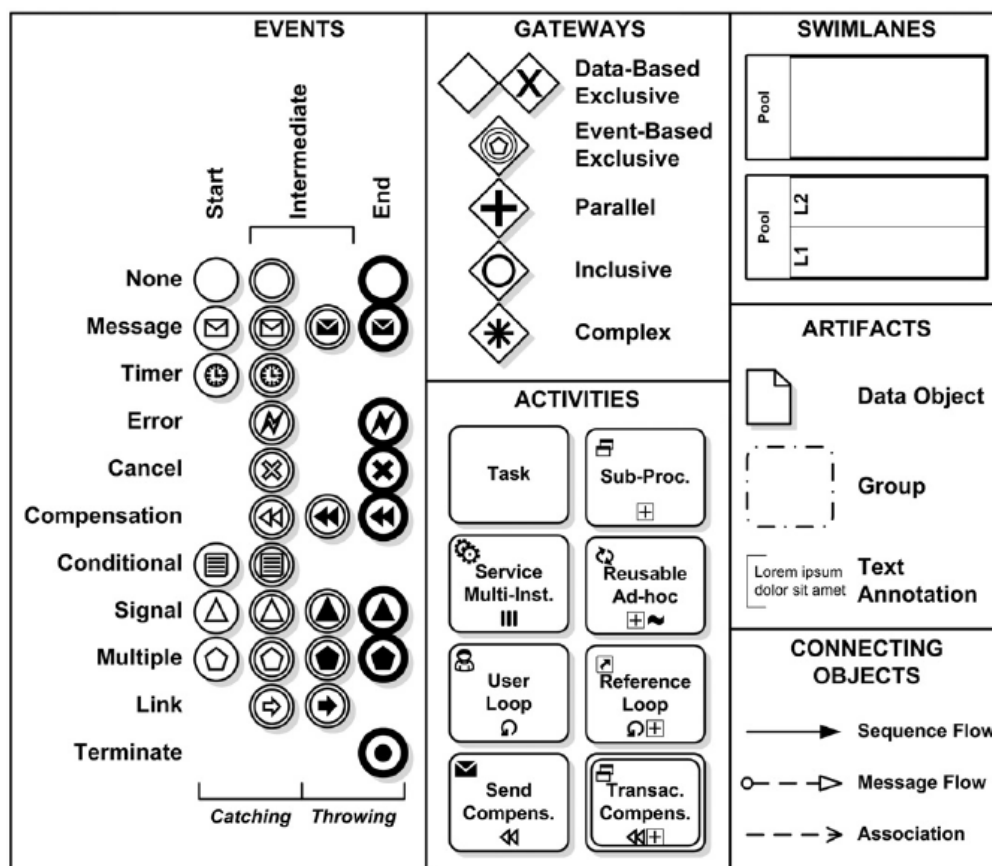


Figura 6 – Conjunto de elementos que compõem a versão BPMN 2.0  
(COMMITTEE et al., 2016).

Como a notação BPM oferece uma maneira de descrever processos de maneira simples, podemos confirmar que utilizar a notação contribui diretamente para registro do processo de execução de experimentos, fornecendo assim um padrão para a construção de pacotes de laboratório.

### 3. IMPLEMENTAÇÃO

A primeira parte do projeto foi destinada a parte básica da notação BPM, onde há um conjunto reduzido de classes de elementos, cada qual com um subclasse, como por exemplo, objetos de fluxo podem ser eventos, atividades ou portões (gateways). Utilizando o conceito de Polimorfismo de Orientação a Objetos, podemos definir que cada elemento que possui um subtipo que consiste uma classe abstrata e seus subtipos a herdarão.

Portanto, ao criar as classes, é necessário uma forma de persistir os dados, armazenar os dados do usuário que utilizar a ferramenta. A serialização é uma técnica que consiste em converter um objeto para uma sequência de bytes e gravar essa sequência de bytes em um arquivo.

Para a persistência dos dados no projeto, será utilizado a biblioteca XStream, que serializa os objetos em Java para XML e desserializa (converte de XML para objeto Java).

Relacionando o modelo relacional com o modelo orientado a documentos, temos que cada registro corresponde a um arquivo (*.xml*) onde cada tabela corresponde a uma coleção, e que a base de dados é representada como uma pasta do sistema de arquivos do sistema operacional.

A persistência dos objetos em XML tem como objetivo que o usuário possa salvar o que foi produzido através da ferramenta, de forma a continuar a elaborar o mesmo projeto em um outro momento.

As atividades descritas nesta seção de implementação utilizaram o ambiente de desenvolvimento integrado NetBeans, sendo implementadas na linguagem de programação Java.

Durante a implementação, foi elaborado um Diagrama de Classes, definindo portanto todas as classes pertencentes ao projeto e todas as relações entre as classes, de modo a permitir que ao analisar o Diagrama de Classes, é possível entender os relacionamentos entre as classes e portanto a estrutura geral do projeto. O Diagrama de Classes foi desenvolvido utilizando a ferramenta online Creately, disponível em [creately.com](https://creately.com).

A Figura 7 representa um fragmento do Diagrama de Classes em que se pode observar o relacionamento das classes que herdam a classe mãe Task, onde podemos visualizar também os atributos pertencentes a cada uma das classes.

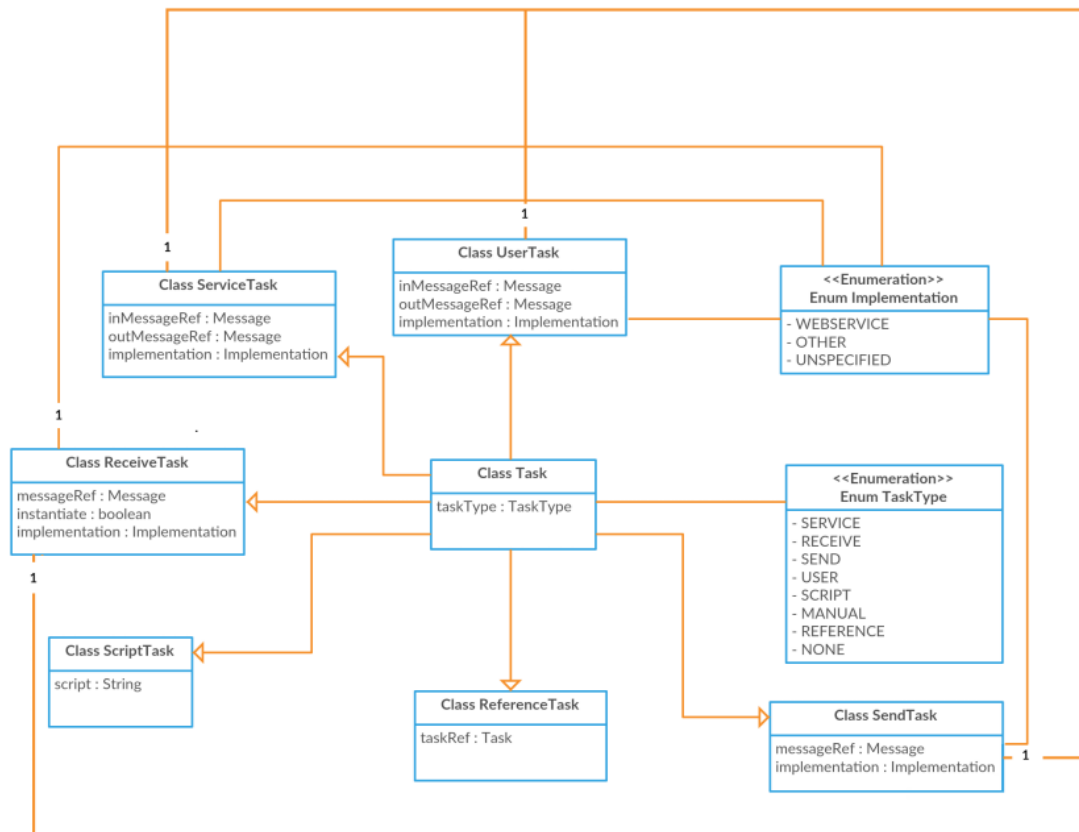


Figura 7 – Fragmento do Diagrama de Classes referente às classes Task  
(Fonte: Próprio autor).

Cada objeto instanciado é referente a um único arquivo, cuja correspondência pode ser verificada na Figura 7 e Figura 8, através da comparação de uma classe na linguagem de programação Java e seu arquivo XML correspondente:

```

1  public class BusinessProcessDiagram {
2  private String id ;
3  private String name ;
4  private String version ;
5  private String author ;
6  private String language ;
7  private String queryLanguage ;
8  private Date creationDate ;
9  private Date modificationDate ;
10 private ArrayList < Pool > pools ;
11 private String documentation ;
12 }
13 .

```

Figura 8 - Estrutura de Dados inicial da classe diagrama BPM (Fonte: Próprio autor).

```

1  <BusinessProcessDiagram>
2  <id>1</id>
3  <name>Selecao dos Grupos</name>
4  <author>Leandro Ungari</author>
5  <language>PT-BR</language>
6  <queryLanguage>BPMN </queryLanguage>
7  <creationDate > 01/05/2017 </creationDate>
8  <modificationDate> 10/05/2017 </modificationDate>
9  <pools>
10 <pool id ="1"> ... </pool>
11 <pool id ="2"> ... </pool>
12 <pool id ="3"> ... </pool>
13 ...
14 <pool id =" n "> ... </pool>
15 </pools>
16 <documentation> Elemento formalizado pela documentacao BPMN 2.0 ...
17 </documentation>
18 </BusinessProcessDiagram>

```

Figura 9 - Estrutura do XML da classe diagrama BPM (Fonte: Próprio autor).

Foi adotado o uso de anotações disponibilizadas pela documentação da biblioteca Xstream para facilitar a leitura do XML, por meio das anotações, podemos renomear classes ou atributos, de modo a produzir um XML mais legível, como por exemplo, isVisible é um atributo booleano, portanto seria impresso no XML como “true” ou “false”, mas por meio das notações, é possível renomeá-los para que seja “yes” ou “no”, tornando a leitura do XML muito mais agradável. Um exemplo deste recurso pode ser observado nas figuras 9 e 10 referente à Classe Element em Java e seu respectivo arquivo XML:

```
1 public class Element{
2     private Point pointer;
3     private boolean isVisible;
4 }
```

Figura 10 - Estrutura de Dados inicial da classe Element (Fonte: Próprio autor).

```
1 <Element>|
2     <pointer>
3         <x>20.0</x>
4         <y>30.0</y>
5     </pointer>
6     <isVisible>yes</isVisible>
7 </Element>
```

Figura 11 - Estrutura do XML da classe Element (Fonte: Próprio autor).

Neste exemplo, foi renomeado a classe para “Element” pois, caso não houvesse a anotação, o XML produzido seria <BPMNelement.Element> seguindo o padrão de <nome\_do\_pacote.nome\_da\_classe> e não <Element>. Entende-se que renomeando para Element, temos um XML muito mais legível e limpo, facilitando as aplicações com o mesmo.

Em relação a ferramenta desenvolvida para a elaboração dos protocolos de experimentação, permitindo a construção de diagramas utilizando a notação BPMN. Esta é composta por dois painéis laterais, um destinado a inserção de elementos que compõem a diagramação. Enquanto o segundo painel é responsável pela edição dos elementos já adicionadas ao esquema do diagrama, manipulando suas respectivas informações.

A Figura 12 apresenta uma visualização geral da interface implementada.

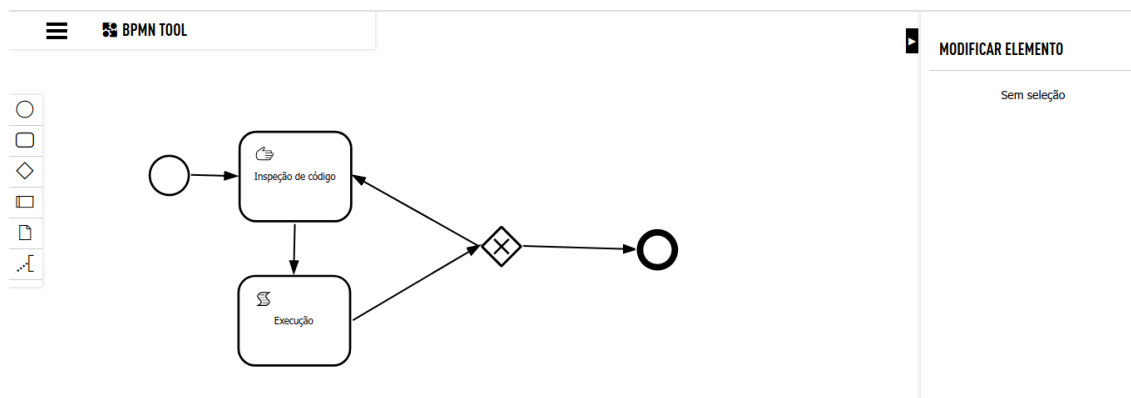


Figura 12 - Visão geral da interface da ferramenta de modelagem.

As Figuras 13, 14 e 15 apresentam a estruturação de menus da ferramenta responsáveis pela inserção de elementos pertencentes a notação processos de negócio, os elementos são eventos, portões e tarefas, respectivamente.

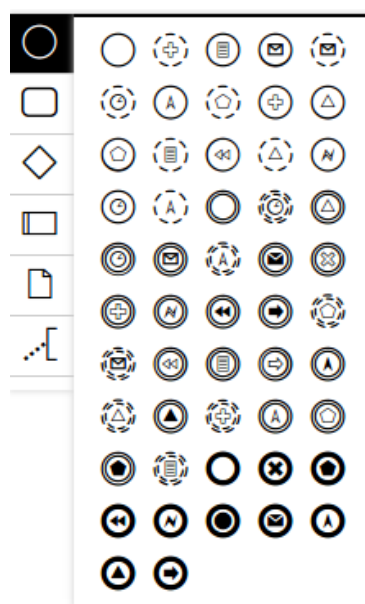


Figura 13 - Elementos da notação pertencentes à categoria de eventos.

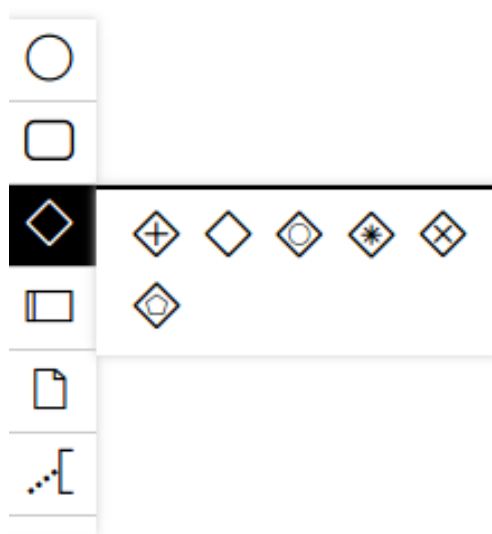


Figura 14 - Elementos da notação pertencentes à categoria de portões.



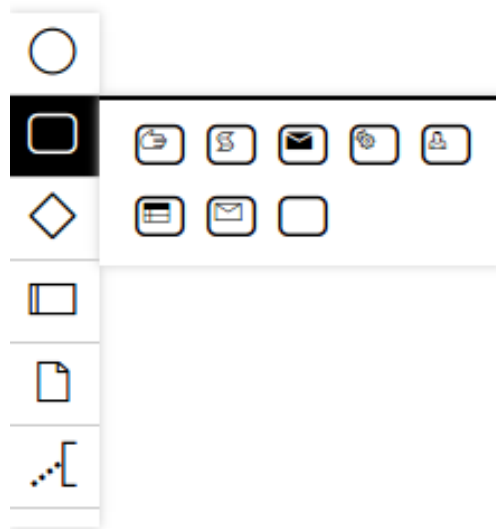


Figura 15 - Elementos da notação pertencentes à categoria de tarefas.

É importante salientar que, apesar de a ferramenta de modelagem de processo de negócio continuar em desenvolvimento, foi possível a aquisição de resultados significativos perante aos objetivos propostos.

#### 4. RESULTADOS

A utilização de bases de dados não-relacionais na construção de pacotes de laboratório, especialmente utilizando o modelo adotado, no caso orientação a documentos, provém mais fácil acesso ao conteúdo destes pacotes, devido a utilização de um formato de arquivo não-proprietário e textual, como XML e JSON, o qual pode ser interpretado sem o uso de uma ferramenta.

Optar por utilizar arquivos XML para fazer a persistência dos dados, provê também uma maior portabilidade entre sistemas, tornando a base de dados dos pacotes de laboratório independente de qualquer sistema gerenciador de banco de dados. Assim, devido a simplicidade da organização destes pacotes de laboratório, diversos recursos presentes em tais sistemas gerenciadores tornam-se desnecessários.

Em relação a utilização da notação de modelagem de processo de negócio, esta foi aplicada a um processo de experimentação referente a parte do projeto de mestrado intitulado “*ModelUIVIZ* - Uma proposta para representação de modelos de interface do usuário utilizando Visualização de Informação”, sob responsabilidade de Livia Cristina Gabos Martins, participante do Laboratório de Pesquisa em Engenharia de Software Aplicada – LaPESA.

Na parte inicial do processo experimentação, um grupo de participantes passaria sob uma separação sem critério em dois grupos distintos. Aplicado a modelo de processo de negócio, este é apresentado na Figura 16.

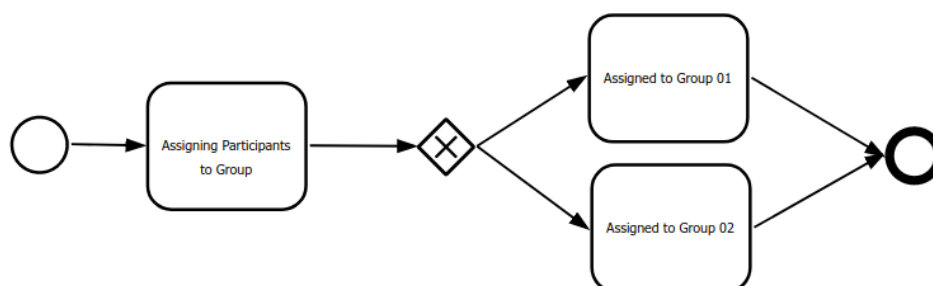


Figura 16 - Seleção de grupos.

Em seguida, finalizada a seleção o grupo A realizaria uma análise sob uma determinada página de um site e elaboraria um conjunto de diagramas referentes a este. Esta etapa pode ser modelada através do diagrama apresentado na Figura 17.

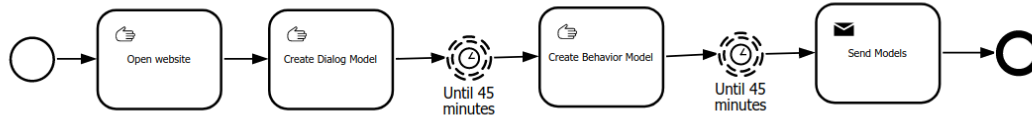


Figura 17 - Análise e diagramação utilizando o site.

Após a execução deste longo processo, o mesmo grupo realizaria um conjunto de atividades utilizando a ferramenta referente ao dado projeto.

Figura 18 - Análise e diagramação utilizando a ferramenta apresentada.



Paralelamente a este conjunto de atividades, os participantes selecionados no grupo B também executaram as mesmas atividades, porém em ordem de processos invertida.

Perante o estudo experimental apresentado acima, o uso da notação de processo de negócio apresentou-se de fácil elaboração e compreensão tanto para a construção quanto análise de diagramas.

A escolha da notação BPM apresentou-se correto pois consiste em uma notação padrão dentre o conjunto de notação do padrão UML e também, plenamente conhecida no âmbito da Engenharia de Software e também muito utilizado em ambientes corporativos para a elaboração de projetos.

Além disso, o uso desta notação permite um simples processo de persistência para o formato XML, e desta forma, pode ser adicionalmente ao corpo estrutural do pacote de laboratório referente ao experimento, promovendo uma maior completude de informações no processo de instanciação de pacotes e também corroborando com o processo replicatório de experimentos.

## **5. CONCLUSÃO**

Com o desenvolvimento deste projeto de Iniciação Científica foi possível perceber que a utilização de uma base de dados não-relacional é benéfica não somente em ambientes com grande volumes de dados, mas também para pequenas aplicações, onde é possível acessar o conteúdo das coleções e também alterar a estrutura organizacional de modo mais fácil, sendo desnecessários os recursos presentes nos sistemas de gerenciamento de banco de dados relacionais. Uma outra vantagem é que o próprio usuário poderá fazer o armazenamento de seu projeto e assim, garantir de que nenhum outro usuário terá acesso ao mesmo.

Por meio da realização do trabalho, foi concluído que uma ferramenta para facilitar a construção de diagramas em Business Process Model pode fortalecer o planejamento em experimentos em Engenharia de Software de modo a prover ao usuário um ambiente iterativo para a construção de tal e prover meios de produção e compartilhamento de pacotes de laboratórios com informações suficientes para se fazer replicações, também fornecendo de certa forma um padrão para os pacotes de laboratório.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

BANKER, K. MongoDB in action. [S.l.]: Manning Publications Co., 2011.

BASILI, V. R.; SHULL, F.; LANUBILE, F. Building knowledge through families of experiments. IEEE Transactions on Software Engineering, v. 25, n. 4, p. 456–473, 1999.

BRAZIL, A. BPM CBOK V3. 0: Guia para o Gerenciamento de Processos de Negócio-Corpo Comum de Conhecimento, 3 a edição . ABPMP Brazil. 2011.

BRITO, R. W. Bancos de dados nosql x sgbds relacionais: análise comparativa. Faculdade Farias Brito e Universidade de Fortaleza, 2010.

CASSANDRA, A. Apache cassandra. Website. Available online at <http://planetcassandra.org/what-is-apache-cassandra>, p. 13, 2014.

CHANG, F. et al. Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems (TOCS), ACM, v. 26, n. 2, p. 4, 2008.

COMMITTEE, O. M. G. B. T. et al. Business Process Model and Notation, version 2.0, 2010. 2016.

CORREIA, A.; ABREU, F. B. e. Enhancing the correctness of bpmn models. In: Improving Organizational Effectiveness with Enterprise Information Systems. [S.l.]: IGI Global, 2015. p. 241–261.

GARCIA, R. E. et al. An ontology for controlled experiments on software engineering. In: Int. Conf. Software Engineering and Knowledge Engineering. [S.l.]: Knowledge Systems Institute Graduate School, 2008. p. 685–690.

KITCHENHAM, B. The role of replications in empirical software engineering—a word of warning. Empirical Software Engineering, Kluwer Academic Publishers, Hingham, MA, USA, v. 13, n. 2, p. 219–221, 2008. ISSN 1382-3256.

MENDONÇA, M. G. de et al. A framework for software engineering experimental replications. In: ICECCS. [S.l.: s.n.], 2008. p. 203–212. 5

MILLER, J. Replicating software engineering experiments: A poisoned chalice or the holy grail. Inf. Softw. Technol., Butterworth-Heinemann, Newton, MA, USA, v. 47, n. 4, p. 233–244, mar. 2005. ISSN 0950-5849. Disponível em: <<http://dx.doi.org/10.1016/j.infsof.2004.08.005>>. 3

MYSQL, A. MySQL. 2001. 9

Pucci Neto, J. et al. Exptool: a tool to conduct, package and replicate controlled experiments in software engineering. In: Proc. 12th. Int. Conf. on Software Engineering Research and Practice. [S.l.: s.n.], 2014. Disponível em <http://worldcomp-proceedings.com/proc/p2014/SER7199.pdf>. 9

ALMEIDA, Maurício Barcellos. Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares. Ciência da informação, v. 31, n. 2, p. 5-13, 2002.

Disponível em <<http://www.scielo.br/pdf/ci/v31n2/12903>>. Acesso em: 10 jun. 2017.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: Guia do Usuário. Tradução de Fábio Freitas da Silva e Cristina de Amorim Machado. 2005. Disponível em: <[https://books.google.com.br/books?hl=pt-BR&lr=&id=ddWqxcDKGF8C&oi=fnd&pg=PR13&dq=uml&ots=fevHkgfJMJ&sig=rV\\_0JvfS8qtOs1P7huHLyR8etto#v=onepage&q=uml&f=false](https://books.google.com.br/books?hl=pt-BR&lr=&id=ddWqxcDKGF8C&oi=fnd&pg=PR13&dq=uml&ots=fevHkgfJMJ&sig=rV_0JvfS8qtOs1P7huHLyR8etto#v=onepage&q=uml&f=false)>. Acesso em: 10 jun. 2017.