

Victor Manuel Moreira Martins

**Integração
de
Sistemas de Informação:
Perspectivas, normas e abordagens.**

Universidade do Minho
Guimarães
2005

Agradecimentos

Aos meus pais e ao meu irmão pelos constantes incentivos, e em especial à minha esposa Cristina, pelo apoio e motivação que sempre manifestou.

Ao meu orientador, Professor Doutor João Álvaro Carvalho, pelos seus comentários, críticas e sugestões e pela disponibilidade sempre demonstrada.

À Oracle Portugal que me proporcionou as condições necessárias para poder realizar este trabalho.

A todos muito obrigado.

Victor Manuel Moreira Martins

Resumo

As Tecnologias da Informação (TI) permitem definir várias arquitecturas de integração e possibilitam diferentes abordagens para um mesmo problema. É uma realidade confusa onde existe uma forte competitividade tecnológica que influencia a adopção de diferentes soluções. Neste âmbito, as TI suportam e controlam a comunicação entre sistemas heterogéneos permitindo a sua compatibilidade e integração. Algumas soluções seguem normas definidas e actualizadas por empresas ou organismos credenciados o que vem facilitar a integração dos Sistemas de Informação (SI).

À primeira vista, a classificação das diferentes soluções para a integração de SI não se revela fácil ou evidente. No entanto as abordagens podem centrar-se quer na informação quer nas aplicações ou nos processos organizacionais. Estas três áreas correspondem a uma qualificação que serve de ponto de partida para a ordenação de ideias neste trabalho. Realça-se também que a integração orientada aos processos organizacionais é fundamental para a real adequação da solução à organização. Esta perspectiva tem influenciado o aparecimento de novas normas técnicas como por exemplo o *Business Process Execution Language* (BPEL). Com a evolução das tecnologias, há uma crescente complementaridade das abordagens existentes o que leva à definição de novas arquitecturas como por exemplo o *Service Oriented Architecture* (SOA). Na implementação do SOA, a norma BPEL permite especificar uma lógica processual de integração de sistemas, disponibilizados como serviços (*Web Services*), e integráveis de acordo com as especificações técnicas correspondentes.

Como ilustração de conceitos, e com objectivo experimental, exemplifica-se uma implementação técnica baseada em algumas das tecnologias e normas descritas neste trabalho. O objectivo deste exemplo é o de mostrar na prática diferentes perspectivas tecnológicas para a integração de SI, implementando duas alternativas diferentes. O BPEL e os *Web Services* são utilizados na segunda alternativa de forma a mostrar as vantagens da sua utilização. No final são tecidas algumas conclusões acerca da catalogação encontrada, do exemplo criado e da evolução tecnológica nesta área de actuação.

Abstract

Information Technologies (IT) allow to define various integration architectures and permit different approaches to the same problem. This is a confusing reality where there is a lot of competitive technologies that influences the adoption of different solutions. In that scope IT supports and controls the communication between heterogeneous systems allowing their compatibility and integration. Some solutions follow standards defined by credited enterprises or organizations facilitating Information Systems (IS) integration.

The classification of the different solutions for IS integration is not easy or evident at the beginning. However the approaches can centralize whether into information, into applications or into business processes. Those three areas correspond to a way of classification that serves as a start to put in order the ideas in this work. There is an emphasis in the process centric integration because of his importance to make the integration solution suitable to the business. This specific view of the IS integration influences the appearance of new standards like, for instance, the *Business Process Execution Language* (BPEL) standard. With the technology evolution there is growing complementarity and convergence of the approaches, and this allows new architecture definitions as for example the *Service Oriented Architecture* (SOA). In that particular case the BPEL standard permits a procedural way of specifying IS integration. Systems are accessible through *Web Services* and can be integrated based on technical specifications defined by the corresponding standards.

To illustrate those concepts, and also as an experimental objective, an IS integration example is created. It is based on some technologies and standards described in this work. In that case, the main objective is to show in a practical way different technological perspectives for IS integration implementing two alternatives. *Web Services* and BPEL are used in the second approach to show their advantages. In the final some conclusions are expressed about the classification, the integration example and the technology evolution in this area.

“It is not the strongest of the species that survive, nor the most intelligent, but the one most responsive to change.”

– **Charles Darwin**

"Things should be made as simple as possible, but no simpler."

- **Albert Einstein**

Índice

Agradecimentos.....	iii
Resumo	iv
Abstract	v
Lista de figuras	ix
Lista de tabelas	xvii
1. Introdução	1
1.1 Objectivos da dissertação.....	3
1.2 Conceitos e pressupostos	4
1.3 Organização dos capítulos	5
2. A integração de Sistemas de Informação	7
2.1 A necessidade de integração de SI.....	8
2.2 As formas e perspectivas de integração de SI.....	10
2.2.1 Integração da informação.....	15
2.2.2 Integração aplicacional	18
2.2.3 Integração de processos	23
2.2.4 Integração inter-organizacional	26
2.3 Os Web Services na integração de SI	27
2.4 As camadas tecnológicas e a sua distribuição.....	28
2.5 Papel das normas para a integração de SI.....	33
2.6 Condicionantes da integração de SI.....	36
2.7 Conclusão.....	38
3. Abordagens para a integração de SI	40
3.1 A Integração da informação	41
3.1.1 Ficheiros, SGBD e Replicação de Informação	42
3.1.2 ETL, TPM, EII, ECM e XML.....	45
3.2 A Integração aplicacional	51
3.2.1 Integração aplicacional dentro da organização	53
3.2.1.1 Plataformas aplicacionais: .NET e J2EE	58
3.2.1.2 Servidores aplicacionais e Integration Brokers.....	61
3.2.2 Integração aplicacional entre organizações	64
3.3 A Integração dos processos organizacionais.....	67
3.3.1 <i>Workflow</i> e <i>Business Process Management</i>	68
3.3.2 <i>Business Process Execution Language</i>	71
3.4 Web Services e a arquitectura Service Oriented Architecture.....	75
3.5 Web Services versus abordagens tradicionais para a integração de SI.....	81
3.6 Relacionamento das perspectivas para a integração de SI.....	82
4. Soluções para a integração de Sistemas de Informação	89
4.1 O Extended Markup Language (XML).....	91
4.2 Soluções para a integração aplicacional dentro da organização	96
4.2.1 <i>Transaction Processing Monitors</i>	96
4.2.2 <i>Remote Procedure Call</i>	97
4.2.3 Componentes COM e JavaBeans.....	98

4.2.4 Objectos distribuídos CORBA, DCOM e Java RMI	100
4.2.5 Messaging (MOM e Message Brokers)	101
4.3 Soluções para a integração inter-organizacional.....	105
4.3.1 EDI.....	105
4.3.2 ebXML.....	106
4.4 Business Process Execution Language (BPEL).....	108
4.5 Os Web Services	111
4.5.1 SOAP	112
4.5.2 WSDL	114
4.5.3 UDDI.....	116
4.6 S�mula	117
5. Implementa�o de um exemplo de integra�o de SI com duas	
abordagens	118
5.1 Estrutura do exemplo	120
5.1.1 Aplica�o de Requisi�es	121
5.1.2 Aplica�o de Recursos Humanos.....	124
5.1.3 Aplica�o de Stocks	125
5.1.4 Descri�o dos pontos de integra�o	126
5.1.5 Processo de aprova�o de requis�es	127
5.1.6 Modelo E-R para o exemplo	128
5.2 Descri�o funcional da primeira abordagem de integra�o	129
5.2.1 Aplica�o de Requisi�es em Oracle Forms Developer 10g.....	129
5.2.2 Acesso � aplica�o de Requisi�es e sua integra�o com Java	131
5.2.3 Integra�o da aplica�o de Requisi�es com solu�o de <i>workflow</i>	135
5.2.4 Integra�o da aplica�o de Requisi�es com solu�o de <i>Messaging</i>	140
5.3 Descri�o funcional da segunda abordagem de integra�o com BPEL.....	144
5.3.1 <i>Web Service</i> para aplica�o de Requisi�es.....	145
5.3.2 <i>Web Service</i> para solu�o de <i>workflow</i> para aprova�o de requis�es ...	150
5.3.3 BPEL para processo de requis�es	154
5.4 S�mula e conclus�es	166
6. Conclus�es.....	169
7. Ap�ndices.....	173
7.1 Cria�o dos utilizadores de base de dados Oracle	173
7.2 Comandos para a cria�o das tabelas e registos na base de dados.....	174
7.3 Database Links.....	177
7.4 C�digo para integra�o do Messaging	177
7.5 Integra�o com Messaging e integra�o com email da aplica�o de Stocks ...	182
7.6 C�digo PL/SQL para <i>Web Service</i> de Inser�o de Requisi�es.....	183
7.7 C�digo PL/SQL para <i>Web Service</i> de <i>Workflow</i>	184
7.8 C�digo de <i>workflow</i> para a integra�o com <i>Requisi�es</i> e <i>Messaging</i>	185
7.9 Classe Java para integra�o com aplica�o de RH	188
7.10 Defini�o BPEL	190
Refer�ncias bibliogr�ficas.....	193

Lista de figuras

Figura 1.1: Evolução e convergência tecnológica	2
Figura 2.1: Principais áreas organizacionais para a integração de SI	7
Figura 2.2: Evolução da integração de SI	8
Figura 2.3: Custo da integração de SI versus nº de aplicações	9
Figura 2.4: Evolução da integração de SI e o BPM	12
Figura 2.5: Valor da integração de SI versus complexidade organizacional	12
Figura 2.6: Visões da Integração	13
Figura 2.7: Pontos de partida para a integração de SI	14
Figura 2.8: Estudo Delphi Group – “Questão: Dificuldade em encontrar a informação?”	15
Figura 2.9: Estudo Delphi Group – “Questão: Factores que dificultam o acesso à informação?”	16
Figura 2.10: Camadas de uma aplicação	19
Figura 2.11: Integração ao nível da camada de apresentação	19
Figura 2.12: Integração ao nível da lógica aplicacional	20
Figura 2.13: Lógica aplicacional de suporte a processos organizacionais	21
Figura 2.14: Integração ao nível do acesso e armazenamento da informação	22
Figura 2.15: Middleware de Integração	22
Figura 2.16: Integração de SI com foco no processo organizacional	24
Figura 2.17: Business Activity Monitoring na gestão e optimização de processos organizacionais	25
Figura 2.18: Web Services	27
Figura 2.19: Camadas tecnológicas	28
Figura 2.20: Camada do Software	28
Figura 2.21: Middleware para a integração	29

Figura 2.22: As 7 camadas da pilha OSI	30
Figura 2.23: Arquitectura Cliente/Servidor	31
Figura 2.24: Arquitectura two-tier	31
Figura 2.25: Arquitectura 3-tier	32
Figura 2.26: Benefícios no uso das normas	34
Figura 2.27: Normas exigidas aos fornecedores de TI	35
Figura 3.1: Sistema de Gestão de Base de Dados	41
Figura 3.2: Envio de dados em ficheiro	42
Figura 3.3: Partilha de repositório de informação	43
Figura 3.4: Customer Data Hub.	43
Figura 3.5: Replicação de informação (adaptado de Oracle Streams 10g)	44
Figura 3.6: ETL (adaptado de ETL study – Marc Songini - Computerworld)	45
Figura 3.7: Integração da informação na camada de apresentação com Middleware.47	
Figura 3.8: Enterprise Content Management(ECM)	48
Figura 3.9: Enterprise Information Integration (EII)	49
Figura 3.10: Panorama da integração da informação	50
Figura 3.11: Integração aplicacional directa	53
Figura 3.12: Modelos aplicacionais (adapitado de [Lewis 2000])	54
Figura 3.13: Interface Aplicacional	55
Figura 3.14: Message Broker	56
Figura 3.15: Comunicação point to point	57
Figura 3.16: Comunicação hub and spoke	58
Figura 3.17: Comunicação do tipo Bus	58
Figura 3.18: Plataforma Microsoft .NET	59
Figura 3.19: Plataforma J2EE	59

Figura 3.20: Funcionalidades do Oracle Application Server baseado em J2EE	62
Figura 3.21: Deployment de código num Application Server	62
Figura 3.22: Acessos diferenciados às aplicações	63
Figura 3.23: Exemplo de um documento EDIFACT	66
Figura 3.24: Caracterização das categorias de processos de Workflow	69
Figura 3.25: BPM para a integração de SI	70
Figura 3.26: Enquadramento das normas BPMN, BPML, BPEL e web services	71
Figura 3.27: Cronologia das normas de orquestração de web services	72
Figura 3.28: Diagrama de normas da WfMC e BPMI.org	73
Figura 3.29: Stack tecnológica do BPEL	74
Figura 3.30: BPEL	74
Figura 3.31: SOAP, UDDI e WSDL	76
Figura 3.32: web services stack (Gottschalk et al., 2002)	77
Figura 3.33: Arquitectura 3-tier	78
Figura 3.34: Arquitectura SOA	78
Figura 3.35: Camadas complementares do SOA	79
Figura 3.36: ESB (adaptado de Fiorano)	80
Figura 3.37: Perspectivas complementares da integração de SI	82
Figura 3.38: Perspectiva da integração inter-organizacional	84
Figura 3.39: Famílias tecnológicas e normas	87
Figura 3.40: Âmbito geral das normas e tecnologias por perspectivas	88
Figura 4.1: Principais soluções e normas por perspectivas da integração de SI	89
Figura 4.2: XML – Estruturação em árvore	91
Figura 4.3: XML – Estruturação em árvore	92
Figura 4.4: DTD	93

Figura 4.5: XSL	94
Figura 4.6: Namespaces	95
Figura 4.7: Monitor Transaccional	96
Figura 4.8: Remote Procedure Call	97
Figura 4.9: Arquitectura DCE	98
Figura 4.10: Aplicação que integra um objecto COM	99
Figura 4.11: Object Request Broker	100
Figura 4.12: Messaging	102
Figura 4.13: Comunicação point to point	103
Figura 4.14: Comunicação hub and spoke	103
Figura 4.15: Comunicação do tipo Bus	104
Figura 4.16: Electronic Data Interchange (EDI)	106
Figura 4.17: Orquestração	109
Figura 4.18: Coreografia	110
Figura 4.19: Estrutura SOAP	112
Figura 4.20: Mensagem SOAP Request	113
Figura 4.21: Mensagem SOAP Response	113
Figura 4.22: Estrutura Lógica WSDL	114
Figura 4.23: Estrutura física WSDL	115
Figura 4.24: WSDL do tipo Request-Response	115
Figura 4.25: UDDI e WSDL	116
Figura 4.26: Invocação de um Web Service	116
Figura 5.1: Integração das 3 aplicações	120
Figura 5.2: Aplicação de Requisições em 3 níveis	121
Figura 5.3: Esquema de integração da primeira abordagem	122

Figura 5.4: Definição do processo de Workflow para suportar aprovação de requisições	123
Figura 5.5: Informação contida na tabela FUNCIONARIOS da base de dados BD_RH	124
Figura 5.6: Classe Java “Valida_BD_RH”	125
Figura 5.7: Informação contida na tabela de ITEMS	126
Figura 5.8: Descrição dos pontos de integração	126
Figura 5.9: Modelo E-R do exemplo	128
Figura 5.10: Écran de introdução da requisição da aplicação de Requisições	130
Figura 5.11: Integração de código Java na aplicação de Requisições	130
Figura 5.12: Integração de informação via Database Link na aplicação de Requisições	131
Figura 5.13: Acesso via Web Browser à aplicação de Requisições com utilizador VICTOR	132
Figura 5.14: Écran de inserção e submissão das requisições	132
Figura 5.15: Informação da integração da aplicação de RH na aplicação de Requisições com Java	133
Figura 5.16: Consulta de informação da aplicação de Stocks na aplicação de Requisições	133
Figura 5.17: Submissão da requisição nº 10 do item nº1 para Workflow com sucesso.	134
Figura 5.18: Processo de aprovação de requisição nº 10 iniciado	135
Figura 5.19: Requisição de item nº 1 recusada automaticamente (Victor do departamento de Vendas)	135
Figura 5.20: Processo de requisição nº 10 recusado	136
Figura 5.21: Regra de negócio aplicada no workflow	136
Figura 5.22: Submissão da requisição nº 11 para item nº2	137
Figura 5.23: Requisição nº 11 em aprovação no workflow	137
Figura 5.24: Notificação para pedido de aprovação de requisição nº11	138

Figura 5.25: Aprovação da requisição nº11 no workflow	138
Figura 5.26: Término de processo de aprovação no workflow da requisição nº 11 ..	139
Figura 5.27: Término de processo de aprovação no workflow da requisição nº 11 .	140
Figura 5.28: Integração com Messaging	141
Figura 5.29: Envio de mensagem no Messaging	141
Figura 5.30: Recepção da mensagem na aplicação de Stocks com operação de DEQUEUE	142
Figura 5.31: Envio de email	142
Figura 5.32: Recepção de email	143
Figura 5.33: Integração com BPEL	144
Figura 5.34: Criação do Web Service para a aplicação de Requisições	145
Figura 5.35: Especificação WSDL do Web Service para a aplicação de Requisições	146
Figura 5.36: Deployment do Web Service no OC4J	146
Figura 5.37: OC4J para disponibilizar os Web Services baseado na plataforma J2EE	147
Figura 5.38: Web Service para a criação das requisições com operação “inserirRequisicao”	147
Figura 5.39: WSDL do Web Service	148
Figura 5.40: Écran de teste para a operação de “inserirRequisicao” da requisição nº33	148
Figura 5.41: Resposta do Web Service após a sua invocação	149
Figura 5.42: Query sobre tabela Requisições com registo da requisição nº33 subemtido por Web Service	149
Figura 5.43: WSDL do Web Service do sistema de workflow	150
Figura 5.44: Web Service para invocação do sistema de workflow de aprovação de requisições	151
Figura 5.45: Écran de teste para a operação de “chamaWorkflow” da requisicao nº44	151

Figura 5.46: Resposta do Web Service após a sua invocação	152
Figura 5.47: Requisição nº44 submetida por Web Service sujeita a aprovação no sistema de workflow	152
Figura 5.48: Requisição nº44 submetida por Web Service	153
Figura 5.49: Criação do projecto BPEL síncrono “REQUISICOES_BPEL” no BPEL Designer	154
Figura 5.50: BPEL Designer – Concepção gráfica e deployment do processo BPEL	155
Figura 5.51: Diagrama BPEL do processo “REQUISICOES_BPEL”	156
Figura 5.52: Associação e passagem de valores entre parâmetros do BPEL e dos Web Services	157
Figura 5.53: Alteração da resposta do Web Service da aplicação de Requisições que retorna nome de requisitante	158
Figura 5.54: Passagem do valor retornado pelo Web Service alterado no ponto anterior	158
Figura 5.55: Deployment do processo BPEL para o BPEL Process Manager	159
Figura 5.56: BPEL Console – Administração e teste dos processos de BPEL via Web.	159
Figura 5.57: Submissão de um formulário em HTML para o processo “REQUISICOES_BPEL”	160
Figura 5.58: Processo BPEL síncrono executado com sucesso para a requisição nº 100	160
Figura 5.59: Monitorização on-line via Web do processo BPEL para a requisição nº 100	161
Figura 5.60: XML na etapa “receiveInput” do BPEL com a requisição nº100	161
Figura 5.61: XML na invocação do Web Service “Inserir_Requisicao” para a requisição nº100	162
Figura 5.62: Verificação da criação da requisição nº 100 na base de dados via BPEL + Web Service	162
Figura 5.63: XML na etapa “Chama_Workflow” do BPEL com a requisição nº100	163

Figura 5.64: Requisição nº100 submetida ao sistema de workflow via BPEL + Web Service	163
Figura 5.75: Requisição nº100 submetida ao sistema de workflow via BPEL + Web Service	163
Figura 5.76: Recepção de email para a requisição nº 100	164
Figura 5.77: Requisição nº 103 em XML	165
Figura 5.78: Submissão directa de XML para o processo “REQUISICOES_BPEL”	165
Figura 5.79: Processo BPEL para a requisição nº103 submetida em XML finalizado	166

Lista de tabelas

Tabela 3.1: Soluções para os modelos aplicativos	55
Tabela 3.2: Comparação simples .NET e J2EE	61
Tabela 3.3: Categorias de processos de Workflow	68
Tabela 3.4: Interação com Web Services	76
Tabela 3.5: Descrição das áreas comuns das perspectivas	83
Tabela 3.6: Nível de integração de SI versus Perspectivas (adaptado de [Schmidt 2000])	85
Tabela 3.7: Formatos e protocolos numa arquitetura SOA	86
Tabela 4.1: Exemplo de transaction set e de message	106
Tabela 5.1: Caracterização resumida das abordagens para a implementação do exemplo.	167

1. Introdução

As Tecnologias da Informação (TI) correspondem hoje em dia a um dos principais pilares das organizações [Earl 1989]. A procura do controlo da informação e da flexibilização tornou a integração de sistemas informáticos uma das grandes prioridades organizacionais [Edwards e Newing 2000]. Por seu turno, a constante evolução das TI criou realidades tecnológicas díspares fomentando a necessidade de partilhar informação e funcionalidades entre sistemas [Houston 1998]. Para as organizações, a questão tecnológica da integração de Sistemas de Informação (SI) tornou-se cada vez mais complexa e é hoje um autêntico desafio quanto à sua flexibilidade, adaptabilidade, implementação, manutenção e gestão.

É comum encontrar organizações cujos Sistemas de Informação (SI) são compostos por diversos sistemas, processos e informação cuja interacção é suportada por canais de comunicação e ligações específicas. A crescente necessidade de integração de SI deriva de factores relacionados com a evolução das organizações, dos mercados e da tecnologia [Carvalho 2001]. Por exemplo, a necessidade de expor na Internet informação residente nos sistemas informáticos internos, a necessidade de partilhar informação entre sistemas heterogéneos ou a automatização de processos organizacionais são algumas das situações que necessitam de soluções específicas. Estas necessidades surgem normalmente no dia-a-dia das organizações devido à sua dinâmica e à constante evolução tecnológica. Esta é uma realidade complexa, onde existe uma forte competitividade ao nível das soluções, e onde as organizações facilmente não encontram as respostas mais adequadas às suas reais necessidades [Allen 2001].

Nas últimas décadas, as empresas seguiram diferentes abordagens com o objectivo de integrar sistemas, quer para partilhar informação residente em diferentes nichos, quer para aproveitar funcionalidades existentes nesses mesmos sistemas. Inicialmente, um dos problemas mais comuns era o facto das aplicações não terem sido concebidas para ser integradas com outras aplicações. Neste sentido, foram encontradas soluções de integração cujas primeiras abordagens não seguiram normas técnicas específicas dada a sua inexistência. É com a crescente necessidade de soluções nesta área, e com a própria evolução tecnológica, que estas especificações foram criadas e aperfeiçoadas

até aos nossos dias. Estas normas vieram facilitar a concepção de soluções, mas há hoje uma realidade tecnológica repleta de alternativas viáveis. A maioria das soluções existentes baseiam-se em normas técnicas o que facilita a sua classificação. No entanto, cada norma tem a sua especificidade e pode, ou não, ser adoptada de acordo com a sua adequação, utilidade e enquadramento nas diversas arquitecturas tecnológicas. Por força da evolução do mercado tecnológico, certas normas sobrepoem-se em algumas áreas, ou são incompatíveis, o que aumenta a dificuldade no entendimento e na escolha da solução mais adequada. Elas são no entanto incontornáveis para se obter soluções certificadas e abertas.

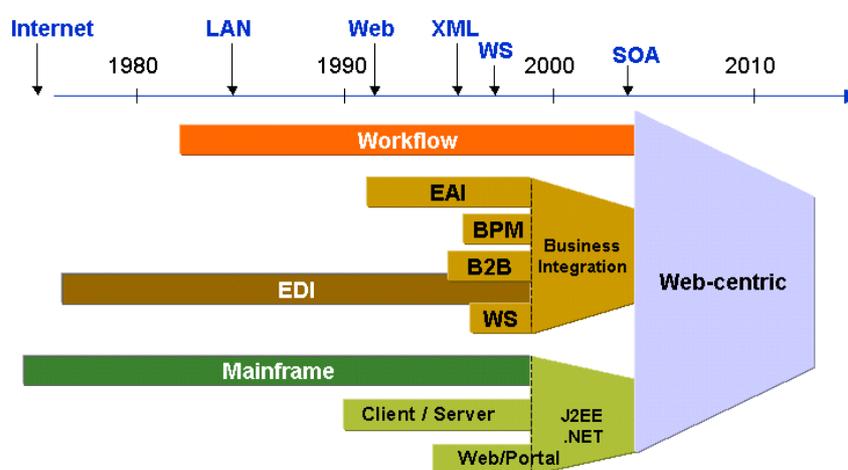


Figura 1.1: Evolução e convergência tecnológica (adaptado de [Dubray 2003])

A integração de SI está normalmente associada aos termos *Enterprise Application Integration* (EAI) [William 2000, Serain 2002] ou *Business Process Management* (BPM) [Hollingsworth 2004] que têm pontos em comum e que são por vezes complementares. Estas são duas correntes influenciadoras que determinaram diferentes visões e soluções para a integração de SI. Por seu turno, o mercado das tecnologias mudou significativamente nos últimos anos dada a crescente competitividade entre as empresas e os mercados das TI. O âmbito da integração de SI deixou de estar exclusivamente na estrutura das aplicações informáticas para se alargar a toda a organização. Para além disso, assistiu-se nos últimos anos a uma convergência e sobreposição de tecnologias que tornou a sua classificação mais difícil (Figura 1.1). Neste sentido, o recente surgimento dos *Web Services* (WS) [Cerami 2002] e da arquitectura *Service Oriented Architecture* (SOA) [Newcomer e Lomow 2004] criou novas alternativas às abordagens mais tradicionais. Neste campo, surgem

novas respostas, que são por vezes complementares às soluções já existentes, e que permitem igualmente suportar este tipo de integração de SI. Hoje em dia, para a área da integração de SI, é possível encontrar diferentes soluções para um mesmo problema, sendo todas elas válidas [Wrazen 2000]. Nestes casos, são as próprias necessidades e os objectivos das organizações que vão determinar a sua real adequação.

Todos estes factores tornaram as abordagens para a integração de SI difíceis de escolher. Neste âmbito, qualquer situação requer uma análise exaustiva, quer ao nível das necessidades da organização, quer ao nível das soluções a adoptar. Para além disso, existe nesta área uma enorme quantidade de informação com diferentes perspectivas e definições, por vezes ambíguas ou desactualizadas, o que dificulta a sua compreensão. Este trabalho pretende contribuir para um melhor entendimento desta complexidade fazendo uma descrição e catalogação das possíveis abordagens e soluções mais adequadas às necessidades existentes nas organizações. Neste trabalho, é feito um enquadramento do tema da integração de SI, uma classificação das abordagens mais comuns, um levantamento das principais normas técnicas relacionadas com cada abordagem e a implementação de um exemplo prático de integração de SI, realçando e comparando duas abordagens.

1.1 Objectivos da dissertação

O principal objectivo deste trabalho é a clarificação e a catalogação das principais perspectivas e áreas de intervenção na integração de SI. Para este efeito foram identificadas diferentes perspectivas que foram por sua vez relacionadas com as principais normas e tecnologias existentes. Para conseguir estes propósitos foi realizado um conjunto de actividades incluindo uma revisão de literatura, uma pesquisa de conceitos, tecnologias e normas e uma implementação prática de um exemplo de integração de SI. Este último ponto procura ilustrar a utilização de tecnologias e normas de acordo com duas abordagens diferentes. Realça-se com este exemplo a importância e utilidade das normas mais recentes como são os *Web Services* e o *BPEL*. Como fruto desta pesquisa obteve-se um conjunto significativo de informação e de figuras maioritariamente em inglês. Algumas destas figuras mais

complexas foram directamente incluídas neste trabalho, pelo receio de não conseguir recriá-las fidedignamente em português.

1.2 Conceitos e pressupostos

As TI têm um papel preponderante para a integração de SI e permitem encarar esta questão de várias formas. Dada a complexidade das TI, e a diversidade de visões na área dos SI, é imperativo definir alguns conceitos e pressupostos sob pena de se perder as ideias orientadoras aqui presentes. É necessário explicar o entendimento dos temas referidos ao longo desta dissertação, quer na área dos SI quer das TI, correndo deliberadamente o risco de simplificar algumas definições. Cada conceito, ou enquadramento, definido poderá ter outras interpretações igualmente válidas, mas servem neste caso de linha orientadora para esta explanação.

O tema da integração de SI refere-se à problemática de integrar sistemas informáticos díspares de forma a poder partilhar os seus recursos, sejam eles dados sejam elas funcionalidades. A natureza de cada sistema vai ditar a abordagem de integração mais adequada. O conceito de *integração* corresponde, neste caso, ao acto de integrar dois ou mais sistemas que podem estar dentro ou fora de uma organização. A integração pode ser feita em vários níveis, como por exemplo, entre diferentes repositórios de informação, dentro de uma mesma aplicação informática e entre duas aplicações. O tema da integração de SI contempla a interoperabilidade entre sistemas no sentido em que são colocados a comunicar e a interagir, independentemente da abordagem tecnológica de suporte.

Considera-se um sistema informático uma solução informatizada, ou um *software*, que permite a gestão e a exploração de informação dentro de uma organização. Um sistema informático pode ser também referido neste trabalho como um *sistema*, um *aplicativo*, uma *aplicação*, um *pacote aplicacional*, uma *solução informática*, um *módulo aplicacional* ou uma *solução aplicacional*.

Uma *arquitectura* corresponde a um padrão de combinação de componentes que está na base de uma solução tecnológica com um determinado fim. Este conceito

identifica que uma determinada solução técnica, ou uma abordagem de integração, tem várias camadas estruturantes e complementares. Essas camadas podem ser lógicas ou físicas e dão corpo a uma ideia ou permitem descrever uma tecnologia. Por exemplo, a arquitectura do tipo “cliente/servidor”, abordada neste trabalho, define duas camadas complementares.

O conceito de *plataforma* referencia uma base tecnológica que suporta e disponibiliza funcionalidades para a concepção de soluções aplicacionais ou para a integração de SI. Por exemplo, a plataforma *Java 2 Platform Enterprise Edition* (J2EE) é uma plataforma de desenvolvimento aplicacional com um conjunto de interfaces programáticas estruturadas em diferentes camadas complementares.

O conceito de *processo* é aqui entendido como um processo organizacional composto por uma sequência de tarefas que podem envolver pessoas, aplicações e/ou documentos. Neste âmbito, o *processo* também pode representar uma base estruturante para a integração de SI, quando este é informatizado e automatizado, permitindo a interligação de várias aplicações informáticas.

O conceito de *Middleware* corresponde a uma arquitectura tecnológica que concentra num *software* centralizado um conjunto de informação, procedimentos e aplicações. Este permite interligar sistemas, disponibilizar informação, transportar dados entre sistemas e fornecer canais de comunicação entre diversas aplicações informáticas.

1.3 Organização dos capítulos

A estrutura deste documento está baseada em 6 partes:

1. Enquadramento do tema e dos conceitos associados que servem de base para esta explanação.
2. Identificação de diferentes perspectivas para a integração de SI, suas condicionantes, tecnologias de base e o papel das normas técnicas.

3. Detalhe e relacionamento das perspectivas, arquiteturas e normas mais importantes.
4. Descrição das principais normas e tecnologias identificadas em cada perspectiva descrita.
5. Apresentação detalhada da implementação de um exemplo de integração de SI utilizando tecnologias e normas apresentadas neste trabalho.
6. Conclusões e sugestões.

2. A integração de Sistemas de Informação

A integração de Sistemas de Informação (SI) [Laudon 1996] é uma área complexa que pode ser vista por diversas perspectivas. De acordo com o tipo de organização [Mintzberg 1979] ou de tecnologia podem ser adoptadas várias interpretações para o seu enquadramento. Entende-se por integração de SI a partilha de informação e processos entre aplicações em rede ou fontes de dados numa organização. É possível encontrar diferentes definições válidas consoante a tecnologia ou os objectivos das soluções adoptadas. Nesta área existem várias abordagens que são adoptadas consoante as características e as necessidades de cada organização. Por seu turno, cada fornecedor tecnológico define a integração de SI consoante a sua visão das organizações, dos seus processos estruturantes e das soluções que fornecem.

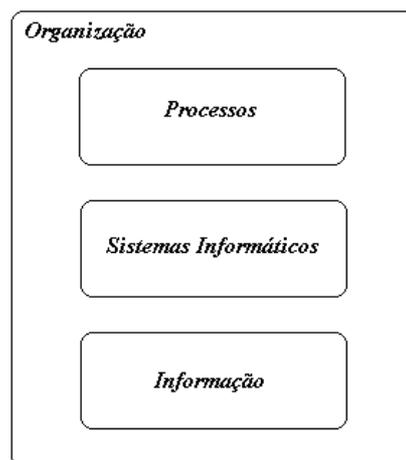


Figura 2.1: Principais áreas organizacionais para a integração de SI.

A informação, os sistemas informáticos e os processos organizacionais são três áreas complementares que devem ser coordenadas e ajustadas sob pena de não se adequarem à dinâmica da organização (Figura 2.1). Neste âmbito os processos correspondem à combinação das actividades necessárias e suficientes para o funcionamento da organização. Qualquer estratégia para a integração de SI tem de ter em conta cada uma destas realidades, quer em termos de análise e estratégia quer em termos de implementação, para obter a solução mais adequada.

2.1 A necessidade de integração de SI

A integração de SI permite às organizações que a tecnologia suporte eficazmente a sua lógica funcional e que estas fiquem melhor preparadas para responder às constantes exigências e mudanças do seu meio ambiente. Cada organização tem diferentes necessidades de integração de SI que dependem do seu tipo de actividade e da realidade tecnológica existente. Em qualquer caso, diferentes abordagens podem ser válidas e complementar-se na definição de uma arquitectura de integração mais abrangente.

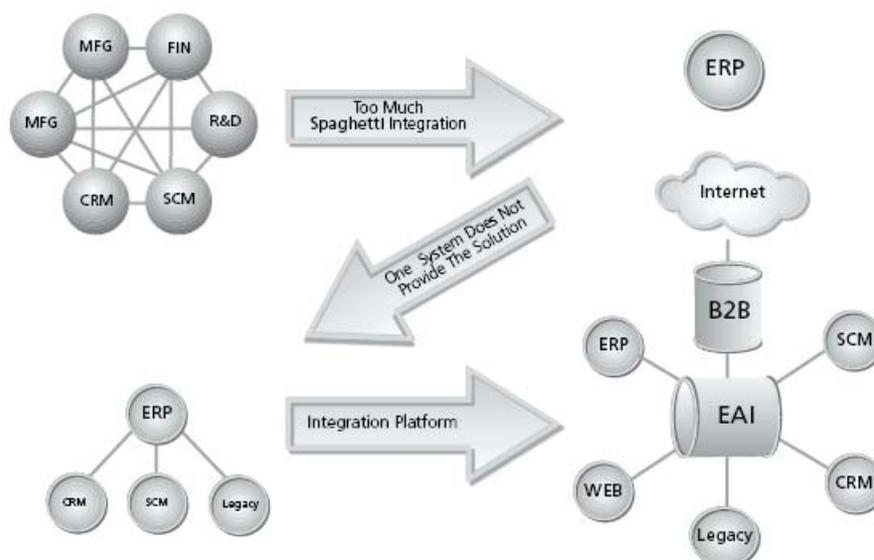


Figura 2.2: Evolução da integração de SI (adaptado de [Magic 2003])

As soluções para a integração de SI determinaram várias fases na sua evolução (Figura 2.2) que passaram pela criação de teias de sistemas interligados, seguida pela adopção de pacotes de software integrado, como são os *Enterprise Resource Planning* (ERP) [Summer 2004], que entretanto foram complementados com a integração de outros sistemas do tipo Client Relationship Management (CRM) [Reynolds 2002], Supply Chain Management (SCM) [Chopra e Meindl 2000] e outras aplicações já existentes na organização [Cherry 2000]. Este tipo de soluções garantiram uma gestão mais eficiente da informação e dos processos das respectivas organizações. Com a crescente necessidade de integração de SI, surgiram soluções específicas, e normalmente denominadas por *Enterprise Application Integration* (EAI), que

permitem suportar e centralizar várias formas de integração de aplicações informáticas.

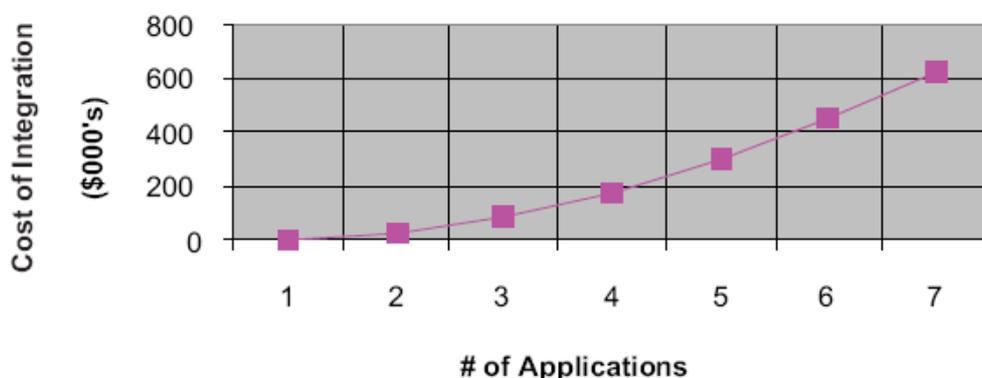


Figura 2.3: Custo da integração de SI versus nº de aplicações (adaptado de [Acharya 2003])

Com o aparecimento da Internet a necessidade de interligar sistemas tornou-se cada vez mais premente. Neste sentido, os responsáveis pelas TI começaram a concluir que era necessário adoptar uma estratégia global [Ulrich 2002] para a integração de todos os seus SI e escolher soluções específicas para o efeito. A empresa consultora Gartner Group demonstrou recentemente que mais de 35% dos investimentos feitos em TI são aplicados na integração de sistemas. Os custos para a integração de SI são normalmente proporcionais à quantidade de sistemas que se quer interligar e ao nível de integração desejado (Figura 2.3). Hoje em dia existem soluções completas que permitem responder a diferentes problemas e seguir diferentes abordagens. Estas soluções específicas são normalmente denominadas de *Integration Middleware* [Serain 2002], ou *Integration Hubs*, e têm um conjunto de funcionalidades que permitem integrar diferentes níveis tecnológicos abordados neste trabalho.

Com a evolução dos mercados e das tecnologias, foram surgindo novas formas de abordar a integração de SI. Existem hoje em dia correntes tecnológicas que defendem diferentes perspectivas e abordagens para a integração de SI. Cada uma delas tem soluções vocacionadas para a sua área chegando por vezes a partilhar funcionalidades das restantes. Esta realidade dificulta a escolha das soluções mais adequadas às necessidades organizacionais e complica o entendimento das tecnologias existentes nesta área.

2.2 As formas e perspectivas de integração de SI

Com a evolução das tecnologias e a constante adaptação das organizações aos novos desafios dos seus meios ambientes surgem diferentes necessidades de integração. Cada uma delas tem diferentes formas de intervenção e tipos de integração. Embora muito centrado na integração aplicacional, é comum designar o *Enterprise Application Integration* (EAI) como a família de referência para as soluções de integração de SI. Por seu turno, os defensores do *Business Process Management* (BPM) alegam que a sua família tecnológica engloba todas as soluções para a integração de SI.

As formas de integração de SI podem ser catalogadas de várias maneiras consoante diferentes critérios. Por exemplo, é possível catalogar genérica e tecnologicamente a integração de SI consoante o seu nível de implementação:

- **Sistemas Integrados de Gestão:** Sistemas aplicacionais fechados e compostos por módulos internos totalmente integrados e autónomos.
- **Informação centralizada:** Diferentes aplicações acedem a repositórios de informação centralizados partilhando o seu conteúdo. Normalmente nestes casos existe um “metamodelo” de dados comum.
- **Aplicações compostas:** Aplicações que estão integradas através das suas interfaces de programação (*Application Programming Interface*) [Hines 1996]. Estas invocam, ou incorporam, entre si funções, métodos ou procedimentos partilhando lógica aplicacional de forma directa.
- **Sistemas Transaccionais:** Sistemas que coordenam entre si as suas transacções operacionais garantindo a actualização com sucesso e sincronizada da informação em cada sistema interligado.
- **Sistemas Distribuídos:** Sistemas autónomos integrados através de serviços aplicacionais que disponibilizam partes de lógica aplicacional. Os serviços aplicacionais estão identificados e catalogados num repositório específico e a sua invocação é feita dinamicamente.

Esta catalogação está centrada nas abordagens técnicas para a integração de aplicações e permite ter uma primeira visão de algumas alternativas subjacentes. Tendo em conta a perspectiva dos processos e da própria organização pode-se por exemplo catalogar da seguinte forma a integração de SI [Hohpe et al. 2003]:

- **Information Portals:** Integração de aplicações ao nível da interface gráfica e da camada de apresentação.
- **Data Replication:** Integração ao nível da informação, actualizando e sincronizando os diferentes repositórios existentes na organização. A informação não está centralizada mas distribuída.
- **Shared Business Functions:** Lógica aplicacional partilhada entre aplicações informáticas que invocam, ou incorporam, procedimentos existentes na camada aplicacional dessas aplicações.
- **Service-Oriented Architectures:** Arquitectura distribuída onde cada sistema aplicacional disponibiliza funções ou procedimentos como serviços disponíveis para serem identificados e chamados.
- **Distributed Business Processes:** A integração aplicacional é feita numa lógica processual. Os processos organizacionais são automatizados e cujas partes são executadas em diferentes aplicações. Os processos representam o eixo da integração de SI.
- **Business-to-Business Integration:** A integração de SI está aqui centrada na organização. O objectivo é permitir a colaboração entre diferentes organizações de forma a partilhar processos e informação de forma automática.

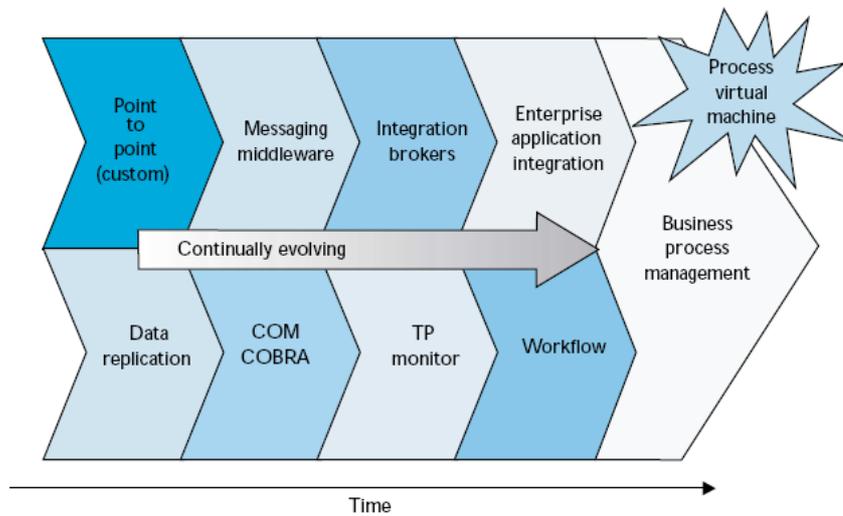


Figura 2.4: Evolução da integração de SI e o BPM (adaptado de [CSC 2002])

Para além das diferentes perspectivas e catalogações existentes, a evolução das TI na área da integração tem revelado que os processos organizacionais são cada vez mais o centro das atenções. Isto deve-se à importância que os processos têm numa organização pela sua natureza estruturante [Alves 1995]. As soluções de BPM mais recentes incluem praticamente todas as tecnologias e conceitos de integração de SI que foram aparecendo ao longo dos tempos (Figura 2.4).

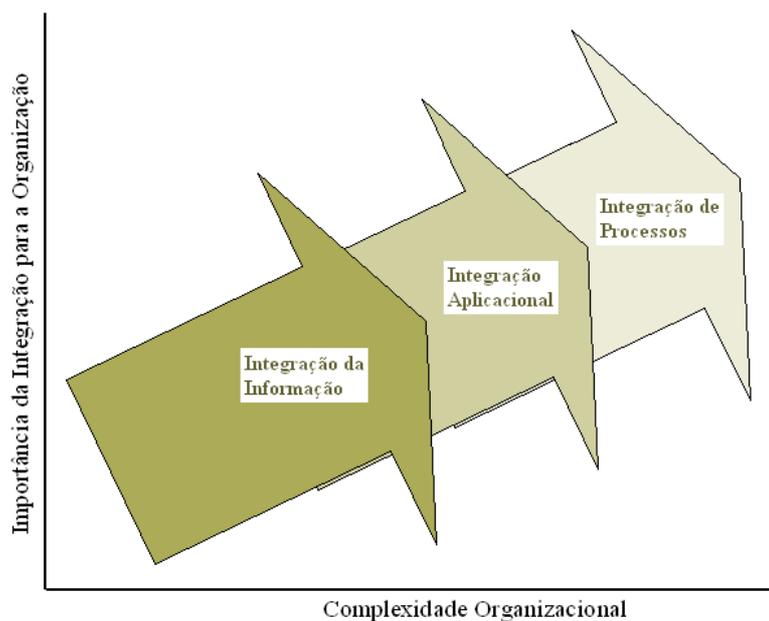


Figura 2.5: Valor da integração de SI *versus* complexidade organizacional

Quanto mais complexa é uma organização mais valor terão os projectos de integração de SI. Neste âmbito, pode-se identificar pelo menos três áreas complementares e com crescente nível de importância para as organizações (Figura 2.5): a integração da informação, a integração aplicacional e a integração de processos.

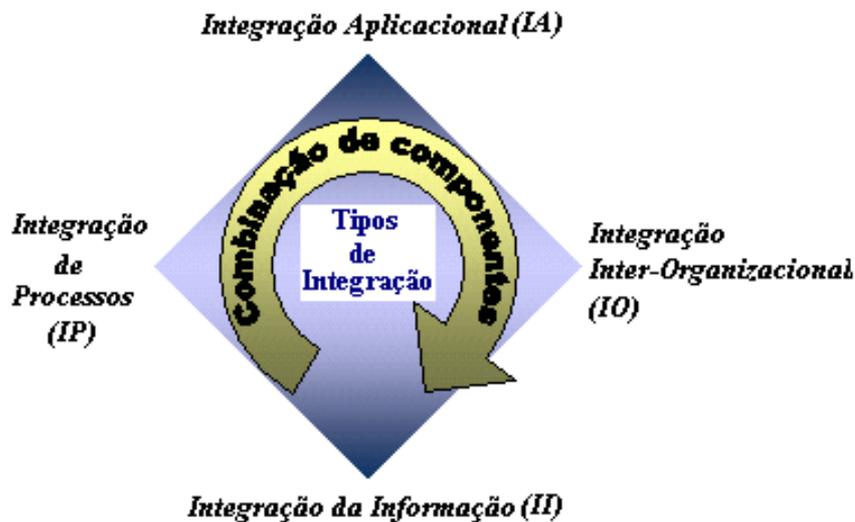


Figura 2.6: Visões da Integração (adaptado de [Integr8 2005])

Existem outras possíveis classificações determinadas pelos níveis técnicos ou conceptuais diferenciados. De qualquer forma o objectivo principal da integração de SI é sempre a obtenção de sistemas que facilitam o acesso a dados e procedimentos sem qualquer barreira funcional. Em consequência, as aplicações resultantes podem corresponder a combinações de componentes de diferentes áreas tecnológicas. Neste âmbito, identificam-se as quatro perspectivas tecnológicas mais comuns e abrangentes (Figura 2.6):

- **Integração da Informação (II):** o centro das atenções nesta perspectiva é a informação, a sua gestão e disponibilização.
- **Integração Aplicacional (IA):** as aplicações são aqui o principal alvo sendo a sua integração o objectivo primordial.

- **Integração de Processos (IP):** os processos organizacionais são o centro das atenções nesta perspectiva, sendo a integração de SI feita numa lógica processual.
- **Integração Inter-Organizacional (IO):** a informação e a sua forma de intercâmbio entre organizações são aqui o alvo desta perspectiva.

Cada perspectiva permite planear e iniciar os projectos para a integração de SI em pontos de partida diferentes (Figura 2.7). É possível começar pela integração ao nível da informação ou dos procedimentos, e optar por uma abordagem *bottom-up*, ou pela situação inversa que centra a sua atenção nos processos das organizações e segue uma abordagem *top-down*.

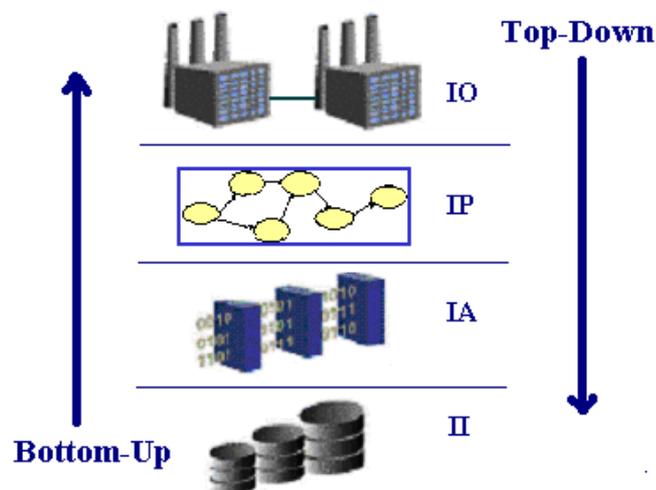


Figura 2.7: Pontos de partida para a integração de SI

Em qualquer dos casos procura-se a integração entre sistemas díspares que precisam de partilhar informação e funcionalidades. Com a evolução das TI, surgiram recentemente normas técnicas como os *Web Services* [Cerami 2002], e arquitecturas como o *Service Oriented Architecture (SOA)* [Newcomer e Lomow 2004], que permitem encapsular funcionalidades e disponibilizar novas camadas de acesso. Os *Web Services* podem ser utilizados em qualquer uma das perspectivas para a integração de SI aqui identificadas. Cada abordagem tem vantagens e desvantagens que devem ser ponderadas sob pena de condicionar as capacidades de gestão e

adaptação das organizações [Levine 2005]. Os pontos seguintes deste capítulo descrevem sumariamente o âmbito de cada uma das quatro principais perspectivas identificadas.

2.2.1 Integração da informação

Ao longo dos tempos, as organizações adoptaram diferentes estratégias para a gestão da informação. Estas foram ganhando consciência que a informação é um dos seus principais recursos [Earl 1989]. A sua consolidação pode ser feita de várias formas e com diferentes objectivos. Não se trata simplesmente de interligar fontes de informação, mas sim de aceder aos repositórios certos, nos momentos certos, e cujo conteúdo seja fidedigno e estruturado [Bertino 1991]. Para isso, existem várias abordagens que proporcionam formas diferenciadas de disponibilizar e apresentar a informação. Todos os esforços de integração nesta área implicam a procura de consistência e de qualidade da informação independentemente das aplicações [Trauth 1989].

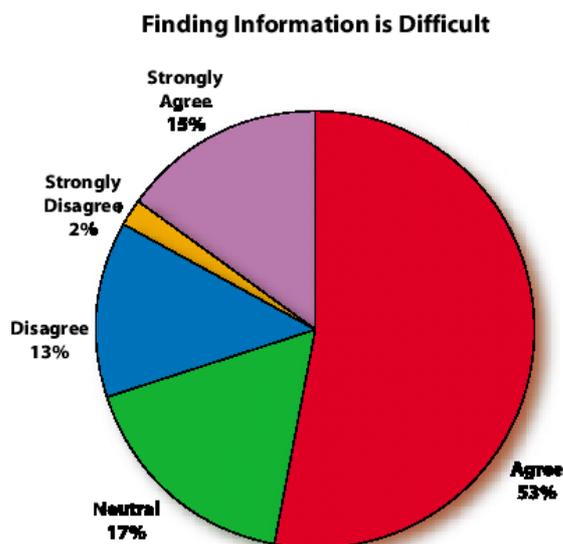


Figura 2.8: Estudo do Delphi Group – “Questão: Dificuldade em encontrar a informação?”

Um estudo do Delphi Group [Delphi 2002] relacionado com os problemas da informação excessiva ou não estruturada nas empresas, revela que 68% dos inquiridos concordam que a informação é difícil de encontrar (Figura 2.8), que 35% não conseguem encontrar a informação válida pelo facto de estar em constante alteração

(Figura 2.9) e que 11% não encontra informação porque simplesmente esta não está disponível. Embora o estudo esteja centrado na informação não estruturada, este revela que a informação é vital para a dinâmica das organizações e que é fundamental tê-la acessível e válida para as pessoas que dela necessitam. A integração de SI ao nível da informação tem precisamente o mesmo objectivo.

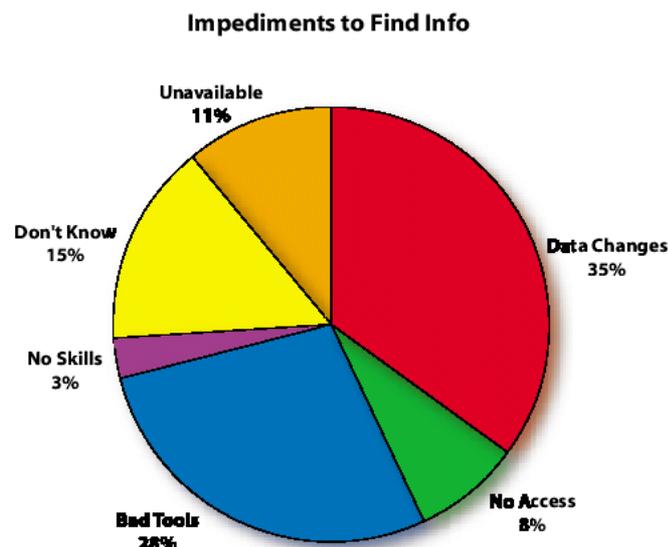


Figura 2.9: Estudo do Delphi Group – “Questão: Factores que dificultam o acesso à informação?”

Uma gestão eficiente da informação torna as organizações mais flexíveis para se adaptarem às mudanças do seu meio circundante, e com isso serem mais competitivas. A forma como elas processam e armazenam este recurso cria necessidades de integração a vários níveis e com diferentes tipos de problemas. Neste âmbito existem soluções que permitem organizar, consolidar e gerir a informação com diferentes propósitos e que correspondem a 2 áreas de intervenção:

- Actualização diária da informação operacional com recurso a soluções transaccionais do tipo *On-Line Transaction Processing* (OLTP) [Claybrook 1992].
- Consolidação dos repositórios de informação para fins de análise e gestão empresarial. Esta é a área dos *Datawarehouses*, dos EIS (*Executive Information Systems*), das soluções multi-dimensionais *Multidimensional On-Line Analytical Processing* (MOLAP)

[Arkhipenkov e Golubev 2001] ou do *DataMining* [Westphal e Blaxton 1998].

As necessidades de integração de informação estão normalmente mais associadas às soluções do tipo OLTP e à natureza operacional do conteúdo dos seus repositórios. Neste âmbito distinguem-se duas situações problemáticas:

- Dispersão da informação na organização.
- Heterogeneidade da informação.

A dispersão da informação engloba situações onde se verifica que esta não está centralizada, conduzindo por vezes à existência de “ilhas de informação”. Neste caso, a integração de duas ou mais aplicações, que precisam de partilhar informação, pode revelar-se complexa dado que esta pode estar repetida e desactualizada. No segundo caso, a heterogeneidade da informação dificulta a interligação entre módulos aplicativos autónomos [Papakonstantinou et al.1995]. Aqui, cada aplicação estrutura a mesma informação de forma diferente o que obriga à sua interpretação, ou tradução, para conseguir uma integração efectiva [Ullman 1997]. Esta situação tem a ver com a modelação lógica da informação, que pode criar estruturas de armazenamento diferentes dificultando a sua partilha.

A integração de SI orientada à informação tem pelo menos três áreas de intervenção possíveis:

1. O armazenamento da informação: permite a integração de SI através da centralização do seu armazenamento.
2. O transporte e troca de informação: permite uma integração de SI ao nível do controlo do seu transporte e da sua troca entre sistemas.
3. A disponibilização e apresentação da informação: permite uma integração de SI com a sua apresentação centralizada num determinado contexto como por exemplo uma página de uma intranet, ou um relatório que agrega várias fontes de informação.

Neste âmbito, a integração de SI orientada à informação pode ser realizada em diferentes áreas que pode passar pela utilização de sistemas de gestão de bases de dados, pela troca de informação entre aplicações, pela criação de repositórios centralizados de informação para efeitos analíticos ou pela apresentação centralizada de informação dispersa numa intranet e acessível com um *web browser*. Os principais benefícios associados a esta abordagem centrada na informação são essencialmente:

- Partilha da informação entre sistemas.
- Consolidação a informação na organização.
- Os repositórios de informação podem manter-se distribuídos e heterogéneos.
- Controlo da integridade e correcção da informação.
- Acesso controlado e centralizado à informação.
- Relatórios de gestão fiáveis, e correspondente visão da organização fidedigna.

As principais dificuldades para a integração da informação estão directamente relacionadas com os sistemas de armazenamento, as formas de acesso, os tipos de comunicação e a segurança.

Independentemente da solução tecnológica utilizada, a integração de SI centrada na informação tem como principal preocupação garantir que esta esteja sempre disponível, correcta e válida para as aplicações ou pessoas que dela necessitam.

2.2.2 Integração aplicacional

As primeiras aplicações informáticas começaram por ser monolíticas, autónomas e fechadas o que traz problemas de integração [William 2000]. Hoje em dia, estas deixaram de estar isoladas e, neste sentido, um utilizador aplicacional espera obter toda a informação, ou efectuar todas as suas operações num mesmo écran e de forma rápida. A integração aplicacional centra-se nas soluções que permitem interligar, ou incorporar, aplicações partilhando informação e funcionalidades [Zahavi 1999]. Esta abordagem é comumente denominada por *Application-to-Application* (A2A) ou também por *Composite Applications* [Woods 2003].

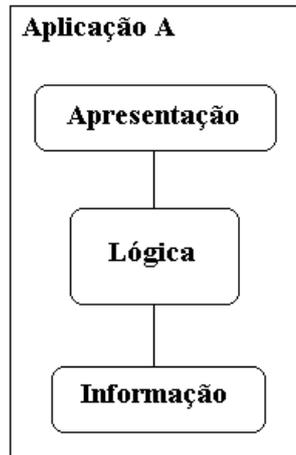


Figura 2.10: Camadas de uma aplicação (adaptado de [Wong 2001])

O principal alvo de intervenção desta área é a própria aplicação. Ao nível da integração directa de aplicações encontram-se diferentes soluções consoante o nível de ligação necessário. Uma aplicação pode conter funcionalidades correspondentes a diferentes estratos tecnológicos complementares. Neste sentido, é possível descrever uma aplicação pelas suas três principais camadas estruturantes (Figura 2.10). Estas correspondem à camada de armazenamento da informação, à camada da lógica aplicacional e à camada de apresentação [Wong 2001]. A camada da lógica funcional de uma aplicação faz a gestão das regras, requisitos e funcionalidades desse sistema, podendo inclusivamente automatizar alguns processos da organização.

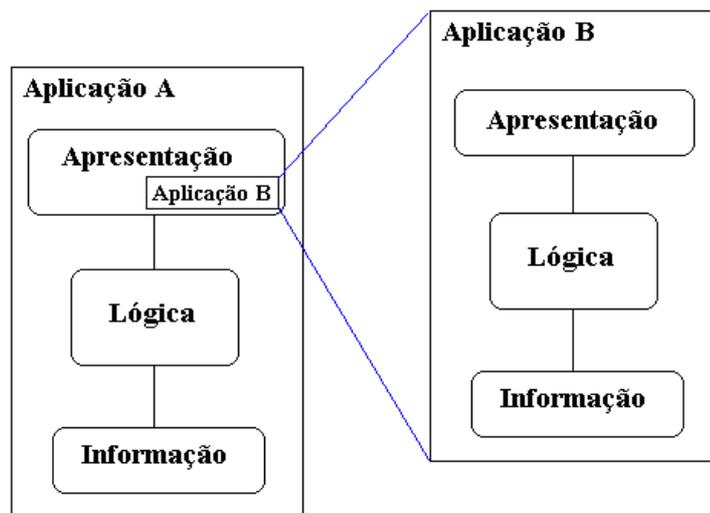


Figura 2.11: Integração ao nível da camada de apresentação (adaptado de [Wong 2001])

De acordo com estas camadas, e centrado o âmbito nas aplicações informáticas, pode-se identificar três grupos de soluções:

- Integração da camada de apresentação
- Integração da lógica aplicacional
- Integração da informação

O principal objectivo da integração da camada de apresentação é a centralização do acesso à informação e às aplicações numa única interface (Figura 2.11). Neste caso, a integração é feita ao nível do que as pessoas visualizam correspondendo a uma nova camada de apresentação que integra toda a informação e aplicações num único écran. Geralmente a ferramenta mais utilizada para aceder a essa nova camada é o *web browser*. Este tipo de integração pode ser feita de forma simples e directamente ao nível do écran ou de forma mais complexa criando uma camada específica de interacção para apresentar somente a informação e as funcionalidades necessárias.

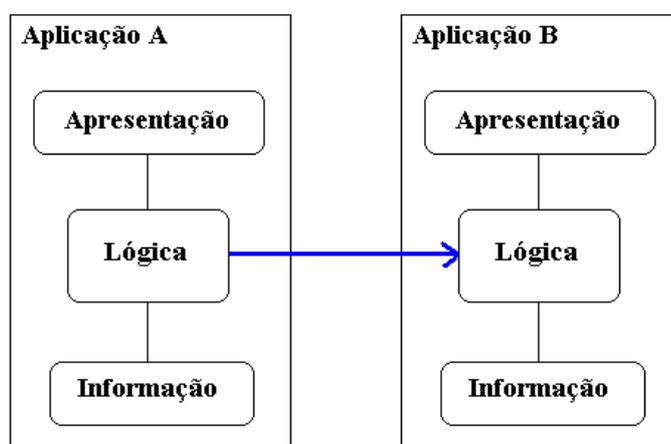


Figura 2.12: Integração ao nível da lógica aplicacional (adaptado de [Wong 2001])

A integração ao nível da lógica aplicacional corresponde à integração directa entre duas ou mais aplicações. Nesta área procura-se interligar aplicações para partilhar funcionalidades e complementar características (Figura 2.12). Existem tecnologias que permitem este tipo de integração e que variam de uma simples invocação de código, à inclusão directa de partes de uma aplicação ou à troca de mensagens estruturadas entre aplicações. Este tipo de integração requer canais de comunicação específicos que suportam os diferentes tipos de interacção. Esta abordagem enquadra

também a integração da lógica aplicacional que implementa regras para a automatização de processos organizacionais. Dentro de uma aplicação, é possível criar funcionalidades com esse propósito, suportando assim os processos, a respectiva sequência de actividades e as regras da própria organização (Figura 2.13). A grande maioria da informação, processos e regras da organização é gerida neste nível aplicacional.

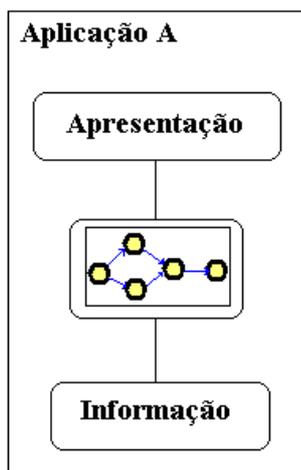


Figura 2.13: Lógica aplicacional de suporte a processos organizacionais (adaptado de [Wong 2001])

A integração de SI centrada na camada da informação das aplicações procura partilhar informação entre cada sistema (Figura 2.14). O principal objectivo é a troca de informação entre sistemas de forma a que cada um tenha acesso à mesma informação e com o mesmo significado. Para isso existem várias tecnologias que garantem a correcção, a integridade, a segurança, a veracidade e a disponibilidade da informação. Um dos problemas mais comuns nesta área tem a ver com a gestão das transacções aplicacionais que determinam a actualização e integridade da informação. Quando esta gestão é feita ao nível da camada da lógica aplicacional pode gerar conflitos no acesso e na partilha da mesma informação em simultâneo. Para isso os SGBD têm funcionalidades específicas ou então existem soluções transaccionais próprias para estas situações. Esta abordagem é comum à perspectiva da integração de SI centrada na informação, no que diz respeito ao seu armazenamento e à sua troca.

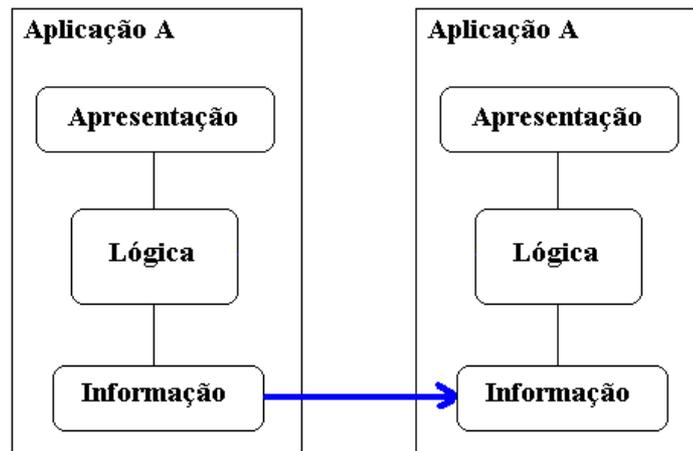


Figura 2.14: Integração ao nível do acesso e armazenamento da informação (adaptado de [Wong 2001])

Ainda neste âmbito, existem soluções que abrangem de forma combinada e centralizada todos os tipos de integração aqui referidos. É na área do *Middleware* [Watt 2002] que se encontram estas alternativas que suportam a integração de sistemas dentro e fora das organizações (Figura 2.15).

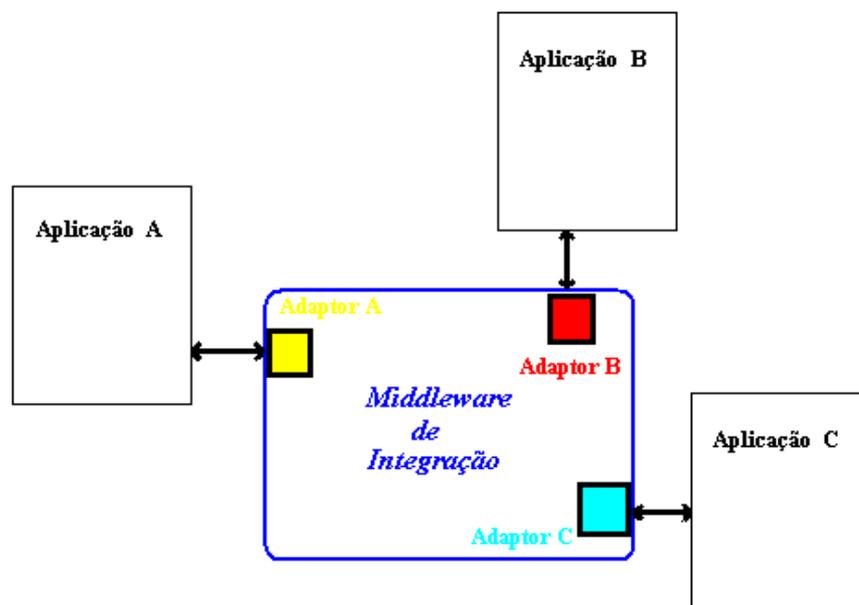


Figura 2.15: *Middleware* de Integração

As soluções de *Middleware* permitem normalmente os três tipos de integração aqui referidos ao nível da estrutura das aplicações. Os principais benefícios associados a esta abordagem de integração de SI orientada às aplicações são essencialmente:

- Tirar melhor partido das funcionalidades dos sistemas existentes
- Reutilização das regras da organização já implementadas nos sistemas
- Tornar os actuais sistemas como fornecedores de serviços para outros sistemas
- Melhora o acesso directo e indirecto às aplicações distribuídas
- Reaproveita sistemas antigos que possam ainda dar o seu contributo
- Facilita futuras integrações de sistemas entre organizações

2.2.3 Integração de processos

Não se tratando directamente de uma camada tecnológica, a automatização dos processos organizacionais representa uma das vertentes da integração de SI. Estes processos correspondem ao modo de funcionamento das organizações, e definem a forma como a informação é tratada e veiculada. A grande valia desta abordagem é o facto de haver uma ligação directa entre os processos da organização e a tecnologia que suporta a sua implementação [Watt 2002]. As organizações modernas tendem a uniformizar e automatizar muitos dos seus processos com o objectivo de tornar mais simples e eficaz o seu controlo [Miers 2005]. A crescente procura de qualidade e excelência pelas organizações levam-nas a certificar-se de acordo com normas de qualidade que se centram na optimização e documentação dos processos organizacionais.

A optimização dos processos organizacionais e o controlo do fluxo de informação é fundamental para a integração de sistemas. Pode-se identificar três tipos diferentes de processos organizacionais [Delphi 2002]:

1. Processo *System-to-System*
2. Processo *Person-to-Person*
3. Processo *Person-to-System*

O primeiro tipo engloba os processos que envolvem transferência de informação entre um conjunto de aplicações que podem incluir várias etapas sequenciais. No segundo caso, encontram-se processos que envolvem pessoas que colaboram para

completar tarefas ou processos da organização. O último tipo caracteriza os casos que envolvem pessoas que dão início a processos organizacionais ou participam neles num determinado ponto.

Neste âmbito, o processo organizacional pode ser o foco para uma estratégia de integração de SI. As soluções informáticas do tipo *Workflow* [Georgakopoulos 1995, Fischer 2003] ou da área do *Business Process Management* (BPM) [Hollingsworth 2004] permitem automatizar este tipo de processos e integrar diferentes aplicações (Figura 2.16). A lógica de integração tem a ver com a sequência de tarefas associadas a um processo que determina a interação ordenada com as diversas aplicações. Esta visão da integração de SI permite definir uma arquitectura orientada aos processos (*Process Oriented Architecture* - POA) [CSC 2003]. É comum as soluções do tipo *Integration Middleware* fornecerem também estas funcionalidades.

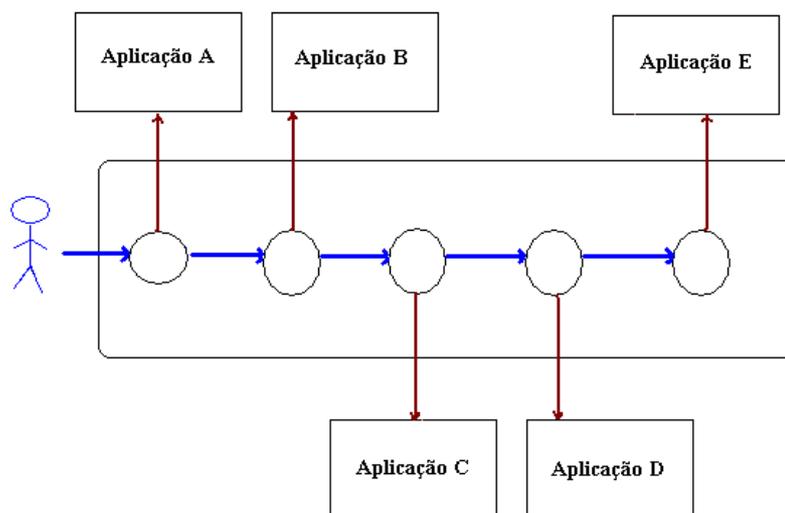


Figura 2.16: Integração de SI com foco no processo organizacional

Em qualquer abordagem para a integração de SI orientada aos processos a monitorização da execução dos processos é fundamental. Esta área é crucial quer para obter informação de gestão quer para verificar a real adequação da tecnologia. Neste campo surge o conceito de BAM (*Business Activity Monitoring*) [McKie 2003, Kochar 2005] no qual se encontra um conjunto de soluções que permitem este tipo de funcionalidades. O BAM (Figura 2.17) disponibiliza informação detalhada acerca de cada passo existente em cada processo e permite diagnosticar em cada momento o

estado de cada processo na organização. Estas soluções têm normalmente 3 camadas complementares:

- *Business Process Optimization*: Baseado na informação obtida da camada inferior é possível otimizar os processos da organização.
- *Business Activity Monitoring*: Informação relativa ao desempenho de cada actividade existente na organização.
- *Business Process Management*: Automatização e gestão dos processos organizacionais.

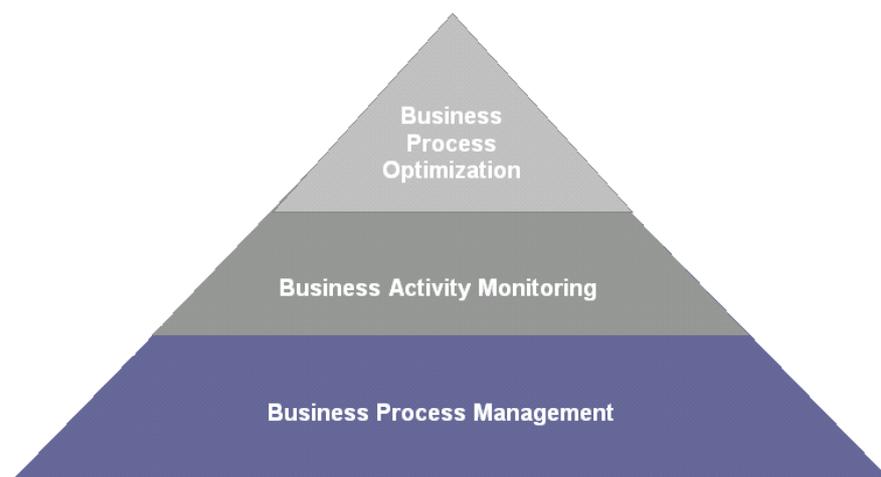


Figura 2.17: *Business Activity Monitoring* na gestão e otimização de processos organizacionais

Os principais benefícios associados a esta abordagem são essencialmente:

- Automatização e execução dos processos organizacionais
- Visão processual da organização e dos seus sistemas
- Melhora e otimiza a dinâmica de processos da organização
- Melhor monitorização dos processos
- Mudanças ou alterações facilitadas de processos
- Relatórios de gestão mais completos e fidedignos
- *Feedback* para a optimização de processos
- Reaproveitamento e integração de sistemas existentes
- Preparação para futuras integrações entre organizações

2.2.4 Integração inter-organizacional

A um nível superior de conectividade, diferentes organizações integram directamente os seus sistemas de informação, seguindo os conceitos de B2B (*Business to Business*) e B2C (*Business to Consumer*) [Bussler 2003] entre outros. A área do B2B contempla as transacções entre organizações que interligam os seus processos e trocam a informação que os suporta. A área B2C abarca todas as empresas ou organizações que vendem os seus produtos ou serviços através da Internet ou Extranet. A solução mais comum é a criação de portais empresariais que disponibilizam esse conjunto de informação e serviços.

Uma arquitectura para B2B permite resolver problemas de integração entre diferentes organizações que colaboram e participam num mesmo processo. Esta abordagem permite a redução de custos e tempo numa cadeia de fornecimentos de bens ou serviços entre organizações. A tecnologia de suporte ao B2B permite um maior controlo dos processos e das interacções entre uma organização e os seus parceiros de negócio. Nesta área existem normas que regulam a estruturação dos documentos de negócio e a forma como são feitas as transacções electrónicas. Trata-se do intercâmbio de informação entre organizações e da definição dos documentos que vão trocar electronicamente de acordo com as respectivas normas e eventuais outros pressupostos.

Os principais benefícios associados a esta abordagem são essencialmente:

- Normalização da informação e documentação trocadas
- Sinergias organizacionais
- Definição de transacções comerciais e operacionais comuns
- Reaproveitamento e integração com sistemas existentes
- Melhoramento e integração das cadeias de valores
- Automatização e coordenação das dinâmicas organizacionais.

2.3 Os Web Services na integração de SI

Independentemente das necessidades de integração, as TI oferecem alternativas tecnológicas que se enquadram em normas técnicas especificadas para o efeito. A integração de SI tira partido destas normas e é cada vez mais orientada para a organização e os seus processos. Com a evolução tecnológica é hoje possível disponibilizar informação ou procedimentos como serviços (Figura 2.18) através de uma camada de abstracção suportada em tecnologia do tipo *Web Services* (WS) [Cerami 2002]. Os WS são “*componentes de software que permitem computação distribuída assenta em normas da Internet*” [ACS 2002] [Lopes e Ramalho 2005].

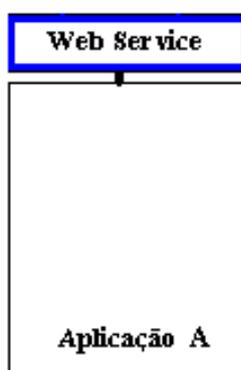


Figura 2.18: *Web Services*

O uso de WS pode fazer sentido em diferentes áreas tecnológicas e conceptuais permitindo compatibilizar soluções díspares, utilizando formatos comuns de troca de informação suportados na norma *Extended Markup Language* (XML) [Ramalho 2002]. Os WS definem regras de interacção e comunicação com cada serviço de forma a obter uma camada de abstracção, suficientemente completa, para poder interagir com o que está encapsulado. Estes funcionam com uma interface baseada em normas técnicas para facilitar a integração de SI.

Os *Web Services* podem ser utilizados em qualquer uma das perspectivas da integração mencionadas nos pontos anteriores tirando partido de todas as vantagens de cada uma delas. Estes criam uma nova camada de acesso aberta sobre cada tecnologia, permitindo tirar partido da sua informação ou dos seus procedimentos.

2.4 As camadas tecnológicas e a sua distribuição

As TI disponibilizam um conjunto vasto de funcionalidades, que podem não estar directamente ligadas ao tema da integração de SI, mas estão subjacentes e associadas directamente a esta área. Neste sentido, é também necessário um enquadramento da tecnologia e de algumas definições utilizadas ao longo dos próximos capítulos.

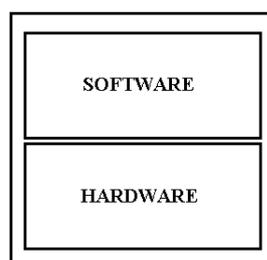


Figura 2.19: Camadas tecnológicas

De forma geral, pode-se visualizar a tecnologia estruturada por camadas complementares. Numa visão técnica simplista identifica-se as camadas básicas do *hardware* e do *software* (Figura 2.19). Por sua vez, a camada base de *software* pode ser decomposta pelo nível dos sistemas operativos e por todo um conjunto de aplicações, ou soluções informáticas, com diferentes fins. Neste âmbito, um sistema de gestão de base de dados (*Database Management System - DBMS*) ou um Servidor Aplicacional (*Application Server - AS*) fazem parte desta segunda decomposição (Figura 2.20).

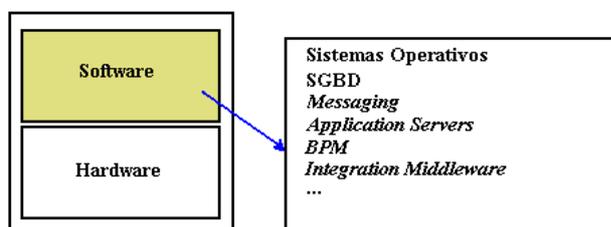


Figura 2.20: Camada do *Software*

É essencialmente dentro da camada de *software* que se enquadram as soluções para a integração de SI. Estas assentam em protocolos de comunicação, outros componentes de *software* e normas tecnológicas. Se identificarmos a integração de SI

como uma camada conceptual esta tem uma relação directa com os protocolos de comunicação das camadas das redes e do *middleware* (Figura 2.21), herdando muitas das suas características tecnológicas [Silva 2003]. Considera-se o *middleware* um conjunto de *software* com funcionalidades que permitem a interacção entre vários sistemas ou aplicações que residem em diferentes máquinas.

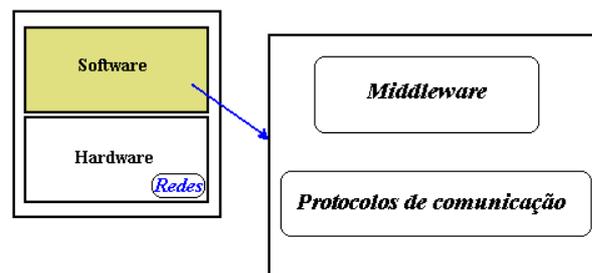


Figura 2.21: *Middleware* para a integração

Como se detalha nos capítulos seguintes a camada do *middleware* é uma peça fundamental para a computação distribuída e para potenciar a comunicação entre sistemas e a sua integração. Esta pode ter várias implementações sendo o seu principal propósito a integração entre componentes de *software*, garantindo o seu acesso e a sua disponibilização. Esta camada permite reduzir o impacto da interligação de diferentes tecnologias de comunicação, sistemas operativos, aplicações e linguagens de programação.

O *Middleware* tem um conjunto de serviços, normalmente assentes em normas técnicas, que suportam por exemplo a execução de código, transacções distribuídas, segurança, balanceamentos de carga, *clustering*, *backups*, comunicações, etc. Nesta camada, pode-se encontrar soluções do tipo *Transaction Processing Monitors* (TPM), *Application Servers* (AS), *Message Brokers* (MB), *Web Server* (WES), *Workflow Server* (WFS) e *Object Request Broker* (ORB) entre outros.

Application	Applications written to run over the network <i>Examples:</i> email, file transfer, order-service, HR, ...
Presentation	Protocol and data conversion, transport-independent policies <i>Examples:</i> encryption, QoS routing, caching, security
Session	Session management, checkpointing, reliable delivery <i>Examples:</i> HTTP 1.1, SSL, SQL, NFS, RPC
Transport	Flow control, error checking, guaranteed delivery <i>Examples:</i> TCP
Network	Logical->physical address translation, route and congestion management <i>Examples:</i> IP, X.25
Data link	Packets-to-bits conversion, data frames <i>Examples:</i> MAC, LLC
Physical	Transmits raw bitstream over physical cable

Figura 2.22: As 7 camadas da pilha OSI

A camada de *middleware* [Schreiber 1995, Bernstein 1996] suporta várias formas de comunicação entre sistemas e componentes. Para que a integração de sistemas de informação seja possível são necessários canais de comunicação que permitem o intercâmbio de informação. Neste âmbito, a tecnologia fornece várias alternativas para a troca de dados entre módulos aplicativos. Estes canais de comunicação assentam nas redes estruturadas que interligam esses sistemas e que permitem estabelecer ligações via protocolos como por exemplo o TCP/IP, o IPX ou NetBEUI. Sobre estas camadas são usados outros protocolos como é o HTTP ou o RPC [Gunter 1995] entre outros. O modelo OSI (Open System Interconnection Model) ilustra cada uma das camadas de suporte às comunicações e troca de informação entre aplicações distribuídas (Figura 2.22). Acima destas camadas podem identificar-se mais quatro camadas para suportar os processos organizacionais e a integração de SI [Carroll 2000]:

- Business Processes (Company-specific Business Processes)
- Business Semantics (Company-specific data definitions and structures)
- Interface Syntax (API, DCOM, CORBA, etc...)
- Integration Middleware (*Event-Driven* ou *Data-Driven*)

Numa visão conceptual, pode-se identificar camadas lógicas dentro da camada de software. A primeira abordagem corresponde à arquitectura Cliente/Servidor [Schussel 1996, Edelstein 1994, Zahavi 1999] (Figura 2.23). A camada do cliente

invoca serviços ou procedimentos fornecidos pela camada do servidor. É seguida uma configuração do tipo *Master/Slave* em que o sistema cliente procura recursos existentes no servidor que está totalmente dedicado a este tipo de funções. Os servidores de ficheiros ou de impressoras são exemplos deste tipo de partilha de recursos e informação.

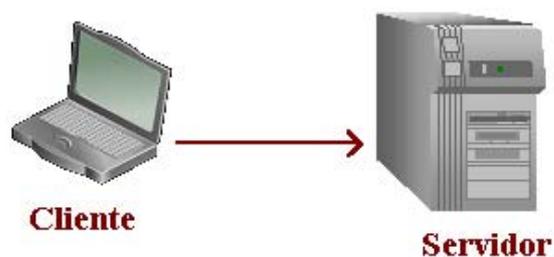


Figura 2.23: Arquitectura Cliente/Servidor

Neste tipo de arquitectura surge o conceito de *Application Programming Interface* (API) que representa uma interface aplicacional para facilitar a integração e troca de informação entre duas ou mais aplicações [Hines 1996] numa configuração do tipo Cliente/Servidor. Neste âmbito surge também o conceito de *Remote Procedure Call* (RPC) que permite a invocação de procedimentos remotos [Rao 1995, Birrell e Nelson 1984] [Gunter 1995] entre aplicações.

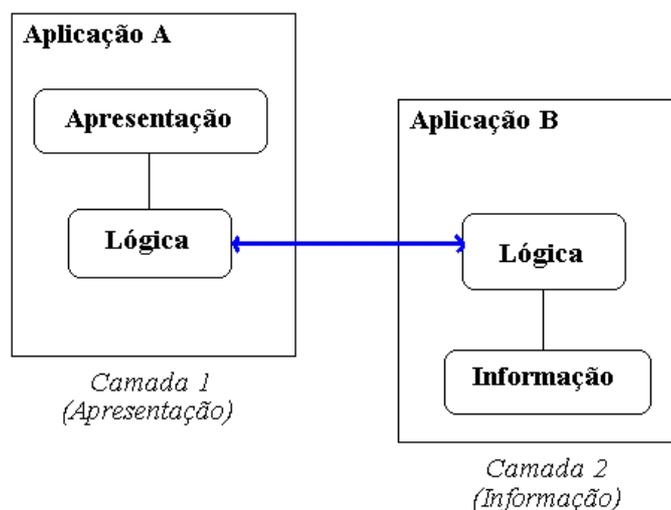


Figura 2.24: Arquitectura *two-tier*

A abordagem das duas camadas [Schussel 1996, Edelstein 1994], também denominado de *two-tier*, enquadra-se no âmbito do Cliente/Servidor mas com uma

diferença na distribuição da lógica aplicacional em duas camadas complementares. Uma parte da programação reside do lado do cliente em conjunto com a camada de apresentação e outra parte está do lado do servidor em conjunto com a camada da informação (Figura 2.24). Como já foi referido, as aplicações são comumente divididas em três camadas lógicas [Eckerson 1995 e Wong 2001] (Figura 2.25):

1. Camada cliente ou de apresentação (*client-tier*)
2. Camada de *business logic* ou processamento (*middle-tier*)
3. Camada da informação ou de base de dados (*information-tier*)

A camada cliente ou *front-end* corresponde à área de apresentação das aplicações e da informação. Por exemplo, numa Intranet o portal empresarial é acessado através de um *web-browser* o que corresponde a esta camada aplicacional.

A camada intermédia do *middle-tier* incorpora a lógica aplicacional com as suas regras e procedimentos. Esta camada tira partido do *middleware* e potencia a comunicação entre sistemas permitindo vários níveis de integração. O próprio *middle-tier* pode estar repartido por vários estratos ou sub-camadas tecnológicas.

A camada da informação contempla todas as soluções de armazenamento e gestão de informação. Aqui encontram-se normalmente os SGBD (Sistemas de Gestão de Bases de Dados) que funcionam como repositórios de informação.

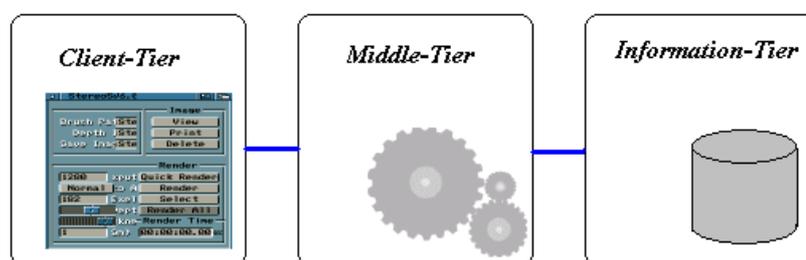


Figura 2.25: Arquitectura 3-tier

Seguindo esta lógica de camadas, esta arquitectura pode ser subdividida em n camadas complementares, também denominada por arquitectura *n-tier* [Chartier 2004, Carnegie 1997]. Esta arquitectura, também designada por *multi-tiered*, permite identificar diferentes camadas complementares, como por exemplo: a camada de

apresentação, a camada *web server* e a camada de informação onde normalmente estão as bases de dados. A camada lógica do *middleware* referida acima pode ela própria estar repartida por *n* camadas intermédias neste tipo de abordagem. Uma estrutura *n-tier* engloba todos estes conceitos e fornece uma plataforma que permite simplificar e unificar diferentes tipos de aplicações e interfaces.

Cada camada pode estar fisicamente distribuída, ou logicamente integrada numa aplicação. Cada uma delas permite diferentes tipos de integração desde o nível da gestão da informação até ao nível da sua apresentação. Em cada nível poder-se-á encontrar funcionalidades complexas e diferentes para a integração de aplicações, processos e informação. Esta visão conceptual é importante para se entender a abrangência tecnológica da integração de SI que permite abordagens diferentes mas igualmente válidas.

2.5 Papel das normas para a integração de SI

A integração de SI corresponde a um mercado muito competitivo e dinâmico onde existem empresas que foram criadas exclusivamente para solucionar os seus problemas específicos. Cada solução tem vantagens e desvantagens consoante a sua flexibilidade e abrangência. Há situações onde uma solução integradora pode não se revelar a mais adequada obrigando a outras alternativas. Nos últimos anos têm surgido normas, quer para documentar, quer para suportar abordagens e tecnologias nesta área. Hoje em dia, ainda existem soluções proprietárias mas a tendência é para adoptar normas técnicas, garantindo a sua abertura e certificação. Neste âmbito, as normas definidas por organismos credenciados representam um factor importante na adaptabilidade e adequação de cada solução.

Pode-se definir uma norma como sendo “*uma categoria específica de tecnologia de informação definida por uma especificação pública ou controlada por uma organização independente ou que esteja em uso há pelo menos um ano por mais de 50% das aplicações na sua classe*”¹. Neste sentido, a criação de normas técnicas tem

¹ John Schmidt, Methodology Committee Chairman, EAI Industry Consortium

dois caminhos paralelos e por vezes concorrentes. Existem as normas “*de facto*” que não são mais do que a adopção generalizada de uma especificação tecnológica, pertencente a um ou mais fornecedores, mas que não está sujeita a um organismo regulador oficial. Por outro lado, as normas denominadas “*de jure*”, ou “formais”, são especificadas por organizações credenciadas para o efeito garantindo a sua independência e a evolução de cada especificação. Em paralelo há esforços de cooperação entre empresas que se unem na criação de uma norma que será posteriormente submetida a um organismo credenciado para a tornar formal.

A adopção de normas específicas na integração de SI tem sido estimulada por um conjunto de factores específicos [Craggs 2003], nomeadamente:

- Integrar processos dentro e entre organizações
- Capacidade de interligar sistemas internos e aplicativos
- Suportar processamento de transacções existentes nos sistemas
- Disponibilização de serviços para fora da organização
- Crescente complexidade das tecnologias para a integração de SI
- Diminuição de custos de integração de SI

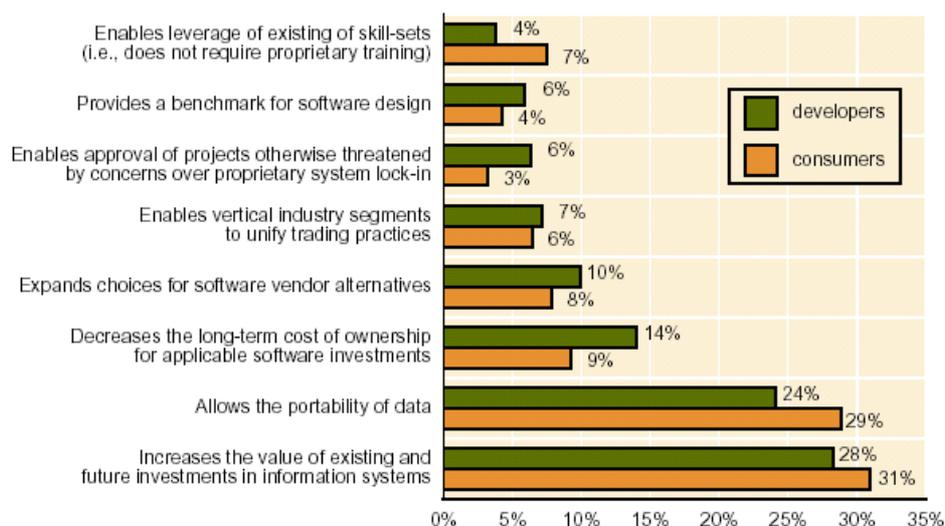


Figura 2.26: Benefícios no uso das normas.

Existem diversas normas de mercado que permitem disciplinar a evolução tecnológica e documentar cada arquitectura com as suas respectivas funcionalidades.

Um estudo recente do Delphi Group², denominado de “*The Value of Standards*”, realça várias vantagens na especificação de normas de mercado onde cerca de 31% das empresas afirmam que “*as normas aumentam o valor dos actuais e futuros investimentos em SI*”, e 29% destas valorizam “*a portabilidade dos dados*” (Figura 2.26). Ainda no mesmo estudo, foi perguntado aos inquiridos quais eram as normas que deveriam estar garantidas nas soluções apresentadas pelos seus fornecedores (Figura 2.27). Destaca-se claramente o XML como norma para a estruturação e portabilidade da informação em vários domínios. Logo a seguir surgem duas normas relativas a plataformas de desenvolvimento de aplicações: o J2EE da SUN com 44% e o .Net da Microsoft com 36%. O protocolo SOAP surge com 35% e está relacionado com os *Web Services* e a troca de informação baseada em XML.

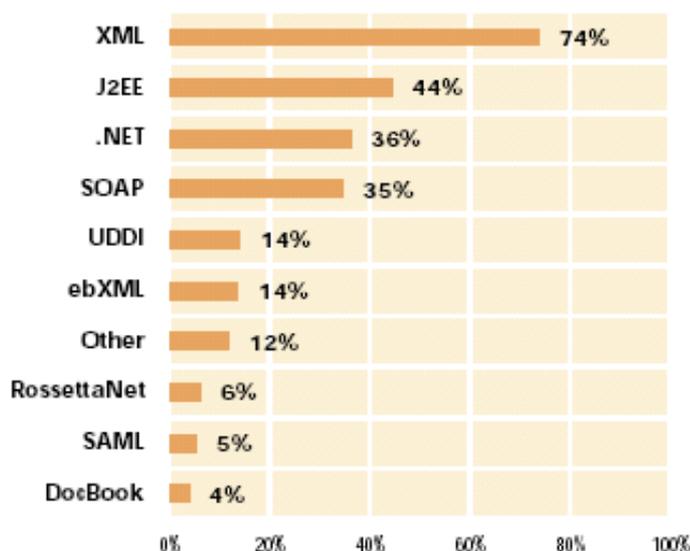


Figura 2.27: Normas exigidas aos fornecedores de TI.

No caso da integração de SI, a identificação das normas a adoptar pode revelar-se difícil e confusa visto que existem vários organismos e empresas, com diferentes interesses, que participam nestas especificações. Geralmente, a evolução do mercado e das próprias tecnologias é que dita a actualidade e a adopção de cada uma das normas que vão surgindo. Estas são normalmente identificadas pelo nome de “*standard*” e são os alicerces para a construção de soluções integradoras abertas e certificadas.

² <http://www.delphigroup.com/research/whitepapers.htm>

2.6 Condicionantes da integração de SI

Dada a natureza complexa dos projectos de integração de SI, há factores que podem determinar o seu fracasso e prejudicar significativamente a organização em causa [Spiers 2005, Chase 2000, Levine 2005]. Algumas das razões mais comuns para que os projectos de integração de SI falhem têm a ver com:

- A implementação é mais complexa do que o esperado.
- Falta de comunicação na organização.
- Falta de metodologias normalizadas e abrangentes para a integração de SI.
- Problemas com a integridade de informação e as transações associadas.
- Modelos de dados heterogéneos.
- Diferenças entre a implementação de uma aplicação e a integração de SI.
- Tecnologias incompatíveis.
- Falta de estratégia na gestão e planeamento dos SI.
- Escolha de soluções menos adequadas.
- Desconhecimento ou falta de documentação da estrutura dos SI em uso.

Os projectos para a integração de SI beneficiam a organização como um todo permitindo melhorar o seu desempenho e dinâmica a todos os níveis [Kelly 2003]. Neste âmbito, existem abordagens na definição de padrões conceptuais para representar tecnologias e acções na fase de análise de um projecto de integração de SI [Yee 2001, Hohpe et al. 2003]. A criação de modelos conceptuais para a integração de SI tem sido uma das abordagens para a modelação deste tipo de sistemas complexos. O *EAI Industry Consortium* tem desenvolvido esforços para a definição de uma metodologia para a integração de SI, e defende que qualquer metodologia potencial nesta área tem de ter em conta cinco diferenças de base face às metodologias tradicionais que abarcam o ciclo de vida de um projecto de desenvolvimento de SI:

- “O todo é maior que a soma das partes”.
- Projectos em constante mutação.
- Não há normas universais e o leque de escolha é alargado.
- Contexto da informação dinâmico com interpretações diversas.

- Todos os detalhes são relevantes.

Independentemente das formas de representação ou das metodologias, para que um projecto desta natureza tenha sucesso tem de se garantir algumas condições fundamentais sob pena de não se obter os resultados desejados:

- Apoio e envolvimento da Direcção da organização
- Análise custo/benefício do investimento a médio e longo prazo
- Gestão de processos organizacionais
- Abordagem e planeamento estruturado para o projecto e sua equipa
- Normas e arquitecturas tecnológicas certificadas
- Potenciar investimentos em SI já existentes

Nesta área, há claramente duas alternativas para a obtenção da solução. A primeira é construí-la à medida, e a outra é comprar um sistema específico. Em qualquer dos casos, é imperativo analisar a adequação, o impacto e o retorno de tal investimento para a organização.

Todas as soluções para a integração de SI suportam e controlam a comunicação entre sistemas heterogéneos permitindo a sua compatibilidade. Em cada uma delas encontram-se diferentes funcionalidades que devem geralmente ter em conta quatro desafios fundamentais [Hohpe et al. 2003]:

- Troca de dados entre sistemas: As soluções para a integração de SI têm de transportar dados entre computadores através de redes de comunicação.
- Utilização de redes de computadores: A computação distribuída tem de lidar com um grande número de problemas que podem gerar atrasos ou interrupções. O envio de dados através de uma rede é normalmente muito mais lento do que a invocação local de um método ou procedimento remoto.

- Heterogeneidade dos sistemas interligados: As soluções de integração de SI precisam de transmitir informação entre sistemas que usam diferentes linguagens de programação, plataformas operacionais e formatos de dados. Estas soluções devem permitir interligar estas diferentes tecnologias.
- Impacto das mudanças dos sistemas interligados: Estas soluções devem também garantir que as mudanças nas aplicações informáticas interligadas não tenham grande impacto na solução integradora encontrada. Para isso devem minimizar as dependências entre sistemas usando a filosofia de *loose coupling* que permite proteger os sistemas das mudanças efectuadas nos outros sistemas com os quais têm ligações.

Para além destas situações, as soluções para a integração de SI devem fornecer documentação clara e inequívoca das suas funcionalidades e das suas bases tecnológicas. Estas devem também seguir as normas de mercado para garantir a sua abertura, adaptabilidade e certificação. Desta forma é reforçada a garantia de independência e credibilidade, protegendo as organizações de dependências tecnológicas ou da inadequação das soluções encontradas [Allen 2001]. A integração de SI é, para além de complexa, uma área cada vez mais estratégica para as organizações no sentido em que pode pôr em causa o seu bom funcionamento e desempenho.

2.7 Conclusão

A integração de sistemas pode revelar-se complexa no seu entendimento e na sua implementação. Os Sistemas de Gestão de Bases de Dados (SGBD), os *Remote Procedure Calls* (RPC), os *Integration Brokers* (IB) ou os *Messaging-Oriented Middlewares* (MOM) são alguns exemplos de soluções tecnológicas abrangentes que permitem integrar aplicações informáticas, directamente ao nível dos seus dados ou dos seus procedimentos. Existem perspectivas mais específicas, como a integração de aplicações, que fornecem um conjunto de funcionalidades para troca de informação síncrona ou assíncrona, garantindo a capacidade de resposta dos sistemas heterogéneos. Por seu turno, a perspectiva da integração de processos dá maior

importância aos processos organizacionais permitindo automatizá-los integrando diferentes tipos de aplicações informáticas. Já a um nível externo, a perspectiva da integração inter-organizacional centra-se na informação e nos processos partilhados entre diferentes organizações.

Face a todos estes factores, pode-se definir diferentes abordagens, ou visões, para a integração de SI de acordo com perspectivas tecnológicas e funcionais abrangentes. O nível da complexidade organizacional vai determinar o tipo de abordagem a seguir face a cada perspectiva. Neste sentido são identificadas, e detalhadas, neste trabalho três orientações para integração de SI que se centram nos seguintes pontos:

- A informação.
- As aplicações informáticas.
- Os processos organizacionais.

Cada visão da integração de SI tem soluções enquadradas em várias arquitecturas tecnológicas e em diversas normas de mercado. Cada uma pode partilhar conceitos e tecnologias semelhantes e chegam mesmo a ser complementares. O surgimento dos *Web Services* criou uma nova visão desta problemática permitindo novas abordagens para as diversas formas de integração de SI. Esta tecnologia possibilita definir uma Arquitectura Orientada aos Serviços (*Service Oriented Architecture - SOA*) que suporta diferentes tipos de integração de sistemas baseando-se nas especificações técnicas e normalizadas dos *Web Services*.

3. Abordagens para a integração de SI

A integração de sistemas de informação nas organizações tem várias vertentes de actuação e diferentes abordagens. Como se referiu no capítulo anterior, existem muitas formas de catalogar a integração e de ilustrar conceptualmente as suas arquitecturas. No entanto, identificou-se que a informação, as aplicações e os processos organizacionais são três elementos complementares para a integração de SI. Neste sentido, e tendo em conta estas três realidades, pode-se identificar pelo menos três perspectivas de integração correspondentes:

1. Perspectiva da integração da informação.
2. Perspectiva da integração aplicacional.
3. Perspectiva da integração dos processos.

Como também foi referido, a evolução das TI potencia novas arquitecturas que permitem criar camadas tecnológicas integradoras para os sistemas de informação existentes. Neste âmbito, os *Web Services* permitem isolar uma aplicação, ou um sistema, e disponibilizá-los como um serviço. É possível desta forma equacionar a integração dos sistemas de informação numa quarta perspectiva mais abrangente:

4. Perspectiva da integração por *Web Services*.

Cada uma destas perspectivas tem partes comuns e pode resolver problemas idênticos. A escolha da solução dependerá da realidade da organização, das suas necessidades, dos seus processos, das TI existentes e da estratégia de integração encontrada

Neste capítulo descreve-se cada uma das perspectivas de forma a poder diferenciá-las e enquadrá-las com o tema da integração. Em complemento, e para cada uma delas, é feita uma associação com as respectivas normas e tecnologias mais comuns existentes no mercado. No final são apresentados alguns quadros que permitem relacionar perspectivas, normas e tecnologias.

3.1 A Integração da informação

Toda a forma de armazenamento, troca e gestão da informação está sujeita a constantes actualizações que podem originar diferentes necessidades de integração. O âmbito desta perspectiva engloba as funcionalidades que permitem ter acesso a informação correcta e actualizada, seja através de sincronismos entre repositórios, seja através da centralização dos acessos a uma única camada de informação. Enquadra-se neste campo a gestão da informação do tipo operacional, que é constantemente actualizada pelas aplicações do tipo *On-Line Transaction Processing* (OLTP), e a consolidação de repositórios de informação com objectivos analíticos, do tipo *Multidimensional On-Line Analytical Processing* (OLAP). Neste último ponto, surgem tecnologias específicas como é o caso dos *Datawarehouses*, dos EIS (*Executive Information Systems*), das soluções multi-dimensionais *Multidimensional On-Line Analytical Processing* (MOLAP) ou do *DataMining*. Apesar dos sistemas OLTP terem objectivos diferentes dos sistemas OLAP, existem neste âmbito necessidades comuns de integração, quer ao nível da gestão e armazenamento da informação, quer ao nível da sua apresentação.



Figura 3.1: Sistema de Gestão de Base de Dados

Devido à importância da informação dentro das organizações foram surgindo tecnologias para o seu armazenamento e exploração. Em primeiro lugar, apareceram os sistemas de gestão de ficheiros [Pereira 1998] associados à linguagem de programação COBOL (*COmmon Business Oriented Language*) que tiveram uma grande difusão no mercado. Para colmatar as lacunas existentes, surgiram os Sistemas de Gestão de Base de Dados (SGBD) que permitiram centralizar o acesso à informação e criar repositórios estruturados de informação e código aplicacional (Figura 3.1). Este tipo de solução também permite transportar e replicar informação

entre repositórios afim de sincronizar o seu conteúdo ou obter um novo repositório com informação estruturada. Estas operações são por exemplo efectuadas através de funcionalidades de ETL (*Extraction-Transformation-Loading*), *Data Replication* e outras formas de troca de informação. Nos pontos seguintes deste capítulo detalha-se algumas das soluções mais comuns para a perspectiva da integração da informação.

3.1.1 Ficheiros, SGBD e Replicação de Informação

Com o objectivo de partilhar informação entre duas aplicações informáticas, existem soluções tecnológicas que permitem enviar dados de uma aplicação para outra, com suporte em diferentes plataformas e métodos. Por exemplo, através de uma rede de computadores, uma aplicação pode enviar informação armazenada num ficheiro de texto electrónico para outra aplicação, que o recebe e interpreta o seu conteúdo (Figura 3.2). A informação contida no ficheiro está estruturada de forma a que ambas as aplicações entendam o seu conteúdo e possam dar-lhe a mesma interpretação. Este exemplo centra-se no envio de informação num único sentido, mas existem soluções mais complexas que permitem gerir a troca de ficheiros nos dois sentidos, e que requerem algoritmos específicos para controlar a integridade e consistência da informação.

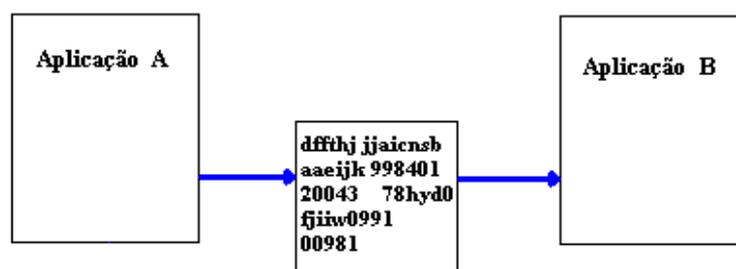


Figura 3.2: Envio de dados em ficheiro

É comum integrar duas ou mais aplicações através da informação que manuseiam, centralizando-a num único repositório. Neste caso, são normalmente usados Sistemas de Gestão de Base de Dados (SGBD) com diferentes formas de armazenar e disponibilizar a informação (Figura 3.3). Esta abordagem permite controlar

eficazmente a integridade e actualidade do seu conteúdo. Para além da informação estruturada, a grande maioria dos SGBD permite também armazenar e partilhar ficheiros e código aplicacional.

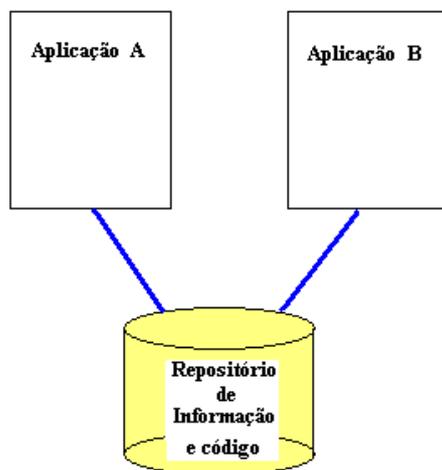


Figura 3.3: Partilha de repositório de informação

Os SGBD funcionam como repositórios de informação seguindo um modelo de dados único e comum a todas as aplicações existentes. Os SGBD seguiram entre outros os modelos hierárquicos, relacionais e *object-oriented* [Pereira 1998]. Neste âmbito, a forma mais comum de acesso à informação usa a linguagem de interrogação SQL (*Structured Query Language*) [Date 2003]. Neste âmbito, a modelação mais comum é a criação de diagramas do tipo *Entity-Relationship Model* (E-R) [Thalheim 2000] que permitem documentar modelos relacionais de dados que são implementados e geridos num motor de base de dados relacional (SGBDR - Sistema de Gestão de Base de dados Relacional). Esta abordagem garante a qualquer sistema que a informação está actualizada e homogénea, sem estar repetida nem dispersa.

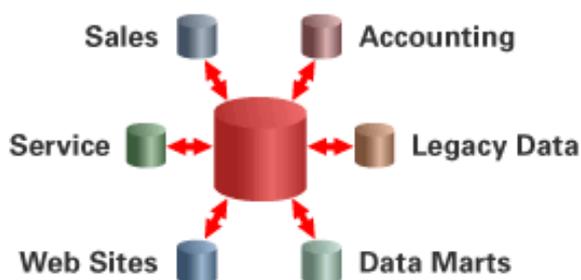


Figura 3.4: *Customer Data Hub*.

Com base nos SGBD surgem soluções do tipo *Data Hub* (DH) cujo objectivo é fornecer repositórios genéricos, com modelos de dados genéricos, abrangentes e previamente definidos, para servir qualquer aplicação dentro de uma organização (Figura 3.4). O exemplo mais comum é o repositório único de clientes de uma empresa, no DH, sobre o qual são feitas todas as actualizações por parte de cada sistema informático.

Apesar da proliferação dos SGDB como primeiro nível de integração da informação, as organizações podem necessitar de consolidar diversas fontes de informação. Existem vários mecanismos de sincronização de informação entre bases de dados. O mais comum centra-se na replicação de dados (*Data Replication*) de forma síncrona ou assíncrona. Este tipo de *software* encarrega-se de actualizar as várias bases de dados para garantir a consistência e a integridade da totalidade da informação. Com diferentes periodicidades, cada base de dados terá uma réplica fiel da informação existente nas restantes (Figura 3.5). A abordagem mais comum encontra-se nas empresas que têm colaboradores cujo trabalho é andar pelo país a vender ou entregar mercadorias. Cada um deles pode ter uma pequena base de dados instalada num computador portátil que lhe permite ir actualizando informação. Os diversos mecanismos de replicação encarregam-se de actualizar a informação entre as n pequenas bases de dados e a base de dados central da organização. Estes mecanismos asseguram a resolução de eventuais conflitos.

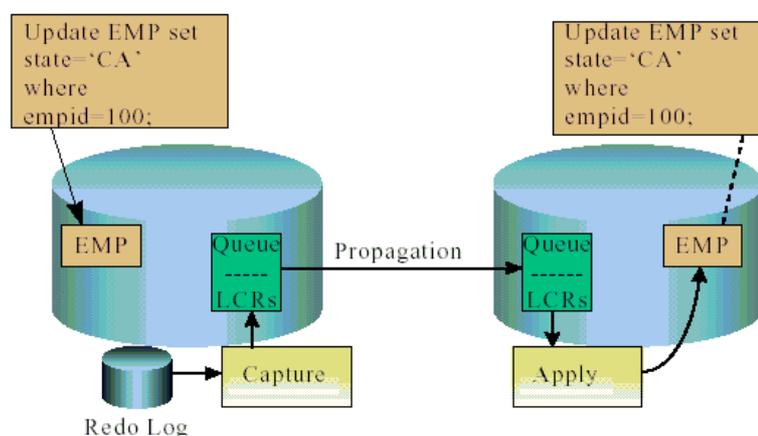


Figura 3.5: Replicação de informação (adaptado de Oracle Streams 10g).

3.1.2 ETL, TPM, EII, ECM e XML

As soluções do tipo ETL (*Extraction-Transformation-Loading*) permitem actualizar de forma assíncrona diferentes fontes de informação. Enquanto a replicação entre bases de dados abrange toda a informação existente, o ETL centra-se em conjuntos de informação identificados. Esta abordagem é normalmente seguida para obter repositórios de informação reorganizados, sobre os quais se efectuam operações analíticas e não transaccionais. As operações de ETL estão normalmente relacionadas com a área dos *Datawarehouses*, ou de *DataMining*, mas podem ser utilizadas para replicação de informação entre repositórios, assumindo neste caso um papel de integração de informação. Para isso, as soluções de ETL evoluíram para permitir um acesso online a um conjunto de informação agregada, aproximando-se das funcionalidades de ECM e EII descritas a seguir.

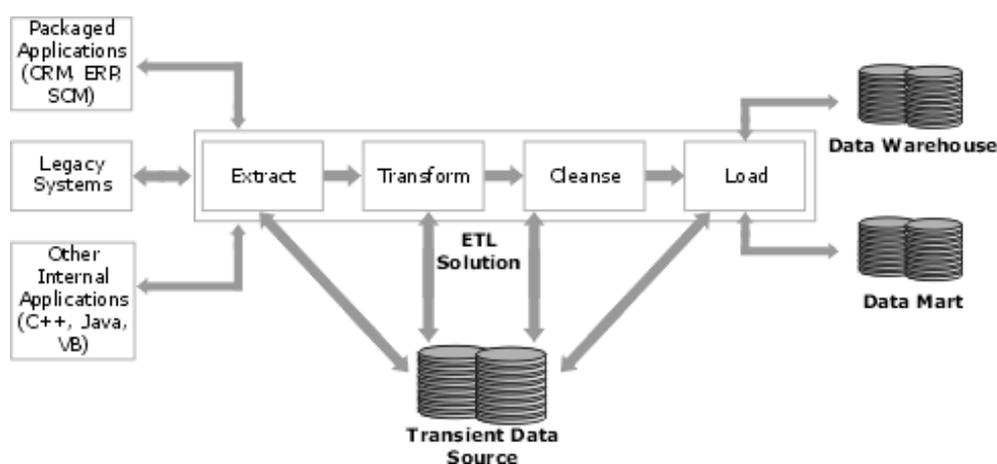


Figura 3.6: ETL (adaptado de ETL study – Marc Songini - Computerworld)

Os procedimentos de ETL executados em modo *batch* permitem extrair e mover dados a partir de diferentes fontes, formatá-los, filtrá-los e carregá-los para outros repositórios de forma a fornecer uma base de informação de gestão (Figura 3.6). Uma das etapas mais importantes tem a ver com a operação de filtragem de dados, ou depuração, que é também designada por *cleansing*. As soluções ETL são úteis para produzir repositórios que serão utilizados em tarefas de análise histórica de informação e suportar decisões de gestão.

A frequência com que se fazem operações de ETL, e pela sua orientação à informação, pode não se revelar suficientemente rápida para a dinâmica da organização, e pode não suportar as reais necessidades de integração. No caso dos sistemas transaccionais do tipo OLTP, tem de haver uma frequência elevada de sincronização da informação entre repositórios. Nestes casos, os SGBD fornecem soluções de replicação, assim como as soluções do tipo *Gateways* de integração ou pode-se também utilizar soluções do tipo TPM (*Transaction Processing Monitors*) [Dickman 1995] que controlam a actualização da informação em diversos repositórios, garantindo a sua integridade, sincronismo e actualidade. Os TPM garantem a consistência da informação distribuída controlando a execução das transacções entre sistemas e garantindo a partilha de informação [Hudson e Johnson 1994]. Este tipo de funcionalidade encontra-se na área das soluções de *Middleware*, em particular nas tecnologias de *Message Queueing* em que se baseiam os *Integration Brokers*. Encontram-se aqui diversas alternativas para troca e partilha de informação.

Os diferentes sistemas envolvidos não interpretam nem gerem forçosamente a informação da mesma forma. Neste sentido, as soluções integradoras de informação devem ter funcionalidades que garantam o sucesso quer do envio quer da recepção da informação. Estas características centram-se no tratamento dos dados veiculados de forma a que cada sistema interveniente entenda o seu conteúdo e lhe dê uma correcta interpretação. As principais funções para o controlo da integração da informação ao nível do *Middleware* são:

- *Data Mapping*: Tratamento da correspondência correcta dos dados entre sistemas.
- *Data Translation*: Tradução do significado da informação entre sistemas.
- *Data Transformation*: Transformação de dados entre sistemas.
- *Data Validation*: Validação da correcção dos dados.
- *Data Routing*: Correcto encaminhamento de dados entre sistemas.
- *Data Common view*: Estrutura de dados comum entre os sistemas.

Estas funções validam, traduzem, transformam e encaminham os dados com o objectivo de garantir que estes são sempre entendíveis e interpretados correctamente pelos sistemas receptores.

Com o objectivo de centralizar o acesso à informação, é possível integrar informação de diversas fontes ao nível da sua apresentação ao utilizador (*User Interface - UI*) (Figura 3.7). Por exemplo, através de um *web browser*, um utilizador poderá aceder, através de um único écran, a informação diversa para a qual tem autorização.

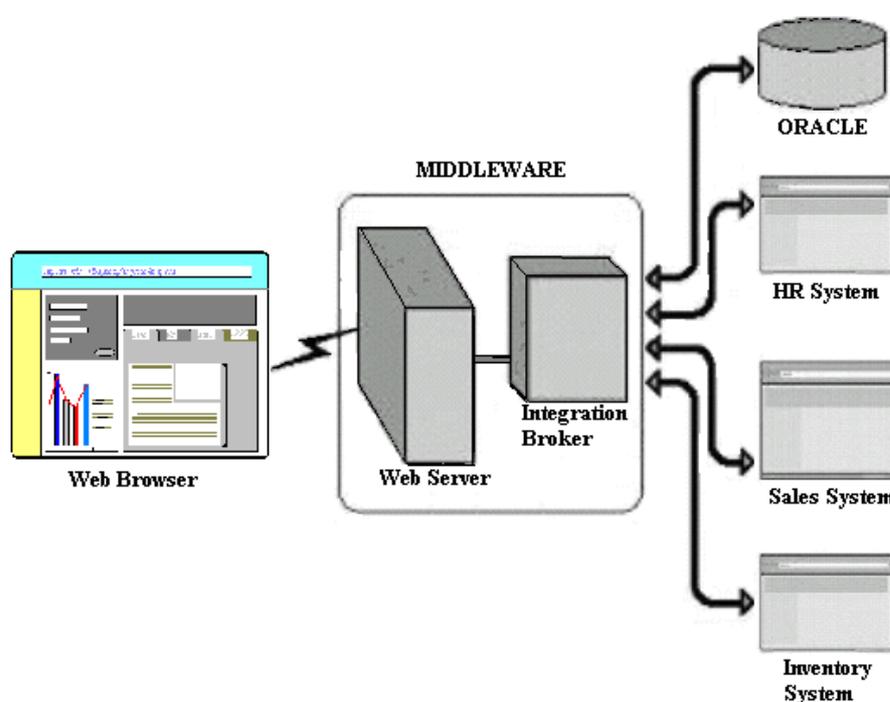


Figura 3.7: Integração da informação na camada de apresentação com *Middleware*.

Dentro das soluções de *Middleware*, a maioria dos *Application Servers* (AS) disponibilizam funcionalidades, na área dos *Web Servers*, designadas por Portal e que incluem esta finalidade. Este tipo de integração da informação está também relacionado com as áreas de ECM (*Enterprise Content Management*) [Jenkins et al. 2004] e de EII (*Enterprise Information Integration*) [Morgenthal 2005].

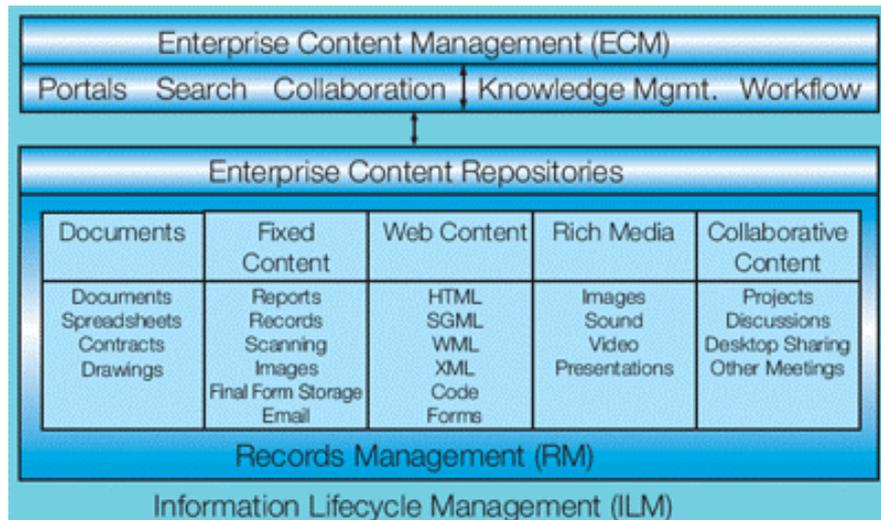


Figura 3.8: *Enterprise Content Management (ECM)*

As abordagens do tipo ECM (*Enterprise Content Management*) tem como objectivo integrar a informação ao nível da camada de apresentação (Figura 3.8). O ECM teve origem em primeiro lugar na gestão documental e depois na gestão de conteúdos para a *Web*. Na área do ECM encontram-se essencialmente os portais empresariais que consolidam informação não estruturada de diversas origens. Considera-se por exemplo como informação não-estruturada os ficheiros de texto ou de imagem. Embora a área do ECM esteja fortemente conotada com a gestão de conteúdos, esta já começa a englobar integração de informação estruturada normalmente armazenada em bases de dados. No ECM encontram-se soluções do tipo “portal” que servem de peça integradora para diferentes fontes de informação, quer sejam aplicações acessíveis via *web*, quer sejam documentos ou outro tipo de informação não estruturada. O objectivo mais comum de uma solução de ECM é a gestão centralizada de conteúdos informativos para dentro de uma organização. O ECM pode ser considerado como um subconjunto do EII (*Enterprise Information Integration*).

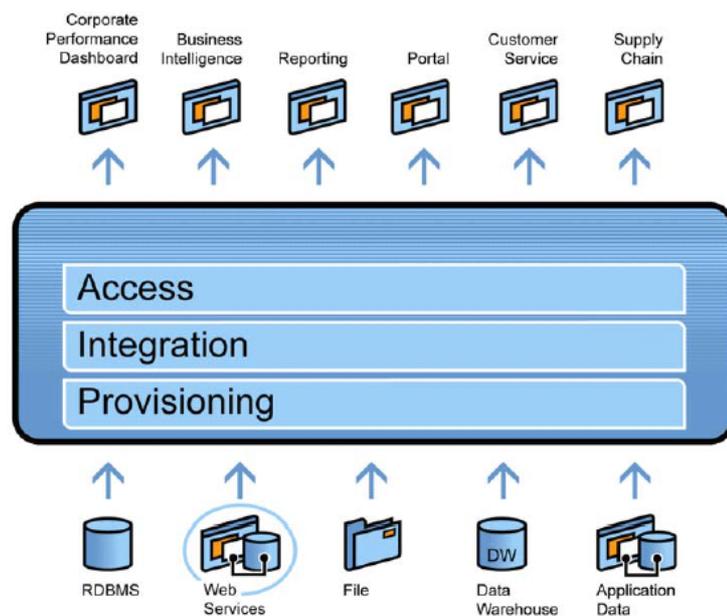


Figura 3.9: *Enterprise Information Integration (EII)*

As soluções de EII (*Enterprise Information Integration*) são mais abrangentes do que o ECM. Estas permitem agregar informação de várias fontes, interagir com ela, como se estivesse num único repositório centralizado, mantendo a sua dispersão e controlando as diferentes operações feitas sobre ela (Figura 3.9). Estas soluções são por vezes designadas de ECM mas têm um leque de funcionalidades e objetivos muito mais abrangentes. O EII fornece um acesso normalizado à informação integrada, com diferentes origens, permitindo ter uma visão lógica e única de toda a informação que está fisicamente armazenada em sistemas heterogéneos. É disponibilizada uma fonte de informação virtual e centralizada fornecendo às aplicações uma única camada de informação e eliminando a complexidade da gestão de diferentes conexões a diferentes fontes de informação. O EII abrange as funcionalidades de ETL, ECM e TPM que permitem a agregação da informação num repositório dinâmico.

Na área do EII surgem também funcionalidades analíticas complementares de *Business Intelligence* (BI) [Vitt et al 2002], onde se incluem as soluções de *Business Activity Monitoring* (BAM) [McKie 2003], que permitem obter em tempo real informação de gestão acerca da organização em causa, e disponibilizá-la de forma legível e estruturada aos respectivos gestores. As aplicações do tipo *Client Relationship Management* (CRM) [Reynolds 2002] tiram partido das soluções de ECM/EII para obter informação consolidada dos clientes da organização em questão.

Por seu turno, muitas das estratégias empresariais para o *Knowledge Management* (KM) [Serrano e Fialho 2003] tiram também partido das tecnologias de ECM/EII para partilhar informação. Cada abordagem vai depender do tipo de informação e do processo de integração pretendido (Figura 3.10).

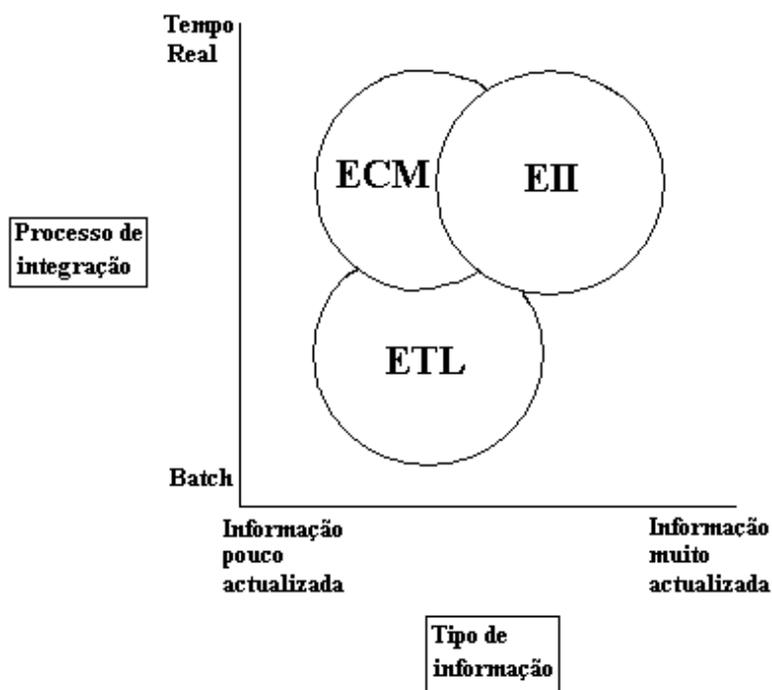


Figura 3.10: Panorama da integração da informação.

Praticamente todas as soluções referidas neste capítulo tiram partido do XML (*eXtended Markup Language*), quer para o armazenamento, quer para a troca de informação. Esta norma define uma linguagem padrão para a representação e troca de informação independentemente das tecnologias. O XML permite obter um grau elevado de portabilidade da informação e é claramente uma norma incontornável para qualquer abordagem na integração de SI.

Salienta-se que a maioria das soluções que integram a informação, quer com ETL, quer com ECM/EII, não têm funcionalidades de automatização de processos tipicamente existentes nas áreas do *Enterprise Application Integration* (EAI) ou do Business Process Management (BPM). O contrário já não acontece dado que existem soluções de EAI e BPM que permitem integrar informação numa lógica de replicação, de ETL ou mesmo de ECM/EII.

A integração da informação é mais complexa do que simplesmente juntar ou interligar fontes de informação. Trata-se por um lado de aceder aos repositórios certos nos momentos certos, e por outro lado, armazenar e sincronizar de forma correcta todas as fontes de informação existentes. Todos os esforços de consolidação e integração nesta área implicam uma procura de consistência e de qualidade da informação. Esta integração tem muitas abordagens válidas que por vezes se sobrepõem. Existem neste sentido diferentes métodos ou soluções para a integração de informação consoante o objectivo pretendido. Cada solução complementa-se com funcionalidades normalmente existentes noutras abordagens, por exemplo, as soluções de ECM e de EII partilham características idênticas e tendem a confundir-se.

Qualquer um dos caminhos referidos neste capítulo permitirá obter uma visão consolidada da informação, quer ao nível do seu armazenamento, quer do seu acesso ou disponibilização. Ainda neste âmbito, começam a surgir normas de mercado como são, por exemplo, o *Web Services for Remote Portlets* (WSRP) especificada pela *Organization for the Advancement of Structured Information Standards* (OASIS), ou o *Java Specification Request* (JSR) 168 definida pelo *Java Community Process* (JCP) que permitem a interoperabilidade de componentes do tipo *portlet* [Hepper 2003] entre diferentes plataformas. Nesta área, o XML é claramente a norma mais presente quer para estruturar a informação no seu armazenamento, quer para a sua troca, distribuição e apresentação.

3.2 A Integração aplicacional

O principal objectivo desta integração centra-se na interligação directa entre aplicações denominado por *Application-to-Application* (A2A) [Bussler 2003]. É comum designar esta área por *Enterprise Application Integration* (EAI) [Serain 2002], mas esta designação é demasiada abrangente e abarca também outras perspectivas ilustradas neste trabalho. Neste caso o âmbito da integração está ao nível dos próprios sistemas aplicacionais que necessitam de interagir ou incorporar componentes e informação.

Uma aplicação pode ser vista como uma combinação de componentes complementares e para cada uma delas existem diferentes formas de integração. Neste caso, o âmbito centra-se na própria estrutura das aplicações onde se pode identificar 3 áreas passíveis de intervenção:

- A camada de apresentação
- A lógica aplicacional
- A informação

Em cada um destes pontos as aplicações podem necessitar de integração para comunicarem, partilharem e integrarem funcionalidades de outros sistemas aplicacionais. Neste sentido, qualquer abordagem para a integração de SI vai depender da forma como as aplicações estão construídas e em que plataformas estão baseadas. Ao longo dos tempos, passou-se de um desenvolvimento em linguagens, por exemplo com o COBOL, para o paradigma dos objectos dominado pela linguagem C++ e as normas CORBA (*Common Object Request Broker Architecture*) [Zahavi 1999, Serain 2002] e COM (*Component Object Model*) [Serain 2002]. Com o surgimento das arquitecturas em n camadas, e da Internet, outras soluções de computação distribuída [Coulouris et al. 2000] dominaram a forma de implementação das aplicações. Resumindo as abordagens mais comuns para este tipo de integração pode-se identificar as seguintes áreas:

- Codificação específica (ex: COBOL, C++ ou Java)
- *Application Program Interface (API)*
- *Remote Procedure Calls*
- *Transacções distribuídas: TP monitors*
- *CORBA, Java RMI e DCOM*
- *Messaging Middleware*
- *Web Services*

Com a evolução das normas e da tecnologia surgiram duas correntes paralelas onde se especificaram plataformas completas de desenvolvimento e disponibilização de aplicações: o J2EE e o .NET. Nelas existem funcionalidades para a integração de aplicações para cada uma das camadas estruturantes. A visão deste tipo de integração

pode ainda mais ser mais alargada, quer no tipo de soluções, quer no seu âmbito, e neste sentido é possível distinguir a integração de aplicações dentro de uma organização e aquela que é implementada entre duas ou mais organizações. Neste último caso, está-se no âmbito das soluções do tipo Business-to-Business Integration (B2Bi) que permitem definir formas de comunicação partilhando informação e aplicações.

3.2.1 Integração aplicacional dentro da organização

A integração aplicacional centra-se nas soluções que permitem a interligação de aplicações para poder partilhar informação e a sua lógica aplicacional. No mercado das TI existem várias ofertas tecnológicas que permitem centralizar a integração de SI nas suas diversas facetas.

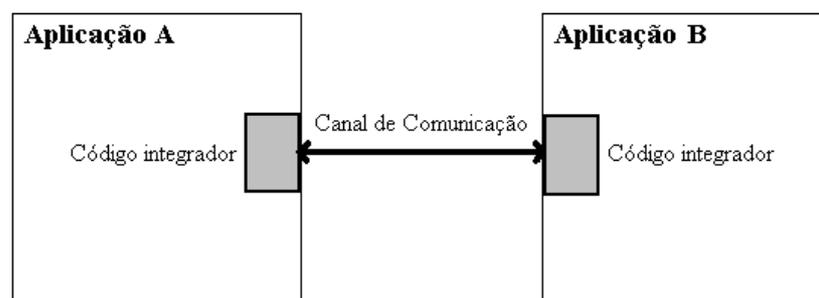


Figura 3.11: Integração aplicacional directa

Uma primeira abordagem para integrar dois sistemas heterogéneos é a sua integração directa (Figura 3.11). Esta solução resolve um problema específico mas pode ter custos elevados e não se revelar suficientemente flexível e adaptável no futuro. No entanto é uma alternativa útil quando se procura aproveitar, de forma simples e directa, as potencialidades de determinadas funcionalidades incorporando-as numa aplicação. Este tipo de interligação pode ser implementado de várias formas que passam pela simples troca assíncrona de ficheiros, pela criação de um canal de comunicação síncrono e dedicado com *sockets* e respectivo código até à incorporação directa de lógica aplicacional externa. Esta última situação pode ser conseguida com os JavaBeans, CORBA ou módulos aplicacionais. Neste caso, o objectivo é o de

partilhar informação e funcionalidades incorporando-as através de uma ligação restrita e dedicada.

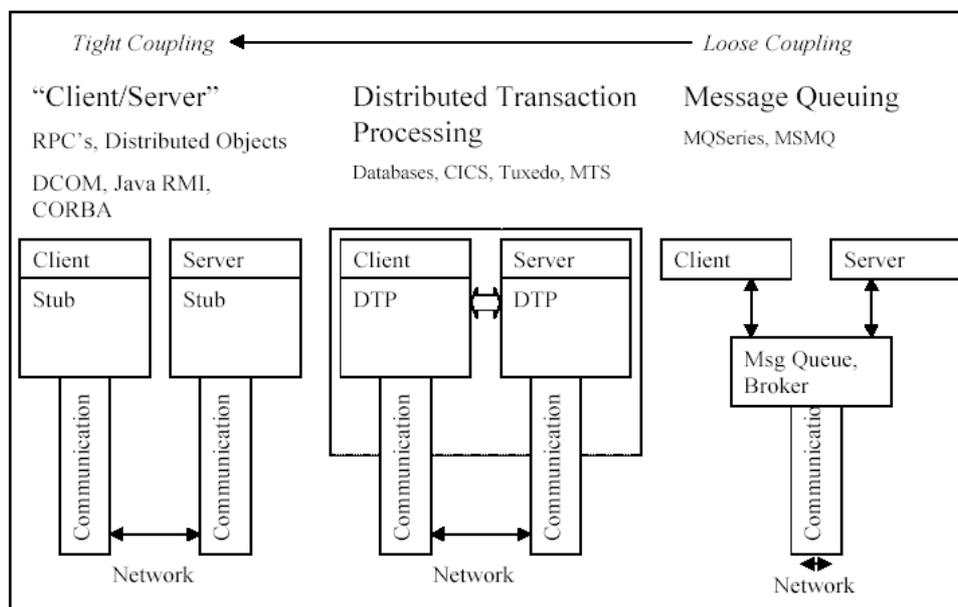


Figura 3.12: Modelos aplicacionais (adaptado de [Lewis 2000])

Mas a forma de conceber aplicações evoluiu com os tempos e criou modelos aplicacionais diferentes. Cada abordagem vai influenciar a proximidade das aplicações determinando diferentes caminhos para a sua interligação. Pode-se identificar pelo menos três situações [Lewis 2000] (Figura 3.12):

1. Cliente/Servidor
2. Transacções distribuídas
3. Messaging

A variação depende do nível de proximidade, ou acoplamento, que vai desde as situações de “acoplamento forte” (*tight coupling*) até às de “acoplamento fraco” (*loose coupling*). Em cada uma das situações haverá tecnologias e abordagens diferentes no que diz respeito à integração aplicacional. Para cada caso existem alternativas de comunicação e partilha de informação e procedimentos (Quadro 3.1).

Tipo	Soluções
Cliente / Servidor	<ul style="list-style-type: none"> • Application Program Interface (API) • Remote Procedure Call (RPC) • Objectos Distribuidos (DCOM, CORBA, Java RMI)
Transacções Distribuidas	<ul style="list-style-type: none"> • Bases de Dados • Monitores Transaccionais
<i>Message Queueing</i>	<ul style="list-style-type: none"> • Message Brokers

Quadro 3.1: Soluções para os modelos aplicacionais

No caso da arquitectura Cliente/Servidor é possível colocar dois sistemas a comunicar directamente com a invocação de funções ou procedimentos remotos (RPC) [Rao 1995, Serain 2002] disponibilizados por interfaces aplicacionais (API) [Hines 1996]. Trata-se de um canal de comunicação síncrono e directo onde cada invocação espera uma resposta. Esta realidade assume que existem sistemas servidores dedicados a responder às solicitações dos sistemas clientes. Este tipo de integração está no nível da lógica funcional das aplicações (Figura 3.13). Para além dos RPC e das API pode-se tirar partido da tecnologia dos objectos distribuídos (DCOM, CORBA, EJB e JavaRMI) que se enquadram geralmente neste tipo de abordagem.

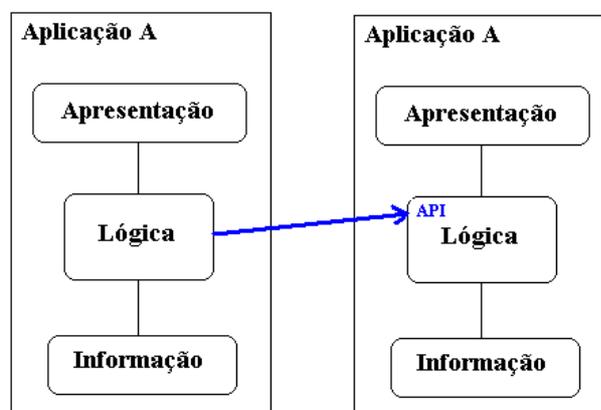


Figura 3.13: Interface Aplicacional

Nos casos em que é necessário controlar transacções aplicacionais, que actualizam informação operacional, pode-se recorrer aos SGBD ou a soluções específicas como são por exemplo os monitores transaccionais. Os SGBD suportam a partilha assíncrona de informação, o acesso a procedimentos armazenados contendo lógica aplicacional. Por seu turno, os monitores transaccionais controlam a execução de transacções aplicacionais permitindo uma integração efectiva entre aplicações. Este tipo de gestão de transacções permite, por exemplo, executar lógica aplicacional remota, controlar transacções distribuídas entre sistemas heterogéneos, tratar eventos despoletando a respectiva sequência de tarefas associadas, aceder e actualizar informação nos SGBD ou controlar a comunicação com as camadas de apresentação das aplicações.

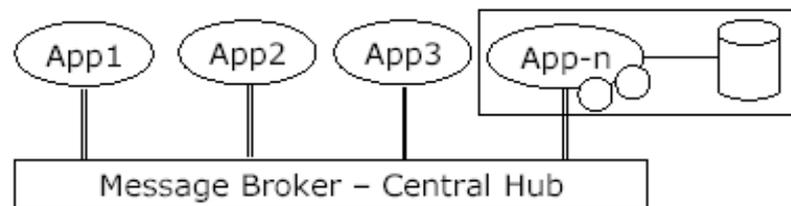


Figura 3.14: Message Broker

As situações em que as aplicações estão num ambiente *loose coupling* a integração pode ter como base a tecnológica o *Messaging* que se encontra nas soluções do tipo *Message-Oriented Middleware* (MOM) ou nas soluções proprietárias do tipo *Message Broker* (MB) (Figura 3.14). Os MB comerciais disponibilizam normalmente um conjunto de adaptadores específicos para aplicações e protocolos de comunicação. Estas soluções operam como uma plataforma de troca de mensagens entre aplicações e são elas próprias denominadas de “plataformas EAI”. Os MOM gerem as filas de mensagens (*Message Queues*) para a troca e o transporte de informação entre as aplicações. As plataformas MOM recebem as diferentes mensagens de diferentes aplicações, transformando o seu conteúdo e mandando-as para as aplicações de destino. Estas soluções têm três tipos de intuitos:

- Desenvolvimento e disponibilização de adaptadores aplicacionais.
- MOM para propagação de mensagens e informação.
- Gestão de regras de encaminhamento e transformação da informação.

Baseado em protocolos de comunicação em rede, as soluções do tipo MOM permitem diferentes formas de comunicação baseadas em mensagens. A informação é encapsulada em mensagens estruturadas que são enviadas de duas formas: síncrona ou assíncrona [Steinke 1995]. Neste âmbito pode-se identificar pelo menos três padrões de comunicação [Hohpe et al. 2003]:

1. *Peer to peer* ou *Point to point*
2. *Hub and Spoke*
3. *Bus* ou *Pipeline* ou *Publish and Subscribe*

A comunicação *Point to Point* permite criar um canal de comunicação assíncrono e específico entre duas aplicações para a troca de mensagens (Figura 3.15).

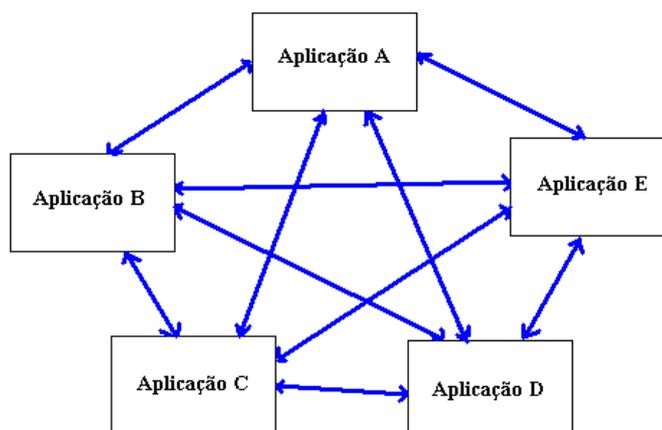


Figura 3.15: Comunicação *point to point*

A topologia *Hub and spoke* centraliza a comunicação num canal de comunicação assíncrono e comum (*Hub*) para troca de informação (Figura 3.16). Esta solução é também frequentemente designada por *EAI Middleware*.

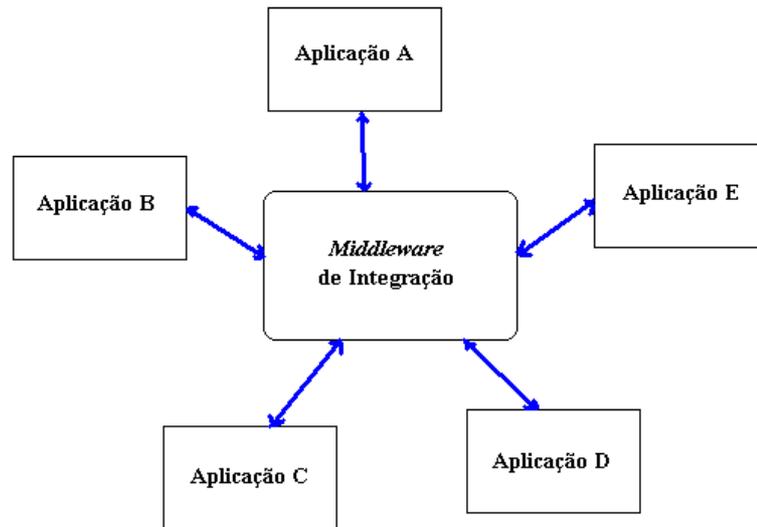


Figura 3.16: Comunicação *hub and spoke*

Na comunicação do tipo Publish/Subscribe são difundidas mensagens para um ou mais destinos que as reconhecem subscrevendo a sua recepção (Figura 3.17). A canal de comunicação é denominado de *Bus* permitindo uma comunicação assíncrona de sistemas federados.

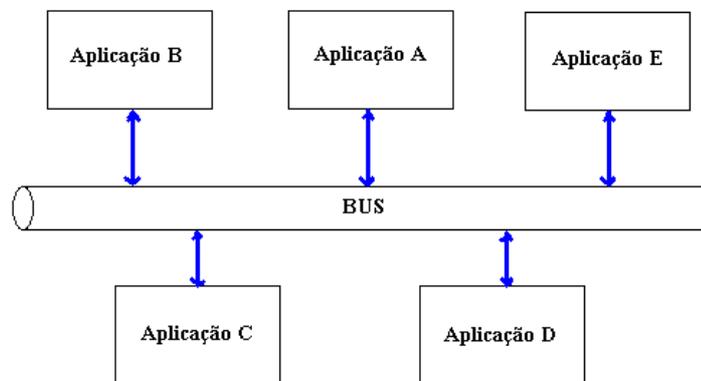


Figura 3.17: Comunicação do tipo *Bus*

3.2.1.1 Plataformas aplicacionais: .NET e J2EE

Ao longo dos tempos o desenvolvimento de aplicações foi acompanhando a tecnologia e a sua evolução. Qualquer aplicação está normalmente alicerçada numa base tecnológica que suporta e executa o seu código. Neste sentido existem plataformas aplicacionais que fornecem um conjunto de especificações, directrizes, suporte tecnológico e ferramentas de análise e de desenvolvimento. Estes ambientes

permitem seguir todo o ciclo de vida de desenvolvimento aplicacional nas suas fases de análise, concepção, implementação e disponibilização. Em complemento, são disponibilizadas funcionalidades de segurança, integração, *performance* e fiabilidade que garantem o bom funcionamento de aplicações distribuídas e assentes em *n* camadas.

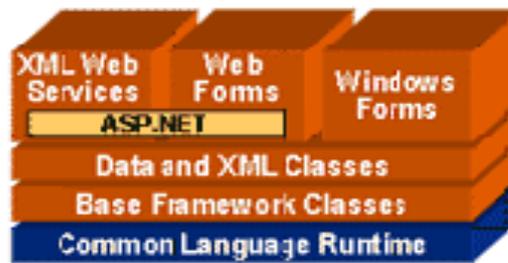


Figura 3.18: Plataforma Microsoft .NET

Neste âmbito as duas plataformas aplicacionais mais comuns são: .NET e J2EE. A primeira assenta numa norma da Microsoft e a segunda numa norma da SUN baseada na linguagem Java. Cada uma está organizada por camadas tecnológicas (Figuras 3.18 e 3.19) que garantem toda o conjunto de funcionalidades desde a gestão do acesso à informação, passando pelas comunicações até à camada de apresentação das aplicações. Cada camada permite várias formas de integração, referidas nos pontos anteriores, disponibilizando-as em pleno ambiente de desenvolvimento aplicacional. Estas plataformas permitem suportar arquitecturas do tipo Cliente/Servidor ou baseadas na *Web*.

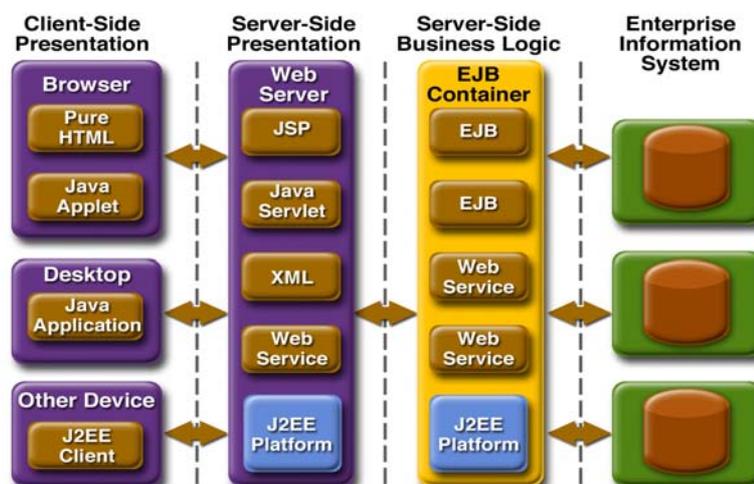


Figura 3.19: Plataforma J2EE

A plataforma .NET fornece um conjunto de bibliotecas de software, ferramentas de desenvolvimento, especificações para codificação, uma área comum de execução de código e várias alternativas de linguagens de programação. A plataforma J2EE fornece um conjunto completo de especificações que ilustra como se deve implementar cada tipo de funcionalidade com a tecnologia Java. Esta plataforma fornece uma área de trabalho completa que abrange as áreas de concepção, desenvolvimento, assemblagem, execução e disponibilização aplicacional. As aplicações resultantes, concebidas em Java, estão baseadas num modelo aplicacional distribuído por n camadas.

Uma plataforma aplicacional disponibiliza essencialmente as seguintes funções:

- Segurança e Autenticação
- Capacidade de crescimento e tolerância a falhas
- Gestão de transacções
- Serviços de directório e gestão de sessões
- Integração aplicacional
- Administração
- Execução aplicacional (*Runtime*)
- Messaging
- Gestão de conexões e comunicações
- *Reporting e Business Intelligence*
- Desenvolvimento, disponibilização e execução aplicacional
- *Web Services*
- Business Process Management e Workflow
- Acessos diferenciados (*Web Browser, Wireless, etc.*)

Como se pode ver no quadro comparativo (Quadro 3.2), é possível fazer algumas comparações simples entre o .NET e o J2EE de forma ter uma visão mais clara sobre algumas diferenças básicas entre estas duas normas.

	J2EE	.NET
Ferramentas de desenvolvimento	- Várias no mercado compatíveis	- Visual Studio.NET
Servidores Aplicacionais	- Oracle, SUN, IBM, BEA	- Microsoft
Sistemas Operativos	- Windows, Unix, Linux	- Windows
Suporte para: Web Services	- XML Messaging (JAXM) - XML Processing (JAXP) - XML Registries (JAXR) - XML based RPC (JAX-RPC) - XML Data Binding (JAXB) - SAAJ Java SOAP API - JSR 109 WS Deployment Model	- SOAP - WSDL - UDDI
Implementação: Web Services	- Componentes EJB - Aplicação Java	- .NET Classes - COM/COM+

Quadro 3.2: Comparação simples .NET e J2EE

A grande maioria dos servidores aplicativos do mercado, assim como aplicações comerciais, estão baseados numa destas duas plataformas.

3.2.1.2 Servidores aplicativos e Integration Brokers

Muitas organizações adoptaram soluções classificadas como *Application Servers* (AS) que incorporaram a grande maioria das alternativas de integração referidas neste capítulo. Um AS é um *software* baseado em componentes, que se situa conceptualmente na camada de *middle-tier* da arquitectura *n-tier* ilustrada no capítulo anterior. Os AS incorporam funcionalidades da camada de *Middleware* (Figura 3.20) e permitem suportar aplicações, disponibilizá-las para a camada cliente, trocar informação entre elas, partilhar lógica aplicacional e integrar funcionalidades.

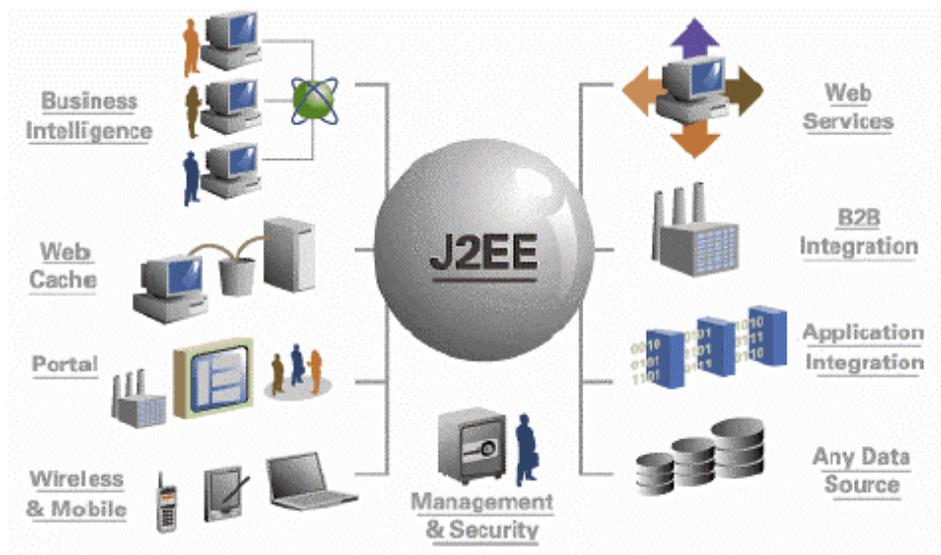


Figura 3.20: Funcionalidades do Oracle Application Server baseado em J2EE

Os AS fornecem uma plataforma de desenvolvimento e disponibilização de serviços que permite aos programadores concentrar-se nas regras da organização, em vez de investir tempo na codificação de funcionalidades básicas e complexas para a camada de *middleware*. Estas soluções disponibilizam normalmente ferramentas de desenvolvimento, serviços de execução e disponibilização de aplicações, funcionalidades para integração de SI e de autenticação e segurança. Os AS têm normalmente três grupos de funcionalidades (Figura 3.21) para o desenvolvimento aplicacional:

1. Ferramentas integradas de desenvolvimento de código.
2. Plataforma de suporte, interpretação, validação e execução do código.
3. Ambiente de *deployment* que armazena, executa e disponibiliza aplicações.

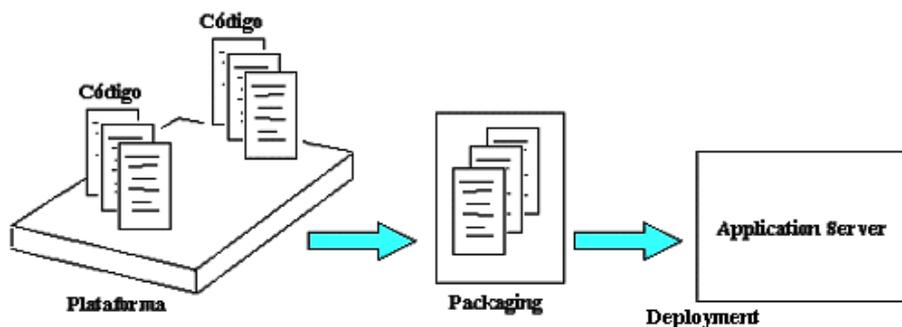


Figura 3.21: *Deployment* de código num Application Server

Os AS permitem suportar plataformas aplicacionais com capacidades de resposta e de crescimento fiáveis, permitindo disponibilizar diferentes pontos de acesso para as aplicações. Para além destas características, estes sistemas incorporam funcionalidades de acesso a diferentes fontes de informação, gestão de segurança e de integração baseadas num canal de comunicação central chamado *Hub*, ou *Bus*, que suportam a tecnologia de *Messaging* já referida.

Alguns servidores aplicacionais disponibilizam componentes de ligação a aplicações específicas denominados por adaptadores ou conectores. Este tipo de software é normalmente proprietário e específico, mas surgem também componentes baseados em normas de mercado como são os *Web Services*. Para além dos adaptadores são também disponibilizadas funcionalidades de automatização de processos organizacionais. Estas soluções são também denominadas por *Integration Brokers* (IB) ou por *Integration Middleware*. Ainda neste âmbito existem soluções isoladas classificadas como *Gateways*, com várias configurações, e que servem exclusivamente para a integração aplicacional ao nível das suas transacções. Estas soluções funcionam como uma ponte, entre dois ou mais sistemas, proporcionando-lhes o controlo da interacção ao nível da informação e dos procedimentos. Através deste tipo *software* centralizado as aplicações conseguem trocar informação e invocar procedimentos entre elas.

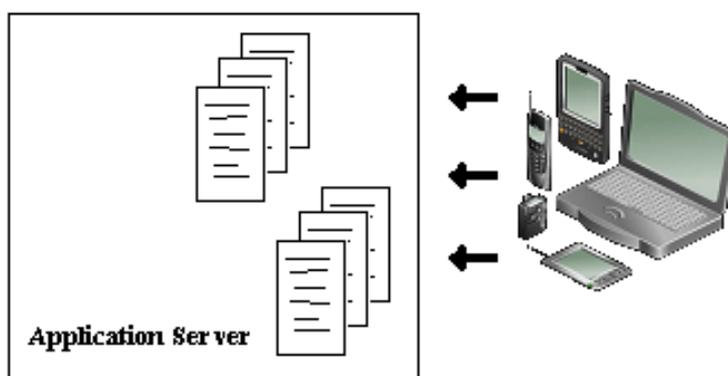


Figura 3.22: Acessos diferenciados às aplicações

Os AS normalmente também permitem a integração de SI ao nível do *User Interface*. Dentro de uma organização, as Intranets permitem incorporar informação e aplicações para as disponibilizar através de um único ponto de acesso através, por

exemplo, de um *web browser*. Alguns AS podem ser utilizados como servidores *Web* tendo mesmo funcionalidades para a criação de portais internos, ou externos, que possibilitam a gestão de conteúdos do tipo ECM ou EII já referidos. Este tipo de integração recorre a componentes do tipo *portlet* [Hepper 2003] que funcionam como pequenos contentores dinâmicos de informação, lógica aplicacional ou aplicações. Muitos destes AS já estão preparados para permitir acessos diferenciados à mesma fonte de informação (Figura 3.22). Com a evolução da tecnologia, os fornecedores de AS incorporaram também funcionalidades de gestão processual que permitem automatizar processos organizacionais mais complexos.

Os AS são hoje uma excelente solução para a integração de aplicações devido ao conjunto rico de funcionalidades que oferecem. A adopção de servidores aplicacionais permite criar uma infraestrutura de integração completa onde se pode criar regras de transformação e troca de informação, integrar aplicações e centralizar o seu acesso.

3.2.2 Integração aplicacional entre organizações

Este tipo de integração é normalmente designado por Business-to-Business Integration (B2Bi) [Bussler 2003]. Trata-se de coordenar a comunicação entre duas ou mais organizações ao nível dos seus SI para que esta seja efectiva e segura. Este tipo de coordenação faz sentido nos relacionamentos existentes entre parceiros comerciais ou do tipo cliente/fornecedor. Este tipo de solução enquadra-se ao nível da dinâmica organizacional no que diz respeito às trocas comerciais, coordenação processual e suporte à cadeia de valor associada às organizações envolvidas. Desta forma é possível aumentar a eficiência dos processos organizacionais, reduzir custos operacionais, otimizar e automatizar a troca de informação.

As soluções do tipo B2Bi devem garantir um conjunto de funcionalidades básicas como são:

- Garantir o suporte às transacções específicas entre as organizações
- Automatizar a troca de dados entre sistemas heterogéneos

- Manter informação histórica dos processos efectuados
- Segurança na troca de dados
- Suportar diferentes conjuntos de formatos de ficheiros, normas e protocolos de comunicação e segurança
- Capacidade de crescimento, balanceamento de carga e tolerância a falhas
- Disponibilizar *reporting* para informação de gestão

A primeira abordagem para o B2B foi a solução EDI (*Electronic Data InterChange*) que se caracteriza por um conjunto de formatos de mensagens estruturadas e a especificação dos seus elementos para permitir a troca electrónica de informação entre empresas ou organizações. Trata-se do intercâmbio de informação entre parceiros autónomos que se associam e definem todos os tipos de documentos que vão trocar electronicamente e formatados segundo normas previamente acordadas. Reconhecido pela ANSI (*American National Standards Institute*), o comité ASCX12 (*Accredited Standards Committee X12*) especificou desde os anos 70 mais de 315 conjuntos de transacções EDI para as diversas situações de troca de documentos entre organizações. Na Europa o directório de transacções EDI UN/EDIFACT (Figura 3.23) abrange as áreas dos transportes, comércio e administração. Com o aparecimento da Internet surge o conceito de EDIINT (*EDI-Internet Integration*) que abrange as especificações para usar EDI sobre a Internet. Neste sentido a comunidade técnica IETF (*Internet Engineering Task Force*) desenvolve esforços onde por exemplo foi definido o RFC 1767 que especifica uma forma de encapsulamento de mensagens EDI para ser enviadas através dos protocolos SMTP e HTTP.

```

UMB+UM0A:1+XXXXX SECURITY:XY+5013546005111:ZZ+971014:1544+600000641
++ORDERS'
UMH+60000000000992+ORDERS:90:1:UN'
EGM+105:ORDERS+16849+971014:1544+00'
RFF+CT+16849REF'
NAD+EY+XXXXX SECURITY:92'
NAD+SE+0033454140:92'
NAD+EY+002440:92'CUX+GEP:IN'
UNS+D'
LIN+1++135-804-
086:BP+CM316X7R474K16&T:M6+21:9000:PK+0.053:CT:1:PK'
PIA+4+AUX:MF'
SCC+1'
QTY+21:9000:PK'
DTM+002+971015'
SCC+1'QTY+21:2000:PK'
DTM+002+971016'
LIN+2++222-804-
086:BP+GM316X7R474K16&T:M6+21:9000:PK+0.053:CT:1:PK'
PIA+4+AUX:MF'
SCC+1'
QTY+21:7000:PK'
DTM+002+971017'
SCC+1'
QTY+21:8000:PK'
DTM+002+971018'
UNS+S'
UNT+25+60000000000992'
UNZ+1+600000641'

```

Figura 3.23: Exemplo de um documento EDIFACT

Após um primeiro esforço para juntar o EDI e o XML com a definição XML/EDI, o comitê técnico OASIS (*Organization for the Advancement of Structured Information Systems*) e a *United Nations/ECE agency* apresentaram o ebXML (*e-business XML*). Trata-se de um método normalizado em XML para registrar e implementar a integração de processos organizacionais, a troca de mensagens e a comunicação de informação entre organizações.

O consórcio RosettaNet promove e implementa um conjunto de especificações para suportar o B2B designado por RNIF (*RosettaNet Implementation Framework*). Este pacote de normas especifica a forma de transporte, encaminhamento e estruturação de mensagens PIP (*Partner Interface Processes*) e eventos de negócio. O nome RosettaNet deriva da conhecida Pedra de Rosetta que tem gravada a mesma mensagem em três línguas e permitiu decifrar os hieróglifos egípcios. Tal como a

pedra esta organização definiu um conjunto de mensagens e procedimentos que permitem às empresas automatizar os seus processos e integrar os seus negócios.

A normalização da troca electrónica de dados entre organizações é claramente uma área que requer soluções para a integração dos SI envolvidos. Existem várias abordagens que se suportam em tecnologias e normas específicas referidas nos pontos anteriores. O XML e os *Web Services* permitem disponibilizar um novo nível de integração nesta área e são hoje em dia normas incontornáveis.

A área da integração de aplicação, dentro e fora das organizações, tem várias soluções que dependem da natureza da construção dos sistemas aplicativos e das próprias organizações. A evolução das TI e das normas acompanham este tipo de necessidades e tem vindo a reforçar as ofertas de alternativas em cada uma das áreas e camadas tecnológicas. Existem plataformas aplicativos completas e com provas dadas no mercado que asseguram o suporte a qualquer tipo de integração necessário. As normas que permitem a portabilidade da informação e uma independência tecnológica, como são os *Web Services* e o XML, têm tido um papel cada vez mais preponderante.

3.3 A Integração dos processos organizacionais

Esta área engloba um conjunto de metodologias e *software* que permite às organizações conceber e especificar detalhadamente processos de negócio tendo em vista a sua optimização e automatização. A concepção destes fluxos automáticos de informação e tarefas exige uma análise muito profunda de cada etapa que estrutura os processos. Estas etapas ou tarefas correspondem normalmente a actividades comuns do negócio como são, por exemplo, as actualizações de informação de clientes, verificações de existência de mercadorias, as aprovações de requisições ou despesas, a recepção de encomendas, etc... A integração de SI centrada nos processos organizacionais procura optimizá-los tornando as organizações mais flexíveis e eficientes. Os processos correspondem à dinâmica organizacional e estruturam a sua actividade diária. A automatização desses processos permite melhorar a eficiência da

organização e facilitar a obtenção de informação de gestão que permite suportar as tomadas de decisão ao nível da administração.

3.3.1 *Workflow e Business Process Management*

Nesta área, as organizações procuram sistemas de automatização de processos que se enquadram nas soluções do tipo ou *Workflow* [Plesums 2002] [Fischer 2003] ou *Business Process Management (BPM)* [Hollingsworth 2004] e que também permitem a integração de SI. Os sistemas de *Workflow* têm como principal objectivo solucionar problemas de controlo dos processos organizacionais. Para tal, podem ser utilizados diversos tipos de tecnologias, para estruturar as diversas etapas, e o trabalho associado. Trata-se da combinação de tarefas individuais numa sequência de actividades, com um objectivo comum, dando suporte aos processos organizacionais. O *Workflow Management Coalition (WfMC)* define o *Workflow* como sendo “a automatização de processos de negócio, no seu todo ou em parte, durante a qual são passados documentos, informação ou tarefas, de um participante para outro para actuar, de acordo com um conjunto de regras procedimentais”. Como se pode constatar, a questão da automatização de processos é claramente a peça central de orientação ao suporte dos fluxos de trabalho.

Processos repetitivos	Processos únicos
<p>Produção</p> <ul style="list-style-type: none"> - Aplicações do negócio - Aplicações para Seguros - Gestão de créditos bancários - ... 	<p>Colaboração</p> <ul style="list-style-type: none"> - Gestão documental de alto nível - Gestão de processos de colaboração - Criação de documentos técnicos - Desenvolvimento de s/w - ...
<p>Administrativo</p> <ul style="list-style-type: none"> - Aprovação de relatórios de despesas - Aprovação de compras - Gestão e aprovação de pedidos de férias - ... 	<p>Ad-hoc</p> <ul style="list-style-type: none"> - Encaminhamento informativo de documentos - Aprovações de pedidos ou requisições - ...

Quadro 3.3: Categorias de processos de *Workflow*

O *Workflow* é geralmente classificado segundo quatro categorias: **ad-hoc**, **colaboração**, **administrativo** e de **produção**. Estas categorias de processos têm em comum a sua importância dentro da organização, mas havendo grandes diferenças relacionadas com o nível de rigidez e estrutura nas definições dos processos. Um *Workflow* de produção tem como finalidade o suporte de processos organizacionais pré-definidos, executando a sequência de actividades de uma forma muito rígida e estruturada (Quadro 3.3). Por outro lado, um *Workflow* ad-hoc dá maior importância à partilha de informação pelos utilizadores, do que ao próprio processo que normalmente não é muito estruturado. Um *Workflow* administrativo tem a ver com processos “burocráticos” e repetitivos encontrados tipicamente nas organizações. Finalmente, um *Workflow* de colaboração caracteriza-se pela dinâmica e pela comunidade de participantes envolvidos no processo. Nestes casos, pode-se estabelecer várias iterações de uma mesma etapa até à chegada a um consenso por parte do grupo envolvido, e só depois passar à etapa seguinte. Os tipos de soluções nesta área são categorizados face ao tipo de processo e o valor acrescentado para a organização, e também face ao relacionamento da complexidade das tarefas com as suas estruturas (Figura 3.24).

A tecnologia de *Workflow* faz parte integrante do *Business Process Management* (BPM) que engloba de forma mais abrangente a classe de tecnologia ou aplicações para a automatização de processos organizacionais. As aplicações específicas de *Workflow* têm como principal objectivo o encaminhamento e a

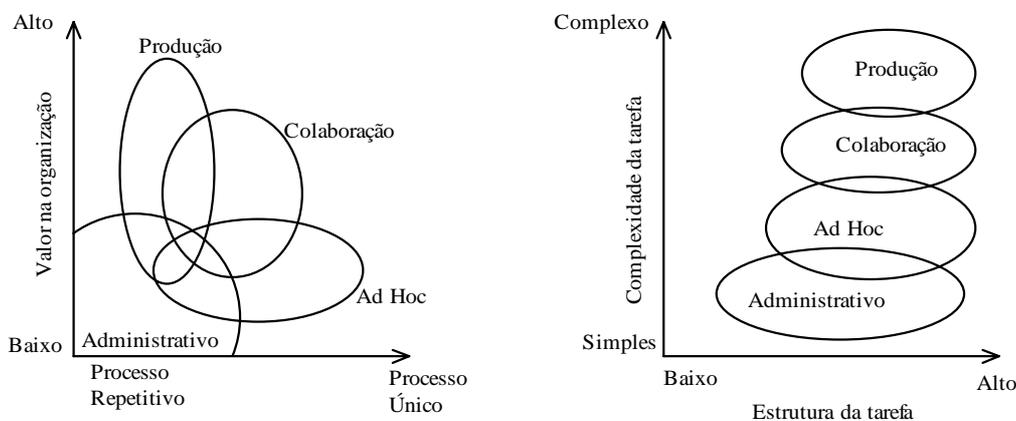


Figura 3.24: Caracterização das categorias de processos de *Workflow*

distribuição de informação e tarefas que envolvem frequentemente pessoas. Para além deste objectivo as soluções de BPM gerem efectivamente, a um nível superior, o processo organizacional como um todo aplicando-lhe regras de negócio, integrando outras aplicações e obtendo as respectivas métricas globais de gestão. O BPM tem cada vez maior importância devido à convergência de diferentes factores. A crescente necessidade da integração funcional das organizações, a maior abrangência da automatização de processos organizacionais e a procura de uma melhor agilidade organizacional estão na origem da procura de soluções de BPM.

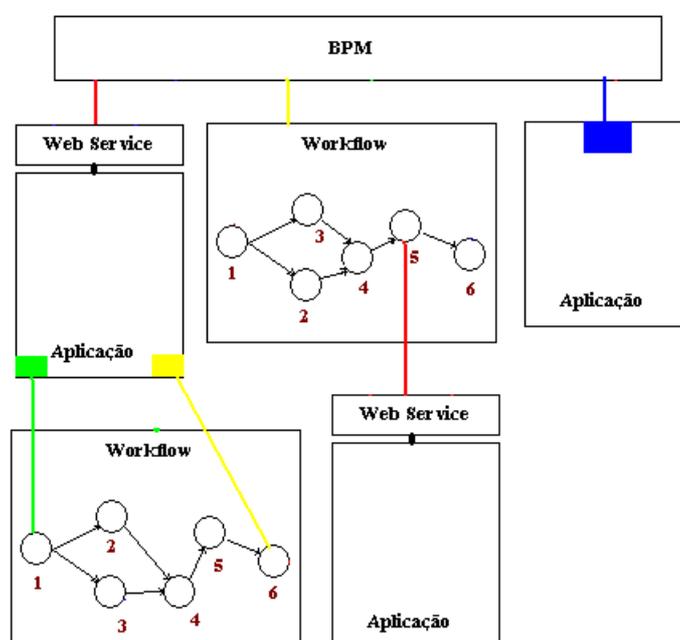


Figura 3.25: BPM para a integração de SI.

As soluções BPM, também denominadas de POM (*Process Oriented Middleware*), permitem coordenar numa lógica processual vários tipos de aplicações de forma a automatizar um processo (Figura 3.25). A coordenação e sincronização de diferentes processos de *Workflow* poderá por exemplo automatizar um processo organizacional de forma completa, quer dentro das organizações ou entre estas. As soluções BPM caracterizam-se por terem várias funcionalidades complementares e que abragem toda a organização a nível tecnológico e de gestão. Pode-se identificar os seguintes grupos de funcionalidades complementares:

1. Gestão de processos *end-to-end*
2. Modelação e documentação dos processos
3. Integração automática ou manual de processos
4. Portabilidade e acessibilidade aos processos
5. Capacidade de crescimento e abrangência da organização
6. Ferramentas de análise e monitorização

As soluções BPM permitem criar arquitecturas funcionais para a integração de sistemas, e são modeladas de acordo com as regras de funcionamento da organização.

3.3.2 Business Process Execution Language

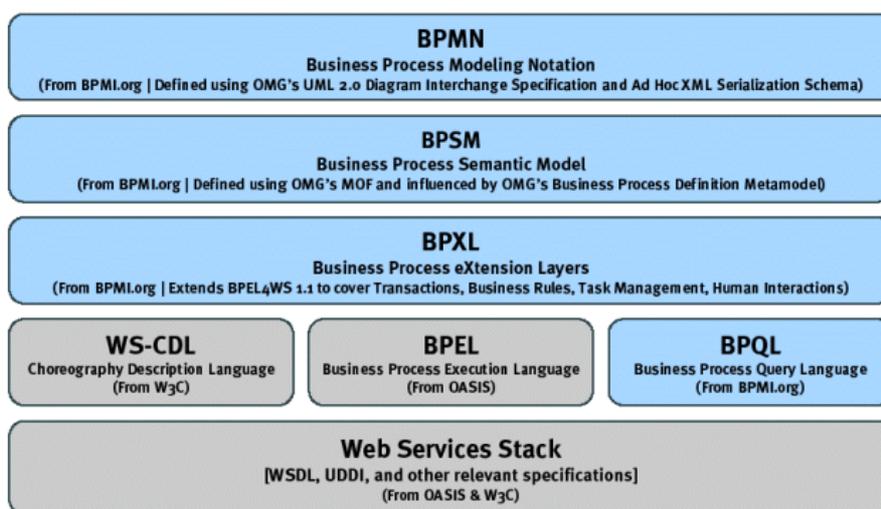


Figura 3.26: Enquadramento das normas BPMN, BPML, BPEL e *web services* (adaptado de BPML.org)

Na área do BPM existem diferentes organismos que especificam e defendem várias normas de mercado (Figura 3.26). Algumas dessas normas complementam-se e permitem criar novas normas mais abrangentes. A evolução destas especificações tem seguido caminhos de convergência que procuram tirar partido da tecnologia dos *web services* criando uma zona nublosa mas complementar com o BPM.

Ao longo dos tempos surgiram várias normas (Figura 3.27) com diferentes propósitos. O WfMC especificou o WPDL (*Workflow Process Definition Language*) e o XPDL (*XML Process Definition Language*) para a definição de processos de *workflow*. O OMG definiu uma extensão ao UML (*Unified Modeling Language*) para a modelação de processos de *workflow* e a proposta BOM/99-10-03 para a definição de processos. O NIST/MIT (*National Institute for Standards and Technologies/MIT*) definiu o PSL (*Process Specification Language*) e o PIF (*Process Interchange Format*). O BPMI.org (*Business Process Management Initiative*) propõe entre outros o BPML (*Business Process Modeling Language*), o BPMN (*Business Process Modeling Notation*) e o BPQL (*Business Process Query Language*). Na iniciativa ebXML.org (*Electronic Business XML*) sugere o BPSS (*Business Process Schedule Specification*) e a OASIS por seu lado definiu o BTP (*Business Transaction Protocol*). Em colaboração com a HP o W3C surge com o WSCL (*Web Services Conversation Language*), em colaboração com a SUN, a BEA e a Intalio definiram o WSCI (*Web Services Choreography Interface*) e mais tarde especifica a norma WS-Choreography.

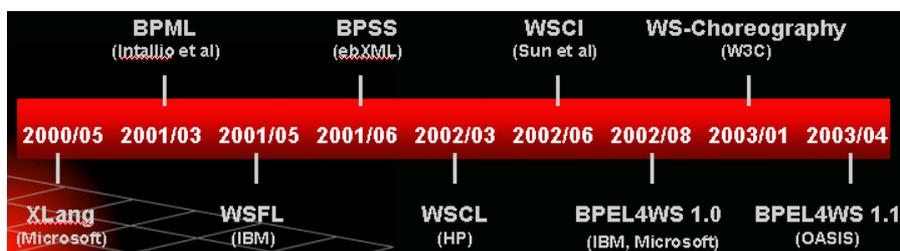


Figura 3.27: Cronologia das normas de orquestração de *web services* (fonte Oracle)

A IBM também participa nesta área com o WSFL (*Web Services Flow Language*) e a Microsoft na mesma área com o XLANG que é baseado no WSDL dos *web services*. A IBM e a Microsoft juntaram esforços para fundir estas especificações numa mais recente chamada BPEL4WS 1.0 (*Business Process Execution Language for Web Services*) ou simplesmente BPEL. Esta convergência incorporou as especificações *WS-Coordination*, *WS-Transaction* e *WS-Security* em resposta à especificação BPML mais abrangente. A agência DARPA (*Defense Advanced Research Projects Agency*) definiu o DAML-S (*DARPA Agent Markup Language - Services*) para a automação de *web services*.

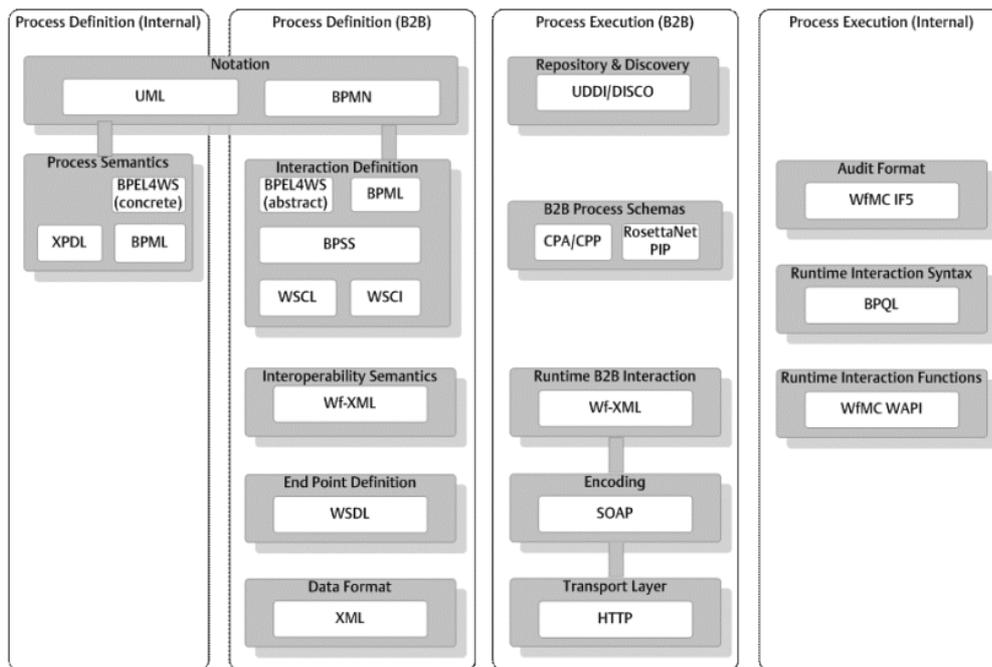


Figura 3.28: Diagrama de normas da WfMC e BPMI.org

Assistindo a uma convergência nesta área, o WfMC e o BPMI.org relacionaram conceptualmente as normas dividindo em áreas de definição e execução internas e externas dos processos incluindo o B2B (Figura 3.28). Centrado o enquadramento nas funcionalidades específicas da definição e execução de processos, as áreas mais relevantes relacionam-se com a definição de processos, a coreografia que permite integrar os *web services*, a execução/interacção dos processos e os próprios *web services*:

- Definição de processos
 - XPDL, BPML e BPEL
- Coreografia de processos
 - WSCI e BPEL
- Execução/Interacção
 - Wf-XML e BPEL
- *Web Services*
 - SOAP, WSDL e UDDI



Figura 3.29: Stack tecnológica do BPEL

Para além de ser uma das normas mais recentes, e baseada numa especificação da OASIS, o BPEL (Figura 3.29) é suportado pela IBM, Microsoft, Oracle, BEA, SAP e Siebel cujas soluções e clientes correspondem praticamente à totalidade do mercado nesta área. Esta realidade levou a Oracle Corporation a comprar a empresa Collaxa, que estava a liderar o mercado nesta área, integrando a sua tecnologia. Esta norma baseada no XML e nos *web services* permite definir a orquestração dos serviços dentro de uma lógica processual e permite a integração ponto a ponto de sistemas ou aplicações. São identificadas as etapas de um processo, a sua interacção com outros sistemas e toda a lógica da sua execução (Figura 3.30). Esta especificação requer um sistema que permita executar os processos definidos por esta via.

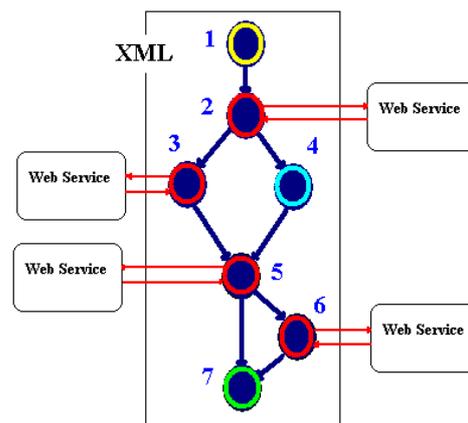


Figura 3.30: BPEL

Numa lógica de continuidade a BEA e IBM juntaram esforços e definiram o BPELJ (BPEL *for Java*). O BPML e o BPEL partilham as mesmas bases quanto aos *web services* (SOAP, WSDL e UDDI), quanto ao XML (XPath e XSDL) e às

especificações WS-Security e WS-Transactions. No entanto o BPMI.org considera o BPEL como um subconjunto do BPML [BPMI 2002] devido ao facto deste contemplar não só a área da execução dos processos mas também a sua própria definição.

Com o objectivo de otimizar a sua dinâmica, as organizações procuram soluções integradoras que permitam automatizar os seus processos reduzir custos operacionais e agilizar os seus procedimentos. As soluções informáticas típicas de *workflow* e de gestão documental são os predecessores das soluções de BPM que abrangem tanto as interações humanas como as integrações com aplicações. Esta abordagem permite identificar e documentar os processos, automatizá-los e executá-los integrando-se com os sistemas existentes sem ter de alterá-los. Estas soluções são por vezes menos dispendiosas do que as abordagens orientadas à informação ou às aplicações que obrigam a outro tipo de implementação. As normas XML, *Web Services* e BPEL vieram dar um grande contributo para o sucesso deste tipo de perspectiva da integração de SI.

3.4 *Web Services e a arquitectura Service Oriented Architecture*

A evolução da tecnologia possibilitou às organizações ver os seus sistemas como componentes com os quais se podem interagir e integrar num processo organizado. É por exemplo possível criar sistemas que permitem iniciar um processo estruturado de uma encomenda numa organização, validar a transacção noutra organização, planear a entrega dos bens com outra organização e ter um suporte pós-venda interno. Os *Web Services* (WS) representam um conjunto estruturado de definições para a integração de diferentes tecnologias, desde o nível dos sistemas operativos, até aos processos organizacionais. Os *Web Services* compatibilizam tecnologias e soluções não integráveis à partida pela sua natureza, utilizando formatos comuns de troca de informação com o XML e aplicando regras de interacção. É assim possível evitar tarefas de reconstrução de sistemas obtendo a sua reutilização e respectiva integração.

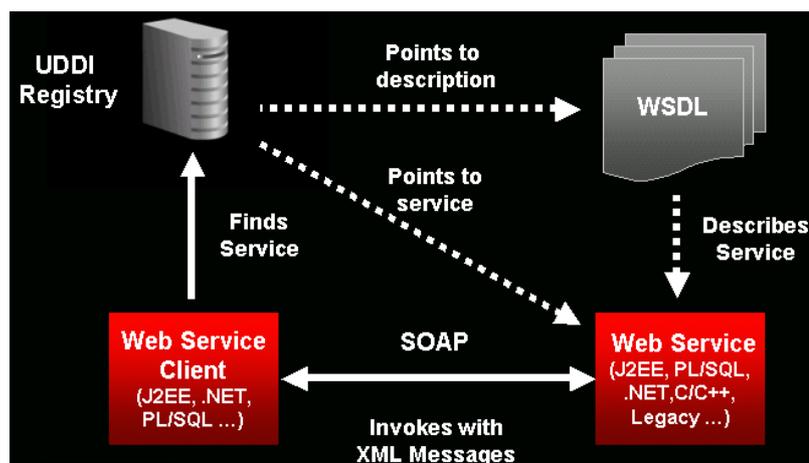


Figura 3.31: SOAP, UDDI e WSDL

O XML é a linguagem base que suporta as principais normas que estruturam os *Web Services* (Figura 3.31):

- SOAP - Simple Object Application Protocol
- WSDL - Web Service Description Language
- UDDI - Universal Definition Discovery Interface

Os *Web Services* funcionam como uma camada de abstracção de sistemas que disponibilizam informação ou procedimentos para os outros sistemas que os invocam (Quadro 3.4).

Funções	Web Service
Interacção	<ul style="list-style-type: none"> • Request/Response
Comunicação	<ul style="list-style-type: none"> • Comunicação síncrona do tipo RPC entre serviços <i>tightly-coupled</i>. • Comunicação assíncrona do tipo ficheiro/documento/mensagem entre serviços <i>loosely-coupled</i>.
Interface de descrição do serviço	<ul style="list-style-type: none"> • Norma WSDL
Protocolo e formato	<ul style="list-style-type: none"> • SOAP e XML
Identificação do serviço	<ul style="list-style-type: none"> • UDDI Registry

Quadro 3.4: Interacção com *Web Services*

O Web Services Interoperability Organization (WS-I) foi fundado em 2002 e tem como principal objectivo garantir a interoperabilidade entre as diferentes abordagens de arquitecturas assentes em Web Services. Esta organização não especifica normas técnicas nesta área mas coopera com outros organismos que o fazem como por exemplo a Organization for the Advancement of Structured Information Standards (OASIS), o World Wide Web Consortium (W3C) e a Internet Engineering Task Force (IETF). O WS-I publicou um conjunto de linhas orientadoras num documento denominado “the WS-I Basic Profile”. Esta publicação centra-se na orientação e recomendação acerca da estrutura das especificações globais dos Web Services (SOAP, WSDL, UDDI, XML e XML Schema) que devem ser definidas e agrupadas para garantir interoperabilidade entre os Web Services e as diferentes tecnologias que os suportam (Figura 3.32).

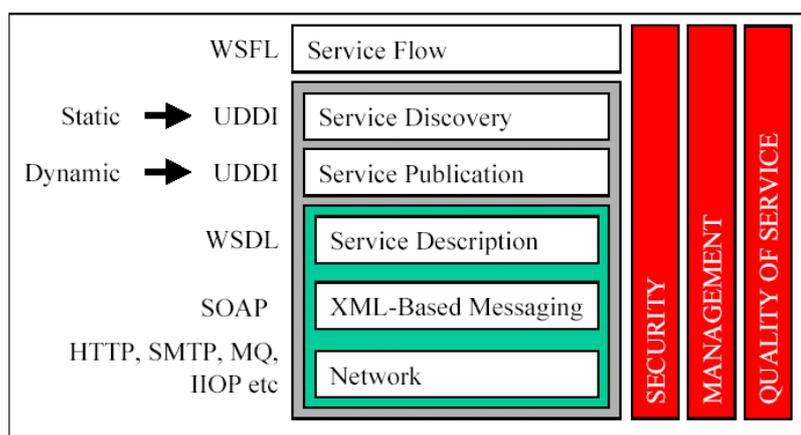


Figura 3.32: *web services stack* (Gottschalk *et al.*, 2002)

Para poder tirar partido de vários *Web Services* de forma combinada, foram definidas três normas para a sua gestão composta, criando uma plataforma denominada por *Web Services Composite Application Framework* (WS-CAF):

- *Web Service Context* (WS-CTX)
- *Web Services Coordination Framework* (WS-CF)
- *Web Services Transaction Management* (WS-TXM)

O WS-CTX disponibiliza um mecanismo para gerir informação comum aos diferentes serviços envolvidos e que deve ser encaminhada através de diversas

operações. Esta funcionalidade identifica um determinado contexto que permite correlacionar as diferentes mensagens XML que circulam entre os sistemas. O WS-CF faz a gestão do contexto referido e garante que as mensagens circulem de forma correcta e chegam aos diferentes pontos de destino. Por fim, o WS-TXM gere a gestão de transacções e a sua interacção com os respectivos sistemas transaccionais.

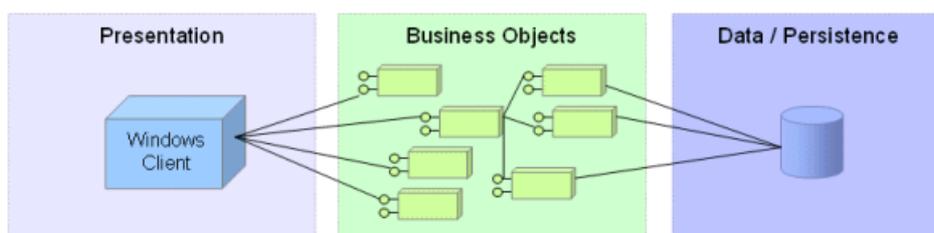


Figura 3.33: Arquitectura 3-tier

Uma integração assente em *web services* denomina-se por *Service Oriented Integration* (SOI) e neste âmbito é possível definir uma arquitectura orientada aos serviços *Service Oriented Architecture* (SOA) [Newcomer e Lomow 2004]. Não se trata de um conceito novo mas pode-se considerar que o SOA é uma evolução da arquitectura 3-tier (Figura 3.33) com a incorporação de serviços que disponibilizam e integram as aplicações de acordo com as suas especificações (Figura 3.34).

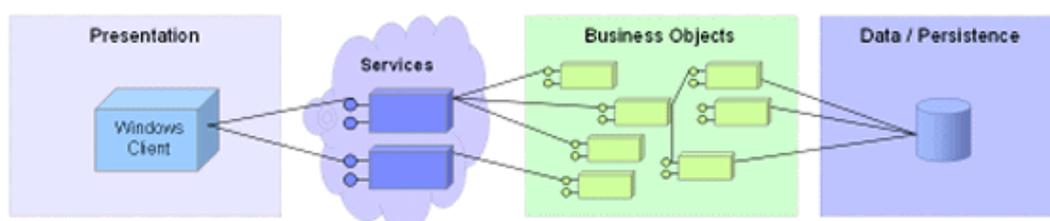


Figura 3.34: Arquitectura SOA

A arquitectura *Service Oriented Architecture* (SOA) permite às organizações disponibilizar as suas aplicações e soluções de *software* como serviços bem definidos, permitindo a sua reutilização e integrando-os em soluções mais abrangentes e flexíveis para o desempenho do negócio. Uma arquitectura SOA disponibiliza um repositório de serviços que podem ou não ser integrados numa solução aplicacional. Cada serviço representa um encapsulamento de componentes aplicacionais que contêm lógica aplicacional que segue as regras do negócio. Ainda no âmbito dos

modelos aplicativos, que vão desde o *tight coupling* até ao *loose coupling*, a arquitetura SOA é composta por camadas complementares (Figura 3.35).

Neste contexto surge o conceito de *Enterprise Service Bus* (ESB) baseado numa arquitetura que herda características dos *Message Brokers* e do SOA e que funciona como uma plataforma empresarial para implementar interfaces de comunicação baseadas em normas.

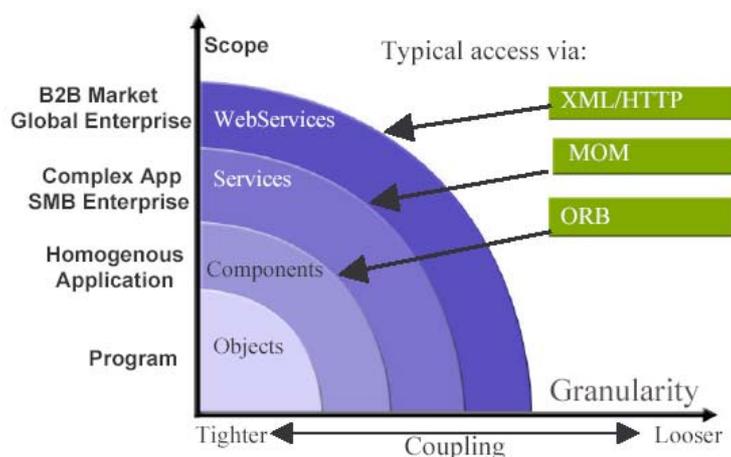


Figura 3.35: Camadas complementares do SOA.

O ESB representa uma espinha dorsal de serviços, mensagens, comunicações, transformações e de segurança sobre a qual se pode acoplar aplicações ou simplesmente interagir com elas. Esta abordagem tira partido da grande maioria das normas e soluções técnicas referidas nos pontos anteriores, em particular do conceito de *Messaging*, dos *Web Services* e protocolos de comunicação. Os ESB seguem os princípios de uma arquitetura orientada aos serviços (SOA), permitindo a integração com diferentes tipos de serviços, e incluem de forma geral as seguintes áreas funcionais (Figura 3.36):

- *Standard-based Communication Infrastructure*
- *Standard-based Connectivity*
- *Standard-based Transformation Engines*
- *Service Oriented Architecture (SOA) for application deployment*
- *Standards-based Security*

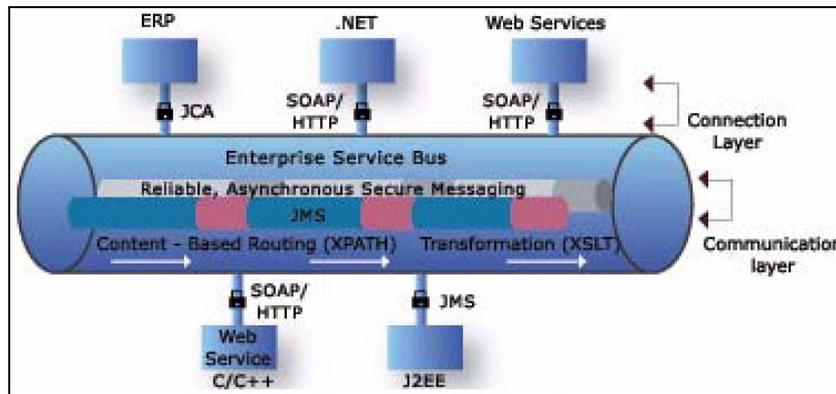


Figura 3.36: ESB (adaptado de Fiorano)

A camada de comunicação baseada num canal centralizado, gere a troca de mensagens garantindo o seu encaminhamento, aplicação de regras e condições, mapeamento e transformação de dados. Estas mensagens são normalmente estruturadas em XML o que permite tirar partido de funcionalidades assentes em XSLT e XPATH. A camada de conexão pode basear-se em diferentes normas como por exemplo o SOAP/HTTP para os *Web Services*, *Java Connector Architecture* (JCA) para adaptadores aplicativos, *Java Message Service* (JMS) para troca de mensagens com uma plataforma J2EE.

Como nas restantes perspectivas para a integração de SI, as normas existentes são muito importantes para determinar a portabilidade e adaptabilidade das soluções encontradas. Neste âmbito, a recente norma BPEL tira partido dos *Web Services* e do XML e é claramente uma solução completa e prática para qualquer situação ao nível da complexidade dos sistemas. Esta norma permite um nível de abstracção muito grande e facilita a integração de SI em qualquer das suas perspectivas, camadas tecnológicas ou arquitecturas. As recentes normas técnicas permitem novas formas de integração nomeadamente aquelas que estão presentes nos conceitos de SOA e ESB.

3.5 Web Services versus abordagens tradicionais para a integração de SI

As perspectivas identificadas nos pontos 3.1, 3.2, 3.3 fazem parte das abordagens tradicionais para a integração de SI. Cada uma delas tem âmbitos e objectivos diferentes embora possam partilhar algumas tecnologias ou tendências como se referiu. Neste âmbito, a grande maioria das soluções para a integração de SI são produtos comerciais que podem estar, ou não, assentes em normas de mercado. Para além disso, há sempre alguma dependência dos fornecedores de tecnologia, quer para a configuração do sistema, quer para a sua manutenção. Por outro lado, é frequente a necessidade de adaptação dos SI existentes à solução integradora escolhida. Estas abordagens também têm vantagens ao nível da maturidade e experiência por parte das empresas que fornecem essas soluções. Neste sentido, e independentemente da confusão de alternativas que possa haver nesta área, a rapidez com que se pode encontrar uma solução adequada é um factor favorável à organização.

Por seu turno, os *Web Services* (WS) oferecem uma arquitectura independente e baseada em normas de mercado regulamentadas e documentadas como são o XML, o WSDL, o UDDI e o SOAP. Numa organização, a adopção de WS permite, por um lado, disponibilizar funcionalidades dos sistemas internos de forma a poder integrá-los, e por outro lado, dá a possibilidade de disponibilizar ou mesmo comercializar esses serviços fora da organização. Enquanto que nas abordagens tradicionais, os pontos de integração estão previamente identificados e coordenados, os WS podem ser escolhidos no momento e consoante a sua funcionalidade ou disponibilidade. Finalmente, a tendência do mercado tecnológico nesta área aponta claramente para a adopção e exploração destas normas recentes devido às vantagens que oferecem. As desvantagens dos WS estão relacionadas com alguma imaturidade ainda existente e algumas lacunas funcionais. Os WS obrigam por exemplo a adoptar tecnologia que os possa orquestrar segundo uma determinada lógica. Aqui surgiu por exemplo a norma BPEL que resulta de uma convergência de especificações e vontades de alguns fornecedores de software.

Neste sentido, nenhuma das abordagens substituiu a outra e “cada caso é um caso”. Em cada situação, as organizações deverão forçosamente analisar o impacto de cada solução face às suas necessidades de integração. No entanto, ambas as abordagens são viáveis e necessárias para obter uma solução de integração de SI completa e abrangente. Apesar da sua independência, cada uma pode complementar a outra, e em conjunto, podem fornecer excelentes soluções nesta área.

3.6 Relacionamento das perspectivas para a integração de SI

Neste capítulo, e com o objectivo de descrever e catalogar as perspectivas descritas anteriormente, foram criados alguns quadros e algumas figuras específicas. Desta forma, é possível relacionar cada uma das realidades descritas neste trabalho realçando áreas e funcionalidades comuns.

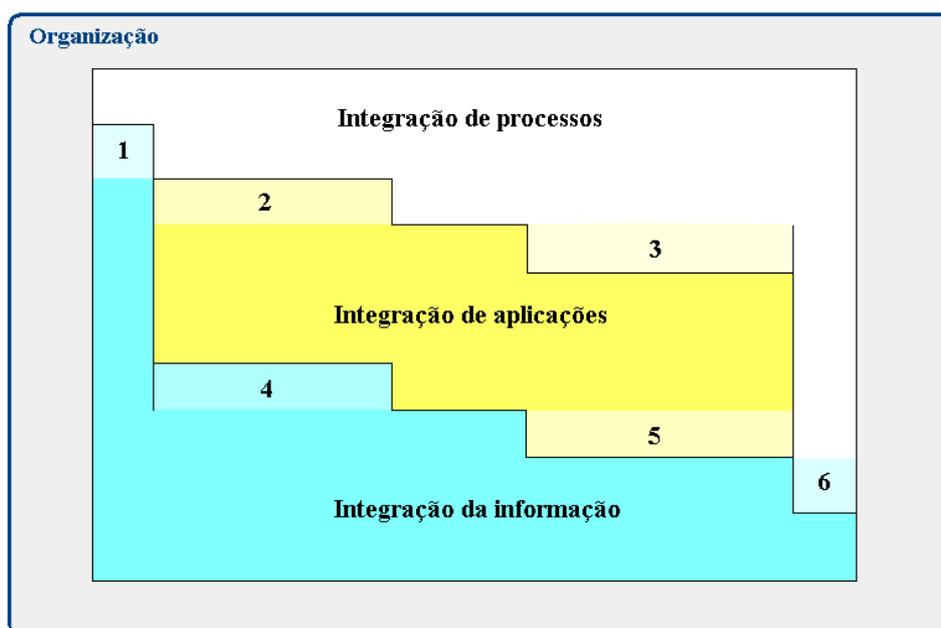


Figura 3.37: Perspectivas complementares da integração de SI

No âmbito de uma organização, pode-se identificar conceptualmente a integração de Sistemas de Informação com três perspectivas complementares: integração de informação, integração de aplicações e integração de processos (Figura 3.37). Entre cada área existem protocolos de comunicação, trocas de informação, funcionalidades e partilha de código. As áreas comuns distinguem-se pelas possíveis

abordagens e tecnologias de base (Quadro 3.5) o que significa que algumas das camadas conceptuais e/ou tecnológicas possam também podem aí estar presentes.

Camada comum	Descrição
1	Troca de informação com o objectivo de interligar processos organizacionais dando origem à sua execução.
2	Incorporação de Lógica aplicacional e regras de negócio na definição de um processo.
3	Implementação de lógica aplicacional dentro de uma aplicação para automatização e controlo do fluxo de processo, etapas, estados.
4	Funcionalidades aplicacionais que entram no âmbito da informação como é a integração de SQL em línguas de programação (ex: Oracle PL/SQL), controlo aplicacional da integridade da informação e as respectivas transacções.
5	Nesta camada pode haver lógica aplicacional implementada ainda na área da Informação como por exemplo os <i>stored procedures</i> dos SGBD.
6	Processos organizacionais cuja automatização permite veicular informação entre sistemas.

Quadro 3.5: Descrição das áreas comuns das perspectivas

As camadas representadas são complementares e correspondem às perspectivas da integração dentro de uma organização. A integração entre duas ou mais organizações (Figura 3.38) pode dar-se ao nível de qualquer uma das camadas aqui descritas, onde se procura integrar informação, aplicações e processos. Com os *Web Services* é possível integrar cada uma das perspectivas entre elas, ou então disponibilizar funcionalidades de cada perspectiva para uma integração com outra organização. Estas possibilidades correspondem às características de uma arquitectura do tipo SOA.

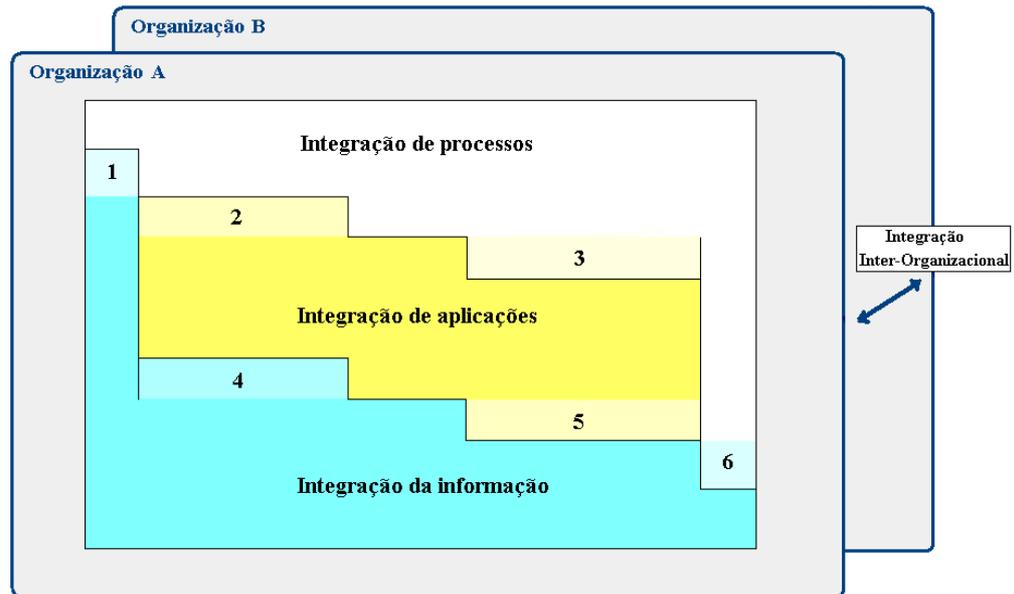


Figura 3.38: Perspectiva da integração inter-organizacional

Pode-se ainda relacionar cada perspectiva pelo nível de integração desejado numa organização que vai desde o nível 0 onde não há integração de sistemas até ao nível 4 que corresponde ao nível de integração entre organizações (Quadro 3.6). Cada perspectiva complementa a anterior e tem a seguinte descrição [Schmidt 2000]:

- **Nível 0 (Inexistente):** Existência de sistemas de informação sem qualquer tipo de integração ou interligação.
- **Nível 1 (Ponto a ponto):** Primeiro nível de integração com sincronização de repositórios e ligação ponto a ponto de sistemas. Sistemas *tight-coupled*.
- **Nível 2 (Estrutural):** Sistemas loosely-coupled. Nível de integração completo em termos de aplicações e repositórios de informação. Toda a estrutura de aplicações está integrada. Introdução de *Middleware*.
- **Nível 3 (Processual):** Introdução de sistemas de automatização de processos específicos para complementar os sistemas integrados existentes. Introdução de informação de gestão e indicadores acerca do comportamento da organização.
- **Nível 4 (Externo):** Estando todos os sistemas internos perfeitamente integrados a todos os níveis, há aqui a introdução do relacionamento externo com outras organizações e os seus sistemas no lógica de *Business-to-Business*.

Nível	Características	Perspectiva
0 - Inexistente	<p>Inserção de dados e sincronização de repositórios de informação manual. Aplicações isoladas e monolíticas.</p> <p>Processos manuais, em papel ou isolados dentro de uma aplicação. Organização isolada.</p>	
1 - Ponto a ponto	<p>Ferramentas de sincronização de repositórios de informação (Replicação) e sistemas <i>tight-coupled</i>.</p> <p>Integração ponto a ponto com API/RPC/ORB ou <i>Messaging</i>.</p>	IA II
2 - Estrutural	<p>Inclui nível Ponto a Ponto. Interfaces de comunicação com Message Hub / Bus. Sistemas <i>loosely-coupled</i></p> <p>Middleware com: Message Brokers. Application Servers. Data transformation Rule processing Transaction Integrity</p>	IA II
3 - Processual	<p>Inclui nível Estrutural. BAM, Business Intelligence e Portais.</p> <p>Informação gerida e partilhada numa lógica de processo.</p> <p>Funcionalidades: Automated Routing, Automated Decision e Workflow</p>	IP IA II
4 – Externo	<p>Inclui nível processual. Infraestrutura de rede e comunicação comuns (ex: Internet).</p> <p>Normalização da informação comum e em XML. Integração de Processos entre organizações. Sistemas aplicacionais comuns.</p> <p>Middleware com: Secure Transaction Data Mapping</p>	IO

Quadro 3.6: Nível de integração de SI versus Perspectivas (adaptado de [Schmidt 2000])

O nível 2 pode por vezes representar o nível 3 caso os sistemas aplicacionais englobem lógica aplicacional que permita automatizar os processos da organização.

O XML é uma norma que pode ser utilizada pela grande maioria das tecnologias ou plataformas existentes. As normas que suportam os *Web Services* baseiam-se no XML nas suas diversas funcionalidades. A arquitectura SOA baseia-se também nos *Web Services* e no XML mas podem ser usadas outras normas equivalentes com os mesmos objectivos de integração (Quadro 3.7).

	Java RMI	CORBA	DCE	Web Services
<i>Invocação de procedimentos ou serviços</i>	Java RMI	CORBA	RPC	JAX-RPC, .NET
<i>Formato da informação</i>	Serialized Java	Common Data Representation	Network Data Representation	XML
<i>Transporte Da informação</i>	Stream	General Inter-ORB Protocol	Protocol Data Units	SOAP
<i>Protocolo de comunicação</i>	Java Remote Method Protocol	Internet Inter-ORB Protocol	RPC Connection-Oriented Protocol (ex:TCP)	HTTP, SMTP
<i>Descrição do serviço</i>	Java Interface	CORBA IDL	DCE IDL	WSDL
<i>Descoberta do serviço</i>	Java Registry	Corba Object Service Naming	Cell Directory Service	UDDI

Quadro 3.7: Formatos e protocolos numa arquitectura SOA

No mercado tecnológico é fácil encontrar definições diferentes para uma mesma sigla ou tecnologia. Por causa de questões de concorrência no mercado das TI é difícil encontrar descrições estanques da cada abordagem, perspectiva ou tecnologia de integração de SI. Mas em termos conceptuais, e de forma genérica, pode-se relacionar algumas das normas e tecnologias mais comuns com as suas famílias tecnológicas (Figura 3.39):

- Object Request Borker (ORB): CORBA, COM/DCOM, RMI
- Distribute Computing Environment (DCE): RPC
- Transaction Processing (TP): TP Monitor, SGBD
- Messaging Oriented Middleware (MOM): Message Queueing e Message Broker

Os *Integration Brokers* estão baseados essencialmente no *Messaging* mas podem incluir outras tecnologias para servir como *Middleware* completo para a integração de SI. Pode conter por exemplo adaptadores para aplicações e ferramentas de monitorização.

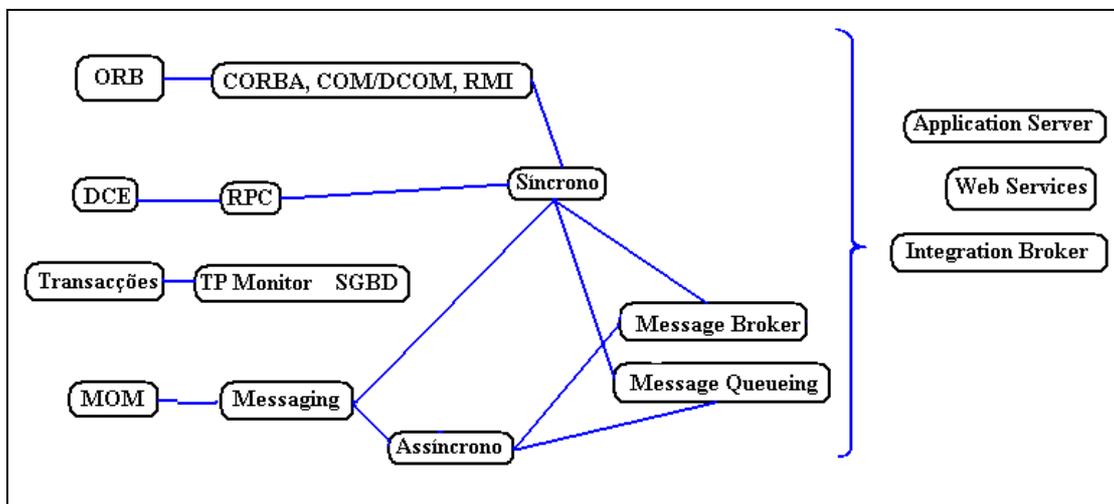


Figura 3.39: Famílias tecnológicas e normas

Na mesma lógica, os *Application Servers* também incluem muitas das tecnologias aqui referidas e permitem diferentes abordagens de integração de SI. Estes são comumente associados à família dos monitores transaccionais, mas são hoje em dia muito mais do que isso.

Os *Web Services* posicionam-se como uma nova alternativa, embora baseados em conceitos tradicionais, para facilitar a integração de sistemas baseada em XML. É uma abordagem que tem tido uma grande aceitação no mercado obrigando os vários fornecedores tecnológicos a incorporá-la nos seus produtos. Com a possibilidade de disponibilizar qualquer informação ou sistema através dos *Web Services* é no entanto necessário dar-lhes alguma lógica e coordenação durante a sua integração. Neste caso o BPEL é a norma de referência com esse mesmo intuito.

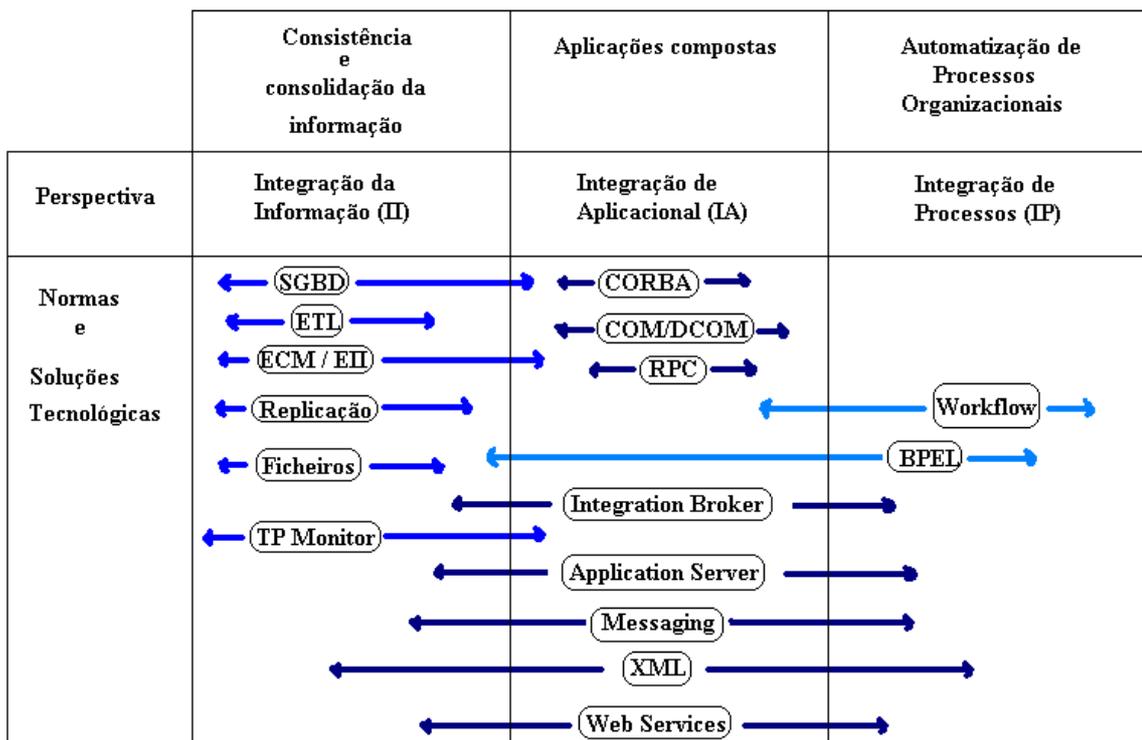


Figura 3.40: Âmbito geral das normas e tecnologias por perspectivas

É quase impossível enumerar todas as normas ou tecnologias para cada situação específica que requer desenvolvimento aplicacional ou integração. A evolução das TI é de tal modo rápida que qualquer enquadramento feito neste sentido poderia ficar obsoleto. Mas as normas ou tecnologias mais comuns ilustradas neste trabalho podem ter aplicabilidade, ou não, consoante a perspectiva adoptada (Figura 3.40). Algumas normas são totalmente abrangentes como é o caso do XML, dos *Web Services* ou do *Messaging* e outras serão mais contextualizadas.

4. Soluções para a integração de Sistemas de Informação

Para a área da integração de SI existem inúmeras soluções que seguem diferentes abordagens e tecnologias. Por um lado, as mesmas soluções podem ser utilizadas na resolução de diferentes problemas, dificultando a sua categorização, e por outro lado, a constante evolução e actualização tecnológica vem agravar essa dificuldade de catalogação [William 2000]. Face a esta realidade, e com o objectivo de simplificá-la, são identificadas, e sumariamente explicadas neste capítulo, algumas das soluções e normas mais comuns, por cada perspectiva descrita nos capítulos anteriores (Figura 4.1). Não está aqui enumerada a totalidade das soluções e normas, que foram surgindo ao longo dos tempos. Isto deve-se ao facto de algumas delas estarem desactualizadas ou terem convergido o que não ajudaria os propósitos de clarificação deste trabalho.

PERSPECTIVAS	SOLUÇÕES / NORMAS
Integração da Informação	XML
Integração Aplicacional	COM e Javabeans CORBA, DCOM e Java RMI Messaging e Message Broker Monitores Transaccionais Remote Procedure Call
Integração Inter-Organizacional	EDI ebXML
Integração de Processos	BPEL
Integração com Web Services	SOAP WSDL UDDI

Figura 4.1: Principais soluções e normas por perspectivas da integração de SI.

A integração da informação numa organização ocorre essencialmente ao nível do seu armazenamento e do seu transporte de uma fonte para outra. Neste capítulo,

destaca-se o XML como a norma mais abrangente para todo o tipo de manuseamento e integração de informação em diversas soluções tecnológicas. Esta norma enquadra-se também nas outras perspectivas mas está associada na sua essência à gestão da informação.

A integração aplicacional segue essencialmente técnicas orientadas aos métodos ou funções programáticas. Existem diversas formas de integração directa, através de invocações de código, que permitem colocar componentes ou objectos a interagir através de canais de comunicação. Estas interacções são na maioria dos casos síncronas e baseadas no tipo *request/reply* entre as aplicações envolvidas. Estas abordagens de integração podem assentar, entre outros, em código aplicacional (COBOL, C++, Java, etc.), nos API (*Application Programming Interface*), nos RPC (*Remote Procedure Call*), nos *TP Monitors* (Transaction Processing Monitors), nos objectos distribuídos, nos componentes do tipo CORBA (*Common Object Request Broker Architecture*), nos Java RMI (*Java Remote Method Invocation*), nos MOM (*Message Oriented Middleware*) e nos *Web Services*. São aqui destacadas as seguintes normas e técnicas: RPC, TP, ORB, COM e *Messaging*.

Quanto à integração entre organizações são realçadas as normas EDI e ebXML. A primeira abordagem para este tipo de integração foi a solução *Electronic Data InterChange* (EDI) sendo ainda hoje uma norma bastante utilizada. Na tentativa de tirar partido do XML surgiu o ebXML que permite estruturar nessa base as trocas de informação entre organizações.

A integração orientada aos processos pode seguir vários caminhos consoante as normas adoptadas. Esta é uma área complexa onde surgem várias especificações resultantes de diferentes correntes no mercado tecnológico. Neste sentido, existem diversas normas, como por exemplo o BPMN, o WSFL ou o XPDL, que seguem diferentes raciocínios e que documentam os processos de formas diferentes. Nesta área tem-se verificado uma convergência de especificações que, ao longo dos tempos, se vão fundindo para criar novas normas mais abrangentes. Por essa razão, e pelo facto de também tirar partido do XML, dá-se destaque à norma BPEL para a integração de processos. Esta norma tem sido consensual e tem aproximado os fornecedores de tecnologia o que também revela a sua importância.

No que diz respeito à integração orientada aos serviços são aqui incluídas as normas respeitante aos *Web Services*: SOAP, UDDI e WSDL. Os *Web Services* são a evolução de muitas das outras técnicas aqui enumeradas que surgiram com a proliferação da Internet e das suas normas de comunicação e invocação de informação e aplicações.

4.1 O Extended Markup Language (XML)

Este capítulo não pretende descrever exhaustivamente a norma XML e todas as suas componentes mas sim dar uma visão geral do seu propósito e da sua estruturação, e permitir entender o seu enquadramento no tema da integração de SI. O organismo World Wide Web Consortium (W3C) definiu o Extended Markup Language (XML) [Ramalho e Henriques 2002] baseando-se na “meta-linguagem” Standard Generalized Markup Language (SGML) da norma ISO 8879. O XML é uma norma que define uma linguagem padrão para a representação de informação independentemente das tecnologias. O XML é uma linguagem de *markup* que consiste num conjunto de regras para formatar e estruturar dados explicando o seu significado. Um documento XML usa uma sintaxe simples e de fácil compreensão constituído por uma árvore de elementos relacionados e rotulados (Figura 4.2). Esta especificação facilita a pesquisa e a manipulação ágil de informação uniforme e independente das aplicações. O XML especifica um padrão que permite codificar o conteúdo, as semânticas e as esquematizações para uma grande variedade de fontes de informação desde as mais simples até as mais complexas.

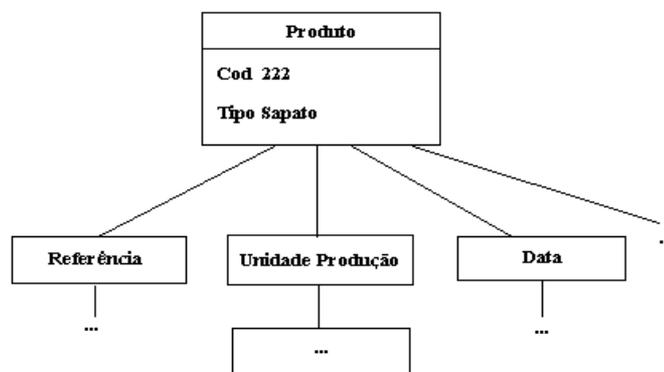


Figura 4.2: XML – Estruturação em árvore

À semelhança do *HyperText Markup Language (HTML)* o XML permite a definição de *tags* ou delimitadores, que caracterizam os dados e a sua formatação, mas de forma ilimitada. É possível criar um conjunto de *tags* ou marcas específicas para obter uma formatação para um determinado conjunto de informação. Cada elemento, ou conjuntos de elementos, está relacionado e estruturado numa lógica de árvore ou hierarquia com dependências e ramificações. Uma folha da árvore será um elemento que pode conter dados como o título de um livro, o código de um produto ou qualquer outro tipo de dado. Com o XML pode-se criar uma camada estruturante para um documento de texto, um registo de base de dados estruturado, um objecto com métodos e dados como são os objectos Java, um registo de dados resultantes de uma pesquisa, a apresentação gráfica de uma aplicação, etc...

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- Edited with XML Spy v4.2 -->
<breakfast_menu>
<food>
<name>Belgian Waffles </name> <price>$5.95</price>
  <description>two of four famous Belgian Waffles with plenty of real maple
  syrup </description> <calories>650</calories>
</food>
<food>
<name>Strawberry Belgian Waffles </name>
<price>$7.95</price>
<description>light Belgian waffles covered with strawberries and whipped
cream </description> <calories>900</calorie>
</food>
</breakfast_menu>
```

Figura 4.3: XML – Estruturação em árvore

O XML permite também separar a camada de apresentação dos dados da sua estrutura. Enquanto que o HTML especifica como o documento deve ser apresentado num *web browser*, o XML define o conteúdo desse documento. No HTML existem por exemplo *tags* para definir o tamanho e a cor da letra, as *tags* de XML descrevem os dados identificando que aquele elemento é um título, um autor, uma data, uma morada, uma referência, etc... Uma estrutura XML pode também ser usado para troca de informação permitindo a integração dos dados com diferentes origens e estruturas (Figura 4.3). É assim possível obter dados de diferentes sistemas, convertê-los numa estrutura XML e distribuí-los directamente às partes intervenientes.

```

<?xml version="1.0"?>
<!DOCTYPE note[
  <ELEMENT note (to,from,heading,body)>
  <ELEMENT to (#PCDATA)>
  <ELEMENT from (#PCDATA)>
  <ELEMENT heading (#PCDATA)>
  <ELEMENT body (#PCDATA)> ]>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend</body>
</note>

```

Figura 4.4: DTD

Os elementos XML são formados por *tags* de abertura `<elemento>`, sendo o mesmo para *tags* de fecho `</elemento>`. Por exemplo a definição do elemento marca de automóvel pode ser representada da seguinte forma: `</marca>Alfa Romeo</marca>`. Um documento XML estrutura-se com pelo menos três partes: uma declaração, o *Document Type Declaration* (DTD) e a instância. A declaração funciona como um cabeçalho descritivo que indica a versão de XML usada, o grupo de caracteres possíveis ou outras características. Um DTD caracteriza-se por um conjunto de declarações que especifica o tipo de estrutura do documento em XML (Figura 4.4). A instância é o documento propriamente dito contendo a informação e as anotações. Na própria instância poderá fazer-se referência ao DTD caso não esteja presente directamente no documento XML. Também existe a abordagem *XML Schema Definition* (XSD) que se posiciona como uma alternativa mais completa e flexível do que o DTD. O XSD foi proposto inicialmente pela Microsoft e passou a ser uma recomendação oficial do W3C em 2001.

O XML pode ainda ser complementado com “folhas de estilo” do tipo *Extensible Style Language* (XSL) para controlar a apresentação dos dados (Figura 4.5). O XML permite estruturar e representar informação de uma forma simples e independentemente da sua origem ou do seu destino. Um documento que contém informação formatada pelo XML pode não ser suficientemente flexível para ser manipulado sobretudo quando não se conhece à partida a sua estruturação. Para facilitar essa interpretação e visualização do conteúdo em XML surge outra norma do W3C: o *eXtensible Stylesheet Language* (XSL). O XSL permitirá substituir as *tags* de

estruturação da informação por marcas de formatação ou transformação semelhante ao que acontece com o HTML.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl"
href="simple_table.xsl"?>
<root>
  <person>
    <forename>meta</forename>
    <surname>mickey</surname>
    <age>21</age>
  </person>
  <person>
    <forename>hey</forename>
    <surname>bukko</surname>
    <age>99</age>
  </person>
</root>

O Ficheiro XSL:
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
<xsl:template match="root">
<table border="1">
  <tr>
    <td>Forename</td>
    <td>Surname</td>
    <td>age</td>
  </tr>
<xsl:apply-templates/>
</table>
</xsl:template>
<xsl:template match="person">
  <tr>
    <xsl:apply-templates/>
  </tr>
</xsl:template>
<xsl:template match="forename">
  <td>
    <xsl:apply-templates/>
  </td>
</xsl:template>
<xsl:template match="surname">
  <td>
    <xsl:apply-templates/>
  </td>
</xsl:template>
<xsl:template match="age">
  <td>
    <xsl:apply-templates/>
  </td>
</xsl:template>
</xsl:stylesheet>
```

Figura 4.5: XSL

Na sua primeira abordagem, o W3C definiu o XSL com ambas as partes de formatação e de transformação do XML. Mais tarde decidiu separar criando o *XML Path Language* (XPath), o *XSL Transformations* (XSLT) e o *XSL Formatting Objects* (XSLFO). Pode-se considerar o XSLFO como sendo igual ao XSL. É uma linguagem de *markup* que descreve a formatação da apresentação de um documento XML e deixando a componente de transformação para o XSLT. O XPath permite localizar informação dentro de um documento XML. Trata-se de uma linguagem que permite identificar partes do documento XML para poder interagir com elas. Por exemplo é

possível ter expressões de localização do tipo Xpath dentro de um XSL. O XSLT permite a transformação de um documento XML noutros formatos ou estruturas. O XSLT caracteriza-se como uma metodologia com regras de transformação de uma estrutura para outra. Esta abordagem é por exemplo utilizada para transformar documentos XML em documentos formatados para uma determinada apresentação, podendo ser complementado com o Xpath e permitir o controlo de troca de informação entre sistemas. O XML Document Object Model (XML DOM) consiste uma interface de programação para documentos em XML. O XML DOM define uma forma de aceder e manipular documentos XML para poder navegar pela sua estrutura modificar os valores dos seus elementos. Pelo facto de não haver restrições de nomeação dos elementos de um documento XML poderão ocorrer conflitos de estruturação entre este tipo de documentos. No XML é possível usar o XML namespaces para controlar eventuais conflitos de descrição dos elementos. Os namespaces permitem resolver conflitos entre dois documentos XML no seu elemento `<table>` com dois namespaces (<http://www.w3.org/TR/html4/> e <http://www.w3schools.com/furniture>) diferenciando cada um deles e mantendo o seu significado (Figura 4.6).

```

Documento XMLn°1:
<table>
<tr>
<td> Apples</td>
<td> Bananas</td>
</tr>
</table>
Documento XMLn°2:
<table>
<name> African Coffee Table</name>
<width>80</width>
<length>120</length>
</table>
Documento XMLn°1 com namespaces:
<html xmlns:hs="http://www.w3.org/TR/html4/">
<htr>
<h:td> Apples</h:td>
<h:td> Bananas</h:td>
</htr>
</html>
Documento XMLn°2 com namespaces:
<f:table xmlns:f="http://www.w3schools.com/furniture">
<f:name> African Coffee Table</f:name>
<f:width>80</f:width>
<f:length>120</f:length>
</f:table>

```

Figura 4.6: Namespaces

4.2 Soluções para a integração aplicacional dentro da organização

Como no capítulo anterior, divide-se este tipo de integração pelas situações que existem dentro da organização e aquelas que procuram a integração entre duas ou mais organizações. Os pontos seguintes enquadram-se no âmbito interno das organizações.

4.2.1 Transaction Processing Monitors

Os monitores transaccionais (MT) são também conotados de *middleware* e correspondem a um ambiente de execução transaccional de aplicações. É uma tecnologia muito popular em ambientes de *mainframes* mas podem ser utilizados noutras situações. O MT fornecem um conjunto de serviços para sistemas num ambiente distribuído como é o cliente/servidor. Estes serviços vão desde o controlo do acesso às bases de dados, controlo de *workflow*, gestão de transacções entre aplicações e até suportam as comunicações com a interface do utilizador. Estes monitores distribuem os n pedidos dos clientes em rotinas de processamento para poder executar as transacções pedidas (Figura 4.7).

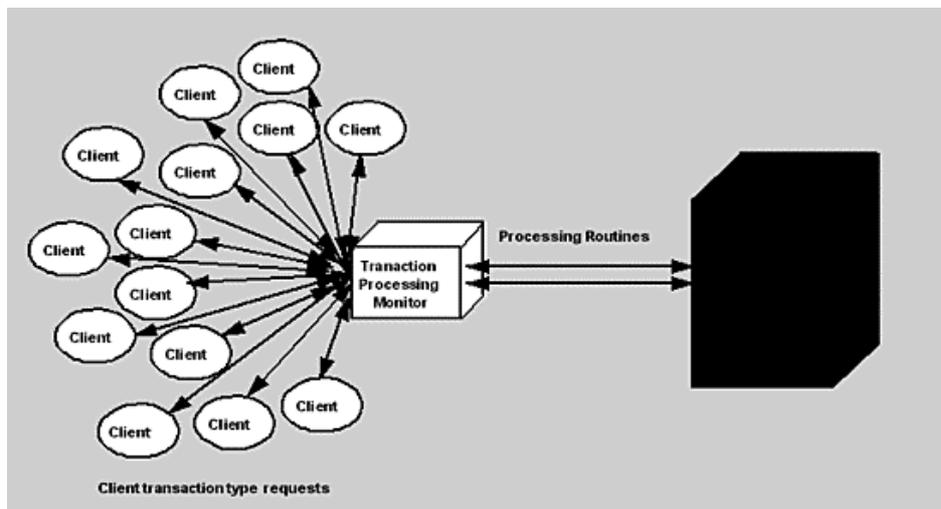


Figura 4.7: Monitor Transaccional

Uma transacção é composta por um conjunto de operações atómicas e só tem sucesso caso todas as operações sejam executadas. Este conceito é aplicado em muitas tecnologias como por exemplo as bases de dados transaccionais ou em motores de

workflow. Um monitor transaccional fornece um conjunto de funções como por exemplo a gestão de transacções falhadas, o balanceamento de carga, a consistência e distorção de informação e níveis de segurança na execução das transacções.

Esta tecnologia permite a integração das aplicações quer no acesso às bases de dados quer na gestão de interacções entre aplicações. É possível tratar uma aplicação como um sistema ou um componente para ser chamado por uma transacção. Uma aplicação pode comunicar com outras aplicações através destas transacções que têm a sua integridade garantida. Os monitores transaccionais permitem suportar sistemas distribuídos e integrar diferentes componentes entre si. São suportados vários modelos de comunicação entre aplicações como o *store and forward*, os RPC e outras formas assíncronas.

Os monitores transaccionais possibilitam a criação de grandes sistemas de informação integrados e suportados por diferentes módulos aplicativos interligados por estas transacções. Esta tecnologia garante a execução destas transacções de forma fiável, eficiente e com capacidade de recuperação.

4.2.2 Remote Procedure Call

Numa arquitectura cliente/servidor, o RPC (Remote Procedure Call) é um protocolo que permite a um programa executar outro programa remoto e armazenado num servidor. O programa cliente envia uma mensagem para o servidor com os respectivos parâmetros e o servidor responde com uma mensagem com a informação resultante da execução.

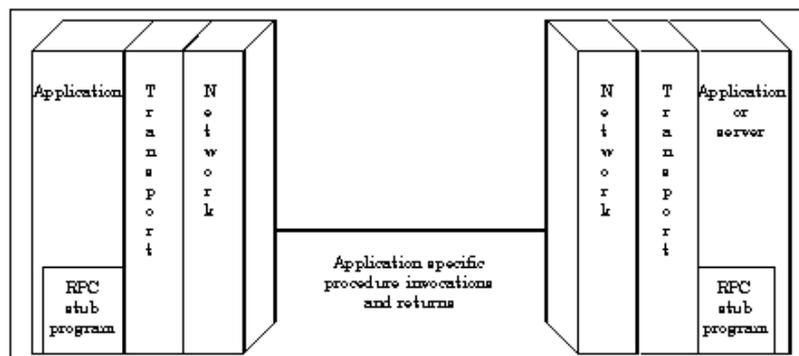


Figura 4.8: Remote Procedure Call

Numa aplicação do tipo cliente/servidor o acesso ao programa remoto é feito por chamadas síncronas do tipo RPC que são embutidas no programa cliente (Figura 4.8). Por ser um tipo de codificação trabalhoso e de difícil gestão surgiram formas automáticas de geração de código para suportar o RPC. Essa geração automática é feita com base numa linguagem de especificação da interface dos procedimentos remotos designada por IDL (*Interface Definition Language*) [Open 1997] e independente das linguagens de programação. Este tipo de abordagem facilita aos programadores a incorporação destes procedimentos quer do lado do cliente quer do lado do servidor. O RPC corresponde a uma das camadas da arquitectura DCE (*Distributed Computing Environment*) definida pelo Open Group (Figura 4.9).

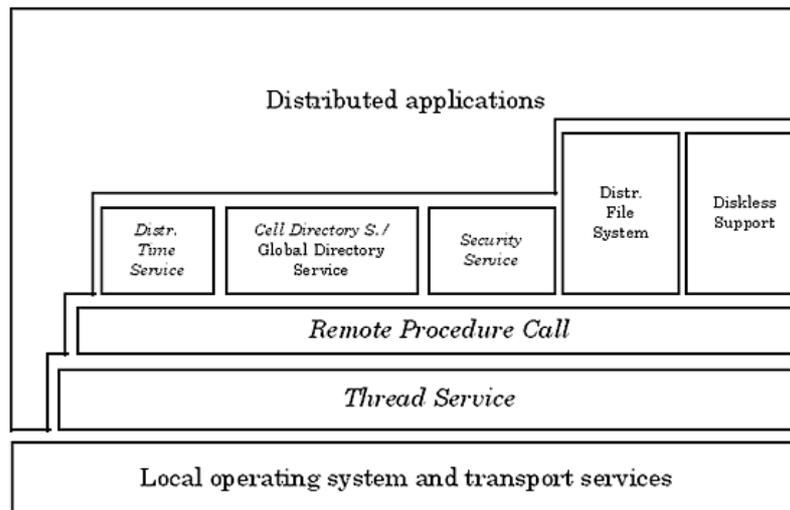


Figura 4.9: Arquitectura DCE

O CORBA e o DCOM são duas metodologias orientadas aos objectos para a comunicação entre objectos que fornecem os mesmo tipo de funcionalidades que o RPC.

4.2.3 Componentes COM e JavaBeans

O desenvolvimento de software baseado em componentes ou objectos tem como objectivo a construção de soluções integrando componentes já existentes. Este tipo de abordagem é referida como CBSE (*Component-based software Engineering*) [Brown 1996]. A especificação COM (*Component Object Model*) [COM 1998],

desenvolvida pela Microsoft, descreve uma plataforma para a integração de componentes de *software* que suporta a sua interoperabilidade e a reutilização. É possível construir aplicações flexíveis com a assemblagem de componentes de diferentes origens que comunicam através do COM (Figura 4.10). Os ambientes em Java tem especificações semelhantes que se enquadram nos componentes *JavaBeans*.

A definição de um componente pode ter várias alternativas mas de uma forma genérica pode-se definir este tipo de objecto como: “Um pequeno objecto binário ou programa que desempenha uma função específica e é concebido de forma a comunicar e integrar com outros componentes e aplicações”³.

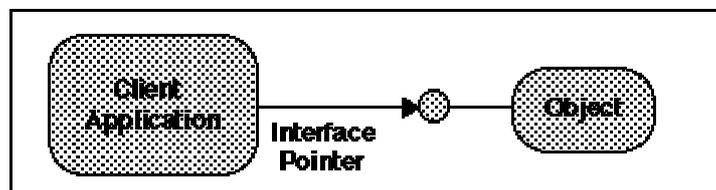


Figura 4.10: Aplicação que integra um objecto COM.

Para introduzir mecanismos de comunicação remota nos próprios objectos, surge DCOM (*Distributed COM*) [Horstman e Kirtland 1997]. Trata-se de uma extensão do COM que permite a interacção entre componentes assentes em protocolos de comunicação em rede. Enquanto que os processos de integração COM são executados na mesma máquina em diferentes endereços, a extensão DCOM permite que a integração de componentes sejam feita através das redes de comunicação e de protocolos como o HTTP. O DCOM foi criado como resposta à especificação CORBA (*Common Object Request Broker Architecture*) do consórcio OMG (*Object Management Group*).

Tanto o COM como o DCOM representam uma abordagem muito específica e baixo nível em termos de codificação e integração de componentes de *software*. As tecnologias OLE (*Object Linking and Embedding*), ActiveX e MTS (*Microsoft transaction Server*) da Microsoft estão assentes em COM e DCOM que estão dependentes do sistema operativo *Windows*. Estes representam um nível superior de disponibilização de serviços aplicativos em termos de integração, portabilidade e

³ <http://www.webopedia.com/TERM/c/component.html>

segurança. O *applet* é o componente em Java equivalente ao ActiveX da Microsoft que permitem executar pequenas aplicações descarregadas para os *web browsers* a partir de servidores *web* remotos. O COM+ é uma evolução da especificação COM, que tira partido das funcionalidades transaccionais e *messaging* do MTS, para a sua utilização e integração em aplicações. O COM+ tem como equivalência no mundo Java os EJB (*Enterprise Java Beans*) que entretanto evoluiu para a plataforma J2EE mais abrangente.

Como se pode concluir do mercado de fornecedores de software, existem duas correntes alternativas suportadas, por um lado, pela Microsoft, e por outro, por um conjunto alargado de outras empresas. Neste caso identifica-se a plataforma .NET da Microsoft e o J2EE defendido pela SUN, Oracle, IBM, Sybase e BEA.

4.2.4 Objectos distribuídos CORBA, DCOM e Java RMI

O conceito ORB (*Object Request Broker*) corresponde a tecnologia ao nível do *middleware* que gere a comunicação e troca de informação entre objectos distribuídos. Um ORB é um componente de acordo com o modelo CORBA (*Common Object Request Broker Architecture*) do consórcio OMG (*Object Management Group*) que gere a integração de objectos num modelo cliente/servidor (Figura 4.11).

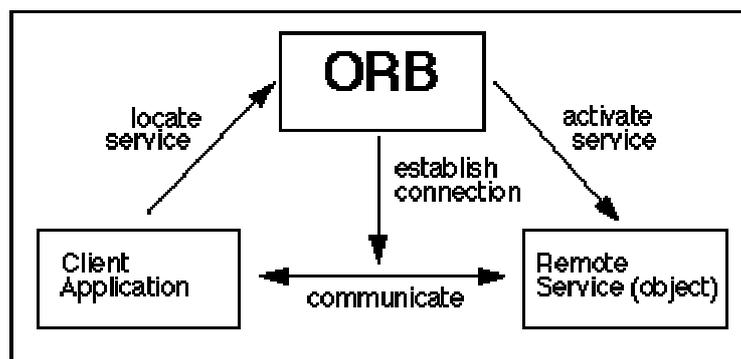


Figura 4.11: Object Request Broker

Um ORB actua de forma a fornecer um directório de serviços disponibilizados por componentes e possibilita as ligações com esses serviços. Um ORB funciona

como um canal de comunicação que controla as interações entre componentes e sistemas de forma a:

- Definir as interfaces de comunicação.
- Localização e activação de objectos remotos.
- Comunicação entre clientes e os objectos.

Existem dois grandes grupos de tecnologias que permitem uma abordagem do tipo ORB:

- O CORBA (*Common Object Request Broker Architecture*)
- Microsoft COM (Component Object Model)

Em 1994 a OMG definiu o CORBA versão 2.0 que incluiu uma especificação mais clara do canal de comunicação ORB denominado de IIOP (Internet Inter-Orb Protocol). O IIOP é baseado nos protocolos da Internet, nomeadamente na norma TCP/IP, e permite estruturar a forma como a informação circula entre os objectos. Pode-se identificar um modelo ORB mais recente com a tecnologia Java denominado de RMI (*Remote Method Invocation*) que permite a execução remota de métodos de objectos Java [Wollrath e Waldo 2005]. O Java RMI é mais simples que o COM/DCOM ou o CORBA mas depende do Java. Por seu lado o COM/DCOM depende do Windows e o CORBA não é tão flexível quanto os restantes apesar de estar implementado em várias linguagens. O CORBA começou a cair em desuso sobretudo pelo facto das comunicações serem síncronas e pelas surgimento dos *web services* e da norma SOAP.

4.2.5 Messaging (MOM e Message Brokers)

O MOM (*Messaging Oriented Middleware*) permite suportar comunicações síncronas e assíncronas entre aplicações num ambiente cliente/servidor. Existem filas de mensagens (*message queues*) que permitem a circulação de informação entre cada sistema ligado ao MOM (Figura 4.12).

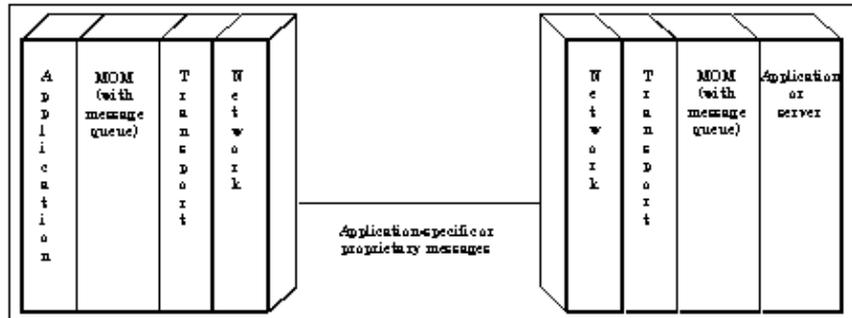


Figura 4.12: *Messaging*

O MOM permite um elevado nível de flexibilidade e independência para a troca de mensagens entre aplicações e sua consequente integração. Este tipo de arquitectura é indicada para aplicações que necessitam de uma integração orientada aos eventos. Na ocorrência de um conjunto de situações ou condições é despoletado um evento que envia uma mensagem para o software de gestão de mensagens e do qual pode ou não esperar uma resposta. Esse software, normalmente conotado como *middleware*, está encarregue de gerir uma fila de mensagens (*message queue*) e de entregar cada uma delas ao seu destino.

A integração entre duas aplicações pode seguir esta filosofia onde surgem várias alternativas com ligações ponto a ponto, *publish/subscribe* ou por um canal de comunicação. Estas soluções são normalmente designadas por *message broker* no caso de gerir as mensagens assincronamente. Os mecanismos de comunicação síncronos ou assíncronos podem ter vantagens e desvantagens consoante as situações e devem ser contempladas na configuração do *software*. No primeiro caso as mensagens são enviadas para o destino do qual é esperada uma resposta para o sistema emissor. No segundo caso o sistema emissor e o sistema receptor não precisam de estar simultaneamente ligados ao canal de comunicação. As mensagens são colocadas em fila até que o sistema destino as receba sem ter de enviar uma resposta em troca. Neste âmbito pode-se identificar pelo menos três topologias [Hohpe et al. 2003] de comunicação:

1. *Peer to peer* ou *Point to point*
2. Hub and Spoke
3. Bus ou Pipeline ou Publish and Subscribe

A comunicação *point to point* (Figura 4.13) permite criar canais de comunicação específicos para cada ponto de integração entre módulos aplicativos. A circulação da informação segue caminhos fixos com pontos de partida e de chegada previamente conhecidos. Esta não é a situação mais desejável devido à sua pouca flexibilidade surgindo nos casos das soluções proprietárias que não seguem normas de mercado.

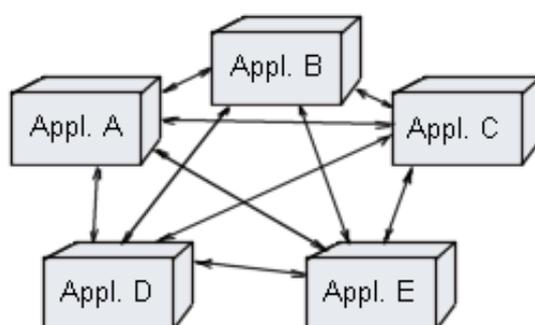


Figura 4.13: Comunicação *point to point*

A comunicação *hub and spoke* (Figura 4.14) permite centralizar o canal de comunicação que funciona como uma ponte comum (*hub*) para troca de informação. Este tipo de solução incorpora funcionalidades de comunicação *hub and spoke* na sua camada de *middleware* e é normalmente utilizada na área de EAI. A camada de *middleware* funciona com gestor de mensagens do tipo *Message Broker* que recebe as mensagens dos sistemas (*spokes*), determina os seus diversos destinos e encaminha as mensagens para os canais de destino.

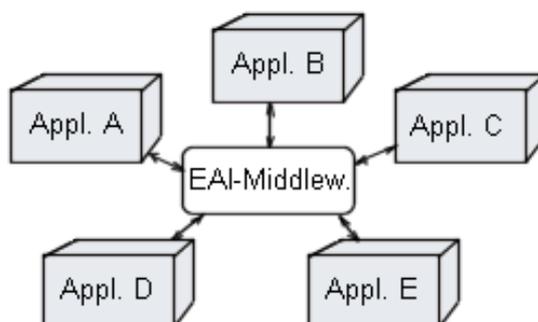


Figura 4.14: Comunicação *hub and spoke*

A comunicação do tipo *bus* (Figura 4.15) permite difundir eventos na forma de mensagens para um ou mais destinos que os reconhecem subscrevendo a sua recepção. Os sistemas catalogam cada mensagem, identificando-a segundo um determinado tópicos (*publish*), submetendo-as numa fila de mensagens (*bus*) em vez de enviá-la para destinos específicos. O canal de comunicação envia a mensagem para todos os sistemas que pediram para receber as mensagens com essa identificação (*subscribe*). Trata-se de uma forma assíncrona de troca de mensagens controlada por uma canal específico de comunicação.

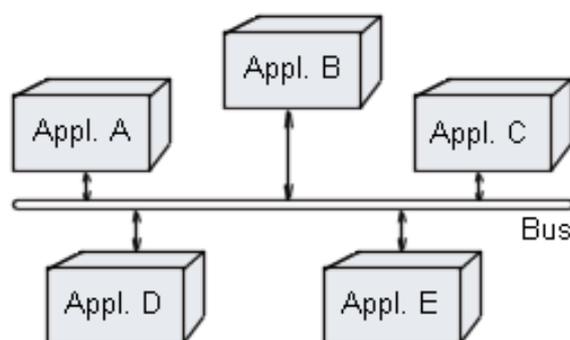


Figura 4.15: Comunicação do tipo *Bus*

Ao contrário dos RPC síncronos as soluções MOM podem acumular mensagens assincronamente e, por alguma razão, não conseguir entregá-las ao seu destino. No entanto as filas de espera do MOM permitem usar o sincronismo para a entrega da informação e voltar para um estado de assincronismo caso alguns dos destinos não estejam a funcionar. As soluções MOM tem interfaces aplicacionais para ser integrados em soluções aplicacionais. Por exemplo, o *Java Message Service (JMS)* é uma API em Java que faz parte da arquitectura J2EE e que permite o acesso às funcionalidades de um *software* do tipo MOM.

Os *message brokers* funcionam como uma ponte integradora entre aplicações permitindo a circulação e troca assíncrona de mensagens. Para além da circulação e do encaminhamento das mensagens, os *message brokers* podem ler o conteúdo delas, decodificá-las, interpretá-las, modificá-las e voltar a codificá-las para depois serem entregues no destino. Trata-se de um conjunto complementar de funcionalidades como são:

- Adaptadores ou *gateways*
- *Metadata directory*
- *Message queue*
- *Audit trail database*
- *Routing service*
- *Transformation and mapping services*

O XML (*eXtended Markup Language*) é uma das linguagens mais comuns na estruturação deste tipo de mensagens. Os *message brokers* comerciais disponibilizam o conjunto de adaptadores já preparados para integrar com diversos tipos de sistemas e aplicações como são o SAP, o Oracle E-Business Suite ou a PeopleSoft entre outros.

4.3 Soluções para a integração inter-organizacional

A integração inter-organizacional contempla as transacções entre organizações que interligam os seus processos e trocam a informação que os suporta. As diferentes organizações integram directamente os seus sistemas de informação, seguindo normas que regulam a estruturação dos documentos de negócio e a forma como são feitas as transacções electrónicas. Os pontos seguintes descrevem sumariamente duas das normas mais comuns: o EDI e o ebXML.

4.3.1 EDI

O EDI (*Electronic Data Interchange*) consiste na troca de documentos de negócio normalizados entre aplicações e as respectivas organizações (Figura 4.16). Documentos como notas de encomenda ou facturas são transmitidos de um sistema para outro de forma electrónica. Para isso as organizações estabelecem um acordo para a transmissão e formato desses documentos.

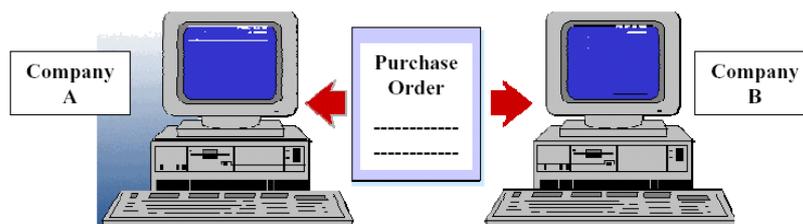


Figura 4.16: *Electronic Data Interchange (EDI)*

As normas EDI especificam as regras de sintaxe dos documentos, da organização dos dados dentro de um documento e da regras de comunicação e troca de informação. O objectivo das normas EDI é a especificação deste tipo de documentos e dos protocolos de transmissão entre as organizações. Uma mensagem EDI normalizada tem o nome de *transaction set* no caso do ASC X12 ou de *message* no caso do UN/EDIFACT (Quadro 4.1).

Documento	ASC X12	UN/EDIFACT
Purchase Order	850	ORDERS message
Invoice	810	INVOIC message
Acknowledgement	997	CONTRL message

Quadro 4.1: Exemplo de *transaction set* e de *message*

Uma mensagem EDI tem um cabeçalho (HEADER) que identifica o documento com informação como por exemplo a data do documento, o nome da empresa e a morada. O corpo do documento (DETAIL) inclui toda a informação relativa ao documento como quantidades de produto, descrições e preços. No final existe um sumário (SUMMARY) que contém informação de controlo e referente ao conjunto de linhas do documento. Cada linha é designada por *segment* que contém elementos chamados *data element*.

4.3.2 ebXML

O ebXML é uma norma para estruturar as trocas de informação entre organizações. Esta norma é suportada pela UN/CEFACT (United Nations Center For Trade Facilitation And Electronic Business) e a OASIS (Organization for the Advancement of Structural Information Standards). Esta norma especifica uma plataforma informacional, para o negócio electrónico, que permite às organizações

encontrarem-se umas às outras e negociarem electronicamente com suporte em mensagens estruturadas em XML. O ebXML permite automatizar as conexões electrónicas entre parceiros comerciais e gerir as suas interacções.

Quando uma empresa quer fazer negócio electrónico com outra empresa segundo a norma ebXML vai seguir 3 etapas:

1. *Implementation Phase*
2. *Discovery of Partner Information and Negotiation Phase*
3. *Transaction Phase*

Na primeira fase, vai haver uma consulta das especificações ebXML para determinados processos de negócio. Após essa consulta e o entendimento dos processos associados, a organização vai implementar os processos escolhidos dentro da sua própria organização segundo a norma. Desta forma vai preparar os seus sistemas de informação para interagir com outras organizações de acordo com os processos descritos no repositório ebXML. Quando o sistema estiver pronto, a organização vai disponibilizar e publicar o seu perfil para que outras organizações interessadas possam também interagir electronicamente com ela. Para isso vai usar o *Collaboration Protocol Profile* (CPP) para publicar o seu perfil de negócio no repositório ebXML.

Na segunda fase, será feito um acordo entre as empresas que queiram colaborar segunda esta norma e para os processos que publicam segundo o CPP. Após verificarem que os perfis e processos estão de acordo com os interesses de cada parte, será elaborado um acordo de cooperação denominado por *Collaborative Partner Agreement* (CPA) que tem a mesma função que um contrato de colaboração.

Na terceira e última fase, será dado início à colaboração electrónica no sentido em que começam a ser trocadas as mensagens XML estruturadas de acordo com a norma ebXML e os CPP/CPA. Toda a lógica de processos e interacção comercial é desta forma automatizada e as organizações colaboram electronicamente.

Para por em prática esta colaboração e integração de SI os *Web Services* podem ser aqui usadas com as normas WSDL, UDDI e SOAP sobre HTTP.

4.4 Business Process Execution Language (BPEL)

O Business Process Management Initiative (BPML.org) e o Workflow Management Coalition (WfMC) juntaram esforços na especificação de três normas para a definição e execução de processos de Workflow: Wf-XML, XPDL e BPML. O Wf-XML é uma variante em XML do WfMC para uma interface de interoperabilidade entre motores de execução de processos de *Workflow*. O *XML Process Definition Language* (XPDL) caracteriza um meta-modelo mínimo de definição portátil de processos de workflow com XML. O Business Process Modeling Language (BPML) é uma meta-linguagem de modelação em XML para os processos organizacionais. Do lado das empresas, a IBM e Microsoft especificaram o BPEL (*Business Process Execution Language for Web Services*) que define uma linguagem para especificar a orquestração de *web services*. O BPEL foi oficialmente submetido pela OASIS em Março de 2003. Tendo em conta que o WfMC está a fazer esforços de convergência entre o XPDL e o BPEL, e sendo o BPEL a norma a que está a prevalecer no mercado inclui-se aqui a sua descrição sumária.

O BPEL é uma norma que permite compor e orquestrar diferentes serviços especificando toda a informação em XML. É possível integrar serviços síncronos e assíncronos segundo uma lógica processual e colaborativa. O BPEL permite conceber e estruturar processos transaccionais que serão executados por sistemas que suportam a automatização de processos definidos em BPEL. Esta norma tem um conjunto de camadas estruturantes como são:

- Web Services e WSDL para definir componentes
- XML como modelo de dados em ambientes *loose-coupling*
- Trocas síncronas e assíncronas de mensagens XML
- Coordenação do fluxo dos processos
- Gestão hierárquica de excepções

- Gestão e compensação de etapas do tipo *long-running*

O BPEL baseia-se nas normas XML e *Web Services* suportando neste contexto as normas SOAP, WSDL, UDDI, WS-Reliable Messaging, WS-Addressing, WS-Coordination e WS-Transaction. Com o BPEL é possível suportar a combinação de aplicações e sistemas acessíveis através dos *Web Services* de duas formas:

- Orquestração
- Coreografia

Os *Web Services* podem ser coordenados segundo uma orquestração que corresponde a uma combinação gerida centralmente numa lógica de processo. Neste sentido, o processo central, que pode também ser ele próprio um *Web Service*, controla a interacção com cada serviço envolvido e coordena a execução das suas diferentes operações. A invocação de cada serviço é da responsabilidade do processo central (Figura 4.17). Esta abordagem é normalmente seguida em processos internos nas organizações.

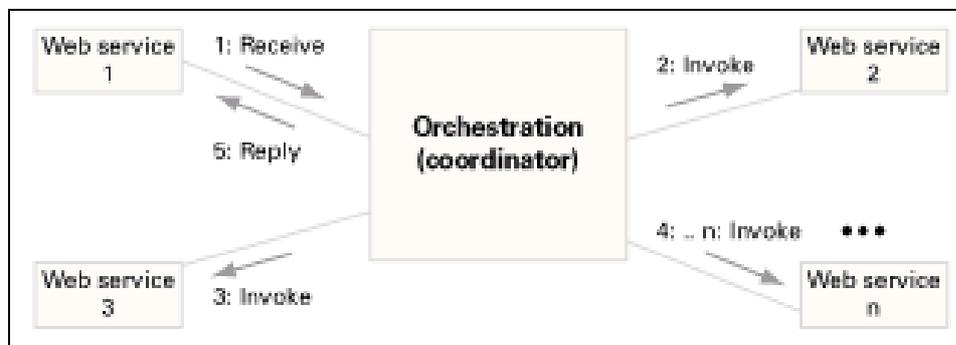


Figura 4.17: Orquestração

Por outro lado, a coreografia dos serviços não está suportada num processo central mas, neste caso, cada serviço está envolvido na própria execução do processo principal e sabe exactamente quando deve executar as suas operações e relacionar-se com os restantes serviços (Figura 4.18). A coreografia consiste numa colaboração organizada entre os serviços e na troca de mensagens. Cada serviço sabe as mensagens XML que deve trocar, quando o deve fazer, que operações deve executar e os processos participa. Embora o BPEL esteja orientado à orquestração, é possível

usar esta especificação para descrever e suportar uma coreografia entre *Web Services*. Este tipo de coreografia pode ser suportada de outras formas, por exemplo, com Java. Face a esta realidade, o consórcio W3C surge com a especificação *Web Services Choreography Definition Language (WS-CDL)* [W3C 2004] que vem complementar o BPEL na área da coreografia dos *Web Services*.

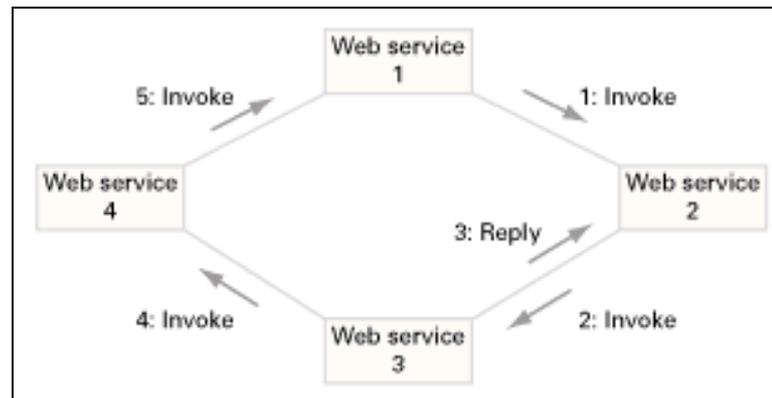


Figura 4.18: Coreografia

Com a norma BPEL é possível seguir duas formas para descrever processos que suportem a orquestração ou a coreografia:

- Processos Executáveis: definição completa de todas as partes de um processo e a sua sequência que será controlada centralmente por um motor de orquestração.
- Protocolos abstractos: permite a especificação pública da troca de mensagens, somente entre cada interveniente, sem a especificação do processo completo mas só parte dele. Esta é a configuração para a coreografia [W3C 2004].

Um processo BPEL consiste em etapas denominadas por ACTIVIDADE. Cada etapa corresponde a uma acção, ou primitiva, que corresponde a uma tarefa básica como são:

- <INVOKE> para invocar um serviço
- <RECEIVE> para esperar por uma invocação externa (ex: receber um pedido)
- <REPLY> geração de uma resposta para operações síncronas

- <ASSIGN> manipulação de variáveis
- <THROW> para indicar uma excepção ou falha
- <WAIT> condição temporal de espera
- <TERMINATE> terminar um processo

Para além destas acções, um processo BPEL suporta estruturas algorítmicas mais complexas como por exemplo:

- <SEQUENCE> conjunto de actividades segundo uma sequência
- <FLOW> fluxos paralelos de execução de actividades
- <SWITCH> condições de execução
- <WHILE> para a definição de ciclos algorítmicos
- <PICK> selecção de fluxos alternativos de processo

É possível complementar estas definições com as tags <PARTNERLINK> e <VARIABLE> para definir links de interligação e variáveis respectivamente.

4.5 Os Web Services

Um Web Service representa uma interface descritivo de um conjunto de funções acessíveis por vários canais heterogéneos. A norma XML é utilizada para a especificação do Web Service que é invocado para executar uma determinada tarefa ou um conjunto de tarefas e obtendo um resultado específico.

O XML é usado como linguagem de base para a especificação dos principais padrões que estruturam os *Web Services*:

- SOAP - Simple Object Application Protocol
- WSDL - Web Service Description Language
- UDDI - Universal Definition Discovery Interface

Um Web Service é descrito através de uma estrutura WSDL que contém os detalhes da interacção que é possível ter-se com ele. Esta descrição contém o formato

das mensagens trocadas e os respectivos protocolos de transporte. O Service Provider ou fornecedor de serviços regista num repositório as definições WSDL para permitir aos sistemas identificar os Web Services e a forma como se interage com cada um. Esse repositório funciona como um directório informativo e é estruturado de acordo com a especificação UDDI. A comunicação estabelecida entre os vários Web Services e as entidades que os invocam é regrada pelo protocolo SOAP que descreve o seu modo de interacção (Figura 4.19). O protocolo SOAP é normalmente utilizado através de invocações no protocolo HTTP, mas poderá suporta-se noutras alternativas.

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Header>
...
</soap:Header>
<soap:Body>
...
<soap:Fault>
...
</soap:Fault>
</soap:Body>
</soap:Envelope>
```

Figura 4.19: Estrutura SOAP

4.5.1 SOAP

O SOAP é um protocolo simples baseado em XML para a troca de informação entre aplicações via o protocolo HTTP. O SOAP é o meio de comunicação onde circulam mensagem simples e é utilizado para aceder a um Web Service e interagir com ele. As aplicações podem normalmente estabelecer um canal de comunicação através dos protocolos CORBA, DCOM ou RPC. Estas formas de interacção não são por vezes eficientes por questões configurações de segurança, de dependências ou de compatibilidade entre as aplicações.

```

POST /rsStock HTTP/1.1
Host: www.stock.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0" ?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.stock.org/stock">
<m:GetStockPrice>
<m:StockName>IBM</m:StockName>
</m:GetStockPrice>
</soap:Body>
</soap:Envelope>

```

Figura 4.20: Mensagem SOAP *Request*

O SOAP utiliza o HTTP que é um protocolo independente e compatível com qualquer *web browser* ou servidor aplicacional. O protocolo HTTP é utilizado como método de comunicação do tipo *request/response* (pedido/resposta) seguindo as regras de estruturação do tipo SOAP. Um pedido SOAP utiliza as regras de *binding* do HTTP que identifica se o pedido é do tipo POST ou GET. Uma mensagem SOAP contém informação estruturada em XML e contém os seguintes elementos:

- SOAP Envelope que identifica a mensagem.
- Header ou cabeçalho com informação.
- Body ou corpo com informação de REQUEST e RESPONSE.
- Fault ou erro com informação descritivas de erros de processamento.

```

HTTP/1.1 200 OK
Content-Type: application/soap; charset=utf-8
Content-Length: nnn<?xml version="1.0" ?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.stock.org/stock">
<m:GetStockPriceResponse>
<m:Price>34.5</m:Price>
</m:GetStockPriceResponse>
</soap:Body>
</soap:Envelope>

```

Figura 4.21: Mensagem SOAP *Response*

Uma mensagem SOAP tem de ser estruturada em XML e não pode conter um DTD ou instruções de processamento XML. Por exemplo, um pedido SOAP de

informação de preço *GetStockPrice* a um determinado servidor, terá como parâmetro *StockName* (Figura 4.20) e receberá a resposta pelo parâmetro *Price* (Figura 4.21). O *namespace* de XML para a função está definida no endereço “http://www.stock.org/stock”.

4.5.2 WSDL

O *Web Service Description Language* (WSDL) descreve os web services e a forma de interagir com eles. As definições dos documentos WSDL são estruturadas em XML e definem onde se encontra o web service e as operações ou métodos que este serviço disponibiliza (Figura 4.22).

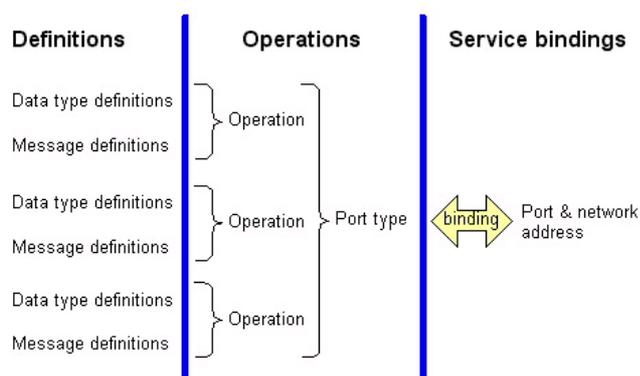


Figura 4.22: Estrutura Lógica WSDL

Um documento WSDL tem elementos do tipo `<type>`, `<message>`, `<portType>` e `<binding>` (Figura 4.23). O elemento `<type>` define o tipo de dados utilizados num web service de acordo com a sintaxe *XML schema* de definição de tipos de dados. O elemento `<message>` identifica os elementos de dados de uma operação funcionando como a descrição dos seus parâmetros. O elemento `<portType>` é o mais importante nesta estrutura definindo o próprio web service, as operações que podem ser feitas e as mensagens associadas. O elemento `<binding>` define o formato da mensagem e o respectivo protocolo para cada *port* (ponto de interação ou operação). O próprio elemento de `<binding>` tem um nome, um estilo (*style*) que pode ser RPC ou DOCUMENT e o protocolo de comunicações que

normalmente é o HTTP. No elemento `<portType>` são definidas todas as operações existentes no webservice e o seu tipo.

```
<definitions>
  <type>
    definition of types .....
  </type>
  <message>
    definition of a message ....
  </message>
  <portType>
    definition of a port .....
  </portType>
  <binding>
    definition of a binding ....
  </binding>
</definitions>
```

Figura 4.23: Estrutura física WSDL

Existem 4 tipos de operações diferentes:

1. One-Way (só recebe uma mensagem)
2. Request-Response (recebe uma mensagem e envia uma resposta) (Figura 4.24)
3. Solicit-Response (envia uma mensagem e espera uma resposta)
4. Notification (envia uma mensagem)

```
<message name="getTerm Request">
  <part name="term" type="xs:string"/>
</message>

<message name="getTerm Response">
  <part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
  <operation name="getTerm">
    <input message="getTerm Request"/>
    <output message="getTerm Response"/>
  </operation>
</portType>
<binding type="glossaryTerms" name="b1">
  <soap:binding style="document"
  transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation>
    <soap:operation
      soap:action="http://example.com/getTerm"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>
```

Figura 4.24: WSDL do tipo *Request-Response*

4.5.3 UDDI

O *Universal Definition Discovery Interface* (UDDI) corresponde a um repositório onde se pode registar ou pesquisar serviços (Figura 4.25). Tem um conjunto de documentos WSDL que permitem identificar e localizar os web services que correspondem aos requisitos de uma determinada funcionalidade.

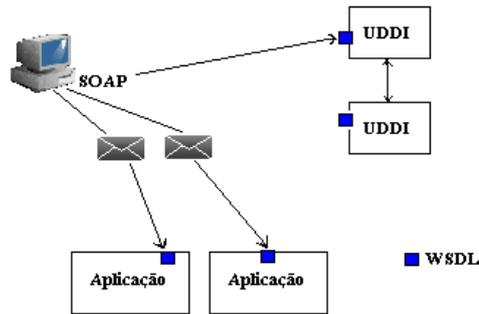


Figura 4.25: UDDI e WSDL

O ciclo de registo e invocação de um webservice tem 5 etapas (Figura 4.26):

1. O *web service* é disponibilizado por um *Service Provider* que o regista num directório ou repositório para o efeito.
2. Ao consultar o repositório
3. Identificar um web service compatível com as funcionalidades desejadas
4. Invocação do *Web Service* directamente ao seu fornecedor.
5. Devolução das respostas pretendidas.

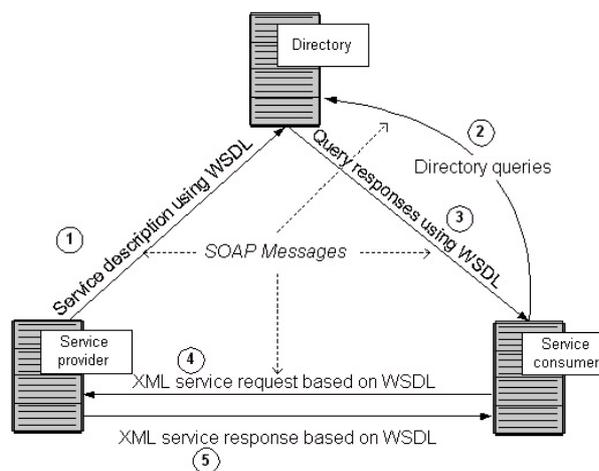


Figura 4.26: Invocação de um *Web Service*

4.6 *Súmula*

As soluções para a integração de SI estão em constante evolução e podem ser aplicadas em diferentes situações. Neste capítulo, foram identificadas, e sumariamente explicadas, algumas das soluções e normas mais comuns nesta área. O principal objectivo foi o de clarificar o papel de cada uma delas e enquadrá-las na área da integração de SI.

No que diz respeito à integração da informação, a norma XML é utilizada em diferentes soluções tecnológicas, permitindo manusear e trocar a informação segundo um padrão independentemente. Para a integração aplicacional existem diversas formas de integração assentes na codificação e no seu processamento. Ainda neste âmbito, existem soluções que se enquadram nas tecnologias de suporte a transacções, ao *Messaging* e aos *Web Services*. No que toca à integração aplicacional entre diferentes organizações foram realçadas as normas EDI e ebXML. Para a integração orientada aos processos foi identificada a norma BPEL sendo, neste momento, uma das especificações com mais destaque no mercado tecnológico para a integração de SI numa lógica de processo. Foram também incluídas neste capítulo as principais normas relativas aos *Web Services*: SOAP, UDDI e WSDL.

5. Implementação de um exemplo de integração de SI com duas abordagens

Neste capítulo é implementado um exemplo de integração de SI identificando as normas seguidas e as soluções técnicas adoptadas. Este exemplo conta com duas implementações diferentes onde se procura demonstrar como se pode combinar diversas tecnologias para conseguir uma integração completa e em diferentes níveis. A primeira abordagem baseia-se num conjunto de soluções mostrando o papel de cada uma na integração de SI. A segunda abordagem aproveita partes da primeira e é complementada com o uso da norma BPEL e de *Web Services*. Desta forma ilustra-se, por um lado, a criação de níveis de abstracção de aplicações e informação, e por outro lado, o papel destas normas no aumento da flexibilidade na integração de SI. Trata-se de um caso hipotético e genérico, mas suficientemente abrangente permitindo abarcar várias formas pontuais de integração.

O exemplo consiste na criação de uma aplicação que necessita de integração com outras 2 aplicações já existentes numa organização. Pretende-se disponibilizar uma nova aplicação, para a gestão de requisições internas, tirando partido da informação e funcionalidades que já existem nas aplicações de gestão de *stocks* e de gestão de recursos humanos. Dado que o principal propósito da implementação deste exemplo é a ilustração de diferentes abordagens de integração, cada aplicação aqui descrita é muito simples, e realça somente os componentes ou a informação que necessita de ser acedida ou integrada.

A abordagem técnica deste exemplo de integração está baseada em várias tecnologias e linguagens, que permitem diferentes níveis de integração abordados nos capítulos anteriores, como são:

- Oracle Database Enterprise Edition 10g
- Oracle Workflow 2.6.3
- Oracle Forms Developer 10g e Oracle Jinitiator
- Java
- Oracle PL/SQL

- SQL e Oracle SQL Plus
- Oracle Advanced Queueing para Messaging Middleware
- Apache, Oracle Container for Java e JDBC
- Case Studio 2.15
- Web Services
- XML
- Oracle BPEL Process Manager 10g

Dado que se está a integrar uma nova aplicação com 2 já existentes, o exemplo integra no seu todo 3 aplicações em concreto. A implementação deste exemplo tem duas partes onde se mostra uma primeira abordagem para a integração das 3 aplicações e se detalha uma segunda abordagem baseada na norma BPEL descrita neste trabalho. No final deste capítulo são tecidas algumas conclusões acerca destas duas implementações.

De acordo com o vasto leque de alternativas tecnológicas, haveria muitos cenários possíveis e diferenciados para a implementação deste caso de integração. Foram definidos dois cenários que permitem mostrar duas alternativas em concreto para este exemplo. Estes cenários foram encontrados tendo como objectivo mostrar a implementação técnica de :

- Uma arquitectura a 3 níveis de uma aplicação.
- O uso de *Messaging Middleware*.
- A automatização de um processo com *workflow*.
- O uso de pelo menos uma *Application Programming Interface* (API).
- A integração aplicacional e de informação ao nível da camada da apresentação.
- Uso de uma das plataformas de desenvolvimento, neste caso o J2EE.
- O uso de uma componente de *Application Server*, neste caso, o OC4J.
- A implementação de BPEL, XML e *Web Services*.

Este capítulo está dividido em quatro partes. Na primeira parte é feita uma breve descrição da estrutura do exemplo e de cada aplicação envolvida. Ao longo

desta primeira descrição detalha-se também, e de forma mais pormenorizada, cada aplicação, a lógica do processo de requisições e as formas de integração implementadas. Na segunda parte do capítulo inclui-se uma descrição passo a passo, seguindo todo o processo de aprovação de requisições, e mostrando ponto a ponto os tipos de integração adoptados. Numa terceira parte, é igualmente feita uma descrição passo a passo da segunda abordagem de integração seguida. Neste caso, enquadra-se a utilização de alternativas para esta integração e detalha-se a sua implementação. Na quarta e última parte, é feita uma breve sùmula da implementação deste exemplo e são tecidas algumas conclusões.

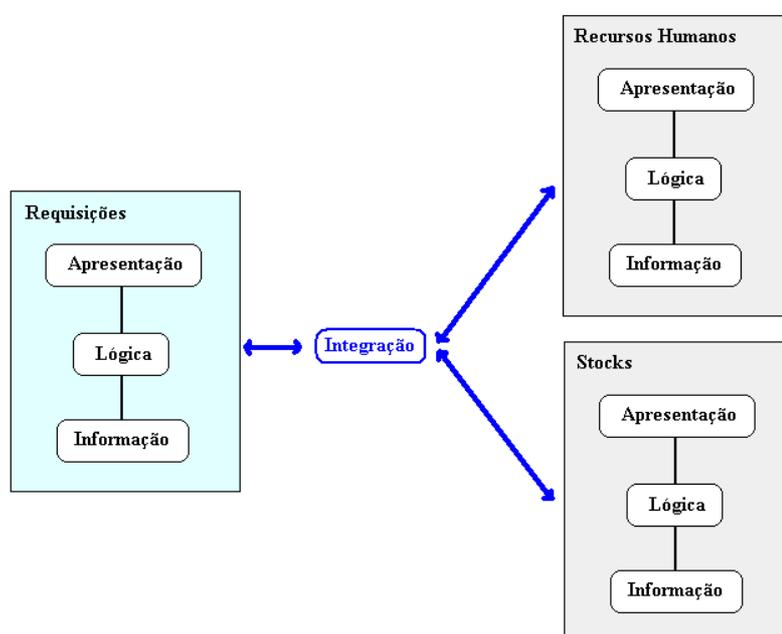


Figura 5.1: Integração das 3 aplicações

5.1 Estrutura do exemplo

A ideia principal consiste na integração de 3 aplicações ao longo de um processo organizacional que automatiza a aprovação de requisições internas de bens. As 3 aplicações a integrar são: aplicação de Requisições, aplicação de Recursos Humanos (RH) e a aplicação de Gestão de Stocks (Stocks) (Figura 5.1). A aplicação de Requisições permite criar e submeter requisições a um processo de *workflow* para aprová-las, a aplicação de Stocks faz a gestão de armazem e aviso de entrega desses

mesmos bens e a aplicação de gestão de Recursos Humanos gere toda a informação relativa aos funcionários assim como os respectivos departamentos.

A aplicação de Requisições assenta na arquitectura a 3 níveis descrita no capítulo 2. Neste sentido, esta pode ser dividida em três partes complementares e correspondentes ao nível do cliente, ao nível da lógica aplicacional e ao nível da informação respectivamente (Figura 5.2). O primeiro nível trata de camada de apresentação desta aplicação que pode ser acedida via *web browser*. O segundo nível trata da execução da lógica aplicacional e baseia-se num servidor aplicacional. O terceiro e último nível trata da disponibilização e armazenamento da informação.

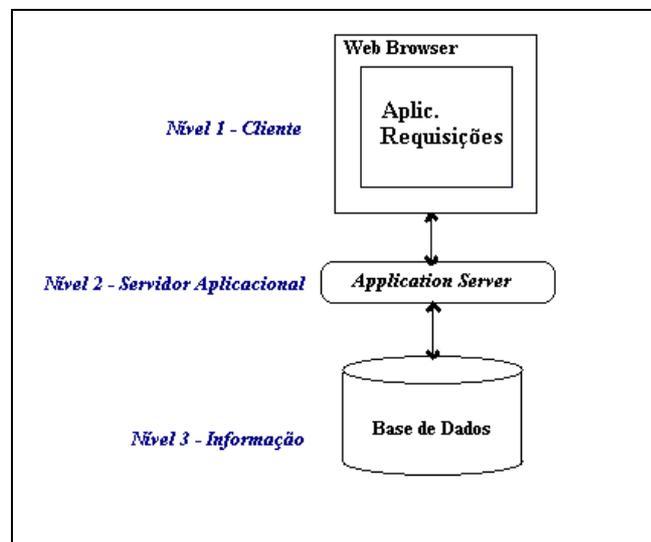


Figura 5.2: Aplicação de Requisições em 3 níveis

Procura-se neste caso criar diferentes tipos de integração descritos nos capítulos anteriores cobrindo a integração de informação, a integração aplicacional, a integração de processos e a utilização de uma alternativa baseada nos *Web Services* e no BPEL.

5.1.1 Aplicação de Requisições

A aplicação de Requisições é programada no ambiente Oracle Forms Developer 10g e segue uma arquitectura a 3 níveis. Esta aplicação é composta (Figura 5.3) pela interface do utilizador acedida por Web Browser, e baseada no Oracle

Jinitiator, por um nível intermédio de execução da lógica aplicacional baseado no servidor aplicacional Oracle Application Server 10g, nas suas componentes OC4J e no Forms Server, e por fim por um nível de repositório de informação disponibilizado pelo motor de base de dados Oracle 10g cujo nome é “BD_REQ”.

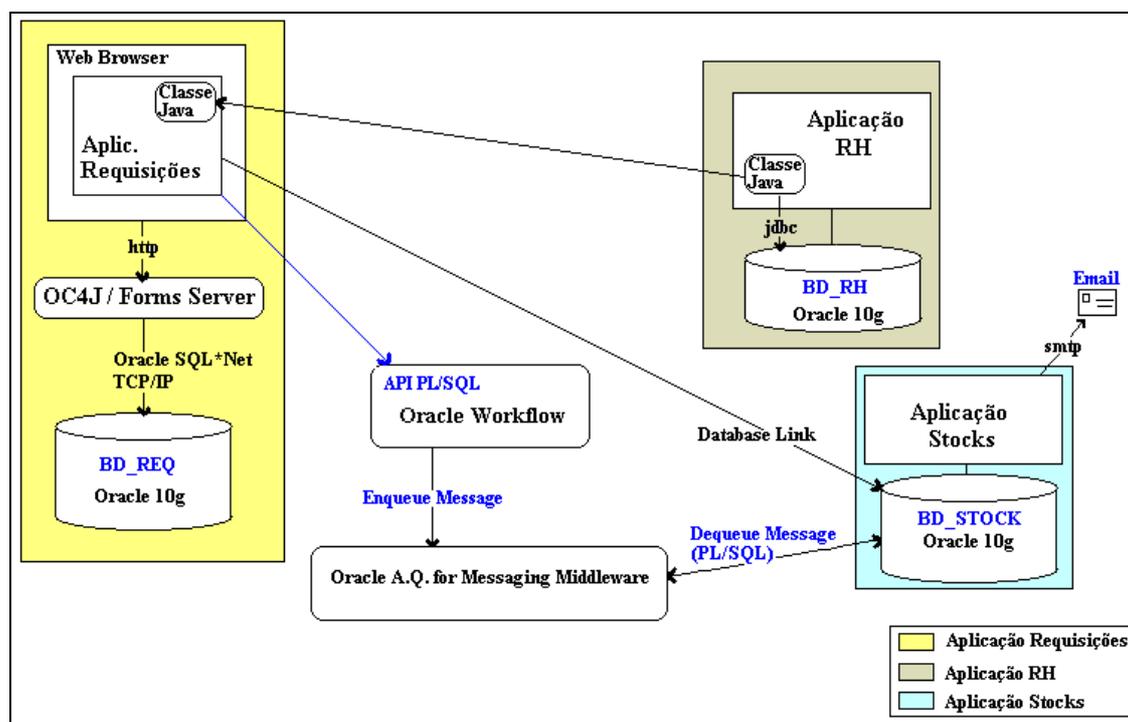


Figura 5.3: Esquema de integração da primeira abordagem

O facto de se poder aceder a esta aplicação via protocolo HTTP torna possível a sua integração ao nível da camada de apresentação por exemplo num Portal. Nesta aplicação foi embutido um objecto programado em Java, que usa a API (*Application Programming Interface*) do Oracle Forms e permite integrar parte da aplicação de Recursos Humanos (RH) de forma a validar os utilizadores e complementar a sua informação ao nível da sua hierarquia (Figura 5.3). Ainda nesta aplicação, é feita uma integração de informação através de um *Database Link* que permite obter dados de outro repositório, neste caso para validar os tipos de bens (ITEM) passíveis de requisição assim como a sua descrição. Desta forma, é possível apresentar informação completa num único écran aos utilizadores da respectiva aplicação após o seu *login*.

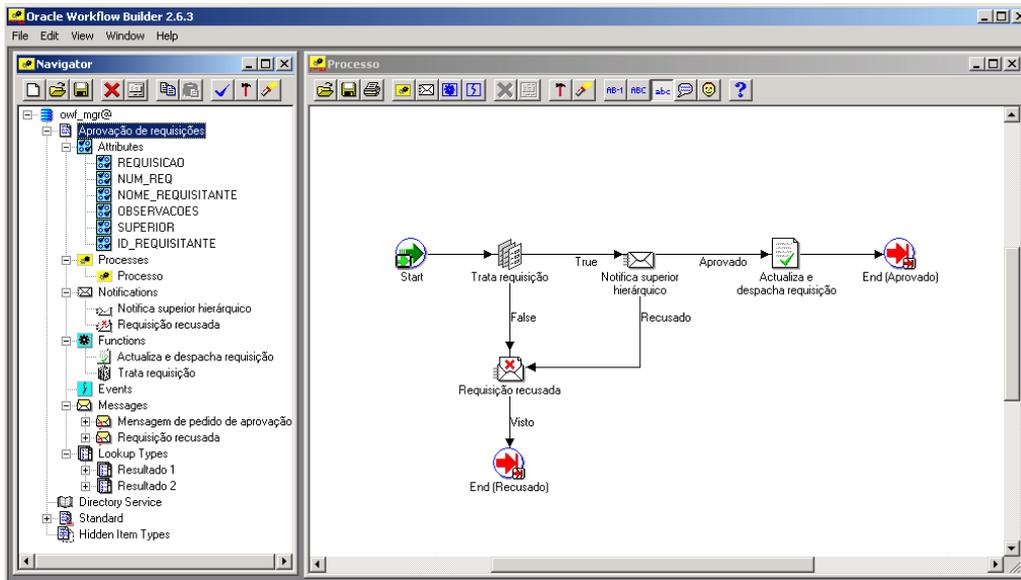


Figura 5.4: Definição do processo de Workflow para suportar aprovação de requisições

Esta aplicação de Requisições integra-se também com um software de *workflow* que suporta o processo de aprovação de pedidos de aprovação para as requisições submetidas. Para este exemplo é utilizada a *Application Programming Interface* (API) em PL/SQL para integrar com a aplicação de Requisições no seu nível de lógica aplicacional (Figura 5.3). O processo implementado recebe da aplicação de Requisições a informação relativa ao bem requisitado, à pessoa que fez essa requisição e ao seu superior. Para dar início ao processo, a aplicação de Requisições invoca o procedimento armazenado “REQ_PLSQL.INICIA_PROCESSO” para o qual passa os parâmetros do número da requisição (NUM_REQ), do utilizador que efectua a requisição (NOME_REQUISITANTE e ID_REQUISITANTE) e do superior hierárquico (SUPERIOR). Após ser iniciado o processo tem 4 etapas principais (Figura 5.4):

- **Trata Requisição:** Este é um ponto de integração ao nível da informação, e onde o processo de workflow vai verificar nas bases de dados das aplicações de Requisições e RH a descrição do tipo de bem requisitado e da identificação do departamento do utilizador em causa. Aqui é implementada uma regra da organização em que no caso do bem requisitado ser do tipo 1 (tipo DESKTOP), e do utilizador pertencer ao departamento número 1 (Vendas), a requisição será automaticamente rejeitada.

- **Notifica Superior Hierárquico:** Esta etapa corresponde à notificação do superior hierárquico da pessoa que efectuou a requisição.
- **Requisição Recusada:** Esta etapa corresponde à notificação da pessoa que efectuou a requisição informando que o pedido de requisição foi rejeitado, automática ou deliberadamente.
- **Actualiza e Despacha Requisição:** Neste ponto a requisição foi aprovada. Esta etapa integra-se com a componente de Messaging deste exemplo para integrar com a aplicação de Stocks. Neste ponto é feito o *enqueue* de uma mensagem contendo a informação da requisição para ser enviada para a aplicação de Stocks que irá tomar conta da requisição aprovada.

5.1.2 Aplicação de Recursos Humanos

A aplicação de Recursos Humanos já existente na organização permite fazer a gestão dos seus empregados. Esta aplicação permite manter uma base de dados denominada “BD_RH” que contem toda a informação dos funcionários, funções, departamentos e as hierarquias de funcionários existentes. As tabelas existentes na base de dados desta aplicação são: FUNCIONARIOS, FUNCOES e DEPARTAMENTOS.

```

C:\WINNT\system32\cmd.exe - sqlplus bd_rh/bd_rh@orcl
SQL>
SQL>
SQL> DESC FUNCIONARIOS
Nome
-----
ID_FUNCIONARIO          NOT NULL  VARCHAR2(10)
ID_DEPARTAMENTO         NOT NULL  VARCHAR2(5)
ID_FUNCAO              NOT NULL  VARCHAR2(5)
NOME_FUNCIONARIO       NOT NULL  VARCHAR2(50)
DATA_CONTRATO          NOT NULL  DATE
ID_SUPERIOR            VARCHAR2(10)

SQL>
SQL>
SQL>
SQL> SELECT * FROM FUNCIONARIOS;
ID_FU  ID_DE  ID_FU  NOME_FUNCIONARIO  DATA_CON  ID_SUPERIO
-----
1      1      2      VICTOR            05.07.15   4
2      2      2      CESAR             99.12.17
3      2      1      CARLOS            99.01.20
4      1      1      EDUARDO           96.09.10
5      3      1      MARIA             00.03.11
SQL>

```

Figura 5.5: Informação contida na tabela FUNCIONARIOS da base de dados BD_RH

Para este exemplo, as requisições foram criadas com o utilizador “VICTOR”, cujo número de funcionário é o 1, e cujo superior hierárquico é o funcionário de nome “EDUARDO” com o número de funcionário 4 (Figura 5.5). Esta aplicação disponibiliza uma API em Java (valida_bd_rh.class) que permite validar quem o superior hierárquico de um determinado funcionário, assim como o nome completo desse funcionário, bastando para isso enviar o parâmetro do número de funcionário (Figura 5.6).

```

valida_bd_rh.java - Notepad
File Edit Format Help
import oracle.jdbc.driver.*;
import java.awt.*;
import java.sql.*;
import java.net.*;
public class valida_bd_rh extends Canvas{
private String data;
public valida_bd_rh()
{ try { DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
} catch (SQLException e){
// Tratamento de erros.
System.out.println("Erro de SQL."); } }
public static String valida_utilizador(String data)
{ Connection conn;
ResultSet rset = null; Statement stmt = null; String valor = null;
System.out.println("Inicio conexão Java à BD aplicação RH");
try { String connect_string="jdbc:oracle:thin:@martins-pt.pt:oracle.com:1521:ORCL";
conn = DriverManager.getConnection (connect_string,"bd_rh","bd_rh");
stmt = conn.createStatement ();
String queryString = "select a.id_funcionario,a.nome_funcionario,b.nome_funcionario
from funcionarios a, funcionarios b
where a.id_superior = b.id_funcionario
and a.nome_funcionario='VICTOR'";
rset = stmt.executeQuery(queryString);
rset.next();
System.out.println("-----");
System.out.println(" ID Funcionario -> " + rset.getString(1));
System.out.println(" Nome Funcionario -> " + rset.getString(2));
System.out.println(" Nome Superior -> " + rset.getString(3));
valor=rset.getString(1);
rset.close();
stmt.close();
System.out.println ("Fecha conexão Aplicação RH.");
conn.close();
System.out.println ("Envia valores para Forms.");
}
catch (Exception e)
{
System.out.println ("Erro.");
e.printStackTrace();
}
return(valor); }
}

```

Figura 5.6: Classe Java “Valida_BD_RH”

5.1.3 Aplicação de Stocks

Esta aplicação tem como propósito gerir o armazem de bens existentes na organização. Para isso tem as tabelas ITEMS, TIPOS_ITEMS, ITEMS_EM_STOCK e FORNECEDORES (Figura 5.7). Esta aplicação tem como função disponibilizar informação acerca dos bens existentes em stock e gerir a entrega dos bens requisitados na aplicação de Requisições. Neste último ponto, existe a integração entre a aplicação de Stocks e a aplicação de Requisições através de um software de *Messaging* que permite uma troca de mensagens informando das requisições aprovadas.

1. Integração aplicacional ao nível da camada de apresentação (Ponto 0): Acesso à aplicação de Requisições através de *Web Browser* na intranet da organização.
2. Integração de Informação (Ponto 1): Entre Aplicação Requisições e RH via componente Java que acede à base de dados de RH via JDBC.
3. Integração aplicacional (Ponto 2): Integração de uma classe Java com lógica aplicacional da aplicação de RH e da aplicação de Requisições programada em Oracle Forms. Este ponto permite validar informação sobre a pessoa que efectua a requisição através da aplicação de RH.
4. Integração de informação (Ponto 3): Através de um *Database Link* a aplicação de Requisições valida informação acerca dos bens requisitáveis na base de dados da aplicação de Stocks.
5. Integração aplicacional e de processos (Ponto 4): Integração da aplicação de Requisições com o software de *workflow* através da sua API. O *software* de *workflow* permite também integrar numa lógica de processo as aplicações de Requisições e de Stocks.
6. Integração aplicacional (Pontos 5 e 6): Integração do software de *workflow* com a o software de *Messaging* que se encarrega da troca de mensagens entre as aplicações de Requisições e de Stocks.
7. Integração aplicacional (Ponto 7): Integração da aplicação de Stocks com o correio electrónico com o protocolo *Simple Mail Transfer Protocol* (SMTP).

5.1.5 Processo de aprovação de requisições

Após a criação de uma requisição válida na aplicação de Requisições, os utilizadores podem submeter as suas requisições a um sistema de *workflow* específico que automatiza o processo de aprovação das mesmas, envolvendo as respectivas hierarquias de funcionários. A informação necessária é veiculada pelo processo para facilitar as tomadas de decisão de quem aprova este tipo de pedidos. Neste âmbito, foi criada uma regra que proíbe qualquer funcionário do departamento de Vendas de requisitar bens do tipo “DESKTOP” cuja chave de identificação é o valor “1”.

Caso a requisição seja aprovada o sistema de *workflow* vai criar uma mensagem, com os dados da requisição aprovada, que submete a um *Middleware* de assente na tecnologia Oracle Advanced Queueing 10g. Esta camada de *software* vai encarregar-se de entregar com sucesso à aplicação de Stocks uma mensagem estruturada que representa a requisição aprovada. Trata-se de um nível de integração entre aplicações suportado em *Middleware* do tipo *Messaging*. Para isso, é feita uma operação de “Enqueue” da mensagem na respectiva *queue* para esta seja entregue ao sistema destino através de uma operação do tipo “Dequeue”. Sendo este processo de trocas de mensagens executado com sucesso, a aplicação de Stocks tem a informação que identifica a necessidade de satisfazer uma requisição. Para isso é feita uma consulta à base de dados que suporta a informação dos bens existentes em armazém, e quando o bem esteja disponível é enviado um email ao respectivo funcionário que efectuou a requisição. Trata-se neste caso de uma integração suportada no protocolo SMTP que permite o envio de informação, neste caso concreto, do envio de um email que permite fechar o ciclo do processo de submissão de requisições.

5.1.6 Modelo E-R para o exemplo

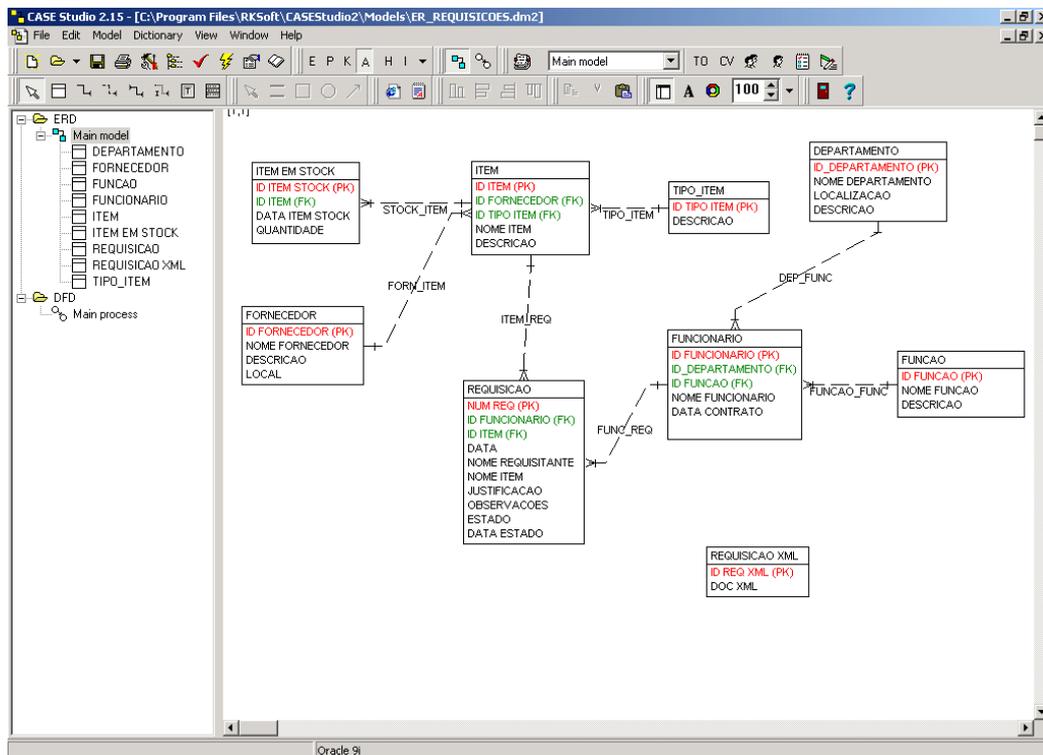


Figura 5.9: Modelo E-R do exemplo.

Este modelo Entidade-Relacionamento define as entidades e respectivos atributos para a implementação das tabelas na base de dados utilizada para a implementação deste exemplo (Figura 5.9). As entidades utilizadas para este exemplo são: ITEM, ITEM_EM_STOCK, TIPO_ITEM, DEPARTAMENTO, FORNECEDOR, REQUISICAO, FUNCIONARIO e FUNCAO.

5.2 Descrição funcional da primeira abordagem de integração

Neste ponto é feita uma descrição funcional de cada componente do exemplo e dos passos seguidos para a criação e submissão de uma requisição.

5.2.1 Aplicação de Requisições em Oracle Forms Developer 10g

A aplicação de Requisições foi desenvolvida no ambiente de programação proprietário Oracle Forms 10g (Figura 5.10). Trata-se de um écran aplicativo que é integrado ao nível da camada de apresentação num ambiente *web*. A partir deste écran é possível criar e submeter uma requisição cuja informação é:

- N° de requisição.
- Data de requisição.
- Identificação do funcionário que cria a requisição.
- Nome do funcionário que cria a requisição.
- Identificação do bem (ITEM) a requisitar.
- Nome do bem a requisitar.
- Justificação da requisição.

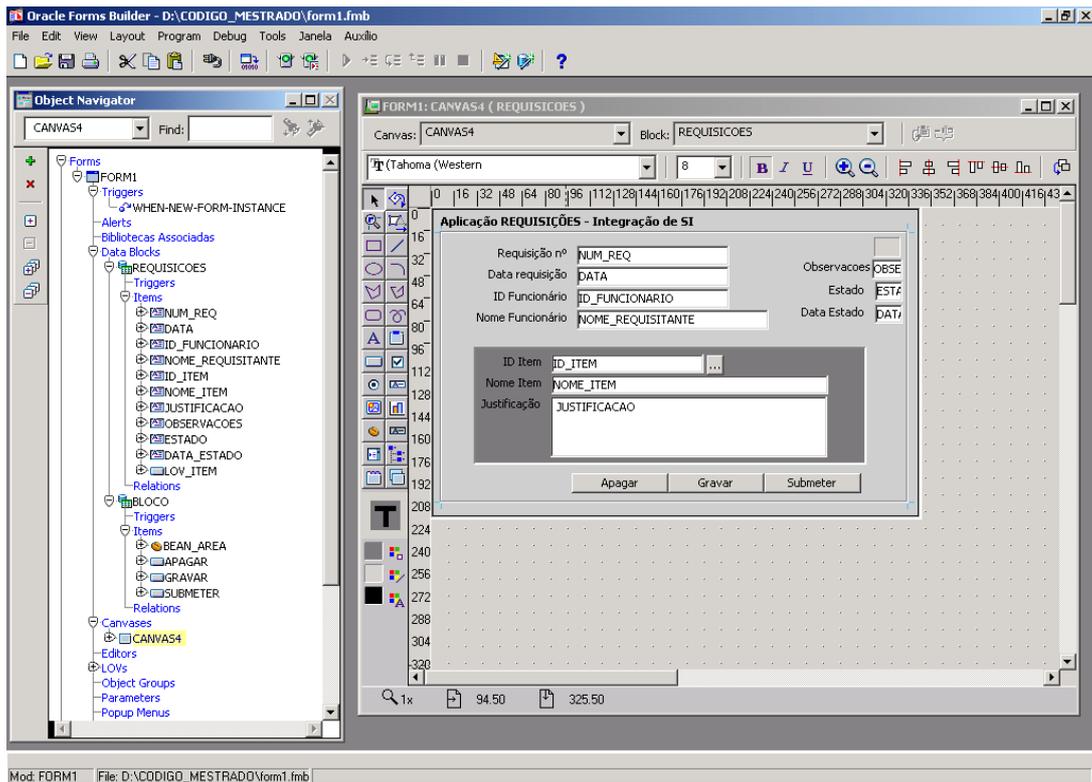


Figura 5.10: Écran de introdução da requisição da aplicação de Requisições

Nesta aplicação é integrada uma classe de Java pertencente à aplicação de RH (Figura 5.11). Para isso é usado o objecto “BEAN AREA” do Oracle Forms.

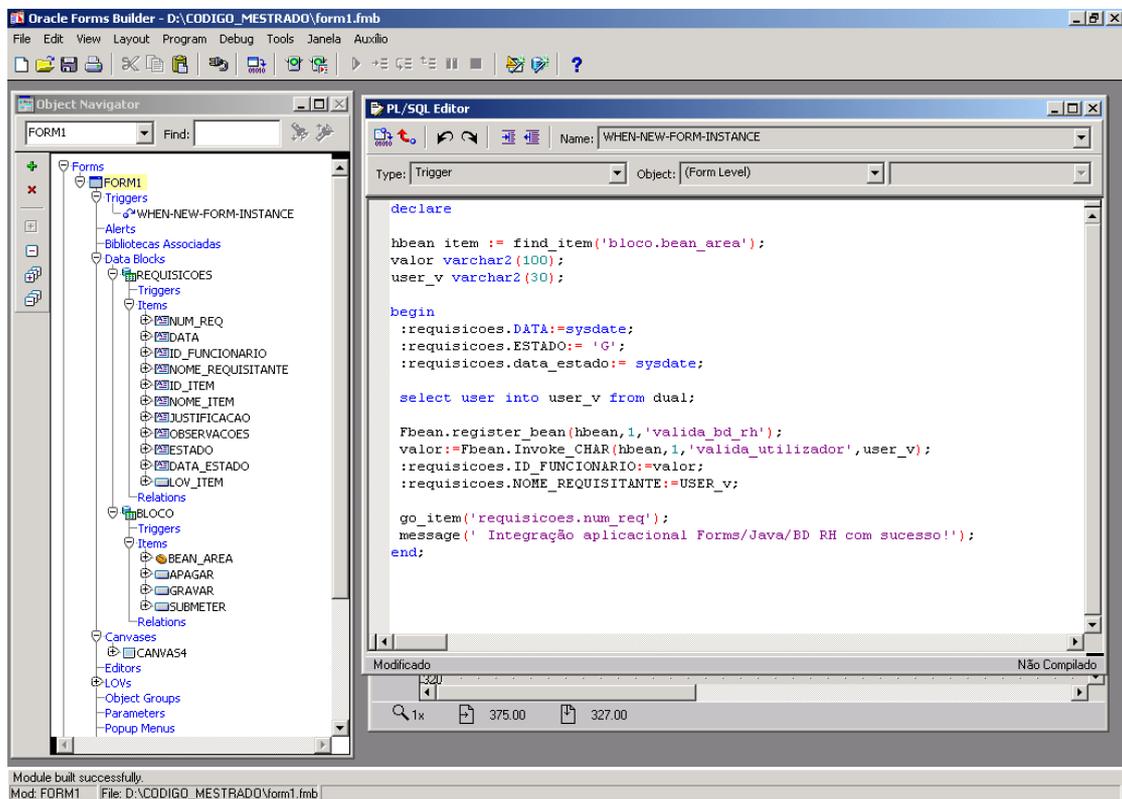


Figura 5.11: Integração de código Java na aplicação de Requisições

Esta aplicação também se integra ao nível da camada de informação com a aplicação de Stocks através de um *Database Link* (DBL_BD_STOCK) para obter informação detalhada acerca dos bens requisitáveis (Figura 5.12).

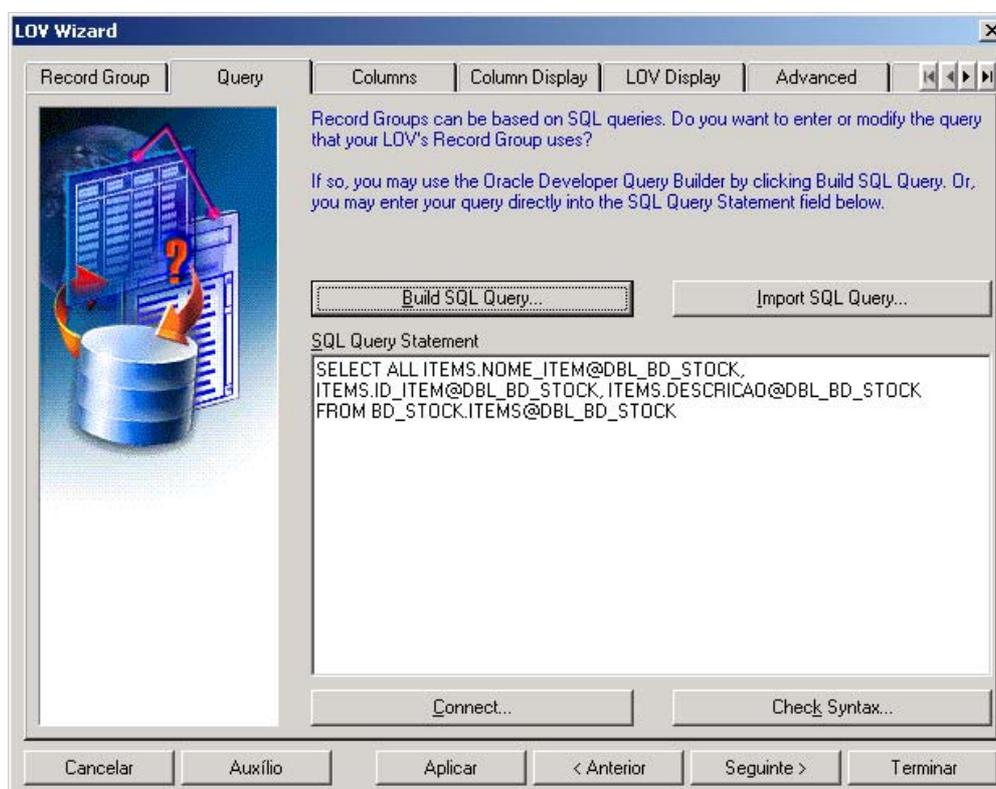


Figura 5.12: Integração de informação via Database Link na aplicação de Requisições

5.2.2 Acesso à aplicação de Requisições e sua integração com Java

O acesso à aplicação de Requisições é feito via Web Browser e através da Intranet da respectiva organização (Figura 5.13). O login é feito com o utilizador “VICTOR”. Esta situação ilustra uma integração ao nível da camada de apresentação.

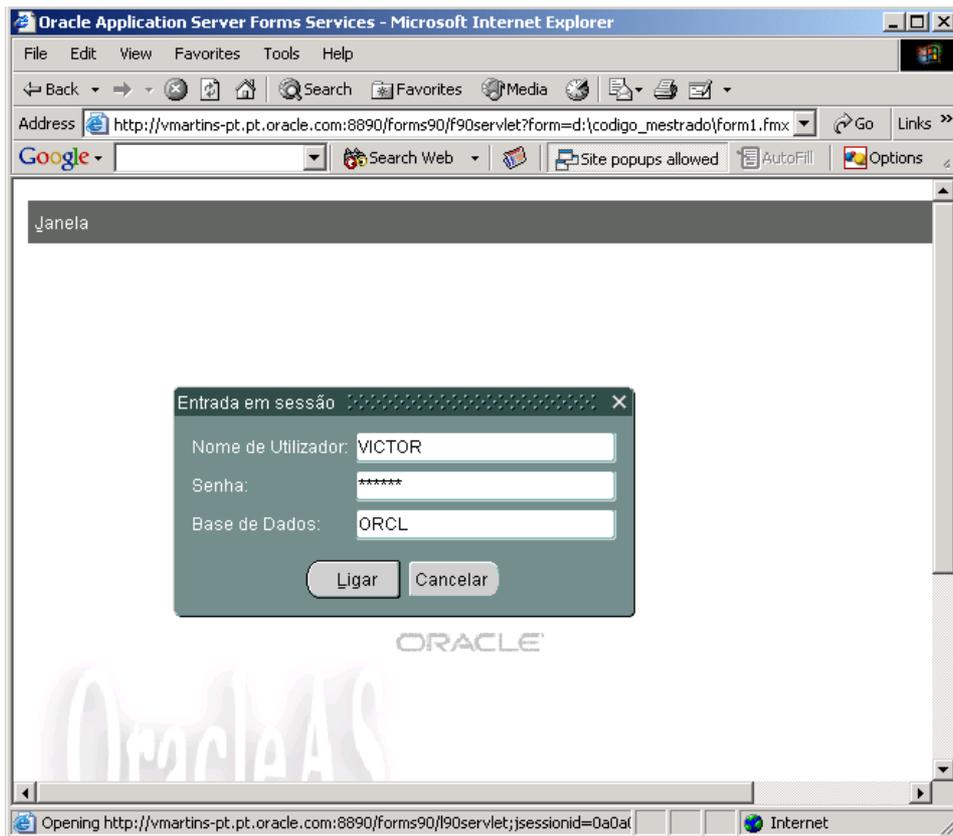


Figura 5.13: Acesso via Web Browser à aplicação de Requisições com utilizador VICTOR

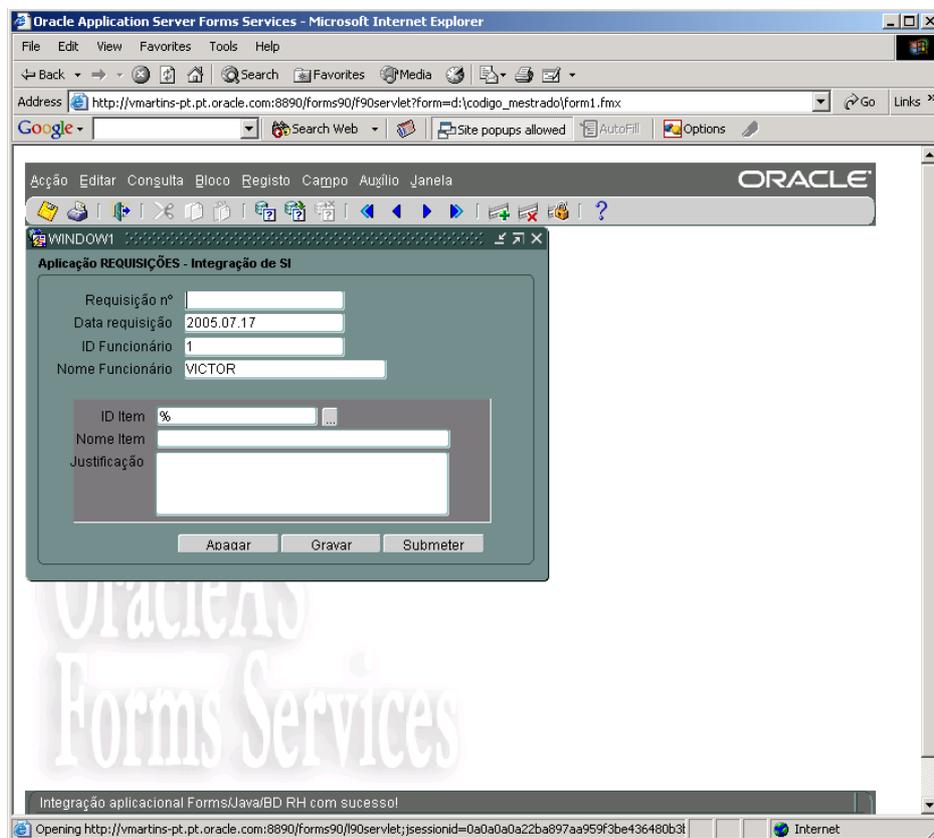


Figura 5.14: Écrã de inserção e submissão das requisições

Ao entrar na aplicação de Requisições o campo do nome do utilizador é automaticamente preenchido (Figura 5.14). Entretanto já foi automaticamente obtida toda a informação acerca do superior hierárquico do utilizador que se autenticou na aplicação com a integração do pedaço de código Java pertencente à aplicação de RH que retorna a respectiva informação (Figura 5.15).

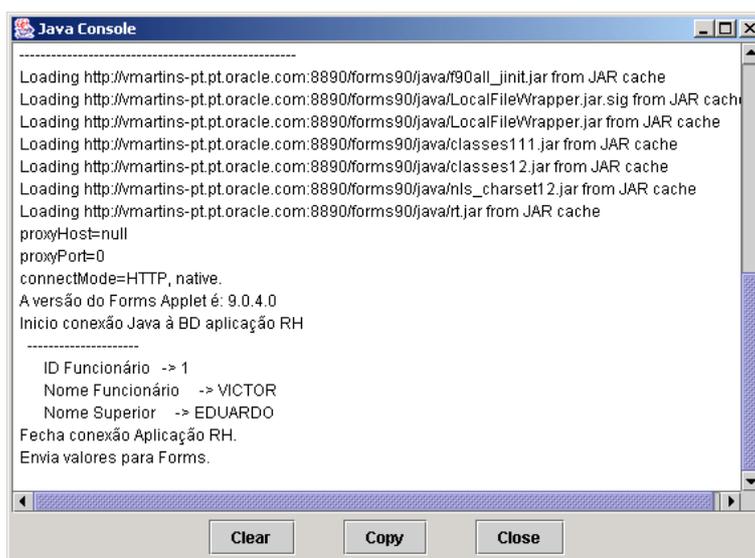


Figura 5.15: Informação da integração da aplicação de RH na aplicação de Requisições com Java

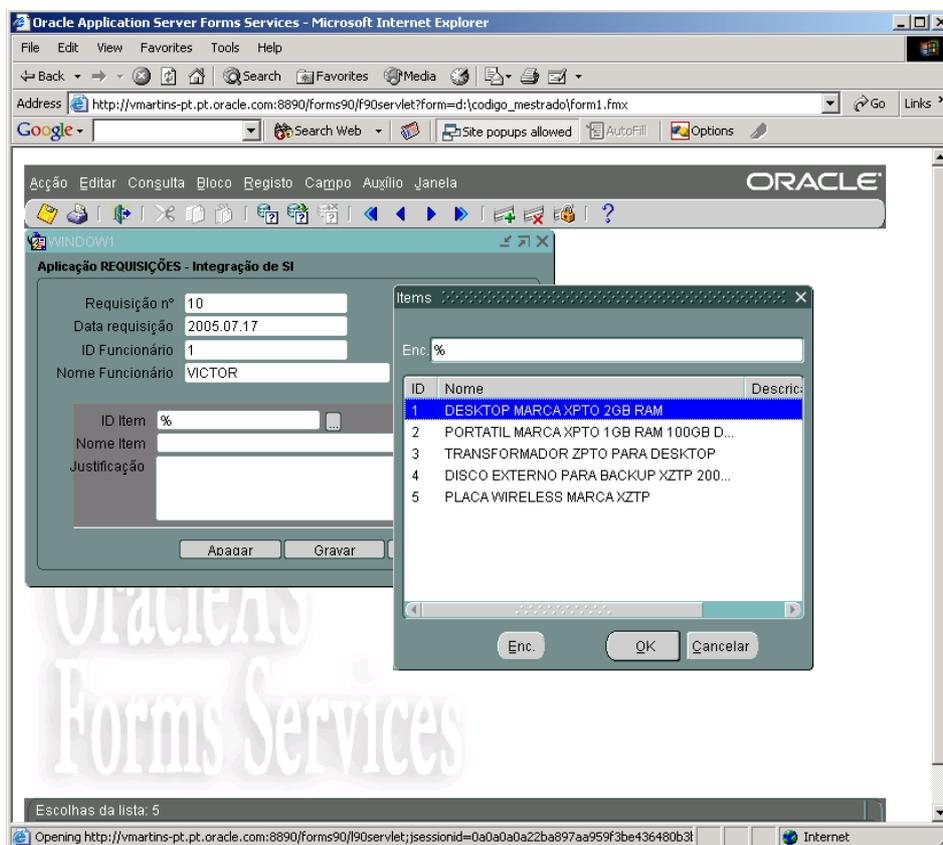


Figura 5.16: Consulta de informação da aplicação de Stocks na aplicação de Requisições

Durante a execução da aplicação de Requisições é feita a integração ao nível da informação com a aplicação de Stocks através de um Database Link que fornece informação a uma lista de valores acerca dos bens requisitáveis (Figura 5.16).

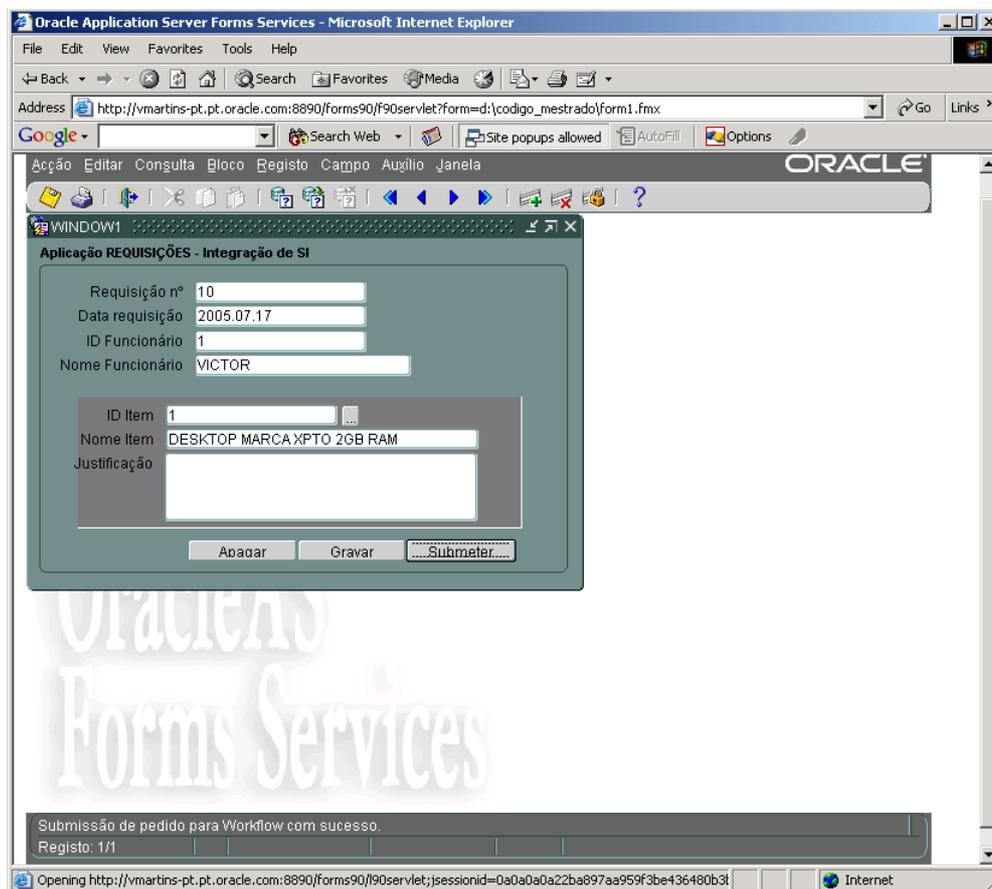
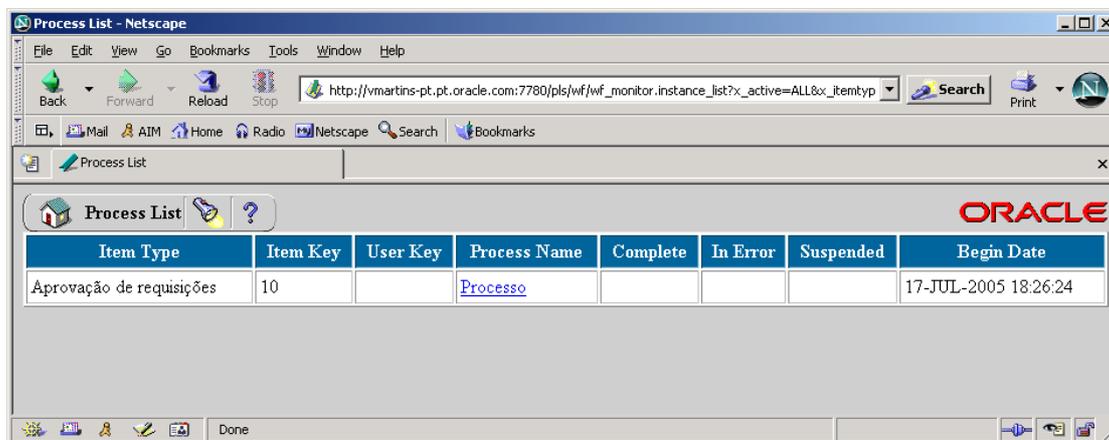


Figura 5.17: Submissão da requisição nº 10 do item nº1 para Workflow com sucesso.

Ao criar e submeter a requisição na aplicação de requisições é mostrada uma mensagem informativa na parte inferior da aplicação que indica se a integração com o software de workflow foi feita com sucesso ou não. Neste caso foi submetido o pedido de requisição nº 10 para o um computador do tipo desktop com o utilizador Victor que pertence ao departamento de Vendas (Figura 5.17).

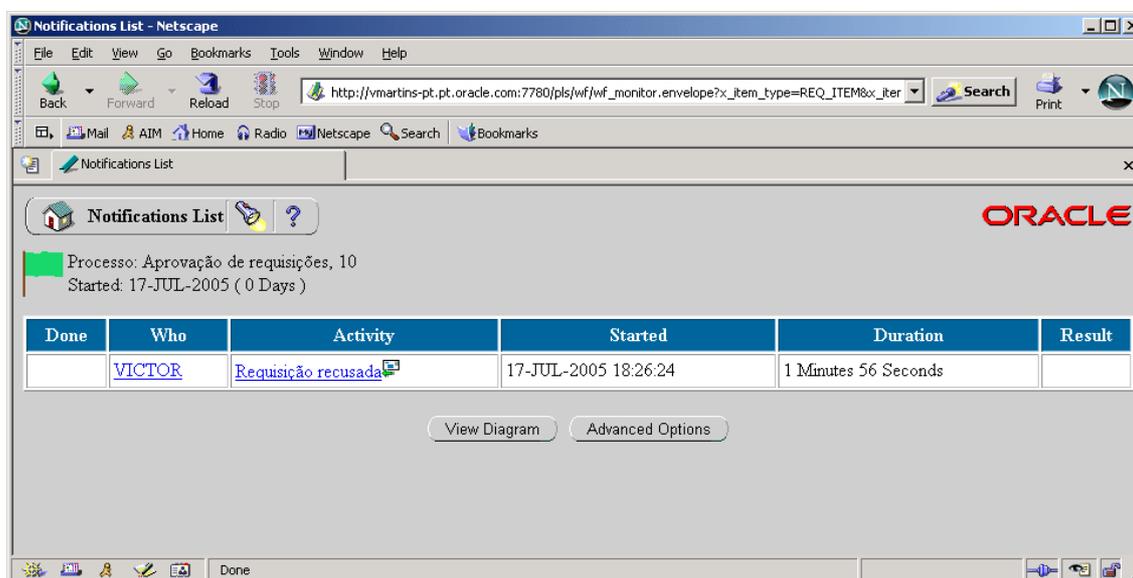
5.2.3 Integração da aplicação de Requisições com solução de *workflow*



Item Type	Item Key	User Key	Process Name	Complete	In Error	Suspended	Begin Date
Aprovação de requisições	10		Processo				17-JUL-2005 18:26:24

Figura 5.18: Processo de aprovação de requisição nº 10 iniciado

Ao submeter a requisição é possível acompanhar a execução do processo de workflow e retirar informação para a sua monitorização. O processo para a requisição nº 10 foi iniciado (Figura 5.18) e aplicou automaticamente a regra de restrição para os bens do tipo “computadores desktop” requisitados por pessoas pertencentes ao departamento de Vendas (Figura 5.19).



Processo: Aprovação de requisições, 10
Started: 17-JUL-2005 (0 Days)

Done	Who	Activity	Started	Duration	Result
	VICTOR	Requisição recusada	17-JUL-2005 18:26:24	1 Minutes 56 Seconds	

[View Diagram](#) [Advanced Options](#)

Figura 5.19: Requisição de item nº 1 recusada automaticamente (Victor do departamento de Vendas)

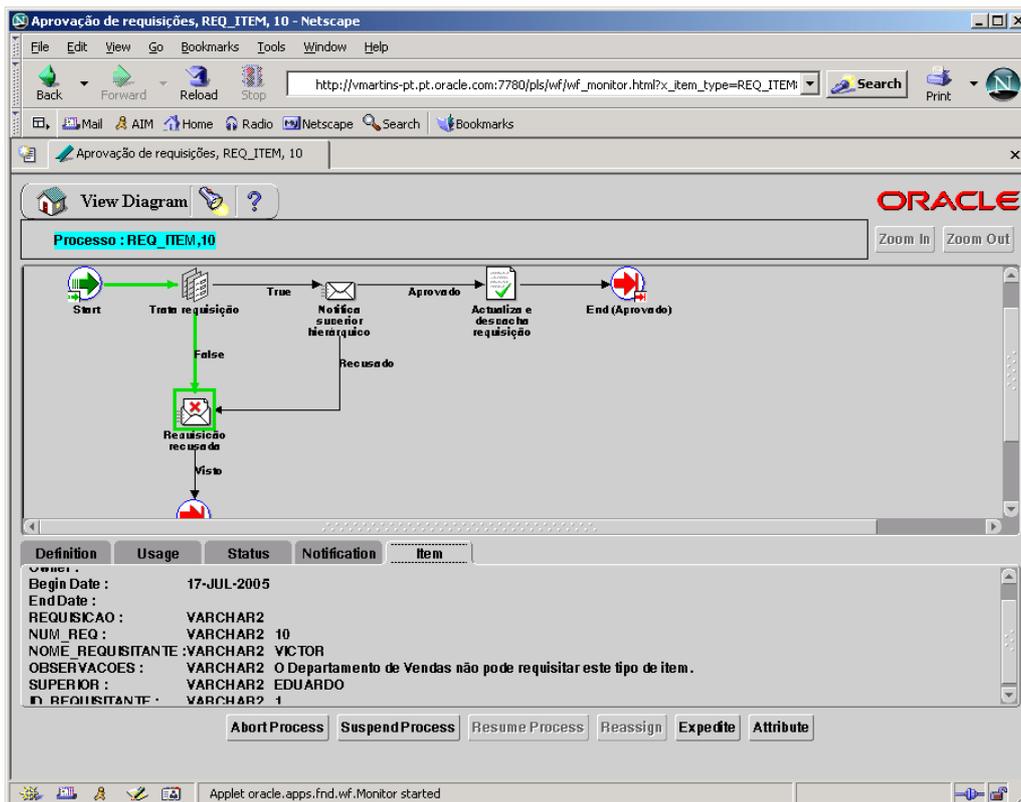


Figura 5.20: Processo de requisição nº 10 recusado.

Esta regra automaticamente aplicada pelo *software* de *workflow* também informa automaticamente o utilizador da razão pela qual houve essa recusa preenchido o campo das observações (Figura 5.20). Esta regra é implementada num procedimento armazenado na base de dados que influencia o caminho seguido pelo processo de *workflow* retornando o resultado verdadeiro (T) ou falso (F) consoante a informação contida na requisição criada (Figura 5.21).

```

...
if (tipo_item_v='1' and id_departamento_v='1' ) then
    wf_engine.SetItemAttrText (itemtype => itemtype,
                              itemkey => itemkey,
                              aname => 'OBSERVACOES',
                              avalue => 'O Departamento de Vendas
                                         não pode requisitar este
                                         tipo de item. ');
    resultout := 'COMPLETE:F';
else
    resultout := 'COMPLETE:T';
end if;
...

```

Figura 5.21: Regra de negócio aplicada no *workflow*

Para efeitos de ilustração é criada uma nova requisição nº11 com o mesmo utilizador mas desta vez com um bem do tipo “computador portátil” com a identificação nº2 (Figura 5.22).

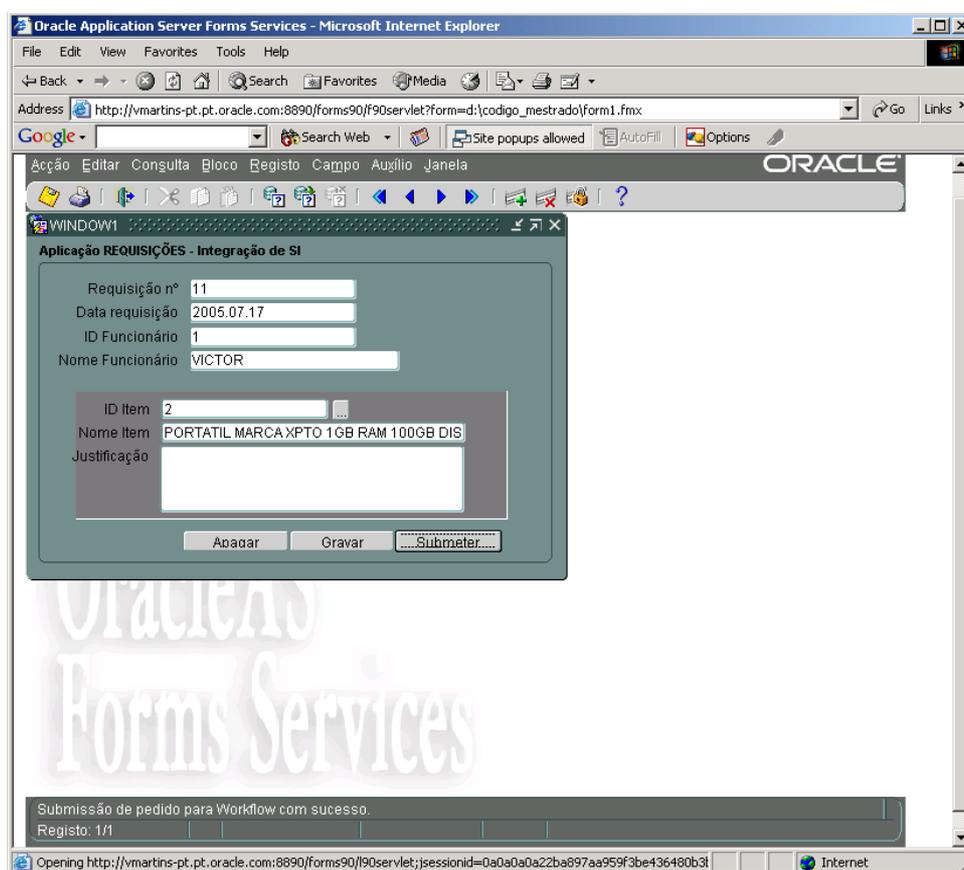


Figura 5.22: Submissão da requisição nº 11 para item nº2

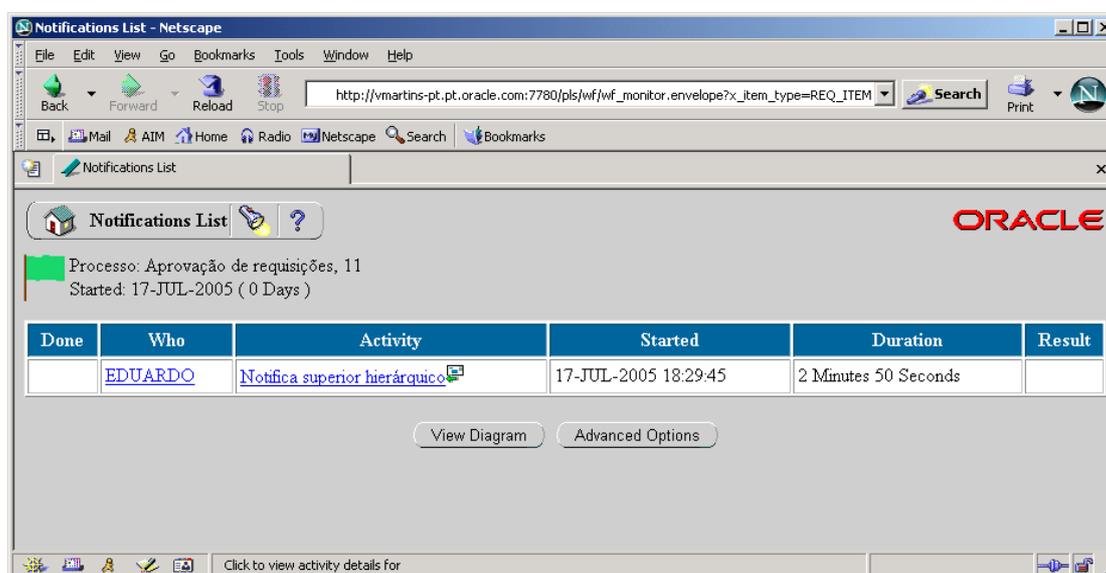


Figura 5.23: Requisição nº 11 em aprovação no workflow

Dado que esta requisição nº11 não colide com a regra automática descrita anteriormente, o processo de workflow segue o seu caminho normal (Figura 5.23) e o superior hierárquico (EDUARDO) da pessoa que criou a requisição (VICTOR) é notificado para dar o seu parecer (Figura 5.24).

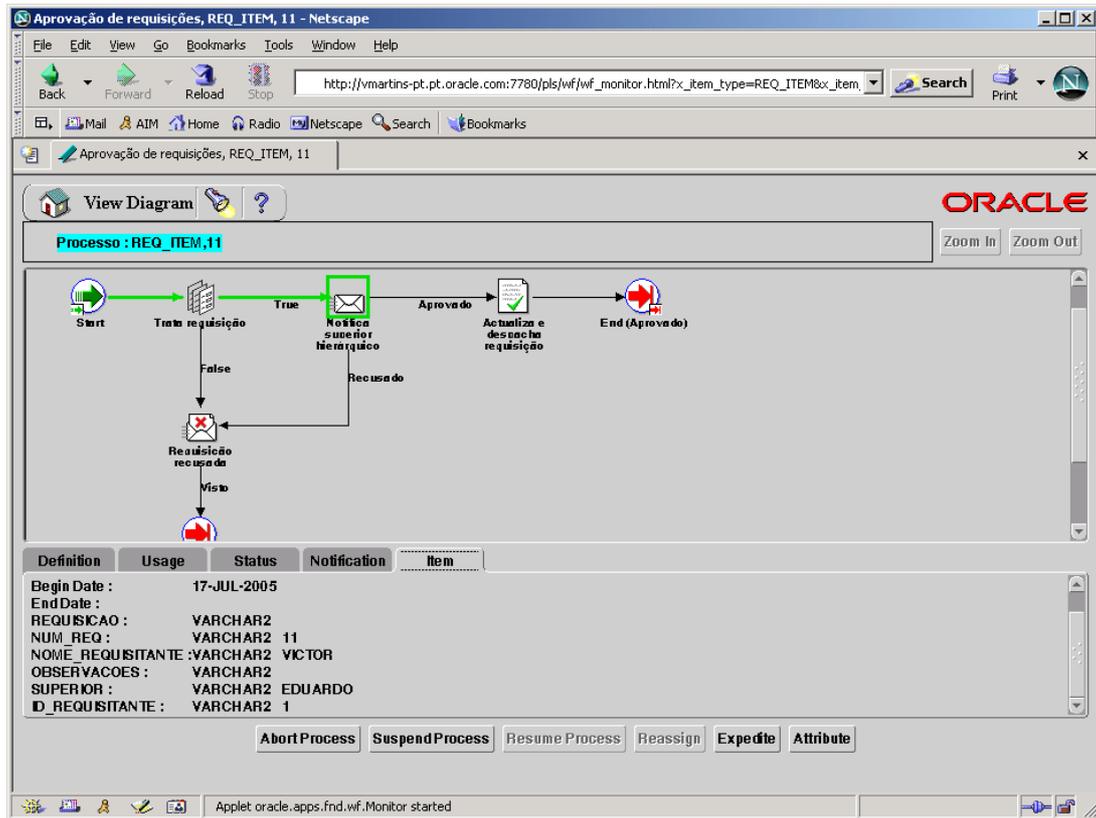


Figura 5.24: Notificação para pedido de aprovação de requisição nº11

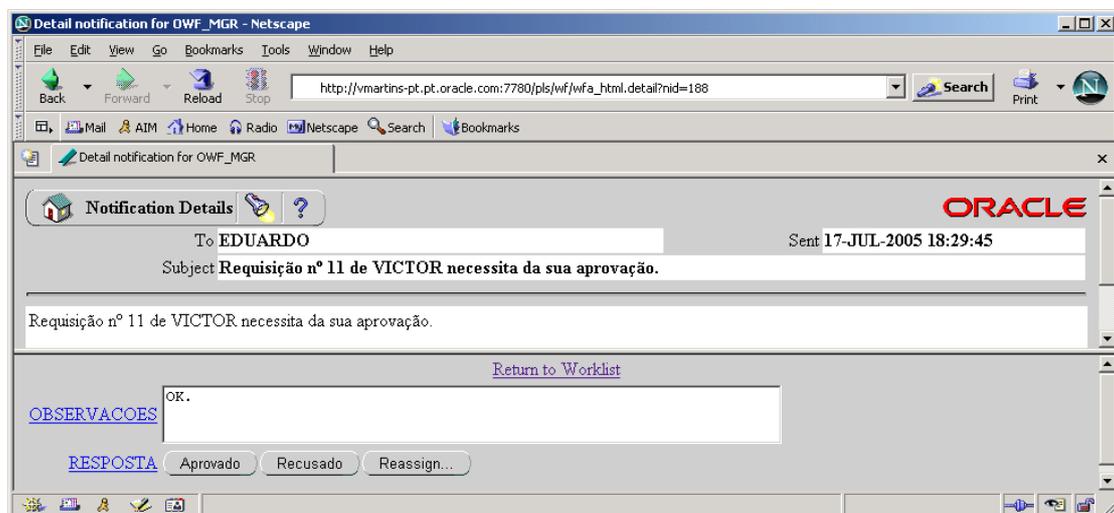


Figura 5.25: Aprovação da requisição nº11 no workflow.

O utilizador EDUARDO vai poder dar o seu parecer acerca da requisição nº11 submetida pelo utilizador VICTOR (Figura 5.25). Neste caso só se mostra a informação acerca do número da requisição e da pessoa que a submeteu. Como o objectivo é mostrar a especificamente a integração entre as diversas aplicações em cada uma das perspectivas descritas neste trabalho não foi incluído na notificação informação mais detalhada acerca da requisição. Para este caso a requisição foi aprovada (Figura 5.26).

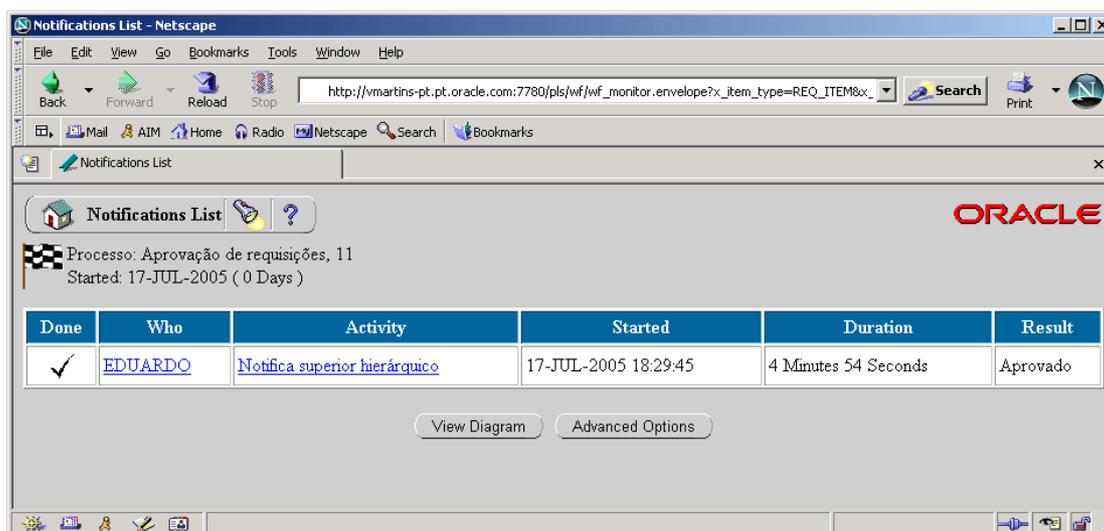


Figura 5.26: Término de processo de aprovação no workflow da requisição nº 11.

Tendo sido aprovada a requisição, o processo de workflow chegou a seu fim mas integrando-se entretanto com a componente de *Messaging* utilizada neste exemplo para enviar uma mensagem à aplicação de Stocks informando da existência de uma requisição aprovada (Figura 5.27).

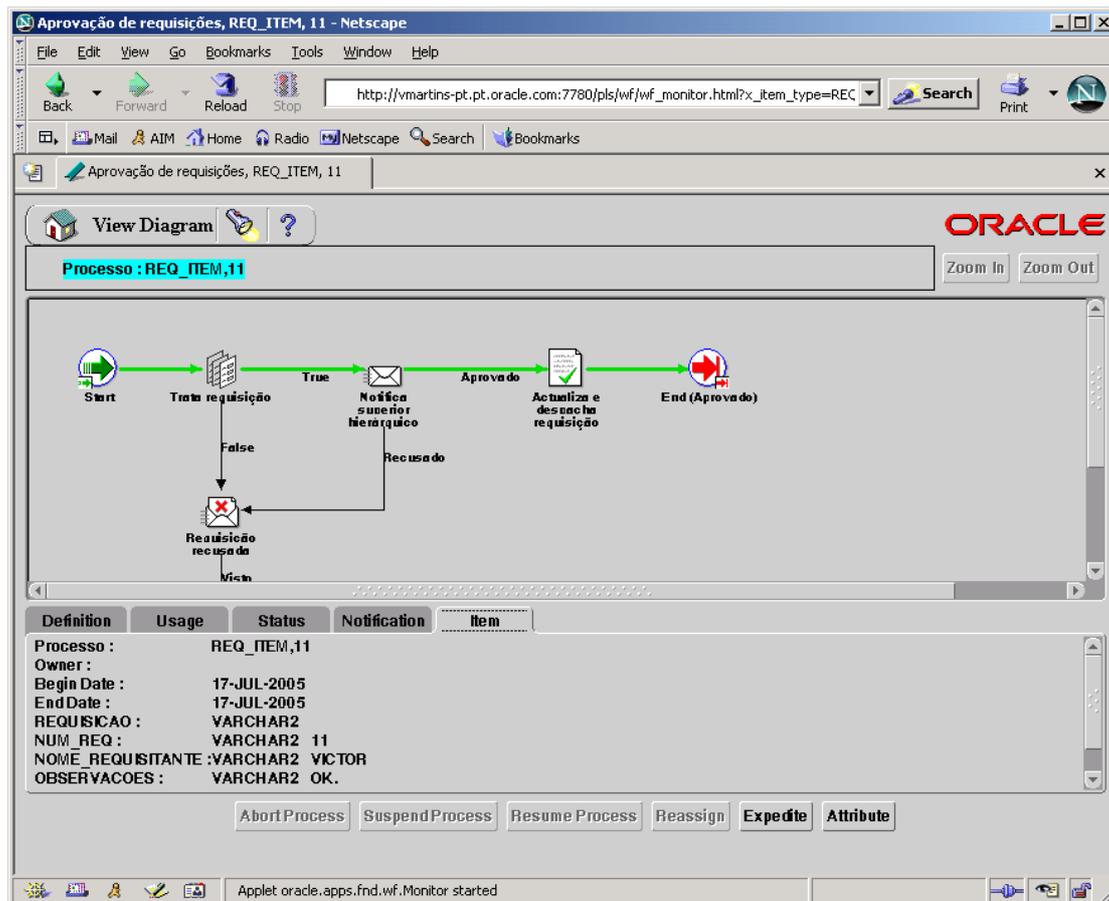


Figura 5.27: Término de processo de aprovação no workflow da requisição nº 11.

5.2.4 Integração da aplicação de Requisições com solução de Messaging

A integração do *workflow* com Oracle Advanced Queueing 10g é feita através de código. Neste fase é enviada a mensagem fazendo o *enqueue* dela no *Middleware* de *Messaging* através da invocação do procedimento `BD_REQ.SUBMETER_MENSAGEM_AQ`, passando como parâmetros o número da requisição, a identificação do bem, a data da requisição e o nome do bem (Figura 5.28).

```

.....
procedure DESPACHA_REQ (itemtype          in varchar2,
                        itemkey           in varchar2,
                        actid             in number,
                        funcmode          in varchar2,
                        resultout         out varchar2) IS

num_req_v varchar2(30);

BEGIN
-- Nesta fase é feito o enqueue no AQ Messaging --

num_req_v := wf_engine.GetItemAttrText(itemtype => itemtype,
                                       itemkey => itemkey,
                                       aname  => 'NUM_REQ' );

bd_req.submeter_mensagem_aq (NUM_REQ_v, ID_ITEM, DATA, NOME_ITEM);
.....

```

Figura 5.28: Integração com *Messaging*

O procedimento invocado efectua uma operação de ENQUEUEE (*DBMS_AQ.ENQUEUEE*) no sistema Oracle Advanced Queueing 10g enviando a mensagem destinada à aplicação *BD_STOCK*, que representa a integração com a aplicação de Stocks, através da fila de mensagens (*queue*) *Q_1* (Figura 5.29).

```

create or replace procedure submeter_mensagem_aq (NUM_REQ in VARCHAR2,
                                                ID_ITEM in VARCHAR2,
                                                DATA in DATE,
                                                NOME_ITEM in VARCHAR2)

is

enqueue_options          DBMS_AQ.enqueue_options_t;
message_properties       DBMS_AQ.message_properties_t;
recipients               DBMS_AQ.aq$_recipient_list_t;
message_handle           RAW(16);
message                  Requisicao_t;
NUM_REQ_V VARCHAR2(10);
ID_ITEM_V VARCHAR2(10);
DATA_V DATE;
NOME_ITEM_V VARCHAR2(50);

BEGIN

message := Requisicao_t(NUM_REQ_V, ID_ITEM_V, DATA_V, NOME_ITEM_V);

recipients(1) := sys.aq$_agent('BD_STOCK', NULL, NULL);

message_properties.recipient_list := recipients;

DBMS_AQ.ENQUEUEE(queue_name => 'BD_REQ_Q_1',
enqueue_options => enqueue_options,
message_properties => message_properties,
payload => message,
msgid => message_handle);

END;

```

Figura 5.29: Envio de mensagem no *Messaging*

```
C:\WINNT\system32\cmd.exe - sqlplus bd_stock/bd_stock@orcl
SQL>
SQL>
SQL>
SQL>
SQL> /
Corpo de pacote criado.
SQL> EXECUTE TRATA_REQUISICOES_AQ.RECEBE_REQUISICAO_MESSAGING;
11 2 05.07.17 PORTATIL MARCA XPT0 1GB RAM 100GB DISCO
Procedimento de PL/SQL concluído com êxito.
SQL>
```

Figura 5.30: Recepção da mensagem na aplicação de Stocks com operação de DEQUEUE.

A aplicação de Stocks faz uma operação de leitura de mensagens invocando a API do sistema de *Messaging* com a operação “DEQUEUE” que está no código “TRATA_REQUISICOES_AQ.RECEBE_REQUISICAO_MESSAGING” (Figura 5.30). É possível neste caso visualizar o conteúdo da mensagem nº11.

```
C:\WINNT\system32\cmd.exe - sqlplus bd_stock/bd_stock@orcl
SQL>
SQL>
SQL>
SQL> execute trata_req.envia_email_item_disponivel('11');
Procedimento de PL/SQL concluído com êxito.
SQL> commit;
Confirmação concluída.
SQL>
```

Figura 5.31: Envio de email

Após validar que o bem existe em stock é feita a integração com o sistema de email da organização através do protocolo de *Simple Mail Transport Protocol* (SMTP) e da API da base de dados “UTL_SMTP” incorporada no código “TRATA_REQ.ENVIA_EMAIL_ITEM_DISPONIVEL” (Figura 5.31). Neste caso é enviado um *email* directamente para a pessoa que requisitou o bem informando-a que o item requisitado já está disponível para ser levantado (Figura 5.32).

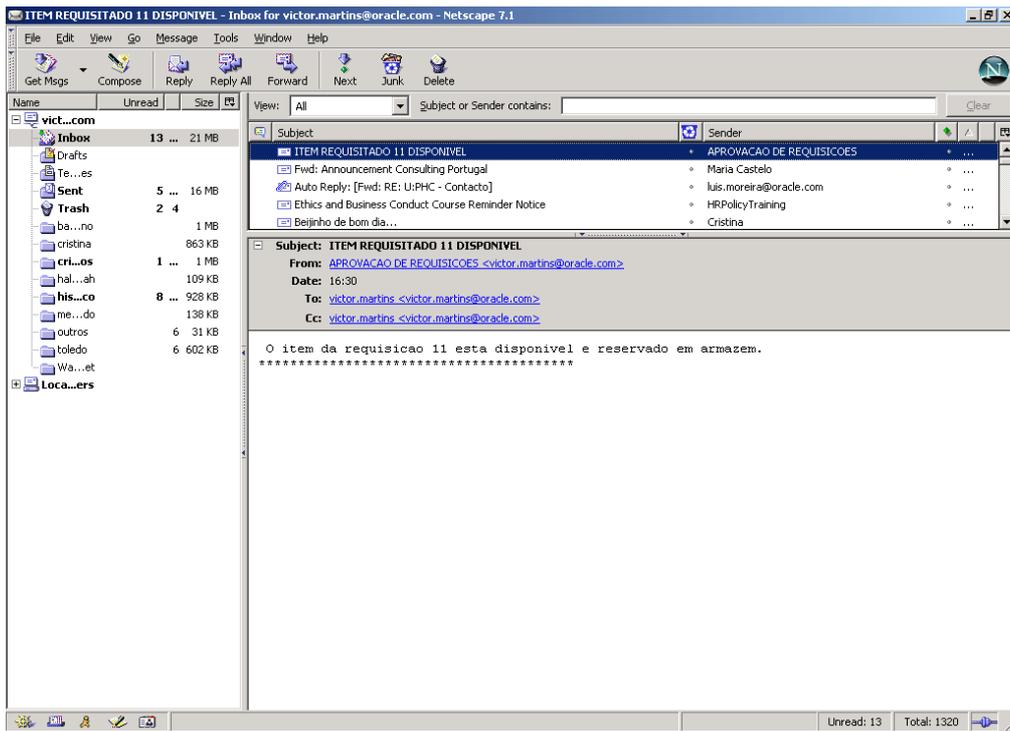


Figura 5.32: Recepção de email

5.3 Descrição funcional da segunda abordagem de integração com BPEL

A lógica de integração das 3 aplicações segue os mesmos princípios e objectivos dos pontos anteriores, mas neste caso, a implementação suporta-se na norma BPEL que permite um nível de integração mais flexível e que tira partido das normas associados aos *Web Services*, nomeadamente o SOAP e o WSDL (Figura 5.33). Para este caso foram criados 2 *Web Services*, que permitem disponibilizar com esta tecnologia respectivamente partes da aplicação de Requisições e do sistema de *workflow* que suporta a aprovação das requisições submetidas.

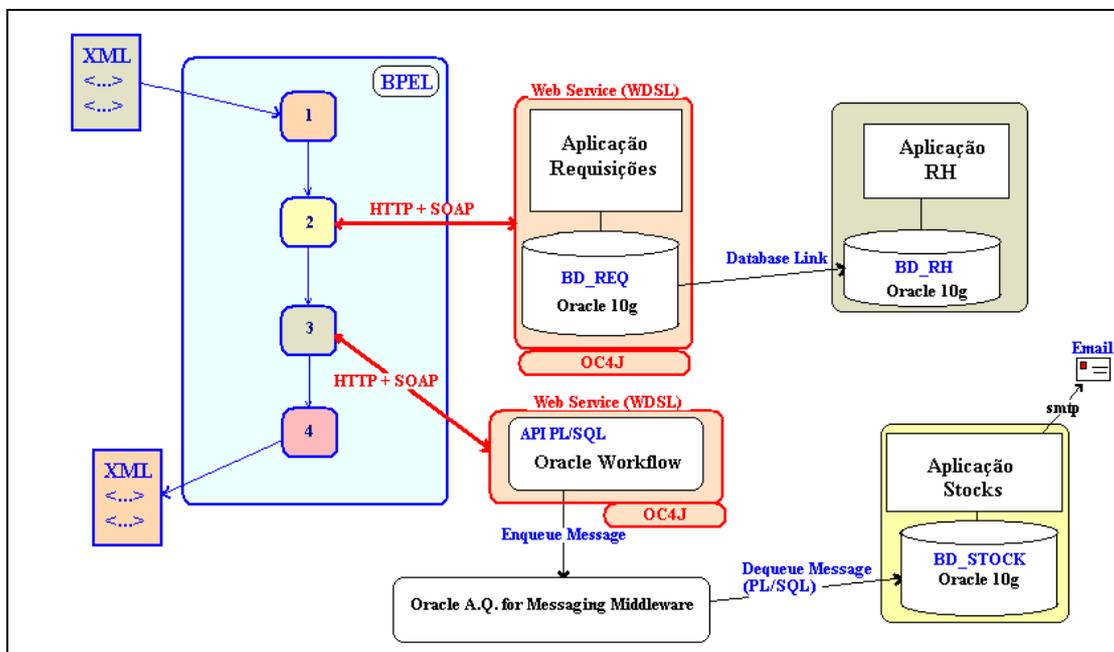


Figura 5.33: Integração com BPEL

Esta abordagem está suportada no BPEL Process Manager que permite criar as suas definições e executá-las de forma a poder integrar os diferentes *Web Services*. Neste caso, o écran da aplicação de Requisições deixa de ser utilizado, passando a ser utilizado um écran em HTML directamente no *Web Browser*. Esse novo écran permite introduzir a informação relativa à requisição e integra-se directamente com o sistema de execução do BPEL. Essa informação pode ser submetida em HTML ou directamente em XML.

5.3.1 Web Service para aplicação de Requisições

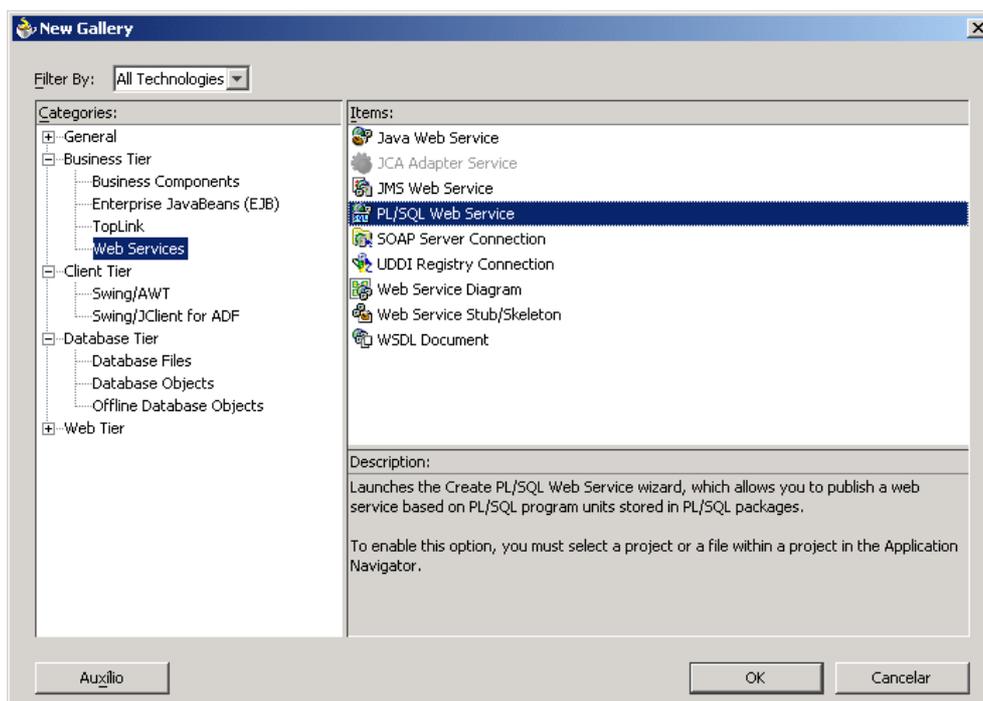


Figura 5.34: Criação do Web Service para a aplicação de Requisições

Para este exemplo é criado o primeiro *Web Service* “*inserir_requisicao*” que permite invocar um procedimento armazenado PL/SQL que recebe informação da requisição em XML e cria o registo na respectiva base de dados BD_REQ na tabela REQUISICOES (Figura 5.34). A identificação deste Web Service é feita através da norma WSDL e que define, entre outras coisas, o seu nome (“*inserir_requisicao*”), as mensagens de *input* e *output* e respectivas operações (“*inserirRequisicao0Request*” e “*inserirRequisicao0Response*”) e informação para a comunicação com o protocolo SOAP, nomeadamente o estilo de operação (*Remote Procedure Call* - RPC) e a sua localização na rede em “*http://vmartins-pt:8893/requisicoes_ws-Project1-context-root/Inserir_requisicao*” (Figura 5.35).

```

<?xml version = '1.0' encoding = 'UTF-8'?>
<!--Generated by the Oracle JDeveloper 10g Web Services WSDL Generator-->
<!--Date Created: Sun Jul 17 23:54:47 BST 2005-->
<definitions
  name="insertir_requisicao"
  targetNamespace="http://insertir_req_ws/insertir_requisicao.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://insertir_req_ws/insertir_requisicao.wsdl"
  xmlns:ns1="http://insertir_req_ws/insertir_requisicao.xsd">
  <types>
    <schema
      targetNamespace="http://insertir_req_ws/insertir_requisicao.xsd"
      xmlns="http://www.w3.org/2001/XMLSchema"
      xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" />
    </types>
    <message name="insertirRequisicaoRequest">
      <part name="numReq" type="xsd:string"/>
      <part name="idRequisitante" type="xsd:string"/>
      <part name="idItem" type="xsd:string"/>
      <part name="justificacao" type="xsd:string"/>
    </message>
    <message name="insertirRequisicaoResponse">
      <part name="return" type="xsd:string"/>
    </message>
    <portType name="insertir_requisicaoPortType">
      <operation name="insertirRequisicao">
        <input name="insertirRequisicaoRequest" message="tns:insertirRequisicaoRequest"/>
        <output name="insertirRequisicaoResponse" message="tns:insertirRequisicaoResponse"/>
      </operation>
    </portType>
    <binding name="insertir_requisicaoBinding" type="tns:insertir_requisicaoPortType">
      <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
      <operation name="insertirRequisicao">
        <soap:operation soapAction="" style="rpc"/>
        <input name="insertirRequisicaoRequest">
          <soap:body use="encoded" namespace="insertir_requisicao" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </input>
        <output name="insertirRequisicaoResponse">
          <soap:body use="encoded" namespace="insertir_requisicao" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
        </output>
      </operation>
    </binding>
    <service name="insertir_requisicao">
      <port name="insertir_requisicaoPort" binding="tns:insertir_requisicaoBinding">
        <soap:address location="http://vmartins-pt:8893/requisicoes_ws-Project1-context-root/insertir_requisicao"/>
      </port>
    </service>
  </definitions>

```

Figura 5.35: Especificação WSDL do Web Service para a aplicação de Requisições

Com a ferramenta de programação Oracle JDeveloper 10g é feito o *deployment* deste *Web Service* numa componente do tipo Servidor Aplicacional, denominado *Oracle Container for Java (OC4J)* ficando desta forma disponível e acessível (Figura 5.36).

Figura 5.36: Deployment do Web Service no OC4J

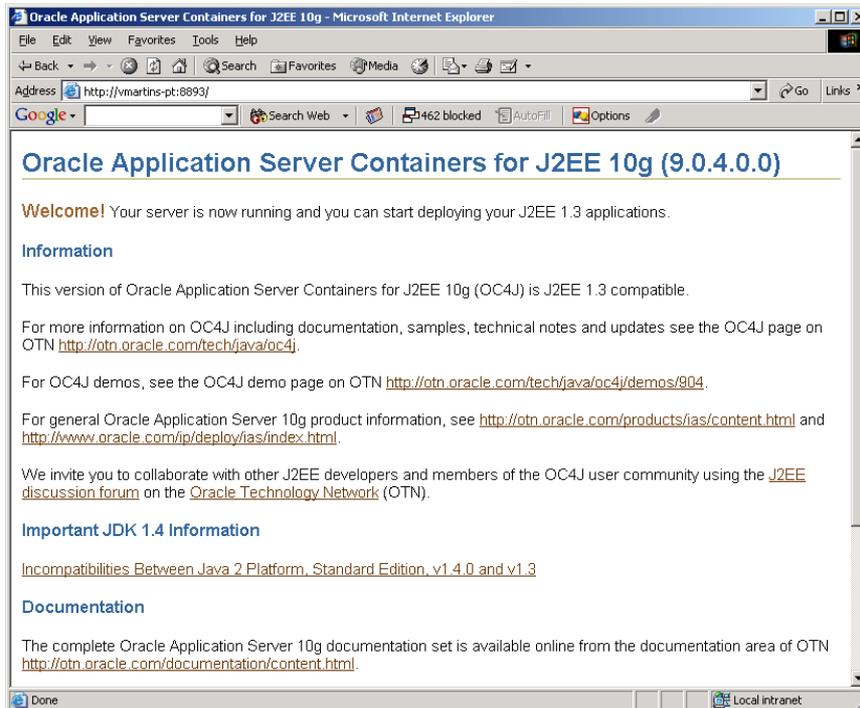


Figura 5.37: OC4J para disponibilizar os *Web Services* baseado na plataforma J2EE

O OC4J é uma componente do servidor aplicativo da Oracle e permite um acesso via HTTP no porto 8893 a um conjunto de serviços e aplicações como são por exemplo os *Web Services* (Figura 5.37). Através do protocolo HTTP pode-se aceder ao Web Service através do URL “http://vmartins-pt:8893/requisicoes_ws-Project1-context-root/Inserir_requisicao” (Figura 5.38).



Figura 5.38: Web Service para a criação das requisições com operação “inserirRequisicao”

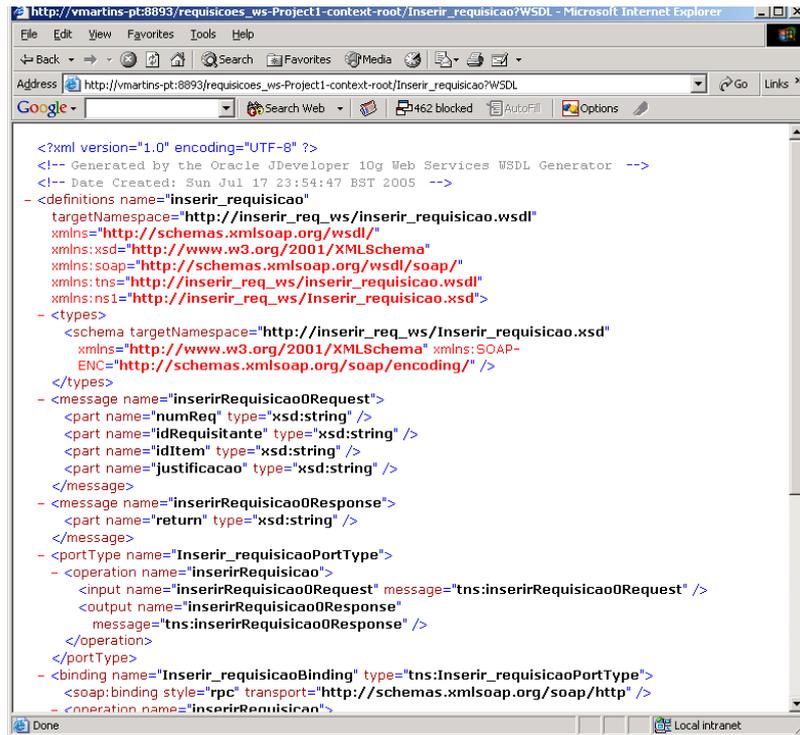


Figura 5.39: WSDL do Web Service

A descrição do Web Service através da norma WSDL está disponível na rede como OC4J e através do endereço: *http://vmartins-pt:8893/requisicoes_ws-Project1-context-root/Inserir_requisicao?WSDL* (Figura 5.39). Pela mesma via é também possível invocar directamente o *Web Service* através de um écran em HTML que permite inserir os parâmetros necessários (Figura 5.40).

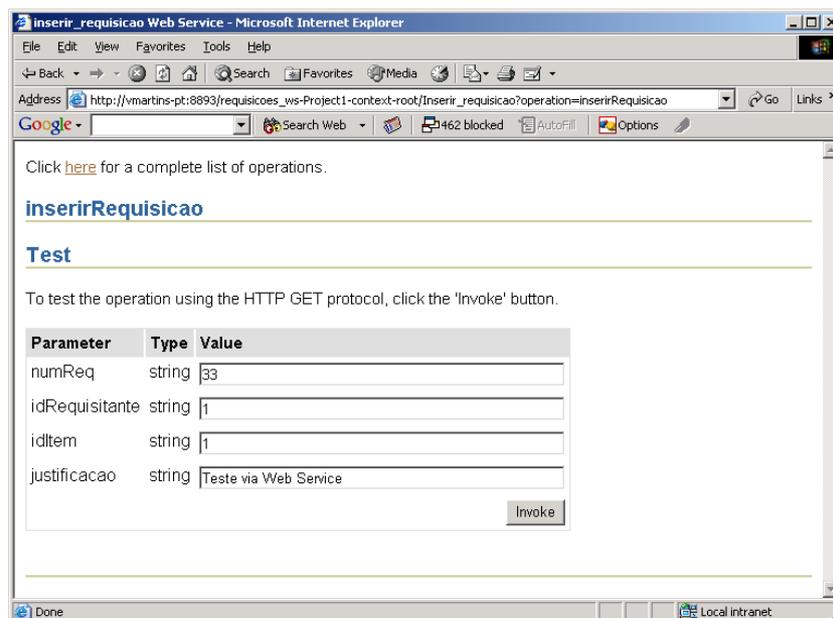


Figura 5.40: Écran de teste para a operação de “inserirRequisicao” da requisição nº33

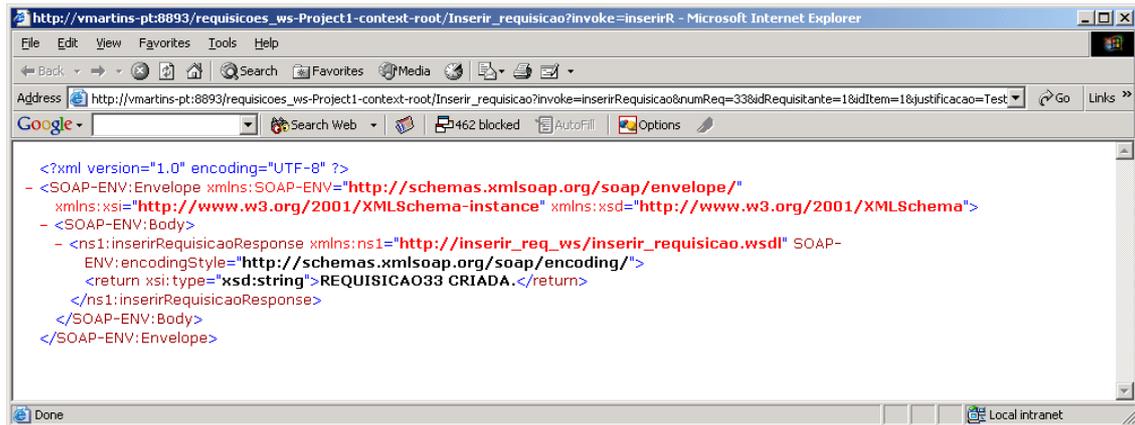


Figura 5.41: Resposta do Web Service após a sua invocação

A invocação directa do *Web Service* “inserir_requisicao” retorna uma mensagem em XML com o resultado da operação e correspondente à definição WSDL (Figura 5.41). Para verificar o real sucesso da operação é feita uma *query* à base de dados na sobre a tabela REQUISICÕES (Figura 5.42): “*select num_req,id_funcionario,id_item,nome_item,justificacao from requisicoes where num_req='33';*”

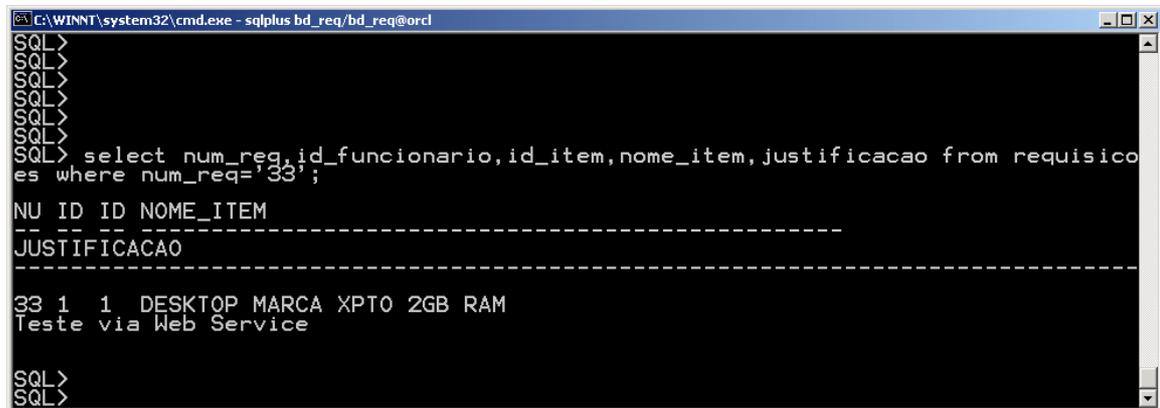
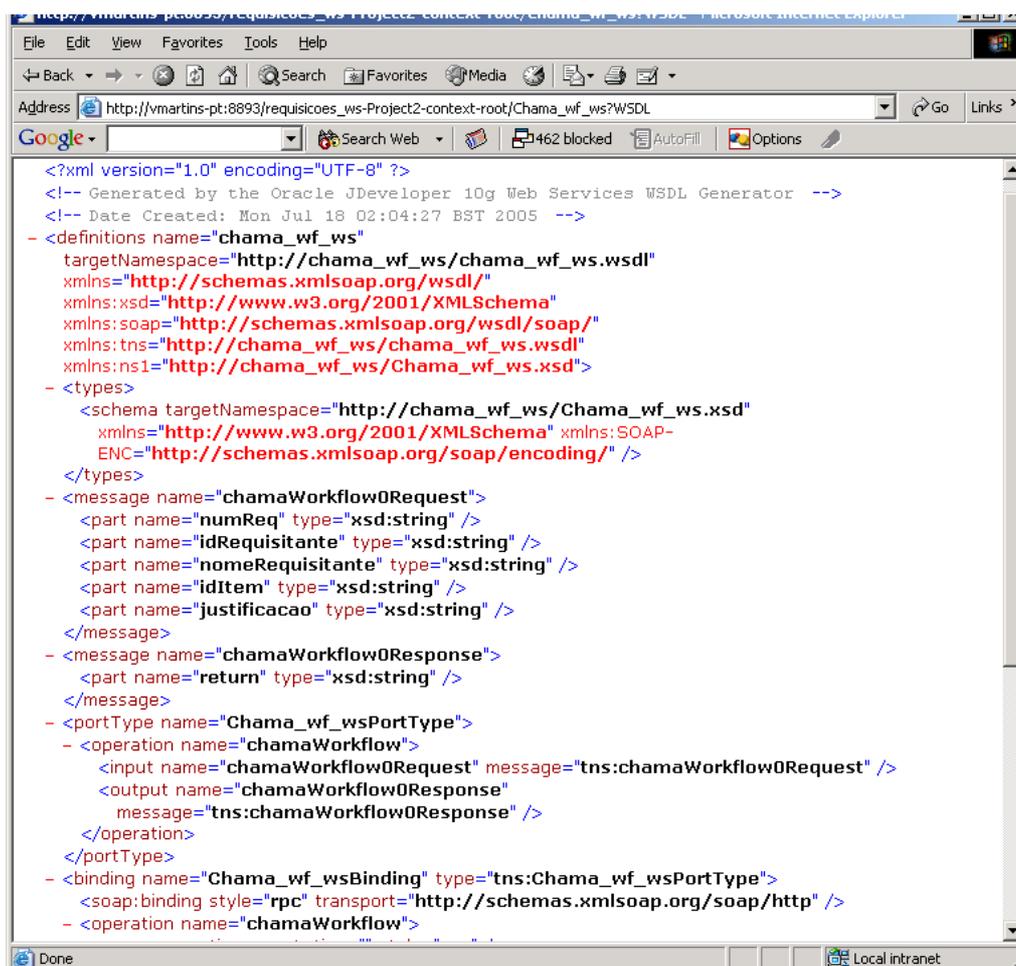


Figura 5.42: Query sobre tabela Requisições com registo da requisição nº33 submetido por *Web Service*

5.3.2 Web Service para solução de *workflow* para aprovação de requisições

Com a mesma lógica e os mesmos procedimentos para a criação do Web Service anterior, é aqui criado um segundo *Web Service* que permite invocar o sistema de *workflow* que permite a integração entre a aplicação de Requisições e a aplicação de Stocks. Este novo Web Service denomina-se “chama_wf_ws” (Figura 5.43).



```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by the Oracle JDeveloper 10g Web Services WSDL Generator -->
<!-- Date Created: Mon Jul 18 02:04:27 BST 2005 -->
- <definitions name="chama_wf_ws"
  targetNamespace="http://chama_wf_ws/chama_wf_ws.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://chama_wf_ws/chama_wf_ws.wsdl"
  xmlns:ns1="http://chama_wf_ws/Chama_wf_ws.xsd">
- <types>
  <schema targetNamespace="http://chama_wf_ws/Chama_wf_ws.xsd"
    xmlns="http://www.w3.org/2001/XMLSchema" xmlns:SOAP-
    ENC="http://schemas.xmlsoap.org/soap/encoding/" />
</types>
- <message name="chamaWorkflow0Request">
  <part name="numReq" type="xsd:string" />
  <part name="idRequisitante" type="xsd:string" />
  <part name="nomeRequisitante" type="xsd:string" />
  <part name="idItem" type="xsd:string" />
  <part name="justificacao" type="xsd:string" />
</message>
- <message name="chamaWorkflow0Response">
  <part name="return" type="xsd:string" />
</message>
- <portType name="Chama_wf_wsPortType">
  - <operation name="chamaWorkflow">
    <input name="chamaWorkflow0Request" message="tns:chamaWorkflow0Request" />
    <output name="chamaWorkflow0Response"
      message="tns:chamaWorkflow0Response" />
  </operation>
</portType>
- <binding name="Chama_wf_wsBinding" type="tns:Chama_wf_wsPortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  - <operation name="chamaWorkflow">
```

Figura 5.43: WSDL do Web Service do sistema de *workflow*

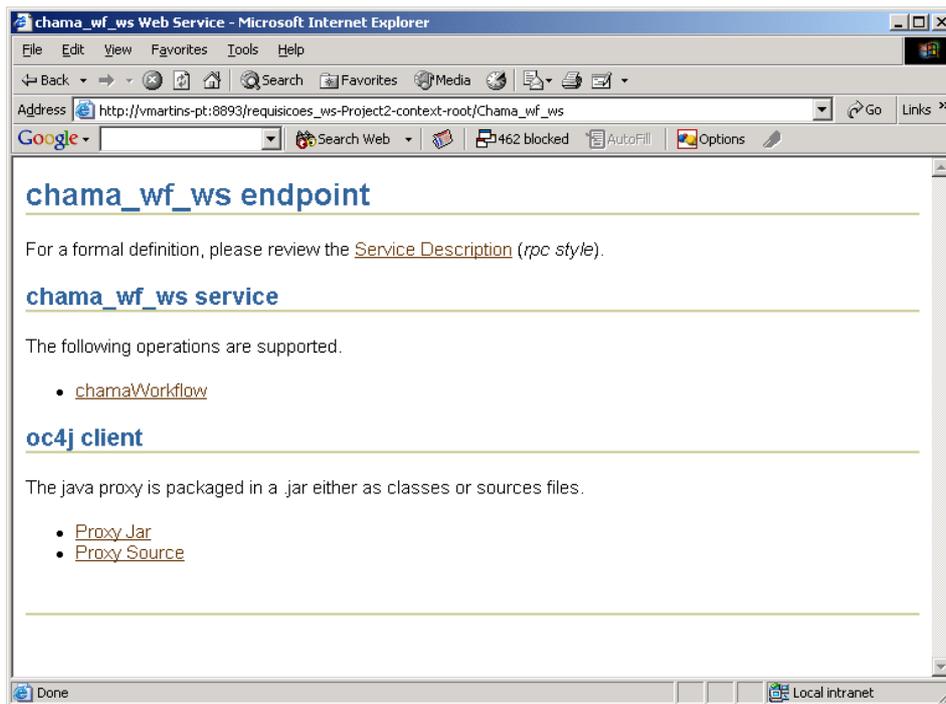


Figura 5.44: Web Service para invocação do sistema de *workflow* de aprovação de requisições

Na mesma ordem de ideias, este segundo *Web Service* pode ser acessado directamente via http no endereço “http://vmartins-pt:8893/requisicoes_ws-Project2-context-root/Chama_wf_ws” (Figura 5.44) e pode também ser invocado directamente pela mesma via (Figura 5.45).

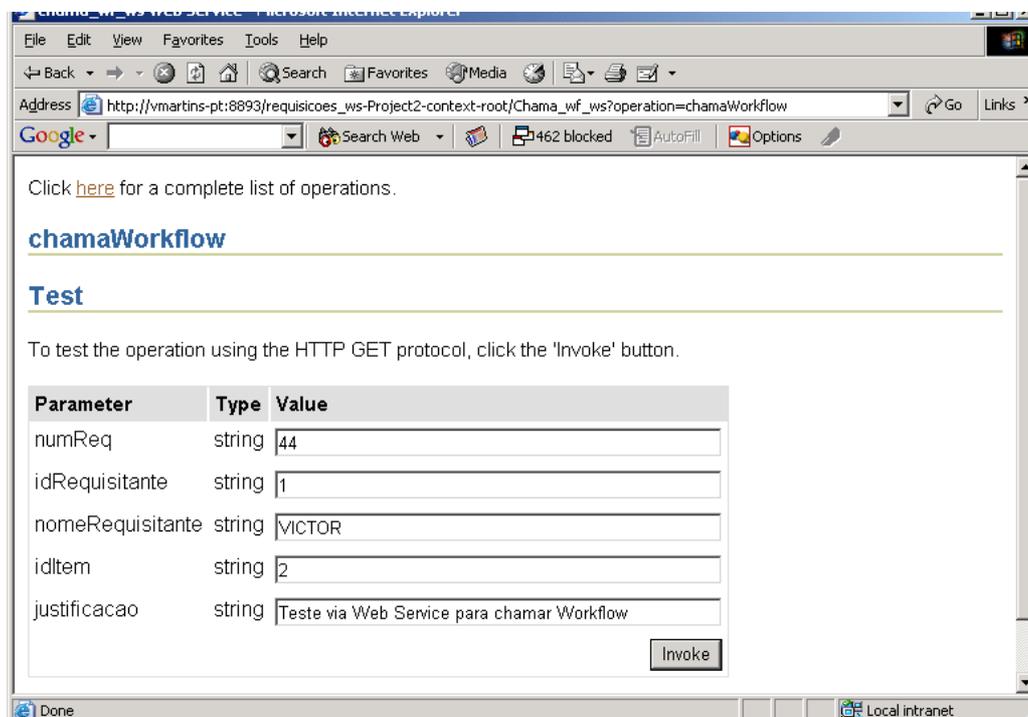


Figura 5.45: Écrã de teste para a operação de “chamaWorkflow” da requisicao nº44

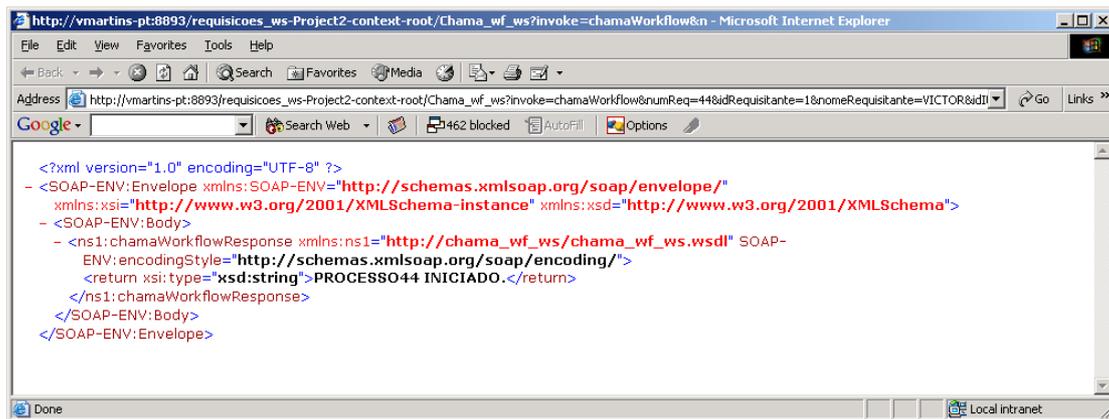


Figura 5.46: Resposta do Web Service após a sua invocação

A invocação directa deste segundo *Web Service* retorna informação em XML indicando que o processo de workflow foi iniciado com sucesso. Esta é a mesma informação que será passada ao sistema que executa o BPEL (Figura 5.46).

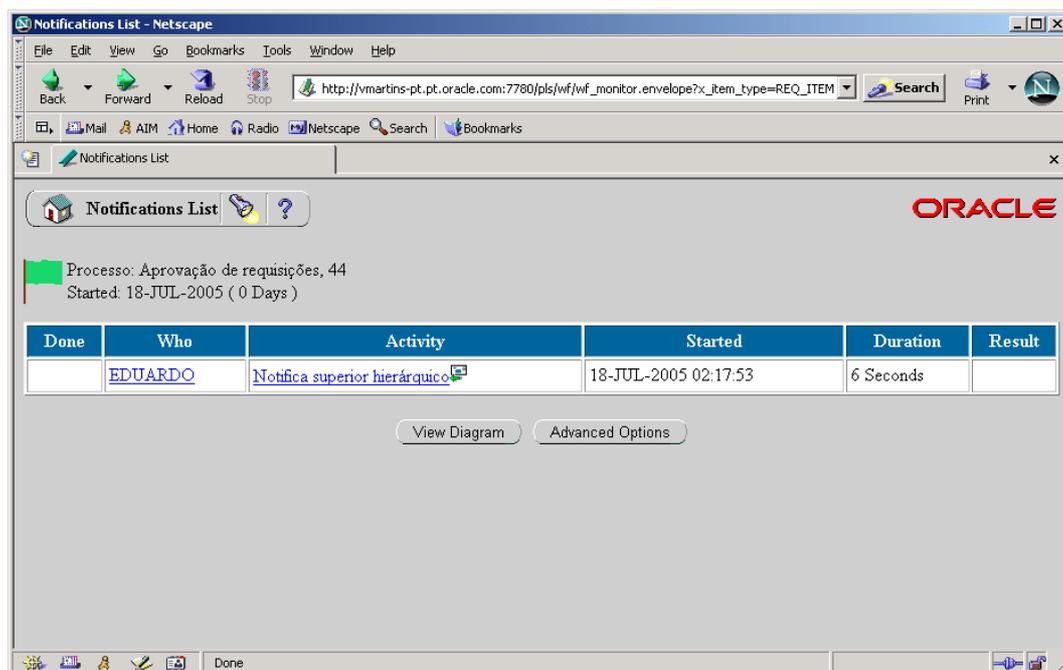


Figura 5.47: Requisição nº44 submetida por *Web Service* sujeita a aprovação no sistema de *workflow*

A invocação directa deste *Web Service* criando a requisição nº 44 despoletou o *software* de *workflow* que segue exactamente os mesmo passos e regras explicadas nos pontos anteriores na primeira abordagem de integração deste capítulo (Figura 5.47).

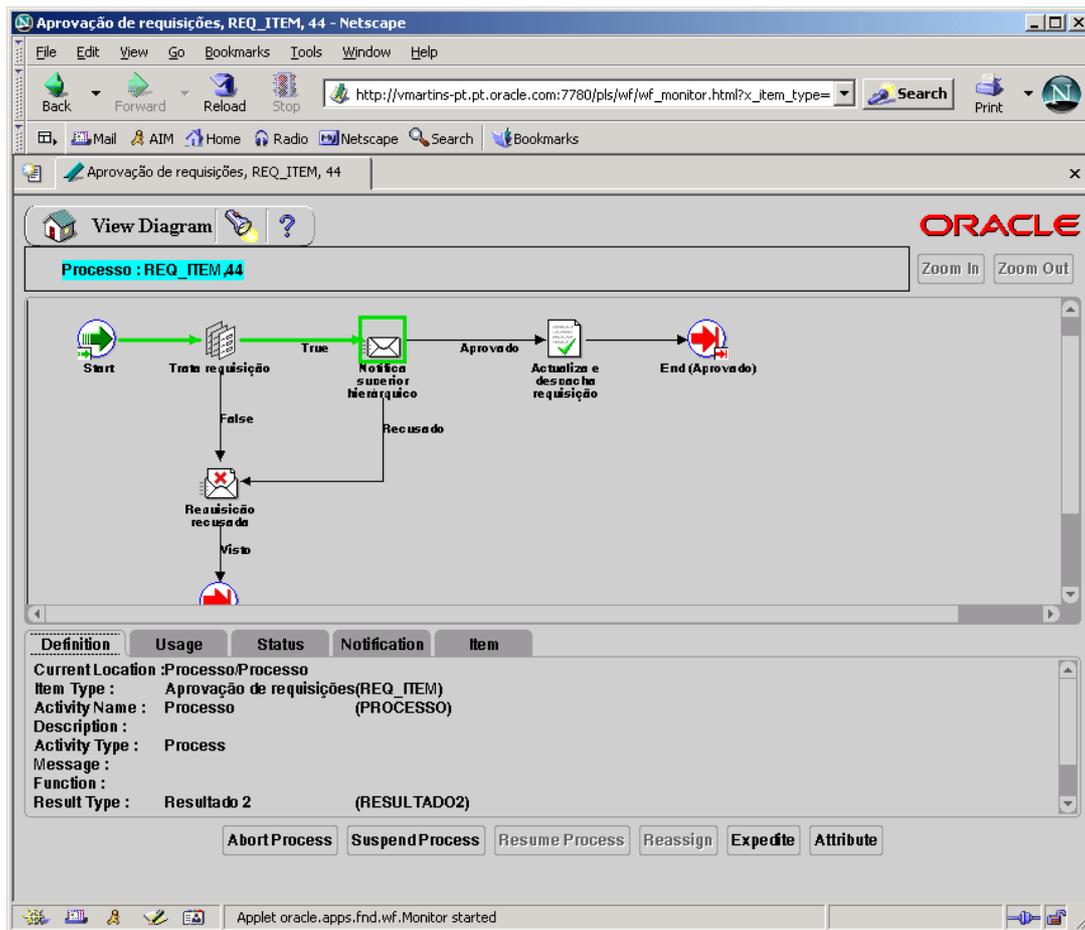


Figura 5.48: Requisição nº44 submetida por Web Service

O processo nº44 iniciado através do Web Service “chama_wf_ws” notificou o superior hierárquico da pessoa que criou a requisição nº 44 (Figura 5.48).

5.3.3 BPEL para processo de requisições

A norma BPEL permite definir em XML a integração de sistemas através da invocação de Web Services numa lógica processual. Para este exemplo vai ser criada uma definição BPEL denominada “REQUISICOES_BPEL” (Figura 5.49). Este processo será do tipo síncrono significando que logo após a invocação do processo é esperada uma resposta. Toda a informação veiculada estará estruturada em XML cujo *namespace* será “http://xmlns.oracle.com/REQUISICOES_BPEL”. Esta definição é implementada em Oracle BPEL Process Manager 10g.

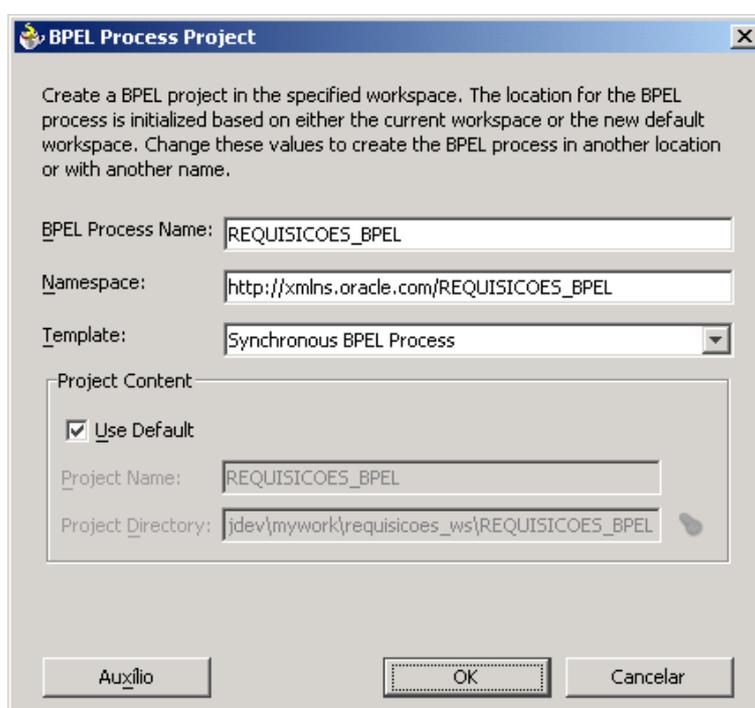


Figura 5.49: Criação do projecto BPEL síncrono “REQUISICOES_BPEL” no BPEL Designer

Esta ferramenta permite conceber graficamente as definições BPEL criando automaticamente a respectiva estrutura em XML, permitindo também fazer o *deployment* da respectiva definição no Oracle OC4J que suporta o Oracle BPEL Process Manager (Figura 5.50).

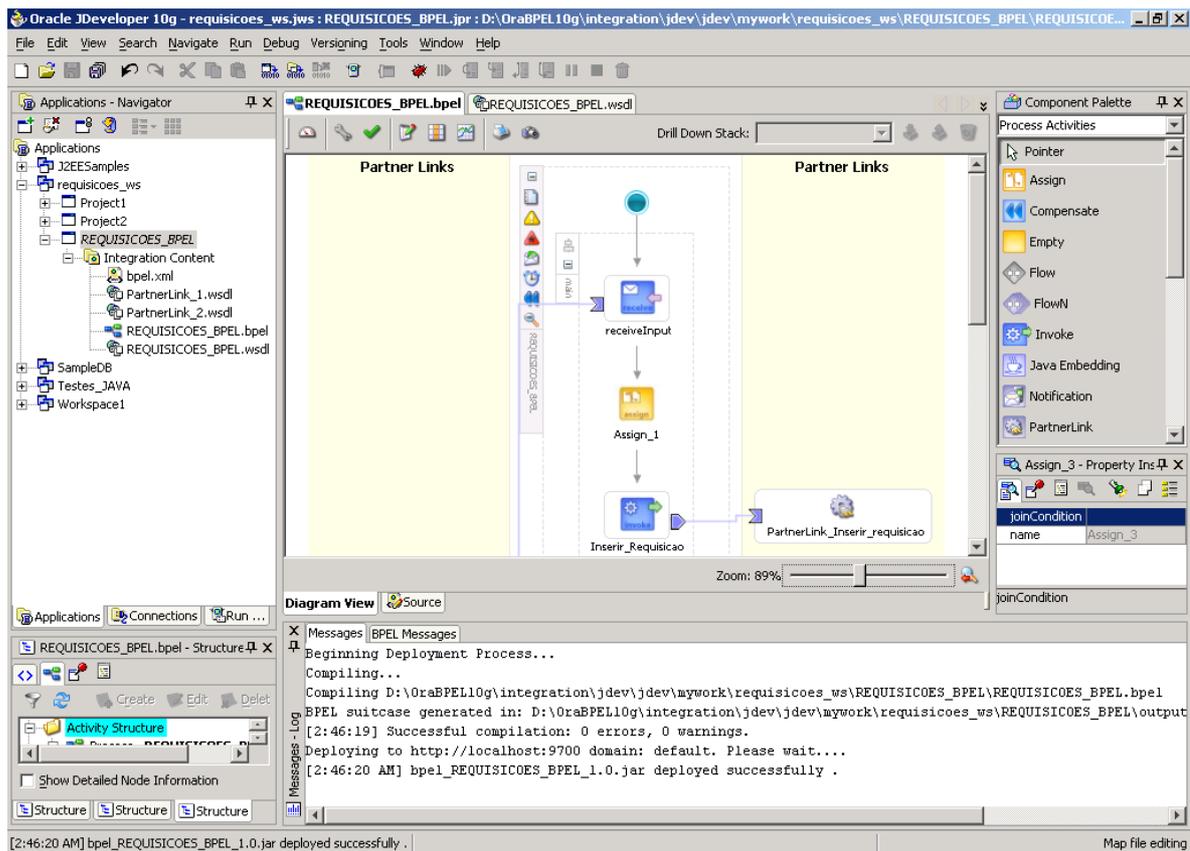


Figura 5.50: BPEL Designer – Concepção gráfica e *deployment* do processo BPEL

A definição gráfica deste processo BPEL está estruturada por 7 etapas processuais e 3 *partnerlinks* que correspondem aos 3 pontos de interacção com este processo BPEL (Figura 5.51). A primeira etapa “receiveInput” trata de receber informação em XML que representa a requisição que se quer criar e submeter para aprovação. A segunda etapa “Assign_1” trata de passar a informação necessária para variáveis internas de forma a passar somente os parâmetros necessários aquando da invocação dos *Web Services*. A terceira etapa “Inserir_Requisicao” trata de invocar de forma síncrona o *Web Service* “inserir_requisicao” armazenando o seu resultado. A quarta etapa “Assign_2” trata de utilizar o resultado encontrado na etapa anterior de forma a poder invocar o próximo *Web Service* com os parâmetros correctos.

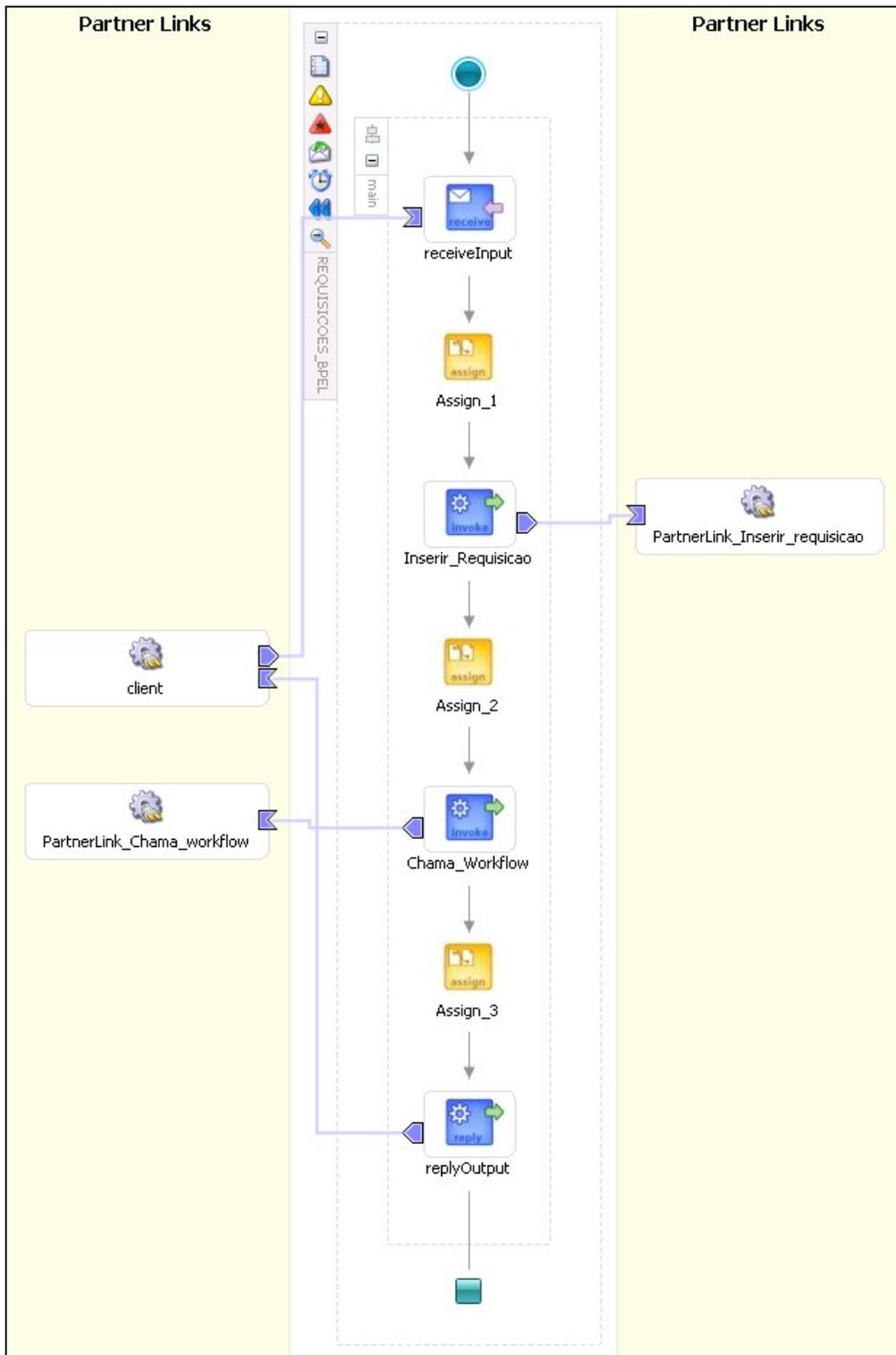


Figura 5.51: Diagrama BPEL do processo "REQUISICOES_BPEL"

A quinta etapa "Chama_Workflow" invoca sincronamente o *Web Service* "chama_wf_ws" de forma a iniciar o processo de *workflow* para aprovação do pedido de requisição e armazena o resultado dessa invocação. A sexta etapa "Assign_3" trata a informação recebida na etapa anterior de forma a poder preparar o resultado final

deste processo BPEL. A sétima e última etapa “replyOutput” retorna o resultado final para o sistema que invocou este BPEL.

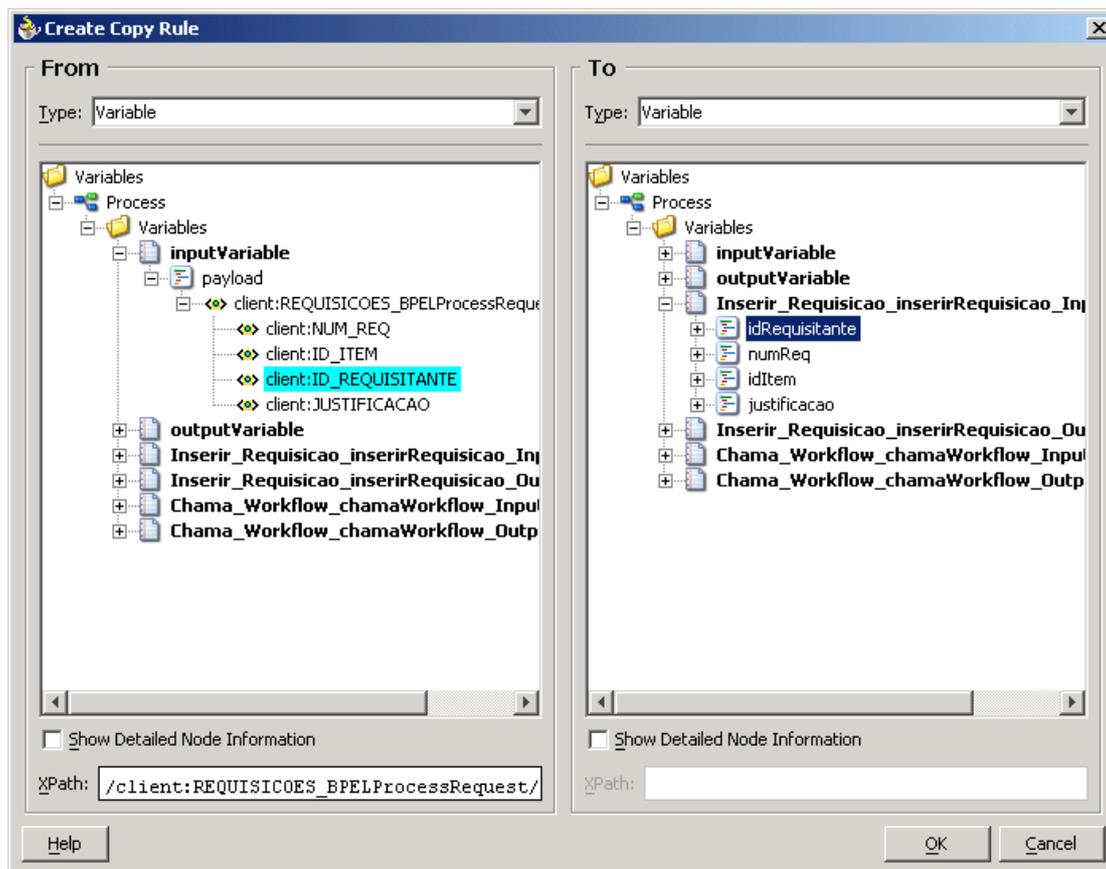


Figura 5.52: Associação e passagem de valores entre parâmetros do BPEL e dos Web Services

A passagem de valores é feita nas etapas do tipo “Assign” que permitem tratar a informação em xml que é veiculada ao longo do processo BPEL. No caso da etapa “Assign_1” são passados os 4 valores existentes na estrutura em XML que está na “inputVariable”. Essa estrutura é normalmente denominada de *payload* e tem toda a informação que é enviada ao sistema que executa as definições BPEL.

Dado que este exemplo em BPEL só recebe o número de identificação da pessoa que cria a requisição, o Web Service “inserir_requisicao” retorna o valor correspondente ao nome completo dessa pessoa (Figura 5.52) e que está na tabela FUNCIONARIOS da base de dados da aplicação de RH.

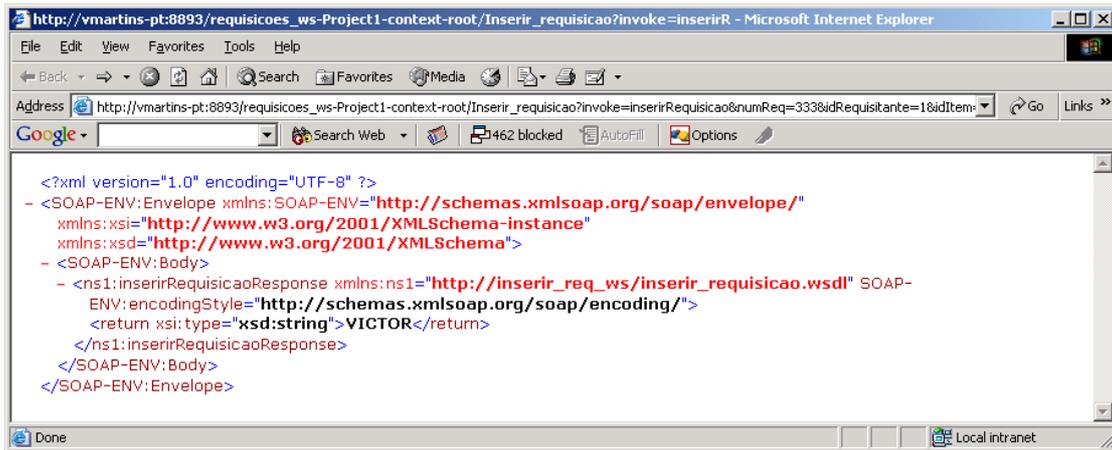


Figura 5.53: Alteração da resposta do Web Service da aplicação de Requisições que retorna nome de requisitante

O valor retornado pelo Web Service invocado é armazenado na variável “return” (Figura 5.53) e automaticamente mapeado para o parâmetro “nome Requisitante” do segundo Web Service “chama_wf_ws” (Figura 5.54).

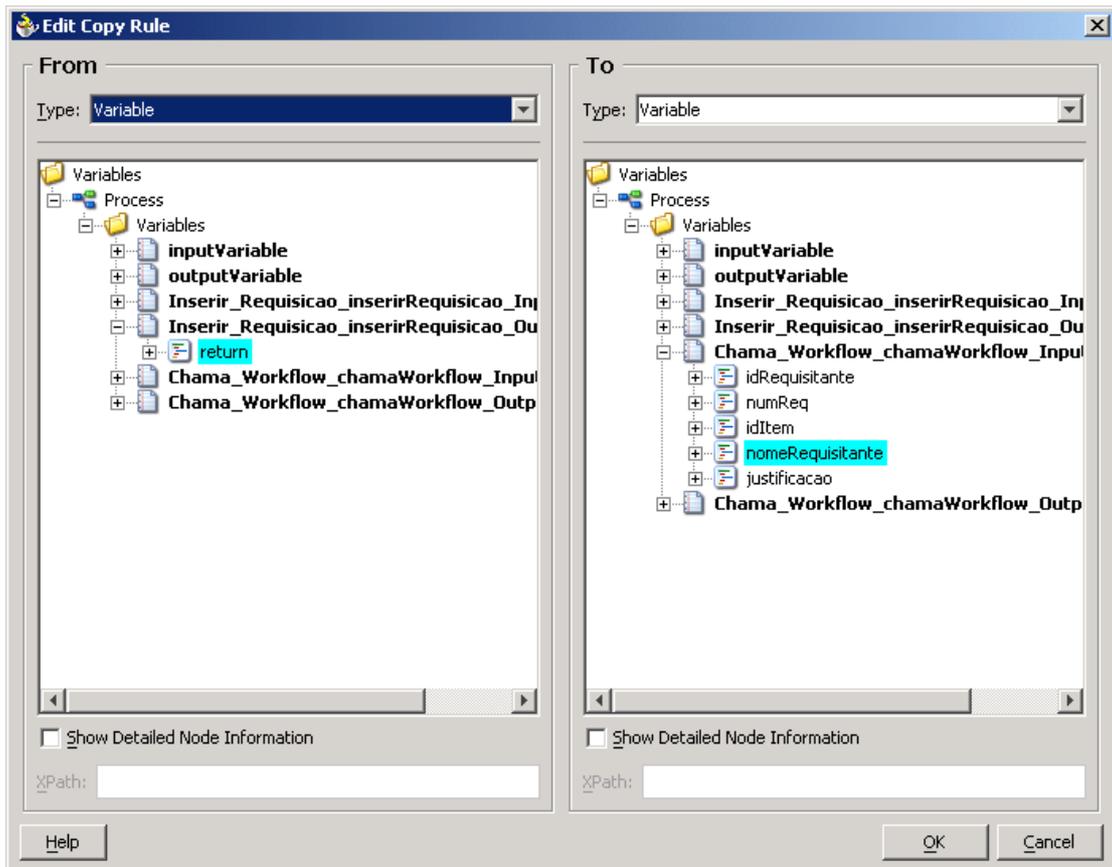


Figura 5.54: Passagem do valor retornado pelo Web Service alterado no ponto anterior

Após a finalização da definição em BPEL da versão 1.0 do processo “REQUISICOES_BPEL” faz-se o deployment dele para o sistema de execução do BPEL “BPEL Server” baseado no OC4J 10g (Figura 5.55).

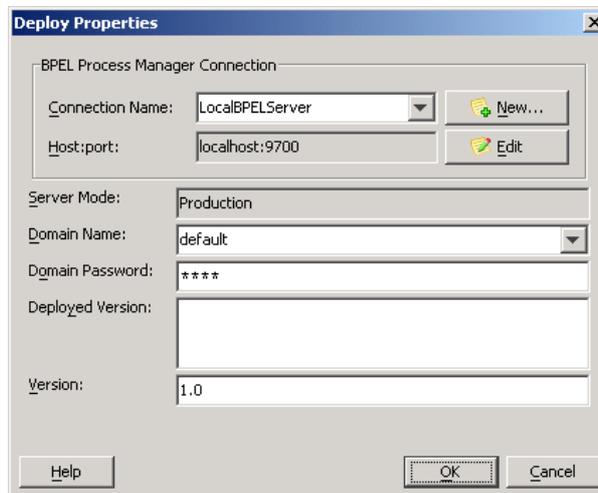


Figura 5.55: *Deployment* do processo BPEL para o BPEL Process Manager

O sistema “BPEL Server” disponibiliza um ambiente de monitorização e testes, denominado de BPEL Console, na a execução dos processos BPEL (Figura 5.56).

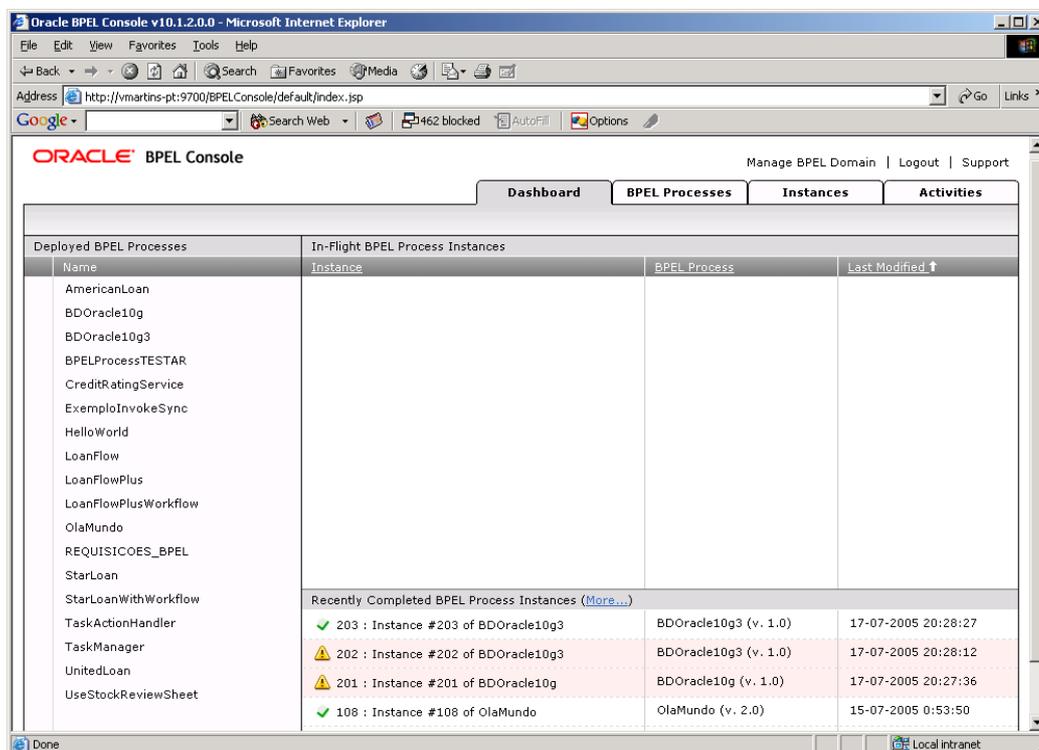


Figura 5.56: BPEL Console – Administração e teste dos processos de BPEL via Web.

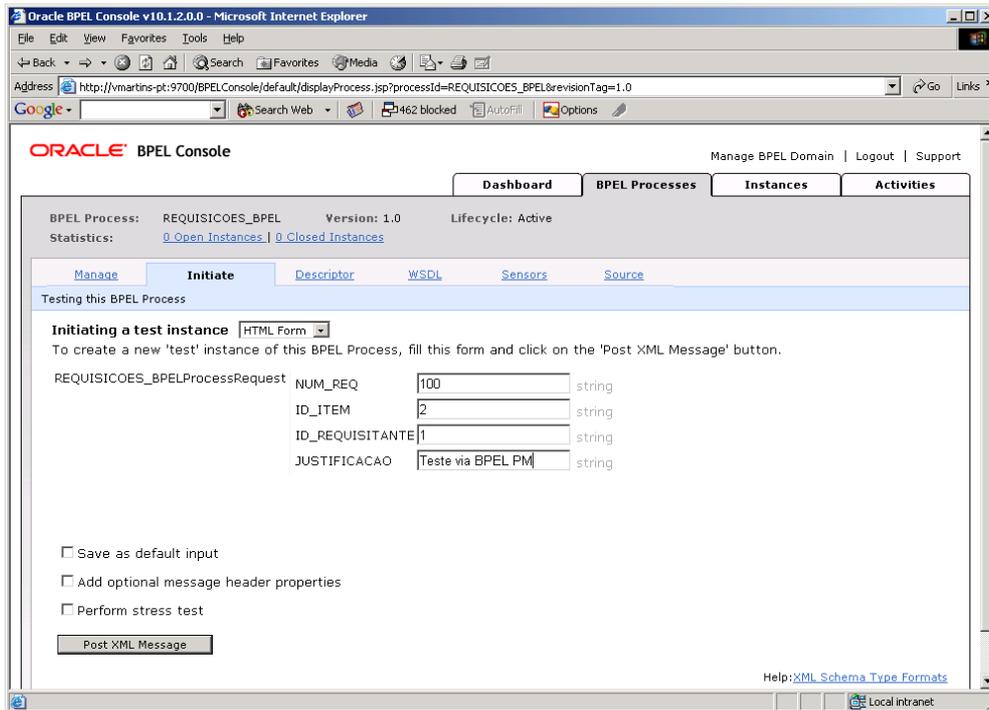


Figura 5.57: Submissão de um formulário em HTML para o processo “REQUISICOES_BPEL”

Para dar início à execução desta segunda abordagem com BPEL é utilizado um écran em HTML que se substitui ao écran da aplicação de Requisições implementado em Oracle Forms 10g. Aqui são introduzidos os valores para a requisição nº100 do bem nº2 pela pessoa nº2 (Figura 5.57). O processo é executado com sucesso retornando a informação relativa ao segundo *Web Service* (Figura 5.58).

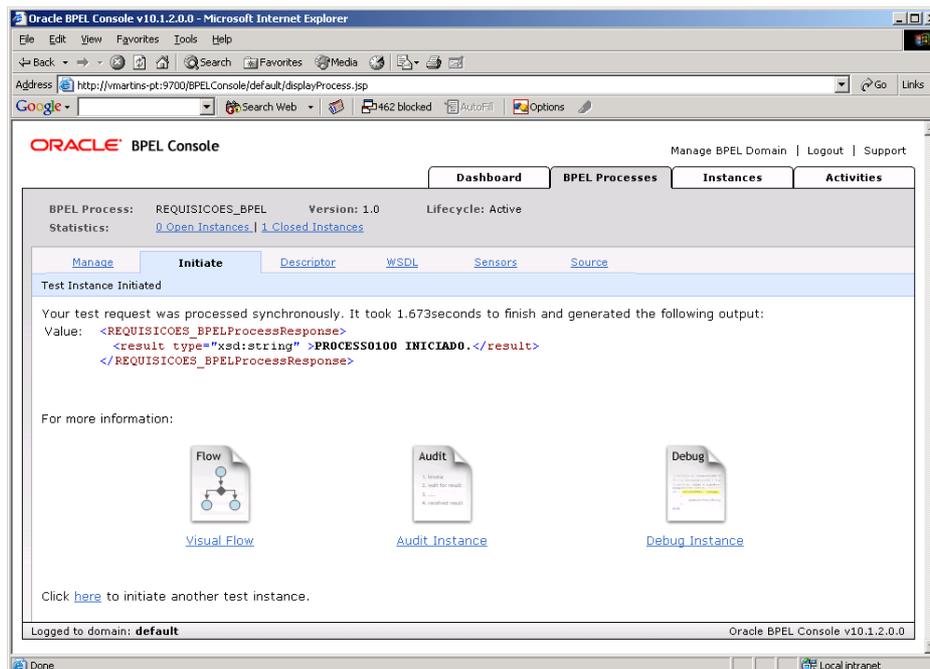


Figura 5.58: Processo BPEL síncrono executado com sucesso para a requisição nº 100

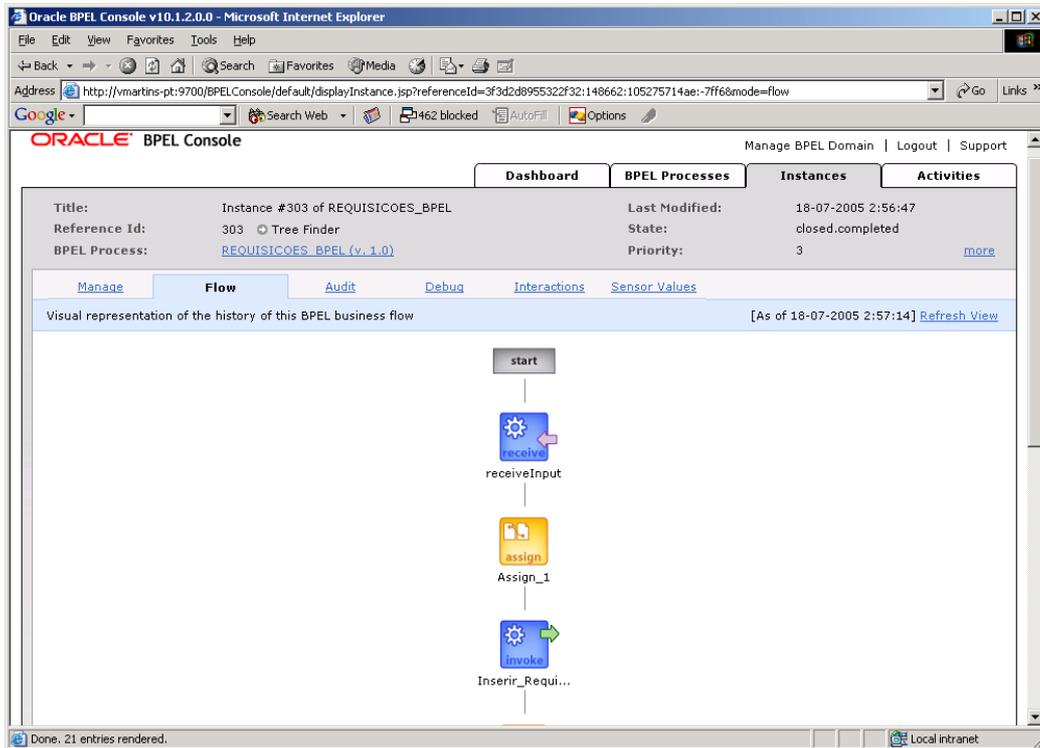


Figura 5.59: Monitorização *on-line* via *Web* do processo BPEL para a requisição nº 100

O ambiente de monitorização do Oracle BPEL Process Manager permite visualizar o processo nº100 na sua forma gráfica e analisar toda a informação veiculada ao longo das suas etapas (Figura 5.59). A informação submetida através do écran em HTML é estruturada internamente em XML (Figura 5.60).

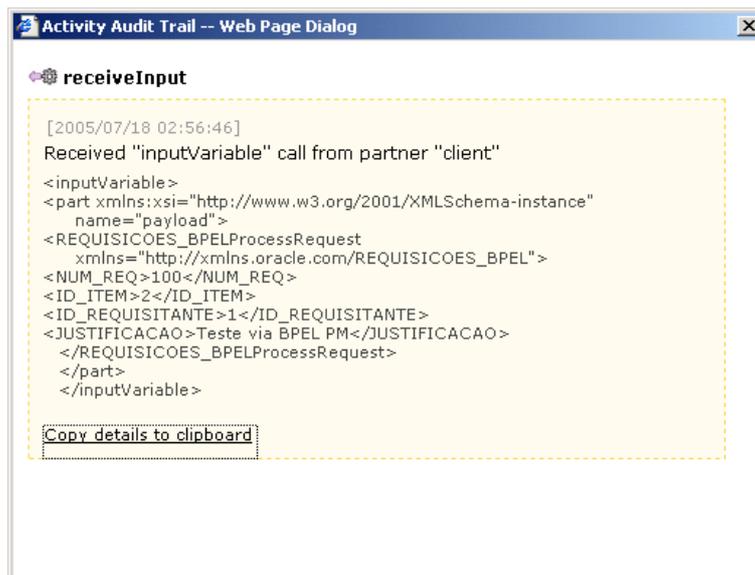


Figura 5.60: XML na etapa “receiveInput” do BPEL com a requisição nº100



Figura 5.61: XML na invocação do Web Service “Inserir_Requisicao” para a requisição nº100

A informação passada ao primeiro *Web Service* também está estruturada em XML (Figura 5.61) assim com resultado obtido aquando da sua invocação na variável “return”. Ao invocar este Web Service o BPEL vai criar um novo registo na base de dados correspondentes à requisição nº100 (Figura 5.62).

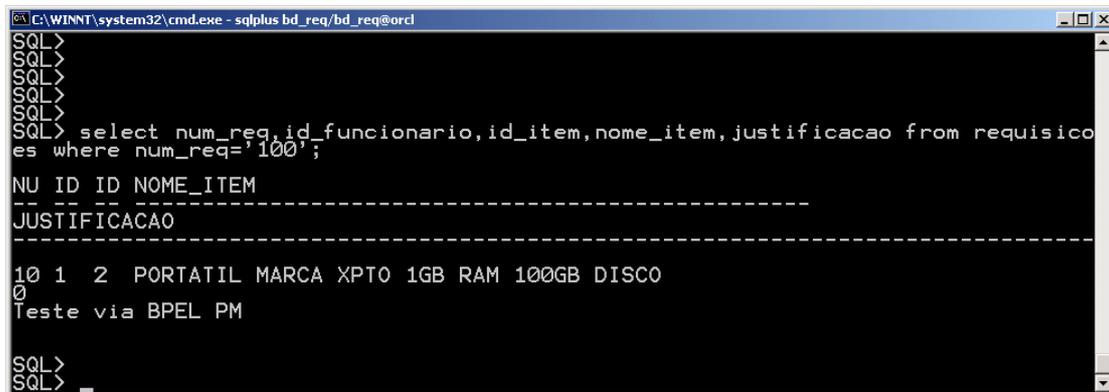


Figura 5.62: Verificação da criação da requisição nº 100 na base de dados via BPEL + *Web Service*

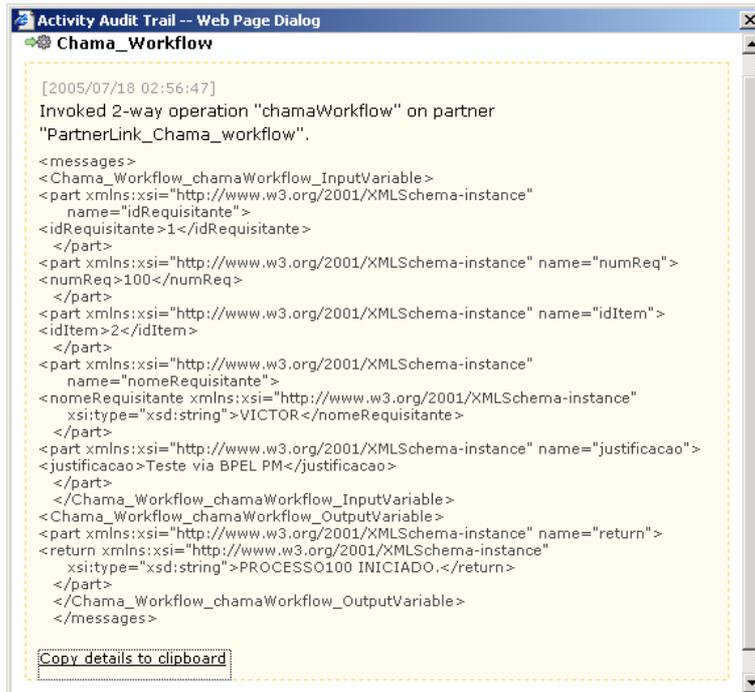


Figura 5.63: XML na etapa “Chama_Workflow” do BPEL com a requisição nº100

A informação passada ao segundo *Web Service* está igualmente estruturada em XML assim como o resultado armazenado na variável “return” (Figura 5.63). O processo de workflow foi entretanto lançado (Figura 5.64) e o resto da aplicação segue exactamente as mesmas etapas da primeira abordagem implementada para este exemplo (Figura 5.65).

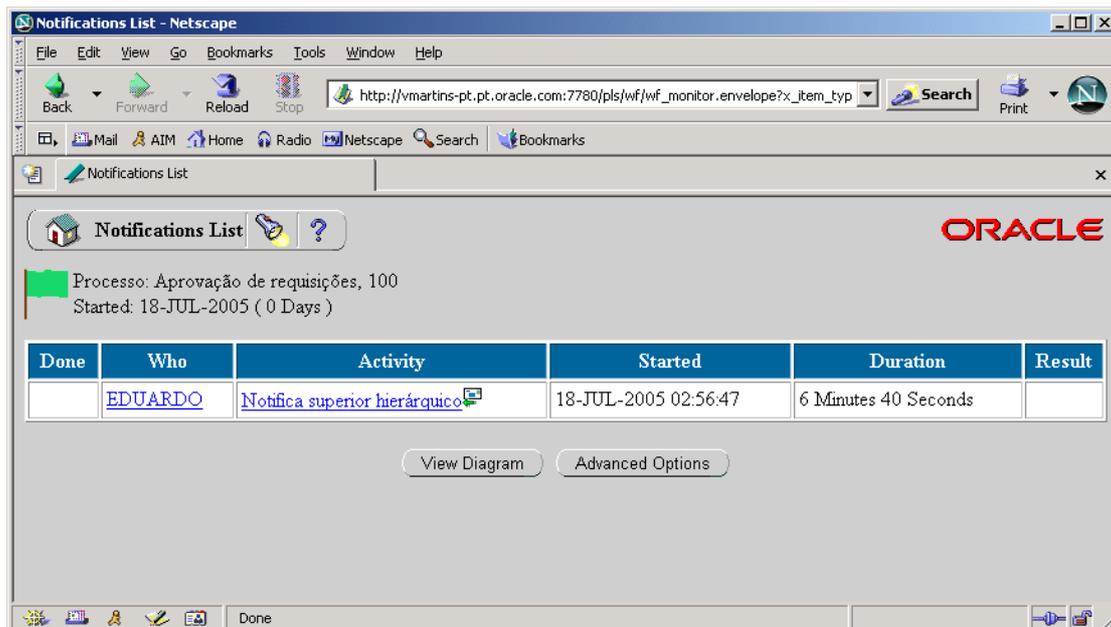


Figura 5.64: Requisição nº100 submetida ao sistema de *workflow* via BPEL + *Web Service*

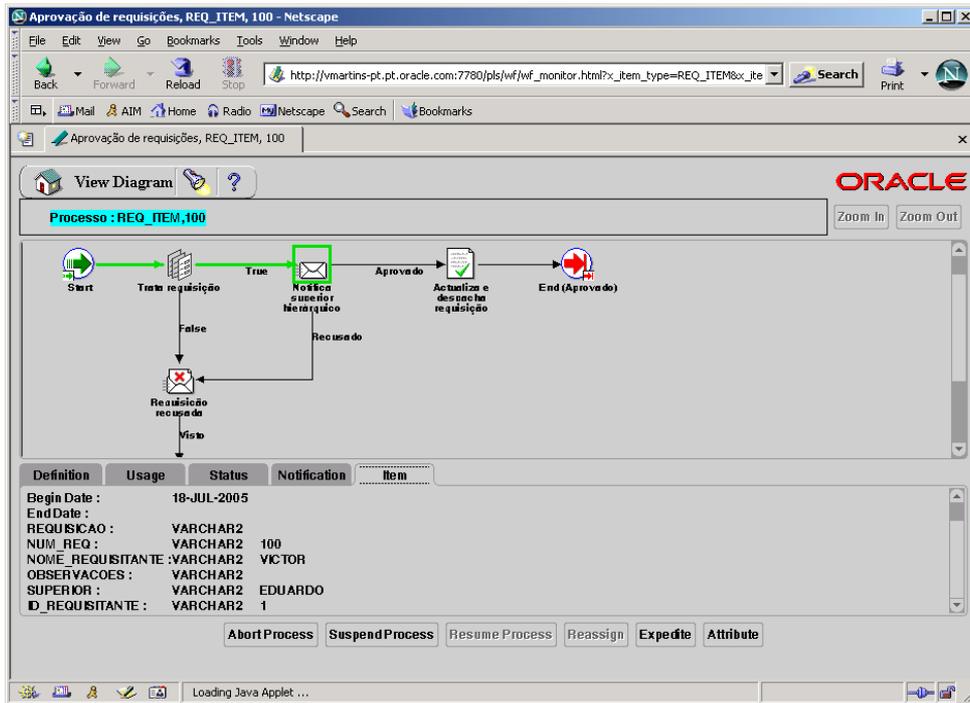


Figura 5.75: Requisição nº100 submetida ao sistema de *workflow* via BPEL + *Web Service*

No final, e tal como na primeira abordagem a aplicação de Stocks envia um email para a pessoa que efectuou a requisição, neste caso via BPEL Process Manager (Figura 5.76).

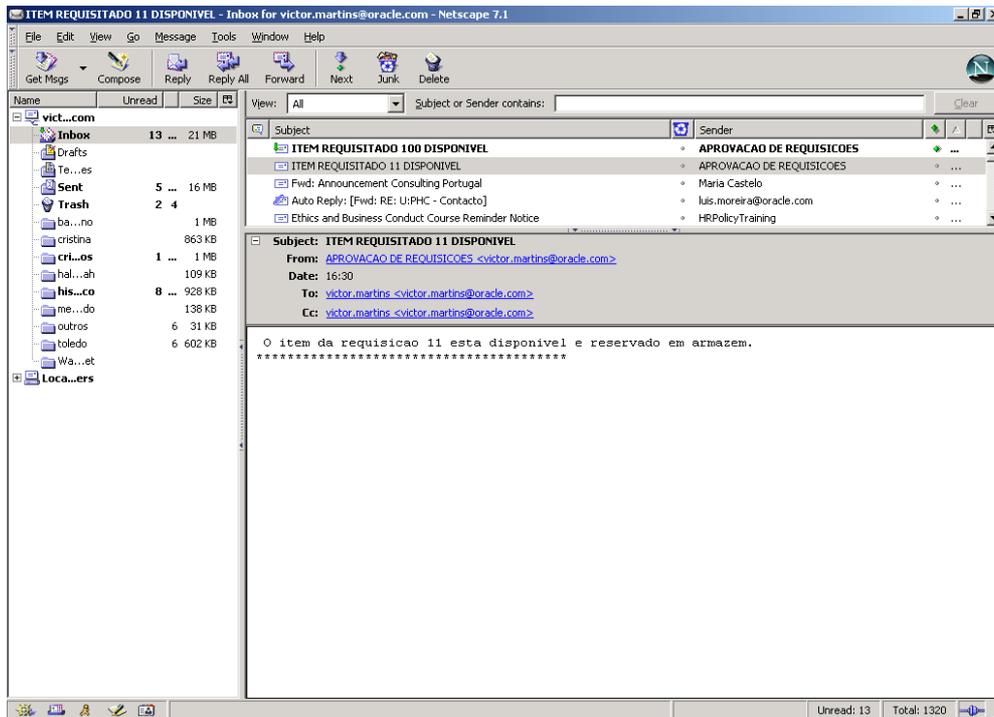


Figura 5.76: Recepção de email para a requisição nº 100

É também possível iniciar um processo BPEL directamente com informação já estruturada em XML (Figura 5.77) e enviá-la desta forma para o sistema que executa o processo BPEL (Figura 5.78).

```
<REQUISICOES_BPELProcessRequest xmlns="http://xmlns.oracle.com/REQUISICOES_BPEL">  
<NUM_REQ>103</NUM_REQ>  
<ID_ITEM>2</ID_ITEM>  
<ID_REQUISITANTE>1</ID_REQUISITANTE>  
<JUSTIFICACAO>Teste via BPEL PM</JUSTIFICACAO>  
</REQUISICOES_BPELProcessRequest>
```

Figura 5.77: Requisição nº 103 em XML

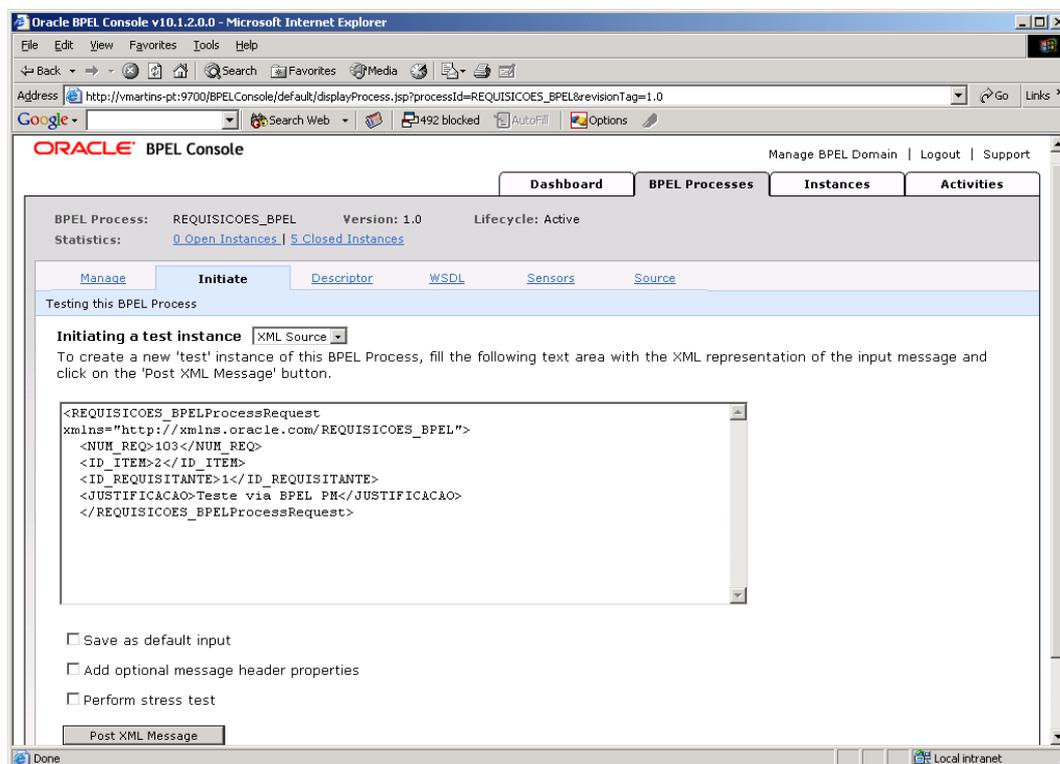


Figura 5.78: Submissão directa de XML para o processo “REQUISICOES_BPEL”

Apesar de se ter enviado directamente XML, as restantes etapas desta abordagem para a integração destas aplicações seguem a mesma lógica sem qualquer alteração e com o mesmo tipo de resultado (Figura 5.79).

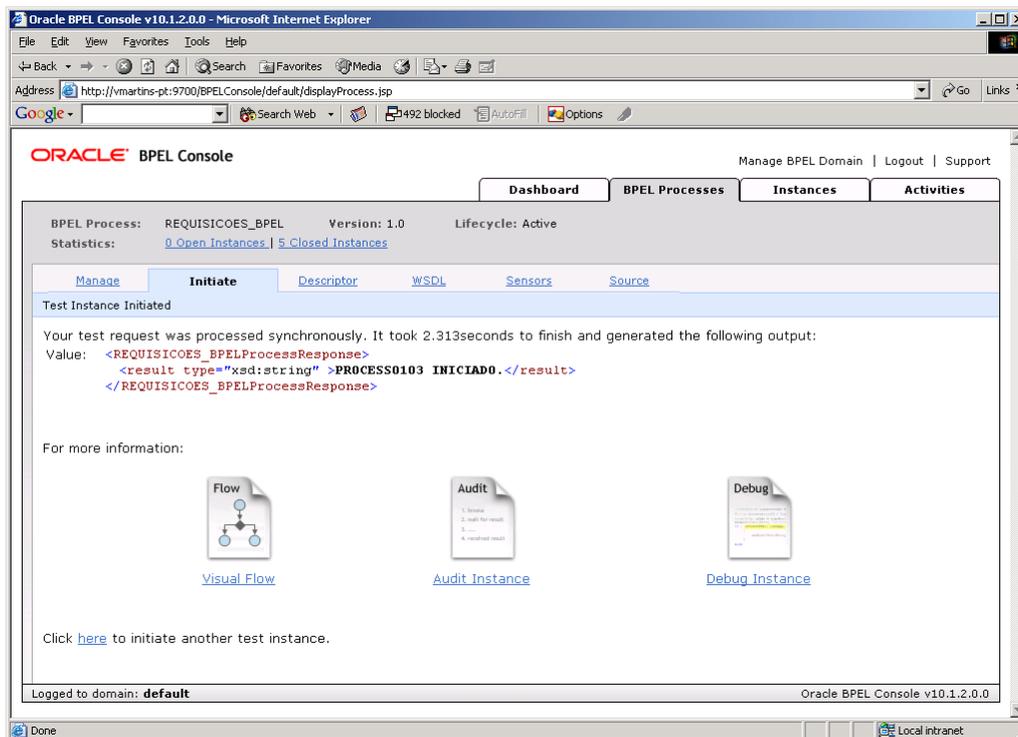


Figura 5.79: Processo BPEL para a requisição nº103 submetida em XML finalizado

5.4 Símula e conclusões

Este exemplo permitiu ilustrar duas abordagens para a integração de 3 aplicações. O uso da tecnologia Oracle deve-se ao facto de ser multi-plataforma, de seguir normas como o J2EE e o .NET e de oferecer uma das primeiras plataformas reconhecidas no mercado para suportar a norma BPEL. Para a implementação técnica deste exemplo de integração, foram seguidas duas abordagens permitindo por um lado realçar soluções técnicas para diferentes necessidades de integração, e por outro lado, ilustrar a utilização de algumas normas mais recentes como são os *Web Services* e o BPEL. Neste caso, as abordagens são complementares dado que algumas partes importantes do exemplo da primeira abordagem foram reutilizadas na segunda.

A primeira abordagem baseou-se em tecnologias comuns para a integração de cada aplicação (Tabela 5.1). A integração de componentes Java, as chamadas a interfaces aplicacionais do tipo API, a integração de informação entre bases de dados com *Database Links*, a integração de processos com *workflow* e a utilização do *Messaging Middleware* mostra-nos uma solução para este exemplo de integração. No que diz respeito aos pontos de integração esta primeira abordagem contém algumas

dependências entre sistemas. Por exemplo, sem a integração na aplicação de Requisições da componente Java, pertencente à aplicação de RH, a criação da requisição não seria feita com a informação correcta. E neste caso não há alternativa senão a da incorporação desta componente aplicacional.

Abordagens para a implementação do exemplo.	Tipo de integração e/ou tecnologia.
<p>Primeira abordagem</p>	<ul style="list-style-type: none"> • Integração aplicacional ao nível da interface com o utilizador. Integração do écran feito em Oracle Forms com uma página HTML no <i>Web Browser</i> (Introdução da informação para dar início ao pedido de requisição). • Integração de informação com componente Java e JDBC entre a aplicação de RH e a aplicação Requisições. • Integração aplicacional entre componente Java e <i>Oracle Forms</i>. • Integração da informação com <i>SQL*Net</i> e <i>Oracle Database Link</i>. • Integração aplicacional e de processos com <i>workflow</i> suportado em <i>Oracle Workflow</i> e a sua API. • Integração aplicacional com <i>Messaging</i> baseado em <i>Oracle Advanced Queueing</i>. • Integração aplicacional com correio electrónico via protocolo SMTP e código PL/SQL.
<p>Segunda abordagem</p>	<ul style="list-style-type: none"> • Integração com aplicação de Requisições via <i>Web Services</i> (Introdução da informação em XML para dar início ao pedido de requisição, suportado na interface <i>web</i> do <i>BPEL Process Manager</i>, substituindo o écran desenvolvido em Oracle Forms). • Integração com componente de <i>workflow</i> via <i>Web Services</i>. • Integração de processos com BPEL e <i>Web Services</i> (Abordagem do tipo <i>Service Oriented Architecture</i>).

Tabela 5.1: Caracterização resumida das abordagens para a implementação do exemplo.

A segunda abordagem pretende ser mais flexível e utilizar soluções mais recentes realçando o seu papel na integração de SI. Com a introdução de *Web Services* criou-se uma camada de abstracção sobre cada aplicação que passa a estar disponível como serviço. Isto permite criar pontos de integração normalizados e suficientemente flexíveis para serem, ou não, integrados noutras aplicações. O uso do BPEL veio dar uma lógica processual à utilização de cada *Web Service* disponibilizando um novo ponto de partida para os pedidos de requisições. Pelo facto de aceitar directamente informação em XML, esta segunda alternativa de implementação proporciona maior flexibilidade e independência aplicacional, ou seja, não é obrigatório utilizar uma aplicação concebida de propósito para esse efeito, como é o caso do écran da aplicação de Requisições desenvolvido em Oracle Forms 10g. Neste caso, a adopção de *Web Services* permite, por um lado, disponibilizar funcionalidades dos sistemas internos de forma a poder integrá-los, e por outro lado, dar maior adaptabilidade do processo de aprovação de requisições que é agora controlado pelo BPEL. Este controlo permite também centralizar a administração e a monitorização deste processo específico, proporcionando um nível superior de acompanhamento do desempenho e da qualidade do processo.

A arquitectura da segunda abordagem segue a filosofia de *Service Oriented Architecture* (SOA) permitindo disponibilizar aplicações e informação como serviços. Esta realidade permite a reutilização e integração destes serviços em soluções mais abrangentes e flexíveis para o desempenho do negócio. Apesar do principal propósito deste exemplo ser a ilustração de algumas formas de integração, suportadas em diferentes normas e tecnologias, a alternativa criada com a norma BPEL é, neste caso concreto, complementar a alguns dos pontos de integração implementados na primeira abordagem. Este é também um dos propósitos do BPEL e dos *Web Services* que permitem reutilizar sistemas que podem por vezes ser classificados de obsoletos pelo simples facto de não se conseguir integrar com novas soluções. Como já foi referido, cada caso de integração de SI terá diferentes abordagens onde o BPEL pode ser uma solução viável.

6. Conclusões

O tema da integração de SI é muito vasto e dinâmico. Como resultado da evolução das TI, e do seu impacto nas organizações, há hoje em dia uma panóplia de ofertas e alternativas para encontrar soluções adequadas. Um dos grandes problemas neste campo é a heterogeneidade de modelos, conceitos e catalogações tecnológicas que não ajuda no enquadramento de problemas e respectivas soluções. Nos últimos anos, tem havido uma convergência de conceitos e tecnologias que torna esta situação ainda mais complexa. Neste sentido, a adopção de normas de mercado garante a certificação e abertura das soluções implementadas nesta área.

O enquadramento da integração de SI centrado na informação, nas aplicações, nos processos e na própria organização permite distinguir quatro perspectivas de grupos tecnológicos nesta área. Em cada um deles é possível tirar partido de normas de mercado como são os *Web Services* que permitem disponibilizar informação e invocar lógica aplicacional através dos protocolos de comunicação mais comuns e utilizados na Internet. A complementaridade destas quatro abordagens com os *Web Services* permite definir uma arquitectura orientadas aos serviços, denominada por *Service Oriented Architecture (SOA)*. A norma *Business Process Execution Language (BPEL)* vem de encontro à necessidade de orquestração dos *Web Services* na implementação do SOA, dando-lhe uma perspectiva processual, e garantindo uma integração a esse nível. A perspectiva da integração entre organizações é mais abrangente tendo como principal intuito facilitar as trocas comerciais suportadas electronicamente, obrigando cada uma das organizações a integrar parte dos seus sistemas.

Um projecto para a integração de SI pode ser muito complexo e de difícil realização dada a sua abrangência e às vertentes organizacionais que engloba. Um projecto desta natureza tem impacto no sistemas de informação internos, nos diferentes departamentos que tiram partido desses sistemas, nas tomadas de decisão baseadas em informação de gestão e nas relações entre as próprias organizações. As perspectivas conceptuais descritas neste trabalho são complementares e permitem em cada uma das suas áreas seguir abordagens de integração enquadradas. Desta forma é

possível reduzir alguma desta complexidade dividindo o âmbito em áreas com diferentes necessidades.

O planeamento de projectos para a integração de SI pode também revelar-se complexo face à sua abrangência. Em cada uma das perspectivas descritas pode-se adoptar diferentes tecnologias, e configurar diferentes soluções baseadas na combinação e integração de componentes para as suportar.

Nestas últimas décadas assistiu-se a uma crescente evolução no objectivo das soluções informatizadas. Houve uma primeira fase cujo objectivo foi a digitalização da informação, recorrendo-se aos sistemas de gestão de bases de dados. Com o amadurecimento destas soluções, o foco passou a ser a criação de aplicações informáticas integradas, como são os *Enterprise Resource Planning* (ERP) que exploram as bases de dados entretanto criadas. Na última década, procurou-se a automatização dos processos organizacionais com o intuito de aumentar a sua eficiência e de integrar aplicações informáticas. Com os *Web Services* e o BPEL abrem-se novas perspectivas para a exploração e integração desses processos automatizados e das suas respectivas aplicações. Torna-se possível a flexibilização total desses processos criando soluções que se auto-adaptam consoante os requisitos e as situações. Esta realidade traz novas necessidades acerca da gestão e monitorização desse tipo de soluções que estão no âmbito das ferramentas de análise e outras tecnologias do tipo *Business Activity Monitorig* (BAM). Com a Globalização esta tendência faz cada vez mais sentido dado que as organizações deixam de estar fisicamente longínquas para estarem electronicamente próximas. A arquitectura SOA responde tecnicamente as estas questões o que a torna incontornável quando se pensa numa solução para a integração de SI. E toda esta evolução tem o seu impacto nas organizações obrigando-as por um lado a estar sempre actualizadas, sob pena de deixarem de ser competitivas, mas trazendo por outro lado novas oportunidades.

Por seu turno, e ao longo destes anos, as normas técnicas foram acompanhando a evolução da tecnologia, permitindo a sua documentação e integração. Assiste-se agora neste âmbito a uma convergência de especificações que ajudam a encontrar as soluções mais flexíveis e independentes dos fornecedores de tecnologia. A norma BPEL é a prova desta convergência que aproxima os

fornecedores da tecnologias e as suas soluções. A crescente adoção e maturidade desta norma dá maior credibilidade às soluções de integração que se baseiam nela. A importância dos processos organizacionais e da sua automatização é aqui reforçada por esta mesma norma.

Para a área da integração de SI existe uma manifesta falta de metodologias abrangentes que permitam definir, planear e documentar por completo uma abordagem para a integração de SI. Existem diversas metodologias isoladas para as diferentes perspectivas e tecnologias descritas neste trabalho, mas não existe nenhuma abordagem global e suficientemente abrangente para esta área. Fica como sugestão a definição futura de uma metodologia abrangente para a área do Planeamento, Desenvolvimento e Gestão da Integração de SI (PDGISI). Esta metodologia permitirá definir, planear e documentar uma abordagem global e fundamentada para solucionar um problema de integração de SI. Associado a esta sugestão fica também a identificação de parâmetros gerais que permitam comparar soluções de integração de SI e avaliar a sua adequabilidade numa organização, identificando as suas vantagens e retornos para a mesma.

A avaliação das estratégias e comparação das soluções para a integração de SI tem também lacunas evidentes. Dada a natureza complexa desta área, e das diferentes alternativas que se apresentam face a um mesmo problema, é difícil avaliar de forma sistemática a real adequabilidade de uma solução. Ainda neste sentido, as soluções são de difícil comparação dada a sua natureza heterogénea e a falta de pontos comparativos.

Para além das questões da diferentes perspectivas encontradas neste trabalho, e das diversas normas técnicas existentes, as soluções baseadas em *Open Source* têm tido alguma presença nesta área embora estejam ainda sujeitas à desconfiança existente devido à sua natureza aberta e não comercial. Esta realidade vem aumentar a complexidade do tema da integração de SI.

Como também já foi referido, as organizações têm cada vez mais necessidades na integração dos seus sistemas devido a uma necessidade de maior flexibilização dos seus sistemas e à crescente concorrência e abertura de mercados. Neste sentido,

surtem novos perfis profissionais nas áreas das tecnologias da informação dedicados à integração de SI.

Como última sugestão, fica o complemento do exemplo de integração, implementado e ilustrado neste trabalho, com tecnologias de análise e monitorização de actividades do tipo *Business Activity Monitoring* (BAM) que ajudará também na identificação dos parâmetros de avaliação referidos neste capítulo.

A integração de SI está em constante evolução e cruza diferentes áreas e disciplinas enquadradas nos Sistemas de Informação. É por isso importante clarificar e definir conceitos, abordagens e metodologias para um melhor entendimento desta realidade complexa. Este trabalho foi realizado com esta linha de pensamento e fica como contributo para a melhor clarificação e compreensão deste tema.

7. Apêndices

7.1 Criação dos utilizadores de base de dados Oracle

```
create user BD_REQ IDENTIFIED BY BD_REQ DEFAULT TABLESPACE USERS  
TEMPORARY  
TABLESPACE TEMP;
```

```
GRANT CONNECT, RESOURCE TO BD_REQ;
```

```
GRANT aq_administrator_role TO BD_REQ;
```

```
grant execute on dbms_aqadm to BD_REQ;
```

```
grant execute on dbms_aq to BD_REQ;
```

```
create user BD_RH IDENTIFIED BY BD_RH DEFAULT TABLESPACE USERS  
TEMPORARY  
TABLESPACE TEMP;
```

```
GRANT CONNECT, RESOURCE TO BD_RH;
```

```
GRANT aq_administrator_role TO BD_RH;
```

```
grant execute on dbms_aqadm to BD_RH;
```

```
grant execute on dbms_aq to BD_RH;
```

```
create user BD_STOCK IDENTIFIED BY BD_STOCK DEFAULT TABLESPACE USERS  
TEMPORARY  
TABLESPACE TEMP;
```

```
GRANT CONNECT, RESOURCE TO BD_STOCK;
```

```
GRANT aq_administrator_role TO BD_STOCK;
```

```
grant execute on dbms_aqadm to BD_STOCK;
```

```
grant execute on dbms_aq to BD_STOCK;
```

7.2 Comandos para a criação das tabelas e registos na base de dados

```
Create table REQUISICOES (  
    NUM_REQ Varchar2(10) NOT NULL ,  
    ID_FUNCIONARIO Varchar2(10) NOT NULL ,  
    ID_ITEM Varchar2(10) NOT NULL ,  
    DATA Date NOT NULL ,  
    NOME_REQUISITANTE Varchar2(50) NOT NULL ,  
    NOME_ITEM Varchar2(50),  
    JUSTIFICACAO Varchar2(100),  
    OBSERVACOES Varchar2(100),  
    ESTADO Varchar2(15) NOT NULL ,  
    DATA_ESTADO Date NOT NULL )  
/  
  
Create table FUNCIONARIOS (  
    ID_FUNCIONARIO Varchar2(10) NOT NULL ,  
    ID_DEPARTAMENTO Varchar2(5) NOT NULL ,  
    ID_FUNCAO Varchar2(5) NOT NULL ,  
    NOME_FUNCIONARIO Varchar2(50) NOT NULL ,  
    DATA_CONTRATO Date NOT NULL)  
/  
  
Create table FUNCOES (  
    ID_FUNCAO Varchar2(5) NOT NULL ,  
    NOME_FUNCAO Varchar2(20) NOT NULL ,  
    DESCRICAO Varchar2(100))  
/  
  
Create table DEPARTAMENTOS (  
    ID_DEPARTAMENTO Varchar2(5) NOT NULL ,  
    NOME_DEPARTAMENTO Varchar2(20) NOT NULL ,  
    LOCALIZACAO Varchar2(50),  
    DESCRICAO Varchar2(100))  
/  
  
Create table ITEMS (  
    ID_ITEM Varchar2(10) NOT NULL ,  
    ID_FORNECEDOR Varchar2(10) NOT NULL ,  
    ID_TIPO_ITEM Varchar2(10) NOT NULL ,  
    NOME_ITEM Varchar2(40),  
    DESCRICAO Varchar2(100))  
/  
  
Create table TIPOS_ITEM (  
    ID_TIPO_ITEM Varchar2(10) NOT NULL ,  
    DESCRICAO Varchar2(100))  
/  
  
Create table FORNECEDORES (  
    ID_FORNECEDOR Varchar2(10) NOT NULL ,  
    NOME_FORNECEDOR Varchar2(30) NOT NULL ,  
    DESCRICAO Varchar2(100),  
    LOCAL Varchar2(5) Default PT NOT NULL )  
/  
  
Create table ITEMS_EM_STOCK (  
    ID_ITEM_STOCK Varchar2(10) NOT NULL ,  
    ID_ITEM Varchar2(10) NOT NULL ,
```

```

        DATA_ITEM_STOCK Date NOT NULL ,
        QUANTIDADE Number NOT NULL )
/

Create table REQUISICAO_XML (
        ID_REQ_XML Varchar2(10) NOT NULL ,
        DOC_XML XMLTYPE)
/

Alter table REQUISICOES add primary key (NUM_REQ)
/
Alter table FUNCIONARIOS add primary key (ID_FUNCIONARIO)
/
Alter table FUNCOES add primary key (ID_FUNCAO)
/
Alter table DEPARTAMENTOS add primary key (ID_DEPARTAMENTO)
/
Alter table ITEMS add primary key (ID_ITEM)
/
Alter table TIPOS_ITEM add primary key (ID_TIPO_ITEM)
/
Alter table FORNECEDORES add primary key (ID_FORNECEDOR)
/
Alter table ITEMS_EM_STOCK add primary key (ID_ITEM_STOCK)
/
Alter table REQUISICAO_XML add primary key (ID_REQ_XML)
/

Alter table REQUISICOES add foreign key (ID_FUNCIONARIO) references
FUNCIONARIOS (ID_FUNCIONARIO)
/
Alter table FUNCIONARIOS add foreign key (ID_FUNCAO) references
FUNCOES (ID_FUNCAO)
/
Alter table FUNCIONARIOS add foreign key (ID_DEPARTAMENTO)
references DEPARTAMENTOS (ID_DEPARTAMENTO)
/
Alter table REQUISICOES add foreign key (ID_ITEM) references ITEMS
(ID_ITEM)
/
Alter table ITEMS_EM_STOCK add foreign key (ID_ITEM) references
ITEMS (ID_ITEM)
/
Alter table ITEMS add foreign key (ID_TIPO_ITEM) references
TIPOS_ITEM (ID_TIPO_ITEM)
/
Alter table ITEMS add foreign key (ID_FORNECEDOR) references
FORNECEDORES (ID_FORNECEDOR)
/
Alter table FUNCIONARIOS add foreign key (ID_SUPERIOR) references
FUNCIONARIO (ID_FUNCIONARIO)
/

grant select on funcionarios to owf_mgr;
grant select on requisicoes to owf_mgr;

```

```

insert into funcoes values ('1','DIRECTOR',null);
insert into funcoes values ('2','FUNCIONARIO',null);

insert into departamentos values ('1','VENDAS','PORTO',null);
insert into departamentos values ('2','MARKETING','PORTO',null);
insert into departamentos values ('3','RH','PORTO',null);

insert into funcionarios values ('3','2','1','CARLOS',to_date('20-01-1999','dd-mm-yyyy'),NULL);
insert into funcionarios values ('4','1','1','EDUARDO',to_date('10-09-1996','dd-mm-yyyy'),NULL);
insert into funcionarios values ('5','3','1','MARIA',to_date('11-03-2000','dd-mm-yyyy'),NULL);

insert into funcionarios values ('1','1','2','VICTOR',sysdate,NULL);
insert into funcionarios values ('2','2','2','CESAR',to_date('17-12-1999','dd-mm-yyyy'),NULL);

insert into tipos_item values ('1','COMPUTADOR DESKTOP');
insert into tipos_item values ('2','MONITOR');
insert into tipos_item values ('3','RATO');
insert into tipos_item values ('4','DISCO EXTERNO');
insert into tipos_item values ('5','TECLADO');
insert into tipos_item values ('6','CD');
insert into tipos_item values ('7','PLACA WIRELESS');
insert into tipos_item values ('8','TRANSFORMADOR');
insert into tipos_item values ('9','COMPUTADOR PORTATIL');

INSERT INTO FORNECEDORES VALUES ('1','FORNECEDOR 1 PT','FORNECEDOR NACIONAL','PORTO');
INSERT INTO FORNECEDORES VALUES ('2','FORNECEDOR 2 FR','FORNECEDOR ESTRANGEIRO','PARIS');

INSERT INTO ITEMS VALUES ('1','1','1','DESKTOP MARCA XPTO 2GB RAM',NULL);
INSERT INTO ITEMS VALUES ('2','1','9','PORTATIL MARCA XPTO 1GB RAM 100GB DISCO',NULL);
INSERT INTO ITEMS VALUES ('3','1','8','TRANSFORMADOR ZPTO PARA DESKTOP',NULL);
INSERT INTO ITEMS VALUES ('4','2','4','DISCO EXTERNO PARA BACKUP XZTP 200GB',NULL);
INSERT INTO ITEMS VALUES ('5','1','7','PLACA WIRELESS MARCA XZTP',NULL);

INSERT INTO ITEMS_EM_STOCK VALUES ('1','5',SYSDATE,5);
INSERT INTO ITEMS_EM_STOCK VALUES ('2','2',SYSDATE,2);
INSERT INTO ITEMS_EM_STOCK VALUES ('3','4',SYSDATE,2);

```

7.3 Database Links

```
create database link dbl_bd_rh connect to bd_rh identified by bd_rh
using 'orcl.pt.oracle.com';
```

```
create database link dbl_bd_stock connect to bd_stock identified by
bd_stock using 'orcl.pt.oracle.com';
```

7.4 Código para integração do Messaging

```
/****** COMANDOS PARA BD_REQ *****
*****/
```

```
CREATE type Requisicao_t as object (
    NUM_REQ Varchar2(10),
    ID_ITEM Varchar2(10),
    DATA Date,
    NOME_ITEM Varchar2(50));
/
```

```
begin
dbms_aqadm.create_queue_table (
Queue_table          => 'QT_1',
Multiple_consumers   => TRUE,
auto_commit          => TRUE,
Compatible           => '8.1',
Queue_payload_type   => 'Requisicao_t');
end;
/
```

```
begin
DBMS_AQADM.CREATE_QUEUE (
queue_name           => 'Q_1',
retention_time       => 3600,
auto_commit          => true,
queue_table          => 'QT_1');
end;
/
```

```
DECLARE
    subscriber sys.aq$_agent;
```

```
BEGIN
```

```
    subscriber := sys.aq$_agent('BD_STOCK', NULL, NULL);
    DBMS_AQADM.ADD_SUBSCRIBER(queue_name => 'Q_1',
                               subscriber => subscriber);
```

```
END;
/
```

```
begin
DBMS_AQADM.START_QUEUE (
```

```

queue_name      => 'Q_1',
enqueue         => TRUE,
dequeue        => TRUE);
end;
/

BEGIN

    DBMS_AQADM.SCHEDULE_PROPAGATION(queue_name => 'BD_REQ.Q_1',
                                     destination => 'dbl_bd_stock');

END;
/

/***** COMANDOS PARA BD_STOCK *****/

CREATE type Requisicao_t as object (
    NUM_REQ Varchar2(10),
    ID_ITEM Varchar2(10),
    DATA Date,
    NOME_ITEM Varchar2(50));
/

begin
dbms_aqadm.create_queue_table (
Queue_table      => 'QT_2',
Multiple_consumers => TRUE,
auto_commit      => TRUE,
Compatible       => '8.1',
Queue_payload_type => 'Requisicao_t');
end;
/

begin
DBMS_AQADM.CREATE_QUEUE (
queue_name      => 'Q_2',
retention_time  => 3600,
auto_commit     => true,
queue_table     => 'QT_2');
end;
/

begin
DBMS_AQADM.START_QUEUE (
queue_name      => 'Q_2',
enqueue         => TRUE,
dequeue        => TRUE);
end;
/

/***** Com BD_REQ - ENQUEUE *****/
*****/

DECLARE

    enqueue_options    DBMS_AQ.enqueue_options_t;
    message_properties DBMS_AQ.message_properties_t;
    recipients         DBMS_AQ.aq$_recipient_list_t;
    message_handle     RAW(16);

```

```

message          Requisicao_t;

NUM_REQ VARCHAR2(10);
ID_ITEM VARCHAR2(10);
DATA DATE;
NOME_ITEM VARCHAR2(50);

BEGIN

message := Requisicao_t(NUM_REQ, ID_ITEM, DATA, NOME_ITEM);
recipients(1) := sys.aq$_agent('BD_STOCK', NULL, NULL);
message_properties.recipient_list := recipients;

DBMS_AQ.ENQUEUE(queue_name => 'Q_1',
enqueue_options => enqueue_options,
message_properties => message_properties,
payload => message,
msgid => message_handle);

commit;
END;
/

-----

create or replace procedure submeter_mensagem_aq(NUM_REQ in VARCHAR2,
                                                ID_ITEM in VARCHAR2,
                                                DATA in DATE,
                                                NOME_ITEM in VARCHAR2) is

Pragma autonomous_transaction;

enqueue_options      DBMS_AQ.enqueue_options_t;
message_properties   DBMS_AQ.message_properties_t;
recipients            DBMS_AQ.aq$_recipient_list_t;
message_handle       RAW(16);
message              Requisicao_t;

NUM_REQ_V VARCHAR2(10);
ID_ITEM_V VARCHAR2(10);
DATA_V DATE;
NOME_ITEM_V VARCHAR2(50);

BEGIN

NUM_REQ_V := NUM_REQ;
ID_ITEM_V := ID_ITEM;
DATA_V := DATA;
NOME_ITEM_V := NOME_ITEM;

message := Requisicao_t(NUM_REQ_V, ID_ITEM_V, DATA_V, NOME_ITEM_V);
recipients(1) := sys.aq$_agent('BD_STOCK', NULL, NULL);
message_properties.recipient_list := recipients;

DBMS_AQ.ENQUEUE(queue_name => 'BD_REQ.Q_1',
enqueue_options => enqueue_options,
message_properties => message_properties,
payload => message,
msgid => message_handle);

```

```

        commit;

END;
/

--NOTA: grant execute on submeter_mensagem_aq to owf_mgr;

/***** Com BD_STOCK - DEQUEUE *****/
*****/

DECLARE
    dequeue_options      dbms_aq.dequeue_options_t;
    message_properties   dbms_aq.message_properties_t;
    message_handle       RAW(16);
    message              Requisicao_t;
    no_messages          exception;
    pragma exception_init (no_messages, -25228);

BEGIN

    dequeue_options.wait := DBMS_AQ.NO_WAIT;

BEGIN

    dequeue_options.consumer_name := 'BD_STOCK';
    dequeue_options.navigation := DBMS_AQ.FIRST_MESSAGE;
    LOOP
        DBMS_AQ.DEQUEUE(queue_name      => 'BD_REQ.Q_1',
                        dequeue_options => dequeue_options,
                        message_properties => message_properties,
                        payload          => message,
                        msgid            => message_handle);

        DBMS_OUTPUT.PUT_LINE (message.NUM_REQ||' '||message.ID_ITEM||'
'|message.DATA||' '||message.NOME_ITEM);

        dequeue_options.navigation := DBMS_AQ.NEXT_MESSAGE;

    END LOOP;
    EXCEPTION
    WHEN no_messages THEN
        DBMS_OUTPUT.PUT_LINE ('Não há mais mensagens. ');
    COMMIT;
END;
END;
/

create or replace function ler_mensagem_aq(identificacao in varchar2)
return varchar2 is

    dequeue_options      dbms_aq.dequeue_options_t;
    message_properties   dbms_aq.message_properties_t;
    message_handle       RAW(16);
    message              Requisicao_t;
    no_messages          exception;
    pragma exception_init (no_messages, -25228);

```

```

    identificacao_v varchar2(10);

BEGIN

    dequeue_options.wait := DBMS_AQ.NO_WAIT;
    --identificacao_v:=identificacao;
    identificacao_v:='BD_STOCK';

BEGIN

-- dequeue_options.consumer_name := 'BD_STOCK';
dequeue_options.consumer_name := identificacao_v;
dequeue_options.navigation := DBMS_AQ.FIRST_MESSAGE;
LOOP
    DBMS_AQ.DEQUEUE(queue_name      => 'BD_REQ.Q_1',
                    dequeue_options => dequeue_options,
                    message_properties => message_properties,
                    payload          => message,
                    msgid            => message_handle);

    DBMS_OUTPUT.PUT_LINE (message.NUM_REQ||' '||message.ID_ITEM||'
'|message.DATA||' '||message.NOME_ITEM);
    return(message.NUM_REQ||'-'||message.ID_ITEM||'-'
'|message.DATA||'-'||message.NOME_ITEM);

    dequeue_options.navigation := DBMS_AQ.NEXT_MESSAGE;

END LOOP;
EXCEPTION
WHEN no_messages THEN
    DBMS_OUTPUT.PUT_LINE ('Não há mais mensagens. ');
    Return('0');
COMMIT;
END;
END;
/

```

7.5 Integração com Messaging e integração com email da aplicação de Stocks

```
CREATE OR REPLACE PACKAGE TRATA_REQ
IS

PROCEDURE RECEBE_REQUISICAO_MESSAGING;

PROCEDURE ENVIA_EMAIL_ITEM_DISPONIVEL (NUM_REQ IN VARCHAR2);

END;

-----
CREATE OR REPLACE PACKAGE BODY TRATA_REQ
IS

PROCEDURE RECEBE_REQUISICAO_MESSAGING IS

a varchar2(100);

begin
a:=bd_req.ler_mensagem_aq('BD_STOCK');
dbms_output.put_line(a);

end;

PROCEDURE ENVIA_EMAIL_ITEM_DISPONIVEL (NUM_REQ IN VARCHAR2)
IS

conn UTL_SMTP.CONNECTION;
crlf VARCHAR2( 2 ):= CHR( 13 ) || CHR( 10 );
mesg VARCHAR2( 1000 );
BEGIN
conn:= utl_smtp.open_connection( 'emeasntp.oracle.com', 25 );
utl_smtp.helo( conn, 'emeasntp.oracle.com' );
utl_smtp.mail( conn, 'victor.martins@oracle.com' );
utl_smtp.rcpt( conn, 'victor.martins@oracle.com' );

mesg:= 'Date: ' || TO_CHAR( SYSDATE, 'dd Mon yy hh24:mi:ss' ) ||
crlf ||
'From: APROVACAO DE REQUISICOES <victor.martins@oracle.com>'
|| crlf ||
'Subject: ITEM REQUISITADO '||NUM_REQ||' DISPONIVEL' || crlf
||
'To: victor.martins <victor.martins@oracle.com>' || crlf ||
'Cc: victor.martins <victor.martins@oracle.com>' || crlf ||
'' || crlf ||
' O item da requisicao '||NUM_REQ||' esta disponivel e
reservado em armazem. '|| crlf ||
'*****';

utl_smtp.data( conn, mesg );
utl_smtp.quit( conn );
END;
end;
```

7.6 Código PL/SQL para Web Service de Inserção de Requisições

```
create or replace package requisicao_ws
is

function inserir_requisicao (NUM_REQ in varchar2,
                           ID_REQUISITANTE in varchar2,
                           ID_ITEM in varchar2,
                           JUSTIFICACAO in varchar2) return
varchar2;

end requisicao_ws;
-----
create or replace package body requisicao_ws
is

function inserir_requisicao (NUM_REQ in varchar2,
                           ID_REQUISITANTE in varchar2,
                           ID_ITEM in varchar2,
                           JUSTIFICACAO in varchar2) return
varchar2
is

nome_funcionario_v varchar2(50);
nome_item_v varchar2(50);
id_item_v varchar2(10);

begin

    id_item_v:=id_item;

    select nome_funcionario
    into nome_funcionario_v
    from bd_rh.funcionarios@dbl_bd_rh
    where id_funcionario=id_requisitante;

    select nome_item
    into nome_item_v
    from bd_stock.items@dbl_bd_stock
    where id_item=id_item_v;

    insert into requisicoes values
    (NUM_REQ,ID_REQUISITANTE,ID_ITEM,sysdate,nome_funcionario_v,nome_item
    _v,JUSTIFICACAO,'','G',sysdate);
    commit;

    return(nome_funcionario_v);

end;
end;
```

7.7 Código PL/SQL para Web Service de Workflow

```
create or replace package workflow_ws
is

function chama_workflow (NUM_REQ in varchar2,
                        ID_REQUISITANTE in varchar2,
                        NOME_REQUISITANTE in varchar2,
                        ID_ITEM in varchar2,
                        JUSTIFICACAO in varchar2) return
varchar2;

end;
-----
create or replace package body workflow_ws
is

function chama_workflow (NUM_REQ in varchar2,
                        ID_REQUISITANTE in varchar2,
                        NOME_REQUISITANTE in varchar2,
                        ID_ITEM in varchar2,
                        JUSTIFICACAO in varchar2) return varchar2
is

nome_superior_v varchar2(50);
nome_item_v varchar2(50);
id_item_v varchar2(10);

begin

    id_item_v:=id_item;

    select b.nome_funcionario
    into nome_superior_v
    from bd_rh.funcionarios@dbl_bd_rh a, bd_rh.funcionarios@dbl_bd_rh
b
    where a.id_superior = b.id_funcionario
    and a.id_funcionario=id_requisitante;

req_plsql.INICIA_PROCESSO(NUM_REQ,NOME_REQUISITANTE, ID_REQUISITANTE ,n
ome_superior_v, 'PROCESSO', 'REQ_ITEM' );

    return('PROCESSO' || NUM_REQ || ' INICIADO. ');

end;
end;
```

7.8 Código de workflow para a integração com Requisições e Messaging

```
create or replace package REQ_PLSQL as

-- PROCEDIMENTO PARA INICIAR PROCESSO DE WORKFLOW COM PARAMETROS DA
APLICAÇÃO FORMS.

procedure INICIA_PROCESSO(NUM_REQ in varchar2,
                           NOME_REQUISITANTE in varchar2,
                           ID_REQUISITANTE IN VARCHAR2,
                           SUPERIOR IN VARCHAR2,
                           PROCESSO in varchar2 default null,
                           Item_Type in varchar2 default null );

-- PROCEDIMENTO CHAMADO PELO WORKFLOW PARA VERIFICAR A REQUISIÇÃO E
TRATA OS SEUS DADOS.

procedure TRATA_REQ (   itemtype      in varchar2,
                       itemkey       in varchar2,
                       actid         in number,
                       funcmode      in varchar2,
                       resultout     out varchar2);

-- PROCEDIMENTO QUE DÁ INICIO À FASE DE ENTREGA DO ITEM REQUISITADO,
OU DADO INICIO À FASE DE ENCOMENDA

procedure DESPACHA_REQ (   itemtype      in varchar2,
                           itemkey       in varchar2,
                           actid         in number,
                           funcmode      in varchar2,
                           resultout     out varchar2);

END REQ_PLSQL;

create or replace package BODY REQ_PLSQL as

-- PROCEDIMENTO PARA INICIAR PROCESSO DE WORKFLOW COM PARAMETROS DA
APLICAÇÃO FORMS.

procedure INICIA_PROCESSO(NUM_REQ in varchar2,
                           NOME_REQUISITANTE in varchar2,
                           ID_REQUISITANTE IN VARCHAR2,
                           SUPERIOR IN VARCHAR2,
                           PROCESSO in varchar2 default null,
                           Item_Type in varchar2 default null ) is

Item_Type varchar2(30) := nvl(Item_Type, 'REQ_ITEM');
ItemKey varchar2(30) := NUM_REQ;

begin
    wf_engine.CreateProcess( Item_Type => Item_Type,
                             ItemKey   => ItemKey,
                             process   => PROCESSO );

    wf_engine.SetItemAttrText (   itemtype => itemtype,
                                  itemkey  => itemkey,
                                  aname    => 'NUM_REQ',
                                  avalue   => NUM_REQ);

--
```

```

wf_engine.SetItemAttrText (   itemtype => itemtype,
                               itemkey  => itemkey,
                               aname    => 'NOME_REQUISITANTE',
                               avalue   => NOME_REQUISITANTE);
--
wf_engine.SetItemAttrText (   itemtype => itemtype,
                               itemkey  => itemkey,
                               aname    => 'ID_REQUISITANTE',
                               avalue   => ID_REQUISITANTE);
--
wf_engine.SetItemAttrText (   itemtype => itemtype,
                               itemkey  => itemkey,
                               aname    => 'SUPERIOR',
                               avalue   => SUPERIOR);
--
wf_engine.StartProcess(       itemtype => itemtype,
                               itemkey => itemkey );
--
COMMIT;
exception
  when others then
    --

    wf_core.context('REQ_PLSQL', 'INICIA_PROCESSO', NUM_REQ, NOME_REQU
ISITANTE, PROCESSO);
        raise;
    --
end INICIA_PROCESSO;

-- PROCEDIMENTO CHAMADO PELO WORKFLOW PARA VERIFICAR A REQUISIÇÃO E -
-- TRATA OS SEUS DADOS.

procedure TRATA_REQ (   itemtype   in varchar2,
                      itemkey     in varchar2,
                      actid       in number,
                      funcmode    in varchar2,
                      resultout   out varchar2)IS

tipo_item_v varchar2(10);
id_funcionario_v varchar2(10);
id_departamento_v varchar2(5);
num_req_v varchar2(30);

BEGIN

if (funcmode = 'RUN') then

    num_req_v := wf_engine.GetItemAttrText(itemtype => itemtype,
                                           itemkey => itemkey,
                                           aname => 'NUM_REQ' );

    select id_item, id_funcionario
    into tipo_item_v, id_funcionario_v
    from bd_req.requisicoes
    where num_req = num_req_v;

    select id_departamento
    into id_departamento_v
    from bd_rh.funcionarios
    where id_funcionario = id_funcionario_v;

```

```
-- Área de código para validação de requisição com parâmetros
errados. Uma regra de negócio simples
-- diz que quem pertence ao departamento de Vendas não pode
requisitar computadores do tipo desktop.
```

```

if (tipo_item_v='1' and id_departamento_v='1' ) then
    --
    wf_engine.SetItemAttrText (   itemtype => itemtype,
                                itemkey  => itemkey,
                                aname    => 'OBSERVACOES',
                                avalue   => 'O Departamento de
Vendas não pode requisitar este tipo de item.');
```

```

    resultout := 'COMPLETE:F';
    --
else
    --
    resultout := 'COMPLETE:T';
    --
end if;
end if;
```

```
END TRATA_REQ;
```

```
-----
-- PROCEDIMENTO QUE DÁ INICIO À FASE DE ENTREGA DO ITEM REQUISITADO,
OU DADO INICIO À FASE DE ENCOMENDA
-----
```

```
procedure DESPACHA_REQ (      itemtype   in varchar2,
                             itemkey    in varchar2,
                             actid      in number,
                             funcmode   in varchar2,
                             resultout  out varchar2)IS
```

```
num_req_v varchar2(30);
```

```
BEGIN
```

```
-- Nesta fase é feito o enqueue no AQ Messaging --
```

```
num_req_v := wf_engine.GetItemAttrText(itemtype => itemtype,
                                       itemkey  => itemkey,
                                       aname    => 'NUM_REQ' );
```

```
bd_req.submeter_mensagem_aq (NUM_REQ_v, ID_ITEM, DATA, NOME_ITEM);
```

```
resultout := 'COMPLETE:';
```

```
END DESPACHA_REQ;
```

```
-----
END REQ_PLSQL;
```

7.9 Classe Java para integração com aplicação de RH

```
import oracle.jdbc.driver.*;

import java.awt.*;
import java.sql.*;
import java.net.*;

public class valida_bd_rh extends Canvas{

private String data;

public valida_bd_rh()
{

try { DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
} catch (SQLException e)
{
// Tratamento de erros.
System.out.println("Erro de SQL.");
}
}

public static String valida_utilizador(String data)
{
Connection conn;
ResultSet rset = null;
Statement stmt = null;
String valor = null;

System.out.println("Inicio conexão Java à BD aplicação RH");

try {
String connect_string="jdbc:oracle:thin:@vmartins-pt.pt.oracle.com:1521:ORCL";
conn = DriverManager.getConnection (connect_string,"bd_rh","bd_rh");
stmt = conn.createStatement ();
String queryString = "select
a.id_funcionario,a.nome_funcionario,b.nome_funcionario from funcionarios a,
funcionarios b where a.id_superior = b.id_funcionario and
a.nome_funcionario=func_v";

rset = stmt.executeQuery(queryString);
rset.next();

System.out.println(" -----");
System.out.println(" ID Funcionário -> " + rset.getString(1));
System.out.println(" Nome Funcionário -> " + rset.getString(2));
System.out.println(" Nome Superior -> " + rset.getString(3));
```

```
valor=rset.getString(1);
rset.close();
stmt.close();
System.out.println ("Fecha conexão Aplicação RH.");
conn.close();
System.out.println ("Envia valores para Forms.");
}
catch (Exception e)
{
    System.out.println ("Erro.");
    e.printStackTrace();
}
return(valor);
}
}
```

7.10 Definição BPEL

```
<process name="REQUISICOES_BPEL"
targetNamespace="http://xmlns.oracle.com/REQUISICOES_BPEL"
xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns:xp20="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.functions.Xpath20" xmlns:ns1="http://inserir_req_ws/inserir_requisicao.wsdl"
xmlns:ldap="http://schemas.oracle.com/xpath/extension/ldap"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:ns2="http://chama_wf_ws/chama_wf_ws.wsdl"
xmlns:bpelx="http://schemas.oracle.com/bpel/extension"
xmlns:client="http://xmlns.oracle.com/REQUISICOES_BPEL"
xmlns:ora="http://schemas.oracle.com/xpath/extension"
xmlns:orcl="http://www.oracle.com/XSL/Transform/java/oracle.tip.pc.services.functions.ExtFunc"><!--
=====
==== --><!-- PARTNERLINKS --><!-- List of services participating in this BPEL
process --><!--
=====
==== -->
<partnerLinks><!--
  The 'client' role represents the requester of this service. It is
  used for callback. The location and correlation information associated
  with the client role are automatically set using WS-Addressing.
  -->
  <partnerLink name="client" partnerLinkType="client:REQUISICOES_BPEL"
myRole="REQUISICOES_BPELProvider"/>
  <partnerLink name="PartnerLink_Inserir_requisicao"
partnerRole="Inserir_requisicaoPortType_Role"
partnerLinkType="ns1:Inserir_requisicaoPortType_PL"/>
  <partnerLink name="PartnerLink_Chama_workflow"
partnerRole="Chama_wf_wsPortType_Role"
partnerLinkType="ns2:Chama_wf_wsPortType_PL"/>
</partnerLinks><!--
=====
==== --><!-- VARIABLES --><!-- List of messages and XML documents used
within this BPEL process --><!--
=====
==== -->
<variables><!-- Reference to the message passed as input during initiation --><!--
  Reference to the message that will be returned to the requester
  -->
  <variable name="inputVariable"
messageType="client:REQUISICOES_BPELRequestMessage"/>
  <variable name="outputVariable"
messageType="client:REQUISICOES_BPELResponseMessage"/>
  <variable name="Inserir_Requisicao_inserirRequisicao_InputVariable"
messageType="ns1:inserirRequisicao0Request"/>
```

```

    <variable name="Inserir_Requisicao_inserirRequisicao_OutputVariable"
messageType="ns1:inserirRequisicao0Response"/>
    <variable name="Chama_Workflow_chamaWorkflow_InputVariable"
messageType="ns2:chamaWorkflow0Request"/>
    <variable name="Chama_Workflow_chamaWorkflow_OutputVariable"
messageType="ns2:chamaWorkflow0Response"/>
</variables><!--
=====
==== --><!-- ORCHESTRATION LOGIC --><!-- Set of activities coordinating the
flow of messages across the --><!-- services integrated within this business process
--><!--
=====
==== -->
<sequence name="main"><!-- Receive input from requestor.
Note: This maps to operation defined in REQUISICOES_BPEL.wsdl
-->
    <receive name="receiveInput" partnerLink="client"
portType="client:REQUISICOES_BPEL" operation="process"
variable="inputVariable" createInstance="yes"/><!-- Generate reply to synchronous
request -->
    <assign name="Assign_1">
        <copy>
            <from variable="inputVariable" part="payload"
query="/client:REQUISICOES_BPELProcessRequest/client:NUM_REQ"/>
            <to variable="Inserir_Requisicao_inserirRequisicao_InputVariable"
part="numReq"/>
        </copy>
        <copy>
            <from variable="inputVariable" part="payload"
query="/client:REQUISICOES_BPELProcessRequest/client:ID_ITEM"/>
            <to variable="Inserir_Requisicao_inserirRequisicao_InputVariable"
part="idItem"/>
        </copy>
        <copy>
            <from variable="inputVariable" part="payload"
query="/client:REQUISICOES_BPELProcessRequest/client:ID_REQUISITANTE"/>
            <to variable="Inserir_Requisicao_inserirRequisicao_InputVariable"
part="idRequisitante"/>
        </copy>
        <copy>
            <from variable="inputVariable" part="payload"
query="/client:REQUISICOES_BPELProcessRequest/client:JUSTIFICACAO"/>
            <to variable="Inserir_Requisicao_inserirRequisicao_InputVariable"
part="justificacao"/>
        </copy>
    </assign>
    <invoke name="Inserir_Requisicao" partnerLink="PartnerLink_Inserir_requisicao"
portType="ns1:Inserir_requisicaoPortType" operation="inserirRequisicao"
inputVariable="Inserir_Requisicao_inserirRequisicao_InputVariable"
outputVariable="Inserir_Requisicao_inserirRequisicao_OutputVariable"/>

```

```

<assign name="Assign_2">
  <copy>
    <from variable="inputVariable" part="payload"
query="/client:REQUISICOES_BPELProcessRequest/client:NUM_REQ"/>
    <to variable="Chama_Workflow_chamaWorkflow_InputVariable"
part="numReq"/>
  </copy>
  <copy>
    <from variable="inputVariable" part="payload"
query="/client:REQUISICOES_BPELProcessRequest/client:ID_ITEM"/>
    <to variable="Chama_Workflow_chamaWorkflow_InputVariable"
part="idItem"/>
  </copy>
  <copy>
    <from variable="inputVariable" part="payload"
query="/client:REQUISICOES_BPELProcessRequest/client:ID_REQUISITANTE"/>
    <to variable="Chama_Workflow_chamaWorkflow_InputVariable"
part="idRequisitante"/>
  </copy>
  <copy>
    <from variable="inputVariable" part="payload"
query="/client:REQUISICOES_BPELProcessRequest/client:JUSTIFICACAO"/>
    <to variable="Chama_Workflow_chamaWorkflow_InputVariable"
part="justificacao"/>
  </copy>
  <copy>
    <from variable="Inserir_Requisicao_inserirRequisicao_OutputVariable"
part="return"/>
    <to variable="Chama_Workflow_chamaWorkflow_InputVariable"
part="nomeRequisitante"/>
  </copy>
</assign>
<invoke name="Chama_Workflow" partnerLink="PartnerLink_Chama_workflow"
portType="ns2:Chama_wf_wsPortType" operation="chamaWorkflow"
inputVariable="Chama_Workflow_chamaWorkflow_InputVariable"
outputVariable="Chama_Workflow_chamaWorkflow_OutputVariable"/>
  <assign name="Assign_3">
    <copy>
      <from variable="Chama_Workflow_chamaWorkflow_OutputVariable"
part="return"/>
      <to variable="outputVariable" part="payload"
query="/client:REQUISICOES_BPELProcessResponse/client:result"/>
    </copy>
  </assign>
  <reply name="replyOutput" partnerLink="client"
portType="client:REQUISICOES_BPEL" operation="process"
variable="outputVariable"/>
</sequence>
</process>

```

Referências bibliográficas

[ACS 2004] Australian Computer Society (ACS), "*Integration using EAI or Web services - which is a better choice?*", Australian Computer Society, <http://www.acs.org.au/Certification/Documents/ITT/2003ITT-Integration.pdf>, 2004.

[Allen 2001] Allen, D., "*Don't Let EAI Tools Get in the Way of Good Design*", http://www.eaiquadrant.com/str/allen_2.html, 2001.

[Alves 1995] Alves, M., "*A Reengenharia dos Processos de Negócio*", Texto Editora, 1995.

[Arkhipenkov e Golubev 2001] Arkhipenkov, S. e D. Golubev, "*Oracle Express OLAP*", Charles River Media, 2001.

[Bernstein 1996] Bernstein, P., "*Middleware: A Model for Distributed Services*", Communications of the ACM, 1996.

[Bertino 1991] Bertino E., "*Integration of Heterogeneous Data Repositories by Using Object-Oriented Views*", Workshop on Interoperability in Multidatabase Systems, 1991.

[Birrell e Nelson 1984] Birrell, A.D. e Bruce J. Nelson, "*Implementing Remote Procedure Calls*", ACM Transactions on Computer Systems, 1984.

[BPMI 2002] BPMI, "*BPML|BPEL4WS: A Convergence Path toward a Standard BPM Stack*", BPMI, <http://www.bpml.org/downloads/BPML-BPEL4WS.pdf>, 2002.

[Brown 1996] Brown, A., "*Preface: Foundations for Component-Based Software Engineering*", IEEE Computer Society Press, http://media.wiley.com/product_data/excerpt/8X/08186771/081867718X-1.pdf, 1996.

[Bussler 2003] Bussler, C., “*B2B Integration: Concepts and Architecture*”, Springer, 2003.

[Carnegie 1997] Carnegie Mellon Software Engineering Institute (SEI), “Software Technology Roadmap”, Carnegie Mellon SEI, <http://www.sei.cmu.edu/str/versions/str.7.28.97.pdf>, 1997.

[Carroll 2000] Carroll, J., “*The Economics of Integration*”, Integrated Design Inc., <http://www.eaiindustry.org/docs/TheEconomicsOfIntegration.pdf>, 2000.

[Carvalho 2001] Carvalho, J.M., “*e-Business & e-Commerce: On & Offline*”, Edições Sílabo, 2001

[Cerami 2002] Cerami, E., “*Web Services Essentials*”, O’Reilly, 2002.

[Chartier 2004] Chartier R., “*Application Architecture: An N-Tier Approach*”, <http://www.15seconds.com/issue/011023.htm>, 2004.

[Chase 2000] Chase P., “*The Seven Hidden Challenges of Application Integration*”, <http://oracle.ittoolbox.com/>, Scribe Software Corporation, 2000.

[Cherry 2000] Cherry Tree & Co., “*Extended Enterprise Applications: Spotlight Report*”, <http://www.cherrytreeco.com>, 2000.

[Chopra e Meindl 2000] Chopra, S. e P. Meindl, “*Supply Chain Management: Strategy, Planning and Operations*”, Prentice Hall, 2000.

[Claybrook 1992] Claybrook, B., “*OLTP: Online Transaction Processing Systems*”, John Wiley & Sons, 1992.

[COM 1998] Stearns D., “*COM: Component Object Model Technologies*”, Microsoft Corporation, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncomg/html/msdn_basicpmd.asp, 1998.

[Coulouris et al. 2000] Coulouris, G., Dollimore, J., Kindberg, T., “Distributed Systems: Concepts and Design (3rd Edition)”, Addison Wesley, 2000.

[Craggs 2003] Craggs, S., “*Raising EAI Standards*”, Saint Consulting Limited, <http://www.eaiindustry.org/docs/Raising%20EAI%20standards.pdf>, 2003.

[CSC 2002] Computer Sciences Corporation (CSC), “*The Emergence of Business Process Management*”, CSC, http://www.ebizq.net/hot_topics/bpm/features/1763.html, 2002.

[CSC 2003] Computer Sciences Corporation (CSC), “*The Architecture rEvolution: Exploring the Network Effect on Infrastructure, Applications and Business Process*”, http://www.csc.com/aboutus/lef/uploads/LEF_ARCHREV.pdf, 2003.

[Date 2003] Date, C.J., “*An Introduction do Database Systems (Eighth edition)*”, Addison-Wesley Publishing Company, 2003.

[Deplhi 2002] Delphi Group, “*Perspectives on Information Retrieval 2002*”, http://www.delphigroup.com/research/ir_perspectives_sum.pdf, 2002.

[Horstman e Kirtland 1997] Horstmann, M. e M. Kirtland, “*DCOM: Distributed Component Object Model*”, Microsoft Corporation, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn_dcomarch.asp, 1997.

[Delphi 2002] Delphi Group (DG), “*BPM2002 Market Milestone Report*”, DG, http://www.delphigroup.com/coverage/bpm_webservices.htm, 2002.

[Dickman 1995] Dickman, A., “*Two-Tier Versus Three-Tier Apps*”, Informationweek 553, <http://www.informationweek.com/553/53olapp.htm> , 1995.

[Dubray 2003] Dubray, J., “*Business Process Engines in Application Architecture*”, Attachmate, http://www.ebpml.org/technoforum_2003_b_eng.ppt, 2003.

[Earl 1989] Earl, M.J., "Management Strategies for Information Technology", Prentice-Hall, 1989.

[Eckerson 1995] Eckerson, W., "Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications", Open Information Systems, 1995.

[Edelstein 1994] Edelstein, H., "Unraveling Client/Server Architecture", <http://www.dbmsmag.com>, DBMS Magazine, 1994.

[Edwards e Newing 2000] Edwards, P. e R. Newing, "Application Integration for e-Business Strategies for integrating key applications, systems and processes", Optima Media Group, http://www.business-intelligence.co.uk/reports/app_int/default.asp, 2000.

[Fischer 2003] Fischer, L., "Workflow Handbook 2003 10Th year anniversary edition", <http://www.wfmc.org>, WFMC, 2003.

[Georgakopoulos et al. 1995] Georgakopoulos D., H. Hornick e A. Sheth, "An overview of workflow management: From process modeling to infrastructure for automation", Distributed and Parallel Databases Journal, 1995.

[Gunter 1995] Gunter, D., "Client/Server Programming With RPC and DCE", QUE, 1995.

[Hepper 2003] Hepper S. e S. Hesmer, "Introducing the Portlet Specification", JavaWorld, <http://www.javaworld.com/javaworld/jw-08-2003/jw-0801-portlet.html>, 2003.

[Hines 1996] Hines, John R., "Software Engineering", IEEE Spectrum, 1996.

[Hollingsworth 2004] Hollingsworth D., "The Workflow Reference Model 10 Years", WfMC, http://www.wfmc.org/standards/docs/Ref_Model_10_years_on_Hollingsworth.pdf, 2004.

[Hohpe et al. 2003] Hohpe G. e B. Woolf, “*Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*”, Addison Wesley, 2003.

[Houston 1998] Houston, P., ”*Building Distributed Applications with Message Queuing Middleware*”, Microsoft Corporation, <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmqqc/html/bldappmq.asp>, 1998.

[Hudson e Johnson 1994] Hudson, D. e J. Johnson, “*Client-Server Goes Business Critical.*” Dennis, MA: The Standish Group International, 1994.

[IEEE 1990] Institute of Electrical and Electronics Engineers, “IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries”, 1990.

[Integr8 2005] Integr8 Consulting, “Services: Styles of Integration”, <http://www.integr8consulting.com>, 2005.

[Jenkins et al. 2004] Jenkins, T., D. Glazer e H. Schaper, “*Enterprise Content Management Technology: What You Need to Know*”, Open Text Corporation, 2004.

[Kelly 2003] Kelly, D., “*Core to your Business*”, Oracle Magazine, <http://www.oracle.com/oramag/index.html>, 2003.

[Kenneth e Laudon 2003] Kenneth C. e Jane P. Laudon, “*Management Information Systems Eighth Edition*”, Prentice Hall, 2003

[Kochar 2005] Kochar, H., “Oracle Business Activity Monitoring”, Oracle Corporation, http://www.oracle.com/technology/products/integration/bam/pdf/BAM_Whitepaper.pdf, 2005.

[Levine 2005] Levine, R., “*EAI Roads to Riches*”, Business Integration Journal, <http://www.bijonline.com/PDF/levine%20july-aug.pdf>, 2005.

[Lewis 2000] Lewis, R. 2000, “*Advanced Messaging Applications with MSMQ and MQSERIES*”, Que Professional, 2000.

[Lopes e Ramalho 2005] Lopes, C. e J. Ramalho, “*Web Services Aplicações Distribuídas sobre Protocolos Internet*”, 2005

[Magic 2003] Magic Software, “The Business of Process Integration”, Magic Software, http://www.ebizq.net/white_papers/3237.html, 2003.

[McKie 2003] McKie, S., “*The Big BAM*”, Intelligent Enterprise Magazine (IEM), http://www.intelligententerprise.com/030718/612feat3_1.jhtml?_requestid=457248, 2003.

[Miers 2005] Miers D., “*BPM—Too Much BP, Not Enough of the M*”, Wfmc, http://www.wfmc.org/information/Miers_too_%20much_BP.pdf, 2005.

[Mintzberg 1979] Mintzberg, H., “*The Structuring of Organizations*”, Prentice-Hall, 1979.

[Morgenthal 2005] Morgenthal, J., “*Enterprise Information Integration: A Pragmatic Approach*”, Lulu Press, 2005.

[Newcomer e Lomow 2004] Newcomer, E. e G. Lomow, Addison-Wesley Professional, “*Understanding Soa With Web Services*”, 2004.

[Open 1997] Open Group, “*CDE 1.1: Remote Procedure Call*”, <http://www.opengroup.org/onlinepubs/9629399>, Open Group, 1997.

[Papakonstantinou et al.1995] Papakonstantinou Y., H. Garcia-Molina e J. Widom, “*Object Exchange Across Heterogeneous Information Sources*”, IEEE International Conference on Data Engineering (ICDE), 1995.

[Pereira 1998] Pereira, José Luis, “*Tecnologia de bases de dados*”, FCA, 1998.

[Plesums 2002] Plesums, C., “*Introduction to Workflow*”, WFMC, http://www.wfmc.org/information/introduction_to_workflow02.pdf, 2002.

[Ramalho e Henriques 2002] Ramalho, J. e P. Henriques, “*XML & XSL da teoria à prática*”, FCA, 2002.

[Rao 1995] Rao, B., Data Communications International, “*Making the Most of Middleware*”, 1995.

[Reynolds 2002] Reynolds, J., “*A Practical Guide to CRM.*”, CMP Books, 2002.

[Serain 2002] Serain, D., “*Middleware and Enterprise Application Integration.*”, Springer, 2002.

[Schmidt 2000] Schmidt, J., “*Enabling Next-Generation Enterprises*”, Business Integration Journal, <http://www.bijonline.com/Article.asp?ArticleID=48>, 2000.

[Schreiber 1995] Schreiber, R., “*Middleware Demystified*”, Cahners Publishing Company, 1995

[Schussel 1996] Schussel, G., “*Client/Server Past, Present and Future*” <http://news.dci.com/geos/dbsejava.htm>, 1996.

[Serrano e Fialho 2003] Serrano, A. e C. Fialho, “*Gestão do Conhecimento: o novo paradigma das organizações*”, FCA, 2003.

[Silva 2003] Silva M., “*A Integração de Sistemas de Informação*”, FCA, 2003.

[Spiers 2005] Spiers, J., “*Ten Ways to Sabotage Your Application Integration Effort*”, <http://www.eaiindustry.org/docs/TenWaystoSabotageYourApplication.pdf>, 2005.

[Steinke 1995] Steinke, S., “*Middleware Meets the Network.*”, LAN The Network Solutions Magazine, 1995

- [Summer 2004] Summer, M., “*Enterprise Resource Planning.*”, Prentice Hall, 2004
- [Thalheim 2000] Thalheim, B., “*Entity-Relationship Modeling: Foundations of Database Technology*”, Springer, 2000.
- [Trauth 1989] Trauth, E.M., “*The Evolution of Information Resource Management.*”, Information & Management, 1989.
- [Ullman 1997] Ullman, J.D., “*Information integration using logical views*”, International Conference on Database Theory (ICDT) vol. 1186, 1997.
- [Ulrich 2002] Ulrich, W., “*Synchronize EAI with Tactical and Strategic initiatives*”, <http://www.eaiindustry.org/docs/SynchronizeEAIwithTacticalandStrategic.pdf>, 2002
- [Vitt et al 2002] Vitt, E., M. Luckevich e S. Misner, “*Business Intelligence*”, Microsoft Press, 2002.
- [Watt 2002] Watt, D, “*E-Business Implementation: a guide to web services, EAI, BPI, e-commerce, content management, portals and supporting technologies*”, Butterworth Heinemann, 2002
- [Westphal e Blaxton 1998] Westphal, C. e T. Blaxton, “*Data Mining Solutions: Methods and Tools for Solving Real-World Problems*”, Wiley, 1998.
- [William et al. 2000] William, R., M. Francis e B. Wiliam, “*Enterprise Application Integration: A Wiley Tech Brief*”, John Wiley & Sons, 2000.
- [Wollrath e Waldo 2005] Wollrath A. e J. Waldo, “*An overview of RMI applications.*”, Sun Developer Network, <http://java.sun.com/docs/books/tutorial/rmi/overview.html>, 2005.
- [Wong 2001] Wong, S., “*Web Services: The Next Evolution of Application Integration*”, Integration Consortium, <http://www.eaiindustry.org/docs/WebServicesTheNextEvolutionofApplicationIntegration.pdf>, 2001.

[Woods 2003] Woods, D., “*Packaged Composite Applications: An O’Reilly Field Guide to Enterprise Software*”, O’Reilly, 2003.

[Wrazen 2000] Wrazen, E., “*EAI Tools are not the Complete Solution*”, <http://www.xephon.com/islibrary/islibrary.php?todo=eq.5.000100&subject=2>, 2000.

[W3C 2004] W3C, “*Web Services Choreography Description Language Version 1.0*”, W3C, <http://www.w3.org/TR/ws-cdl-10,2004>.

[Yee 2001] Yee A., “*Integrating Your e-Business Enterprise*”, SAMS, 2001.

[Zahavi 1999] Zahavi, R., “*Entreprise Application Integration with CORBA*”, John Wiley & Sons, 1999.