

Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP)

Uso de *Business Process Model* como apoio à condução e à replicação de experimentos controlados em Engenharia de Software

RELATÓRIO CIENTÍFICO PARCIAL

Bolsista: Leandro Ungari Cayres

Orientador: Prof. Dr. Rogério Eduardo Garcia

Processo: 2015/18238-9

Período do relatório: 01/12/2016 a 31/05/2017

Vigência: 01/12/2016 a 30/11/2017

Presidente Prudente

Maio de 2017

Sumário

	Sumário	i
	Lista de ilustrações	ii
	Lista de tabelas	iii
1	INTRODUÇÃO	1
2	PROCESSO EXPERIMENTAL: SEUS DADOS E METADADOS . .	3
2.1	Pacotes de Laboratório	5
2.2	Bancos de Dados Não-Relacionais	7
3	MODELAGEM DE PROCESSO DE NEGÓCIO	11
3.1	<i>Business Process Modeling and Notation</i>	11
3.2	Protocolos de experimentação usando BPM	13
4	PROTÓTIPO DA INTERFACE DE MODELAGEM	17
5	PLANO INICIAL E ATIVIDADES REALIZADAS	21
5.1	Cronograma Inicial	21
5.2	Atividades Realizadas	21
6	CONSIDERAÇÕES FINAIS	23
	REFERÊNCIAS BIBLIOGRÁFICAS	25

Lista de ilustrações

Figura 1 – Processo de Experimentação – adaptado de (WOHLIN et al., 2012).	4
Figura 2 – Ontologia para Experimentos Controlados (GARCIA et al., 2008).	6
Figura 3 – Ontologia para Pacotes de Laboratório (GARCIA et al., 2008).	7
Figura 4 – Exemplo de diagrama BPMN – adaptado de (WESKE, 2012).	12
Figura 5 – Conjunto de elementos que compõem a versão BPMN 2.0 (COMMIT- TEE et al., 2016).	12
Figura 6 – Processo de atribuição de participantes aos grupos.	13
Figura 7 – Diagrama BPM referente às atividades dos participantes do primeiro grupo.	14
Figura 8 – Diagrama BPM referente às atividades dos participantes do segundo grupo.	15
Figura 9 – Interface de modelagem da notação.	17
Figura 10 – Menu de opções da ferramenta.	18
Figura 11 – Painel de elementos da categoria de eventos.	18
Figura 12 – Painel de elementos	19
Figura 13 – Exemplo de elemento <i>MessageTask</i>	19

Lista de tabelas

Tabela 1 – Banco de Dados Não-Relacional e sua tecnologia de armazenamento. . .	9
Tabela 2 – Análise Comparativa do Modelo Relacional e Não-Relacional (BRITO, 2010).	10
Tabela 3 – Cronograma de Atividades	21

1 Introdução

Este relatório tem como objetivo apresentar as atividades desenvolvidas pelo bolsista Leandro Ungari Cayres referente ao projeto registrado junto à FAPESP com número 2016/17477-2, sob orientação do Prof. Dr. Rogério E. Garcia, período compreendido entre dezembro de 2016 a maio de 2017.

O objetivo geral deste projeto consiste em prover uma interface capaz de apresentar visualmente o protocolo de um experimento, utilizando a notação BPM (*Business Process Modeling*). Ou seja, deve-se: (1) prover ao experimentador a possibilidade de planejar seu experimento utilizando a notação BPM e; (2) prover ao replicador a possibilidade de visualizar o protocolo contido no PL, também utilizando a notação BPM.

Como objetivo específico, considera-se a modificação da camada de apresentação (interface) da ferramenta *OntoExpTool*, incorporando o modelo gráfico BPM à ferramenta, assim como as adequações necessárias na camada de controle. É importante ressaltar que objetiva-se constituir um sistema de software que permita a concepção e a troca de pacotes de laboratórios, para apoiar a condução de experimentos controlados.

Como questões de investigação (em nível de Iniciação Científica), tem-se: o uso de BPM contribui para a definição do protocolo de experimentos? Quais recursos devem ser utilizados para o uso da tecnologia? Como suplantam limitações para a tecnologia a ser utilizada?

Para apresentar as atividades desenvolvidas até o momento, este relatório encontra-se dividido em 5 capítulos, além desta introdução:

- No capítulo 2 é apresentada uma breve revisão sobre Engenharia de Software Experimental, com foco em experimentos controlados, para contextualização do projeto de pesquisa.
- No capítulo 3 é descrito um breve histórico da notação BPM, assim como, apresentando os elementos que compõem esta notação.
- No capítulo 4 é apresentado o andamento das atividades de implementação da interface de modelagem.
- No capítulo 5 é apresentado o plano de atividades inicial para o projeto. É, também, apresentado o que foi cumprido até o presente momento, bem como as atividades que serão efetuadas a partir desta etapa.
- Por fim, no Capítulo 6 são apresentadas as considerações finais do presente relatório.

2 Processo Experimental: seus dados e metadados

A Engenharia de Software Experimental busca medir e avaliar modelos e tecnologias em contextos práticos, objetivando a obtenção de um corpo de conhecimento. Porém, resultados oriundos de um único experimento não podem estabelecer tal corpo de conhecimento, de modo a determinar fatos sobre um dado fenômeno devido às mais diversas variações que podem ser introduzidas no processo experimental, tais como influências culturais ([BASILI; SHULL; LANUBILE, 1999](#); [MILLER, 2005](#); [SHULL et al., 2003](#)). Perante a isto, se faz necessário que dados sobre estudos independentes sejam armazenados e compartilhados.

O ato de experimentação consiste no centro do processo científico, a qual representa uma atividade na área de pesquisa científica e se utiliza de métodos de investigação experimental para a condução de experimentos ([TRAVASSOS; GUROV; AMARAL, 2002](#)).

Segundo [Basili, Shull e Lanubile \(1999\)](#), a experimentação ajuda a determinar a eficácia de métodos e de teorias propostas. Somente experimentos verificam as teorias, podem explorar os fatores críticos, e dar luz ao fenômeno novo para que as teorias possam ser formuladas e corrigidas.

A execução de um experimento requer uma sequência de atividades estabelecidas previamente, que por meio das quais 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.

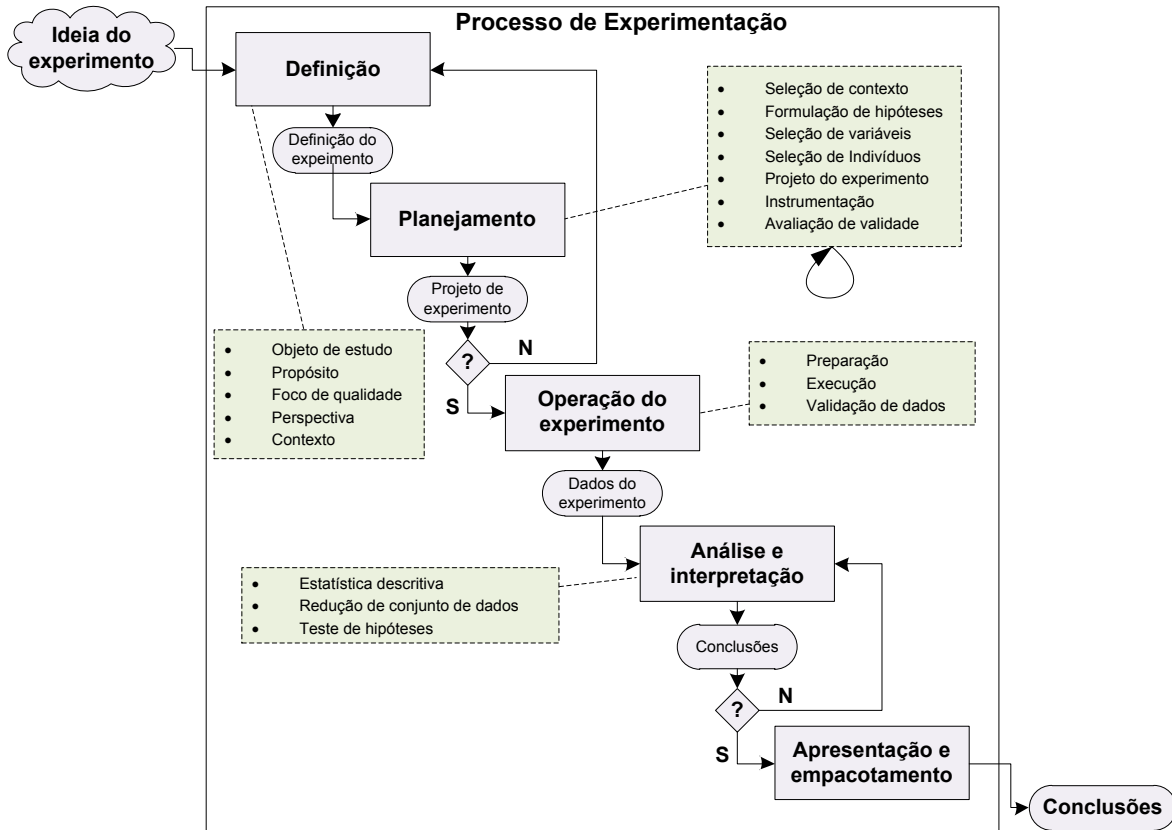


Figura 1 – Processo de Experimentação – adaptado de (WOHLIN et al., 2012).

Como apoio, os estudos experimentais atuam como ferramentas para obtenção dos dados necessários através de todo o processo de desenvolvimento de software, almejando resultados objetivos e significativos de forma a alcançar melhorias no processo. Também segundo Wohlin et al. (2012), tais estudos podem ser divididos 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. Nesta categoria, não controle de execução ou medidas.
- 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 sobre todos os elementos que o compõem. Podem ser efetuados experimentos controlados

em ambiente universitário, de forma a reduzir custos e riscos do que aplicá-los diretamente à indústria.

Somente experimentos verificam as teorias, podem explorar os fatores críticos, e dar luz ao fenômeno novo para que as teorias possam ser formuladas e corrigidas. O processo experimental proporciona de modo sistemático, disciplinado, e controlada a avaliação de processos e de atividades humanas ([TRAVASSOS; GUROV; AMARAL, 2002](#)).

Através da replicação de experimentos, pesquisadores adquirem conhecimento adicional a respeito dos conceitos estudados. Para que se possa replicar experimentos, é necessário que seu empacotamento seja realizado apropriadamente. Uma replicação em contextos diferentes sujeita o experimento a variações, dentre os quais fatores humanos, socioeconômicos e ambientais do âmbito que o experimento é realizado, possibilitando maior precisão na verificação da validade de hipóteses ([SHULL et al., 2003](#)). E para a replicação de um experimento, tanto intra quanto extra-grupo ([MENDONÇA et al., 2008](#)), é de suma importância a organização do registro das atividades de um experimento controlado, que são mantidas no chamado *Pacote de Laboratório*.

2.1 Pacotes de Laboratório

Dentro do âmbito da Engenharia de Software Experimental, a todo momento diversas pesquisas, técnicas e ferramentas são desenvolvidos para validar teses ou otimizar soluções, porém tais recursos ou informações isoladas não formam um corpo de conhecimento consistente, faz-se necessário compartilhá-los entre os grupos de pesquisa por meio do uso de pacotes de laboratório.

A condução de experimentos controlados ou de suas respectivas replicações, ambas necessitam de validação. A execução destes experimentos, tanto no papel de experimentador quanto de replicador, podem influenciar de forma positiva ou negativa sob muitos aspectos, principalmente em relação ao nível de experiência do condutor do experimento. Por fim, o conjunto de informações obtidos através da execução do processo experimental compõem uma nova instância de pacote de laboratório ([GARCIA et al., 2008](#)).

Diversos pesquisadores relatam dificuldades na revisão de pacotes de laboratório, como problemas no compartilhamento de conhecimento entre grupos de pesquisa devido uma falta de padronização para a integração de um conhecimento novo e/ou isolado ao conhecimento comum ([SCATALON; GARCIA; CORREIA, 2011](#)). Desta forma, é imprescindível uma boa definição e construção de um pacote de laboratório com o uso de uma estrutura específica de simples compreensão, possibilitando inclusive o uso de ontologias para seu desenvolvimento.

[Garcia et al. \(2008\)](#) propõem o uso de uma ontologia para apoiar a atividade de

Empacotamento processo de Experimentação no contexto de Engenharia de Software, que descreve os conceitos que compõem um pacote de laboratório para experimentos controlados, chamada *ExperOntology*, apresentada na Figura 2. 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. As regras (axiomas) da *ExperOntology* não são apresentados aqui, mas podem ser observados em (GARCIA et al., 2008).

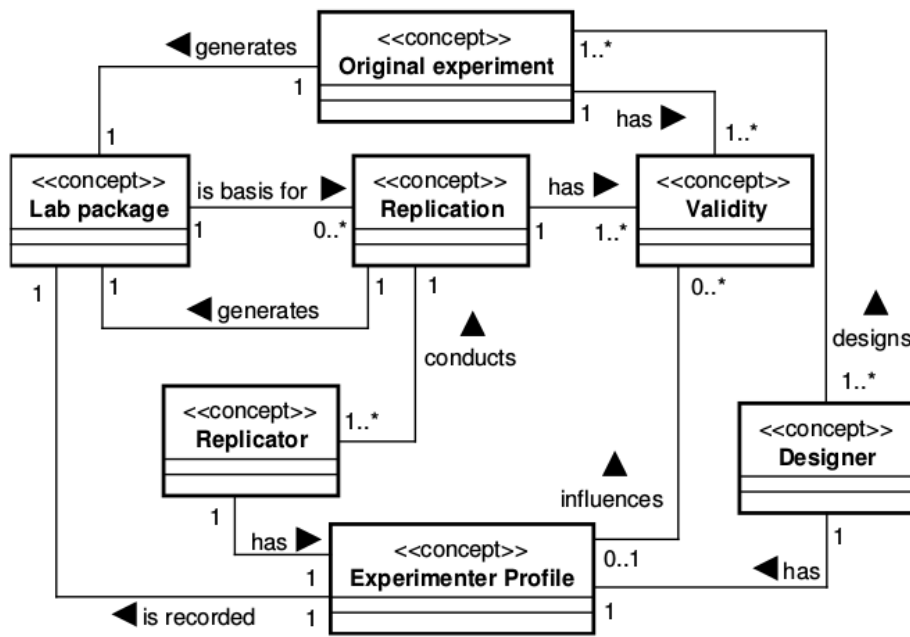


Figura 2 – Ontologia para Experimentos Controlados (GARCIA et al., 2008).

A *ExperOntology* visa a definir o principais conceitos de experimentos controlados desde a fase de definição até a análise de resultados, sendo importante ressaltar a evolução do experimento e o uso de pacotes de laboratório para armazenamento (GARCIA et al., 2008).

Na Figura 3 estão descritos os conceitos que devem fazer parte de um pacote de laboratório, segundo a *ExperOntology*. Inicialmente, é definido um propósito, um contexto, um objeto de estudo e o foco a serem considerados no estudo controlado; em seguida, é elaborada de uma hipótese inicial, a qual juntamente com uma objeto de estudo e um contexto formam a formalização da hipótese. A partir deste ponto, 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.

O principal objetivo das fases de definição e planejamento é estabelecer um modelo experimental que satisfaça todos os requisitos para a fase de análise. O resultado destas

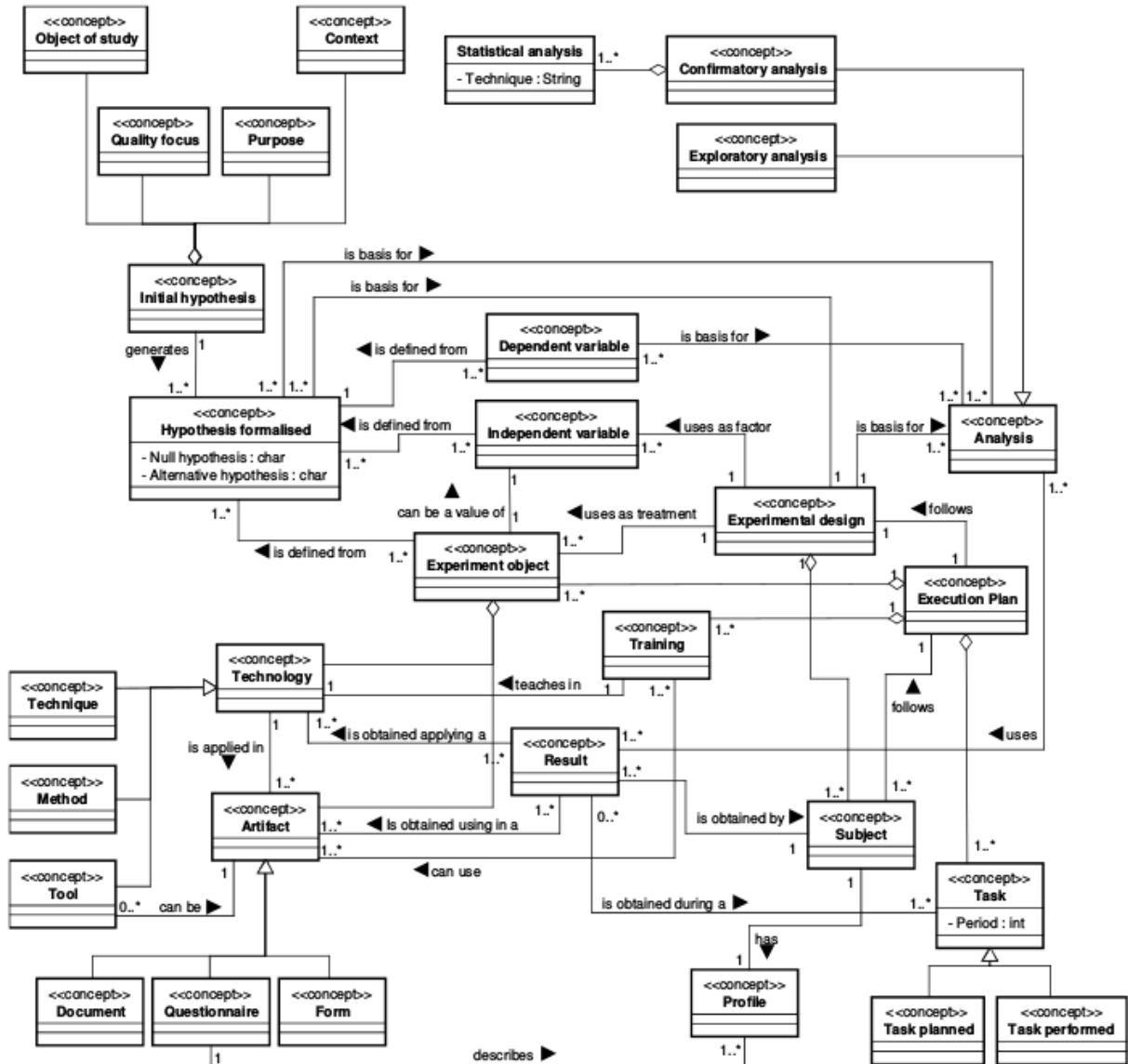


Figura 3 – Ontologia para Pacotes de Laboratório (GARCIA et al., 2008).

fases culmina em um modelo experimental e no plano de execução, o qual permite a definição de um ambiente controlado, realização de testes das hipóteses e minimização de ameaças para a validação do experimento.

Todos os conceitos apresentados devem ser mantidos no Pacote de Laboratório e, portanto, devem ser instanciados em uma base de dados ou outro meio equivalente para persistir os dados.

2.2 Bancos de Dados Não-Relacionais

Diversas tecnologias para armazenamento de dados têm sido propostas para a persistência de dados, como os Sistemas Gerenciadores de Banco de Dados (SGBDs).

O modelo relacional de dados tem sido utilizado em larga escala desde a sua criação. Este modelo tem como característica a utilização de tabelas e tuplas para o armazenamento de dados, assim como o uso de chaves primárias para garantia de unicidade (identificação única de elementos de dados) ([BRITO, 2010](#)).

Mediante ao crescente número de aplicações, o volume de dados tem aumentado exponencialmente nos últimos anos, e com tal crescimento limitações do modelo relacional tem ficado evidente, principalmente quanto à eficiência na recuperação de dados e escalabilidade ([TOTH, 2011](#)). Para este projeto, a principal limitação é a necessidade de se ter uma estrutura de tabelas pré-definidas, com seus respectivos atributos, o que pode ser uma restrição, pois experimentos controlados podem ter variações nos dados tratados, difícil de ser acomodados em uma estrutura relacional.

Como alternativa, há soluções tecnológicas alternativas que priorizam flexibilidade quanto ao armazenamento, desse modo, não empregam regras presentes no modelo relacional tradicional. Um dessas alternativas é o Modelo Não-Relacional (NoRel), o qual não tem como objetivo substituir o modelo relacional por completo, mas somente em casos em que seja mais vantajoso utilizá-lo, como em ambientes de Big Data. Alguns representantes de base não-relacionais são: Cassandra ([CASSANDRA, 2014](#)), Dynamo ([SIVASUBRAMANIAN, 2012](#)), MongoDB ([BANKER, 2011](#)) e BigTable ([CHANG et al., 2008](#)).

Devido à inexistência de regras para a organização dos seus dados, 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).

Por fim, de modo a esclarecer as reais vantagens e desvantagens do uso de uma base de dados não-relacional, na Tabela 2.2 é apresentado um comparativo com o tradicional modelo relacional analisando os quesitos de consistência, escalonamento e disponibilidade (BRITO, 2010).

Perante esta conceituação, a camada de persistência será elaborada utilizando o banco de dados *MySQL* (MYSQL, 2001) devido à utilização da ferramenta *OntoExp-Tool* (Pucci Neto et al., 2014), mantendo assim a compatibilidade para os modelos já

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

	Relacional	Não-Relacional
Escalonamento	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 predefinido, o modelo possui maior flexibilidade o que favorece a inclusão transparente de outros elementos.
Consistência	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.
Disponibilidade	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.

criados. Para a persistência da camada de interface será utilizada uma estrutura orientada a documentos, utilizando *eXtensible Markup Language* (XML). Tal decisão foi tomada pois a estrutura orientada a documentos se mostrou adequada tanto para o armazenamento, quanto para a transferência de protocolos experimentais entre experimentador e replicador. Adicionalmente, essa organização estrutural permite que os dados de um experimentos sejam mantidos em uma base de dados. e seu protocolo, expresso em BPMN (armazenado em XML) possa ser compartilhado, sem expor os dados. O compartilhamento do protocolo juntamente com os dados do experimento (dados coletados) podem ser compartilhados, desde que desejado pelo experimentador (responsável pela criação do experimento e, conseqüentemente, do seu Pacote de Laboratório).

3 Modelagem de Processo de Negócio

Para a elaboração dos modelos de processos de negócio, é relevante o conhecimento de todos os elementos envolvidos na execução de processos, tais como atores, clientes internos e externos, recursos e limitações; por meio dos quais é estabelecido um modelo que propicia um melhor entendimento, organização e representação, seja esta uma visão contextual abstrata ou com alto nível de detalhamento, conforme necessário.

Tipicamente, a representação de um modelo de processos inclui ícones, que representam atividades, eventos, decisões, condições e outros elementos do processo (BRAZIL, 2011). Esses elementos podem conter informações sobre os relacionamentos entre eles e com o ambiente, bem como sobre o comportamento de cada elemento na cadeia de processos.

3.1 *Business Process Modeling and Notation*

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. Após a padronização, foram lançadas algumas versões subsequentes, atualmente encontrando-se na versão 2.0 (COMMITTEE et al., 2016).

Segundo Correia e Abreu (2015), *Business Process Modeling and Notation* (BPMN) é atualmente a notação de modelagem de processos de negócio mais usada entre profissionais da área, devido a sua flexibilidade e abrangência. Consiste em uma tecnologia recente e pode ser utilizada por profissionais com diferentes níveis de conhecimento técnico. A formalização dos conceitos de modelagem de processos de negócio é baseada em um metamodelo construído a partir da linguagem *Unified Modeling Language* (UML).

A construção de processos de negócio através desta notação, utiliza um conjunto de padrões denominado “metamodelo para definição de processos de negócio” (COMMITTEE et al., 2016). Nesse sentido, a especificação BPMN provê uma notação gráfica para representar processos de negócio por meio de um diagrama chamado “diagrama de processo de negócio”. Esse diagrama é elaborado a partir de um conjunto de elementos gráficos que compõem diagramas simples de serem desenvolvidos e compreendidos.

A Figura 4 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

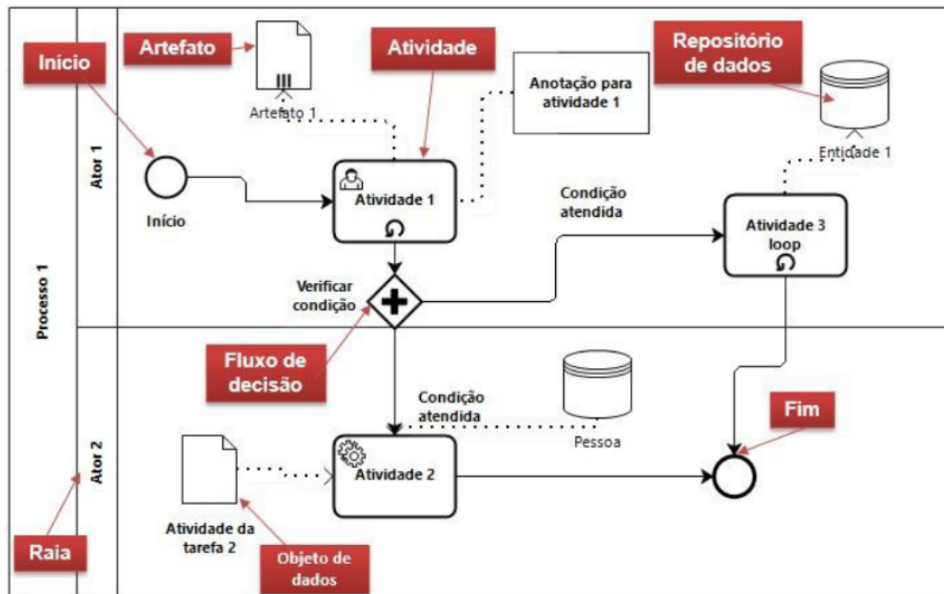


Figura 4 – Exemplo de diagrama BPMN – adaptado de (WESKE, 2012).

repositórios de dados.

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*). Na a Figura 12 há uma visão geral dos elementos da atual versão do BPMN.

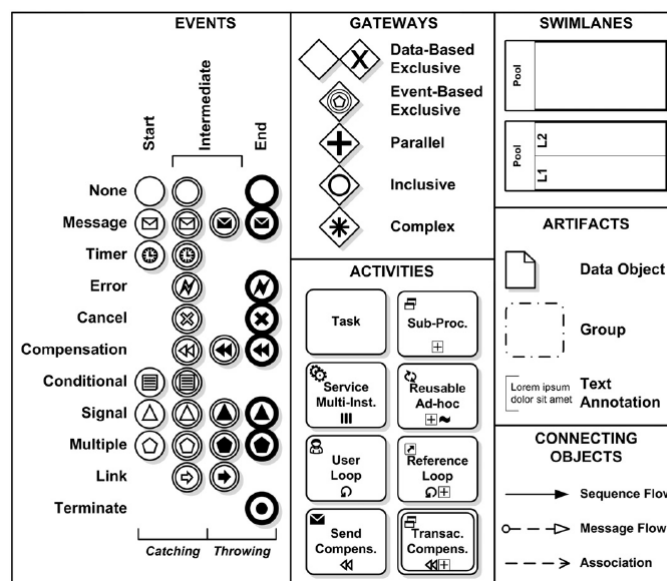


Figura 5 – Conjunto de elementos que compõem a versão BPMN 2.0 (COMMITTEE et al., 2016).

3.2 Protocolos de experimentação usando BPM

Compreender o projeto do experimento e revisar suas informações é fundamental não só para a execução do experimento, mas também para sua replicação. Diversos pesquisadores tem realizado projetos pilotos que ocasionam somente aumento de custos (KITCHENHAM, 2008).

A utilização da notação de processo de negócio pode contribuir diretamente tanto na concepção quanto na execução do protocolo de estudo experimental. Primeiramente, devido à facilidade de compreensão do uso que compõem a respectiva notação, principalmente pelo fato de consistir em um padrão UML. Em segundo ponto, a utilização de pacotes de laboratório para o armazenamento das informações relativas aos protocolos realizados em cada experimento.

Através destes modelos de processo a seguir, é possível observar a sequência de execução das atividades, os fatores limitantes, tais como tempo, e também a estruturação de cada um dos processos e suas subdivisões.

Nas figuras 6, 7 e 8 são apresentados os diagramas de processo de negócio referentes ao protocolo de experimentação aplicado como parte do projeto de mestrado intitulado “ModelUI_{VIZ} - 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.

A representação do modelo de diagrama de processo de negócio também deve ser armazenado no pacote de laboratório, viabilizando o compartilhamento do protocolo juntamente com os dados referentes ao experimento, tais como hipóteses, variáveis dependentes e independentes.

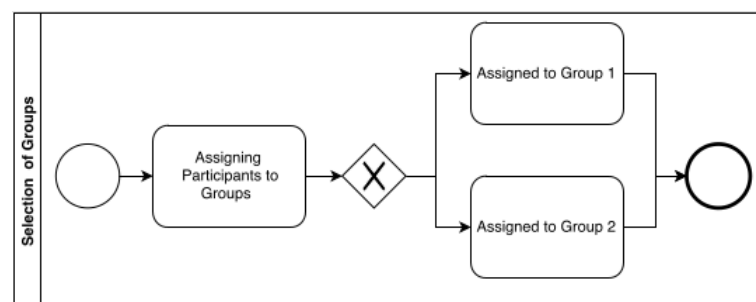


Figura 6 – Processo de atribuição de participantes aos grupos.

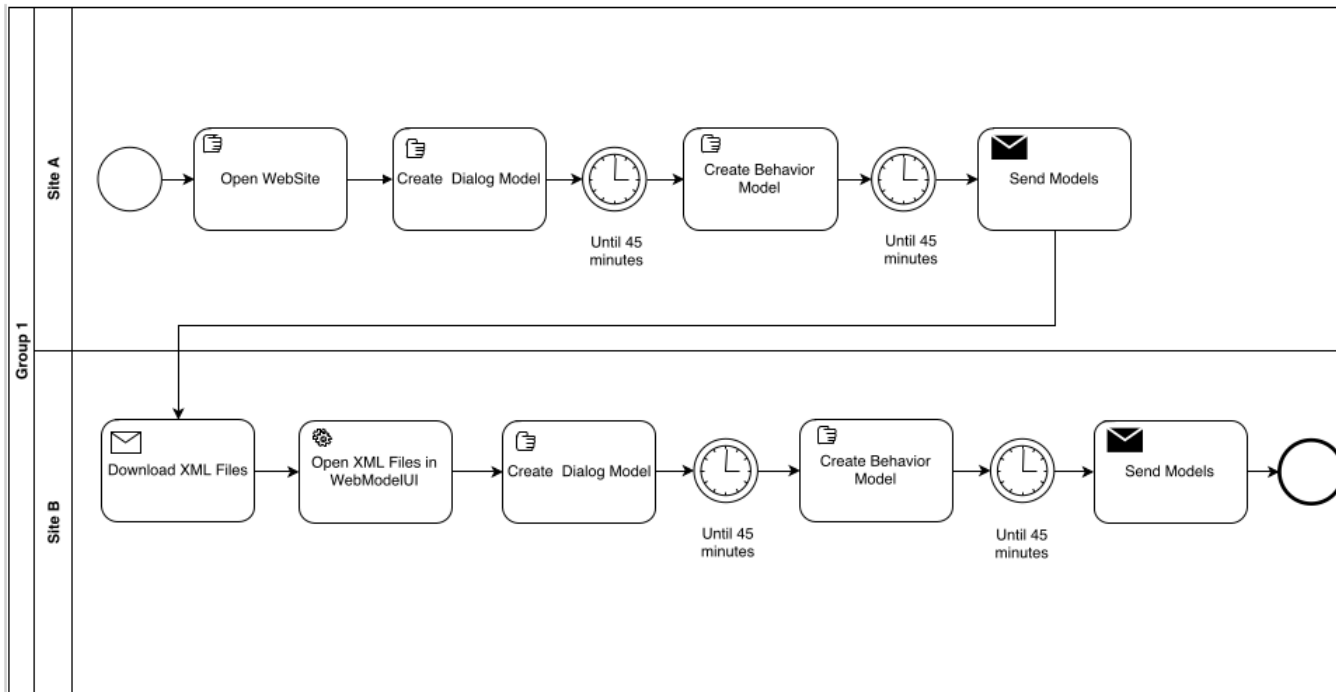


Figura 7 – Diagrama BPM referente às atividades dos participantes do primeiro grupo.

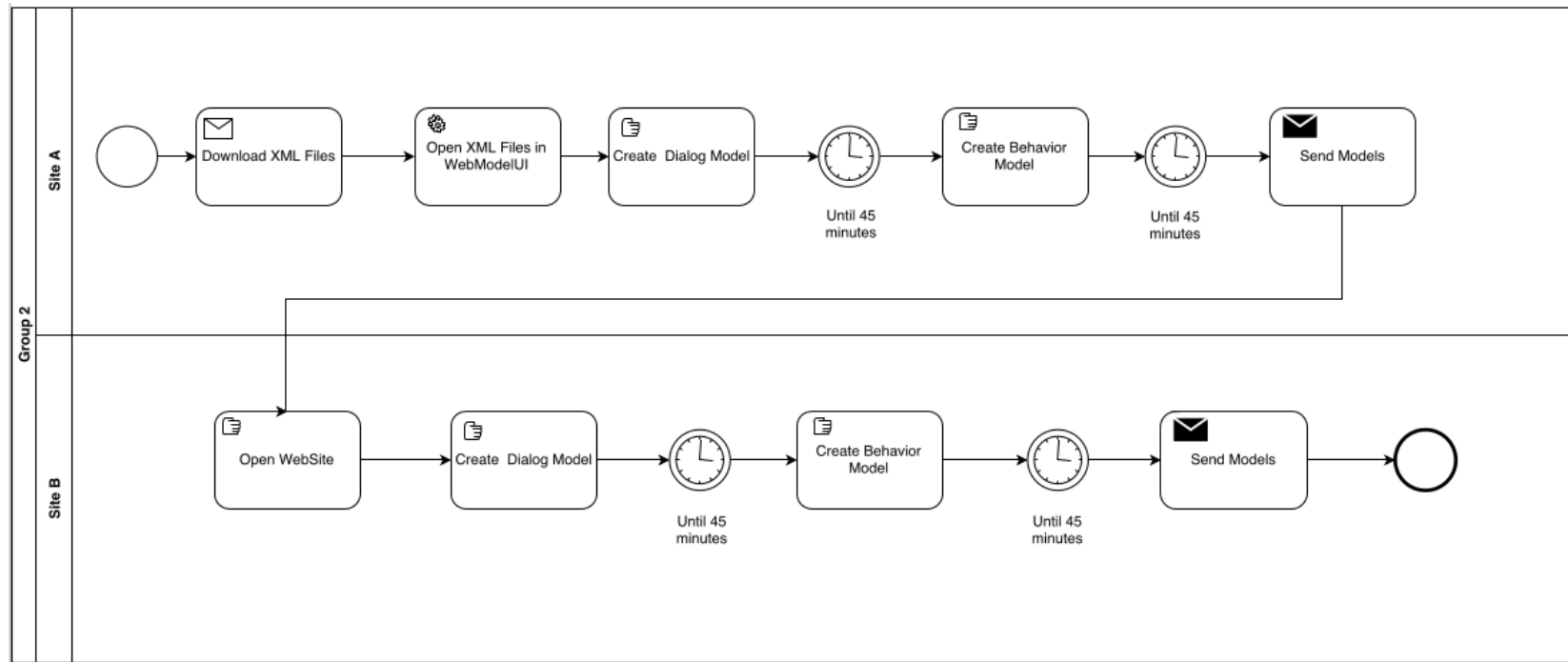


Figura 8 – Diagrama BPM referente às atividades dos participantes do segundo grupo.

4 Protótipo da Interface de Modelagem

Neste capítulo é apresentada a interface da implementação parcial (situação atual do desenvolvimento) tanto da camada de interface para modelagem gráfica quanto da camada de persistência para a notação. Primeiramente, em relação à camada de interface, trata-se de uma plataforma *web* e, para sua implementação, foram utilizados a linguagem de marcação HTML 5 (*HyperText Markup Language*), o CSS 3 (*Cascading Style Sheets*) e a linguagem *JavaScript*. Adicionalmente, as bibliotecas *JQuery* e *SVGPanZoom* são utilizadas para facilitar a implementação de recursos gráficos e execução de requisições HTTP/Ajax para a comunicação com o servidor, e manipulação dos elementos gráficos por meio de operações de escala e deslocamento, respectivamente.

A ferramenta foi projetada para ser simples e intuitiva, com foco na elaboração dos diagramas. A página inicial é apresentada na Figura 9.

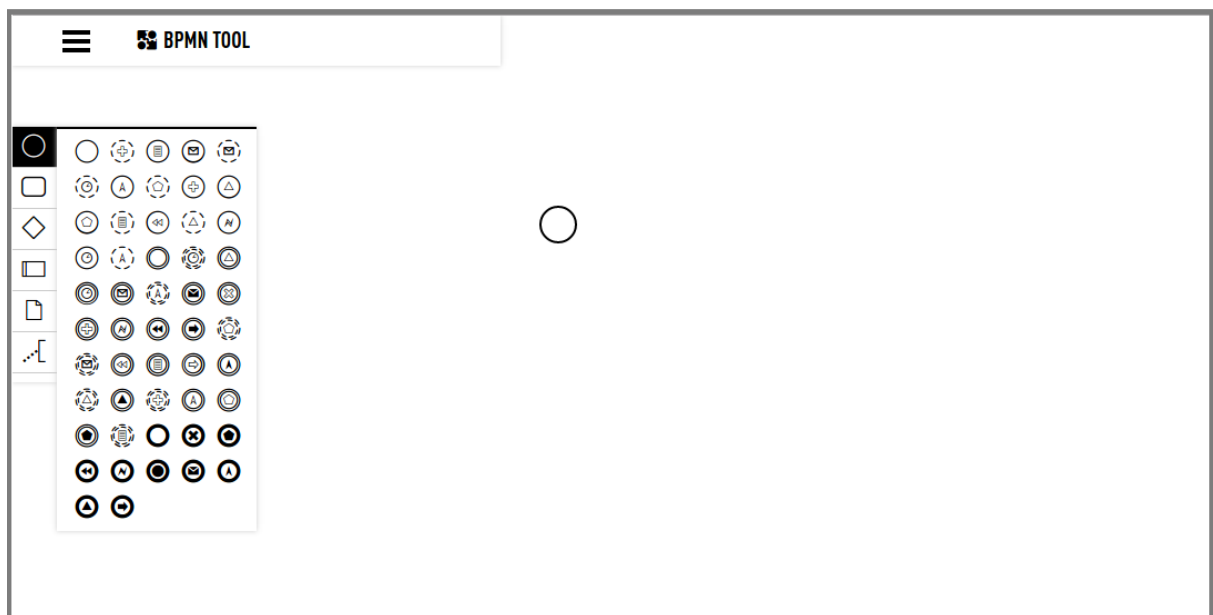


Figura 9 – Interface de modelagem da notação.

Para a criação de um novo modelo de diagrama de processo de negócio, é necessário somente realizar a inserção dos componentes do diagrama, ou caso já haja um modelo representado, o usuário deve selecionar a opção "*Menu -> New*" para iniciar a representação. O menu da ferramenta é apresentado na Figura 10.

Ainda em relação à Figura 10, é possível observar outras opções. Na opção "*Menu -> Open*", o usuário abrir um modelo de processo de negócio elaborado previamente nesta ferramenta, armazenados em arquivos com a extensão *.bpmn*. A opção "*Menu -> Save as*"

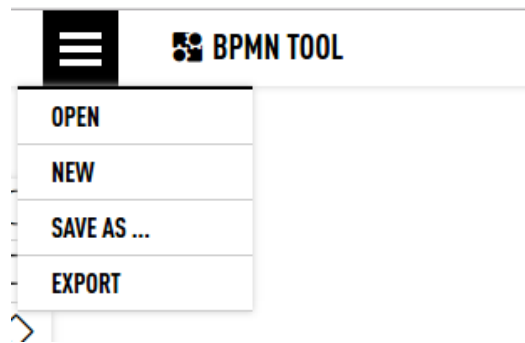


Figura 10 – Menu de opções da ferramenta.

permite o armazenamento do modelo no mesmo formato citado anteriormente, sendo este um diagrama concluído ou em processo de criação. Por fim, a opção “Menu -> Export” visa armazenar o modelo em formato de imagem vetorial, no formato SVG (*Scalable Vector Graphics*), de forma que, o usuário, mesmo sem acesso a esta ferramenta, possa ter acesso ao conteúdo do modelo, ao menos, para leitura de seu conteúdo.

Para a construção do modelo de processo de negócio, o usuário faz uso de elementos tais como: *Events*, *Gateways*, *Tasks*, *Pools/Lanes*, *Data Objects* e *TextAnnotations*. Estes elementos são encontrados na ferramenta pelo menu lateral, no qual estão presentes todos os itens dessa categorias listadas anteriormente.

Na Figura 11, todos os eventos e suas derivações são listados.

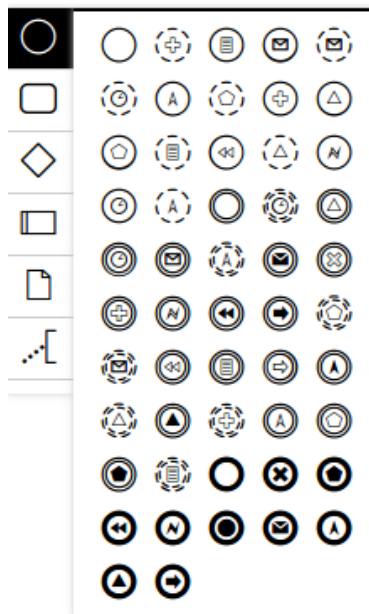


Figura 11 – Painel de elementos da categoria de eventos.

Na Figura 12(a) são apresentados os elementos que compõem a categoria de portões. E na Figura 12(b) são exibidos os elementos referentes a piscinas e raia.

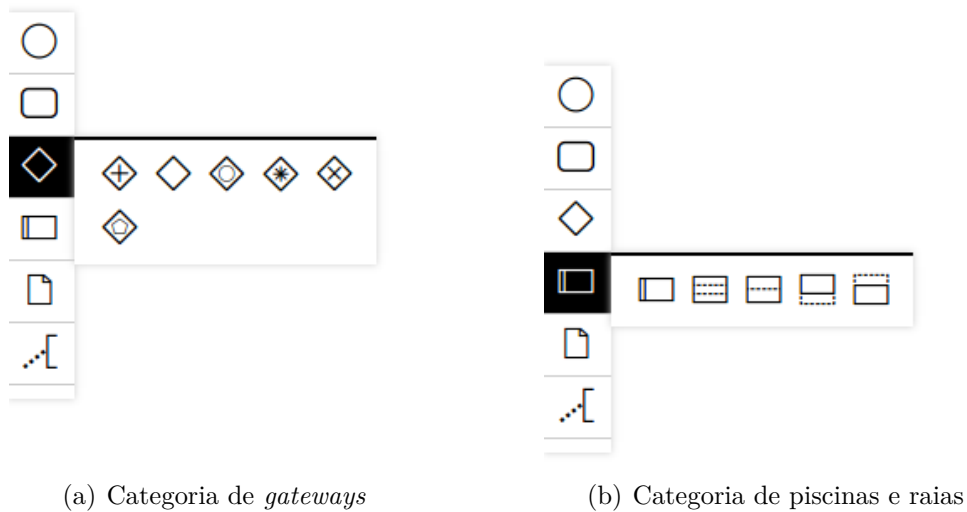


Figura 12 – Painel de elementos

Cada elemento pertencente à notação será persistido no formato XML para composição do pacote de laboratório. A figura 13 representa uma tarefa da categoria de mensagens, este elemento será armazenado no modelo apresentado na listagem 4.1

Figura 13 – Exemplo de elemento *MessageTask*.Listagem 4.1 – Descrição do elemento *Mensagem*

```

1 <messagetask id="23">
2   <type>task</type>
3   <from>15</from>
4   <to>42</to>
5   <value>Download XML Files</value>
6 </messagetask>

```

Em relação ao servidor da plataforma *web* em desenvolvimento, está sendo utilizada a linguagem de programação *Java*, o servidor *GlassFish Server 4.1* e o *JSP - Java Server Pages*. Adicionalmente, para o suporte à serialização dos dados no formato XML (*eXtensible Markup Language*) é utilizada a biblioteca *XStream*. Esta parte da aplicação provê suporte tanto à leitura e escrita dos arquivos, quanto à validação dos diagramas, em termos de completude e corretitude, e também a extração do conteúdo de pacotes de laboratório e criação do respectivo modelo de processo. Aliás, a estrutura do arquivo XML deverá conter tanto elementos para representar o modelo visual BPMN (como coordenadas para posicionamento de elementos gráficos na interface), quanto elementos que representam os conceitos do Pacote de Laboratório.

Atualmente, em relação à camada de persistência, todos os elementos compõem a primeira versão da notação já foram implementados, e está em andamento a implementação dos elementos da segunda versão da notação BPM. Atualmente, foram implementadas trinta e uma classes, divididas em seis pacotes conforme a categorização dos elementos da notação. Tem sido adotada uma abordagem evolutiva, portanto os diagramas, assim como o modelo de classes, têm sido, também, alvo de contante evolução.

A seguir, na Listagem 4.2 é apresentada a classe *BusinessProcessDiagram* (apenas dados privados, que devem ser armazenados), a qual representa o componente básico de um diagrama BPMN, sendo responsável pela organização de todo o diagrama. O armazenamento desse elemento em pacote de laboratório está representado na Listagem 4.3.

Listagem 4.2 – Estrutura de Dados inicial da classe diagrama BPM

```
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 }
```

Listagem 4.3 – Estrutura do XML da classe diagrama BPM

```
1 <businessprocessdiagram id="1245" name="Selecao dos grupos">  
2     <author>Leandro Ungari</author>  
3     <language>pt-br</language>  
4     <queryLanguage>BPMN</queryLanguage>  
5     <creationDate>01/05/2017</creationDate>  
6     <modificationDate>10/05/2017</modificationDate>  
7     <pools>  
8         <pool id="1">...</pool>  
9         <pool id="2">...</pool>  
10        <pool id="3">...</pool>  
11        ...  
12        <pool id="n">...</pool>  
13    </pools>  
14    <documentation>Elemento formalizado pela documentacao BPMN 2.0 ...  
15    </documentation>  
16 </businessprocessdiagram>
```


5 Plano inicial e atividades realizadas

5.1 Cronograma Inicial

O cronograma de atividades foi organizado de acordo com a Tabela 3. As atividades foram divididas mensalmente, com início em dezembro de 2016. Nessa tabela as letras 'x' correspondem às atividades previstas e as letras 'X' às atividades já realizadas.

- *Atividade 1*: Revisão Bibliográfica (e da tecnologia a ser utilizada);
- *Atividade 2*: Definição dos requisitos da camada de interface;
- *Atividade 3*: Desenvolvimento da camada de interface;
- *Atividade 4*: Avaliação da interface desenvolvida;
- *Atividade 5*: Relatório Parcial;
- *Atividade 6*: Relatório Final.

Tabela 3 – Cronograma de Atividades

Atividades	Período											
	2016	2017										
	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov
1	X	X	X	X								
2			X	X								
3				X	X	x	x	x	x			
4									x	x	x	x
5						X						
6											x	x

5.2 Atividades Realizadas

As atividades desenvolvidas até o presente momento foram as seguintes: foi realizada uma breve revisão bibliográfica sobre Engenharia de Software Experimental, com foco em experimentos controlados e o uso de Pacotes de Laboratório para a transferência de dados entre grupos de pesquisa, considerando o conhecimento adquirido em iniciações científicas anteriores. Em seguida, foi aprofundado o estudo em relação ao uso da notação BPM e como esta poderia ser aplicada em contexto de estudos experimentais.

Atualmente, a partir dos requisitos da camada de interface para modelagem, foi iniciada a implementação desta assim como da camada de persistência.

Em relação às próximas atividades, o próximo passo consiste em finalizar a construção da camada de persistência e da interface gráfica de modelagem. Em seguida, realizar a integração com a ferramenta *OntoExpTool* e, desta forma, avaliar o uso da notação para a construção de protocolo de experimentação. Por fim, será realizada a escrita do relatório científico final.

6 Considerações Finais

Nessa primeira parte do projeto, além dos estudos teóricos, foi implementada a camada de persistência referente à primeira versão da respectiva notação, assim como classes e métodos auxiliares armazenar no formato *XML*, conforme a estruturação de um pacote de laboratório. Paralelamente, tem sido desenvolvida a camada de interface para a construção dos diagramas de processo de negócio.

Para uma avaliação mais precisa do uso da notação BPM na elaboração de processo de elaboração de protocolos de experimentos, é necessária a conclusão das duas atividades citadas acima, assim como a transferência dos dados de um pacote de laboratório para a representação gráfica.

A continuidade das atividades corresponde ao encerramento da implementação da camada de persistência assim como da interface de modelagem. Em seguida, será estabelecida a integração da interface de modelagem BPM com a ferramenta *OntoExpTool*, de modo a visualizar graficamente o protocolo do experimento armazenado no pacote de laboratório.

A execução das atividades do projeto está dentro do esperado, desse modo, a conclusão das atividades deve ocorrer dentro dos prazos estabelecidos, assim como a elaboração do relatório científico final deste projeto de iniciação científica.

Referências Bibliográficas

- BANKER, K. *MongoDB in action*. [S.l.]: Manning Publications Co., 2011. 8
- 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. 3
- BRAZIL, A. *BPM CBOK V3. 0: Guia para o Gerenciamento de Processos de Negócio-Corpo Comum de Conhecimento, 3ª edição*. ABPMP Brazil. 2011. 11
- BRITO, R. W. Bancos de dados nosql x sgbd's relacionais: análise comparativa. *Faculdade Farias Brito e Universidade de Fortaleza*, 2010. iii, 8, 9, 10
- CASSANDRA, A. Apache cassandra. *Website*. Available online at <http://planetcassandra.org/what-is-apache-cassandra>, p. 13, 2014. 8
- 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. 8
- COMMITTEE, O. M. G. B. T. et al. *Business Process Model and Notation, version 2.0*, 2010. 2016. ii, 11, 12
- 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. 11
- 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. ii, 5, 6, 7
- 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. 13
- 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

SCATALON, L. P.; GARCIA, R. E.; CORREIA, R. C. M. Packaging controlled experiments using an evolutionary approach based on ontology. In: *International Conference on Software Engineering and Knowledge Engineering (SEKE 2011)*. [S.l.: s.n.], 2011. p. 408–413. [5](#)

SHULL, F. et al. Replicated studies: building a body of knowledge about software reading techniques. In: *Lecture notes on empirical software engineering*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2003. p. 39–84. [3](#), [5](#)

SIVASUBRAMANIAN, S. Amazon dynamodb: a seamlessly scalable non-relational database service. In: ACM. *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. [S.l.], 2012. p. 729–730. [8](#)

TOTH, R. M. Abordagem nosql—uma real alternativa. *Sorocaba, São Paulo, Brasil: Abril*, v. 13, 2011. [8](#), [9](#)

TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. A. G. *Introdução à Engenharia de Software Experimental*. [S.l.], 2002. [3](#), [5](#)

VAISH, G. *Getting started with NoSQL*. [S.l.]: Packt Publishing Ltd, 2013. [8](#), [9](#)

WESKE, M. *Business Process Management: Concepts, Languages, Architectures*. New York: Springer Heidelberg Dordrecht, London, 2012. [ii](#), [12](#)

WOHLIN, C. et al. *Experimentation in Software Engineering: An introduction*. Boston, USA: Kluwer Academic Publishers, 2012. [ii](#), [3](#), [4](#)