

# P2BPMN: PROTÓTIPO PARA CONSTRUÇÃO DE MAPA DE PROCESSOS A PARTIR DE ARQUIVOS MS-PROJECT

André Luiz Lunelli, Mauro Marcelo Mattos – Orientador

Curso de Bacharel em Ciência da Computação  
Departamento de Sistemas e Computação  
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil

to.lunelli@gmail.com, mattos@furb.br

**Resumo:** *O presente projeto descreve um protótipo de uma ferramenta, que permite o mapeamento de um conjunto de atividades de um projeto especificadas em MS-Project, para uma notação em BPMN de tal forma que, a empresa possa ter uma visão de fluxo de processos a partir do detalhamento de um determinado projeto. São detalhados alguns aspectos da especificação e apresentados os resultados obtidos. O projeto foi avaliado por um pequeno conjunto de especialistas e as considerações apresentadas. Como um projeto piloto, os resultados obtidos foram satisfatórios, o que indica um caminho de possibilidades de expansões no sentido da construção de uma ferramenta mais completa, que permita transformar conhecimento tácito representado em modelos de gerenciamento de projetos em modelos gráficos de mapas de processos.*

**Palavras-chave:** *Gerência de projetos. Gerência de processos. Modelagem de negócios. Mapa de Processos. BPMN.*

## 1 INTRODUÇÃO

Eficiência e agilidade são atributos constantemente perseguidos por empresas e estes objetivos são alcançados na medida em que os processos de negócio interagem entre si. Neste cenário, o método de gestão de processos de negócio, Business Process Management, ou BPM, vem sendo adotado desde o início da década de 1990 para melhorar continuamente as atividades das empresas, a partir da integração de sistemas (CATELLI; SANTOS, 2004). Conforme Basso (2018), a BPM pode também ser usada para representar o comportamento de processo de software, já que se refere a processos de negócio.

Por outro lado, conforme Antoni, Forsberg e Kammerlind (2000) o conceito de gerenciamento de projetos moderno “[...]tem suas origens na década de 1930 e está intimamente relacionado com a eclosão da Segunda Guerra Mundial e sua demanda por sistemas militares complexos.” (p.1, tradução nossa), já que as organizações tradicionais da época tinham problemas para lidar com a complexidade técnica e organizacional dos empreendimentos em combinação com a pressão de tempo que prevalece em uma situação de hostilidades. Além disso, continuam os autores, “O objetivo de um processo é capturar as experiências de projetos, analisá-las, avaliá-las e filtrá-las, a fim de alimentá-las ativamente de volta às organizações do projeto.” (p.1, tradução nossa).

Neste contexto, a adoção de técnicas de gerenciamento de projetos tem por objetivo a obtenção de resultados efetivos com o uso adequado de recursos disponíveis. Para Baldam, Valle e Rozenfeld (2014, p.23), a ideia geral de processos surgiu com o fortalecimento da produção em massa, pois esta baseia-se em racionalização, inclui tarefas repetitivas, uso de ferramentas específicas e técnicas de execução em tempo mínimo. Carvalho e Notari (2017, p.17) qualificam o cenário atual da seguinte maneira, “Gerenciar bem projetos dentro da empresa tornou-se questão de sobrevivência das organizações.”.

Smith e Fingar (2007) caracterizam a BPM como uma, metodologia que abrange diversos conceitos desde o desenvolvimento de um plano de negócio até o controle gerencial da organização. Essa metodologia baseia-se em um conjunto de técnicas que unifica os aspectos de Gestão de Negócios e Tecnologia da Informação, com enfoque na otimização dos resultados através da melhoria e integração dos processos. Além disso, a BPM permite modelar um processo existente, testar inúmeras variações, gerenciar melhorias e ou inovações que a organização pretenda seguir e retornar os resultados destas análises dos processos com rapidez.

Flores e Sepúlveda (2011) afirmam que apesar da existência de padrões de modelagem de processos como Business Process Management Notation (BPMN), continua sendo difícil especificar restrições e relações temporais complexas entre tarefas de um dado processo, o que impede a especificação e subsequente automação de processos onde essas restrições são relevantes. Conforme Kostelac, Matrljan e Dobovicek (2011, p. 1119, tradução nossa):

Muitas organizações não estão obtendo um bom relacionamento entre processos e projetos. Os projetos têm uma má reputação – as chances de sucesso geralmente não são boas. O que geralmente acontece quando os projetos começam a vacilar é que existem forças invisíveis que tentam transformar o projeto em um processo. As chances de uma transformação bem-sucedida não são boas, a menos que a própria transformação seja reconhecida como um processo. Ao revisar as definições e

a literatura, torna-se aparente que o pensamento correto é que todo o trabalho é um processo e que os projetos se encaixam na estrutura do gerenciamento de processos. Se você não pode descrever o que você está fazendo como um processo, você não sabe o que está fazendo.

O projeto tem por objetivo a construção de um protótipo de uma ferramenta web que permita o mapeamento de um conjunto de atividades de um projeto especificado em MS-Project para uma notação em BPMN. De tal forma que a organização ou indivíduo, possa ter uma visão de fluxo de processos a partir do detalhamento de um determinado arquivo MS-Project.

A pesquisa foi realizada em duas etapas. A primeira etapa envolveu a prova de conceito do software e desenvolvimento de um protótipo para validação dos conceitos estudados. A segunda etapa utilizou uma abordagem qualitativa, com finalidade de validação preliminar, disponibilizada em rede social profissional (LinkedIn).

O trabalho está organizado como segue: a próxima seção apresenta os principais aspectos conceituais relacionados ao escopo do projeto. A seção três apresenta a descrição do protótipo construído. A seção quatro apresenta os resultados obtidos e finalmente são apresentadas as conclusões do trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta definições e conceitos importantes, com a finalidade de fundamentação e aprofundamento acerca dos assuntos abordados neste trabalho. Para os devidos fins, serão abordados: gerenciamento de projetos, MS-Project e XML, notação BPMN, biblioteca Bpmn.io e trabalhos correlatos, assim respectivamente.

### 2.1 GERENCIAMENTO DE PROJETOS

Conforme Vargas (2007), um projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo e envolve um determinado número de atividades coordenadas e controladas com datas de início e término. A natureza temporária dos projetos indica que eles têm um início e um término definidos sendo que o término é alcançado quando os objetivos de projeto são atingidos ou quando o projeto é encerrado porque os seus objetivos não serão ou não podem ser alcançados, ou quando a necessidade de projeto deixar de existir.

De acordo com o Project Management Institute (PMI, 2017), a atividade de gerenciamento de projetos é definida como a utilização do conhecimento, das habilidades e das ferramentas e técnicas nas atividades do projeto para garantir seus requisitos. As boas práticas de gestão de projetos utilizam a aplicação e integração dos 49 processos definidos no Guia PMBOK (especificado pelo PMI) os quais estão divididos logicamente nos seguintes grupos de processos:

- a) iniciação, formado por processos que indicam a autorização formal para iniciação de um novo projeto;
- b) planejamento, identifica riscos, custos, agendamento de atividades e datas de entrega e a maneira como tudo deve ocorrer para que o projeto seja concluído conforme combinado;
- c) execução de processos, envolve a coordenação de pessoas, realização das atividades de acordo com plano e fornece uma análise de realização do projeto;
- d) monitoramento e controle, analisa a fase atual do projeto com base nas atividades realizadas e pendentes, para realizar ações corretivas para que o projeto siga o planejado;
- e) encerramento, finaliza formalmente um projeto.

Tendo em vista qualificar o profissional na área de gerenciamento de projetos, o PMI possui uma série de certificações. A Tabela 1 mostra a evolução de profissionais qualificados no mundo e inclui o total do Brasil em dezembro de 2018 na última coluna. A coluna totalizadora demonstra a crescente necessidade de mão de obra qualificada para resolver problemas em projetos.

Tabela 1 – Evolução das qualificações em gerenciamento de projetos no mundo e no Brasil

Ano	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2018
Membros	259172	309715	334019	371575	397453	437794	454032	478493	472269	500461	556839	10843
PMP	273547	361238	412503	467390	510434	593074	639237	694534	745981	827960	884518	17445
CAPM	3988	10263	13272	16491	20157	24196	27168	30474	32868	34504	37258	615
PgMP		393	511	678	834	985	1161	1438	1788	2173	2550	18
PMI-SP		261	408	578	809	1036	1268	1448	1603	1782	1910	45
PMI-RMP		242	588	1195	1805	2507	3003	3443	3886	4450	5051	192
PMI-ACP				517	2063	4366	7282	10351	14021	19295	24998	342
PfMP							183	286	403	506	640	9
PMI-PBA							216	569	1127	2020	3073	24

Fonte: Adaptado de Sotille (2019).

## 2.2 MS-PROJECT E XML

O Microsoft Project, ou apenas MS-Project, é uma ferramenta desenvolvida pela Microsoft (MICROSOFT, 2019) com a finalidade de administrar o tempo do projeto, levantar atividades, duração, sequenciamento de tarefas, direcionamento de recursos, acompanhamento, levantamento de custos, definição de cronogramas e grande variedade de artefatos como gráficos e relatórios (SANTOS, 2013).

Uma das características do MS-Project é a possibilidade de desdobramento das atividades de um projeto em subatividades. A Figura 1 apresenta um exemplo deste recurso e o Quadro 1 mostra um pequeno exemplo para ilustrar aspectos que são utilizados no escopo do trabalho, destacando-se os atributos OutlineNumber e OutlineLevel, que servem para indicar o posicionamento e profundidade de uma tarefa, o que indica que filho ela é e a qual tarefa pai ela pertence.

Figura 1 – Exemplo de detalhamento das atividades de um projeto

Task Name	Duração	Início	Término	Pred	Nomes dos recursos
Develop Project Charter	0,5 dias?	Seg 21/05/18	Seg 21/05/18	5	Project Manager; Acceptor
▾ Develop Preliminary Project Scope Statement	4 dias?	Seg 21/05/18	Sex 25/05/18	2	
Conduct Planning Workshop	1 dia?	Seg 21/05/18	Ter 22/05/18		Project Director
Document Project Costs and Benefits	0,5 dias?	Ter 22/05/18	Ter 22/05/18	8	Application Architect Technical
Develop High Level Work Breakdown Structure	1 dia?	Qua 23/05/18	Qua 23/05/18	9	Project Manager
Conduct Peer Review	1 dia?	Qui 24/05/18	Qui 24/05/18	10	Project Manager
Prepare Preliminary Project Scope Statement	0,5 dias?	Sex 25/05/18	Sex 25/05/18	11	Project Manager; Acceptor
▾ Planning	13 dias?	Sex 25/05/18	Qua 13/06/18	1	
▸ Set Up Project Environment	4 dias?	Sex 25/05/18	Qui 31/05/18		
▾ Define Scope	4 dias?	Qui 31/05/18	Qua 06/06/18	14	
Document Scope Management	1 dia?	Qui 31/05/18	Sex 01/06/18		Project Manager

Fonte: Microsoft Project Office 365 (2019).

Quadro 1 – Elementos e dados do MS-Project em XML

```

1  <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
2  <Project xmlns="http://schemas.microsoft.com/project">
3    <Title>Project Management for MS Website</Title>
4    <Author>Aluno</Author>
5    <CreationDate>2018-05-16T21:10:00</CreationDate>
6    <StartDate>2018-05-16T08:00:00</StartDate>
7    <FinishDate>2018-06-21T12:00:00</FinishDate>
8    //...
9    <Tasks>
10     <Task>
11       <Name>Project Management for MS Website</Name>
12       <OutlineNumber>0</OutlineNumber>
13       <OutlineLevel>0</OutlineLevel>
14       <Start>2018-05-16T08:00:00</Start>
15       <Finish>2018-06-21T12:00:00</Finish>
16       <Duration>PT212H0M0S</Duration>
17       <ManualStart>2018-05-16T08:00:00</ManualStart>
18       <ManualFinish>2018-06-21T12:00:00</ManualFinish>
19       <ManualDuration>PT212H0M0S</ManualDuration>
20       <Work>PT224H0M0S</Work>
21       <Estimated>1</Estimated>
22       <EarlyStart>2018-05-16T08:00:00</EarlyStart>
23       <EarlyFinish>2018-06-21T12:00:00</EarlyFinish>
24       <LateStart>2018-05-16T08:00:00</LateStart>
25       <LateFinish>2018-06-21T12:00:00</LateFinish>
26       <WorkVariance>13440000.00</WorkVariance>
27     </Task>
28   </Tasks>
29   //...

```

Fonte: elaborado pelo autor.

### 2.3 NOTAÇÃO BPMN

Conforme o grupo responsável pela padronização, gerenciamento e evolução de especificações relacionadas a BPMN, o OMG (Object Management Group), o uso da notação BPMN é extensa. A modelagem de processos de negócios é usada para comunicar uma ampla variedade de informações para uma ampla variedade de usuários finais de diferentes áreas (OMG, 2011). Assim como há esta variedade de usuários, também existe uma variedade de aplicações em torno desta notação, consumindo e produzindo informação e oferecendo novos diferenciais de mercado. Por conseguinte, há a necessidade de padronização na criação de arquivos da notação BPMN.

Os símbolos do diagrama BPMN são separados em 4 grupos principais: objetos de fluxo, conectividade, raias e artefatos, conforme a Figura 2. Objetos de fluxo são responsáveis pela definição do comportamento do processo. Dentro do grupo de fluxo, tem-se os elementos para: eventos, atividades e porta de entrada. Objetos de fluxo definem a ordem e o fluxo do processo, bem como associações a diversos elementos da BPMN. As raias são utilizadas para organizar determinado grupo de aspectos que um processo tem, subdividindo em 2 elementos, piscina e raia. Os artefatos representam informações importantes dentro do processo, porém não estão ligados diretamente a elementos, como anotações ou representações virtuais de objetos no mundo físico ou interação com usuário (LUCID SOFTWARE INC, 2019).

Figura 2 – Elementos básicos da BPMN

Elemento	Símbolo gráfico	Elemento	Símbolo gráfico
Evento de início		Base de dados	
Evento intermediário		Fluxo de seqüência	
Evento de fim		Fluxo de mensagem	
Tarefa		Associação	
Subprocesso		Pool	
Gateway		Lane	
Objeto de dados		Grupo	
		Anotação de texto	

Fonte: Giroto (2016).

O Core da especificação da BPMN inclui os elementos mais fundamentais da especificação, estruturado em camadas: Foundation, Service e Common (OMG, 2011, p. 49). Todo documento BPMN deve ter o elemento `<bpmn:definitions>` como nodo raiz e todos os outros elementos devem estar envolvidos por ele (OMG, 2011, p. 475). Os elementos principais para garantir a correta visualização dos modelos em BPMN são: `<bpmn:process>` e `<bpmndi:BPMDiagram>`. O elemento `<bpmn:process>` é responsável por definir os elementos que constituem o modelo, sejam eles eventos de inicialização, tarefas, portas de entrada, eventos de finalização, entre outros e como se relacionam entre si. Esse relacionamento é definido pelos atributos `sourceRef` e `targetRef` do elemento `<bpmn:sequenceFlow>` e identificado com o atributo `id` para ser desenhado posteriormente na forma de uma seta, onde o curso da direção é definido do atributo `sourceRef` que identifica o elemento de origem, para o atributo `targetRef` que identifica o elemento destino da seta. A maioria dos elementos visuais devem estender da classe `BaseElement`, para padronizar o uso do atributo `id`, que permite que todos os elementos possam ser identificados e manipulados. Inclusive para a correta visualização de origem e destino das setas deve-se respeitar o recuo do elemento `<bpmn:sequenceFlow>` para que não fique sobreposta por outro elemento, como no caso de uma seta dentro de um subprocesso apontando para outro subprocesso mais à dentro, como mostra o exemplo do Quadro 2 nas linhas 4 e 5.

Quadro 2 – Elementos de definição com fluxo de seqüência

```

1 <?xml version="1.0"?>
2 <bpmn:definitions xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL" xmlns:bpmn="http://www.omg.org/spec/BPMN/20100524/MODEL">
3   <bpmn:process id="process_1" isExecutable="">
4     <bpmn:sequenceFlow id="sequenceFlow_5cbb5f710a1d9" sourceRef="StartEvent_5cbb5f710a178" targetRef="TaskActivity_5cbb5f7080b3b"/>
5     <bpmn:sequenceFlow id="sequenceFlow_5cbb5f710a258" sourceRef="TaskActivity_5cbb5f7080b3b" targetRef="EndEvent_5cbb5f710a19b"/>
6     <bpmn:startEvent id="StartEvent_5cbb5f710a178" name="">
7       <bpmn:outgoing>sequenceFlow_5cbb5f710a1d9</bpmn:outgoing>
8     </bpmn:startEvent>
9     <bpmn:subProcess id="TaskActivity_5cbb5f7080b3b" name="Initiating, Dt.inicio 16/05/2018">
10      <bpmn:outgoing>sequenceFlow_5cbb5f710a258</bpmn:outgoing>
11      <bpmn:incoming>sequenceFlow_5cbb5f710a246</bpmn:incoming>
12      <bpmn:sequenceFlow id="sequenceFlow_5cbb5f710a2a0" sourceRef="StartEvent_5cbb5f70b95cd" targetRef="TaskActivity_5cbb5f7081a13"/>
13      <bpmn:sequenceFlow id="sequenceFlow_5cbb5f710a309" sourceRef="TaskActivity_5cbb5f7081a13" targetRef="TaskActivity_5cbb5f70873f3"/>
14      <bpmn:sequenceFlow id="sequenceFlow_5cbb5f710a7e1" sourceRef="TaskActivity_5cbb5f70873f3" targetRef="EndEvent_5cbb5f708dc57"/>
15      <bpmn:startEvent id="StartEvent_5cbb5f70b95cd" name="">
16        <bpmn:outgoing>sequenceFlow_5cbb5f710a2a0</bpmn:outgoing>
17      </bpmn:startEvent>
18    </bpmn:subProcess>
19  </bpmn:process>
20 </bpmn:definitions>
  
```

Fonte: elaborado pelo autor.

Posteriormente, tem-se a definição do elemento `<bpmndi:BPMNDiagram>`. O diagrama para representação ou BPMN Diagram Interchange, tem por objetivo definir como representar as formas existentes no nodo `<bpmn:process>`. Desta maneira, os elementos para representação na tela como `<bpmndi:BPMNEdge>` e `<bpmndi:BPMNShape>`, estão envolvidos por um nodo `<bpmndi:BPMNPlane>` e indicam a representação visual do modelo, como indicado no Quadro 3. Todavia, a representação não se preocupa na definição de cores e estilos, isso fica a critério do software de terceiros, interagir com o modelo. O nodo `<bpmndi:BPMNShape>` representa na tela os elementos básicos, como tarefas e subprocessos. O atributo `bpmnElement` é utilizado para identificar o elemento localizado no nodo `<bpmn:process>` e delimita os traços do elemento com o nodo `<Bounds>` que contém os atributos altura, largura e a posição x e y, como mostra o Quadro 3. Definido como os elementos da tela estão representados, é necessário representar o relacionamento entre os elementos e a direção dos elementos com o nodo `<bpmndi:BPMNEdge>` (OMG, 2011, p. 375). Este elemento é uma representação de uma seta indicadora de fluxo, contendo dois nodos filhos `<di:waypoint>` que possuem os atributos x e y. Deve-se definir dois nodos `<di:waypoint>` pois o primeiro indica a origem e a segunda o destino, como mostra o Quadro 3. Tendo respeitado tais definições, considera-se possível criar um arquivo BPMN válido para representação.

Quadro 3 – BPMN Diagram Interchange

```

94 // ...
95 <bpmn:process>
96 <bpmndi:BPMNDiagram id="BpmnDiagram 1">
97   <bpmndi:BPMNPlane bpmnElement="Process 1" id="BpmnPlane 1">
98     <bpmndi:BPMNShape bpmnElement="StartEvent 5cbb5f710a178" id="StartEvent_5cbb5f710a178_di">
99       <dc:Bounds height="36" width="36" x="300" y="25"/>
100     </bpmndi:BPMNShape>
101     <bpmndi:BPMNEdge bpmnElement="sequenceFlow_5cbb5f710a1d9" id="sequenceFlow_5cbb5f710a1d9_di">
102       <di:waypoint x="318" y="61"/>
103       <di:waypoint x="318" y="86"/>
104     </bpmndi:BPMNEdge>
105 // ...

```

Fonte: elaborado pelo autor.

## 2.4 BIBLIOTECA BPMN.IO

Conforme Rohten (2016), há várias ferramentas disponíveis no mercado que permitem a modelagem na notação BPMN, tais como: jBPM, Makesense, Alfresco, Camunda, Bpmn.io, Bonita BPM. Todavia, como o objetivo é representar a modelagem em ambiente web, foi escolhido a ferramenta Bpmn.io.

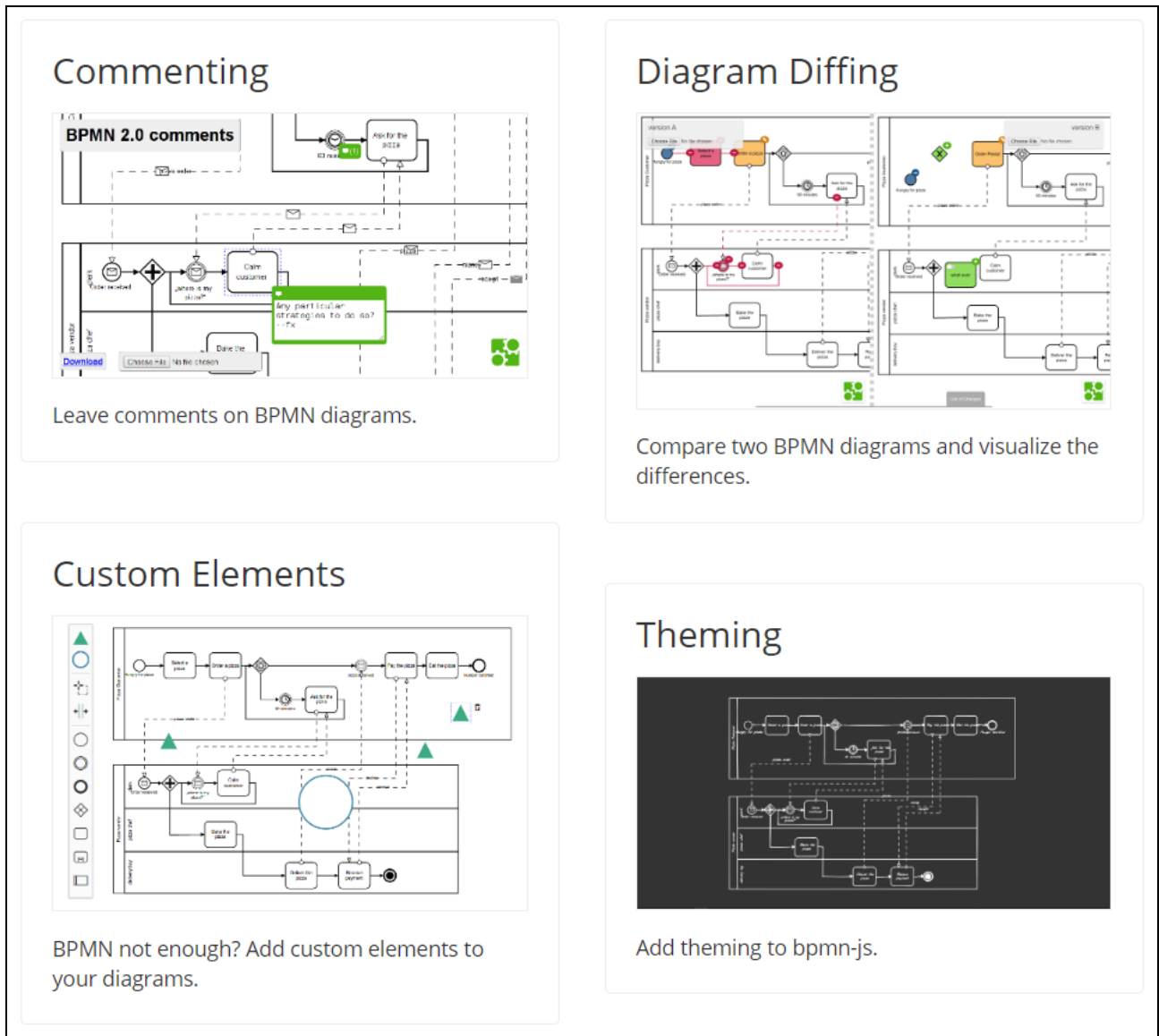
A biblioteca Bpmn.io consiste em um conjunto de ferramentas e pacotes para renderização e edição de documentos BPMN, CMMN (Case Management Model and Notation) e DMN (Decision Model and Notation). Ela é escrita em Javascript, modularizada e não necessita de servidor *back-end* para funcionar, o que a torna flexível para ser incorporada em qualquer projeto (BPMN.IO, 2019). Uma vez que este trabalho apenas se utiliza do pacote BPMN, nos próximos parágrafos são abordados sobre a arquitetura do pacote Bpmn-js.

O pacote Bpmn-js baseia-se em duas bibliotecas: Bpmn-moddle e Diagram-js (BPMN.IO, 2019). A biblioteca Bpmn-moddle conhece acerca dos padrões do XML da BPMN 2.0. Ela permite ler e escrever documentos em formato compatível com o padrão da BPMN e acessar informações por trás de formas relacionadas a representação no diagrama. Em essência a biblioteca oferece uma interface simplificada para validação do esquema BPMN na forma de duas funções: `fromXML`, cria a árvore BPMN como um objeto Javascript dado uma `string XML`, `toXML`, cria uma `string XML` a partir de um objeto Javascript. Posteriormente, a biblioteca Diagram-js tem como responsabilidade garantir a interação do usuário com o modelo e representar as formas da árvore Javascript da BPMN consumida pelo método `fromXML` do pacote Bpmn-moddle.

A biblioteca é extensível, de tal forma que é possível escrever *plug-ins* e disponibilizar o código separadamente da biblioteca. Na Figura 3 são ilustrados alguns exemplos de extensão da biblioteca, são eles:

- commenting*, responsável por permitir inserir comentários dinâmicos e sobrepostos aos elementos da tela. Facilitando a confecção do modelo, permitindo que antes que uma alteração seja feita do documento, mais usuários possam apreciar a alteração e demonstrar seu parecer;
- diagram diffing*, responsável por comparar dois arquivos BPMN e mostrar a diferença entre ambos;
- custom elements*, responsável por permitir, disponibilizar novos elementos ao modelador;
- theming*, responsável por permitir alterar os estilos padrão da biblioteca.

Figura 3 – Exemplos de extensão da biblioteca



Fonte: Adaptado de Bpmn.io (2019).

## 2.5 TRABALHOS CORRELATOS

Nesta seção são apresentados os trabalhos correlatos ao projeto proposto. São eles: o software Advantage Microsoft Project Plug-in (2018) o qual realiza a operação reversa à proposta deste trabalho, como mostra o Quadro 4; a dissertação de mestrado de Moura (2017) que gera cenários de teste a partir de modelos BPMN, conforme mostra o Quadro 5; e o trabalho de conclusão de curso de Souza (2013) que permite ao usuário gestor, criar fluxos de processo de negócio através de uma interface web, conforme é apresentado no Quadro 6.

Quadro 4 – Advantage Microsoft Project Plug-in

Referência	Active Modeler (2018)
Objetivos	Permitir exportação de processos BPMN para tarefas no MS-Project.
Principais funcionalidades	A ferramenta possibilita que os diagramas criados pelo modelador ActiveModeler sejam exportados para o MS-Project. Após o mapeamento ser feito, é possível visualizar todos os processos e subprocessos do diagrama listados no MS-Project como tarefas. A ferramenta não permite acesso ou edição online. O programa é capaz de exportar as seguintes informações: tarefas, dependências de tarefas, coordenadas do modelo, duração, notas e resumo de tarefas. Concluída exportação, tais informações encontram-se localizadas no MS-Project, com um gráfico de Gantt selecionado, após clicar com botão direito no menu, na opção Task Information. Para cada tarefa exportada os seguintes campos são preenchidos: WBS (Work breakdown Structure) code, que é utilizado para demarcar a indentação das tarefas, nome, custo

	e duração.
Ferramentas de desenvolvimento	ActiveModeler
Resultados e conclusões	A aplicação cria um arquivo MS-Project composto por todas as tarefas encontradas no arquivo BPMN, de acordo com o autor.

Fonte: elaborado pelo autor.

Quadro 5 – Dissertação de Mestrado de Jéssica Lasch de Moura

Referência	Moura (2017)
Objetivos	Gerar código de cenários de teste automatizados para Sistemas de Gerenciamento de Processos de Negócio (BPMS).
Principais funcionalidades	O trabalho de Moura (2017), propõe gerar códigos de cenários para executar testes automatizados de aplicações Web, implementadas com o apoio de um BPMS, a partir de modelos BPMN, visando abreviar o esforço de construção de elementos para teste automatizado.
Ferramentas de desenvolvimento	Cucumber, Lettuce, JDave e Selenium.
Resultados e conclusões	A aplicação cria uma tabela de caminhos de execução a partir da BPMN e gera código de cenários para os testes, utilizando os fluxos obtidos como entrada, a serem executados utilizando as ferramentas de desenvolvimento de teste, de acordo com o autor.

Fonte: elaborado pelo autor.

Quadro 6 – Trabalho de conclusão de curso de Eduardo Leopoldino de Souza

Referência	Souza (2013)
Objetivos	O software permite a automatização de processos de negócio em soluções de gestão de incidentes utilizando BPMN.
Principais funcionalidades	O trabalho de Souza (2013) dispõe para o usuário um editor gráfico de modelos de processo que possibilita a simulação de modelos criados pelo usuário e sua posterior publicação e execução em ambiente de produção.
Ferramentas de desenvolvimento	GoJS e Objectflow
Resultados e conclusões	O objetivo do trabalho foi atingido, utilizando o sistema 0800net como base para gerenciamento dos fluxos o editor gráfico. O trabalho limitou-se a representar e executar os conceitos básicos de <i>workflows</i> , de acordo com o autor.

Fonte: elaborado pelo autor.

### 3 DESCRIÇÃO DO PROTÓTIPO

A seguir são apresentados os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF), especificação, descrição de criação do arquivo, arquitetura, entre outros aspectos técnicos relevantes na concepção do projeto.

#### 3.1 REQUISITOS

No Quadro 7 são descritos os Requisitos Funcionais (RF) e no Quadro 8 são descritos os Requisitos Não Funcionais (RNF).

Quadro 7 – Requisitos Funcionais

RF	Requisito Funcional
RF01	permitir usuário fazer <i>login</i> na aplicação;
RF02	apresentar ao usuário seus arquivos já enviados;
RF03	permitir ao usuário enviar arquivo MS-Project para gerar o mapeamento;
RF04	permitir ao usuário visualizar individualmente cada processo BPMN gerado;
RF05	permitir ao usuário manipular individualmente cada processo BPMN gerado;
RF06	permitir ao usuário salvar suas alterações em arquivo BPMN ou SVG;
RF07	Permitir ao usuário fazer <i>logout</i> da aplicação.

Fonte: elaborado pelo autor.

Quadro 8 – Requisitos Não Funcionais

RNF	Requisito Não Funcional
RNF01	disponibilizar aplicação HTML5 para manipular a BPMN resultante;
RNF02	servidor web deverá ser implementado na linguagem PHP respeitando os padrões de codificação como SOLID e interoperabilidade, de acordo com a documentação disposta em PHP-FIG para que a aplicação possa ser disposta em componentes;

RNF03	criar testes de software automatizados para garantir a qualidade;
RNF04	utilizar ambiente PhpStorm para desenvolvimento da solução;
RNF05	Usar sistema de versionamento e controle de código Git.

Fonte: elaborado pelo autor.

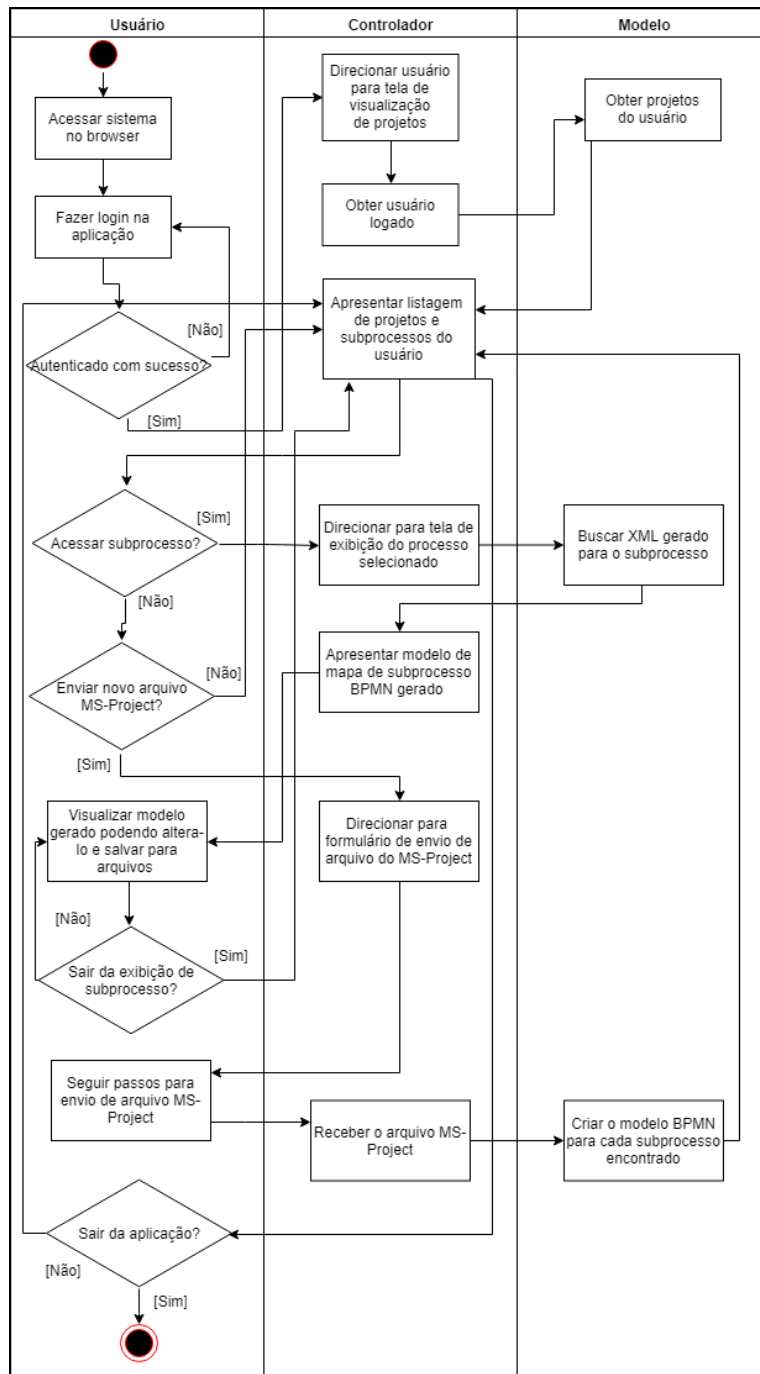
### 3.2 ESPECIFICAÇÃO

Nesta seção são apresentados os diagramas de atividade e diagrama de classes elaborados com as ferramentas.

#### 3.2.1 DIAGRAMA DE ATIVIDADES

Esta seção apresenta na Figura 4 o diagrama de atividades do sistema, que está dividido em três colunas: usuário, controlador e modelo. A coluna usuário, lista as ações disponíveis no *front-end* e as colunas controlador e modelo refletem as ações executadas na *back-end* da aplicação.

Figura 4 – Diagrama de atividades



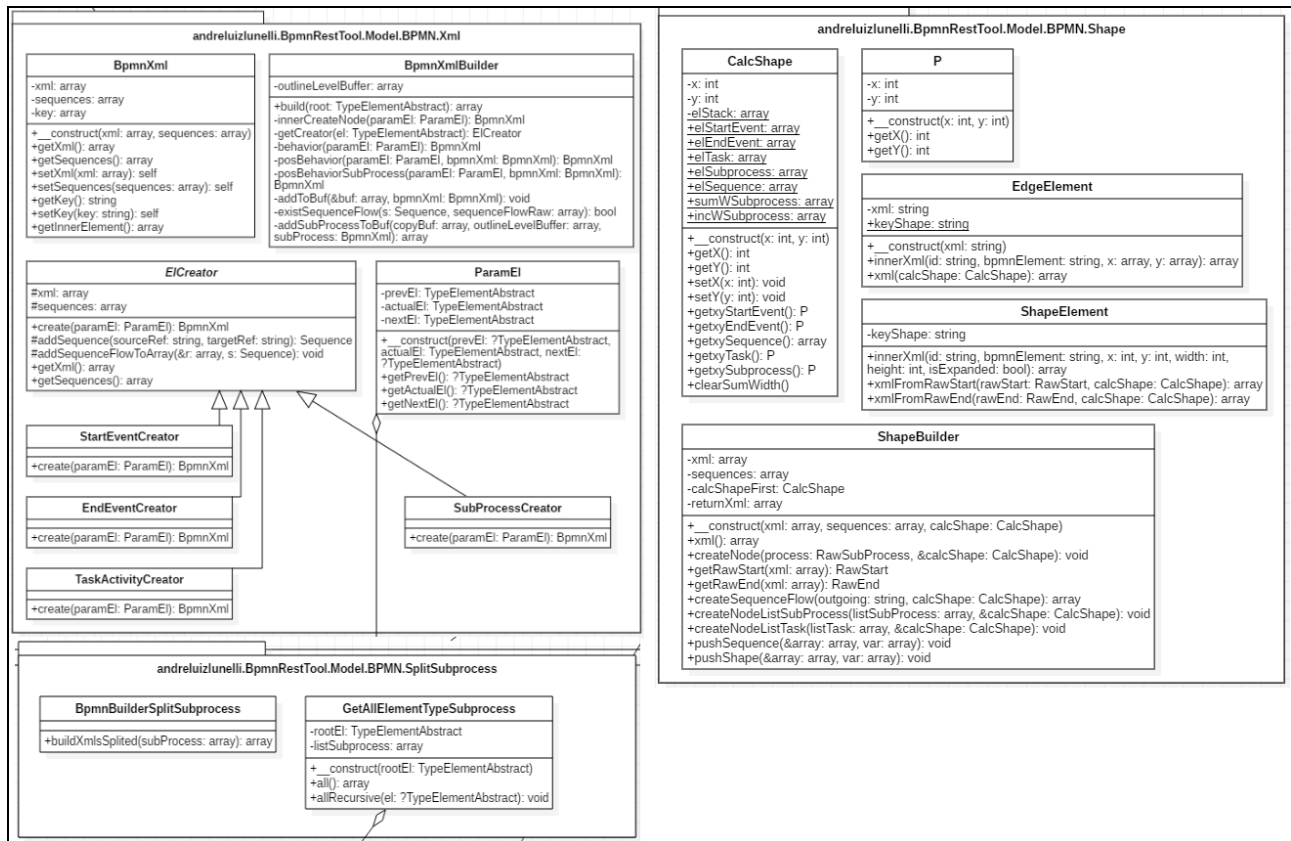
Fonte: Elaborado pelo autor.



### 3.2.2 DIAGRAMA DE CLASSES

Na Figura 5 é apresentado o pacote de fontes responsável por escrever no arquivo BPMN. Inicialmente o processo se dá com a classe `GetAllElementTypeSubprocess` que busca todos os subprocessos da árvore e passa para o método `buildXmlsSplited` da classe `BpmnBuilderSplitSubprocess` que gera um BPMN para cada subprocesso. Uma vez que o método `buildXmlsSplited` é chamado, continua-se o processo com a classe `BpmnBuilder`, que inicia uma nova instância de `Bpmnxmlbuilder` que é responsável por criar o XML de definições, o nodo `process`. Após a criação do XML de definições, usa-se a classe `ShapeBuilder` para criar o XML com definições de posicionamento na tela, calculando: largura, altura e posicionamento x e y.

Figura 5 – Diagrama de classes do projeto



Fonte: Elaborado pelo autor.

Detalhamento das classes:

- `BpmnXmlBuilder`: recebe um nodo `TypeElementAbstract` para gerar xml cru, tanto o xml do nodo quanto seus elementos de sequência;
- `BpmnXml`: objeto que detêm o retorno do xml cru criado pelo `ElCreator`;
- `ElCreator`: cria o nodo com base no tipo da instância, e chama sua classe especialista;
- `ParamEl`: utilizado para respeitar a indentação dos nodos anterior e próximo;
- `CalcShape`: acumulador responsável por marcar a largura e altura de um elemento;
- `EdgeElement`: cria o elemento `BPMNEdge`;
- `ShapeElement`: cria o elemento `BPMNShape`.

### 3.3 CRIAÇÃO DO ARQUIVO

Inicialmente o esforço da pesquisa foi para encontrar uma biblioteca em Php que interprete arquivos com extensão `.mpp` (Microsoft Project). Não encontrado a biblioteca que interprete `.mpp`, foi necessário definir como restrição a utilização do arquivo XML exportado do MS-Project.

Tendo mapeado os padrões de leitura de arquivo exportado do MS-Project e obtido a lista de tarefas contida no XML, foi possível assemelhar um padrão para criação do arquivo BPMN. Como restrição no protótipo é imposto que, um subprocesso não pode ser precedido ou sucedido por uma tarefa e vice-versa. De tal forma que:

- a lista de tarefas deve ser precedida por elemento de inicialização;
- a lista de tarefas deve ser sucedida por elemento de finalização;

- c) a lista de tarefas deve ser envolvida por um elemento de subprocesso;
- d) a lista de subprocessos deve ser antecedida por elemento de inicialização ou subprocesso;
- e) a lista de subprocessos deve ser sucedida por um elemento de finalização ou subprocesso.

O processo de criação do arquivo consiste em armazenar a lista de tarefas do MS-Project, criação de uma árvore binária onde é definido o metadado contendo dois nodos: um para o próximo elemento e um nodo indicando o elemento que contém um subprocesso. Para criar a árvore de metadados é necessário percorrer a lista armazenada de tarefas do MS-Project, verificando a propriedade `OutlineLevel` definida para cada tarefa do arquivo exportado como nível de estrutura. Cada tarefa lida é uma iteração e cada iteração armazena-se o nodo anterior, alimentando então os nodos da árvore. Desta maneira, as seguintes regras são aplicadas quando percorrido a lista de tarefas:

- a) o primeiro elemento da árvore deve ser um elemento de inicialização;
- b) se o `OutlineLevel` da tarefa atual menos o `OutlineLevel` da tarefa anterior for igual a 1, significa que a tarefa atual é um subprocesso da tarefa anterior;
- c) se o `OutlineLevel` da tarefa atual for igual ao do `OutlineLevel` anterior, adiciona-se uma nova tarefa em seqüência;
- d) se o `OutlineLevel` da tarefa atual menos o `OutlineLevel` da tarefa anterior for igual a -1, significa que a tarefa atual pertence ao subprocesso pai anterior;
- e) se o `OutlineLevel` da tarefa atual menos o `OutlineLevel` da tarefa anterior for menor que -1, deve-se executar a busca do `OutlineLevel` na árvore para adicionar o nodo na indentação correta.

A próxima etapa é montar o XML no padrão de modelagem BPMN. Esse padrão consiste em um arquivo com dois nodos principais para representar o diagrama: `process` e `BPMNDiagram`. Dentro do nodo `process` estão informados todos os elementos que serão desenhados na tela. Para cada elemento que compõe a tela, deve ser gerado um identificador único, o qual, consiste no nome do tipo do elemento, concatenado com uma *hash*, essa identificação é necessária para informar no `BPMNDiagram` a posição x e y do elemento na tela. A estrutura interna que compõe cada elemento desenhado na tela é definida por: identificador do elemento seguido pelos elementos `incoming`, `outgoing` e `sequenceFlow`.

Para visualização correta dos elementos foi necessário respeitar a indentação do XML dentro do nodo `process`. Cada subprocesso contém dentro de si no mínimo um elemento de inicialização, uma tarefa e um elemento de finalização. Esses elementos devem estar envolvidos dentro do XML do subprocesso, assim como todos os outros subprocessos dentro de um subprocesso devem estar envolvidos, essa indentação deve ser respeitada para que nenhum elemento seja sobreposto na tela. Para garantir essa indentação foi necessário no momento de criação do elemento saber o elemento anterior, elemento atual e o elemento seguinte. Tendo essas informações, torna-se possível respeitar a indentação do XML dentro do nodo `process`.

Como último passo do processo de criação do arquivo BPMN, foi necessário calcular as coordenadas x, y, largura e altura para cada elemento dentro do nodo `process` e informar dentro do nodo `BPMNDiagram`. Para efetuar esses cálculos definiu-se um tamanho fixo para largura e altura dos elementos de: inicialização, finalização, seqüência e tarefa. Cada elemento é processado dentro do contexto de um subprocesso e valores atuais de: x, y, largura e altura. Quando o contexto de um subprocesso é finalizado, os atributos de coordenadas são atualizados contendo o valor atualizado de largura e altura da soma de todos os elementos.

### 3.4 ARQUITETURA

Nesta seção é abordado o conjunto de camadas e tecnologias que compõem o software, bem como a utilização de uma arquitetura cliente-servidor. Como a maioria de aplicações web, o software é dividido em duas grandes partes: cliente e servidor, ou *front-end* e *back-end*. Essa arquitetura foi escolhida por alguns motivos: a biblioteca que articula a representação e modelagem da BPMN é escrita em JavaScript, o código que cria a BPMN é feito em PHP, o usuário não necessita de processo de instalação apenas acesso à web, facilitando o processo de teste e validação da ferramenta por outros usuários.

#### 3.4.1 FRONT-END

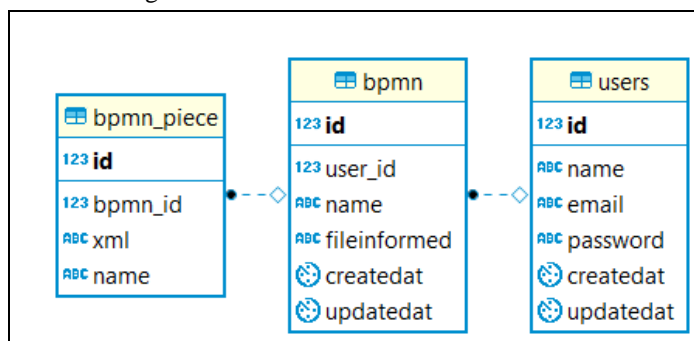
O *front-end* é a camada responsável por garantir a interação do usuário. A camada disponibiliza funções como: submeter arquivos MS-Project e manipular modelos BPMN, podendo alterar de alguma forma os processos, ordem ou adicionando novos elementos, como portas de entrada, eventos e setas. Também disponibiliza a opção de exportar BPMN, via download nas extensões, SVG ou BPMN.

O software é composto de várias tecnologias que funcionam em conjunto para garantir a usabilidade do sistema. Como destaque, a utilização do *framework* WebPack, utilizado para transpilar código de Typescript para Javascript, SASS para CSS e definição de dependências do projeto, como a biblioteca para articular a visualização da BPMN, Bpmn.io e a biblioteca Bootstrap, que disponibiliza os estilos e scripts para dinamização da tela.

### 3.4.2 BACK-END

O *back-end* é responsável por garantir a autorização e autenticação do sistema, receber arquivos MS-Project e gerar a BPMN resultante. Quando um arquivo é submetido ao servidor, é gerado um registro na tabela `bpmn` (Figura 6) do banco de dados, informando o nome do arquivo, usuário que submeteu e conteúdo do arquivo informado. A tabela `bpmn` é composta de outra tabela `bpmn_piece`, que detém todos os subprocessos que foram produzidos na árvore de subprocessos da BPMN, possibilitando assim o acesso do usuário a cada subprocesso individualmente.

Figura 6 – Modelo entidade relacionamento



Fonte: Elaborado pelo autor.

Toda interação com o banco de dados: criação, atualização de tabelas, seleção em tabela, criar, atualizar e excluir registros é executada a partir do Framework Doctrine 2 como ORM (Object Relational Mapper). Para armazenamento foi utilizado o banco PostgreSQL.

A arquitetura interna para fornecer acesso as páginas é criada com a biblioteca SlimFramework que fornece a estrutura para implementação de padrões de projeto como MVC e REST. Destas estruturas, destacam-se os padrões de Middleware e HTTPRouter que facilitaram a criação de rotas e autenticação no sistema.

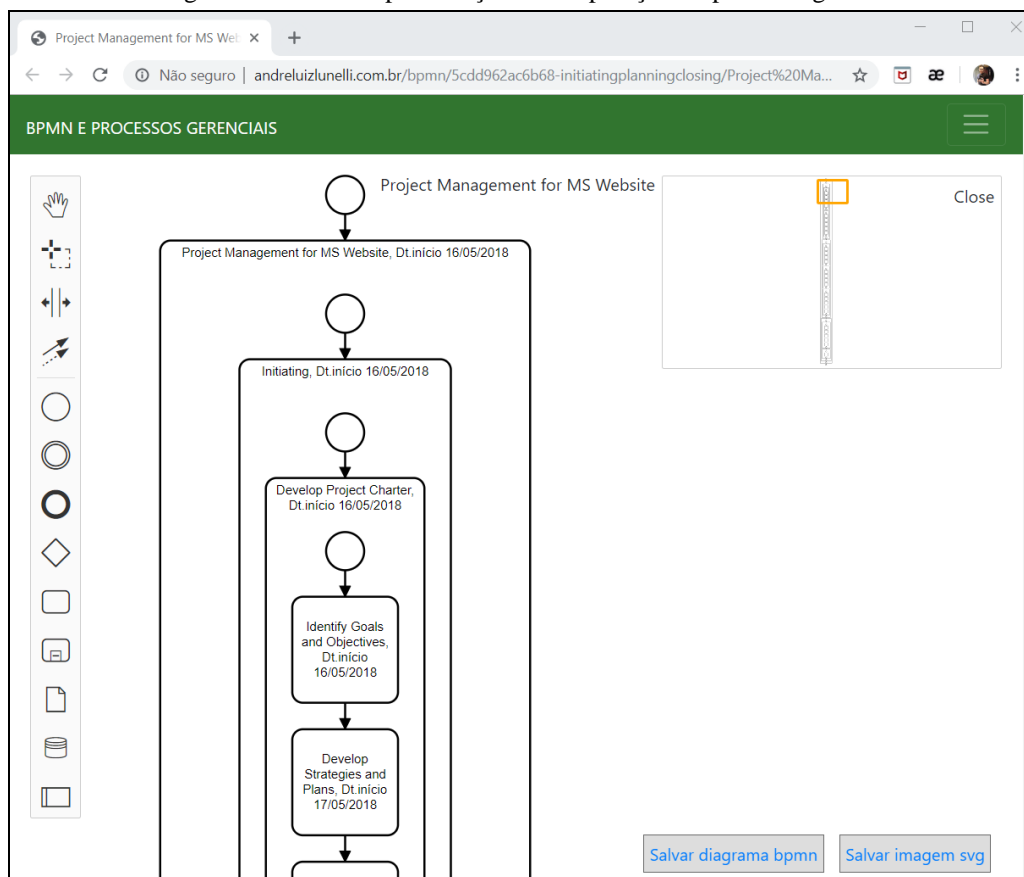
Toda estrutura de pastas e código no PHP é controlada por um gerenciador de dependências, Composer. Essa ferramenta para controle de dependências, também disponibiliza uma implementação para *autoload* de classes, a `Psr-4`, que é responsável por eximir a responsabilidade do programador fazer `include` de classes e trabalhar apenas com *namespaces*.

## 4 RESULTADOS

Através deste trabalho, tornou-se possível a análise de como a gerência de projetos pode estar formatada como processo. Apenas submetendo o arquivo exportado do MS-Project, cada tarefa encontrada é apresentada como fluxo em forma de subprocesso BPMN e pode ter seu conteúdo alterado utilizando a paleta de elementos. Existe a possibilidade de incrementar o fluxo com outros elementos de tarefas, condicionais, *swim lanes* e outros elementos de conformidade ao padrão BPMN, podendo navegar pelo fluxo arrastando o diagrama ou utilizando o mapa. Posteriormente, no fim da sua utilização, a opção de salvar como BPMN ou SVG, como mostra a Figura 7.

O sistema por ter sido idealizado para plataforma web, permite que todo arquivo enviado e BPMN gerado, possa ser manuseado apenas pelo usuário que originou o registro na base de dados. Esse aspecto, assim como o relacionamento da base, classifica o software como um sistema *Multi Tenant*, pois uma única instância é capaz de lidar com vários clientes (MANDUCA, 2013), conforme mostra a Figura 8.

Figura 7 – Tela de representação e manipulação do processo gerado



Fonte: elaborado pelo autor.

Figura 8 – Tela de login da aplicação

Fonte: Elaborado pelo autor.

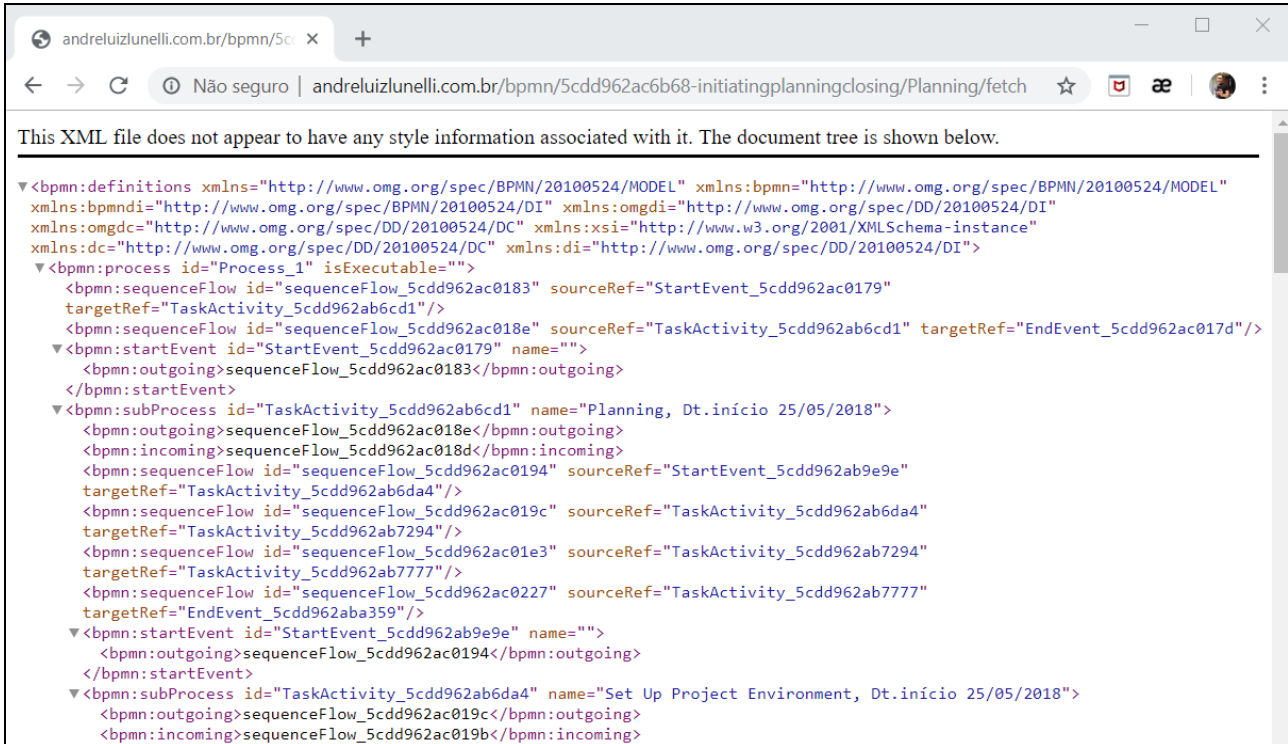
Como resultado secundário, o projeto viabilizou a disponibilização de um conjunto de classes, que podem ser utilizadas por aplicativos de terceiros, para obterem informações a partir de um arquivo MS-Project, tais como:

- árvore de elementos e lista de subprocessos;
- funcionalidades de consumo de um arquivo de MS-Project com a classe `ProjectMapper` que encapsula a biblioteca `QueryPath`, possibilitando resgatar todas as informações no arquivo, incluindo título do arquivo, nome e a listagem de todas as tarefas;
- criação de um metadado, representado por uma árvore binária com a classe `BpmnMetadataBuilder` que tem como parâmetro o resultado obtido com a classe `ProjectMapper` por conter as informações relativas ao MS-Project. O metadado contém informações do tipo do nodo atual, se é um evento, uma tarefa ou um subprocesso. De modo que todo nodo está indentado dentro do nodo `outgoing`, e quando o nodo

subprocess existir a interpretação que se segue é que o subprocesso está dentro do fluxo anterior informado no nodo outgoing. Essa árvore também pode ser facilmente exportada para JSON;

- d) há a possibilidade de acessar ao XML gerado pelo protótipo, informando na url da página de visualização da BPMN ao acrescentar a string /fetch, no final da url como mostra a Figura 9.

Figura 9 – Solicitação de acesso ao XML gerado



Fonte: Elaborado pelo autor.

Após a finalização desta primeira versão do protótipo realizou-se uma pesquisa com alguns usuários que se dispuseram a dedicar um tempo para avaliar o software. O Quadro 9 apresenta as considerações destes usuários.

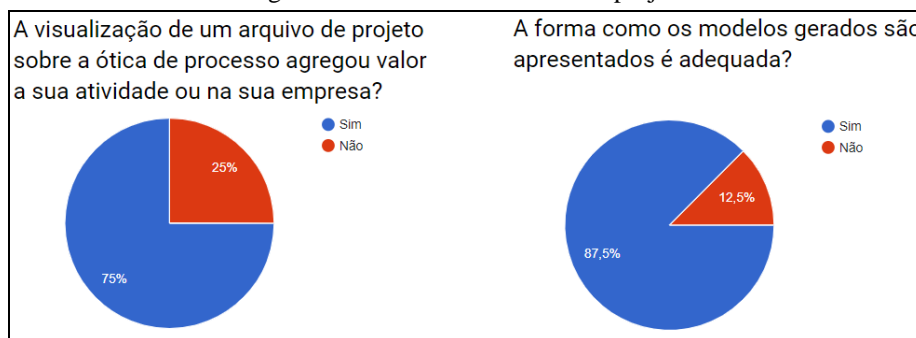
Quadro 9 – Sugestões para evolução da ferramenta

Trazer a porcentagem relacionada ao tempo de entrega do projeto.
Colocar no início da plataforma um modelo de projeto.
Bem legal a ideia, tudo que vier pra facilitar na transmissão da informação é bem-vindo!
Está perfeitamente funcional. Os modelos apresentados são claros e de fácil manipulação. A performance é perfeitamente adequada para o que se pretende. O mapa lateral fornece uma visão holística do projeto e é possível identificar facilmente qual etapa do processo em relação ao todo está sendo analisado. Apesar de ser uma aplicação implementada para um público bem específico, desconhecia no mercado, soluções que fornecessem o desenho de um projeto, em layout BPMN, o que faz dele um diferencial. É notório também e necessário destacar a preocupação do desenvolvedor com o usuário conseguir usar sozinho a aplicação. Excelente trabalho!
Eu recomendaria a melhoria na parte visual do app web e um retorno mais amigável em caso de estourar um erro, por exemplo, enviar um e-mail ou Whatsapp para o autor do trabalho.

Fonte: elaborado pelo autor.

Embora a amostragem de avaliadores tenha sido pequena, as estatísticas obtidas nos levam a crer que o projeto tem perspectiva de futuro pois: 75% dos usuários consideraram o diagrama de processos fornecido relevante para seu cenário de trabalho e 87,5% consideraram o modelo apresentado como adequado, conforme mostra Figura 10.

Figura 10 – Gráfico relevância do projeto



Fonte: Elaborado pelo autor.

## 5 CONCLUSÕES

Este artigo apresentou o desenvolvimento de uma ferramenta que a partir de um arquivo de MS-Project gera um modelo BPMN com base nas tarefas obtidas do arquivo. Esta ferramenta mostrou-se promissora pois auxilia na análise de como o usuário ou organização, lida com os projetos e onde pode ser mitigado esforço.

Após a idealização da ferramenta, foi empregado esforço em como cada elemento BPMN poderia ser representado dentro de um contexto de MS-Project, porém alguns casos, tais como elementos condicionais, não puderam ser atendidos pois o MS-Project não dispõe de tal característica. Uma limitação do protótipo é que ele não é capaz de importar um projeto que possua mais de 2000 tarefas tendo em vista que nesta situação há um erro de vazamento de memória.

Considera-se que os objetivos do projeto foram alcançados e podem ser acessados a partir do link: <http://andreluizlunelli.com.br>. A ferramenta foi desenvolvida utilizando o ambiente PhpStorm, a linguagem PHP 7.3, Apache 2.4 e banco de dados PostgreSQL 9.6. O código fonte da aplicação está disponível em <https://github.com/andreluizlunelli/bpmn-rest-tool>.

Como extensões ao trabalho propõe-se:

- exportar para ferramentas como Bizagi e Visio;
- suportar envio de arquivos grandes do MS-Project, com mais de 2000 tarefas;
- implementar funcionalidade de ações condicionais junto aos campos customizados disponibilizados pelo MS-Project, que devem ser predefinidos antes e indicados pelo usuário antes da submissão do arquivo;
- o trabalho desenvolvido se mostrou coeso e extensível o que permite configurá-lo para operar em um modelo FaaS (Function as a Service).

## REFERÊNCIAS

- ACTIVE MODELER. **ActiveModeler.com - BPMN based BPM and Workflow system**. 2019. Disponível em: <<http://www.activemodeler.com/MS-Project.htm>>. Acesso em: 17 mai. 2018.
- ANTONI, Marc; FORSBERG, Torbjörn; KAMMERLIND, Peter. **Process management in a project environment - the relation between different roles.**: Paper presented at Project Management Institute Annual Seminars & Symposium. 2000. Disponível em: <<https://www.pmi.org/learning/library/process-management-project-environment-relation-1879>>. Acesso em: 27 jun. 2019.
- BALDAM, Roquemar; VALLE, Rogerio; ROZENFELD, Henrique. **Gerenciamento de processos de negócio BPM: Uma referência para implantação prática**. Rio de Janeiro: Elsevier Editora Ltda., 2014. 777 p.
- BASSO, Raquel Mainardi Pillat. **BPMNT: A Proposal for Flexible Process Tailoring Representation in BPMN**. 2018. 186 f. Tese (Doutorado) - Curso de Engenharia de Software, Engenharia de Sistemas e Computação, Instituto Alberto Luiz Coimbra de Pós-graduação e Pesquisa de Engenharia, Rio de Janeiro, 2018. Disponível em: <<https://www.cos.ufrj.br/uploadfile/publicacao/2825.pdf>>. Acesso em: 6 jun. 2019.
- BPMN.IO (Comp.). **Walkthrough | bpmn.io**. 2019. Disponível em: <<https://bpmn.io/toolkit/bpmn-js/walkthrough/>>. Acesso em: 24 maio 2019.
- BPMN.IO. **Bpmn-js: Explore examples that use, modify and extend bpmn-js.** 2019. Disponível em: <<https://bpmn.io/toolkit/bpmn-js/examples/>>. Acesso em: 29 jun. 2019.
- CARVALHO, Izabella Antonioli de; NOTARI, Luiza Meirelles Ninhaus. **Gerenciamento de projetos: aplicação do planejamento no setor de óleo e gás**. 2017. 120 f. TCC (Graduação) - Curso de Engenharia Química, Engenharia Química e de Petróleo, Universidade Federal Fluminense, Niterói, 2017. Disponível em: <<https://app.uff.br/riuff/handle/1/5544>>. Acesso em: 6 jun. 2019.

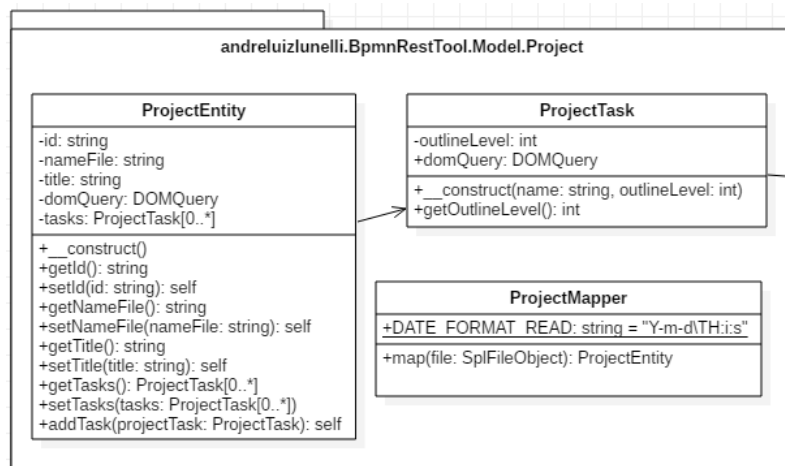
- CATELLI, Armando; SANTOS, Edilene Santana. Mensurando a criação de valor na gestão pública. **Revista de Administração Pública**, Rio de Janeiro, v. 38, n. 3, p. 423 a 450, jan. 2004. ISSN 1982-3134. Disponível em <<http://bibliotecadigital.fgv.br/ojs/index.php/rap/article/view/6544/5128>>. Acesso em: 12 mai.19.
- FLORES, Camilo; SEPÚLVEDA, Marcos. Temporal Specification of Business Processes through Project Planning Tools. **Springer for Research & Development**, [S.L.], v. 66, p.85-86, 2011.
- GIROTTO, Alysson Nathan. **Uma proposta para derivar Casos de Uso a partir de modelos BPMN com suporte computacional**. 2016. 167 f. TCC (Graduação) - Curso de Bacharelado em Ciência da Computação, Centro Tecnológico, Unioeste - Universidade Estadual do Oeste do Paraná, Cascavel, 2016. Disponível em: <<http://inf.unioeste.br/~tcc/2016/TCC%20-%20Alysson%20Nathan%20Giroto.pdf>>. Acesso em: 27 jun. 2019.
- KOSTELAC, Drazen; MATRJAN, Damir; DOBOVICEK, Sandro. Relationship between Processes and Project Management. **DAAAM International vienna**, Vienna, v. 22, n. 1, p. 1119-1120, out. 2011. Disponível em: <[https://bib.irb.hr/datoteka/642224.RELATIONSHIP\\_BETWEEN\\_PROCESSES\\_AND\\_PROJECT\\_MANAGEMENT.pdf](https://bib.irb.hr/datoteka/642224.RELATIONSHIP_BETWEEN_PROCESSES_AND_PROJECT_MANAGEMENT.pdf)>. Acesso em: 12 mai.19.
- LUCID SOFTWARE INC. **Símbolos e notação de diagramas BPMN**. 2019. Disponível em: <<https://www.lucidchart.com/pages/pt/simbolos-e-notacao-de-diagramas-bpmn>>. Acesso em: 27 jun. 2019.
- MANDUCA, Alexandre Michetti. **Transforme aplicações web em serviços Multi Tenant**. 2013. Disponível em: <<https://www.devmedia.com.br/transforme-aplicacoes-web-em-servicos-multi-tenant/28881>>. Acesso em: 31 maio 2019.
- MICROSOFT. **Microsoft Project**. 2019. Disponível em: <<https://products.office.com/pt-br/project/project-and-portfolio-management-software>>. Acesso em: 31 maio 2019.
- MICROSOFT Project Office 365. [S. l.], 2019. 1 CD-ROM.
- MOURA, Jéssica Lasch de. **Geração Automática de Códigos de Cenários de Testes de Aplicações de Gerenciamento de Processos de Negócio a partir de Modelos BPMN**. 2017. 66 f. Dissertação (Mestrado) – Curso de Ciência da Computação, Centro Tecnológico, Universidade Federal de Santa Maria, RS, 2017.
- OMG. **FORMAL/2011-01-03: Business Process Model and Notation**. 2.0 ed. [s. L.], 2011. 508 p. Disponível em: <<https://www.omg.org/spec/BPMN/2.0/PDF>>. Acesso em: 20 maio 2019.
- PLANTUML in a nutshell. 2019. Disponível em: <<http://plantuml.com/>>. Acesso em: 31 maio 2019.
- PMI (Eua) (Org.). **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK®)**. 6. ed. Newtown Square, Pennsylvania: Project Management Institute, Inc., 2017. 555 p.
- ROHTEN, Luis Augusto Melo. **Integrando a Internet das Coisas às Aplicações: uma Modelagem de Aplicação WSN Orientada a Serviço e Orquestrada em BPMN**. 2016. 58 f. TCC (Graduação) - Curso de Ciência da Computação, Centro Tecnológico, Universidade Federal do Espírito Santo, Vitória, 2016. Disponível em: <<http://www.multicast.com.br/ucb/projeto-integrado/artigos-exemplo/tcc-luis-rohten.pdf>>. Acesso em: 27 jun. 2019.
- SANTOS, Angela Maria dos. **Gerenciamento de Tempo de Projeto - Implantação do Software Erp SAP: Utilização do MS-Project**. 2013. 50 f. Dissertação (Mestrado) - Curso de Mba em Gerência de Sistemas Logísticos, Universidade Federal do Paraná, Curitiba, 2013. Disponível em: <<https://hdl.handle.net/1884/48520>>. Acesso em: 4 jun. 2019.
- SOTILLE, Mauro. **Estatísticas sobre o PMI**. 2019. Disponível em: <<https://blog.pmtech.com.br/dados-estatisticos/>>. Acesso em: 27 jun. 2019.
- SOUZA, Eduardo Leopoldino de. **Automatização de Processos em Aplicações de Gestão de Incidentes Utilizando BPMN**. 2013. 60 f. TCC (Graduação) - Curso de Sistemas de Informação, Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2013. Disponível em: <<http://dsc.inf.furb.br/arquivos/tccs/monografias/TCC2013-1-06-VF-EduardoLdeSouza.pdf>>. Acesso em: 29 maio 2019.
- UMLWRITER. **UmlWriter: Create UML class diagrams from your PHP source.** 2019. Disponível em: <<http://php5.laurent-laville.org/umlwriter/>>. Acesso em: 31 maio 2019.
- VARGAS, Ricardo. **Manual prático do plano de projeto utilizando o PMBOK guide**. Rio de Janeiro: Brasport, 2007. 256 p.

## APÊNDICE A – DIAGRAMAS DE ESPECIFICAÇÃO

Nesta seção serão apresentados os pacotes usados para consumir o Project: suas tarefas e atributos em XML. Da mesma forma serão apresentados também os pacotes relacionados a criação do arquivo BPMN: sua árvore de elementos e representação XML em conformidade com o padrão BPMN. Para criação dos diagramas foi utilizado a ferramenta PlantUML (2018) que cria a representação gráfica dos pacotes e classes, em conjunto com a biblioteca UmlWriter (2018), que cria a definição consumida pelo PlantUML, para criação da representação gráfica.

Na Figura 11 é apresentado, o pacote responsável por obter as tarefas do Project e seus atributos em uma lista de tarefas. Posteriormente a instância de `ProjectEntity` que contém as informações do Project e suas tarefas, é usada como parâmetro para classe `BpmnMetadataBuilder`.

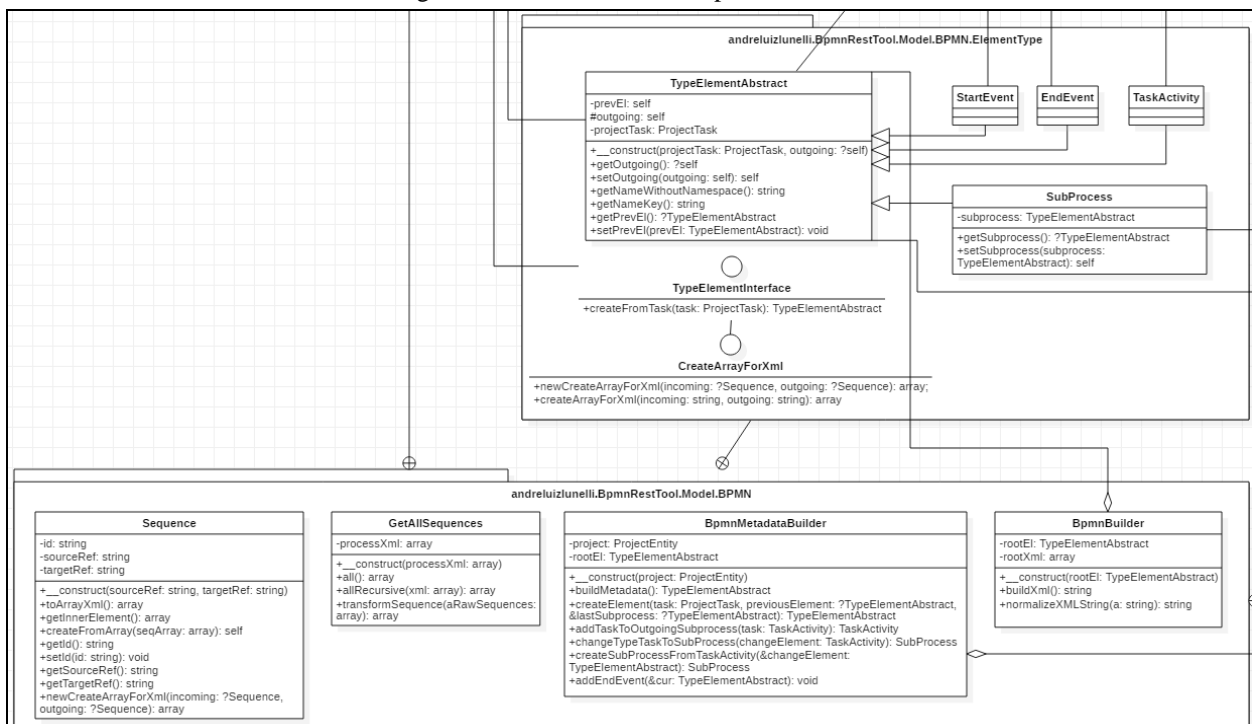
Figura 11 – Fontes do pacote Project



Fonte: Elaborado pelo autor.

Na Figura 12 é apresentado as classes de domínio no pacote raiz BPMN. Que contém a lógica para criação dos metadados da árvore de elementos, com a classe `BpmnMetadataBuilder`, utilizando como base os dados do Project informados pela classe `ProjectEntity`. A árvore de metadados é formada por nodos do elemento `TypeElementAbstract`.

Figura 12 – Fontes da raiz do pacote BPMN



Fonte: Elaborado pelo autor