UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DJEISON AGUIR SELZLEIN

DESENVOLVIMENTO DE APLICAÇÕES BPM ORIENTADO A MODELOS UTILIZANDO-SE NOTAÇÃO BPMN

TRABALHO DE DIPLOMAÇÃO

MEDIANEIRA
2015

DJEISON AGUIR SELZLEIN

DESENVOLVIMENTO DE APLICAÇÕES BPM ORIENTADO A MODELOS UTILIZANDO-SE NOTAÇÃO BPMN

Trabalho de Diplomação apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas – COADS – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Everton Coimbra de Araújo.

MEDIANEIRA

2015

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Diretoria de Graduação e Educação Profissional
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

TERMO DE APROVAÇÃO

Desenvolvimento de aplicações BPM orientado a modelos utilizando-se notação BPMN

Por

Djeison Aguir Selzlein

Este Trabalho de Diplomação (TD) foi apresentado às 09:10 h do dia 19 de novembro de 2015 como requisito parcial para a obtenção do título de Tecnólogo no Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, *Campus* Medianeira. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Everton Coimbra de Araújo
UTFPR – *Campus* Medianeira
(Orientador)

Prof. M. Sc. Ricardo Sobjak
UTFPR – *Campus* Medianeira
(Convidado)

Prof. Marcio Angelo Matte
UTFPR – *Campus* Medianeira
(Convidado)

Prof. M. Sc. Juliano Rodrigo Lamb
UTFPR – *Campus* Medianeira
(Responsável pelas atividades de TCC)

A folha de aprovação assinada encontra-se na Coordenação do Curso.

RESUMO

SELZLEIN, Djeison Aguir. DESENVOLVIMENTO DE APLICAÇÕES BPM ORIENTADO A MODELOS UTILIZANDO-SE NOTAÇÃO BPMN. Trabalho de Diplomação (Tecnologia em Análise e Desenvolvimento de Sistemas). Universidade Tecnológica Federal do Paraná. Medianeira 2015.

Assim como o processo de desenvolvimento de software tem se tornado moroso, exigindo grande esforço por parte de equipes de desenvolvimento, também os processos de negócio desempenhados nas mais diversas organizações se tornam complexos conforme essas organizações crescem. BPM ou *Business Process Modeling* é a atividade que permite o gerenciamento e modelagem de processos de negócio. Ferramentas para desenvolvimento de aplicações BPM que utilizem uma abordagem orientada a modelos, tais como o LSPS, proporcionam uma redução na complexidade no processo de desenvolvimento de software e também a possibilidade de otimização, execução e monitoramento de processos de negócio. O presente trabalho teve como objetivo apresentar os conceitos implementados por essas ferramentas, focando especificamente o LSPS, e demonstrar a utilização deste no processo de desenvolvimento de uma aplicação demonstrativa. Após desenvolvimento deste trabalho, verificam-se os benefícios da utilização de ferramentas BPM, tais como controle e monitoramento de processos de negocio e agilidade no processo de modelagem.

Palavras-Chave: Desenvolvimento Orientado a Modelos, Aplicações BPM, LSPS, Linguagem de Modelagem GO-BPMN

ABSTRACT

SELZLEIN, Djeison Aguir. MODEL-DRIVEN BPM APPLICATIONS' DEVELOPMENT USING BPMN NOTATION. Trabalho de Diplomação (Tecnologia em Análise e Desenvolvimento de Sistemas). Universidade Tecnológica Federal do Paraná. Medianeira 2015.

Just like software development process has become cumbersome demanding huge effort from development teams, also business processes performed in several different organizations become complex as organizations grow. BPM or Business Process Modeling is the activity that allows business processes management and modeling. Tools for BPM applications development, which make use of an model-driven approach such as LSPS, provide a complexity reduction in the process of software development as well as the possibility of business processes optimization, execution, and monitoring. The present paper aims to present the concepts related to these tools, specifically LSPS. This paper also demonstrates usage of LSPS in the process of development of a demo application. With the conclusion of this paper, it is clear the benefits of using BPM tools, such as business process control and monitoring, and process modeling agility.

Keywords: Model-driven Development, BPM Applications, LSPS, GO-BPMN Modeling Language

LISTA DE SIGLAS

BPM	*Business Process Management*
BPMN	*Business Process Model and Notation*
BPMS	*Business Process Management Suite*
COO	*Chief Operating Officer*
EJB	*Enterprise Java Beans*
EAR	*Enterprise Archive*
GO-BPMN	*Goal Oriented Business Process Modeling Notation*
IDE	*Integrated Development Environment*
JEE	*Java Platform, Enterprise Edition*
LSPS	*Living Systems Process Suite*
MDD	*Model Driven Development*
PDS	*Process Design Suite*
SOAP	*Simple Object Access Protocol*

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Empresas, de maneira geral, possuem atividades desempenhadas diariamente. Essas rotinas, também são conhecidas como processos de negócio e constituem a essência dessas empresas, uma vez que é desempenhando seus processos de negócio que produzem serviços ou bens. Toda empresa, independentemente do setor em que atua, possui processos de negócio.

BPM (*Business Process Management*) é a disciplina que envolve qualquer combinação de modelagem, automação, execução, controle, medição e optimização do fluxo de atividades de negócio, em suporte à metas de empresas, abrangendo sistemas, empregados, clientes e parceiros dentro e além das fronteiras da empresa (Palmer, 2015).

Definir modelos de negócios é o processo de identificação, definição e representação de processos de negócio visando suportar e promover a comunicação de forma clara sobre tal processo (Palmer, 2015). Modelagem de processos de negócios é uma das principais atividades relacionadas a BPM sendo que uma vez que se tenha um modelo definido, é possível analisá-lo para que sejam aplicadas medidas visando sua otimização.

Buscando a representação de modelos de negócio de maneira que todos os indivíduos envolvidos no processo de negócio pudessem compreendê-lo, desde analistas de negócio até os desenvolvedores de software que o implementam, a OMG (*Object Management Group*) criou o padrão BPMN (*Business Process Model and Notation*). Dessa maneira, BPMN cria uma ponte entre design do processo de negócio e sua implementação (Object Management Group, 2011).

Uma vez que BPM é a atividade de gerenciamento de processos de negócio e a notação BPMN é utilizada como linguagem para modelagem destes processos, é interessante que aplicações BPM sejam implementadas utilizando-se uma abordagem de desenvolvimento adequada, que tenha como foco os modelos criados.

Model Driven Development (MDD ou desenvolvimento orientado a modelos) é uma abordagem de desenvolvimento de software que tem como foco principal os modelos do sistema. Dessa maneira, o desenvolvimento de uma aplicação é orientado a seguir os modelos definidos previamente. Segundo Lima e Silva Júnior (2009), a principal vantagem do desenvolvimento orientado a modelos está em "representar as funcionalidades e/ou recursos do sistema auxiliando a tomada de decisões sobre quais as melhores ações devem ser aplicadas para atingir as metas do sistema".

Considerando os benefícios proporcionados pela abordagem de desenvolvimento orientado a modelos e analisando BPMN a partir da perspectiva que essa notação visa a definição e representação de modelos de negócio, o desafio torna-se integrar estas tecnologias de maneira a fazer uso das qualidades de ambas. A utilização de MDD tendo como linguagem para modelagem o BPMN permite a criação de aplicações BPM de alta complexidade, e fácil gerenciamento de alterações.

Living Systems Process Suite (LSPS) é uma suíte de ferramentas para o design, execução, monitoramento e manutenção de modelos de processos de negócio. LSPS tem como linguagem para criação de seus modelos a GO-BPMN (*Goal Oriented Business Process Modeling Notation* ou Notação para Modelagem de Processos de Negócio Orientada a Metas) (Whitestein Technologies AG, 2015).

GO-BPMN também permite a criação de modelos baseados em conceitos de metas a serem atingidas (definidas por objetivos ou *Goals*) e *como* atingi-las (definidos por planos ou *Plans*) (Whitestein Technologies AG, 2015).

De maneira geral, LSPS é uma ferramenta que integra desenvolvimento orientado a modelos e BPMN de maneira que analistas de negócio possam realizar BPM fazendo uso dessa plataforma. Neste ambiente, o usuário modela o processo de negócio utilizando a notação GO-BPMN, que estende BPMN e então a própria plataforma gera o código-fonte, na linguagem de programação Java, da aplicação BPM correspondente.

Este trabalho tem como objetivo expor o LSPS como um ambiente de desenvolvimento orientado a modelo que utiliza a notação BPMN para criação de aplicações BPM. Além disso, é objetivo também apresentar tal ferramenta e demonstrar o uso da mesma no desenvolvimento de uma aplicação exemplo sobre uma loja virtual, na qual o processo de negócio modelado represente o processamento de pedidos de produtos, realizados por clientes.

1.1 OBJETIVO GERAL

Demonstrar os benefícios da utilização de MDD tendo BPMN como notação no desenvolvimento de aplicações BPM. Como prova de conceito, desenvolver aplicação demonstrativa utilizando a ferramenta LSPS.

1.2 OBJETIVOS ESPECÍFICOS

Como etapas para conclusão do objetivo geral, os seguintes objetivos específicos são propostos:

- Desenvolver referencial teórico sobre MDD, BPMN e BPM, suas respectivas motivações, conceitos e objetivos;
- Elaborar embasamento sobre a ferramenta LSPS abordando a maneira como implementa os conceitos em estudo;
- Utilizar a ferramenta LSPS para implementação de aplicação BPM para loja virtual, criando modelos de processos de negócio que representem o processamento de pedidos e gerando-se interface de usuário;
- Avaliar os benefícios obtidos pela utilização do LSPS como plataforma para desenvolvimento de aplicações BPM.

1.3 JUSTIFICATIVA

Da mesma maneira que empresas crescem fisicamente, seus processos internos também evoluem e se tornam complexos. Os processos de negócio desempenhados em uma empresa constituem a essência desta. Portanto, se não há gerência e controle sobre os processos de negócio de uma empresa, esta mesma está suscetível a falhas e mal aproveitamento de recursos.

Neste contexto, BPM se apresenta como prática que permite modelagem, otimização, automação, controle e análise de processos de negócio auxiliando analistas de negócio (Palmer, 2015).

Assim como processos de negócio se tornam complexos, os requisitos de sistemas de informação também ganham complexidade. Segundo Lucrédio (2009), complexidade de software é “o esforço necessário para tanto (construção de software), utilizando tecnologias centradas em código-fonte, é muito grande”.

MDD ou desenvolvimento orientado a modelo surge como abordagem que se propõe a resolver os problemas de complexidade de software. A abordagem de desenvolvimento orientada a modelos traz as seguintes melhorias e vantagens (Lima e Silva Júnior, 2009):

- Produtividade – Geração automática de código-fonte;
- Portabilidade – Flexibilidade para transferência de aplicações de uma plataforma para outra;
- Qualidade – Considerando que a modelagem é independente de plataforma tecnológica, o foco principal da construção do sistema fica nas suas funcionalidades e recursos;
- Redução nos custos de desenvolvimento – Como o foco do desenvolvimento está nos modelos, quando há necessidade de alteração em alguma funcionalidade, apenas o modelo é modificado e o código-fonte é gerado automaticamente.

Considerando os benefícios proporcionados pelo MDD e a importância da prática de BPM, a ideia de uma ferramenta que permita a combinação destes dois conceitos, tendo como modelo o BPMN, se torna altamente interessante para a evolução da maneira como se escreve software.

1.4 ESTRUTURA DO TRABALHO

O presente trabalho é composto por cinco capítulos. No primeiro capítulo é realizada uma introdução sobre o assunto abordado neste trabalho. São expostos os objetivos do mesmo, a que se destina e por último uma justificativa.

O segundo capítulo objetiva uma explanação e obtenção de referencial teórico, apresentando os conceitos relacionados a BPM, BPMN e MDD, sua evolução e importância. Também foi realizada uma análise e conceituação sobre a plataforma LSPS. Essa análise visa expor as características dessa ferramenta e de que maneira ela implementa os conceitos de BPM, BPMN e MDD.

No terceiro capítulo são abordados os materiais e métodos utilizados no desenvolvimento deste trabalho, sendo que neste capítulo o foco é a ferramenta LSPS e a arquitetura dos projetos criados na mesma.

No quarto capítulo é apresentado o resultado e discussão resultantes do desenvolvimento de uma aplicação BPM demonstrativa representando uma loja virtual, onde o processo de negócio baseia-se no processamento de pedidos realizados por clientes.

Ao final, no quinto capítulo, a são apresentadas as considerações finais sobre o desenvolvimento de aplicações BPM utilizando-se desenvolvimento orientado a modelos e notação BPMN. A aplicação criada foi analisada como prova de conceito e ocorre um levantamento dos benefícios da utilização do LSPS no processo de desenvolvimento de aplicações BPM bem como para organizações que utilizem BPM.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados os conceitos e tecnologias necessários para a compreensão e embasamento do objetivo proposto.

2.1 BPM

Segundo Palmer (2015), "*Business Process Management* é a disciplina envolvendo qualquer combinação de automação, execução, controle, medição e otimização do fluxo de atividade de negócio, em suporte a metas de empresas, abrangendo sistemas, empregados, clientes e parceiros dentro e além das fronteiras da empresa". Argumenta-se que BPM permite que empresas sejam mais eficientes e capazes de mudança do que uma abordagem focada em funcionalidade.

BPM existe para trazer a tona as informações pertinentes de como os processos são executados para que melhorias possam ser realizadas e os processos possam ser gerenciados para uma melhor tomada de decisão e visão de negócio como um todo (Barrios, 2013). Assim, uma empresa precisa ter conhecimento de seus processos e rotinas para que se possa realizar medições e o gerenciamento destes. Uma empresa que não tem conhecimento sobre seus processos e organização tende ao desperdício de recursos, tanto humanos como financeiros.

Ainda segundo Barrios (2013), dentre os benefícios da gestão de processos vale ressaltar a visualização da organização com o foco nos processos orientados ao cliente e a estratégia da organização suportando e promovendo a facilidade na tomada de decisões sobre os processos, além é claro, da transparência que BPM proporciona ao negócio uma vez que todas as atividades realizadas em uma companhia podem ser vistas como processos e modeladas como tal.

2.2 BPMN

BPMN é a notação gráfica utilizada na modelagem de processos de negócio. Essa notação permite a representação de processos de maneira detalhada, passando desde o início do processo, pelas atividades desempenhadas nele, estruturas de decisão e finalmente o ponto que determina seu fim. A modelagem de processos a partir do uso de BPMN é importante para automatização dos mesmos sendo que permite o descobrimento desses processos revelando suas falhas e expondo pontos que podem ser otimizados (Nogueira Arantes, 2014).

Segundo a OMG (*Object Management Group*), instituição mantenedora do BPMN, o principal objetivo do BPMN é prover uma notação que seja prontamente compreensível por todos os envolvidos no processo de negócio, desde os analistas de negócio, que esboçam os primeiros rascunhos do processo, até os desenvolvedores responsáveis por implementar a tecnologia que irá desempenhar tais processos e finalmente aos administradores que irão gerenciar e monitorar esses processos. Dessa maneira, BPMN cria uma ponte para preencher o vazio entre o design de processos de negócio e a implementação dos mesmos (Object Management Group, 2011).

2.2.1 Elementos BPMN

Um dos principais objetivos da notação BPMN é prover um mecanismo simples e compreensível para a criação de modelos de processos de negócio, mas ao mesmo tempo, ser capaz de lidar com a complexidade encontrada em processos de negócio. A abordagem escolhida para comportar esses requerimentos contrários organiza os elementos BPMN nas seguintes categorias básicas (Object Management Group, 2011):

1. Objetos de Fluxo (*Flow Objects*);
2. Dados (*Data*);
3. Objetos de Conexão (*Connection Object*);
4. Raia de Piscina (*Swimlanes*);
5. Artefatos (*Artifacts*).

Neste trabalho, somente as categorias Objetos de Fluxo, Objetos de Conexão e Raia de Piscina foram abordados pois contém os principais elementos gráficos que definem o comportamento de um processo de negócio.

2.2.2 Objetos de Fluxo BPMN

Objetos de Fluxo são os principais elementos gráficos na definição do comportamento de processos de negócio. Existem três Objetos de Fluxo: Evento, Atividade e Decisão.

Evento é o primeiro elemento definido sob a categoria Objetos de Fluxo. Um Evento é uma ação que acontece durante o curso do Processo. Esses eventos afetam o fluxo do processo e geralmente tem uma causa (gatilho) ou impacto (resultado). Um Evento é definido graficamente como um círculo com o centro aberto para permitir marcadores internos que definem se tal evento possui um gatilho ou resultado. Existem três tipos de eventos que se diferenciam pelo momento em que afetam o fluxo: Início (*Start*), Intermediário (*Intermediate*) e Fim (*End*). A Figura 1 representa a notação para Evento.

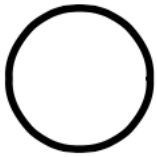

Figura 1 – Evento, disparado durante a execução de um processo.
Fonte: (Object Management Group, 2011).

Uma atividade é um termo genérico para o trabalho que uma companhia realiza em um processo. Atividades podem ser atômicas ou não atômicas (composta). Os tipos de atividades que são parte do Modelo de Processo são: Sub processo (*Sub-Process*) e Tarefa (*Task*) os quais são representados por um retângulo arredondado. Atividades são utilizadas em ambos Processos e Coreografias padrões. A Figura 2 representa a notação para Atividade.

Figura 2 - Notação para Atividade.
Fonte: (Object Management Group, 2011).

Decisões são usadas para controlar a divergência e convergência de um Fluxo de Sequência em um Processo e em uma Coreografia. Assim, decisões determinam ramificação, bifurcação, mescla e encontro de caminhos. Marcadores internos determinam o tipo de comportamento de controle. Na Figura 3 verifica-se a representação para a notação decisão.

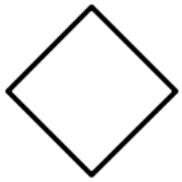

Figura 3 – Notação para elemento Decisão
Fonte: (Object Management Group, 2011).

2.2.3 Objetos de Conexão BPMN

Existem quatro maneiras de se conectar Objetos de Fluxo entre si. Essas quatro maneiras são representadas por quatro diferentes Objetos de Conexão. Os três Objetos de Fluxo mais importantes são: Fluxos de Sequência, Fluxos de Mensagem e Associação.

Fluxos de Sequência são utilizados para mostrar a ordem em que Atividades serão realizadas em um Processo ou Coreografia. A Figura 4 representa a notação gráfica para o elemento Fluxo de Sequência.

Figura 4 – Notação de um Fluxo de Sequência
Fonte: (Object Management Group, 2011).

Fluxos de Mensagem são utilizados para mostrar o fluxo de mensagens entre participantes que estejam preparados para recebe-los. Em BPMN, duas *Pools* diferentes em

um Diagrama de Colaboração irão representar dois Participantes. A Figura 5 representa a notação gráfica de um Fluxo de Mensagem.

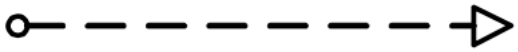

Figura 5 – Notação de Fluxo de Mensagem.
Fonte: (Object Management Group, 2011).

Associações são utilizadas para ligar informação e Artefatos com elementos gráficos BPMN. *Text Annotations* e outros Artefatos podem ser associados com elementos gráficos. Uma seta na Associação pode indicar a direção do fluxo. A Figura 6 representa ambas Associações não-direcionadas e direcionadas.

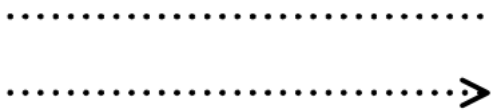

Figura 6 – Associações.
Fonte: (Object Management Group, 2011).

2.2.4 Raia de Piscina (*Swimlanes*) BPMN

Existem duas maneiras de agrupar elementos de modelagem primários usando Raias de Piscina: Piscinas (*Pools*) e Raias (*Lanes*).

Piscina é a representação gráfica de um Participante em uma Colaboração. Ela também age como Raia e contêiner gráfico para particionar um conjunto de Atividades de outras Piscinas. Uma Piscina pode conter detalhes internos, na forma de Processo que será executado. A Figura 7 apresenta a notação gráfica para o elemento Piscina.

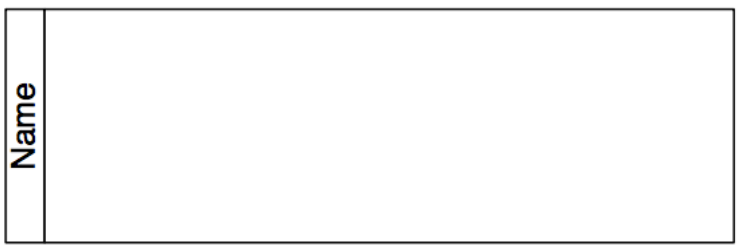

Figura 7 - Piscina.
Fonte: (Object Management Group, 2011).

Raias são sub partições dentro de um Processo, às vezes dentro de uma Piscina, e estendem-se por toda a extensão do Processo, verticalmente ou horizontalmente. Raias são utilizadas para organizar e categorizar Atividades. A Figura 8 representa a notação para o elemento Raia.

Figura 8 - Raia.
Fonte: (Object Management Group, 2011).

2.3 LINGUAGEM DE MODELAGEM GO-BPMN

A linguagem GO-BPMN (*Goal-Oriented Business Process Modeling Notation* ou Notação para Modelagem de Processos de Negócio Orientada a Metas) foi desenvolvida pela Whitestein Technologies e é uma linguagem de modelagem para especificação e modelagem de processos de negócio. GO-BPMN estende de BPMN 1.2 especificado pela OMG e permite a aplicação de ambas abordagens: orientação a metas e uso do BPMN clássico para modelagem de processos (Whitestein Technologies AG, 2015).

O escopo da linguagem de modelagem GO-BPMN abrange (Whitestein Technologies AG, 2015):

- Mecanismos e elementos para organização de modelos e recursos;
- Contextos para modelos;
- Elementos específicos para modelagem de modelos orientados a metas;
- Elementos comuns do BPMN para modelagem de processos;
- Mecanismos e elementos para modelagem de modelos organizacionais;
- Mecanismos e elementos para modelagem de modelos de tipos de dados;
- Mecanismos e elementos para visualização.

Um *Goal Model* ou Modelo de Meta é uma hierarquia de *Goals* (Metas) e *Plans* (Planos) e suas conexões com o processo baseado em metas. A Meta especifica o que precisa ser cumprido pelo processo enquanto o Plano especifica como será procedido para que a Meta seja alcançada (Whitestein Technologies AG, 2015).

O elemento *Achieve Goal* ou Meta é o elemento de modelagem que representa a condição ou estado que se busca alcançar. Metas geralmente representam os objetivos explícitos do processo, como produzir um carro ou enviar uma mensagem. Um Alcançar Meta é representado pelo elemento gráfico exibido na Figura 9.

Figura 9 - Notação Gráfica de Metas.
Fonte: (Whitestein Technologies AG, 2015).

Plan ou Plano é um elemento na hierarquia de Metas que especifica o que deve ser feito para que a Meta diretamente acima de si na hierarquia seja alcançada. Um Plano encapsula um modelo de Plano, que é um fluxo coerente de Eventos e Atividades conectadas por Fluxos. Um Plano deve ter exatamente um elemento superior na hierarquia de Metas e este elemento deve ser uma Meta e não deve ter fluxos de saída, tendo assim a função de elemento folha na hierarquia de Metas. A Figura 10 representa a notação definida para Planos (Whitestein Technologies AG, 2015).

Figura 10 - Notação Gráfica de Planos.
Fonte: (Whitestein Technologies AG, 2015).

2.4 LSPS *EXPRESSION LANGUAGE*

A LSPS *Expression Language* ou Linguagem de Expressão LSPS é uma linguagem funcional estaticamente tipada, usada em modelos da suíte *Livings Systems Process Suite* para computar e processar valores. Não é uma linguagem de programação completa, sendo que não é possível definir modelos de negócio exclusivamente na LSPS *Expression Language*. Por outro lado, essa linguagem é suportada em muitos pontos da modelagem de processos sendo

que propriedades dos elementos GO-BPMN são definidas por expressões nesta linguagem (Whitestein Technologies AG, 2015).

2.5 DESENVOLVIMENTO ORIENTADO A MODELOS

Ao longo dos anos, o processo de desenvolvimento de software tem se tornado complexo, exigindo muito esforço quando se é escolhida uma abordagem centrada em código-fonte. Segundo France e Rumpe (2007) o esforço na construção manual de sistemas de software complexos, pode ser comparado a construção de pirâmides no Egito: "A admiração está principalmente baseada na apreciação pelo esforço requerido para se lidar com significantes complexidades acidentais que surgem do uso de tecnologias inadequadas".

Desenvolvimento Orientado a Modelos ou *Model Driven Development* (MDD) é uma abordagem de desenvolvimento de software que foca na elaboração de modelos do sistema. Seguindo esta premissa, o desenvolvimento de software é orientado a seguir modelos criados previamente ao processo de implementação.

MDD visa resolver o problema de complexidade no desenvolvimento de software defendendo a criação de modelos como base para implementação. A proposta do MDD é reduzir a distância entre o problema a ser solucionado e sua solução. A complexidade em lidar com essa distância é lidada por meio do uso de modelos de alto nível que os desenvolvedores transformarão em aplicações reais (France; Rumpe, 2007).

2.5.1 Benefícios do Desenvolvimento Orientado a Modelos

A implementação de uma arquitetura orientada a modelos tem como objetivo, à longo prazo, uma flexibilidade em termos de Truyen (2006):

- Obsolescência tecnológica: novas infraestruturas de implementação podem ser mais facilmente integradas e suportadas por designs existentes;
- Portabilidade: funcionalidade existente pode ser mais facilmente migrada para novos ambientes, como ditado pelas necessidades das empresas;

- Produtividade: automatiza várias tarefas tediosas e livra desenvolvedores e arquitetos para que foquem sua atenção na lógica central do sistema;
- Qualidade: a separação formal de preocupações, implicadas por esta abordagem, somada à consistência e confiabilidade dos artefatos produzidos contribuem para uma melhora na qualidade do sistema em geral;
- Integração: a produção de pontes de integração com sistemas legados ou de terceiros é grandemente facilitada;
- Manutenção: a disponibilidade de design de forma legível por máquina dá aos analistas, desenvolvedores e testadores acesso direto à especificação do sistema, simplificando a manutenção;
- Teste e simulação: modelos podem ser diretamente validados junto aos requisitos, assim como testados junto a várias infraestruturas. Também podem ser usados para simulações de comportamento do sistema em desenvolvimento;
- Retorno de investimento: empresas podem extrair maior valor de seus investimentos em ferramentas.

2.6 BUSINESS PROCESS MANAGEMENT SUITE

Business Process Management Suite (BPMS) é um tipo de ferramenta que permite a gestão de processos, incluindo atividades de modelagem, execução, controle e monitoração de processos. Esse tipo de ferramenta destina-se a simplificar e otimizar processos de negócio e fluxos de trabalho para torna-los mais eficientes e adaptáveis a ambientes em constante mudança. Assim, BPMS é uma ferramenta valorosa para desenvolvimento e melhoramento de processos de negócio, sendo que pode armazenar dados sobre estes de maneira que possam ser analisados pelas empresas (Janssen, 2015).

Versões comerciais de BPMS tendem a focar a otimização de processos de negócio, movendo companhias dos processos à base de “papel e caneta” para processos automatizados. Essa ferramenta também monitora os processos visando a garantia de que os processos sejam adequadamente executados. Isso permite a exposição dos “gargalos” que causam desperdícios de recursos e destaca possíveis deficiências nos processos (Janssen, 2015).

Sem a utilização de ferramentas BPMS, os processos de uma empresa podem ser vistos como implícitos, uma vez que cada empregado armazena em sua mente o funcionamento da empresa. Essas pessoas também têm suas próprias listas de afazeres interagindo tanto com humanos quanto com sistemas. Com o uso de BPMS, os processos são explícitos e portanto mais adaptáveis que a abordagem anterior, afinal, existem ferramentas, metodologias e técnicas que permitem a adaptação de processos. Nesse ambiente, gerentes de negócio podem assumir direta responsabilidade pelos processos (Whitestein Technologies AG, 2014).

2.6.1 Living Systems Process Suite

Assim como automação de processos se espalha por meio de companhias e cadeias de suprimento, mais instâncias surgem com fluxos de processos que devem se adaptar continuamente à situação e contexto no qual estão sendo desempenhados. Dessa maneira, uma ferramenta BPMS moderna deve fazer mais do que simplesmente automatizar o modelo de fluxo de trabalho. Ela deve também ser capaz de assistir de forma inteligente os participantes do negócio de maneira a auxiliá-los na tomada de decisões em determinados contextos (BPM.Com, 2015).

Living Systems Process Suite (LSPS) foi criado pela *Whitestein Technologies* como uma ferramenta BPMS, agnóstica em relação a domínio e orientada a metas. Ela permite o design, execução, análise e ágil otimização de processos de negócio. LSPS tem sido usado em diversos domínios, como industrial, bancário e seguros, todos em setores público e privado (BPM.Com, 2015).

"Whitestein tem tido grande cuidado em produzir uma plataforma BPM que ofereça tudo que seus clientes precisam e esperam" (BPM.Com, 2015).

Algumas das principais características que identificam o LSPS são (BPM.Com, 2015):

- Gerenciamento de processos: LSPS constrói aplicações que combinam modelos baseados em BPMN com interfaces de usuário ricas, *Business Intelligence* (BI) integrada, e robustas funcionalidades de gerenciamento para melhora de processos. Também conta com controle de acesso baseado em atribuições (papéis) de usuário, monitoramento de processos, regras de negócio proativas e gerenciamento de decisões;

- SOA/Integração de Sistemas Legados: LSPS constrói aplicações puramente Java em uma elegante arquitetura de *deploy*, estável e escalável. Ele se encaixa, sem esforço, como camada de orquestração, gerenciando serviços web, sistemas legados e integração interna de organizações. Sua abordagem orientada a metas provê camadas de abstração adicionais para orquestrar e governar um panorama completo para alinhar dados, infraestrutura e processos com metas da organização;
- Processos Orientados a Metas: usa metas de negócio para dirigir, rastrear e medir processos. Metas são criadas diretamente no modelo de processo e são usadas para governar e monitorar desempenho de processos. BPMN Orientado a Metas enriquece largamente o uso de BPMN por separar política de procedimento: Metas descrevem o que um processo irá fazer – seus marcos e objetivos – enquanto planos de atividades descrevem como cada meta pode ser alcançada;
- Melhoria Contínua: modelos de Processos podem ser alterados em qualquer momento de forma imediata. Isso promove melhora operacional contínua e minimiza esforço, atraso e custos na mudança de processos;
- Design de Processos Colaborativo: Equipes usando LSPS criam e melhoram modelos de processos em um ambiente colaborativo que reúne as competências de profissionais de negócios e de tecnologia da informação durante todo o ciclo de vida do processo. Ferramentas de mescla garantem que contribuições individuais, possivelmente conflitantes sejam resolvidas em um processo final estável. Uma versão integrada online de gerenciamento de versões dá acesso à todos os recursos de processos. Isso provê um histórico completo de mudança e a possibilidade de se voltar à uma versão anterior em qualquer momento.
- Bibliotecas de Aplicação: Toda aplicação de processos criada com LSPS pode ser criada como uma solução independente. Entretanto, toda aplicação tipicamente tirará vantagem de camadas de bibliotecas que promovam, ou possivelmente exijam, o reuso de elementos de modelo, modelos e padrões;
- Escalabilidade e Desempenho: O Mecanismo de Execução em tempo real do LSPS é encapsulado como um *Enterprise Archive* (EAR) pronto para publicação (*deployment*) em servidores de aplicações JEE (*Java Platform, Enterprise Edition*). Isso implica que aplicações de processos orientados a

modelos se beneficiam em termos de escalabilidade, desempenho e disponibilidade, do acesso direto ao servidor JEE, sem ineficiências introduzidas pela separação do motor de processos da aplicação servidora.

Buhler (2008) em seu trabalho realizado junto ao Departamento de Assuntos de Veteranos dos Estados Unidos da América utiliza LSPS como ferramenta para implementação de uma aplicação para processamento de reivindicações e destaca algumas vantagens do uso dessa plataforma:

- É orientado a modelos BPMN e centrado em Java. É também baseado no Eclipse IDE (*Integrated Development Environment*) e amigável ao desenvolvedor;
- Provê melhorias no controle de código fonte tradicional e rotinas de compilação do Maven tornam possível sua integração com o processo de compilação da integração contínua.
- Aplicações criadas com LSPS são publicadas como JEE EAR tornando-as compatíveis com rotinas de publicação existentes;
- Possibilidade de integração de Tarefas Customizadas (*Custom Tasks*) do LSPS com serviços EJB (*Enterprise JavaBeans*) ou serviços externos baseados em SOAP (*Simple Object Access Protocol*);
- A aplicação administrativa do LSPS, console de gerenciamento de processos e relatórios do painel de instrumentos (*dashboard*) estão disponíveis para operadores via navegador web, facilitando diagnósticos remotos.

3 MATERIAL E MÉTODOS

Este capítulo apresenta os materiais e métodos utilizados ao longo do desenvolvimento deste trabalho. A aplicação destes no desenvolvimento de software demonstrativo, resulta em uma aplicação que será visualizada como um dos resultados deste trabalho.

Para este trabalho foi definido que a aplicação demonstrativa represente uma loja virtual onde são processados pedidos de produtos realizados por clientes. Esta aplicação demonstrativa foi inspirada na aplicação modelo *OnlineShop* provida juntamente ao LSPS versão 2.7, tendo sido realizadas diversas modificações para suprir os objetivos deste trabalho. O processo de negócio utilizado é detalhado e definido ao longo do processo de criação do mesmo no desenvolvimento da aplicação LSPS discorrido neste trabalho.

3.1 FERRAMENTAS

A plataforma LSPS, descrita a seguir é utilizada para o desenvolvimento deste trabalho, está disponível em <https://www.whitestein.com/user/register/cloud>. Após o registro para avaliação da ferramenta pode então ser feito o *download* da mesma e instalação, de forma gratuita.

3.1.1 Living Systems Process Design Suite

Living Systems Process Design Suite (PDS) é a IDE disponibilizada pelo LSPS. O PDS é um ambiente baseado no Eclipse *framework* que disponibiliza ferramentas para modelagem de modelos de processos de negócio, gerenciamento e monitoração em tempo real.

A Figura 11 é a representação de uma visão geral sobre a arquitetura do ambiente LSPS com o PDS na esquerda contendo os modelos de negócio. Também está representada a conexão com bancos de dados, geração de interface de usuário ao topo e à direita o *Business*

Activity Monitoring ou Monitor de Atividade de Negócios para monitoramento de atividades dos processos.

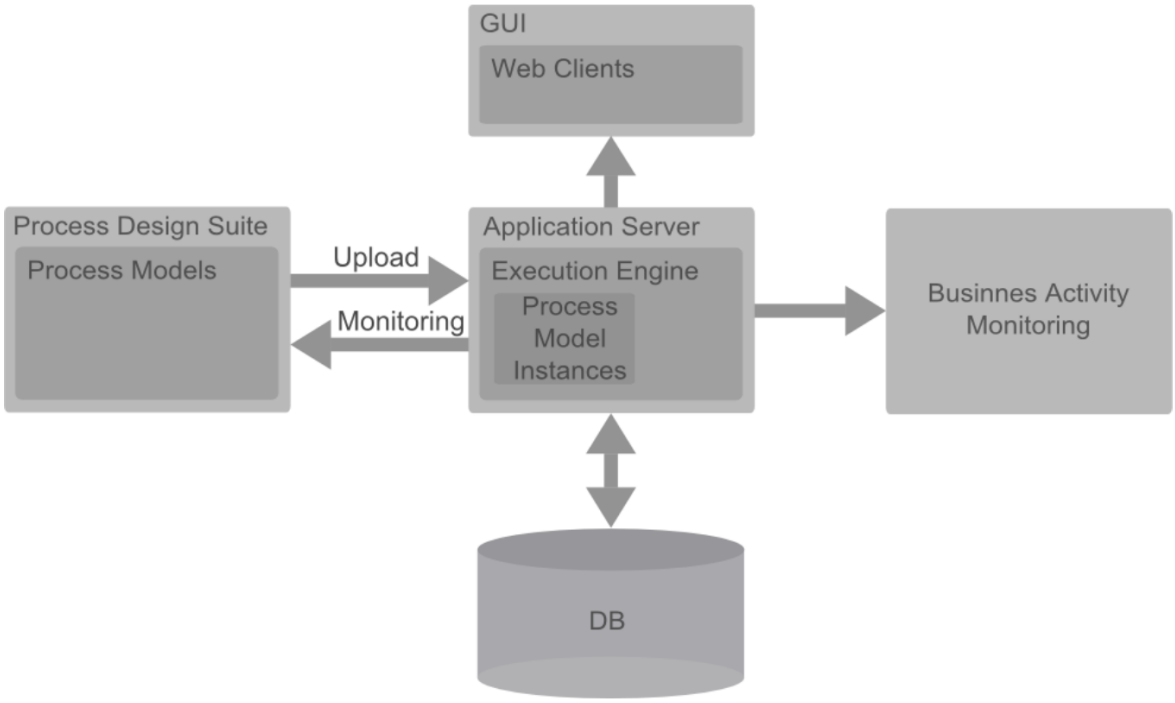

Figura 11 - Visão geral do LSPS.
Fonte (Whitestein Technologies AG, 2015).

O principal componente da plataforma LSPS é o *Execution Engine*, ou Mecanismo de Execução que contém e gerencia os modelos de negócio bem como dados relacionados à estes os quais podem ou não ser persistidos em bancos de dados.

Modelos de negócio são criados e editados utilizando-se o PDS. Essa ferramenta também permite a publicação (*deploy*), execução dos módulos de negócio no Mecanismo de Execução e também trabalhar com os dados de execução. Leia-se como dados de execução os dados referentes a modelos publicados.

O PDS é entregue ao usuário juntamente com os seguintes *thin clients* ou clientes magros baseados na web (Whitestein Technologies AG, 2015):

- *Process Application* ou Aplicação de Processo que é a aplicação padrão para execução de tarefas de usuário gerada pelas instâncias de modelo;
- *Process Management Console* ou Console de Gerenciamento de Processos, é uma aplicação administrativa para envio, execução e monitoramento de modelos e suas respectivas instâncias;
- *Business Activity Monitor* (BAM) ou Monitor de Atividade de Negócios é uma aplicação para geração de relatórios sobre os dados de execução.

Um típico fluxo de trabalho dentro do ambiente PDS se dá por meio da seguinte sequência de etapas (Whitestein Technologies AG, 2015):

1. Usuário cria um modelo utilizando-se o PDS;
2. Este modelo é então carregado, ou enviado, ao mecanismo de execução que é então publicado em um servidor de aplicações suportado pelo LSPS;
3. Uma instância do modelo é então criada sendo definida pelo próprio modelo. Essa instância existe em seu próprio contexto com seus próprios dados de execução;
4. A instância de modelo executa os processos como definido pelo modelo. Durante a execução ela pode armazenar ou consultar dados de um armazém de dados, como um banco de dados, requisitar dados de sistemas externos assim como atores humanos;
5. Uma vez que todos os processos sejam finalizados, a instância de modelo também finaliza-se.

Os dados sobre instâncias de modelo e seus contextos podem ser usados pelos relatórios gerenciados pelo BAM.

Modelos ou Modelos de Processos são recursos *standalone* que podem ser instanciados no Mecanismo de Execução em seus respectivos contextos de execução. Para que um modelo seja criado, é necessário antes criar um módulo. Módulo, por sua vez, é uma construção abstrata pois não é representado por elemento de modelagem, mas sim, por todos os recursos que o compõem, incluindo outros módulos importados. Módulos precisam necessariamente ser executáveis para serem instanciados e considerados modelos. (Whitestein Technologies AG, 2015)

Todos os módulos devem ser criados dentro de um projeto GO-BPMN sendo que um projeto pode conter um número arbitrário de módulos. Esses projetos são contêineres *top-level* para módulos e demais artefatos durante a modelagem. (Whitestein Technologies AG, 2015)

Um projeto GO-BPMN contém os seguintes itens:

- Módulos;
- Bibliotecas;
- Configurações de atualização de modelos;
- Arquivos e pastas genéricas.

Um projeto também pode referenciar outros projetos o que permite módulos importarem módulos de outros projetos. São os módulos que contêm arquivos que armazenam

recursos como processos, tipos de dados e variáveis. A estrutura organizacional de um projeto GO-BPMN pode ser verificada na Figura 12. Nesta representação verificam-se a definição de um módulo, chamado "*Module* 1.0", suas dependências aninhadas sob "Module Imports", definições de processo e variáveis.

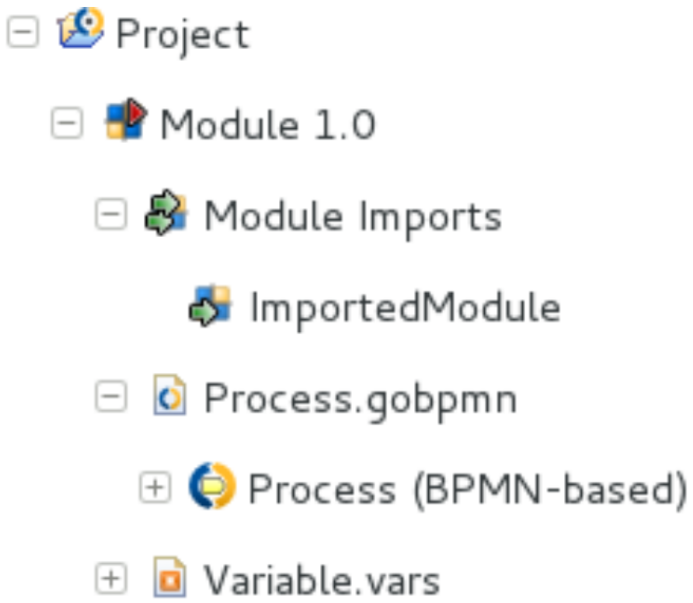

Figura 12 - Estrutura de projeto GO-BPMN.
Fonte (Whitestein Technologies AG, 2015).

A interface de usuário do PDS opera sobre dados em uma particular área de trabalho ou *workspace*. Área de trabalho é um diretório onde recursos utilizados no trabalho com o PDS ficam armazenados. Leia-se por recursos, diretórios e arquivos que podem constituir projetos GO-BPMN. A seleção de área de trabalho a qual se deseja utilizar é feita ao iniciar a ferramenta PDS, podendo ser trocada a qualquer momento.

O eixo central para dados de usuário é chamado *workspace*. Pode-se pensar na plataforma de trabalho como uma ferramenta que permite usuários navegarem e manipularem o *workspace* (Eclipse, 2013).

Uma perspectiva define o conjunto inicial e disposição de visualizações ou *views* na janela *Workbench*. Cada perspectiva provê um conjunto de funcionalidades destinadas a cumprir um tipo específico de tarefa ou trabalho com tipos específicos de recursos. Por exemplo, a perspectiva Java combina *views* que geralmente são usadas quando se edita arquivos fonte Java (Eclipse, 2013).

Living Systems Process Design Suite contém três perspectivas especificamente criadas para essa ferramenta (Whitestein Technologies AG, 2015):

- Modelagem ou *Modeling*, para criação de modelos;
- Gerenciamento ou *Management*, para gerenciamento de modelos em tempo de execução;
- Navegação de Modelos ou *Model Browsing*, para apresentação de modelos.

Workbench ou bancada é o conjunto de views, barras de ferramentas, menus e editores utilizados juntamente com os recursos da área de trabalho. Workbench acomoda as perspectivas como apresentado na Figura 13. A parte superior da bancada exibe barras e menus com comandos para a bancada inteira (Whitestein Technologies AG, 2015):

- Barra de menus: provê acesso a funcionalidades para a bancada como um todo, incluindo comandos para perspectivas;
- Barra de ferramentas principal: contém botões com funcionalidades aplicáveis ao conteúdo atualmente ativo da bancada (também disponível na barra de menus;
- Gerenciador de Perspectivas: permite a mudança de perspectivas.

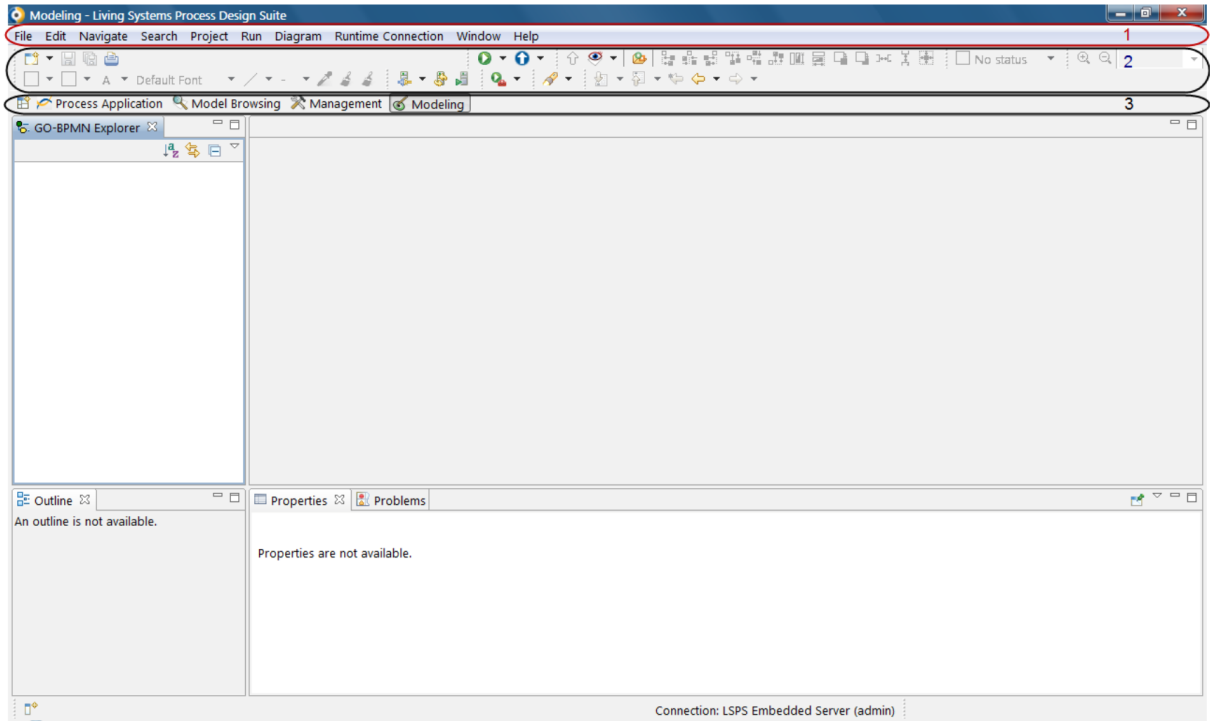

Figura 13 - *Workbench* - Áreas destacadas: 1 - Barra de Menus; 2 - Barra de Ferramentas Principal; 3 - Gerenciador de Perspectivas.
Fonte (Whitestein Technologies AG, 2015).

A maioria das perspectivas são compostas de uma área de edição e uma ou mais *views*. Pode-se associar diferentes editores a diferentes tipos de arquivos (Eclipse, 2013). Para trabalhar com arquivos de recursos, tanto de definições como configurações, pode-se usar tanto editores embarcados como editores externos. Para edição de recursos relacionados ao

PDS existem editores embarcados disponibilizados pela própria ferramenta que são executados por padrão quando um recurso é aberto (Whitestein Technologies AG, 2015). Na Figura 14 é possível verificar o editor embarcado na ferramenta PDS para edição de modelos GO-BPMN e na aba oculta está o editor para Tipos de Dados ou *Data Type Editor*.

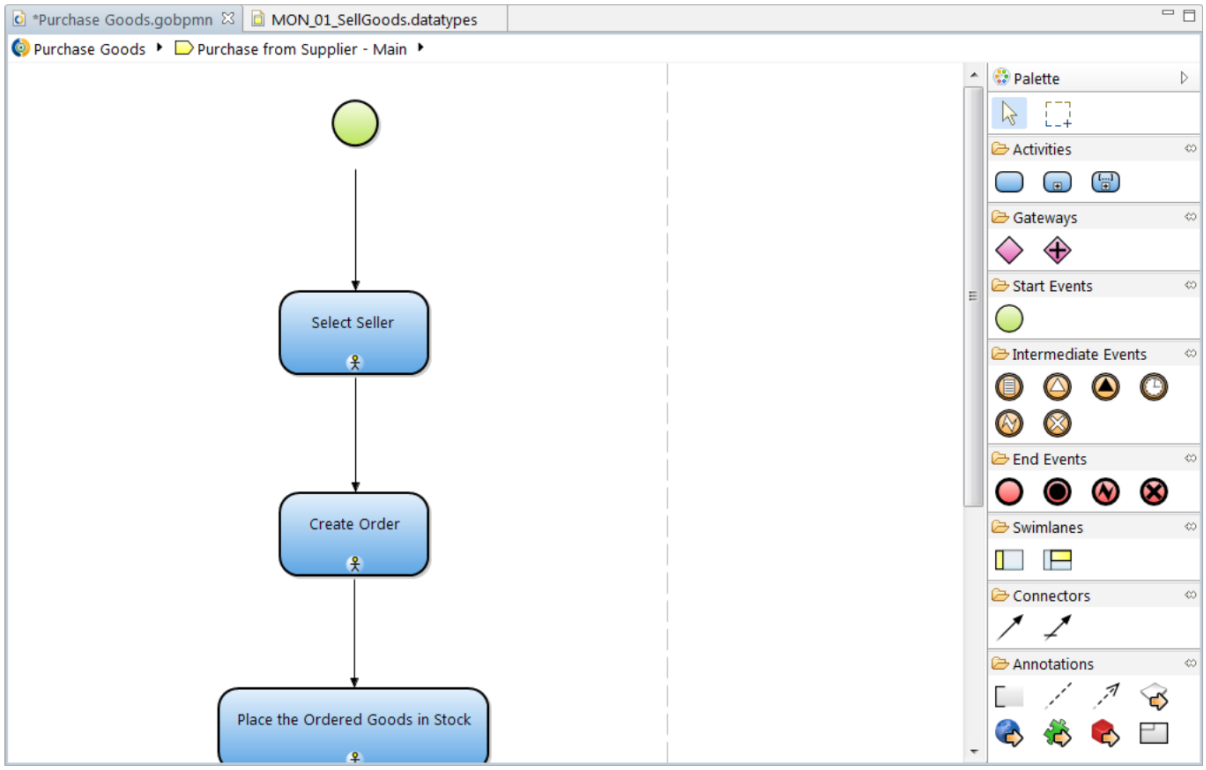

Figura 14 - Editor GO-BPMN e *Data Type*.
Fonte (Whitestein Technologies AG, 2015).

Views suportam editores e proveem apresentações alternativas de recursos, bem como maneiras de se navegar sobre a informação na *Workbench*. Por exemplo, a *view Project Explorer* e outras *views* de navegação exibem projetos e seus respectivos recursos com que se está trabalhando. (Eclipse, 2013)

GO-BPMN *Explorer* é uma *view* para navegação sobre os recursos de uma área de trabalho. Essa visualização é exibida por padrão na perspectiva de Modelagem e contém uma árvore expansível com projetos GO-BPMN disponíveis na área de trabalho atual. Na Figura 15 pode ser visualizada a *view* GO-BPMN *Explorer*.

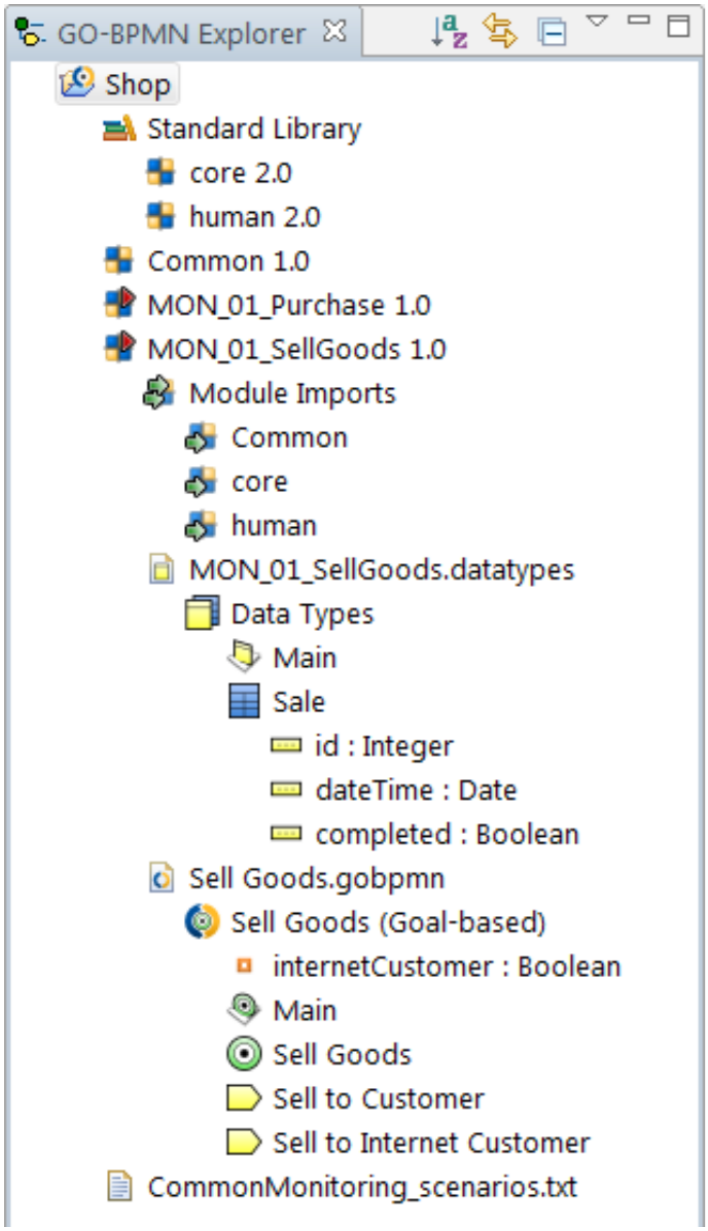

Figura 15 - GO-BPMN *Explorer*.
Fonte (Whitestein Technologies AG, 2015).

3.1.2 Aplicação de Processos LSPS

O *Living Systems Process Application* ou Aplicação de Processos LSPS é a aplicação *front-end* padrão pela qual pode-se criar instâncias de modelo e participar em sua execução. Usuários podem submeter, rejeitar, anotar e salvar seus afazeres ou *todos*, criar instâncias de modelo, pesquisar por instâncias de modelo, trabalhar em documentos e definir aplicações e

configurações de usuário. A Aplicação de Processos é parte da plataforma LSPS e pode ser customizado pelo usuário (Whitestein Technologies AG, 2015).

O ambiente de Aplicação de Processos é renderizado baseado em componentes Vaadin. Em resumo, a interface dessa aplicação exibe os menus à esquerda de maneira que o conteúdo a ser exibido ao usuário seja relativo ao item de menu selecionado à esquerda. Por padrão, do menu de navegação podem ser acessadas as seguintes páginas (Whitestein Technologies AG, 2015):

- Lista de *Todos* ou Afazeres: exibe os afazeres alocados para o usuário corrente;
- Documentos ou *Documents*: exibe os documentos alocados ao usuário corrente;
- Executar Modelo ou *Run Model*: exibe os modelos que o usuário corrente pode instanciar;
- Configurações ou *Settings*: exibe a janela de configurações para a aplicação;
- Sair ou *Log Out*: remove o usuário corrente e navega para a página de *login*.

Um Afazer é uma entidade apresentada na interface de usuário juntamente a especificação do trabalho que o usuário final deve realizar. É gerado quando um processo requer intervenção humana: no *background* o fluxo do processo atinge uma tarefa do tipo *user task*. O Afazer é então designado a um usuário ou grupo de usuários e aparece em suas respectivas listas de Afazeres. O usuário abre um Afazer, realiza as atividades que este demanda e uma vez que concluídas estas tarefas o usuário submete este Afazer que se torna completo e permite então que o fluxo do processo proceda.

3.1.3 Estrutura de Aplicações GO-BPMN

A *view* GO-BPMN *Explorer* exemplificada na Figura 15 suporta a organização de aplicações GO-BPMN em uma organização independente, isto é, um projeto. Um projeto contém estruturas específicas (Whitestein Technologies AG, 2014):

- Bibliotecas: coleções somente-leitura de módulos GO-BPMN. Um exemplo é a *Standard Library* ou Biblioteca Padrão provida pela plataforma LSPS;
- Módulos: reusáveis, são partes independentemente executáveis do processo de negócio. Podem ser organizadas em hierarquias de pastas. Módulos definidos

como executáveis são processos que podem iniciados e contém lógica de processos;

- Pastas: mantém organizados módulos e outros recursos. Por exemplo documentos tais como especificações de processos.

A Biblioteca Padrão está disponível em qualquer projeto GO-BPMN. Esta biblioteca é composta de três módulos principais (Whitestein Technologies AG, 2015):

- *core*: este módulo contém tipos de dados, funções e tipos de tarefas usados para reflexão de modelo, para acessar o contexto de execução de instância de modelo e manipulação fundamental com tipos de dados primitivos em GO-BPMN. Módulo necessário para correto funcionamento de modelos GO-BPMN;
- *human*: módulo que define um conjunto de tipos de dados, funções e tipos de tarefas usados para refletir o modelo organizacional, para acessar dados de usuário a partir do modelo de processo e para gerar Afazeres de diferentes tipos. Este módulo importa o módulo *core*;
- *ui*: módulo que define um conjunto de tipos de dados, funções e *hints* de apresentação de UI (*User Interface* ou Interface de Usuário). Os tipos de dados especificados em sua maioria representam componentes de interface de usuário, eventos relacionados à UI e *listeners*. Todos usados para definir formulários em modelos GO-BPMN.

Aplicações GO-BPMN são compostas por vários recursos, ou entidades, cada um com funções específicas. Esses entidades são definidas em arquivos com extensões específicas à plataforma LSPS, dessa forma, segue lista destas entidades seguidas pela extensão correspondente e o que cada uma representa (Whitestein Technologies AG, 2014) e (Whitestein Technologies AG, 2015):

- Definição de Processos (.gobpmn): modelos de processos GO-BPMN. Arquivos com esta extensão armazenam modelos de processos modelados utilizando-se a notação GO-BPMN;
- Definição de Organização (.orgmodel): estrutura organizacional envolvida no processo. Modelo organizacional que uma companhia use, também é modelada utilizando-se a notação GO-BPMN;

- Modelo de Tipos de Dados (.datatypes): definições de dados específicos do domínio de negócio. Um modelo de tipos de dados é a hierarquia dos tipos de dados customizados e seus relacionamentos em um modelo;
- Definição de Consultas (.queries): consultas para se adquirir dados de negócio de um banco de dados persistente. Uma consulta pode ser definida de forma padrão acessando o banco de dados através dos registros compartilhados definidos no Modelo de Tipos de Dados ou através de consultas nativas definidas pelo comando "SELECT" de bancos de dados;
- Definição de Documentos (.docs): visualizações independentes elaboradas sobre os dados de negócio. É um recurso que define número arbitrário de documentos;
- Definição de Formulário (.form): definição de formulários web para entrada de dados pelo usuário final e interação em processos desempenhada também pelo usuário final. Cada definição de formulário é referente à um único formulário;
- Definição de Funções (.funcs): funções específicas do modelo. Função é uma expressão reusável escrita em uma das linguagens suportadas que desempenhe procedimento específico;
- Definição de Variáveis (.var): variáveis de contexto específicas do modelo. Um módulo pode conter múltiplas definições de variáveis;
- Definição de Constantes (.consts): constantes específicas do modelo. É um recurso que contém definições de uma ou múltiplas constantes;
- Definição de Tipos de Tarefa (.tasks): arquivo de definição de tipos de tarefas customizados. Todo Tipo de Tarefa define um conjunto de atributos, parâmetros e sua classe de implementação.

3.1.4 Java

A linguagem de programação orientada a objetos Java foi desenvolvida por uma equipe de desenvolvedores liderada por James Gosling na *Sun MicroSystems* (atualmente pertence a Oracle). Foi lançada em 1995 e faz parte da Plataforma Java (Pereira, 2009).

O objetivo inicial do time liderado por Gosling objetivava inicialmente criar um interpretador para pequenos dispositivos como aparelhos eletrônicos em geral. A ideia não deu certo por motivos financeiros e conflitos de interesses. Em 1995 a Sun percebeu que poderia usar o Java para rodar pequenas aplicações dentro de navegadores web, podendo-se programar para mais de uma plataforma utilizando-se a mesma linguagem de programação. Foi então que a *Sun* lançou o Java 1.0 (Caelum, 2015).

3.1.5 Vaadin

Segundo Grönroos (2015), o *framework* Vaadin é um framework para desenvolvimento de aplicações Web baseado em Java, destinado a ser usado na criação e manutenção de interfaces de usuário. Vaadin também suporta dois diferentes modelos de programação: lado cliente e lado servidor.

Vaadin possui um mecanismo de comunicação entre cliente e servidor, sendo que este permite que uma aplicação seja totalmente desenvolvida sem se preocupar com peculiaridades como esta, comuns no mundo Web (Carvalho, 2014).

Na Figura 16 é representada a arquitetura de aplicações desenvolvidas utilizando-se Vaadin. O mecanismo de execução do Vaadin é executado no navegador, renderizando a interface, composta em sua maioria por *Widgets*, e enviando a interação de usuário ao servidor. A arquitetura da aplicação consiste em um *server-side framework* ou *framework* no lado servidor e um mecanismo no *client-side* ou lado cliente (Grönroos, 2015).

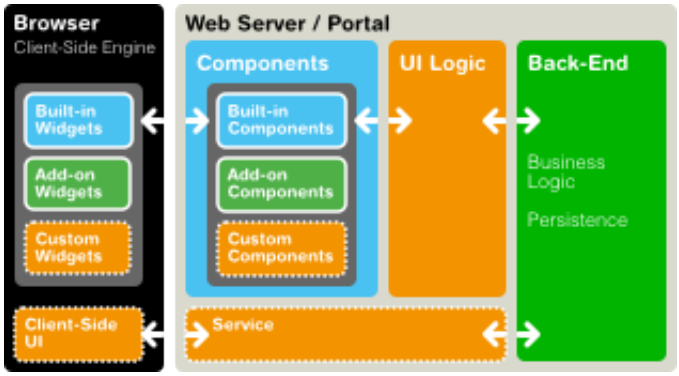

Figura 16 - Arquitetura do *Framework* Vaadin.
Fonte: (Grönroos, 2015).

4 RESULTADOS E DISCUSSÃO

Neste capítulo é apresentada a aplicação BPM resultante da utilização da ferramenta LSPS no processo de desenvolvimento de aplicações BPM, por meio do desenvolvimento orientado a modelo tendo BPMN como notação padrão.

Como resultado, é apresentada a aplicação desenvolvida sendo exposta a possibilidade de execução de BPM, ou seja, o gerenciamento de processos de negócio por meio da plataforma LSPS.

Nas subseções seguintes é detalhado o processo de desenvolvimento de uma aplicação BPM utilizando-se a ferramenta LSPS e uma abordagem orientada a modelos. Dessa maneira, verifica-se a realização da atividade BPM pela modelagem e gerenciamento de processos de negócio por meio da ferramenta trabalhada.

4.1 CRIAÇÃO DE APLICAÇÃO LSPS

O processo de criação de uma aplicação LSPS se dá pela definição inicial do domínio de solução escolhido, isso significa definir o foco da aplicação a ser desenvolvida, bem como o que se espera atingir.

Primeiramente, é necessário que seja criada uma aplicação na plataforma LSPS. Após iniciada a ferramenta, é necessário definir uma *workspace* onde será armazenada a aplicação criada posteriormente, bem como os projetos GO-BPMN com seus modelos de negócio. Tendo uma *workspace* definida o próximo passo é a criação da aplicação LSPS por meio de menus padrões proporcionados pela ferramenta. Após preenchimento dos parâmetros para a criação da nova aplicação, a ferramenta PDS gera um comando Maven que é então executado criando os artefatos que compõem a aplicação. Pode ser verificado na Figura 17 que a aplicação foi criada com êxito e os subprojetos que a compõem podem ser visualizados na *view* GO-BPMN Explorer.

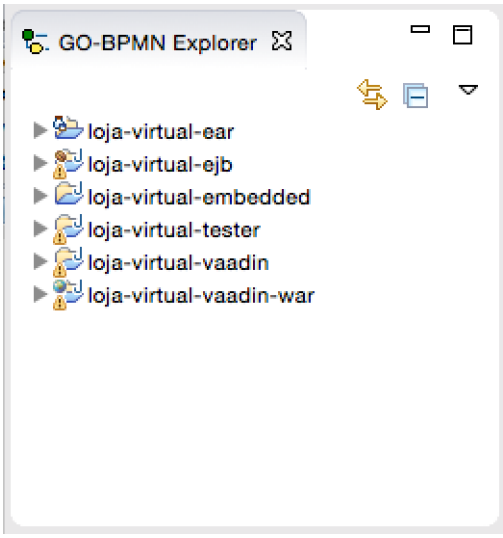

Figura 17 - Aplicação LSPS.

A aplicação criada servirá como base para publicação do processo modelado, sendo que esta contém as interfaces padrão geradas em Vaadin, projeto para implementação de testes, projeto contendo servidor embarcado para testes, EJB e EAR, que é o arquivo a ser usado para publicação da aplicação.

Como próxima etapa, é criado o projeto GO-BPMN, que deverá conter os módulos GO-BPMN que compõem um projeto. Dentro deste projeto, por sua vez, é criado o módulo GO-BPMN que organiza sob sua estrutura, os arquivos de definição de recursos específicos da plataforma. Na Figura 18 pode ser visualizado o projeto citado anteriormente, chamado loja-virtual-model, com um módulo aninhado chamado loja-virtual-principal, que como o nome sugere, será o principal módulo neste projeto e conterá o modelo de processo de negócio.

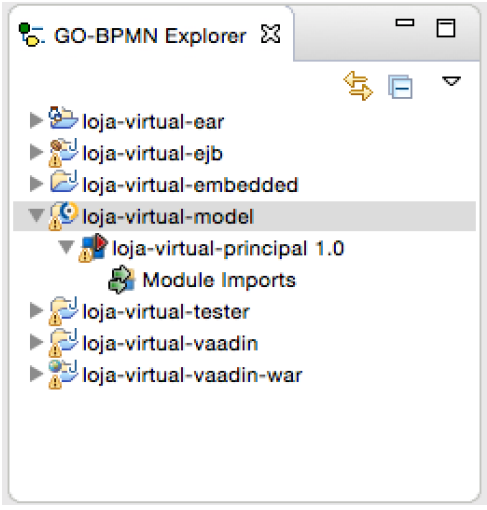

Figura 18 - Módulo GO-BPMN.

4.2 DEFINIÇÕES DE RECURSOS

Um projeto GO-BPMN é composto por vários recursos definidos em arquivos específicos da plataforma LSPS. Estes recursos servem como apoio para o processo de negócio, e são utilizados em todas as atividades de um processo, fazendo parte de sua composição.

Para melhor gerenciamento e organização do projeto, é criado novo módulo GO-BPMN, chamado loja-virtual-base, que armazene as definições de recursos separadamente. Este módulo é definido como não executável sendo que será somente utilizado como dependência pelo módulo principal que contém o processo de negócio.

Tendo definida esta arquitetura organizacional, é iniciado o processo de criação e modelagem dos arquivos de recursos. O primeiro recurso definido é o de Definição de Organização que representa a estrutura organizacional envolvida no processo de negócio. A estrutura utilizada na aplicação implementada neste trabalho pode ser visualizada na Figura 19.

A modelagem desta estrutura organizacional se dá por meio do desenvolvimento orientado a modelo, sendo que o desenvolvedor seleciona os componentes gráficos, disponibilizados pela *view* Palette que também está representada na Figura 19, e os inclui na definição.

Verifica-se que esta estrutura contém três unidades organizacionais, sendo elas Loja Virtual, Produção e Logística, que podem ser vistos como departamentos dentro de uma organização. Também se verifica a presença de papéis organizacionais que são atribuídos a pessoas envolvidas. Sob a unidade Produção encontram-se os papéis de Responsável de Produção e Trabalhador, enquanto sob a unidade organizacional Logística encontra-se o papel Agente de Entregas. Sob a unidade Loja Virtual, aninham-se as unidades Produção e Logística, bem como o papel de Agente Contábil. Além destes recursos citados, este modelo também apresenta os papéis de Cliente, que como o nome sugere, é atribuído a clientes da aplicação, e de Conselho de Diretores que não é relacionado a nenhuma unidade.

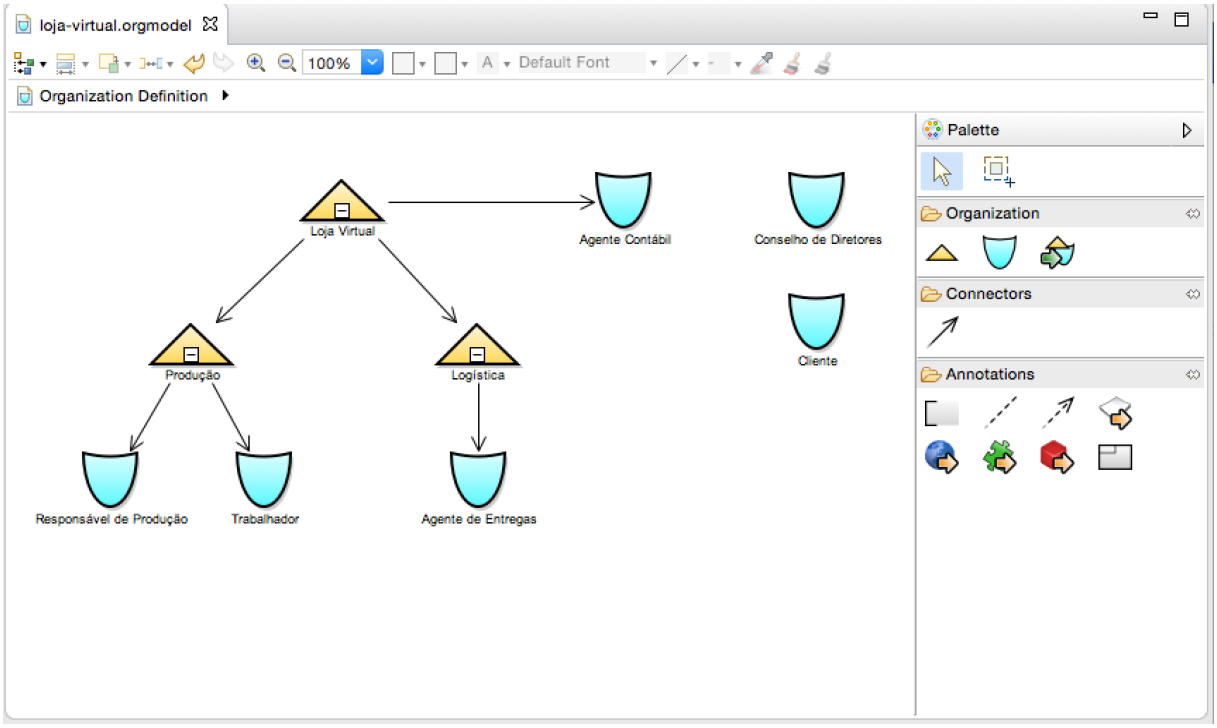

Figura 19 - Estrutura Organizacional.

Após definição da estrutura organizacional, é criado o Modelo de Tipos de Dados. Este modelo é a representação das entidades e seus relacionamentos presentes no processo de negócio. Mantendo em mente que a aplicação desenvolvida neste trabalho é voltada ao processamento de pedidos. É neste arquivo de definições de tipos de dados que se define o que é um Pedido, determinando os atributos que os caracterizam, bem como seus relacionamentos com outras entidades que também compõem esse modelo.

De maneira semelhante a definição de estrutura organizacional, a ferramenta PDS oferece editor padrão para este tipo de arquivo, sendo que o desenvolvedor neste caso também tem a disposição a *view* Palette que provê os componentes adequados para cada tipo de definição de recurso.

O modelo de tipos de dados utilizado no desenvolvimento da aplicação demonstrativa deste trabalho, e representado na Figura 20, é composto por várias entidades, dentre as principais encontram-se:

- Endereco: entidade que representa um endereço físico;
- Cliente: entidade que representa pessoa física que cria pedidos na aplicação. Estende do tipo *human::Person* que pertence a biblioteca *human* provida pelo LSPS;
- Produto: representa produtos que podem ser incluídos em pedidos;

- Pedido: entidade que representa o registro de pedidos de produtos realizados por clientes;
- Cartão de Crédito: entidade que representa cartões de crédito a serem usados como métodos de pagamento de faturas de pedidos.

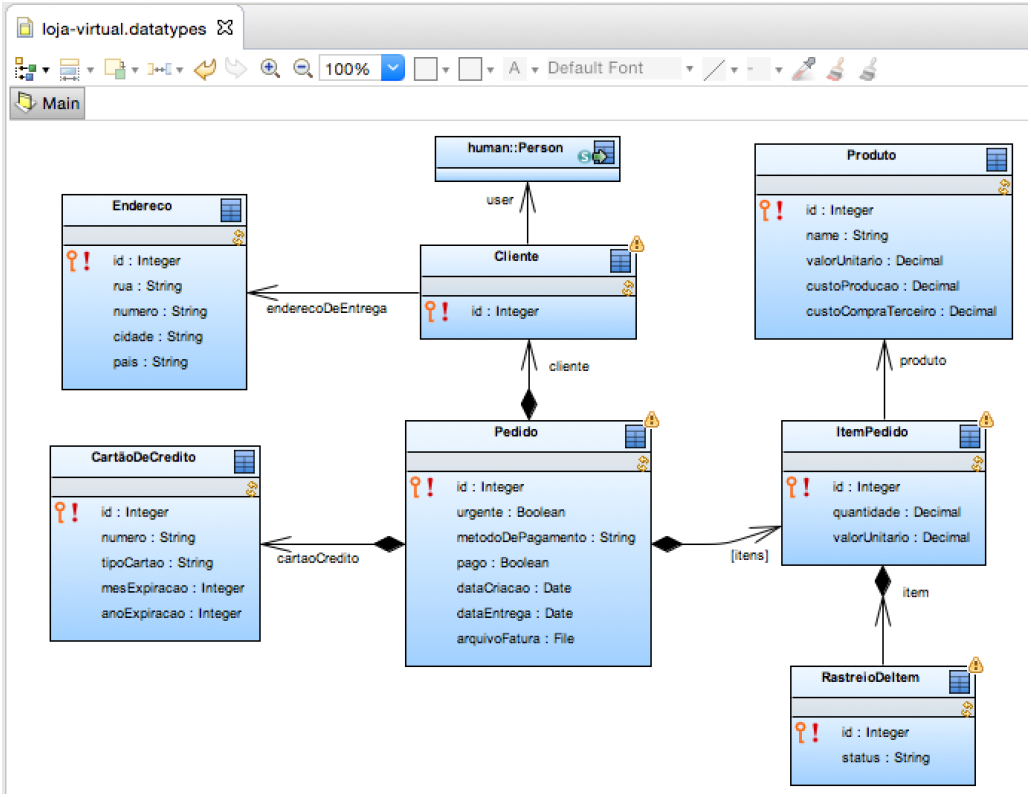

Figura 20 - Modelo de Tipos de Dados.

O próximo arquivo de definição de recursos a ser definido é o de Definição de Constantes, onde são criadas constantes que podem ser acessadas e ter seus respectivos valores lidos pela aplicação. Como o nome sugere, o valor de uma constante é imutável, somente sendo editado em tempo de design e se mantendo o mesmo quando a aplicação se encontra em execução.

O arquivo de definição de constantes utilizado neste projeto pode ser visualizado na Figura 21. Neste arquivo são definidas quatro constantes:

- PROVEM_DE_COMPRA: indica que determinado produto provém de compra junto a terceiros;
- PROVEM_DE_PRODUCAO: indica que determinado produto provém de produção própria da companhia;

- PROVEM_DE_ESTOQUE: indica que a companhia tem determinado produto em estoque, dispensando a necessidade de outros meios de produção ou compra;
- LIMITE_DE_CUSTO: indica o valor máximo que a companhia está disposta a gastar para ter determinado produto à disposição.

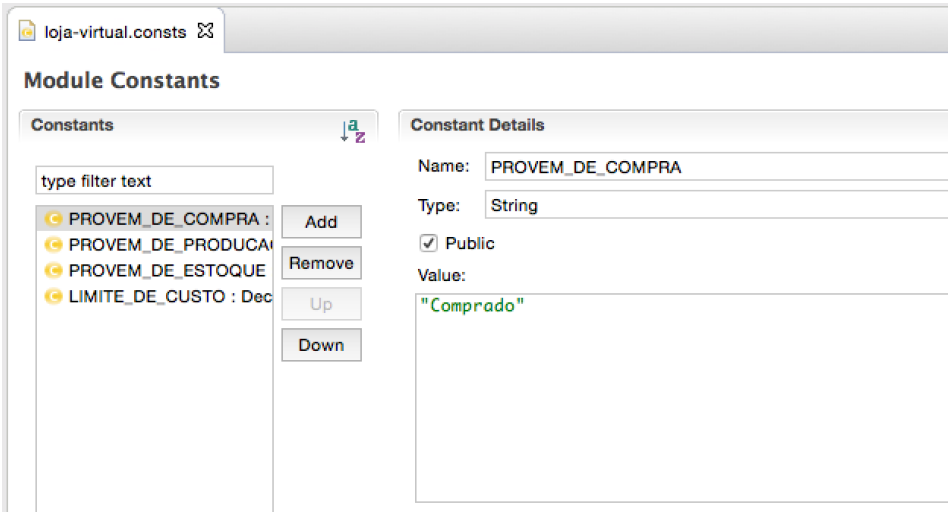

Figura 21 - Definição de Constantes.

Definição de Funções também é importante em uma aplicação LSPS. Funções permitem automatizar certos procedimentos em chamadas simples que podem ser utilizadas sempre que necessário. Na Figura 22 é apresentado o editor padrão do PDS com o arquivo de definição de funções aberto exibindo as funções definidas para esta aplicação e implementadas em *Expression Language*. Verifica-se na esquerda do editor a lista de funções criadas sendo que na direita do editor encontram-se os detalhes da função selecionada da listagem na esquerda. Funções possuem vários atributos, tais como nome, tipo de retorno, tipos genéricos, parâmetros e implementação. A implementação de uma Função pode ser em Java, por meio da implementação de um EJB na aplicação LSPS ou por meio de *Expression Language* como representado na Figura 22.

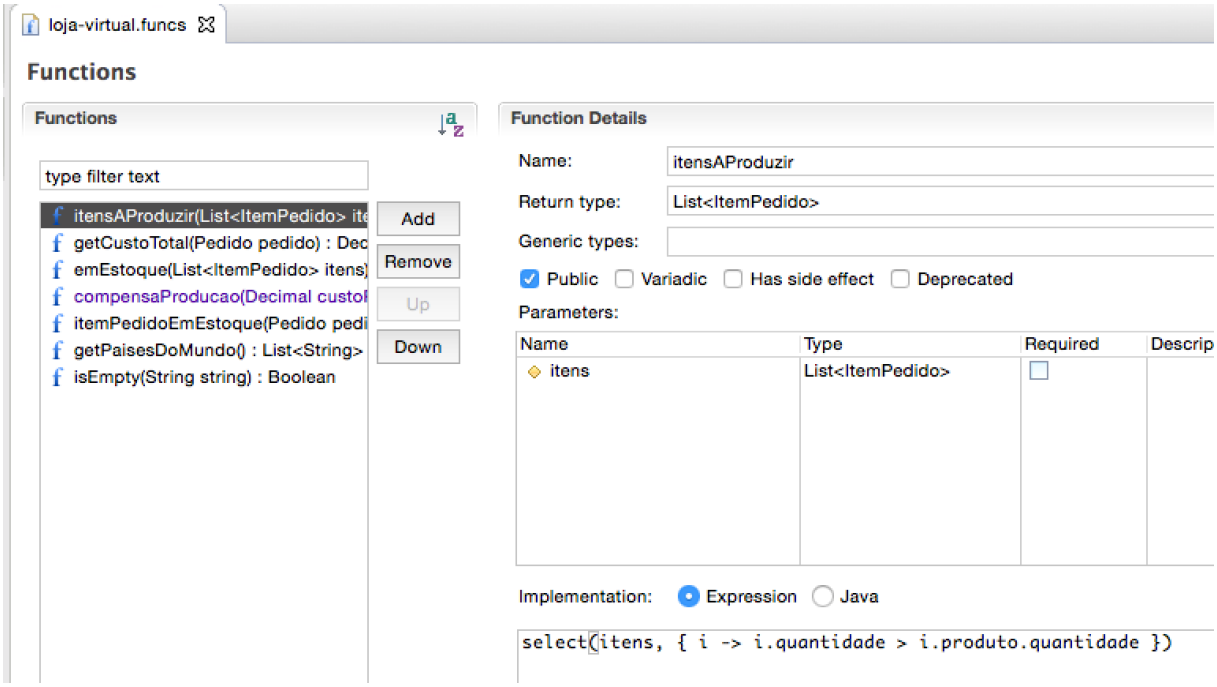

Figura 22 - Definições de Funções.

Tendo os recursos acima definidos, passa-se ao processo de criação da interface Web que permite a entrada de dados pelos usuários da aplicação. A Definição de Formulários permite a criação destas interfaces de usuário por meio da modelagem de formulários, sem necessidade de codificação árdua de telas. Cada Definição de Formulário representa um único formulário Web, portanto, a maioria das aplicações é composta por inúmeras Definições de Formulário.

Buscando organizar e isolar estas definições de recursos das demais previamente criadas, estas são aninhadas em uma pasta *forms* e estão dispostas na *view* GO-BPMN Explorer como representado na Figura 23. Um único módulo GO-BPMN pode conter várias Definições de Formulário. Estas definições podem ser utilizadas em Definições de Documentos ou em Atividades do processo GO-BPMN. Quando os formulários são utilizados como Atividades no modelo de processo de negócio eles são parte principal na realização dos Afazeres, sendo que quando um usuário inicia um Afazer, o formulário associado é renderizado e então ocorre a interação entre usuário e aplicação, sendo que quando o formulário é submetido pelo usuário, o processo de negócio prossegue tendo concluído este determinado Afazer e seguindo para a próxima Atividade a ser desempenhada.

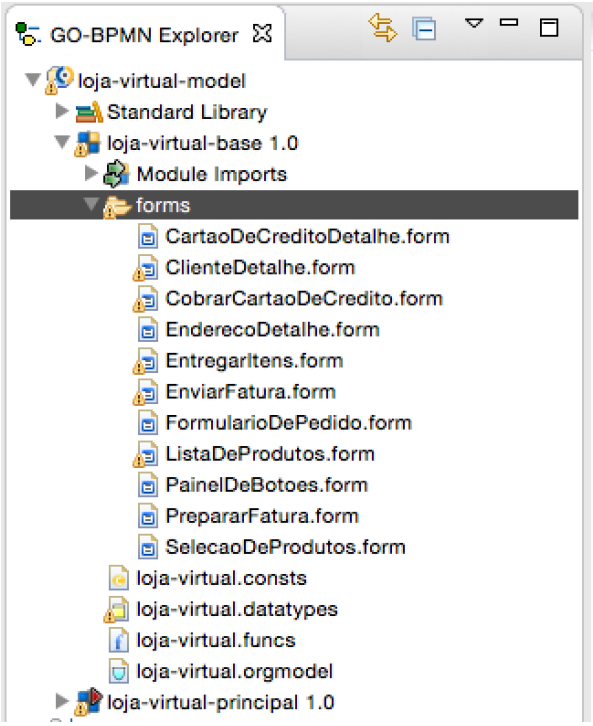

Figura 23 - Definições de Formulário.

Uma Definição de Formulário pode ser modelada com o editor padrão fornecido pela ferramenta PDS, assim como os demais recursos exclusivos do LSPS. Na Figura 24 verifica-se este editor padrão apresentando a Definição de Formulário EnderecoDetalhe, cuja definição possui campos para inserção de valores correspondentes aos atributos de Endereço, tais como: Rua, Número, Cidade, e País.

Na *view* Palette na parte direita da Figura 24 encontram-se os componentes de interação que podem ser inseridos em uma Definição de Formulário. Estes componentes encontram-se subdivididos em categorias que representam seus objetivos e são disponibilizados pela biblioteca *ui* encontrada na Biblioteca Padrão provida pela plataforma LSPS. Dentre as categorias existentes, as principais utilizadas no desenvolvimento deste trabalho são:

- *Container Components* ou Componentes Contêiner: componentes cujo objetivo é auxiliar na organização e disposição gráfica de outros elementos que podem ser aninhados nestes Contêineres;
- *Input Components* ou Componentes de Entrada: componentes utilizados para receber entrada de dados pelo usuário final;
- *Output Components* ou Componentes de Saída: componentes utilizados para saída de dados pelo sistema, quando se objetiva exibir dados ao usuário;

- *Action Components* ou Componentes de Ação: permitem que usuários realizem ações, como submeter um documento ou navegar dentro da aplicação;
- *Custom Components* ou Componentes Customizados: componentes customizados disponíveis em determinado formulário. Usuários da plataforma podem criar Componentes Customizados que tenham comportamento diferenciado dos componentes padrão.

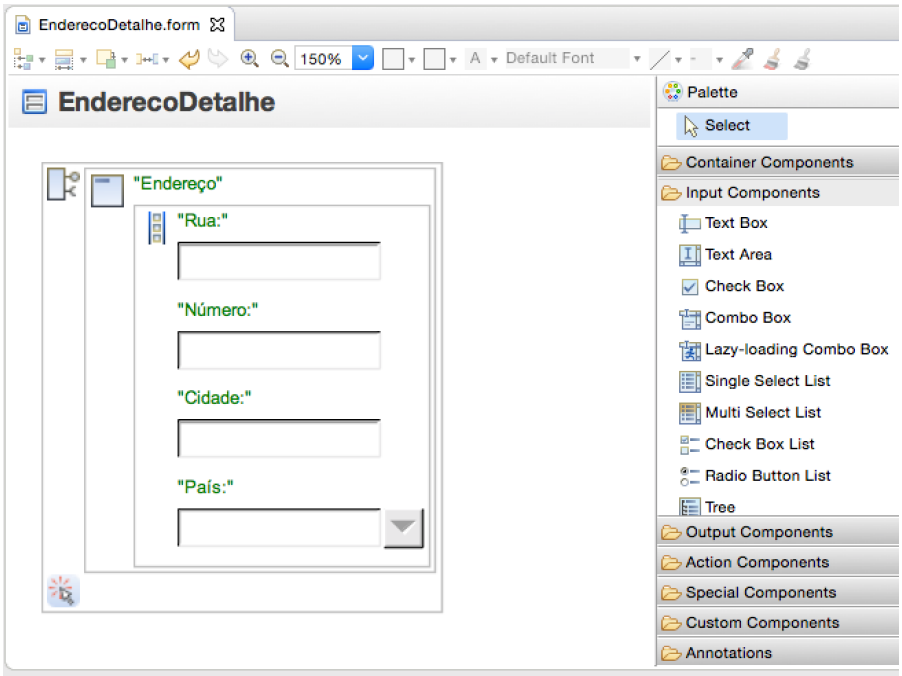

Figura 24 - Definição de Formulário EnderecoDetalhe.

Como citado anteriormente, as Definições de Formulário podem ser utilizadas tanto em Atividades de usuário, gerando *Todos*, como em Documentos. Documentos permitem que usuários acessem dados e formulários de maneira independente ao processo de negócio, ou seja, esse tipo de recurso sempre estará disponível ao usuário, independentemente do estado de execução do processo de negócio, portanto o usuário pode acessá-lo sempre que necessário.

A título de exemplo, a Figura 25 apresenta o arquivo de Definição de Documentos criado no desenvolvimento deste trabalho. Em um arquivo deste tipo, podem ser definidos vários Documentos. Cada Documento definido possui propriedades específicas e é relacionado a uma Definição de Formulário. Neste exemplo, utilizam-se os seguintes atributos na definição do Documento:

- *Name*: define o nome do Documento, que é um identificador único;

- *Parameters*: define os parâmetros do Documento que poderão ser passados à Definição de Formulário associada a ele;
- *UI Definition*: definição de Formulário associada ao Documento. Este Formulário será exibido ao usuário quando o Documento for aberto;
- *Access Rights*: determina quais usuários, ou papéis de usuário têm acesso ao Documento.

O Documento ProdutosDoc, criado na Figura 25 renderiza o formulário ListaDeProdutos, que exibe a listagem dos produtos cadastrados na aplicação. Outro detalhe é a propriedade Access Rights que determina que somente usuários com o papel de Responsável de Produção terão direito de acesso a este Documento.

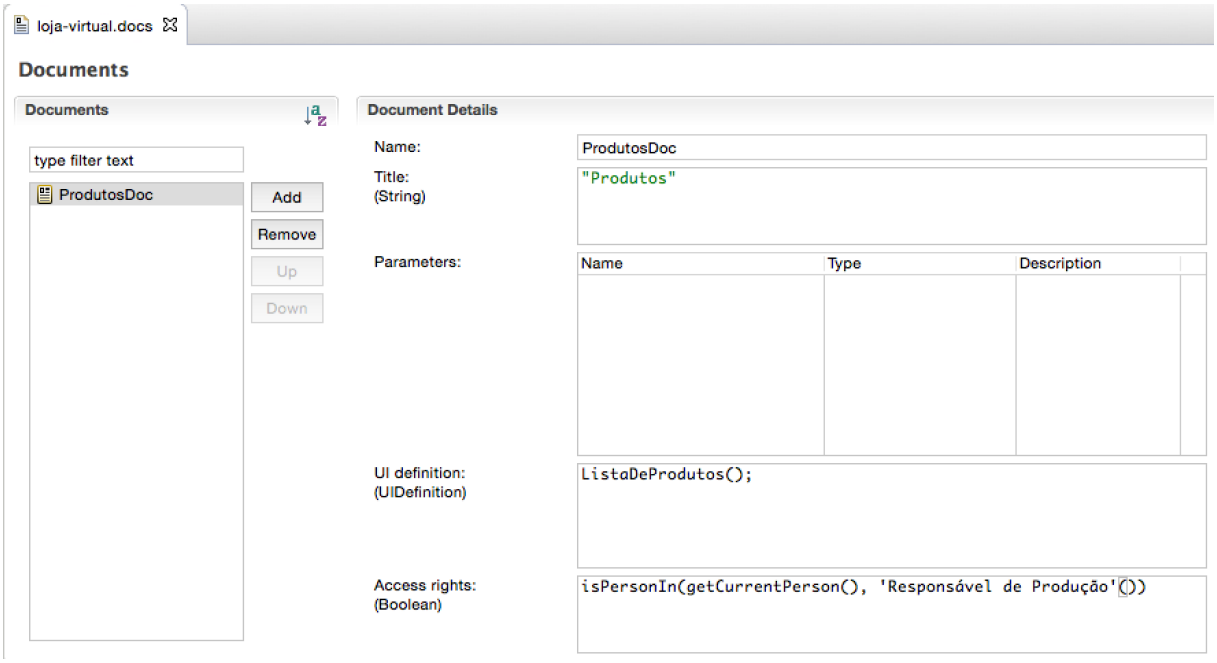

Figura 25 - Definição de Documentos.

4.3 MODELANDO PROCESSOS DE NEGÓCIO

Tendo-se concluído o processo de definição dos recursos utilizados pela aplicação LSPS, inicia-se o processo de modelagem do processo de negócio. Objetivando organização, a Definição de Processo é criada no módulo loja-virtual-principal, módulo este que em sua criação, foi configurado como executável, desta maneira sendo possível executá-lo e monitorá-lo. A Definição de Processo criada é nomeada ProcessoDePedidos e do tipo de

processo GO-BPMN, ou seja, é orientado a metas. Isso significa que neste processo são definidas metas e planos, estes últimos devem ser executados a fim de cumprir as metas.

Para gerenciamento do processamento de pedidos de uma loja virtual é definido um modelo de processo que seja claro e objetivo, tendo como meta final o processamento de pedido concluído com sucesso. Neste contexto, a Figura 26 apresenta o modelo GO-BPMN definido para o processo de negócio utilizado na aplicação Loja Virtual. Este modelo foi criado utilizando-se o editor de processos padrão provido pela ferramenta LSPS, tendo a *view* Palette em auxílio que disponibiliza os componentes adequados definidos para este tipo de modelagem.

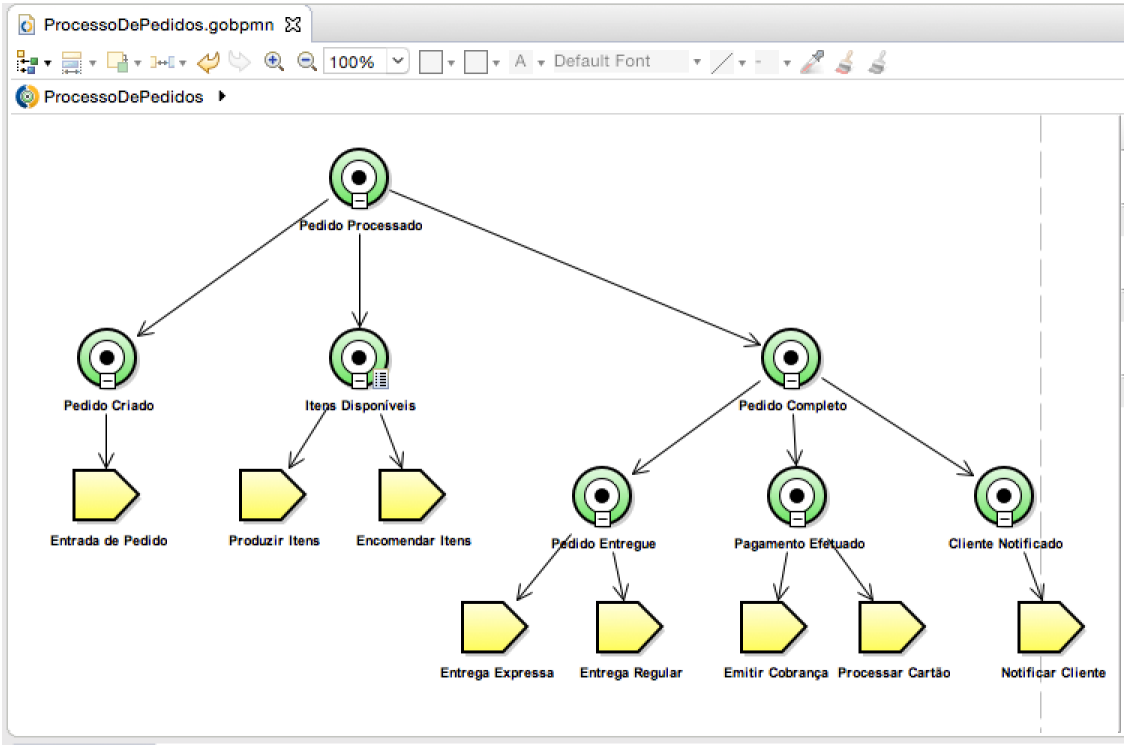

Figura 26 - Definição de Processo GO-BPMN para Loja Virtual.

Analisando-se a Figura 26 verifica-se a existência de sete diferentes metas. A principal, ao topo, definida como Pedido Processado determina se o processamento do pedido está concluído, uma vez que esta meta alcança o estado de *ACHIEVED*, ou Alcançada, o processo estará concluído, mas para isso, as metas definidas sob Pedido Processado devem ser cumpridas primeiro. As metas diretamente conectadas a Pedido Processado são Pedido Criado, Itens Disponíveis e Pedido Completo, este último aninha mais outras três metas que devem ser cumpridas.

A meta Pedido Criado determina que para que ela seja dada como Alcançada, é necessário o processamento do Plano chamado Entrada de Pedido. Este Plano de execução é

definido por um modelo BPMN. Dessa maneira, para que Pedido Criado seja Alcançada, é necessário executar o processo definido no Plano Entrada de Pedido. O Plano Entrada de Pedido, como já dito, define um processo. Este processo representa o início do processamento de pedidos, uma vez que é nele que o usuário registra um Pedido, que é o ponto inicial no processo de negócio da Loja Virtual.

Metas podem ser configuradas para executarem somente em determinadas condições. Este é o caso da meta Itens Disponíveis que tem como pré-condição de execução o fato de Pedido Criado estar em estado Alcançado. Também apresenta condição de desativação como Pedido Criado estar em estado Alcançado, já existirem itens em estoque suficientes para suprir o Pedido em processamento e ela própria, Itens Disponíveis, não estar em execução. Assim, esta meta somente será executada quando o pedido em processamento tiver sido criado e poderá ser desativada a qualquer momento que as três afirmações citadas anteriormente forem verdadeiras. Quando uma meta é desativada, ela passa a ser ignorada pelas metas que dependem desta para serem atingidas. Objetivando ilustrar a descrição realizada acima, a Figura 27 apresenta a configuração implementada para a Meta Itens Disponíveis.

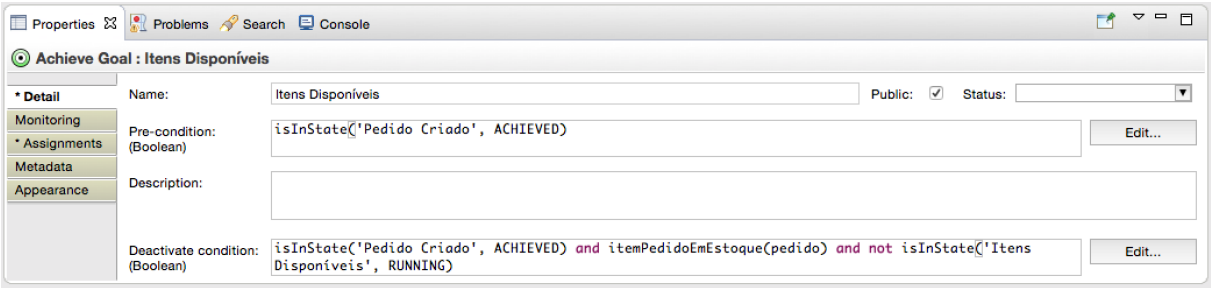

Figura 27 - Propriedades da Meta Itens Disponíveis.

Uma vez que a meta Pedido Criado é atingida e não existem itens em estoque suficientes para suprir o pedido em processamento, a meta Itens Disponíveis é então ativada e um de seus Planos será executado. Sob Itens Disponíveis existem dois Planos definidos, sendo eles Produzir Itens a ser executado quando a Loja Virtual opta por produzir os itens em falta no pedido e Encomendar Itens que é executado quando a Loja Virtual opta por adquirir os produtos em falta junto a fornecedores. Supondo que para Loja Virtual seja mais barato produzir os itens, a política padrão adotada neste modelo de processo de negócio é seguir com a produção dos itens em falta, somente sendo adquiridos os itens de fornecedores quando a produção falhar.

O plano Produzir Itens é então executado. Este é definido por um processo BPMN que possui uma Atividade iterativa sobre os itens do pedido, sendo assim, será executada para todos os itens em falta. Esta Atividade iterativa é definida como sub processo e é composto pelas Atividades Requisitar Produção de Item e Adicionar Item Produzido ao Estoque. Caso estas duas Atividades não sejam concluídas dentro de uma data limite previamente determinada, um componente *Timer* irá disparar e determinará que o item em produção será adquirido junto a um fornecedor para garantir que o pedido seja processado em tempo hábil.

A Figura 28 apresenta o modelo de processo de negócio criado para o Plano Produzir Itens que será executada para todos os produtos do Pedido que a Loja Virtual não tenha em estoque. Também se verifica a presença dos elementos BPMN Piscina e Raias, sendo uma Raia atribuída a Sistema, estando esta Raia a cargo de execução do software e outra atribuída a Trabalhador, papel de usuário criado na Definição de Estrutura Organizacional para Loja Virtual.

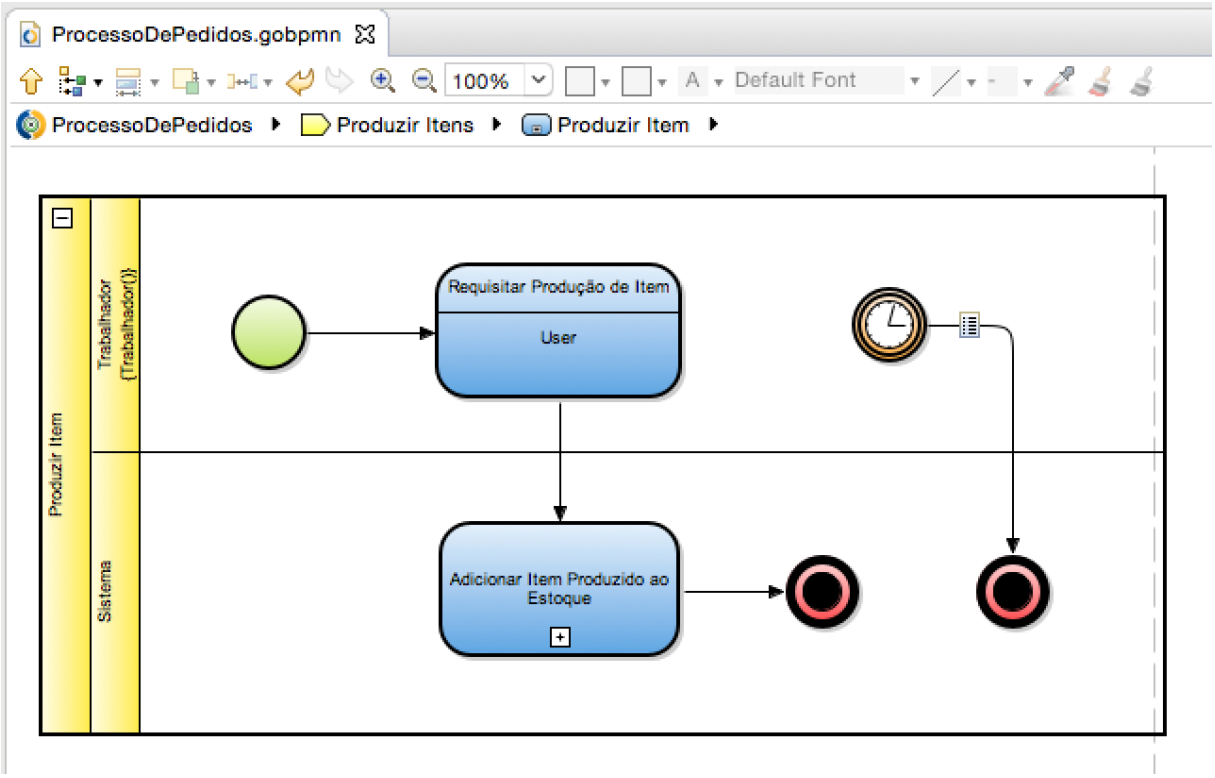

Figura 28 - Processo BPMN Produzir Item.

Caso a produção de itens falhe, por não ser concluída em tempo hábil, é ativado então o Plano Encomendar Itens. Este Plano funciona de maneira semelhante a Produzir Itens, só que neste caso a Atividade principal, Requisitar Itens em Falta Automaticamente é desempenhada pelo usuário com o Papel Responsável de Produção. Segue-se a execução

desta Atividade, as atividades de Atualizar Estoque com Itens Adquiridos e Atualizar Preço Produtos Adquiridos, ambas são iterativas e desempenhadas pelo Sistema, dessa maneira sem interação do usuário. Atualizar Preço Produtos Adquiridos é um sub processo iterativo tendo em seu sub processo uma única Atividade do tipo AtualizarPrecoProdutoTerceiro. Este tipo é customizado e foi criado especificamente para esta aplicação. Tipos customizados de Atividades são definidos em arquivos de Definição de Tarefas e podem ser implementados utilizando-se a linguagem de programação Java ou *Expression Language* da ferramenta LSPS.

Na Figura 29 é apresentada a implementação da tarefa customizada por meio da classe Java AtualizarPrecoProdutoTerceiro. Esta classe implementa a interface *ExecutableTask*, que é requisito para que uma classe possa ser utilizada para Definição de Tarefas e posteriormente referenciada em modelos BPMN. O principal método a se destacar nesta implementação é o *start*, que é chamado quando a Atividade deste tipo customizado é acionada. É neste método que ocorre a execução da tarefa. Neste caso, esta tarefa obtém o objeto "item" por meio do parâmetro recebido e então atualiza o atributo "valorUnitario" deste item.

AtualizarPrecoProdutoTerceiro.java

```
/**
 * @author djeisonselzlein
 *
 */
public class AtualizarPrecoProdutoTerceiro implements ExecutableTask {

    @Override
    public Result processInput(TaskContext context, Object input) {
        return null;
    }

    @Override
    public Result start(TaskContext context) {
        try {
            RecordHolder produto = (RecordHolder) context
                    .getParameter("item");

            Decimal valorAtual = (Decimal) produto.getProperty("valorUnitario");
            valorAtual = Decimal.valueOf(valorAtual.doubleValue()
                    + valorAtual.doubleValue() * 0.1);
            produto.setProperty("valorUnitario", valorAtual);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return Result.FINISHED;
    }

    @Override
    public void terminate(TaskContext context, TerminationReason reason) {
        // Método não utilizado nesta tarefa
    }
```

Figura 29 - Implementação Java de Tarefa Customizada.

A terceira meta necessária para que um Pedido seja processado é Pedido Completo que somente é ativado quando a meta Itens Disponíveis está em estado Alcançado ou

Desativado (caso existam itens suficientes em estoque para suprir o pedido). Esta meta, por sua vez, possui três metas filhas que precisam ser processadas para que seja Alcançada:

- Pedido Entregue: meta cujo objetivo é processar o método de entrega dos itens do Pedido. Ativado quando sua meta mãe (Pedido Completo) é ativada. Pode ser executada em dois Planos diferentes, Entrega Regular e Entrega Expressa. Caso um pedido seja marcado como "urgente", o Plano Entrega Expressa será executado, caso contrário, Entrega Regular. Ambos Planos são definidos por processos BPMN;
- Pagamento Efetuado: meta que busca processar a forma de pagamento do Pedido. Ativada também juntamente a meta Pedido Completo. Executada em dois Planos diferentes, Emitir Cobrança ou Processar Cartão. Processar Cartão é acionado quando o método de pagamento selecionado é por meio de cartão de crédito, caso contrário, Emitir Cobrança é executado, gerando uma fatura impressa a ser entregue no endereço de entrega do Pedido;
- Cliente Notificado: meta que visa a notificação ao cliente que realizou a solicitação do Pedido. Ativado quando ambas metas Pedido Entregue e Pagamento Efetuado encontram-se Alcançadas. O plano a ser executado nesta meta é Notificar Cliente, cujo processo é iniciado pelo sistema que disponibiliza um *Todo* com uma simples mensagem de notificação informando que o Pedido solicitado foi processado e enviado ao endereço de entrega provido.

4.4 EXECUÇÃO E MONITORAMENTO DE PROCESSOS

Após processo de definição de recursos relevantes ao processo de negócio e modelagem deste último, é momento de se carregar estes modelos no servidor embarcado e executá-los visando demonstração da funcionalidade como um todo, abrangendo execução e monitoramento do processo modelado.

Primeiramente, é iniciado o servidor embarcado, provido pela ferramenta LSPS. Se iniciado com sucesso, o servidor irá imprimir resultados do processo de publicação da aplicação LSPS. Na Figura 30 está representada a saída impressa pelo servidor embarcado ao final do processo de início do mesmo. Podem ser verificados os endereços nos quais o

servidor registrou determinadas aplicações. Neste exemplo é importante ressaltar o endereço http://localhost:8080/loja-virtual no qual está registrada a aplicação Loja Virtual e o endereço http://localhost:8080/lsps-management-vaadin que por sua vez permite acesso ao console de administração da aplicação LSPS criada.

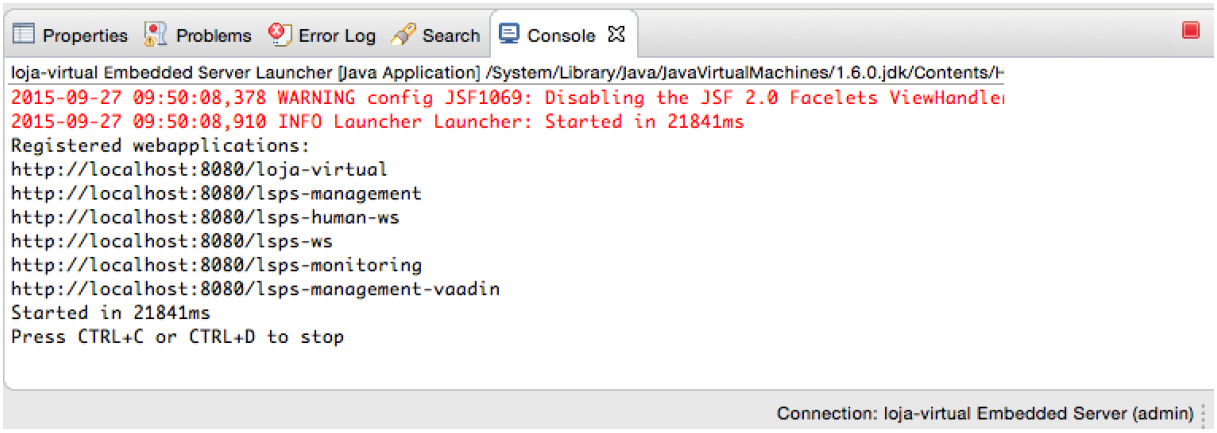

Figura 30 - *View* Console exibindo saída do servidor embarcado.

Uma vez que o servidor embarcado é iniciado, é necessário que os módulos da aplicação sejam carregados no servidor para então serem acessados pela interface Web registrada acima. A aplicação LSPS é acessível a qualquer momento após o servidor ser iniciado, porém não haverá módulos para execução, caso não sejam carregados, e a interface não conterá funcionalidade alguma, além daquelas geradas ao criar-se uma aplicação LSPS.

Objetivando-se a disponibilização dos modelos e processos desenvolvidos, é feito o carregamento (*upload*) dos módulos loja-virtual-base e loja-virtual-principal ao servidor. Este processo geralmente leva poucos segundos para ser concluído e as saídas de resultado serão impressas também na *view* Console. Uma vez que os módulos forem carregados, é possível então acessar a aplicação pelo endereço http://localhost:8080/loja-virtual.

O modelo de processos desenvolvido é baseado em uma estrutura organizacional, sendo assim, é necessária a definição de usuários de cada Papel de usuário que esteja envolvido no processo de negócio. Quando uma aplicação LSPS é criada, três usuários padrão são criados: *admin* (administrador da aplicação), *guest* (convidado) e *processAgent* (agente de processos). No caso da aplicação Loja Virtual assim como no cenário de aplicações comerciais, são necessários mais usuários com diversos papéis.

O console de administração LSPS, que por padrão é registrado no endereço http://localhost:8080/lsps-management-vaadin, permite o gerenciamento de diversos recursos, entre eles, usuários que têm acesso a aplicação. Para esta demonstração foram criados

usuários utilizando-se nomes fictícios para cada um dos papéis envolvidos no processo, como representado na Quadro 1.

Quadro 1 - Usuários da aplicação Loja Virtual

Usuário	Nome	Papel Modelado
marcela	Marcela Paes	Cliente
joao	João Lima	Responsável de Produção
tadeu	Tadeu Farias	Trabalhador
valter	Valter Guimarães	Agente de Entregas
maria	Maria Góes	Agente Contábil

Como definido no modelo de processo de negócio, o processo é iniciado quando um usuário com papel Cliente registra um pedido na Loja Virtual. Nesta demonstração, a usuária Marcela Paes desempenha este papel registrando um pedido. Inicialmente é realizado o processo de autenticação na aplicação informando usuário e senha definidos. Acessando o menu *Run Model* se tem acesso aos modelos disponíveis para execução de acordo com carregamento anteriormente feito. Ao acessar este menu e clicar no módulo listado chamado loja-virtual-principal, o mecanismo de execução do LSPS cria uma nova instância de processo do modelo selecionado. Dessa maneira, o modelo ProcessoDePedidos entra em estado de execução na instância criada executando-se as tarefas previstas neste modelo a fim de atingir a meta especificada.

A primeira atividade no processo ProcessoDePedidos é o cumprimento da meta Pedido Criado que é atingida pela execução do Plano Entrada de Pedido. Neste Plano está definida a Atividade de Registrar Pedido, que é realizada pelo usuário que iniciou a execução do modelo de processo. Esta atividade é disponibilizada para cumprimento por meio de um Afazer a ser exibido no menu *Todo List*. Na Figura 31, pode ser visualizado o formulário de entrada de pedidos exibido ao ser iniciado o Afazer pela usuária Marcela Paes. Após devido preenchimento, ocorre a submissão deste formulário pela usuária. Tendo-se submetido este formulário, o Plano Pedido Criado é dado como Atingido. Na Figura 31 pode-se verificar o formulário renderizado em Vaadin para entrada de pedidos renderizado ao usuário quando o Afazer é ativado.

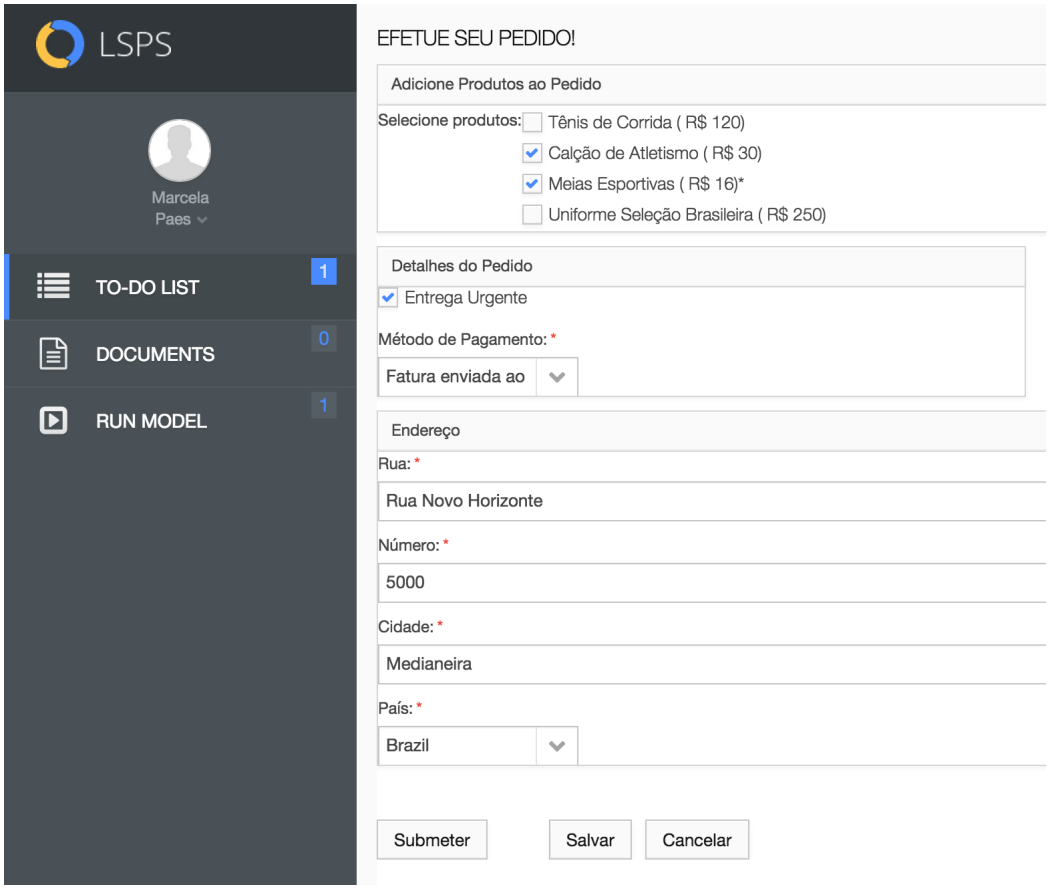

Figura 31 - Formulário de Submissão de Pedidos.

Após conclusão da execução do Plano Entrada de Dados, a meta Pedido Criado é concluída. A meta Itens Disponíveis é então ativada, sendo que sua condição para ativação é a conclusão da meta Pedido Criado. Considerando que a política padrão estabelecida é que itens em falta sejam produzidos, o Plano a ser executado para conclusão desta meta é Produzir Itens. Como pode ser visualizado na Figura 31, o pedido em processamento foi configurado como urgente e portanto, se a produção de itens não for concluída dentro da data limite, a produção é tida como falha e os itens em falta devem ser adquiridos junto a terceiros. Este caso é ilustrado nesta execução e pode ser visualizado por meio do monitorado de processos.

O monitoramento de processos na ferramenta LSPS é feito por meio da perspectiva *Management* ou Gerenciamento. Esta perspectiva dispõe de *views* especificas que auxiliam o acompanhamento da execução de modelos de processos de negócio. Esta perspectiva também permite o gerenciamento de módulos no servidor, visualização de pessoas relacionadas ao processo, acompanhamento em tempo real da execução dos diagramas modelados.

Um dos recursos mais interessantes presentes na perspectiva de Gerenciamento é a *view Model Instances* ou Instâncias de Modelo que permite o rastreio e manipulação das instâncias de modelo criadas. Também permite a visualização do estado de cada instância,

quando foi iniciada e a qual modelo se refere. É possível ainda acessar detalhes sobre cada instância listada. Esses detalhes trazem propriedades da instância tais como, quem a iniciou, variáveis do modelo e seus respectivos valores também atualizados enquanto o processo está em execução e por último, mas mais importante, o processo em si com as metas e planos de execução e seus respectivos estados de execução.

Na Figura 32 pode-se visualizar o estado atual de execução do processo de negócio da maneira como se tem até o ponto de execução atual, depois de Marcela realizar a solicitação de Pedido e a Produção de Itens falhar acionando o Plano Encomendar Itens. Neste diagrama os estados dos componentes do processo são identificados por cores. Assim sendo, verifica-se que a meta Pedido Criado e seu plano de execução encontram-se em estado "Alcançado", tendo sido concluídos com sucesso. As metas Pedido Processado e Itens Disponíveis, bem como o plano de execução Encomendar Itens encontram-se "Em Execução", sendo que o plano de execução Produzir Itens possui estado "Falhou". "Pronto" é o estado em que se encontra a meta Pedido Completo, sendo que aguarda o término da execução de Itens Disponíveis para que seja então ativada. As metas filhas de Pedido Completo estão "Inativas" pois aguardam ativação de sua meta mãe para que sejam processadas.

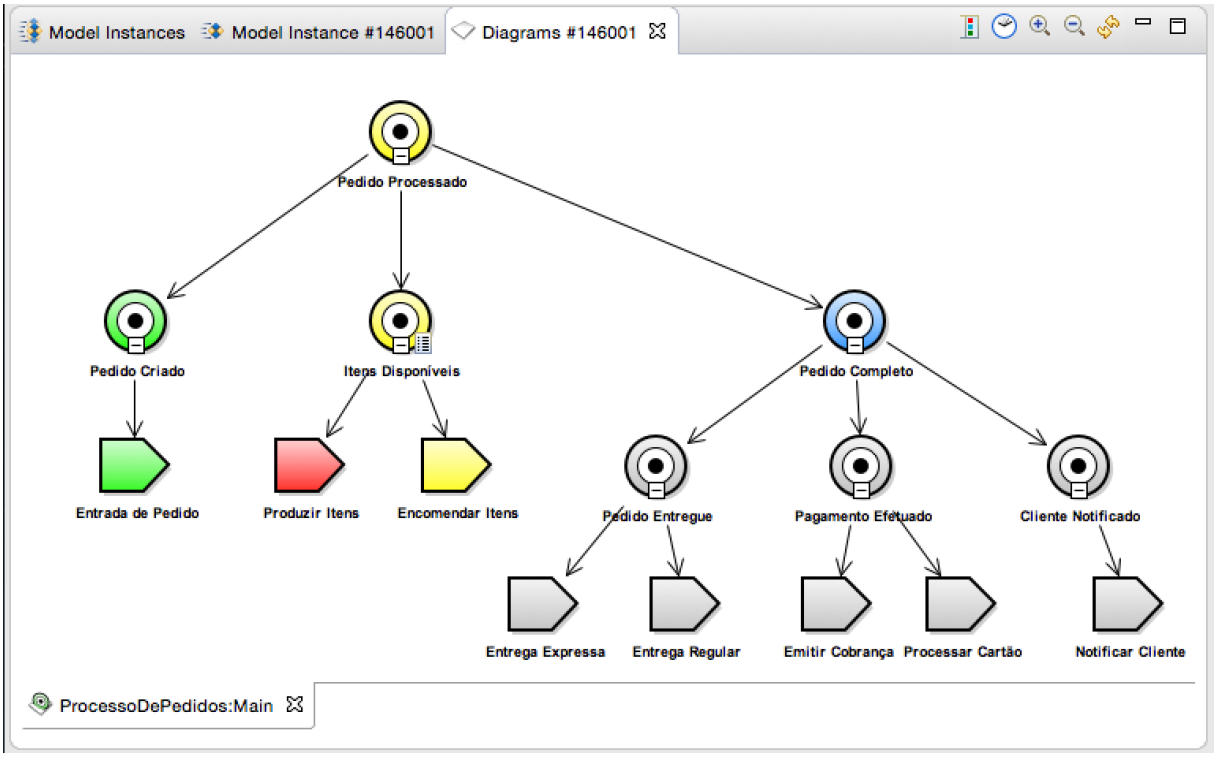

Figura 32 - Monitoramento do processo ProcessoDePedidos.

Como descrito anteriormente, o Plano Encomendar Itens tem como executor inicial um usuário cujo Papel seja Responsável de Produção. Para que o processo prossiga, é

necessário que um usuário com este Papel se autentique na aplicação e submeta a requisição automática de itens junto a fornecedores. Nesta aplicação o usuário João Lima possui este Papel e deverá submeter este formulário.

Ainda, na visualização do diagrama de execução de processos, é possível navegar entre Planos de execução e verificar suas respectivas execuções de maneira detalhada. Quando o usuário João submeter formulário de requisição de produtos o sistema executará as tarefas Atualizar Estoque Com Itens Adquiridos e Atualizar Preço Produtos Adquiridos. A tarefa Atualizar Preço Produtos Adquiridos é a tarefa customizada implementada em linguagem Java especificamente para este modelo, mas que pode ser reusada em modelos posteriores. A Figura 33 apresenta como fica o diagrama de monitoramento de processo após submissão do formulário de requisição de produtos e execução das tarefas do sistema. A coloração esverdeada indica o estado de conclusão da execução, assim como no diagrama GO-BPMN anterior.

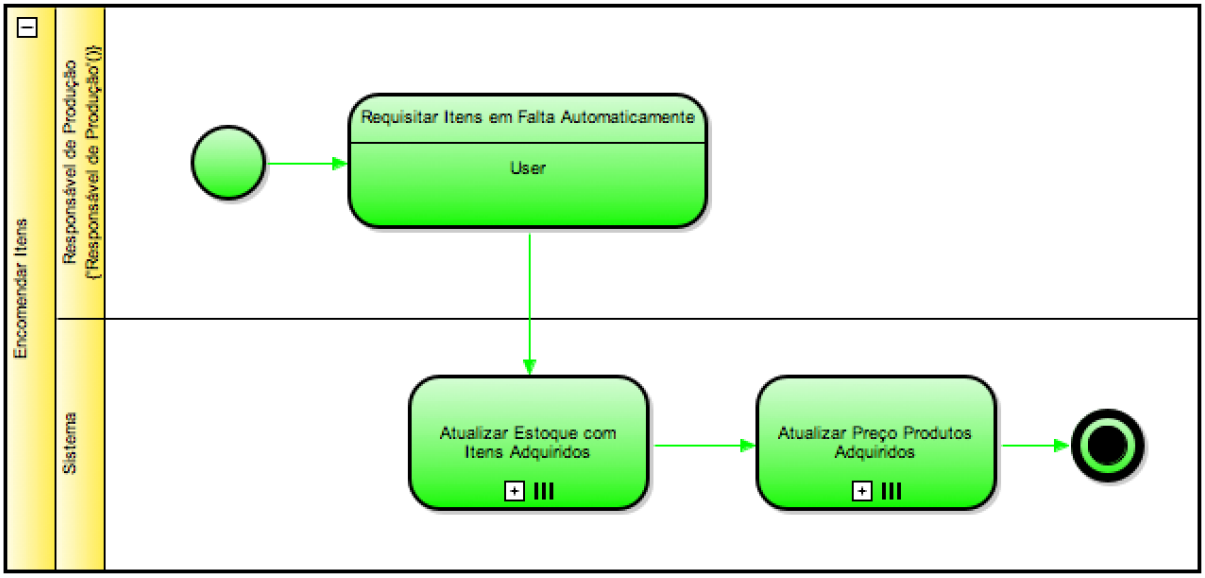

Figura 33 - Plano Encomendar Itens Executado Com Sucesso.

Após conclusão do Plano Encomendar Itens, a meta Itens Disponíveis é definida como Alcançada e é ativada a meta Pedido Completo. Pedido Completo possui três metas filhas sendo que as metas Pedido Entregue e Pagamento Efetuado são ativadas imediatamente após a meta mãe ser ativada. A última meta, porém, aguarda a execução das anteriores, uma vez que essa é a meta Cliente Notificado, cujo Plano de execução contém uma única tarefa a ser executada pelo cliente que submeteu o pedido. Esta tarefa somente exibe uma notificação em tela informando o cliente sobre a conclusão do processamento do Pedido realizado.

A meta Pedido Entregue pode ser realizada por dois diferentes Planos de execução Entrega Expressa e Entrega Regular, dependendo se um Pedido é urgente ou não. Esses dois Planos possuem modelos de processo BPMN muito semelhantes, ambos possuem uma atividade inicial Atualizar Estoque, executada pelo Sistema que deverá deduzir do estoque as quantidades de itens que estão no Pedido em processamento e também a atividade Enviar Produtos ao Cliente que é executada por um Agente de Entrega. A diferença entre esses dois Planos é que Entrega Expressa aciona uma agência de entregas enfatizando a urgência da entrega enquanto o outro Plano segue uma metodologia de entrega regular.

Pagamento Efetuado é a meta que busca concretizar monetização do Pedido em processamento. Também pode ser desempenhado por dois diferentes Planos. Emitir Cobrança é executado quando o cliente opta por receber a cobrança no endereço de entrega do pedido. Esse Plano é composto por duas atividades, sendo a primeira executada por um Agente Contábil e chamada Preparar Cobrança. Nesta atividade, o Agente Contábil deverá anexar ao Afazer um documento de cobrança que será enviado ao Cliente juntamente com a entrega na atividade seguinte Enviar Cobrança, desempenhada por um Agente de Entregas. O segundo plano possível é Processar Cartão, executado quando o Cliente opta por pagamento em cartão de crédito. Este segundo Plano é mais simples, sendo composto somente pela Atividade de Cobrar Cartão de Crédito Manualmente, onde um Agente Contábil deverá submeter a cobrança ao cartão de crédito informado pelo Cliente ao realizar o Pedido.

Na situação demonstrativa criada neste trabalho, os Planos de Entrega Expressa e Emitir Cobrança são executados para cumprimento de suas respectivas metas pelos usuários fictícios criados que portam tais Papeis necessários para execução das atividades. Uma vez executados estes Planos, as metas Pedido Entregue e Pagamento Efetuado são então Atingidas.

A meta Cliente Notificado tem como pré-condição para ativação que as metas Pagamento Entregue e Pagamento Efetuado sejam Atingidas. Neste momento é ativada a meta Cliente Notificado e o Plano de execução para esta meta define uma única atividade de usuário, onde o Cliente visualiza uma confirmação de que o seu pedido foi processado. Nesta aplicação, Marcela visualiza o resultado da sua solicitação e submete o formulário apresentado com tal resultado.

Neste ponto, todas as metas e planos de execução necessários foram atingidas e/ou executadas. Assim sendo, o Pedido encontra-se processado e a meta principal é definida como executada. O resultado desta execução é apresentado no diagrama de monitoramento do processo e está representado na Figura 34.

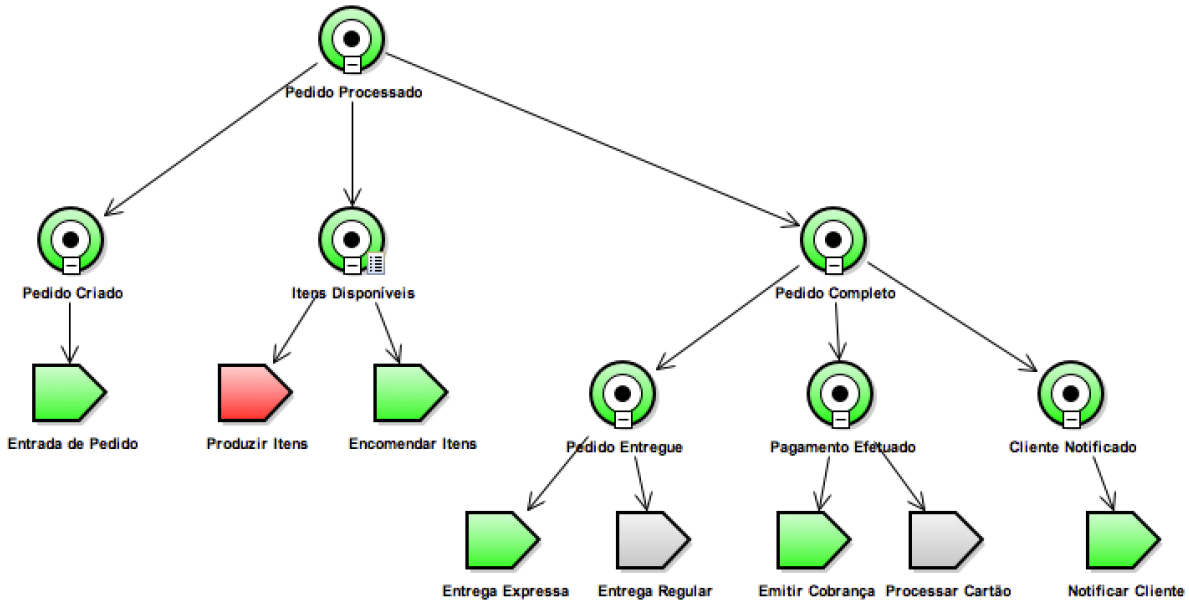

Figura 34 - ProcessoDePedidos Concluído

A título de demonstração, neste trabalho somente uma instância de processo foi executada, mas em uma situação real de utilização da ferramenta, várias instâncias podem ser executadas simultaneamente e monitoradas de maneira igual ao exemplificado.

5 CONSIDERAÇÕES FINAIS

Neste capítulo serão apresentadas as conclusões e benefícios abstraídos a partir desenvolvimento deste trabalho e aplicação demonstrativa. Também neste capítulo são apresentadas as sugestões para trabalhos futuros envolvendo ferramentas de desenvolvimento de aplicações BPM que façam uso de uma abordagem orientada a modelos.

5.1 CONCLUSÃO

Um dos principais benefícios da utilização da abordagem apresentada é o seu nível de abstração. Neste trabalho ocorre a modelagem de um processo de negócio de um segmento específico do mercado, mas esta abordagem de desenvolvimento pode ser utilizada nas mais diversas áreas de aplicação. Isto é verdade devido ao fato de que todo negócio é definido por um processo, ou seja, toda organização possui uma forma de trabalho definida e etapas que devem ser cumpridas para realização dos objetivos desta. A utilização da modelagem de processos de negócio e execução acomoda esta gama de possibilidades.

Dentre os benefícios da utilização de uma ferramenta como LSPS verifica-se a agilidade e clareza na criação de aplicações BPM e modelagem de processos. Isso é permitido pelo mecanismo de execução que permite a execução e monitoramento de modelos, assim sendo, o próprio diagrama do modelo de processo criado é executado e monitorado, sem obrigatoriedade de manipulação direta de código fonte.

Dinamismo é outra característica importante proporcionada pelo LSPS. Isso se verifica na criação de modelos de negócio, que provê ao usuário da ferramenta inúmeros recursos para desempenho desta atividade. Também é importante destacar a possibilidade da implementação em Java de tarefas customizadas, permitindo a incorporação destas implementações como atividades dentro do processo BPMN.

No ambiente corporativo, as organizações podem ser muito beneficiadas pela implementação de soluções de software que coloquem BPM em prática, como o LSPS. O gerenciamento de processo de negócios permite a administradores o controle sobre o processos. Quando o processo de negócio é modelado conforme é desempenhado sem otimização, ficam claros os pontos falhos do processo, onde está ocorrendo desperdício de

recursos e os pontos que trazem prejuízo ou causam falha na execução do processo. Desta maneira, a implementação da prática BPM permite organizações otimizarem suas estruturas.

5.2 TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO

É proposto como continuação deste trabalho, o estudo sobre outras ferramentas similares ao LSPS que possibilitem o desenvolvimento de aplicações BPM, utilizando-se uma abordagem orientada a modelos a fim de conhecimento sobre alternativas e comparativo de peculiaridades de cada ferramenta.

Outro tópico a ser estudado são as interfaces de conexão do LSPS com ferramentas externas também importantes no ciclo de desenvolvimento de software, tais como bancos de dados, para persistência dos dados gerados e necessários às aplicações criadas e servidores web para publicação destas aplicações. Este tópico destaca-se devido ao fato de que é de essencial importância que ferramentas como o LSPS possam se conectar a sistemas legados de maneira a absorver dados destes e causar o mínimo impacto à organização que o utilize e também proporcionando um paralelismo entre aplicações

REFERÊNCIAS

Barrios, B. **Gerência de Processos de Negócio: Benefícios e Desafios!**, 2013. Disponível em <http://www.bpmvision.com.br/gerencia-de-processos-de-negocio-beneficios-e-desafios/>. Acesso em 25 de Ago. de 2015.

BPM.Com. **Whitestein Technologies**, 2015. Disponível em <http://bpm.com/vendors/113-featured-vendors/672-vendors-whitestein-technologies>. Acesso em 28 de Ago. de 2015.

Buhler, P. A. **Successful Business Process Management U.S. Department of Veteran Affairs**, 2008.

Caelum. **O que é Java**, 2015. Disponível em <http://www.caelum.com.br/apostila-java-orientacao-objetos/o-que-e-java/#2-2-uma-breve-historia-do-java>. Acesso em 27 de Set. de 2015.

Carvalho, M. S. **Vaadin: Programação para a web com jeito de desktop**, 2014. Disponível em <http://www.devmedia.com.br/vaadin-programacao-para-a-web-com-jeito-de-desktop-revista-java-magazine-95-parte-1/22979>. Acesso em 27 de Set. de 2015.

Eclipse. **Eclipse Documentation - Current Release**, 2013. Disponível em <http://help.eclipse.org/mars/index.jsp>. Acesso em 1 de Set. de 2015

France, R., e Rumpe, B. **Model-driven Development of Complex Software: A Research Roadmap**. Minneapolis, MN, EUA, 2007.

Grönroos, M. **Book of Vaadin** (Vol. 1). Turku, Finlândia: Vaadin Ltd, 2015.

Janssen, C. **Business Process Management Software (BPMS)**, 2015. Disponível em <https://www.techopedia.com/definition/28519/business-process-management-software-bpms>. Acesso em 28 de Ago. de 2015.

Lucrédio, D. **Uma Abordagem Orientada a Modelos para Reutilização de Software**. São Carlos, São Paulo, Brasil: USP - São Carlos, 2009.

Nogueira Arantes, R. **Introdução ao Business Process Modeling Notation (BPMN)**, 2014. Disponível em <http://www.devmedia.com.br/introducao-ao-business-process-modeling-notation-bpmn/29892>. Acesso em 25 de Ago. de 2015.

Object Management Group. **BPMN 2.0**, 2011. Disponível em <http://www.omg.org/spec/BPMN/2.0/>. Acesso em 16 de Jul. de 2015.

Palmer, N. **What is BPM?**, 2015. Disponível em <http://bpm.com/what-is-bpm>. Acesso em 16 de Jul. de 2015.

Pereira, A. P. **O que é Java?**, 2009. Disponível em <http://www.tecmundo.com.br/programacao/2710-o-que-e-java-.htm>. Acesso em 27 de Set. de 2015.

Lima, C. D., e Silva Júnior, J. P. **Relacionando Engenharia de Requisitos à Engenharia de Software Orientada a Modelos**, 2009. Disponível em <http://www.cin.ufpe.br/~in1020/arquivos/monografias/2009_1/carlos_josias.pdf>. Acesso em 16 de Jul. de 2015.

Truyen, F. **The Fast Guide to Model Driven Architecture**, 2006. Disponível em <http://www.omg.org/mda/mda_files/Cephas_MDA_Fast_Guide.pdf>. Acesso em 27 de Ago. de 2015.

Whitestein Technologies AG. **Expression Language**. Cham, Zug, Suíça, 2015.

Whitestein Technologies AG. **Getting Started Guide** (1 ed.). Cham, Zug, Suíça, 2015.

Whitestein Technologies AG. **GO-BPMN Modeling Language**. Cham, Zug, Suíça, 2015.

Whitestein Technologies AG. **LSPS Training Foundations**. Cham, Zug, Suíça, 2014.

Whitestein Technologies AG. **Process Application Guide**. Cham, Zug, Suíça, 2015.

Whitestein Technologies AG. **Process Design Suite User Guide**. Cham, Zug, Suíça, 2015.

Whitestein Technologies AG. **Standard Library Specification**. Cham, Zug, Suíça, 2015.