

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

EDIJUNIOR RELL

IMPLEMENTAÇÃO DE PROCESSOS DE NEGÓCIOS COM BPEL

TRABALHO DE DIPLOMAÇÃO

MEDIANEIRA

2013

EDIJUNIOR RELL

IMPLEMENTAÇÃO DE PROCESSOS DE NEGÓCIOS COM BPEL

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

Orientador: Prof. M. Sc. Alan Gavioli

MEDIANEIRA

2013



TERMO DE APROVAÇÃO

IMPLEMENTAÇÃO DE PROCESSOS DE NEGÓCIOS COM BPEL

Por

EdijuniorRell

Este Trabalho de Diplomação (TD) foi apresentado às **10:30** h do **dia 21 de fevereiro de 2013** 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, Câmpus Medianeira. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado com louvor e mérito.

Alan Gavioli
Prof.(a) Orientador(a)

Nelson M. Betzek
Membro titular

Ricardo Sobjak
Membro titular

AGRADECIMENTOS

Agradeço primeiramente a meus pais e irmã pelo apoio e incentivo a concluir este curso, pelas suas lições de vida em conselhos e ações que me deram força para nunca desistir dos sonhos e desejos.

Acredito ser o conhecimento a base de toda grande realização, e por isso aos professores tenho muito a agradecer, que são grandes mestres da educação e a base do conhecimento que foi necessário para a realização deste, partiu destes.

Por fim, agradeço de forma especial a amiga Karen Luana, pelo apoio no início dessa caminhada, nos quais pareceu ser a parte mais complicada, ela sempre deu forças para continuar, e também aqueles que estiveram sempre apoiando e foram a melhor companhia nas aulas e fora delas, aos colegas Felipe Joseph Vettorazzi, Aline Ferreira, Matheus Hoyer Bresciani, Fernanda Cristina Girelli, Eduardo Sganderla, Thainá Mariani, Fabio Kniphoff.

RESUMO

RELL, Edijunior. **Implementação de processos de negócios com BPEL**. 2012. 68. Trabalho de Conclusão de Curso de Tecnologia em Análise e Desenvolvimento de Sistemas – Universidade Tecnológica Federal do Paraná. Medianeira, 2013.

Este trabalho apresenta o resultado de pesquisas relacionadas a conceitos importantes e necessários para o entendimento da estrutura do desenvolvimento de processos de negócios. Para facilitar o entendimento, expõe conceitos úteis de um conjunto de tecnologias como SOA, *Web Services* e BPEL focado no estudo do funcionamento dos processos de negócios. Ainda aborda uma implementação de um processo usando a Oracle Soa Suíte 11g, descrevendo suas funcionalidades no processo e destacando a importância do seu uso no meio corporativo. Por fim, um estudo de caso detalhado com cada parte do uso das tecnologias envolvidas no desenvolvimento prático do processo de negócio abordado neste trabalho.

Palavras-chaves: Oracle Soa Suíte 11g. WSDL. SOA. *Web Services*.

ABSTRACT

RELL, Edijunior. **Implementação de processos de negócios com BPEL**. 2012. 68. Trabalho de Conclusão de Curso de Tecnologia em Análise e Desenvolvimento de Sistemas – Universidade Tecnológica Federal do Paraná. Medianeira, 2013.

This study presents the results of research related to important concepts and necessary for understanding the structure of the development of business processes. To facilitate understanding, exposes useful concepts for a set of technologies such as SOA, Web Services and BPEL focuses on the study of the functioning of business processes. Also discusses an implementation of a process using the Oracle Soa Suite 11g, describing the features in the process and highlighting the importance of its use in the corporate environment. Finally, a detailed case study with each party's use of the technologies involved in the practical development of the business process discussed in this work.

Keywords: Oracle Soa Suíte 11g. WSDL. SOA. Web Services.

LISTA DE TABELAS

Tabela 1 – Obstáculos comuns na implementação de um processo de negócio	19
Tabela 2 – Símbolos de um fluxograma	21

LISTA DE QUADROS

Quadro 1 – Descrição do estudo experimental utilizado	33
---	----

LISTA DE FIGURAS

Figura 1 – Funcionamento da orquestração.....	30
Figura 2 – Funcionamento da coreografia.....	30
Figura 3 – Visão geral dos componentes da SUITE.....	34
Figura 4 – Janela inicial para criação de uma nova aplicação no JDeveloper 11g ...	37
Figura 5 – Escolha das tecnologias envolvidas nos projetos da aplicação	38
Figura 6 – Sugestão de templates do JDeveloper para se trabalhar com projeto SOA	39
Figura 7 – Criação de um processo BPEL no JDeveloper 11g.....	40
Figura 8 – Importação de schema exemplo para um processo BPEL.....	41
Figura 9 – Paleta de componentes utilizáveis em um processo BPEL.....	42
Figura 10 – Localização do WSDL para serviço de consulta crédito do cliente... ..	43
Figura 11 – Configuração do link para consulta de crédito.....	43
Figura 12 – Aspecto do componente scope no projeto.....	44
Figura 13 – Atribuição de regras e definições de funções envolvendo variáveis.. ..	45
Figura 14 – Uso de funções na manipulação de variáveis.....	46
Figura 15 – Configurações do componente Invoke.....	47
Figura 16 – Estrutura do scopo responsável por acessar uma atividade de consulta de crédito.....	47
Figura 17 – Representação do componente Flow.....	48
Figura 18 – Partner Link para primeiro fornecedor com especificação de WSDL.. ..	49
Figura 19 – Configuração para invocação de serviço no primeiro fornecedor.....	50
Figura 20 – Configuração do componente Receive para um Partner Link especificado.....	51
Figura 21 – Estrutura do primeiro sequence do componente Flow para execução de tarefas paralelas.....	52
Figura 22 – Estrutura do segundo sequence do componente Flow para execução de tarefas paralelas.....	53
Figura 23 – Componente Flow com duas sequencias montadas exemplificando o acesso simultâneo a dois serviços.....	53
Figura 24 – Processo de negócio completo com expansão dos componentes de agrupamento utilizados.....	54
Figura 25 – Opção de deploy do projeto desenvolvido no JDeveloper 11g.. ..	55
Figura 26 – Controle de versões para deploy do projeto.....	56
Figura 27 – Pesquisa de servidores ativos para deploy do projeto.....	57
Figura 28 – Servidor ativo e pronto para receber o novo processo.....	58
Figura 29 – Confirmação de deploy do processo no servidor de aplicações	58

Figura 30 – Estrutura do processo de negócio no servidor referenciando os Web Services utilizados.....	59
Figura 31 – Entrada de dados para teste de Web service..	59
Figura 32 – Web Services sendo executados no console BPEL prontos para receber requisitos de processamento de informações.....	60
Figura 33 – Acompanhamento da execução do processo de negócio.....	61
Figura 34 – Modelagem do processo de negócio vista através do console do servidor Web logic.....	62

LISTA DE SIGLAS

BPEL	<i>Business Process Execution Language.</i>
BPEL4WS	<i>Business Process Execution Language For Web Services</i>
BPMN	<i>Business Process Modeling Notation</i>
CCM	<i>Corba Component Model.</i>
CEO	<i>Chief Executive Officer.</i>
CORBA	<i>Common Object Request Broker Archicture</i>
DCE	<i>Distributed Computing Environment</i>
EAI	<i>Enterprise Application Integration</i>
EJB	<i>Enterprise Java Beans</i>
ESB	<i>Enterprise Service Bus</i>
JCA	<i>Java Connector Architecture</i>
JMS	<i>Java Messaging System</i>
OASIS	<i>Organization for the Advancement of Structured Information</i>
ORB	<i>Objetct Request Brokers</i>
OSB	<i>Oracle Service Bus</i>
RMI	<i>Remote Method Invocation</i>
RPC	<i>Remote Procedure Call</i>
SOA	<i>Service Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
UDDI	<i>Universal Description Discovery and Integration</i>
WSDL	<i>Web Services Description Language</i>
WSFL	<i>Web Services Flow Language</i>
XLANG	<i>Extensible Language</i>
XML	<i>Extensible Markup Language</i>
XPATH	<i>XML Path language</i>
XSLT	<i>Extensible Stylesheet Language Transformation</i>

SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	OBJETIVO GERAL.....	14
1.2	OBJETIVOS ESPECÍFICOS	14
1.3	JUSTIFICATIVA	14
1.4	ESTRUTURA DO TRABALHO	15
2	PROCESSOS DE NEGÓCIOS COM BPEL	16
2.1	INTRODUÇÃO.....	16
2.2	MODELAGEM DE PROCESSO DE NEGÓCIO	18
2.3	FLUXOGRAMAS	21
2.4	CRIAÇÃO DE FLUXOGRAMAS FUNCIONAIS	22
2.5	AUTOMATIZAÇÃO DE PROCESSOS DE NEGÓCIOS.....	22
2.6	SOA	23
2.6.1	Web Services	25
2.7	O PADRÃO BPEL	26
2.8	CARACTERÍSTICAS DO BPEL	27
2.9	BPEL PARA COMPOSICAO DE SERVIÇOS	28
2.10	ORQUESTRAÇÃO E COREOGRAFIA	29
2.11	PROCESSOS EXECUTÁVEIS E PROCESSOS ABSTRATOS.....	31
2.12	ORACLE SOA SUITE 11G	32
2.12.1	Oracle Fusion Middleware	33
2.12.2	Jdeveloper para desenvolvimento com BPEL	33
2.12.3	Oracle Service Bus (OSB)	34
2.12.4	Oracle BPEL Process Manager.....	34
3	ESTUDO EXPERIMENTAL	34
3.1	DESENVOLVIMENTO DA APLICAÇÃO.....	35
3.1.1	Criação de um projeto SOA.....	36
3.1.2	Criação de um processo BPEL	38
3.1.3	Conexão de links externos para consumir os serviços.....	40
3.1.4	Organização do processo de negócio com escopos	43

3.1.5	Invocação de serviços	43
3.1.6	Trabalho com fluxos paralelos.....	47
3.1.7	Deploy do projeto	55
4	CONSIDERAÇÕES FINAIS	65
4.1	CONCLUSÃO	65
4.2	TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO	66
	REFERÊNCIAS BIBLIOGRÁFICAS	67

1 INTRODUÇÃO

Atualmente, a ascensão da Internet e de tecnologias voltadas a sistemas distribuídos, instiga os desenvolvedores a oferecerem soluções cada vez mais ideais ao mundo corporativo baseado nessas tecnologias. Esses sistemas buscam facilitar o gerenciamento de empresas, auxiliando na tomada de decisões importantes com base em dados específicos, por exemplo, uma análise nos processos de negócios da empresa. Através dessa análise é possível melhorar, por exemplo, o atendimento ao cliente ou até mesmo eliminar uma despesa desnecessária que estava imperceptível no meio do processo.

Cada processo de negócio tem como resultado um produto ou serviço. No meio corporativo, pode-se utilizar as linguagens BPMN (*Business Process Modeling Notation*) ou a BPEL (*Business Process Execution Language*) que será o foco deste trabalho para escrever tais processos.

Também conhecida como BPEL4WS, a BPEL surgiu da combinação de duas linguagens anteriores a ela: a IBM Web Service FlowLanguage (WSFL) e Microsoft XLANG (*Extensible Language*) (WEERAWANA, 2005).

As interações entre *Web Services* podem ser descritas de duas formas: por processo de negócio executável e por processo de negócio abstrato. Processo de negócio executável possui o atual comportamento de um participante em uma interação de negócios. Processos de negócios abstratos são a especificação parcial dos processos que não se destinem a serem executados. Um processo abstrato pode escnos quaisr alguns detalhes operacionais. Resumindo, processos desempenham um papel descritivo, com mais de um caso de uso possível, incluindo o comportamento observável ou modelo de processo. BPEL é destinado para modelar o comportamento de ambos os processos de negócios: os executáveis e os abstratos. Ao fazer isso, ele estende o modelo de interação de *Web Services* e lhe permite suportar transações de negócios. BPEL define um modelo de integração interoperável que deve facilitar a expansão da integração de processos automatizados. (ANDREWS et al., 2003).

A partir de um conjunto de serviços que cooperam entre si para solucionar um determinado problema, um mecanismo para controlar a execução desses serviços na forma de um processo é necessário. Esse mecanismo pode ser

implementado utilizando um barramento de serviços e uma linguagem específica, como BPEL. O barramento de serviços permite a integração de serviços e a composição de novas aplicações (SANTANA et al., 2009).

BPEL é uma linguagem que implementa uma lógica de alto nível, e fornece o suporte para as interações assíncronas (computacionais ou humanas) (ROCHA; VERDE; PEREIRA, 2006).

1.1 OBJETIVO GERAL

Demonstrar, por meio de um estudo experimental, a implementação de processos de negócios utilizando a linguagem BPEL.

1.2 OBJETIVOS ESPECÍFICOS

Como objetivos específicos, destacam-se:

- Apresentar os conceitos e as utilizações atuais da linguagem BPEL;
- Apresentar ferramentas para implementação de processos de negócios com BPEL;
- Demonstrar a aplicabilidade da linguagem BPEL por meio de um estudo experimental;
- Analisar os resultados obtidos com o estudo experimental.

1.3 JUSTIFICATIVA

Padrões permitem menor custo de propriedade através de portabilidade do conhecimento – em vez de usar complexas tecnologias proprietárias, BPEL permite que as melhores práticas, padrões, experiência e formação sejam aproveitadas a partir de uma variedade de fornecedores, bem como acesso a recursos experientes no modelo BPEL e tecnologia (GEYER, 2007).

Para cenários nos quais as atividades são mais complexas, devido a intervenções humanas (*workflow*) e orquestração para controlar como e quando os fluxos de dados vão sendo trocados entre sistemas, o BPEL é a melhor opção (BARROS, 2008).

Em sua versão atual, o BPEL 2.0, é uma especificação aceitável para os clientes construírem processos de missão crítica (GHALIMI, 2006). Atualmente, um dos fornecedores - a Oracle - cita em seu *Website* alguns clientes com histórias de sucesso com o BPEL, tais como: Claro do Brasil, Samsung Electro-Mechanics da Coreia do Sul, Warid Telecom do Paquistão, entre outras.

1.4 ESTRUTURA DO TRABALHO

O trabalho foi dividido em quatro capítulos que abordarão os seguintes temas:

O capítulo dois enfatiza os processos de negócios, descrevendo um pouco de história da criação e como está evoluindo no mundo corporativo. Este capítulo também disponibiliza um pouco de conteúdo sobre as tecnologias utilizadas em conjunto com processos de negócios na linguagem BPEL.

O capítulo seguinte traz um estudo experimental envolvendo um exemplo referente a venda de produtos, nos quais são utilizadas várias tecnologias em conjunto para atender ao estudo proposto e demonstrar o funcionamento do processo de negócio envolvido no trabalho.

No capítulo quatro são apresentadas as conclusões deste trabalho.

2 PROCESSOS DE NEGÓCIOS COM BPEL

2.1 INTRODUÇÃO

O processo de negócio é uma sequência de interações entre diversas entidades internas e/ou externas a uma organização, que tem como finalidade modelar o negócio da organização. Um bom processo de negócio deve maximizar e flexibilizar a reestruturação e/ou a integração de novas estruturas de negócio. Isto torna os processos complexos e dinâmicos, implicando constantes alterações com o objetivo da otimização. Mais uma vez, as linguagens de definição processual podem minimizar os inconvenientes do caráter dinâmico dos processos organizacionais (ROCHA; VERDE; PEREIRA; 2006).

Eles são atividades que transformam insumos como conhecimento e matérias-primas em produtos e serviços que criam valor para os clientes. Uma organização depende em grande parte sobre o quão bem realiza seus processos de negócios. Os processos de negócios constituem todas as atividades que uma empresa faz, usando pessoas, tecnologia e informações, para cumprir sua missão, estabelecer metas, medir o desempenho, servir consumidores e enfrentar desafios inevitáveis que surgem ao fazer seu trabalho. Tais processos determinam a eficácia e eficiência das operações de sua empresa e também a satisfação do cliente (MCDONALD, 2010).

Tecnicamente, para McDonald (2010) um processo de negócio é um conjunto de passos que uma empresa realiza para agregar valor para os clientes. Um processo é composto por três componentes básicos:

- *Inputs*: eles começam o processo. Por exemplo, quando se está construindo uma bicicleta, os inputs são as rodas, pneus, parafusos entre outros.
- *Atividade*: estes transformam inputs em outputs. No exemplo da bicicleta, atividades seriam a inclusão do quadro, prender as rodas e ajustar as engrenagens.
- *Outputs*: também são conhecidos como consequência, são considerados os resultados das atividades – nesse exemplo, o produto final da construção da bicicleta.

Os processos são mais fáceis de entender quando se considera bens físicos, como no exemplo bicicletas. Entretanto existem processos em todas as empresas que não são apenas físicos. Por exemplo, em uma empresa que presta serviços de consultoria de gestão, existem os *inputs* (como o conhecimento de um consultor), atividades (como, fazer uma pesquisa com os funcionários do cliente para obter uma visão geral hierarquia da empresa) e *outputs* (como um plano de mudança na organização dos setores da empresa) (MCDONALD, 2010).

Os Processos de negócios também podem ser entendidos como uma série de eventos que reúnem pessoas, tecnologia e informação, de forma a criar saídas valiosas. Nessa estrutura, têm-se pessoas realizam atividades de processos baseado nas entradas que recebem (como o pedido de um cliente), estas pessoas estão muitas vezes trabalhando com a tecnologia (aplicações de *software* e da Internet) e a informação fornecida pelos clientes, que podem ser mantidas em banco de dados ou na mente das pessoas. Essas informações podem ser à entrada de um processo (como o número de peças em uma loja) ou uma saída (por exemplo, um relatório de estoque) (MCDONALD, 2010).

Aplicativos corporativos e sistemas de informação tornaram-se ativos fundamentais para as empresas. Elas dependem deles para ser capaz de realizar operações de negócios. Os sistemas de informações das empresas podem melhorar a eficiência das empresas através da automação de processos de negócios. O objetivo de quase todas as empresas é que o aplicativo que ele usa deve fornecer suporte abrangente para processo de negócios. Isto significa que as aplicações devem se alinhar com os processos de negócios (JURIC, KRIZEVNIK 2010).

Toda organização contém um grande número de processos de negócios. Eles podem ser simples realizados em um único departamento (como o registro de um cliente no sistema), ou podem ser complexos que envolvem vários departamentos (por exemplo, o desenvolvimento de produtos). Dessa maneira, quando um processo é melhorado, obtém-se vantagens para a empresa na forma de redução de custos, eficiência e rentabilidade. Embora este requisito não parecesse muito difícil de cumprir, a situação do mundo real mostra um quadro diferente. Os processos de negócios são geralmente dinâmicos, nos quais frequentemente as empresas têm de melhorar, otimizar e adaptar os processos de negócios aos seus clientes, e assim melhorar a agilidade de toda a empresa. Toda mudança em um processo de negócio deve ser refletida nos aplicativos que oferecem suporte para

eles, para manter-se uma harmonia no sistema como um todo. A partir da perspectiva de sistemas de negócios, é importante que a parte de T.I. forneça aplicativos que suportam os processos de negócios desde o início até o fim. Esse suporte deve ser flexível, de modo que os processos comerciais possam ser modificados rapidamente (JURIC, KRIZEVNIK 2010).

As pessoas relacionadas com a organização realizam muitos processos diferentes a cada dia, que em muitos casos são invisíveis. Neste sentido muitas pessoas não pensam coincidentemente ou percebem o impacto que tem sobre o desempenho de uma organização. Melhorar os processos de negócios ajuda a compreender como efetivamente uma equipe está atendendo as necessidades dos clientes e outros departamentos na empresa. Isso ajuda a economizar tempo e dinheiro da empresa, simplificando os processos excessivamente complexos e caros (MCDONALD, 2010).

2.2 MODELAGEM DE PROCESSO DE NEGÓCIO

Processos de negócios são modelados através de entrevistas, discussões com os *stakeholders*¹ e como resultado de sessões de *brainstorming*². Logo, para entender os mecanismos de um determinado negócio, deve-se realizar o levantamento para a modelagem, iniciando-se com encontros e entrevistas com os empregados e gerentes da organização (ERIKSSON; PENKER, 2000).

A criação de um documento inicial para especificação de informações importantes sobre o novo processo constitui uma fase extremamente importante para o futuro do novo processo.

Garimela, Lees e Williams (2008) apontam o proprietário do processo como o responsável por realizar a documentação do processo de negócio. Outro elemento da documentação de processos, nas palavras dos mesmos autores, são as regras de negócio.

A estes elementos podemos adicionar outros importantes para a documentação de um processo, a saber: tempos de execução, custos envolvidos,

¹ Termo utilizado para se referenciar à parte interessada.

² Reuniões para geração de novas ideias, conceitos e soluções para qualquer assunto.

metas de sucesso, e se estão implementados através de tecnologia de informação, além de seu fluxo de informação (DÁVALOS, 2006).

É importante considerar maneiras de cortar custos. Discutir ideias que considerem o corte de custos e também o tempo necessário para concluir o processo são algumas dicas que sempre se deve considerar ao melhorar um processo (MCDONALD, 2010).

Antes de implementar um novo processo, é útil entender os obstáculos comuns para a implementação de um processo e assim pode tomar medidas preventivas para evitá-los. A tabela 1 mostra alguns obstáculos comuns.

Tabela 1 – Obstáculos comuns na implementação de um processo de negócio.

Obstáculo	Explicação e exemplo	Como evitar
Resistência dos empregados	Qualquer processo de mudança leva a suposição implícita de que as pessoas estavam fazendo a coisa errada no processo antigo. Pode esperar alguma resistência ao novo processo de empregados, principalmente se não há nenhuma maneira fácil para eles para ver os benefícios de fazer a mudança.	Reconhecer as preocupações dos funcionários, apontando como o novo processo resolve os problemas levantados pelo processo antigo. Ajude os funcionários a ver como irão se beneficiar da adoção do novo processo, e explicar claramente as mudanças nas cargas de trabalho e remuneração que virão com o novo processo.
Resistência dos gerentes	Se um processo redesenhado atravessa vários departamentos ou unidades, as disputas territoriais podem surgir quando se trata de tempo para implementação. Especificamente, os gestores de algumas unidades podem não estar dispostos a mudar a maneira como eles trabalham, a fim de colocar o novo processo em ação.	Esta questão precisa ser tratada antes da implementação – durante o desenvolvimento e teste se possível. Os gerentes precisam ver que o seu apoio ao novo processo irá tornar seu trabalho mais fácil ou capacitá-los a apresentar melhor desempenho. Para ganhar o seu apoio, destacar os problemas de negócios que o processo atual está causando, sua atuação no processo antigo, a sua capacidade para melhorar esse desempenho usando o novo processo, e como o novo processo irá beneficiá-los.
Nenhum campeão para conduzir o projeto.	Toda mudança exige um processo de campeão – a pessoa que fornece suporte executivo ao projeto. Em casos de mudança do processo principal, o CEO ou um líder de alto nível pode ser o campeão. O campeão fornece o poder executivo para fazer alterações ao processo e para redistribuir responsabilidades.	Se o projeto nunca teve um campeão, não deveria ter começado no primeiro lugar. Se o campeão mudou ou perdeu o interesse, no entanto, deve-se lembrar-lhe do valor do negócio, que os resultados irão gerar. Certifique-se o campeão demonstre seu apoio para o projeto para outros gerentes e os funcionários envolvidos.

Fonte: McDonald, 2010.

McDonald (2010) destaca alguns passos importantes a serem seguidos na modelagem de um processo de negócio:

- Comunica-se – passar a mensagem sobre a implementação futura do novo processo. Dar a todos – gestores, trabalhadores e outras pessoas envolvidas no processo – tempo para entender o da necessidade do novo processo, o que é, e como as coisas vão ser diferentes e melhores, e o que vai acontecer quando o novo processo tornar-se parte das operações diárias.
- Educar e familiarizar – construir uma compreensão mais detalhada de como o novo processo funciona por meio de prática e simulação. Durante a prática é importante procurar os problemas e planejar como eliminá-los.
- Projeto piloto – se ainda suspeita-se que há alguma dificuldade no projeto piloto, ele ainda precisa ser trabalhado nesta etapa. Durante a execução de um projeto piloto, procura-se simular circunstâncias reais do negócio. Isso gera um valioso *feedback*³ que ajudará a melhorar o processo em questão e é uma ótima oportunidade de observar o novo processo em ação.
- Implementar – colocar o novo processo para a produção, declarando o início de novas operações. Começa a reunir dados sobre o desempenho de novas medidas relacionadas com o processo, não esquecendo que haverá uma curva de aprendizado e que é necessário considerar um tempo de aprendizagem.
- Excluir os laços com o passado – Remover os artefatos do processo antigo para reduzir qualquer tentação de continuar usando os mesmos hábitos. Esses artefatos podem ser equipamentos, sinalização e outras formas.

Mesmo pequenas melhorias em um processo relativamente simples podem pagar grandes dividendos de uma organização. Dominar os conceitos básicos de melhoria de processos de negócios ajuda a aguçar a vantagem competitiva da sua empresa (MCDONALD, 2010).

³Termo utilizado para se referenciar ao retorno de informações.

No caso de se requerer apenas modificações, o tempo envolvido na alteração está relacionado alguns fatores. O mais importante deles, além da complexidade e do tamanho da modificação, é o estado da aplicação a ser alterada. Se uma aplicação tem uma arquitetura bem definida e foi construída tendo em vista futuras modificações, então será mais fácil de modificar. No entanto, cada modificação para a aplicação faz a sua arquitetura menos robusta em relação às mudanças futuras. Normalmente cada nova alteração a torna mais difícil, demorada e muitas vezes resultam em erros inesperados (JURIC, 2006).

2.3 FLUXOGRAMAS

Através de fluxogramas tem-se uma representação gráfica das etapas do processo, no qual cada símbolo tem sua particularidade dentro do contexto. Os símbolos utilizados na confecção de um fluxograma estão apresentados na tabela 2.

Tabela 2 – Símbolos de um fluxograma.

Símbolo	Exemplo	Significado
Caixa		Uma atividade que uma pessoa ou um sistema executa
Diamante		Uma análise ou decisão que uma pessoa ou tecnologia deve fazer
Seta		Direção do fluxo de trabalho; A seta pode aparecer entre os símbolos de fluxograma
Triângulo		Depósito ou armazenamento de materiais ou informações
Grande D		Um atraso, como falhas de equipamento ou simplesmente à espera de informações.
Oval		O começo e o fim de mapa de processo.
Círculo		Uma referência cruzada para outro processo.

Fonte: McDonald, 2010.

2.4 CRIAÇÃO DE FLUXOGRAMAS FUNCIONAIS

McDonald (2010) definiu alguns passos simples para ajudar na criação de um fluxograma simples e funcional:

- Definir os limites do processo – através de uma equipe de melhoria de processos de negócios, o primeiro passo é identificar as atividades ou decisões que marcam o início e o final do processo;
- Registrar os cargos das pessoas envolvidas no processo – devem ser listados os cargos de todas as pessoas que trabalham no processo.
- Criar metas – separar os trabalhos com linhas, estas tornam-se metas e permitem acompanhar o trabalho das pessoas e ajuda a identificar os desequilíbrios do trabalho e dos participantes do processo.
- Adicionar detalhes dos processos – para cada cargo no mapa, inserir uma caixa que representa o que a pessoa faz no processo. Dentro da caixa, usar combinações verbo-substantivo para descrever o que a pessoa faz. Por exemplo, “receber formulário”. Inserir diamantes que representam as decisões que as pessoas tem que fazer durante a realização do processo. Dentro dos diamantes, usar perguntas para representar as decisões. Por exemplo, “o arquivo está completo?”.
- Mostrar a sequência de atividades – numerar cada caixa e diamante no fluxograma para indicar a sequência em que as atividades são realizadas durante o processo. Deve-se certificar que cada caixa ou diamante tem um número único.

2.5 AUTOMATIZAÇÃO DE PROCESSOS DE NEGÓCIOS

Durante muitos anos a indústria de software vem buscando arquiteturas eficientes, tecnologias e métodos que fariam a realização de algumas tarefas o mais simples e rápido possível. Para Juric (2006), através dessa visão, destacam-se alguns pontos para a automatização eficiente de processos de negócios através de TI:

- Fornecer uma maneira padronizada para expor e acessar a funcionalidade de aplicativos como serviços. A exigência de expor

funcionalidade de aplicativos e acessá-las remotamente resultou em arquiteturas distribuídas, que surgiram ao longo do tempo. A última arquitetura distribuída, que combina as comunicações síncronas e assíncronas, é Web Services. Ela é a arquitetura mais adequada distribuída para expor a funcionalidade de aplicativos como serviços.

- Integração entre serviços e aplicações - Esta integração é necessária porque os sistemas de informação corporativos consistem geralmente em várias aplicações diferentes, que abordam determinadas funções, tarefas e processos de negócios não inteiros. Atingir a integração eficiente está relacionado com a definição e realização de arquiteturas de integração, que são muitas vezes bastante complexas, especialmente em grandes empresas. Os melhores métodos e práticas para arquiteturas de integração de construção são hoje conhecidos como Arquiteturas Orientadas a Serviços (SOA).
- Composição dos serviços expostos em Processos de Negócios – o aspecto final é a composição de serviços expostos de aplicações integradas nos processos de negócio.

A linguagem mais popular, comumente aceita, e especializada para a definição do processo de negócio é BPEL. Esta promete desenvolver os processos de uma maneira fácil e eficiente, rapidamente adaptando-se às necessidades das empresas em muito esforço.

2.6 SOA

Sistemas de informação precisam suportar as mudanças de uma forma rápida e eficiente. No entanto, eles também precisam se adaptar ao rápido desenvolvimento de novas tecnologias. A maioria dos sistemas de informação empresariais são heterogêneos, contendo uma variedade de diferentes sistemas, aplicações, tecnologias e arquiteturas. A integração dessas tecnologias é crucial uma vez que apenas sistemas integrados de informação pode entregar valores de negócios, como uma eficiente tomada de decisão com suporte ao acesso instantâneo à informação e a integridade dos dados, juntamente com custo reduzido

de desenvolvimento de software e manutenção. Para gerenciar os problemas relacionados com as mudanças de requisitos, desenvolvimento de tecnologia e integração, diferentes métodos têm sido propostos e utilizados ao longo do tempo. Arquitetura orientada a serviços é uma abordagem arquitetônica relacionada com a integração, desenvolvimento e manutenção de sistemas de informação complexos empresariais (JURIC, 2006).

A camada *Web service* expõe as funcionalidades da camada de aplicação como um serviço. Desta forma, múltiplos processos de negócios podem ser reusados com serviços, o que representa a promessa SOA (BLANVALET, 2006). Integração entre as aplicações evoluiu desde os primeiros anos com métodos bem definidos e princípios, muitas vezes referidos como EAI (*Enterprise Application Integration*). EAI inicialmente focada em integração de aplicações dentro das empresas (JURIC, 2006).

Erl (2009) define SOA como uma arquitetura que visa aprimorar a eficiência, a agilidade e a produtividade de uma empresa, posicionando os serviços como os principais meios para que uma solução lógica seja representada no suporte à realização dos objetivos estratégicos associados à computação orientada a serviços.

SOA melhora e amplia a flexibilidade de métodos anteriores de integração (EAI) e arquiteturas distribuídas, e foca na reutilização de aplicações e sistemas existentes, interoperabilidade eficiente e integração de aplicações, e composição dos processos de negócio de serviços (funcionalidades) fornecido pelos aplicativos. (JURIC, KRIZEVNIK 2006).

SOA define os conceitos, arquitetura e estrutura do processo, para permitir que o custo-eficiente para o desenvolvimento, integração e manutenção de sistemas de informação, através da redução de complexidade, e estímulo de integração e reutilização (JURIC, 2006).

SOA é o estilo de arquitetura que suporta serviços de baixo acoplamento para permitir flexibilidade de negócios de maneira interoperável. SOA consiste em um conjunto composto de negócios alinhados com serviços que suportam uma abordagem flexível e dinâmica reconfigurável para realização de processos utilizando descrições de interface baseados em serviços (JURIC, 2006).

Segundo Juric e Krizevnik (2010) SOA é mais do que apenas um conjunto de tecnologias. SOA não é diretamente relacionado com qualquer tecnologia, embora seja mais frequentemente aplicada com *Web Services*. No

entanto, usando serviços Web não é suficiente para construir SOA. Devem-se usar serviços da Web de acordo com os conceitos que SOA define.

Os mais importantes conceitos de SOA são:

- Serviços;
- Interfaces;
- Troca de mensagens;
- Suporte para comunicação síncrona e assíncrona;
- Baixo acoplamento;
- Registros;
- Qualidade de serviço;
- Composição de serviços em processos de negócios;

2.6.1 Web Services

Web Services tornou-se uma tecnologia usada para a interoperabilidade e integração de aplicações e sistemas de informação. *Web Services* fornecem a base tecnológica para assegurar que as aplicações de diferentes plataformas de software, sistemas operacionais e linguagens de programação interajam. São construídos em linguagens de programação e usam XML (*Extensible Markup Language*) entre outros tipos para representar os dados e facilitar a integração (JURIC, 2006).

Além das várias vantagens, os serviços de Web também tem um par de desvantagens. Um deles é o desempenho, que não é tão bom como a de arquiteturas distribuídas que utilizam protocolos binários para a comunicação (SiRF). A outra é que os serviços de Web simples não oferecem infraestrutura e qualidade de serviço - como segurança, transações e outros, que tenham sido fornecidos por modelos de componentes por vários anos. *Webs services* preenchem esta lacuna importante, introduzindo especificações adicionais relatadas por Juric (2006):

- *WS-Security*: Endereços de autenticação e mensagem de nível de segurança, e permite a comunicação segura com serviços Web.
- *WS-Coordination*: Define um quadro de coordenação para os serviços Web e é a base para *WS-AtomicTransaction* e *WS-BusinessActivity*.

- Especificações de transações (*WS-AtomicTransaction* e *WS-BusinessActivity*): Especifica o suporte para transações distribuídas com serviços Web. *AtomicTransaction* especifica curta duração, e *BusinessActivity* especifica execução de transações comerciais, também chamado de transações de compensação.
- *WS-ReliableMessaging*: Fornece suporte para comunicação confiável e entrega de mensagens entre serviços Web através de protocolos de transporte.
- *WS-Addressing*: Especifica coordenação mensagem e roteamento.
- *WS-Inspection*: Fornece suporte para a introspecção dinâmica de descrições de serviços Web.
- *WS-Policy*: especifica como as políticas são declaradas e trocadas entre colaborando serviços Web.
- *WS-Eventing*: Define um modelo de evento para notificação assíncrona de partes interessadas para serviços Web.

2.7 O PADRÃO BPEL

Business Process Execution Language (BPEL) é um padrão para a definição de processos de negócios com a composição de serviços. Tem a capacidade de definir os processos de negócio executáveis e abstratos abrindo novas portas na gestão de processos de negócios (JURIC, 2006).

IBM, BEA e Microsoft desenvolveram a primeira versão do BPEL em agosto de 2002. Desde então, SAP e Siebel juntaram-se, o que resultou em diversas modificações e melhorias da versão 1.1 em março de 2003. Em Abril de 2003, BPEL foi submetido a OASIS (Organização para o Avanço de Padrões de Informação Estruturada) para fins de padronização. Em Abril de 2007 foi disponibilizado o Padrão BEPL 2.0. Desde então, isto levou a aceitação ainda mais ampla da indústria. BPEL representa uma convergência de duas linguagens, a WSFL (Linguagem de Fluxo de Serviços) e XLANG. WSFL foi inventada pela IBM e baseia-se no conceito de grafos. XLANG foi projetado pela Microsoft e é uma linguagem estruturada em blocos. BPEL combina as duas abordagens e oferece um rico vocabulário para a descrição dos processos de negócio (JURIC, 2006).

BPEL não constitui apenas mais um produto ou padrão lançado pela indústria de software, mais sim uma maneira de garantir que as organizações tenham a habilidade de modificar suas aplicações conforme possíveis mudanças em seu horizonte de negócio (BLANVALET, 2006).

BPEL é uma linguagem utilizada para a composição, orquestração e coordenação dos serviços da Web. Ela fornece um vocabulário rico para expressar o comportamento de processos de negócios (JURIC, 2006).

2.8 CARACTERÍSTICAS DO BPEL

Com BPEL, podem-se definir os processos de negócios simples e complexos. Até certo ponto BPEL é semelhante às linguagens de programação tradicionais. Oferece construções, tais como loops, variáveis, atribuições, etc., que permitem definir processos de negócio de uma forma algorítmica. BPEL é uma linguagem especializada focada na definição de processos de negócios. Portanto por um lado oferece construções, que fazem a definição de processos relativamente simples. Por outro lado, é menos complexo do que linguagens de programação tradicionais, o que simplifica a aprendizagem (JURIC, 2006).

BPEL implementa a lógica de alto nível, e fornece o suporte para as interações assíncronas (computacionais ou humanas) (ROCHA, VERDE, PEREIRA, 2006). Segundo Juric (2006), as construções BPEL mais importantes estão relacionadas com a invocação de serviços *Web*. Pode-se invocar operações quer em sequência ou em paralelo. Também pode-se esperar por retornos de chamada. BPEL fornece um rico vocabulário para tratamento de falhas, que é muito importante, pois o processo de negócio robusto precisa reagir a falhas de uma maneira inteligente. BPEL também fornece suporte para longa duração do processo e compensação. A seguir estão listadas as características mais importantes que BPEL fornece. Ainda segundo Juric, 2006. Com BPEL pode-se:

- Descrever a lógica dos processos de negócio através da composição de serviços;
- Compor os processos de negócios maiores de processos menores e serviços;

- Diferenciar operações síncronas e assíncronas (muitas vezes chamadas de longa duração) nos serviços, e gerencia de *callbacks* que ocorram em épocas posteriores;
- Chame operações de serviço em sequência ou em paralelo compensando seletivamente atividades concluídas em caso de falhas;
- Manter múltiplas atividades transacionais de longa duração, que também são passíveis de interrupção;
- Retomar as atividades interrompidas ou não minimizando o trabalho a ser feito;
- Enviar mensagens de rota de entrada para os processos e atividades apropriadas;
- Correlacionar os pedidos dentro e através de processos de negócios, programando atividades baseadas no tempo de execução e definição da ordem de execução;
- Executar atividades em paralelo e definir como os fluxos paralelos para trabalhar com base nas condições de sincronização;
- Estrutura de processos de negócios em vários escopos exibindo mensagens relacionadas com o tempo e evento;

2.9 BPEL PARA COMPOSICAO DE SERVIÇOS

A adoção geral de soluções de negócios de automação de processos exige um padrão e uma linguagem especializada para a composição de serviços em processos de negócios que fornece a capacidade de expressá-los de uma forma padronizada, utilizando uma linguagem comumente aceita. BPEL é uma linguagem destacável, podendo ser o padrão dominante. O objetivo principal do BPEL é padronizar o processo de automação entre serviços Web (JURIC, 2006).

Esta linguagem de execução utiliza a tecnologia XML para acessar dado, enquanto que os Web Services são utilizados para serialização⁴ de dado (BLANVALET, 2006).

⁴ Serialização – é uma técnica de persistência de dados através de uma sequencialização destes para posterior armazenamento em um arquivo XML, um arquivo texto e até um banco de dados.

Nas empresas, BPEL é usada para padronizar a integração de aplicações empresariais e ampliar a integração de sistemas isolados anteriormente. Entre as empresas, BPEL permite a integração mais fácil e eficaz com parceiros de negócios. BPEL estimula as empresas a definir melhor seus processos de negócios, que por sua vez leva à otimização de processos de negócios existentes, a reengenharia, e a seleção dos processos mais adequados, contribuindo assim para a otimização da organização. As definições de processos de negócios descritos em BPEL não influenciam sistemas existentes. BPEL é a tecnologia-chave em ambientes nos quais funcionalidades já estão ou estarão expostas através de serviços Web (JURIC, 2006).

2.10 ORQUESTRAÇÃO E COREOGRAFIA

Na orquestração, um processo central (que pode ser outro serviço *Web*) assume o controle sobre os serviços *Web* envolvidos e coordena a execução de diferentes operações sobre os serviços da *Web* envolvidas na operação. Isto é feito de acordo com as exigências da orquestração. Os serviços *Web* envolvidos não sabem (e não precisa saber) que eles estão envolvidos em uma composição e que é parte de um processo maior de negócios. Através da figura 1, observa-se que apenas o coordenador central da orquestração sabe disso, então a orquestração é centralizada, com definições das operações de cada serviço *Web* e a ordem de invocação deles (JURIC, 2006).

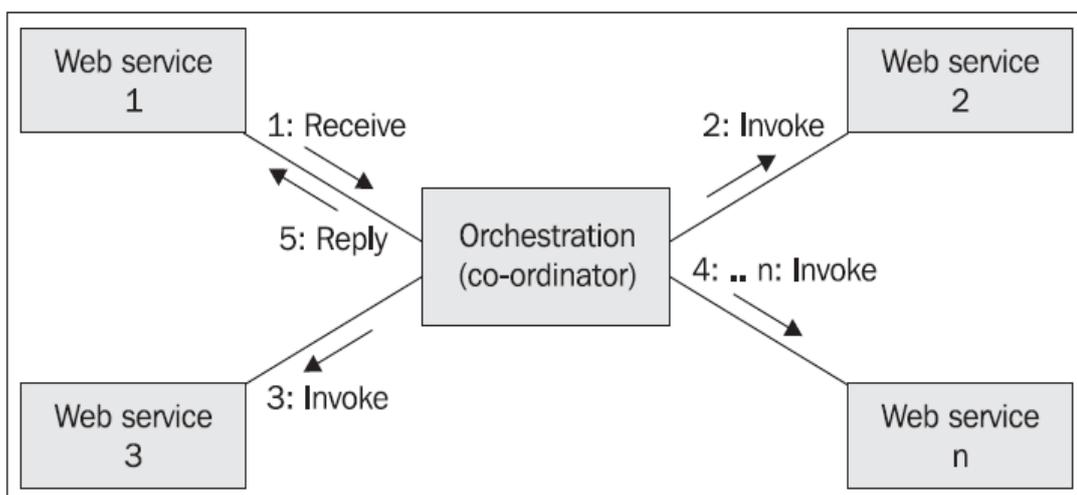
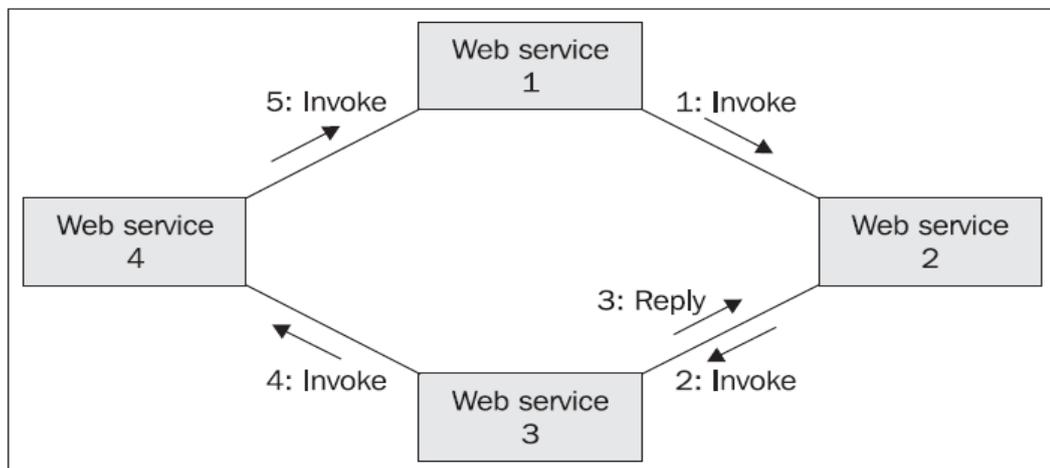


Figura 1 – Funcionamento da orquestração;

Fonte: Juric (2006)

Coreografia por outro lado não depende de um coordenador central. Em vez disso, cada serviço Web envolvido na coreografia sabe exatamente quando para de executar suas operações e com quem deve interagir. Coreografia é um esforço colaborativo focado na troca de mensagens em processo de negócios públicos. Todos os participantes da coreografia precisam estar cientes dos processos de negócios, operações a executar e a troca de mensagens. Coreografia na composição de serviços Web é como mostrada na figura 2 (JURIC, 2006):



**Figura 2 – Funcionamento da Coreografia;
Fonte: Juric (2006)**

Orquestração entre processos representa a visão da organização de seus processos, seu negócio específico. Coreografia entre processos é o comportamento esperado entre a interação de participantes do negócio, nos quais estes participantes podem ser regras de negócio ou uma entidade de negócio específica, como por exemplo, os correios ou uma transportadora. Já a colaboração entre processos é uma espécie de coreografia que define um conjunto de interações ordenadas entre os participantes de um processo (WHITE; MIERS, 2008).

Para Juric e Krizevnik (2010), do ponto de vista da composição de serviços Web para executar processos de negócios, a orquestração tem uma vantagem sobre a coreografia. Orquestração é um paradigma mais flexível, embora a linha entre a orquestração e coreografia está a desaparecer. Orquestração tem as seguintes vantagens:

- Sabe-se exatamente quem é responsável pela execução de todo o processo comercial.
- É possível incorporar serviços *Web*, mesmo aqueles que não estão cientes de que eles são parte de um processo de negócio.
- Podem-se fornecer cenários alternativos quando ocorrem falhas.

BPEL fornece suporte para orquestração e coreografia através de processos de negócio executáveis e abstratos.

2.11 PROCESSOS EXECUTÁVEIS E PROCESSOS ABSTRATOS

Segundo JURIC 2006, com BPEL, pode-se descrever processos de negócios de duas formas distintas:

- Especificando detalhes exatos de processos de negócios. Tais processos são chamados de processos de negócio executáveis e siga o paradigma de orquestração. Eles podem ser executados por um motor de orquestração.
- Especificando a troca de mensagens pública entre as partes apenas. Tais processos são chamados de processos de negócios abstratos. Eles não incluem os detalhes internos de fluxos de processos e não executável. Eles seguem o paradigma de coreografia.

Processos de negócios executáveis são processos que compõem um conjunto de serviços existentes. Processos executáveis são importantes porque são a resposta direta para o problema da automação de processos de negócios através de TI. Com os processos executáveis BPEL, é possível especificar o algoritmo exato da composição de serviços de uma maneira relativamente simples e direta, e executá-lo em um BPEL compatível. Processos executáveis preenchem a lacuna entre as especificações de processo de negócios e o código responsável pela sua execução. Ao se definir um processo de negócio executável em BPEL, define-se um novo serviço *Web* que é uma composição de serviços existentes. A interface do novo serviço de *Web* do BPEL composto usa um conjunto de tipos de portas, através do qual se fornece operações como qualquer outro serviço *Web*. Para chamar um processo de negócio executável, é chamado o serviço *Web* composto

resultante. Podemos ver que os processos de negócios executáveis são a forma mais importante de usar BPEL. Na maioria dos casos, BPEL é usado para especificar processos executáveis (JURIC, 2006).

Ainda segundo Juric (2006), processos comerciais Abstratos, por outro lado, não são realizáveis. Eles especificam a troca de mensagens pública entre as partes apenas de os aspectos externamente observáveis do comportamento do processo. A descrição do comportamento observável externamente de um processo comercial pode estar relacionada com um único serviço ou um conjunto de serviços. Também pode descrever o comportamento de um participante em um processo de negócio. Processos abstratos geralmente serão definidos principalmente para dois cenários:

- Para descrever o comportamento de um serviço sem saber exatamente em qual processo de negócio que irá participar.
- Para definir protocolos de colaboração entre os vários partidos e descrever precisamente o comportamento externo de cada partido.

Processos abstratos são raramente utilizados. O cenário mais comum é usá-los como um modelo para definir os processos executáveis. Processos abstratos podem ser usados para substituir conjuntos de regras geralmente expressos em linguagem natural, que é muitas vezes ambígua (JURIC 2006).

2.12 ORACLE SOA SUITE 11G

Possui um conjunto completo de componentes de infraestrutura para modelagem, implantação e gerenciamento de aplicações compostas. Permite que serviços sejam criados, gerenciados e orquestrados em aplicações compostas e processos de negócio (GALDINO, 2009). Na Figura 3 tem-se uma visão geral dos componentes da suíte:

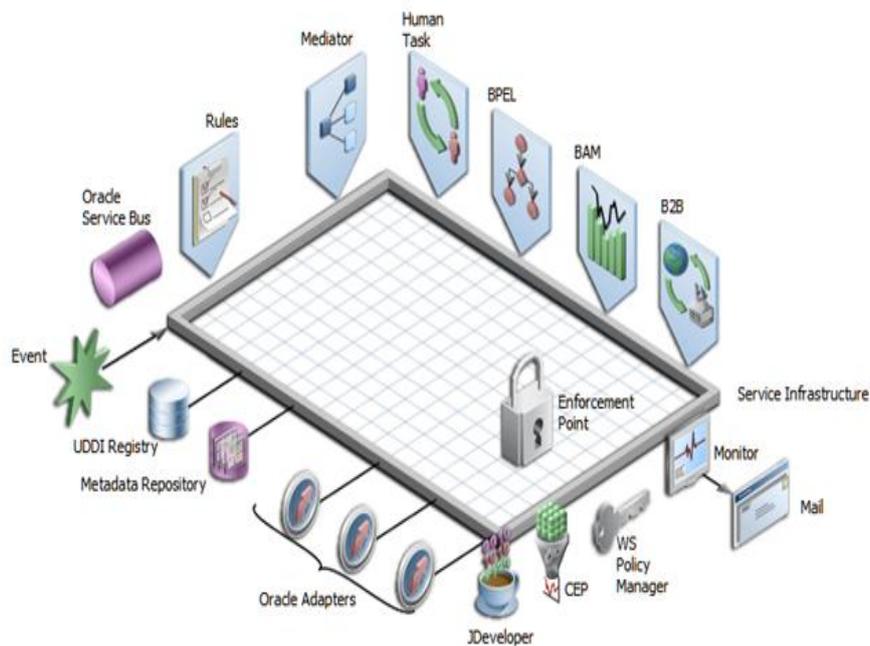


Figura 03 – Visão geral dos componentes da SUITE

Fonte: Galdino (2009)

2.12.1 Oracle Fusion Middleware

Middleware que é uma categoria de soluções tecnológicas para atuação em Arquitetura Distribuída cooperação entre múltiplos processos e máquinas, interagindo através de uma infraestrutura de rede para suportar e simplificar soluções distribuídas, complexas e interoperáveis agregando valor ao negócio, então podemos dizer que o Oracle *Fusion Middleware* é uma plataforma unificada que atende com um conjunto de ou soluções que atendem a diversos níveis, desde a infraestrutura, passando pelo banco de dados tendo a tecnologia de Middleware como base para a construção de aplicações. Como benefício, o Oracle Fusion Middleware, traz uma unificação com os servidores de aplicação, além de grande agilidade para o negócio dos seus clientes.

2.12.2 Jdeveloper para desenvolvimento com Bpel

A Oracle apresenta o JDeveloper com diversas funções, entre elas ele é um ambiente de desenvolvimento para o Oracle SOA Suite que permite aos desenvolvedores a construção de diagramas para processos em BPEL, e também é

um editor declarativo que fornece uma maneira fácil para definir regras de negócio para o Oracle SOA Suite.

2.12.3 Oracle Service Bus (OSB)

O Oracle Service Bus (OSB) é projetado para conectar, mediar e gerir as interações entre os serviços heterogêneos, aplicações legadas, aplicativos empacotados e várias instâncias Enterprise Service Bus (ESB) através de uma rede de serviços em toda a empresa. O OSB fornece capacidades integradas de gestão e de acompanhamento e suporte *out-of-the-bod* integração com produtos da família SOA. O Oracle Service Bus é um componente central no Oracle SOA Suíte como uma espinha dorsal para mensagens SOA.

2.12.4 Oracle Bpel Process Manager

Oracle BPEL Process Manager é uma parte integrante da Solução Oracle de Integração de Negócios, que fornece uma solução abrangente, baseado em padrões e fácil de usar para criar, implementar e gerenciar processos entre aplicativos de negócios com ambos fluxos de trabalho, automatizado e humano – tudo em uma arquitetura orientada a serviços. Seu suporte nativo para padrões como BPEL, XML, XSLT (Extensible Stylesheet Language Transformation), XPATH (XML Path language), JMS (Java Messaging System), JCA (Java Connector Architecture) e Serviços Web torna esta uma solução ideal para a criação de processos de negócios integrados que são verdadeiramente portátil através de plataformas.

3 ESTUDO EXPERIMENTAL

O estudo experimental utilizado neste trabalho é um exemplo apresentado pela Oracle para ser desenvolvido em BPEL e que neste trabalho está adaptado ao Oracle Soa Suite. A descrição deste estudo é dada no Quadro 1.

Quadro 1 – Descrição do estudo experimental utilizado.

Uma empresa fabrica equipamentos e os vende através de varejo na Web e também através de representantes. Esta empresa não pode fornecer os mesmos serviços e respostas a todos os clientes. Com o tempo, ela viu o aumento das vendas porém as margens de lucro vinham caindo nos últimos tempos, e então decidiu implementar mecanismos para melhorar a logística dos produtos e melhorar o serviço ao consumidor e assim administrar melhor este processo.

Com isso foram utilizadas as ferramentas disponíveis para implementar um processo de negócio que recebe encomendas feitas à empresa através de qualquer canal. Este processo define qual o melhor fornecedor para atender a cada pedido com base nas regras de negócio do processo da empresa.

O estudo de caso apresenta um processo sofisticado que utiliza serviços externos e fluxos paralelos de execução. Os serviços e fluxos representam a identificação do cliente e o pedido passado aos fornecedores respectivamente.

3.1 DESENVOLVIMENTO DA APLICAÇÃO

Para o desenvolvimento do estudo experimental proposto, inicialmente foi instalado todo o pacote Oracle Suíte 11g e Oracle Process Manager.

Segundo o exemplo as encomendas para a empresa são feitas por diversos meios, como internet, telefone ou pessoalmente. O sistema interpreta esses pedidos e os encaminha para o processo responsável por atender o pedido com base nas regras de negocio da empresa.

Na etapa seguinte, o cliente é identificado e o sistema faz a verificação de crédito e prossegue com o processo.

Os Itens do pedido são passados para os dois fornecedores para serem comparados os preços. Ambos os fornecedores têm um prazo para respnos quaisr com a cotação de preços. Devido à confidencialidade do negócio, os detalhes do cliente do pedido não são passados junto com a lista de produtos.

O processo de cotação de preços retorna os preços e então é passado ao gerente de negócios para decidir de qual fornecedor tal venda será atendida com base nas duas cotações.

3.1.1 Criação de um projeto SOA.

A criação de um projeto representando alguns processos de negócios em BPEL no JDeveloper 11g consiste em primeiramente criar uma nova aplicação, que consiste na mesma ideia de *workspace* de outros programas utilizados para desenvolvimento, como no caso do Eclipse IDE. A figura 4 exibe a estrutura da janela principal para a criação de uma nova aplicação no JDeveloper 11g.

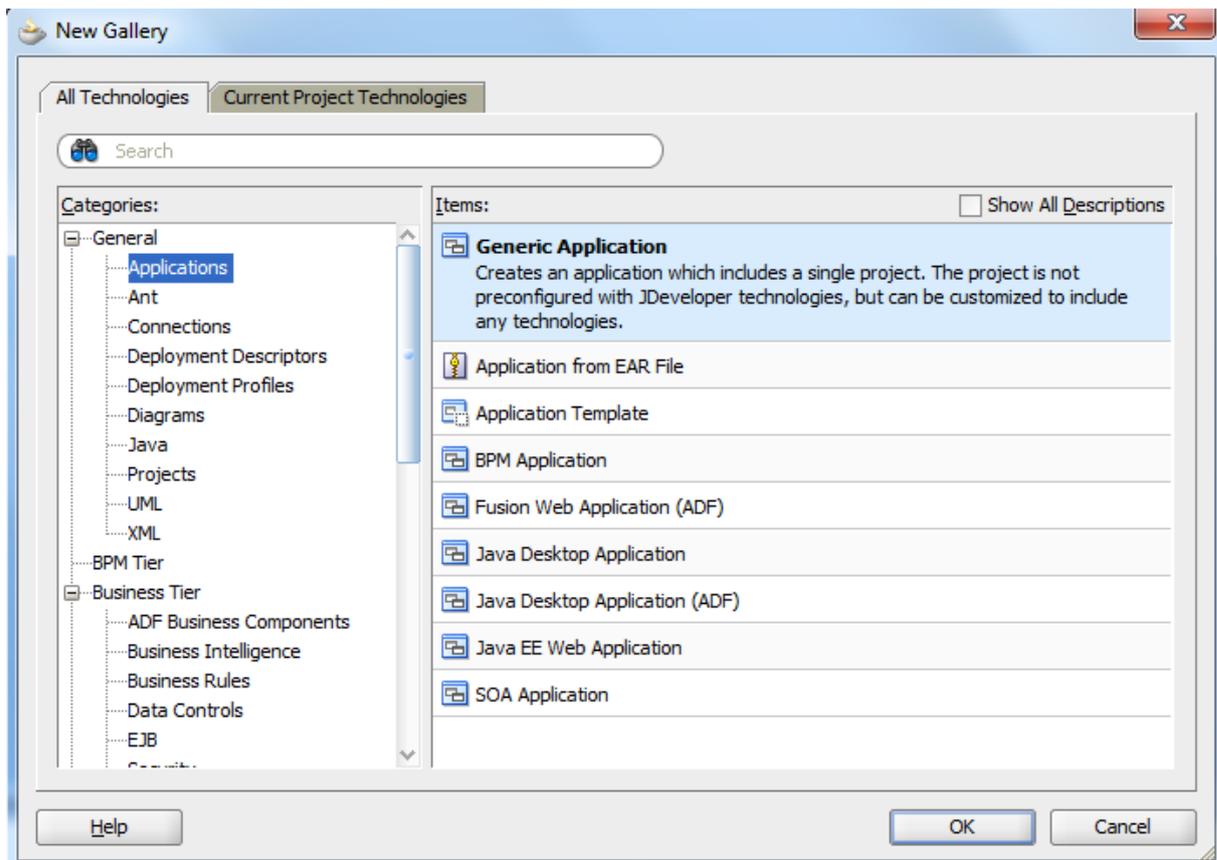


Figura 4 – Janela inicial para criação de uma nova aplicação no JDeveloper 11g.

Fonte: Autoria Própria.

A ideia de se escolher uma aplicação genérica é utilizada para poder suportar vários tipos de projetos nessa mesma aplicação e assim poder escolher ter

suporte para várias tecnologias. Uma dessas tecnologias escolhidas para este trabalho foi SOA, nos quais o projeto pode acessar serviços remotamente. A figura 5 mostra algumas tecnologias que podem ser selecionadas para se trabalhar no mesmo projeto.

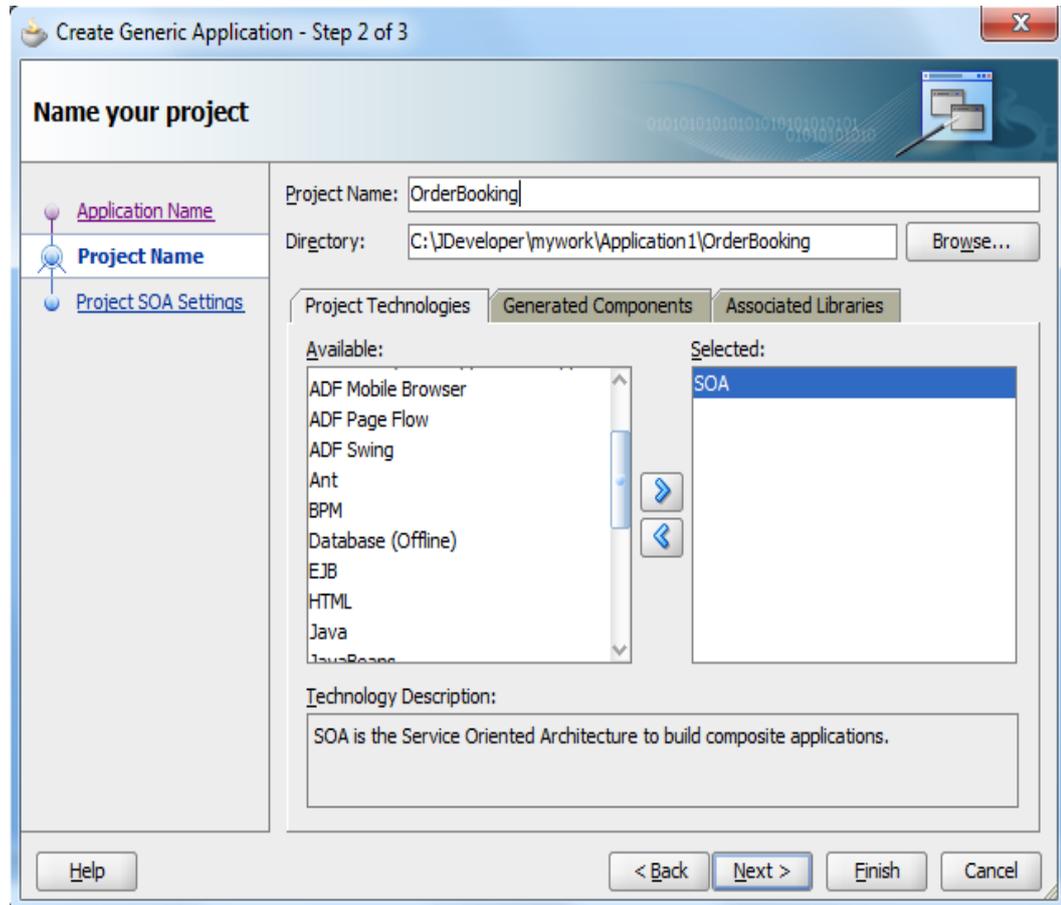
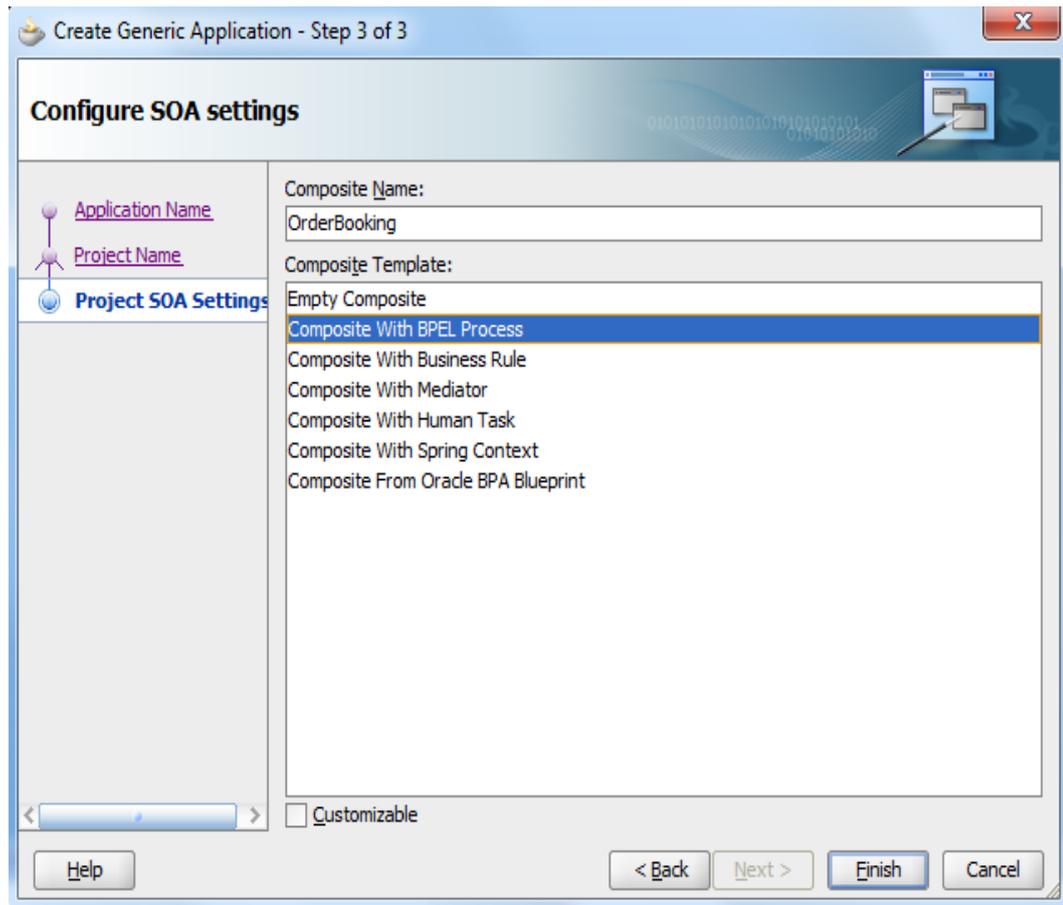


Figura 5 – escolha das tecnologias envolvidas nos projetos da aplicação.

Fonte: Autoria Própria.

A linguagem BPEL trabalha através de SOA para acessar os serviços disponibilizados por *Web Services*, e quando selecionada esta tecnologia para o ambiente de trabalho, são oferecido alguns *templates* para criação de um projeto com ambas tecnologias. A figura 6 mostra os *templates* disponíveis para o projeto.



**Figura 6 – sugestão de templates do JDeveloper para se trabalhar com projeto SOA.
Fonte: Autoria Própria.**

3.1.2 Criação de um processo Bpel

A criação de um processo BPEL envolve algumas configurações iniciais importantes, como o tipo do processo, se síncrono ou assíncrono. É necessário especificar a versão da linguagem que será utilizada para evitar alguns problemas de recursos no decorrer do projeto. Todos os aspectos das configurações podem ser vistos na figura 7:

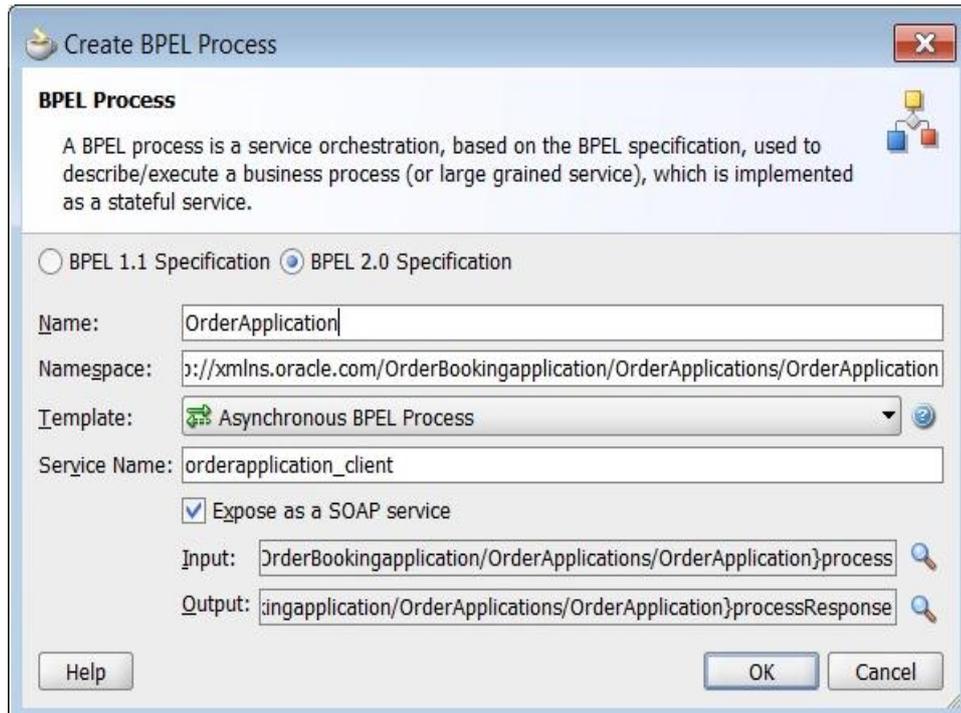


Figura 7 – criação de um processo BPEL no JDeveloper 11g.
Fonte: Aatoria Própria.

Através dos campos Input e Output declaram-se as variáveis responsáveis por informações do processo, no qual é possível também importar alguma estrutura de outro projeto no qual já tenha algumas informações necessárias a esse processo. A figura 8 mostra algumas variáveis disponíveis de outro projeto.

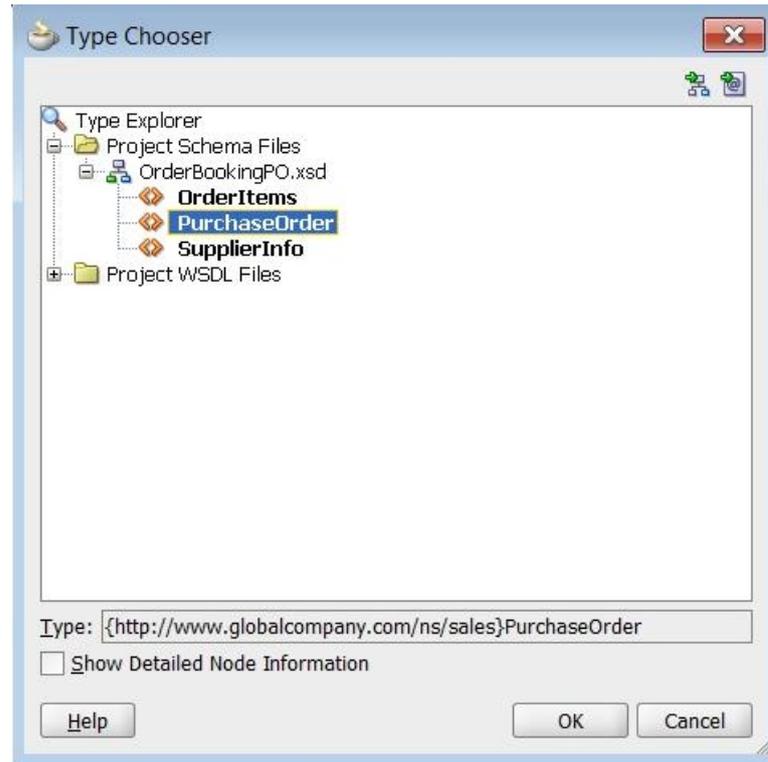


Figura 8 – importação do schema exemplo para o processo BPEL.

Fonte: Autoria Própria.

Com isso a estrutura inicial do processo é criada e está pronta para começar a implementar os detalhes do processo.

3.1.3 Conexão de links externos para consumir os serviços

A criação de um processo BPEL exige componentes para dar funcionalidades ao processo e assim ampliar suas funcionalidades. O JDeveloper possui uma área nos quais essas ferramentas ficam disponíveis ao desenvolvedor. Para utilizar qualquer um deles, pode-se arrastar e soltar na janela de designer. A figura 9 mostra os componentes disponíveis:

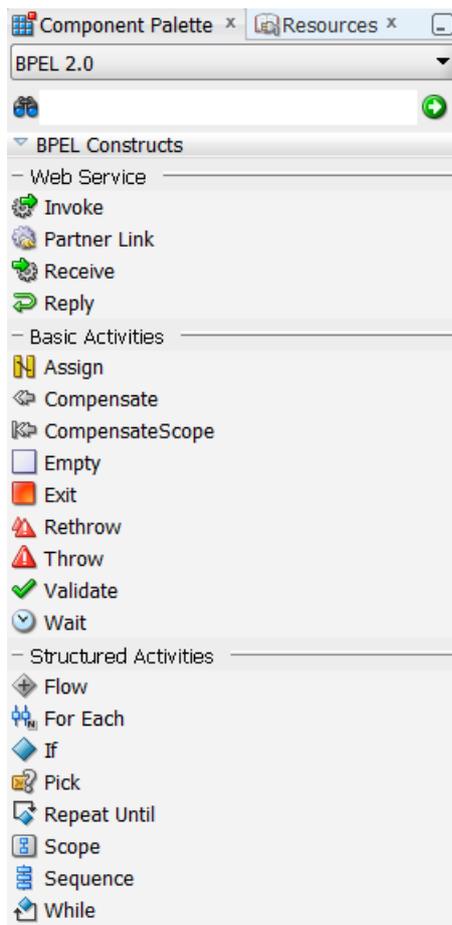


Figura 9 – paleta de componentes utilizáveis em um processo BPEL.

Fonte: Autoria Própria.

A conexão do processo BPEL com serviços disponibilizados por um *Web service* é feita através do componente *Partner Link* no qual é indicado o caminho do serviço que deve ser acessado.

Este primeiro processo deve-se configurar para que possa ser feita a consulta de crédito do cliente. A principal etapa dessa configuração é o WSDL, no qual este é a descrição de nos quais o serviço está disponível para ser processado.

A obtenção do WSDL deve ser feita no servidor nos quais o *Web Services* está disponível e respondendo às requisições. Com o servidor BPEL iniciado, tem-se acesso ao WSDL através do seu console que está exibido na figura 10:

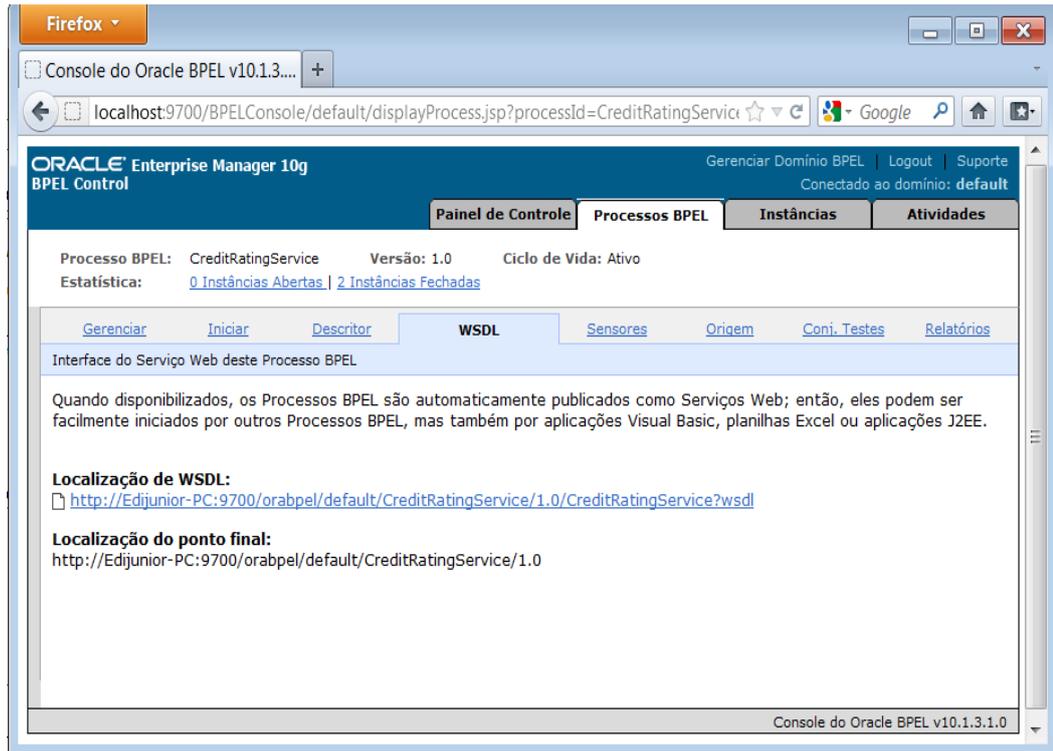


Figura 10 – localização do WSDL para serviço de credito do cliente.

Fonte: Autoria Própria.

Após a obtenção do WSDL, esta pronta para ser feita a configuração no *Partner Link* do processo BPEL. A figura 11 mostra como é configuração do *Partner link*.

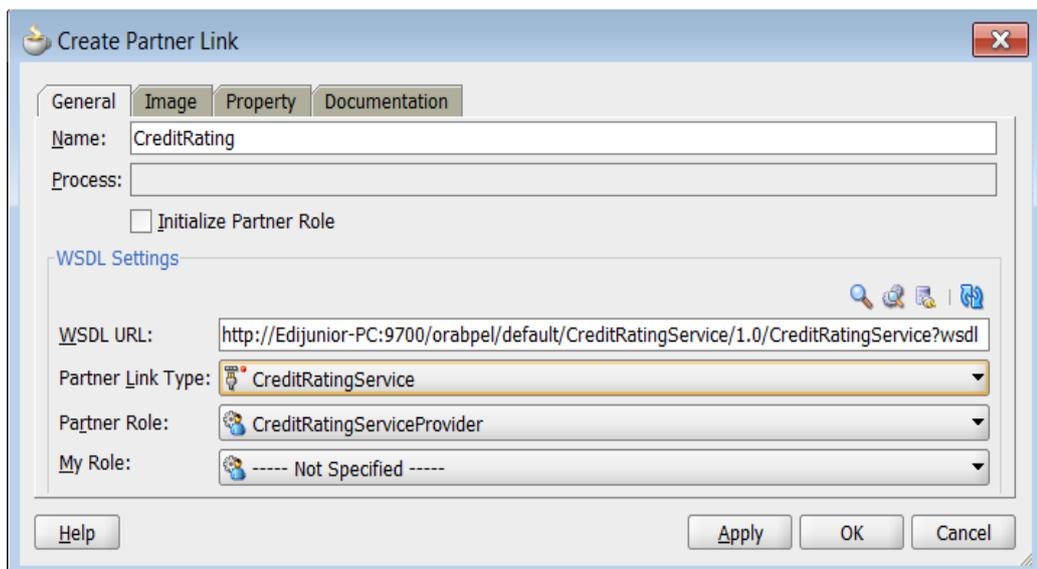


Figura 11 – configuração do link para consulta de crédito.

Fonte: Autoria Própria.

A criação dessa atividade permite que os dados da solicitação do processo de compra sejam enviados para a consulta de crédito do cliente e assim receber uma resposta e dar prosseguimento ao processo.

3.1.4 Organização do processo de negócio com escopos

Uma maneira eficiente de organizar os componentes dentro do processo é a utilização de escopos para separar as diferentes interações decorrentes da ação a ser executada pelo processo.

Disponível na aba de componentes, o *Scope* está disponível para utilização assim que arrastado para a área de design. A figura 12 mostra o componente no projeto:

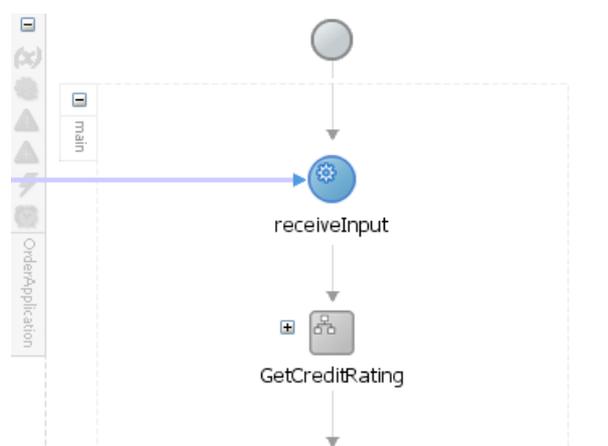


Figura 12 – aspecto do componente Scope no projeto.

Fonte: Autoria Própria.

3.1.5 Invocação de serviços

Dentro do escopo é possível utilizar componentes para acessar um serviço disponível através do *Partner Link*, e para que isso seja possível é necessária também uma preparação de dados para saber qual será responsável por guardar o dado manipulado por aquele processo. O componente *Assing* é responsável por administrar as variáveis que serão utilizadas para o envio de uma solicitação para o serviço de crédito. Através dele declara-se a variável que será responsável por uma ação. Um exemplo desta ação está na figura 13:

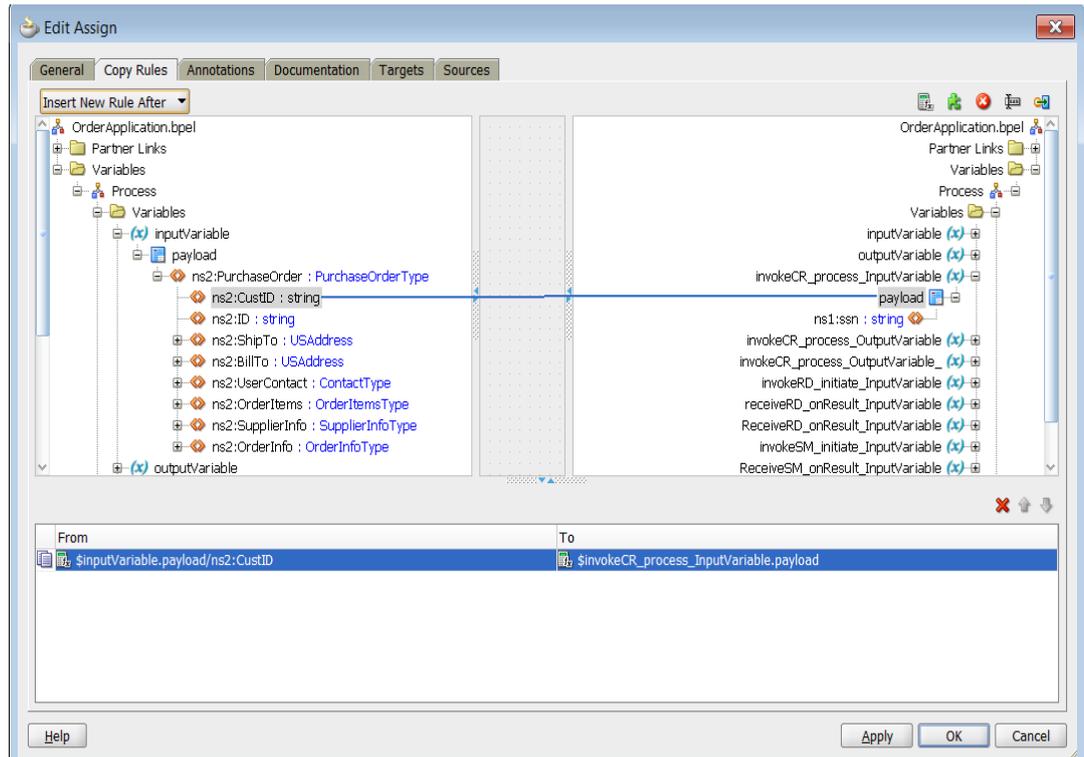


Figura 13 – atribuição de regras e definições de funções envolvendo variáveis.
Fonte: Autoria Própria.

É possível também atribuir uma expressão a uma variável, entre tantas disponíveis, foi usada a “concat” como exemplo para concatenar alguma String ao retorno de um serviço. A figura 14 exhibe como ficou a expressão.

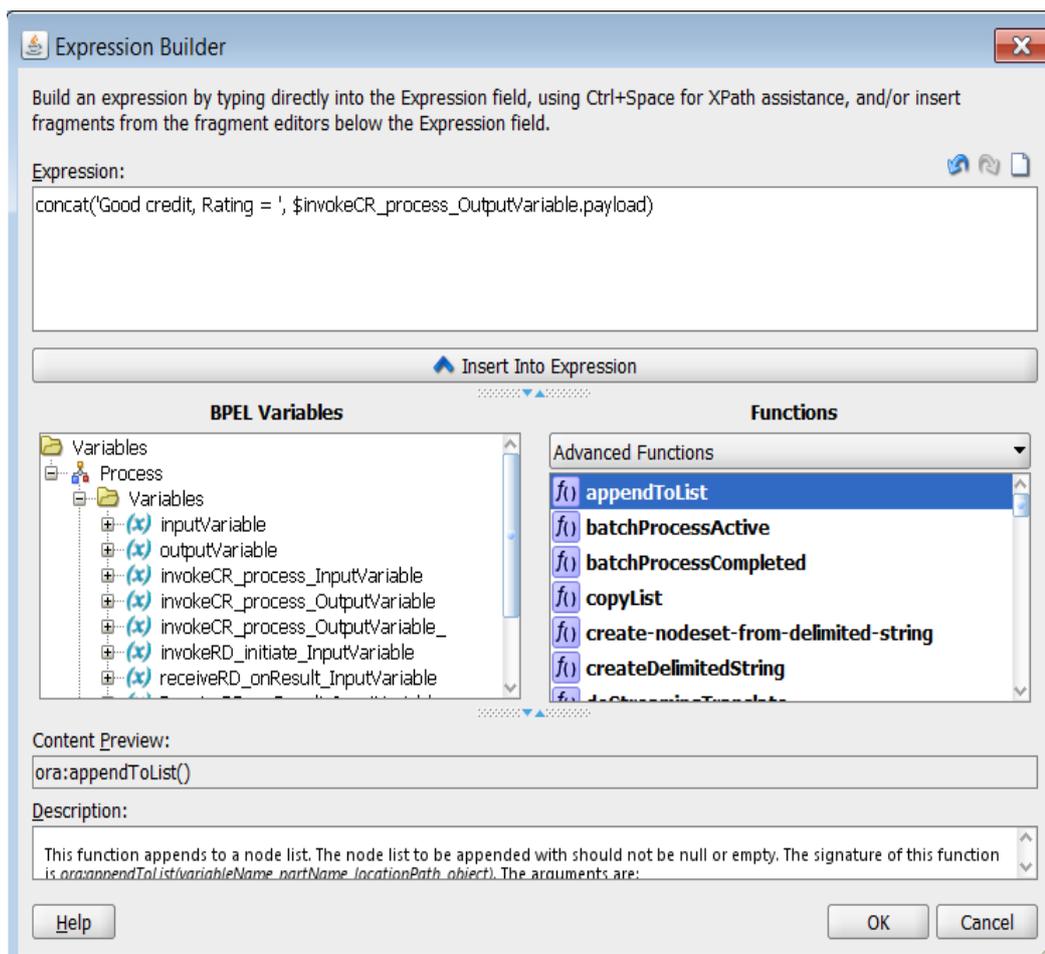


Figura 14 – uso de funções na manipulação das variáveis.

Fonte: Autoria Própria.

O componente *Invoke* é responsável por chamar o *Partner link* e assim conseguir estender o chamado até o serviço disponível no *Web service*. Em suas configurações basta indicar qual é o *Partner Link* a ser acessado e criar variáveis para receber a resposta do serviço. Como mostrado na figura 15:

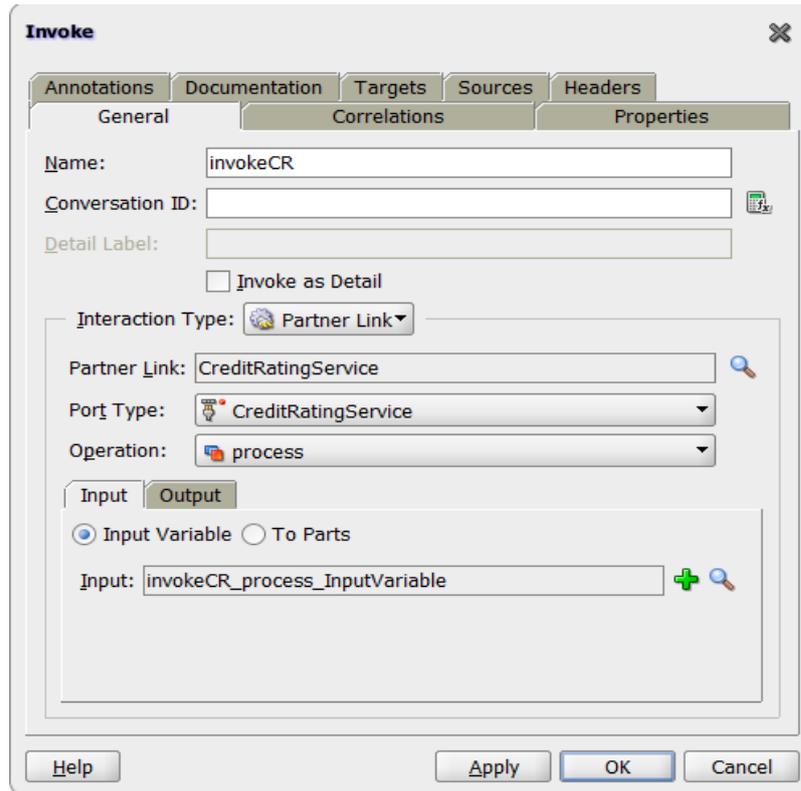


Figura 15 – configurações do componente Invoke.
Fonte: Autoria Própria.

As atividades dentro de um escopo referem-se geralmente a mesma ação, sendo assim melhor organizado o processo separando-o por partes. Um escopo não tem um limite de atividades, mas sempre sua estrutura quando expandida fica semelhante a apresentada na figura 16:



Figura 16 – estrutura do scopo responsável por acessar uma atividade de consulta de crédito.
Fonte: Autoria Própria.

3.1.6 Trabalho com fluxos paralelos

Fluxos paralelos permitem executar mais de uma ação ao mesmo tempo, diminuindo o tempo final de espera por um resultado da execução de ações do processo.

Para a representação deste mecanismo, criou-se um processo para pedir um orçamento em dois dos fornecedores da empresa. Os dois orçamentos são encaminhados no mesmo momento, cada um deles representado por uma sequência dentro do componente *Flow*, mostrado na figura 17.

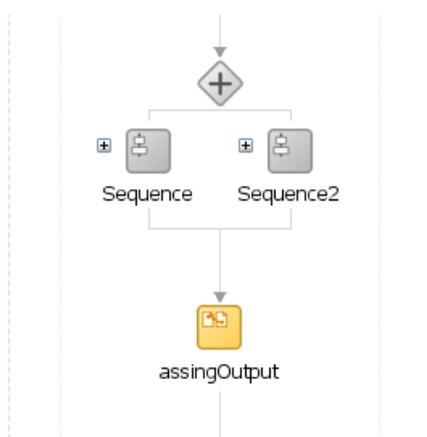
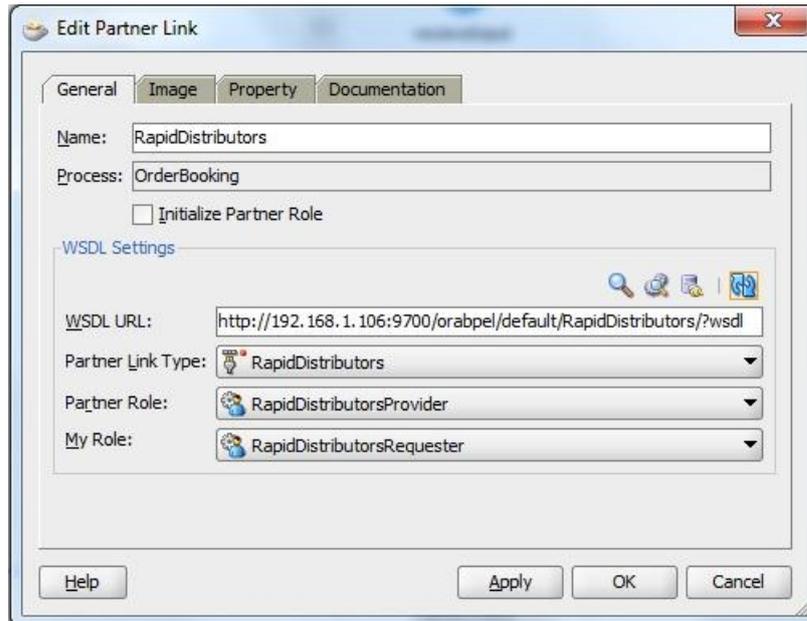


Figura 17 – representação componente Flow.
Fonte: A autoria Própria.

A definição dos componentes para cada *Sequence* é feita com base nos recursos que eles devem acessar, neste caso os dois serão semelhantes, pois terão a ideia de fazer orçamentos, porém utilizando serviços diferentes que serão definidos por *Partner Links* acessando diferentes WSDL, um para cada respectivo fornecedor. Um exemplo da configuração é exibido na figura 18.



**Figura 18 – Partner Link para primeiro fornecedor com especificação de WSDL.
Fonte: Autoria Própria.**

Para consumir o serviço especificado com o WSDL indicado no *Partner link* da figura 18, tem-se que ter um componente para fazer tal invocação. As configurações do componente responsável por tal invocação são mostradas na figura 19.

The image shows a dialog box titled "Invoke" with a close button (X) in the top right corner. The dialog has several tabs: "Annotations", "Documentation", "Targets", "Sources", and "Headers". Below these, there are three sub-tabs: "General", "Correlations", and "Properties". The "General" sub-tab is active and contains the following fields and options:

- Name:** invokeRD
- Conversation ID:** (empty field)
- Detail Label:** (empty field)
- Invoke as Detail
- Interaction Type:** Partner Link (dropdown menu)
- Partner Link:** RapidDistributors (text field with a search icon)
- Port Type:** RapidDistributors (dropdown menu)
- Operation:** initiate (dropdown menu)
- Input/Output:** A section with two tabs: "Input" (selected) and "Output". Under "Input", there are two radio buttons: "Input Variable" (selected) and "To Parts". Below the radio buttons is a text field containing "invokeRD_initiate_InputVariable" with a green plus icon and a search icon to its right.

At the bottom of the dialog, there are four buttons: "Help", "Apply", "OK", and "Cancel".

Figura 19 – configuração para invocação do serviço no primeiro fornecedor.
Fonte: Autoria Própria.

Cada uma das sequências para consulta de preço possui também um componente *Receive* para receber a informação do servidor. A configuração é relativamente simples e mostrada na figura 20:

The image shows a 'Receive' configuration dialog box with the following details:

- Tabbed Interface:** Documentation, Targets, Sources, Headers, General (selected), Correlations, Properties, Annotations.
- Name:** ReceiveRD
- Conversation ID:** (empty field)
- Create Instance:**
- Interaction Type:** Partner Link
- Partner Link:** RapidDistributors
- Port Type:** RapidDistributorsCallback
- Operation:** onResult
- Radio Buttons:** Variable (selected), From Parts
- Variable:** ReceiveRD_onResult_InputVariable
- Buttons:** Help, Apply, OK, Cancel

Figura 20 – configuração do componente receive para um partner Link especificado.

Fonte: Autoria Própria.

Tendo base nos componentes utilizados até este ponto do trabalho, pode-se visualizar na figura 21, o resultado dessas ações na estrutura do processo de negócio e assim melhor esclarecer a estrutura do componente Flow.

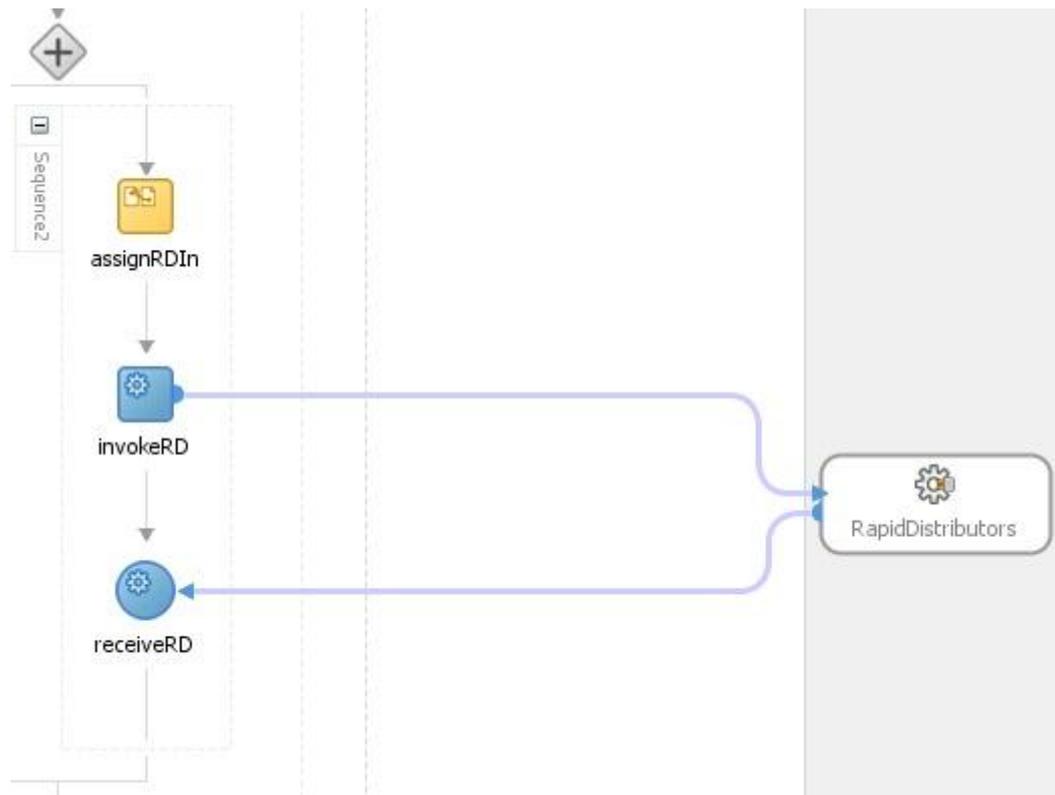


Figura 21 – estrutura do primeiro Sequence do componente Flow para execução de tarefas paralelas.

Fonte: Autoria Própria.

Os componentes do segundo *Sequence* utilizado dentro do *Flow* tem os mesmos princípios, alterando-se apenas o caminho do WSDL no *Partner link*, configuração do *Invoke* para o novo *Partner link* e o componente *Receive* para o tratamento da resposta vinda do serviço. O resultado segue na figura 22.

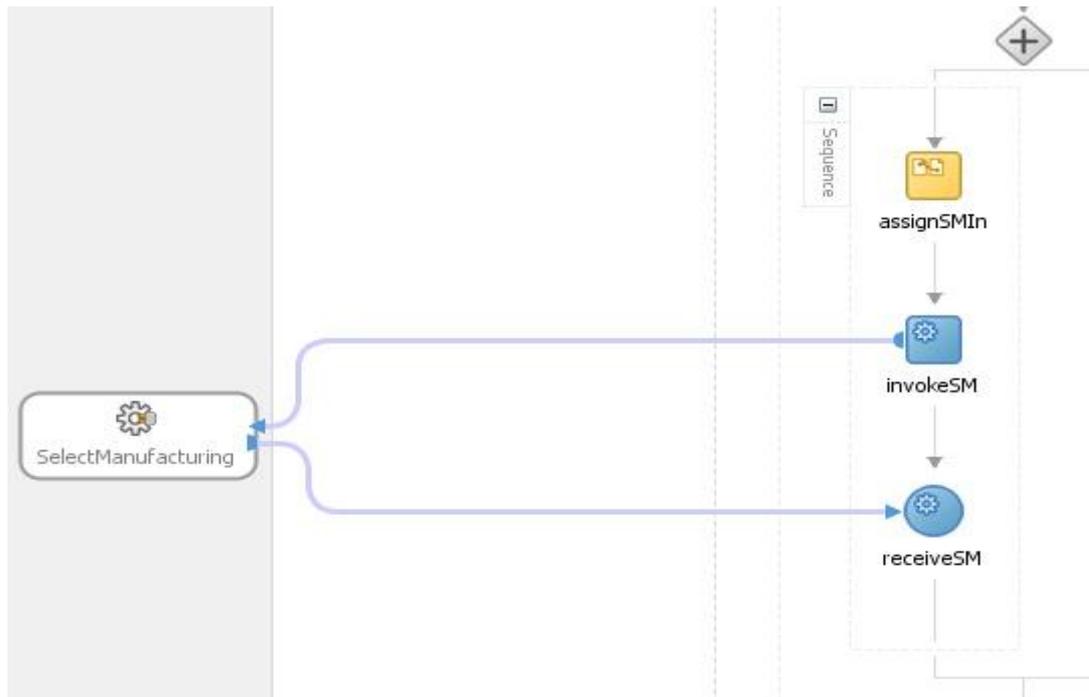


Figura 22 – estrutura do segundo *Sequence* do componente *Flow* para execução de tarefas paralelas.

Fonte: Autoria Própria.

O componente *Flow* permite mais de cinquenta sequencias de execução paralela, isso garante suporte para a robustez do JDeveloper para implementar e gerenciar processos de negócios, independente da complexidade que este apresentar. O componente montado com as duas sequências de pesquisa de preço a fornecedores é apresentado a seguir na figura 23:

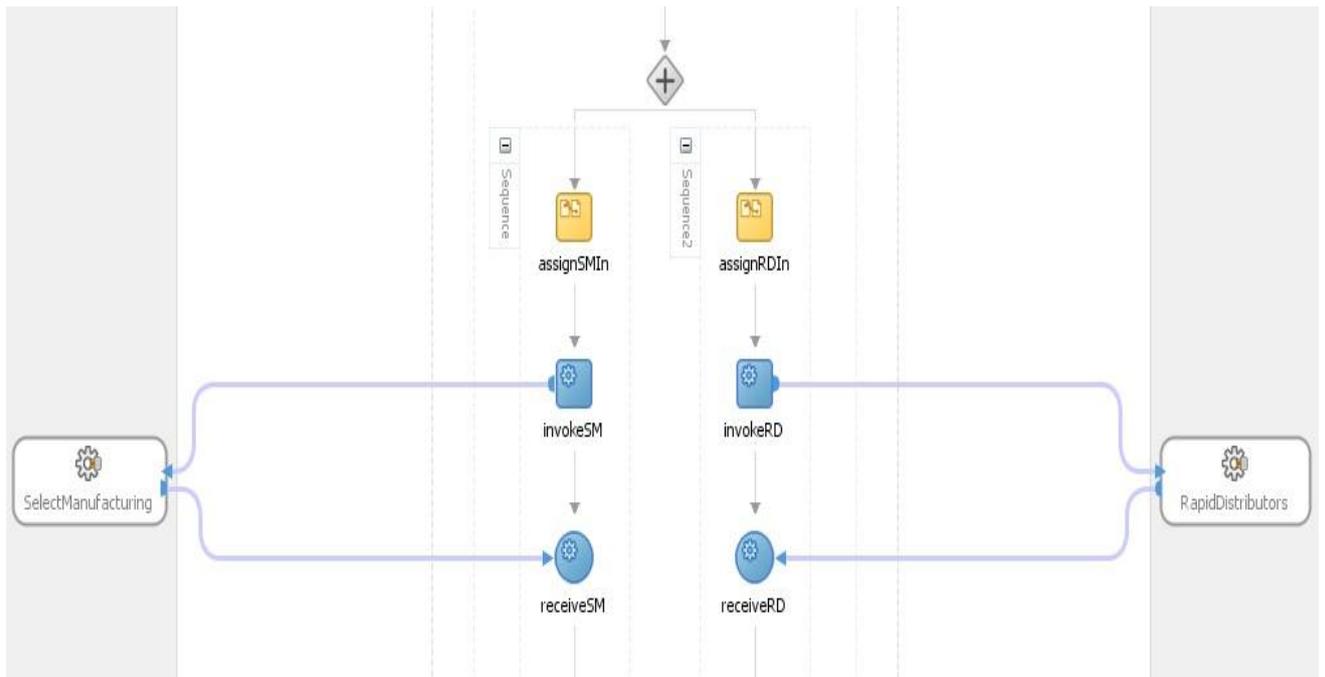


Figura 23 – Componente *Flow* com duas sequencias montadas exemplificando o acesso simultâneo a dois serviços.

Fonte: Aatoria Própria.

Assim, a modelagem para demonstração do processo de negócio com BPEL chega ao fim, tendo um visual bastante simples e de fácil entendimento dos componentes utilizados. A figura 24 exibe a configuração final do processo.

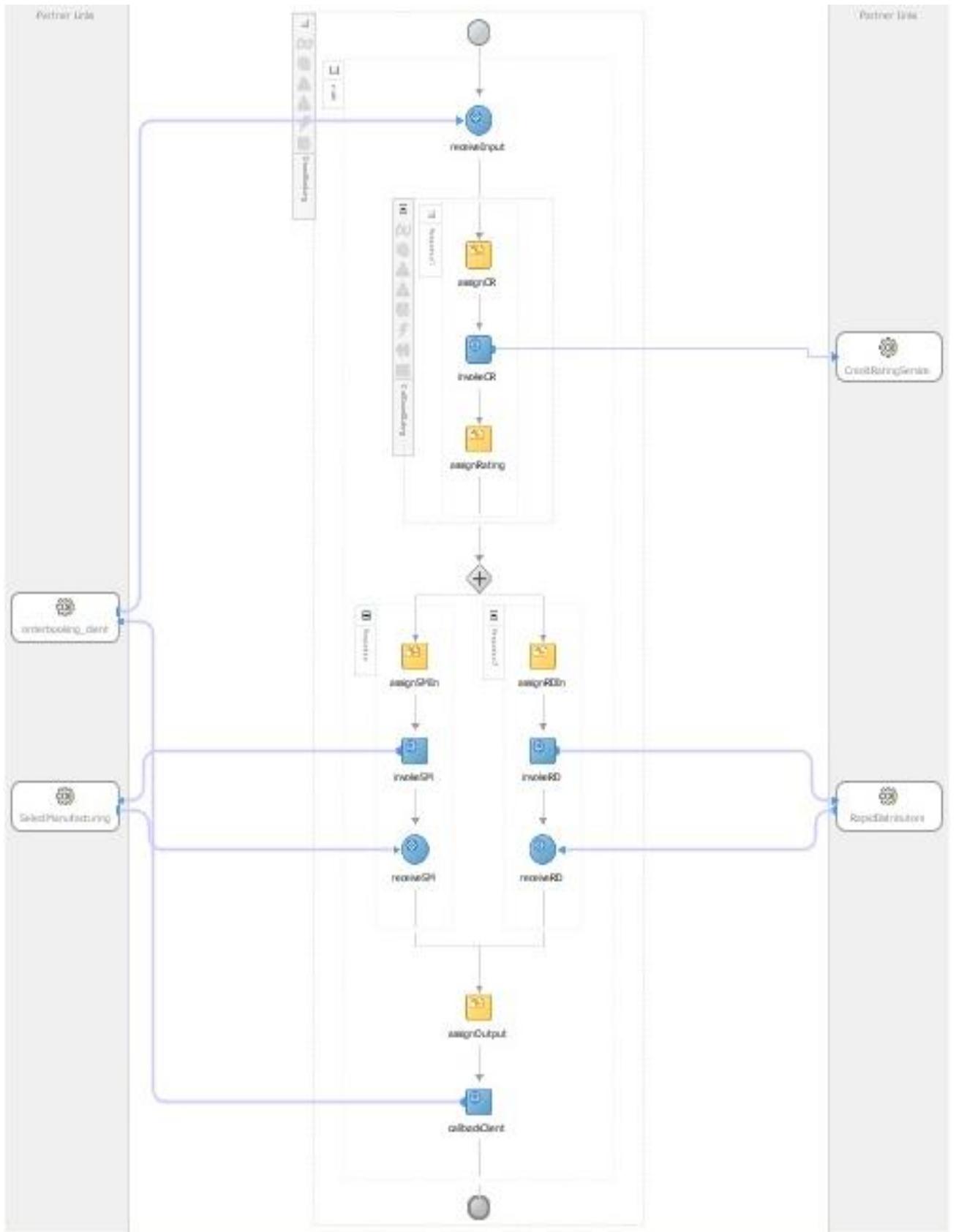


Figura 24 – Processo de negócio completo com expansão dos componentes de agrupamento utilizados.
Fonte: Autoria Própria.

3.1.7 Deploy do projeto

Feita a modelagem é necessário fazer o *deploy* para então começar funcionar e aproveitar os recursos oferecidos dos processos de negócios para uma administração eficiente da empresa.

Neste caso o *deploy* foi feito direto do JDeveloper para um servidor *Weblogic*, podendo ser feito também por meio do *prompt* de comando. A figura 25 mostra como fazer o *deploy* no JDeveloper.

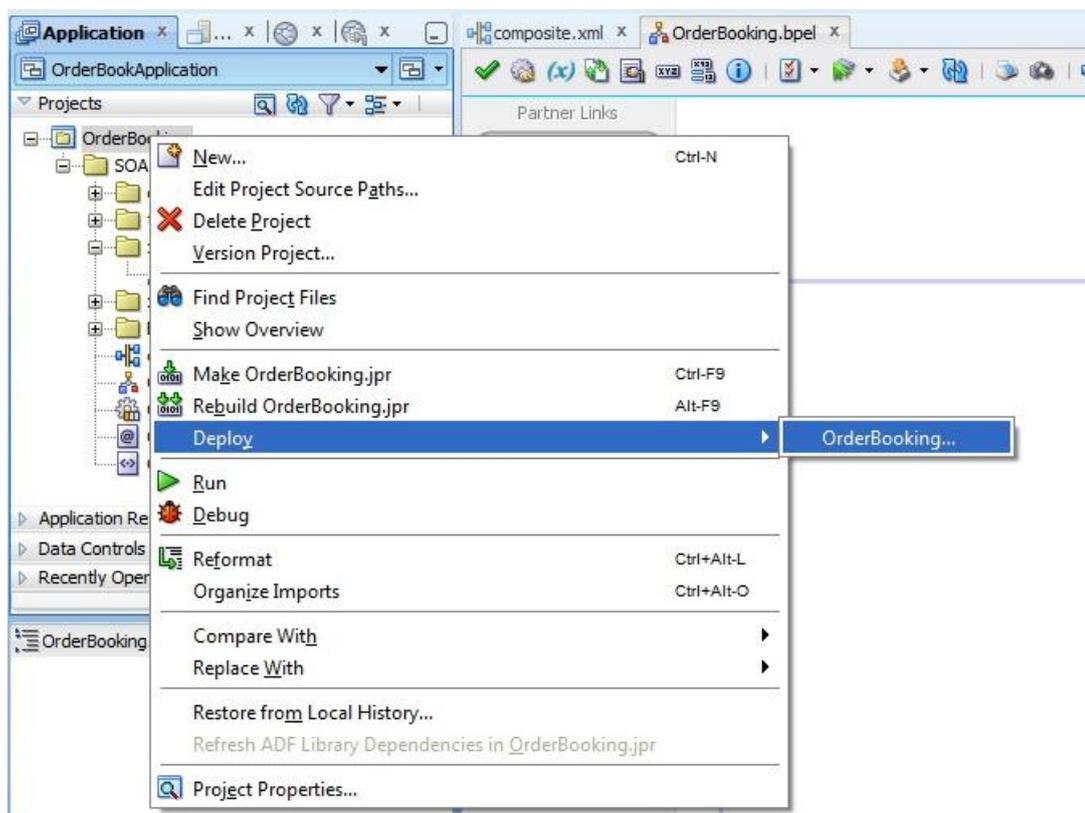


Figura 25 – Opção de *deploy* do projeto apresentado através do JDeveloper.

Fonte: Autoria Própria.

É necessário fazer o *deploy* de todas as versões construídas do projeto, para que sejam melhor testadas e com isso é possível ter várias versões rodando no servidor. Para isso não confundir ou causar outros problemas, pode-se fazer o controle de versões nas próprias configurações do *deploy*, indicando qual é a versão que se está trabalhando no momento. A figura 26 exhibe as opções de configuração para versionamento.

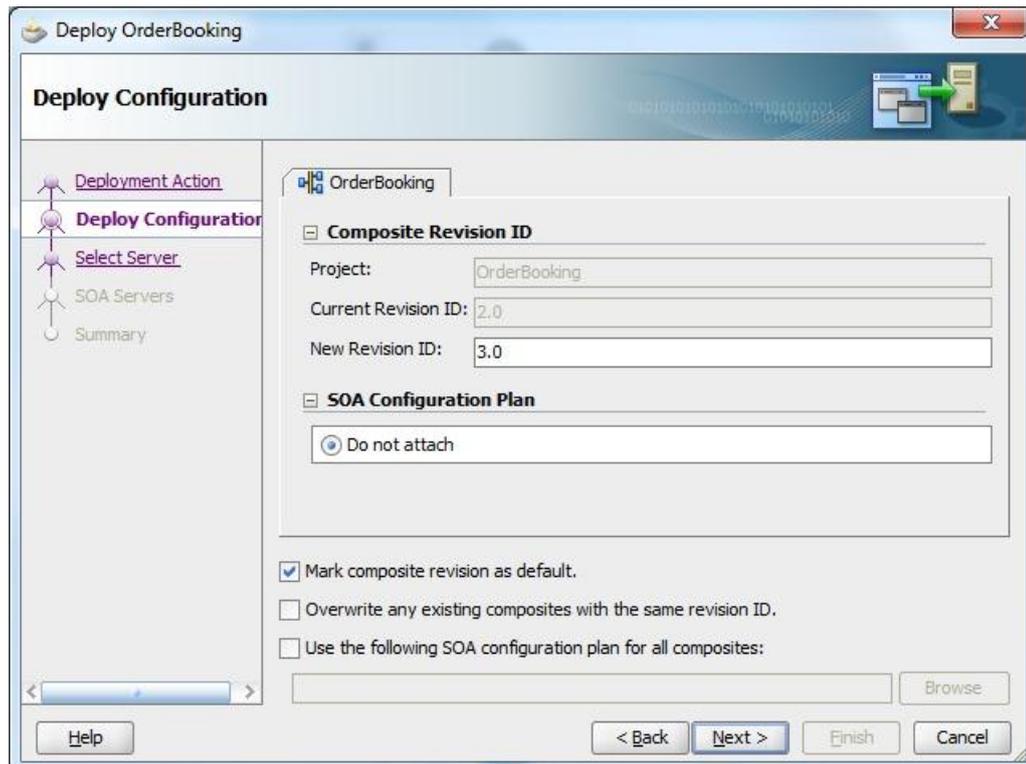


Figura 26 – Controle de versões para deploy do projeto.

Fonte: Autoria Própria.

Para o *deploy* ser concluído é necessário estar com o servidor *Weblogic* ativo e configurado para que o JDeveloper o encontre e possa fazer a conexão e o envio do projeto. Na figura 27 é mostrada a busca feita por servidores ativos.

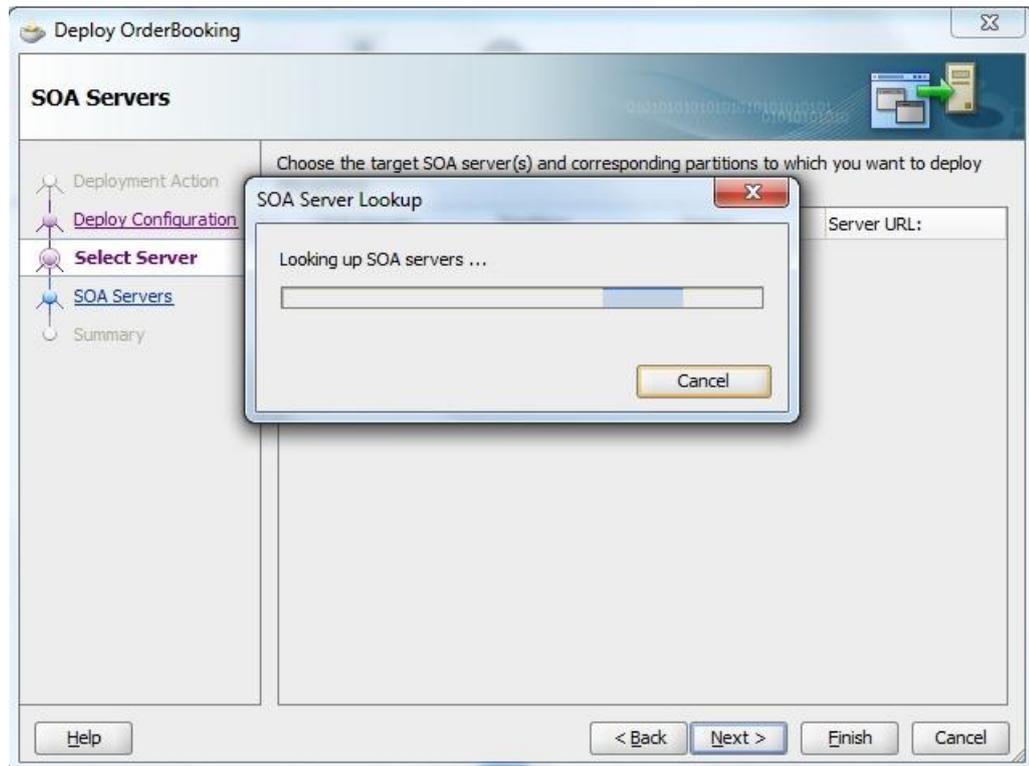


Figura 27 – Pesquisa de servidores ativos para conclusão do *deploy* do projeto trabalhado.

Fonte: Autoria Própria.

Caso o servidor não esteja ativo ou esteja sem suporte para aplicações SOA, não será possível a conclusão do trabalho. Porém se tudo estiver rodando e pronto para ser utilizado, ele será apresentado e basta o desenvolvedor escolher em qual deles deseja fazer o *deploy* de seu processo. A figura 28 mostra o servidor disponível para receber a aplicação.

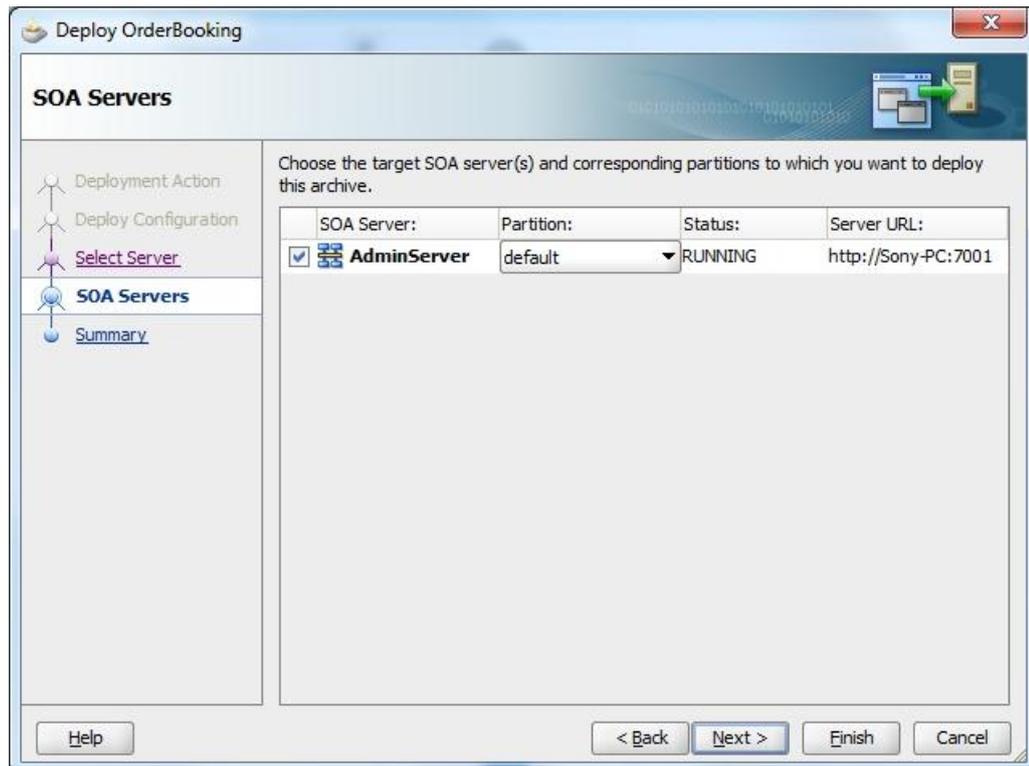


Figura 28 – Servidor ativo e pronto para receber o novo processo.

Fonte: Autoria Própria.

Para se ter certeza que o processo foi corretamente enviado ao servidor de aplicações é preciso verificar na aba SOA – Logo a presença da confirmação “*Build successful*” como mostrado na figura 29, para então poder acessar o console do servidor *Weblogic* e testar o processo.

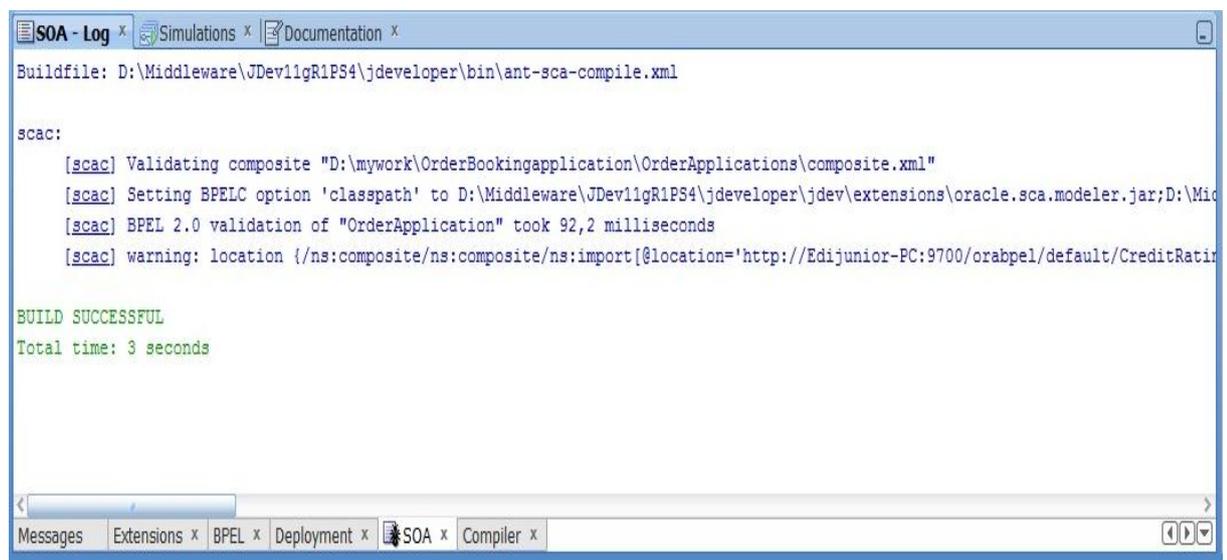


Figura 29 – Confirmação de *deploy* do processo no servidor de aplicações aplicações.

Fonte: Autoria Própria.

Fazer o *deploy* significa que agora o processo está sendo gerenciado por um servidor de aplicações e outras pessoas podem ter acesso a ele por meio de conexões externas e assim poder visualizar a execução e sugerir mudanças, melhorando o processo. A figura 30 mostra a estrutura do processo no servidor de aplicações.

The screenshot displays the Oracle Enterprise Manager 11g Fusion Middleware Control interface. The main content area shows the configuration for the 'OrderApplications [3.0]' composite. The 'Falhas e Mensagens Rejeitadas Recentes' section is currently empty, indicating no recent errors. The 'Métricas do Componente' table shows the following data:

Nome	Tipo de Componente	Total de Instâncias	Instâncias em Execução	Instâncias com Falhas Recuperável	Instâncias com Falhas Não Recuperável
OrderApplication	BPEL	0	0	0	0

The 'Serviços e Referências' table shows the following data:

Nome	Tipo	Uso	Falhas	Total de Mensagens	Tempo Médio de Processamento (seg)
orderapplication_client_ep	Web Service	Serviço	0	0	0,000
RapidDistributors	Web Service	Referência	0	0	0,000
RapidDistributor_2	Web Service	Referência	0	0	0,000
CreditRatingService	Web Service	Referência	0	0	0,000

Figura 30 – Estrutura do processo de negócio no servidor referenciando os *Web Services* utilizados.

Fonte: Autoria Própria.

Com a confirmação do processo sendo executado no servidor é possível testa-lo e verificar o fluxo de requisições aos serviços indicados através dos WSDL indicados nos componentes *Partner link* e acessados por componentes *Invoke*.

Para o teste é preciso preencher alguns campos com as informações necessárias para a execução do serviço, como exibido na figura 31.

Testar Web Service ?
Testar Web Service

Use esta página para testar qualquer WSDL, inclusive os que não estão na farm. Para testar um Web service, informe o WSDL e clique em **Fazer Parse de WSDL**. Quando a página for atualizada com os detalhes do WSDL, primeiro selecione o Serviço, selecione a Porta e, em seguida, selecione a Operação que deseja testar. Especifique os parâmetros de entrada e clique em **Testar Web Service**.

WSDL: Fazer Parse de WSDL

Opção de Autenticação Básica HTTP para o Acesso do WSDL

Serviço:

Porta:

Operação:

URL de Ponto Final: Editar URL de Ponto Final

Solicitação

Resposta

Segurança

Qualidade do Serviço

Opções de Transporte HTTP

Opções de Teste Adicionais

Argumentos de Entrada

View em Árvore

Nome	Tipo	Valor
[-] *payload	PurchaseOrderType	
*CustID	string	<input type="text"/>
*ID	string	<input type="text"/>
[-] *ShipTo	USAddress	
*BillTo	USAddress	
*UserContact	ContactType	
[-] *OrderItems	OrderItemsType	
*SupplierInfo	SupplierInfoType	
*OrderInfo	OrderInfoType	

Figura 31 – Entrada de dados para teste de *Web Services*.

Fonte: Autoria Própria.

Há outro servidor BPEL que está hospedando o *Web Services*. Nos quais foi feito o *deploy* de um *Web Service* de exemplo para geração dos WSDL e o uso no processo de negócio demonstrado neste trabalho. A figura 32 mostra o console BPEL.

The screenshot shows the Oracle Enterprise Manager 10g BPEL Control interface. The browser window title is 'Console do Oracle BPEL v10.1.3.1.0'. The URL is 'localhost:9700/BPELConsole/default/index.jsp'. The page header includes 'ORACLE Enterprise Manager 10g BPEL Control' and navigation links for 'Gerenciar Domínio BPEL', 'Logout', and 'Suporte'. The user is logged in as 'default'. The main navigation tabs are 'Painel de Controle', 'Processos BPEL', 'Instâncias', and 'Atividades'. The 'Instâncias' tab is active, displaying a table of BPEL processes and instances.

Processos BPEL Disponibilizados		Instâncias de Processos BPEL Em Execução 1 - 1	
Nome	Instância	BPEL_Process	Última Modificação ↑
CreditRatingService	3 : Instance #3 of SelectManufacturing	SelectManufacturing (v. 1.0)	04/06/12 20:55:43
RapidDistributors			
SelectManufacturing			
TaskActionHandler			
TaskManager			
Instâncias de Processos BPEL Recém-Concluídos (Mais...)			
	✓ 4 : Instance #4 of RapidDistributors	RapidDistributors (v. 1.0)	04/06/12 20:56:11
	✓ 2 : Instance #2 of CreditRatingService	CreditRatingService (v. 1.0)	04/06/12 20:55:39
	✓ 1 : Instance #1 of CreditRatingService	CreditRatingService (v. 1.0)	04/06/12 19:34:48

At the bottom left, there is a button labeled 'Disponibilizar Novo Processo'. At the bottom right, the version 'Console do Oracle BPEL v10.1.3.1.0' is displayed.

Figura 32 – Web Services sendo executados no console BPEL prontos para receber requisições de processamento de informação.

Fonte: Autoria Própria.

Com o processo rodando é possível analisar o que está acontecendo e para um gerente de negócio isso fica viável para absorver problemas e sugerir mudanças úteis para o melhoramento do processo. O acompanhamento da execução pode ser visto no console do servidor *Weblogic*, na figura 33.

Rastreamento de Fluxo > Instância de OrderApplication

Instância de OrderApplication

Esta página mostra os detalhes da instância do processo BPEL.

Trilha de Auditoria | Fluxo | Valores Sensor | Falhas

Expandir um nó de payload para exibir os detalhes.

AssingCR	04/06/2012 17:55:39	Variável atualizada "invokeCR_process_InputVariable"
	04/06/2012 17:55:39	Designação concluída
invokeCR	04/06/2012 17:55:39	Started invocation of operation "process" on partner "CreditRatingService".
	04/06/2012 17:55:39	Operação bidirecional foi chamada "process" no parceiro "CreditRatingService".
		<payload>
AssingRating	04/06/2012 17:55:39	Variável atualizada "inputVariable"
	04/06/2012 17:55:39	Designação concluída
<GetPriceQuote (127)>		
<Sequence (128)>		
AssingSMIn	04/06/2012 17:55:39	Variável atualizada "invokeRD_initiate_InputVariable"
	04/06/2012 17:55:39	Designação concluída
invokeSM	04/06/2012 17:55:39	Started invocation of operation "initiate" on partner "RapidDistributor_2".
	04/06/2012 17:55:40	Operação unidirecional foi chamada "initiate" no parceiro "RapidDistributor_2".
		<payload>
ReceiveSM (pendente)		
	04/06/2012 17:55:40	Aguardando "onResult" de "RapidDistributor_2". Callback assíncrono.
<Sequence2 (147)>		
AssingRDIn	04/06/2012 17:55:39	Variável atualizada "invokeCR_process_InputVariable"
	04/06/2012 17:55:39	Designação concluída
invokeRD	04/06/2012 17:55:40	Started invocation of operation "initiate" on partner "RapidDistributors".
	04/06/2012 17:55:40	Operação unidirecional foi chamada "initiate" no parceiro "RapidDistributors".
		<payload>
ReceiveRD (pendente)		
	04/06/2012 17:55:41	Aguardando "onResult" de "RapidDistributors". Callback assíncrono.

Figura 33 – acompanhamento da execução do processo de negócio.

Fonte: Autoria Própria.

Através deste mesmo console é possível analisar o fluxo sendo executado e se obter maiores detalhes da modelagem do processo. A figura 34 exibe esse fluxo.

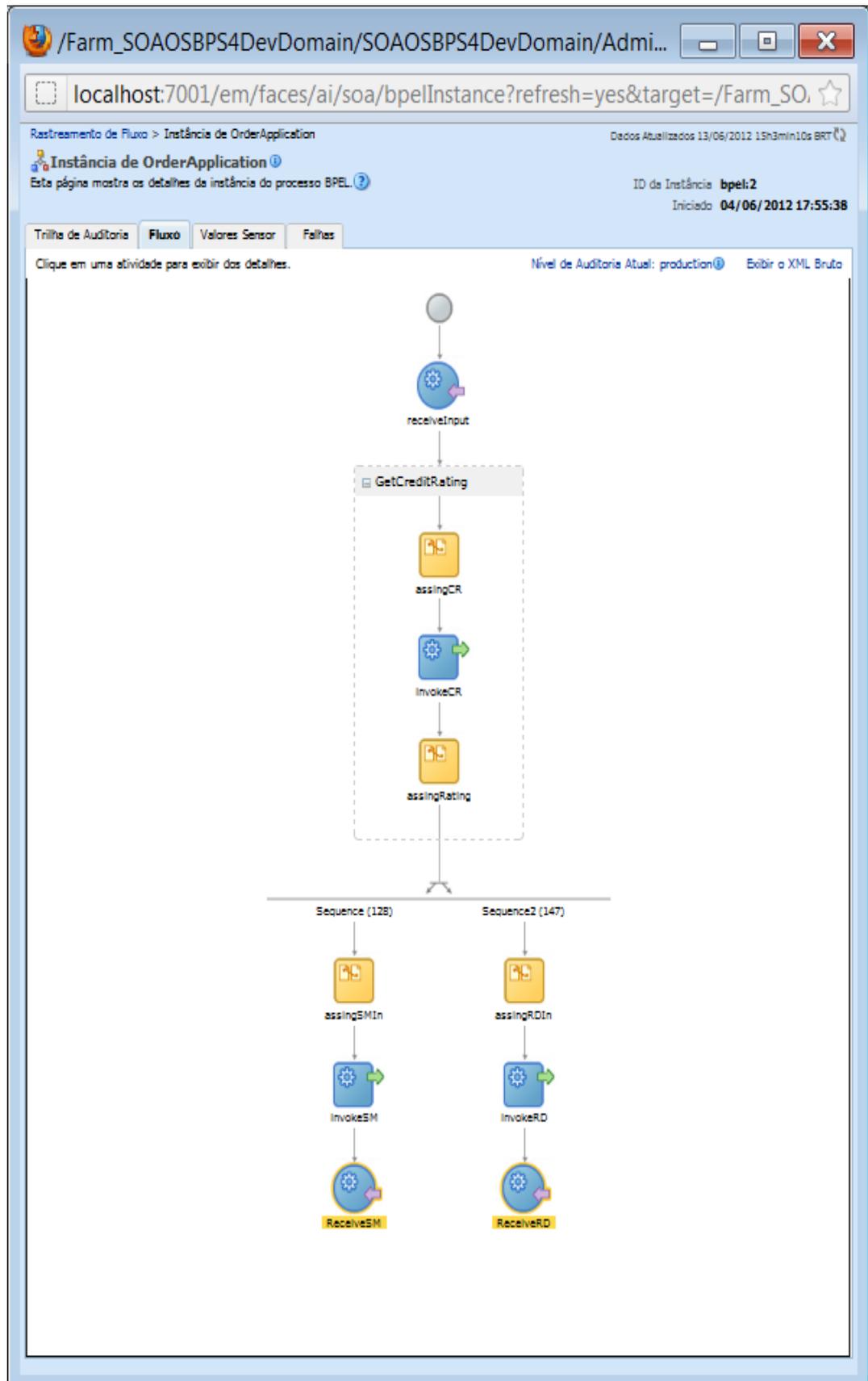


Figura 34 – modelagem do processo vista através do console do servidor *Weblogic*.
Fonte: Autoria Própria.

A modelagem vista através do servidor não pode ser alterada online, ficando responsável por mudanças o grupo responsável pela modelagem e implementação do processo através de ferramentas de desenvolvimento. Assim o processo de negócio está pronto e pode ser conferido por gerentes ou quaisquer outros que desejam obter informações sobre o funcionamento da empresa.

4 CONSIDERAÇÕES FINAIS

4.1 CONCLUSÃO

Este trabalho abordou principalmente processos de negócios e suas vantagens quando bem organizados e modelados. Mostrou também a importância da modelagem adequada desses processos para o sucesso das grandes empresas, sendo ainda uma oportunidade para pequenas e médias organizações adotarem e melhorem seus negócios.

A atenção dada ao assunto no mundo corporativo chama a atenção e desperta o interesse por não ser algo comumente usado ou divulgado, ficando por enquanto seu uso restrito a grandes corporações.

A aplicação da tecnologia estudada em um estudo experimental rendeu conhecimento sobre a integração de sistemas diferentes, nos quais se destacou o uso de ferramentas e tecnologias que facilitam essa integração, reduzindo custos para empresas manterem seus sistemas e mesmo assim aplicar novas tecnologias mantendo seus sistemas atuais e atendendo suas necessidades.

O presente trabalho também foi importante para apresentar novos horizontes de uma área de grande importância no mercado e que une duas áreas importantes: a área de Tecnologia da Informação, com suas inúmeras aplicações e maneiras de melhorar, e a área de Administração de Empresas, responsável por conceitos importantes de mercado.

Ambos os recursos utilizados não foram de fácil instalação (Oracle Soa Suíte e BPEL *Process Manager*), o que permite destacar aqui a necessidade de certa persistência e dedicação para a realização de atividades similares às que foram apresentadas no capítulo 3.

4.2 TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO

Processos de negócios são de grande importância para as organizações conforme apresentado neste trabalho, porém ainda são modelados quase que exclusivamente pelas grandes corporações. Neste trabalho foi apresentada uma estrutura relativamente nova nessa área, tanto na linguagem na versão BPEL 2.0 quanto nas ferramentas Oracle utilizadas, ficando aberto a novas experiências de uso para destacar novas funcionalidades ou algum melhoramento de outras já existentes.

Destacada a importância de um bom controle sobre os processos de negócios, o desenvolvimento de ferramentas gratuitas para facilitar a inserção do conceito no mercado surge como sugestão para futuros trabalhos, e assim despertar o interesse de pequenas e médias empresas para o assunto.

REFERÊNCIAS BIBLIOGRÁFICAS

ANDREWS, Tony; CUBERA, Francisco; DHOLAKIA, Hitesh; GOLAND, Yaron; KLEIN, Johannes; LEYMANN, Frank; LIU, Kevin; ROLLER, Dieter; SMITH, Doug; THATTE, Satish; TRICKOVIC, Ivana; WEERAWARANA, Sanjiva. **Business Process Execution Language for Web Services Version 1.1**. 2003. Acessado em 15 de Fevereiro de 2012.

BARROS, Breno. **Devo usar BPEL ou ESB**. 2008. Disponível em <<http://soa-journal.blogspot.com/2008/06/devo-usar-bpel-ou-esb.html>> Acessado em 03 de Fevereiro de 2012.

BLANVALET, Stany et al. **BPEL Cookbook: Best Practices for SOA-based integration and composite applications development**. Birmingham: Packt Publishing, 2006.

DÁVALOS, Ricardo Villarroel. **Modelagem de Processos . 2. ed**. Palhoça: Unisulvirtual, 2006.

ERIKSSON, H. E.; PENKER, M. **Business Modeling with UML**. New York: John Wiley & Sons, 2000.

ERL, Thomas. **SOA: Princípios de Design de Serviços**. São Paulo: Pearson Prentice Hall, 2009.

GALDINO, Fernando. **O que é Soa Suite 11g?**. 2009. Disponível em <https://blogs.oracle.com/galdino/entry/o_que_soa_suite_11g > Acessado em 05 de Abril de 2012.

GARIMELA, Kiran; LEES, Michael; WILLIAMS, Bruce. **BPM Basics**. Indianapolis: Wiley Publishing, 2008.

GEYER, Carol; **POR QUE USAR BPEL**. 2007. Disponível em <<http://bpel.xml.org/why-use-bpel>> Acessado em 28 de Março de 2012.

GHALIMI, Ismael. **BPM 2.0**. 2007. Disponível em <<http://pt.scribd.com/doc/59091000/7/BPEL>> Acessado em 03 de Fevereiro de 2012.

JURIC, Matjaz B. **Business Process Execution Language for Web Services, Second Edition**. 2006.

JURIC, Matjaz B; KRIZEVNIK, Marcel. **WS-BPEL 2.0 for SOA Composite Applications with Oracle SOA Suite 11g**. 2010.

MCDONALD, Mark. **Improving Business Process**. 2010.

ROCHA, Gilberto; VERDE, Isidro Vila; PEREIRA, Rui Humberto. **Implementação de um modelo baseado em XML para suporte da dinâmica processual de negócio**. 2006. Acessado em 13 de Fevereiro de 2012.

SANTANA, Fabiana S; FERREIRA, Jefferson C; STANGE, Renata L; SARAIVA, Antônio M. **Uso de Wrappers e Web Services para a construção de modelagem de nicho ecológico em SOA**. 2009. Acessado em 03 de Março de 2012.

WEERAWARANA, Sanjiva. **Web Services plataforma architecture: SOAP, WSDL, WSPolicy, WS – Addressing, WS – BPEL, WS-Reliable Messaging, and more**. 2005. Acessado em 10 de Fevereiro de 2012.

WHITE, Stephen A; MIERS, Derek. **BPMN Modeling Reference Guide**. 2008. Disponível em <<http://pt.scribd.com/doc/66046505/7/As-Linguagens-de-Execucao-e-os-Processos-de-Negocio>> Acessado em 15 de Abril de 2012.