

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Utilização do Oracle BPEL Process Manager

Ivo André da Rocha Marinho

Relatório de Projecto realizado no âmbito do
Mestrado Integrado em Engenharia Informática e Computação

Orientador: Prof. Gabriel Torcato David

Co-orientador: Eng. Manuel Machado

Julho de 2008

Utilização do Oracle BPEL Process Manager

Ivo André da Rocha Marinho

Relatório de Projecto realizado no âmbito do
Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas publicas pelo Júri:

Presidente: Rosaldo José Fernandes Rossetti (Eng.)

Arguente: José Carlos Leite Ramalho (Eng.)

Vogal: Gabriel de Sousa Torcato David (Eng.)

15 de Julho de 2008

Resumo

Será apresentado neste documento o projecto final de estágio com o título “Utilização do Oracle BPEL Process Manager”, desenvolvido por Ivo André da Rocha Marinho, aluno finalista do Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto.

A realização deste projecto possibilitou a reformulação de um Sistema de Informação, de modo a implementar os conceitos e objectivos da Arquitectura SOA, tornando-o num sistema distribuído e tirando partido de todas as vantagens desta arquitectura. Para isto, foi necessário analisar e encontrar soluções de integração de uma nova linguagem de definição e automatização de processos de negócio, denominada de Web Services Business Process Execution Language WS-BPEL, ou apenas BPEL como é mais conhecida.

Pretendia-se com este estudo, especificar e implementar uma nova arquitectura distribuída, que utilizasse a linguagem BPEL como o standard de definição dos processos de negócio que compõem os diversos módulos existentes neste sistema. Para isto, foi ainda necessário criar novas soluções de evolução e conversão para BPEL, dos actuais processos de negócio que se encontram definidos em Oracle Workflow. Deste modo, foi especificada para este projecto, a reformulação de um módulo do Sistema de Informação, convertendo todo o seu processo de negócio para BPEL e reformulando todo o seu motor de lógica e interfaces de modo a integrar-se com a nova arquitectura distribuída.

Todos os componentes desenvolvidos para este projecto serão devidamente apresentados e detalhados ao longo deste documento. Dada ainda a especial importância e o crescimento de interesses nas suas tecnologias e conceitos utilizados, este reflecte-se como um documento inovador, cujos conteúdos apresentam igualmente soluções inovadoras e referem detalhes de integração e implementação de extrema importância, tanto para a evolução dos restantes módulos deste sistema como para o desenvolvimento de futuras aplicações de software de alto nível.

Abstract

This document will present the final stage project with the title "Using Oracle BPEL Process Manager", developed by Ivo André da Rocha Marino, finalist student for the Integrated Masters in Information Technology and Computer Engineering in the Faculty of Engineering of the University of Oporto.

The completion of this project enabled the reformulation of an Information System in order to implement the concepts and objectives of the SOA Architecture, making it a distributed system and taking advantage of all the strengths of this architecture. For this, it was necessary to analyze and find solutions to integrate a new language for definition and automation of business processes, called Web Services Business Process Execution Language WS-BPEL, or just BPEL as is better known.

The aim of this study was to specify and implement a new distributed architecture, which use the BPEL language as the main standard definition for all the business processes defined in this system. For this, it was also necessary to create new solutions, which develop and converts to BPEL the existing business processes that are defined with Oracle Workflow. Thus, it was specified for this project, the recasting of an example module of the Information System, converting its business process to BPEL and reformulating the entire engine and interfaces to integrate them with the new distributed architecture.

All components developed for this project, will be properly presented and detailed throughout this document. Although, given the importance and the growth of interests that the technologies and concepts used in this project have now, this is also reflected as a innovative document, whose contents represents new solutions and details of integration and implementation that are very important for the evolution of the remaining modules of the system and for the development of future applications in high-level software.

Agradecimentos

Embora a realização deste projecto tenha sido, pela sua finalidade académica, um trabalho individual, há que realçar a necessidade de expressar os meus sinceros agradecimentos a um conjunto de pessoas que contribuíram para o seu sucesso.

Deste modo, estou especialmente agradecido ao Eng. Manuel Machado e ao Prof. Gabriel Torcato David, por toda a orientação e confiança depositada em mim para a concretização deste trabalho e pela disponibilidade e amizade que sempre me demonstraram.

Aproveito ainda para agradecer a todos os restantes colaboradores da instituição, em especial ao Glaucio Guerra e ao Eng. António Cunha, de quem obtive um total apoio e auxílio fundamentais para a elaboração deste trabalho, e às demais pessoas que estiveram envolvidas neste projecto.

Por fim, não poderia deixar de dar um especial agradecimento a toda a minha família e amigos, por todo o amor, amizade e conselhos demonstrados, que me permitiram ultrapassar diversas dificuldades e inculcar o estudo, o trabalho e a realização profissional como os passos fundamentais para um futuro cheio de sucessos.

A todos os meus mais sinceros agradecimentos.

Ivo André da Rocha Marinho

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	A Instituição	2
1.3	Projecto	2
1.4	Motivação e Objectivos	3
1.5	Organização do Documento	4
2	Motivação do Projecto	7
2.1	Plano e Calendarização	9
2.2	Tecnologias e Ferramentas Utilizadas	9
3	Arquitectura Orientada a Serviços	11
3.1	Tecnologia BPEL	13
3.1.1	Principais Características e Funcionalidades	16
3.1.2	Exemplo de um Processo de Negócio	16
3.1.3	Aplicações e Servidores Disponíveis	19
3.2	Oracle BPEL Process Manager	21
3.2.1	Arquitectura do Oracle BPEL Process Manager	22
3.2.1.1	Desenho de Processos BPEL	23
3.2.1.2	Servidor BPEL	25
3.2.1.3	Consola BPEL	26
3.2.1.4	Repositório de Dados	27
3.2.2	Comparação do Oracle BPEL com Outros Servidores	28
3.3	Resumo e Conclusões	29
4	Análise do Sistema na Versão Oracle Workflow	31
4.1	Processos de Negócio	32
4.2	Modelo Relacional de Dados	34
4.3	Interfaces	34
4.4	Administração de Workflow	35
4.5	Resumo e Conclusões	36
5	Transformação de Oracle Workflow para BPEL	39
5.1	Lista de Actividades BPEL	41
5.2	Normas de Conversão de Workflow para BPEL	41
5.3	Análise Comparativa entre o BPEL e o Oracle Workflow	45
5.4	Resumo e Conclusões	46

CONTEÚDO

6	Especificação de Evolução de um Módulo	49
6.1	Serviço de Identificação	49
6.2	A Arquitectura SIBPEL	50
6.3	Especificação do Novo Trouble Tickets	54
6.3.1	Requisitos Funcionais	54
6.3.2	Requisitos Não Funcionais	55
6.3.3	Casos de Utilização	56
6.3.3.1	Actores	56
6.3.3.2	Módulo de Reportar Problemas	57
6.3.3.3	Módulo de Gestão de Problemas	57
6.4	Especificação do Sistema de Administração BPEL	58
6.4.1	Casos de Utilização	59
7	Projecto e Implementação	61
7.1	Arquitectura Lógica do Sistema	61
7.1.1	Decomposição Horizontal	61
7.1.1.1	Interface para Utilizador	61
7.1.1.2	Lógica de Negócio	63
7.1.1.3	Acesso a Dados	64
7.1.1.4	Camada de Dados	65
7.1.1.5	Segurança	65
7.1.2	Decomposição Vertical	65
7.2	Arquitectura Física do Sistema	66
7.3	Sistema SIBPEL	67
7.4	Estrutura de Serviços	74
7.5	O Novo Trouble Tickets	76
7.5.1	Definição do Processo de Negócio BPEL	76
7.5.2	Detalhes da Integração do Módulo com BPEL	80
7.5.3	Modelo Relacional de Dados	81
7.5.4	Motor de Lógica	82
7.5.5	Novas Interfaces	83
7.5.5.1	Reportar Problema	84
7.5.5.2	Detalhes Ticket	86
7.5.5.3	Pesquisa de Tickets e Tarefas	89
7.5.5.4	Listagens Rápidas	89
7.6	O Sistema de Administração BPEL	90
7.6.1	Consola BPEL	90
7.6.2	Gestor de Tarefas	92
8	Testes e Resultados	95
8.1	Avaliação Global do Projecto	95
8.2	Caso de Utilização Simples	97
8.3	Caso de Utilização Completo	99
9	Conclusões e Perspectivas de Trabalho Futuro	103
9.1	Satisfação dos Objectivos	103
9.2	Trabalho Futuro	104

CONTEÚDO

Referências	107
A Diagramas	109
A.1 Actividades do Processo de Negócio BPEL	109
A.2 Processo de Negócio BPEL de um Ticket	109
A.3 Processo de Negócio BPEL de uma Tarefa	109
A.4 Modelo Relacional do Novo Módulo de Trouble Tickets	109

CONTEÚDO

Lista de Figuras

2.1	Diagrama de Gantt do Plano de Realização do Projecto	10
3.1	Visão geral da Arquitectura SOA	13
3.2	Tecnologias do Projecto integradas na Arquitectura SOA	13
3.3	Exemplo da Agregação de Serviços num Processo de Negócio	15
3.4	Exemplo da Implementação Real de um Processo BPEL	20
3.5	Arquitectura do Oracle BPEL Process Manager	23
3.6	Oracle BPEL JDeveloper Designer	24
3.7	Da Interface Gráfica para Código Fonte BPEL	24
3.8	Apresentação do Oracle BPEL Console	27
3.9	Apresentação do Oracle Worklist	28
3.10	Modelo Relacional da Camada de Base de Dados do Oracle BPEL	30
4.1	Processo de Negócio Principal do Trouble Tickets em Oracle Workflow	33
4.2	Sub-Processo de Negócio do Trouble Tickets em Oracle Workflow	34
4.3	Modelo Relacional de Dados do Trouble Tickets na Versão Workflow	35
4.4	Formulários Originais do Módulo de Trouble Tickets	36
5.1	Exemplo de um Processo Definido em Oracle Workflow	41
5.2	Exemplo de um processo definido em BPEL	42
6.1	Arquitectura SIBPEL	51
6.2	Diagrama Geral de Pacotes dos Casos de Utilização do Novo Módulo de Trouble Tickets	57
6.3	Diagrama de Casos de Utilização do Pacote Reportar Problemas	58
6.4	Diagrama de Casos de Utilização do Pacote Gestão de Problemas	59
6.5	Diagrama de Casos de Utilização do Sistema de Administração BPEL	60
7.1	Decomposição Horizontal	62
7.2	Decomposição Vertical	66
7.3	Arquitectura Física	67
7.4	Diagrama de Estados dos Tickets e Tarefas	77
7.5	Sistemas de Pesquisa e Selecção de Problemas e Utentes	86
7.6	Sistemas de Especificação de Tarefas, Mensagens e Documentos	87
7.7	Apresentação do Formulário Detalhes Ticket	88
7.8	Apresentação do Pesquisa de Tickets e Tarefas	89
7.9	Apresentação da Consola BPEL do SIGARRA	91
7.10	Representação do Fluxo de Actividades de uma Instância BPEL na Consola	93

LISTA DE FIGURAS

7.11	Apresentação do Gestor de Tarefas BPEL do SIGARRA	94
8.1	Exemplo Simples de Utilização do Sistema	99
8.2	Exemplo Completo de Utilização e Funcionamento do Sistema	101
A.1	Diagrama de Actividades do Novo Processo de Negócio dos Trouble Tickets	110
A.2	Núcleo Central do Processo de Negócio em BPEL de um Ticket	111
A.3	Processo de Negócio em BPEL de uma Tarefa	112
A.4	O Novo Modelo Relacional do Módulo dos Trouble Tickets	113

Lista de Tabelas

5.1	Lista das Principais Actividades BPEL	47
7.1	Lista de Serviços do Web Service BPELProvider	74
7.2	Lista de Serviços do Web Service BPELConsole utilizados pela Consola BPEL	75
7.3	Lista de Serviços do Web Service BPELConsole utilizados pelo Gestor de Tarefas	76

LISTA DE TABELAS

Abreviaturas e Símbolos

PSI:	Instituição Projecto de Sistemas de Informação
FEUP:	Faculdade de Engenharia da Universidade do Porto
MIEIC:	Mestrado Integrado em Engenharia Informática e Computação
CICA	Centro de Informática Prof. Correia de Araújo
UP	Universidade do Porto
WS-BPEL:	Web Services Business Process Execution Language
Oracle BPM:	Oracle BPEL Process Manager
Oracle AS:	Oracle Application Server
OASIS:	Organization for the Advancement of Structured Information Standards
PL/SQL:	Procedural Language/Structured Query Language
XML:	eXtensible Markup Language
SOA:	Service-Oriented Architecture
WSDL:	Web Services Description Language
SOAP:	Simple Object Access Protocol
AJAX:	Asynchronous Javascript And XML
JSP:	Java Server Pages
HTML:	HyperText Markup Language
HTTP:	Hypertext Transfer Protocol
CSS:	Cascading Style Sheets
UML:	Unified Modeling Language
XPATH:	XML Path Language
API:	Application Programming Interface
Processo BPEL:	Processo de Negócio definido em BPEL
Instância BPEL:	Instância que processa o fluxo de actividades de um Processo BPEL
TTS:	Módulo dos Trouble Tickets
SI:	Sistema de Informação
Tarefa:	Funcionalidade do módulo dos Trouble Tickets
tarefa:	Funcionalidade da linguagem BPEL

ABREVIATURAS E SÍMBOLOS

Capítulo 1

Introdução

Este documento apresenta os detalhes da realização do projecto “Utilização do Oracle BPEL Process Manager”, desenvolvido pelo aluno Ivo André da Rocha Marinho, no âmbito do seu projecto final do Mestrado Integrado em Engenharia Informática para a Faculdade de Engenharia Informática e Computação.

Ao longo deste documento serão demonstradas as necessidades da elaboração deste projecto no meio empresarial em que foi desenvolvido, comparadas as vantagens da sua aplicação relativamente ao sistema precedente e terminando com a apresentação dos principais detalhes da sua especificação, implementação e avaliação final.

Neste primeiro capítulo constitui uma introdução ao projecto e uma apresentação da instituição onde foi realizado, incluindo ainda a estrutura de capítulos que compõem este documento.

1.1 Enquadramento

O desenvolvimento do projecto “Utilização do Oracle BPEL Process Manager” foi proposto para a realização de um estágio final, conforme o estipulado pelo actual plano de estudos para os alunos finalistas do Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto.

A sua realização decorreu durante o segundo semestre do ano lectivo 2007/2008, com início em Fevereiro de 2008 e fim em Julho de 2008, no Projecto de Sistemas de Informação (PSI) na Faculdade de Engenharia da Universidade do Porto (FEUP).

O autor deste projecto é o aluno Ivo André da Rocha Marinho, finalista do Mestrado Integrado em Engenharia Informática e Computação da Faculdade de Engenharia da Universidade do Porto, contando ainda com a orientação e supervisionamento do Prof. Gabriel Torcato David, por parte da FEUP, e do Eng. Manuel Machado por parte do PSI.

1.2 A Instituição

O presente projecto de estágio foi proposto e desenvolvido no Projecto de Sistemas de Informação (PSI) da Faculdade de Engenharia da Universidade do Porto. É constituída maioritariamente por colaboradores da FEUP pertencentes aos quadros do Centro de Informática Professor Correia de Araújo (CICA), cujas funções e responsabilidades distribuem-se pelas diversas áreas de interesse desta instituição.

O seu principal objectivo é o desenvolvimento do SIGARRA, que teve início em 1996. Esta aplicação consiste num completo sistema de gestão de instituições de ensino. Dados ainda os seus interesses na integração das mais recentes tecnologias, rapidamente o SIGARRA se destacou como um sistema fulcral para seu sucesso, sendo actualmente utilizado por diversas instituições de ensino em Portugal.

Tendo como principais objectivos, o desenvolvimento de produtos e serviços na área de Sistemas de Informação, dos seus sistemas destacam-se, para além do SIGARRA, a Gestão Integrada de Contra-Ordenações para a Câmara Municipal do Porto e o Sistema de Informação do Sindicato de Trabalhadores da Função Pública do Norte.

Como estagiário nesta instituição e de forma a cumprir a realização de todos os objectivos propostos, fui integrado numa equipa de desenvolvimento e investigação, com um total de dois colaboradores do PSI, estando particularmente alocado as responsabilidades do estudo da nova linguagem e tecnologia BPEL e de proceder a sua integração no módulo do SIGARRA.

1.3 Projecto

Este projecto tem como título “Utilização do Oracle BPEL Process Manager”, que consiste na utilização de uma nova tecnologia desenvolvida pela Oracle, para gestão e desenho de processos de negócio utilizando igualmente uma muito recente tecnologia de standards baseada em Web Services, denominada por Business Process Execution Language (BPEL).

Esta recente tecnologia BPEL adopta o WS-BPEL 2.0 OASIS Standard, definindo-se como uma nova linguagem para desenho e informatização de processos de negócio baseados na arquitectura orientada a serviços SOA, igualmente baseada nas mais recentes tecnologias tais como a Web Services Description Language WSDL, XML, Java e comunicações via SOAP.

Sendo esta uma tecnologia bastante recente e com grandes perspectivas de crescimento de importância e adopção, foi necessário realizar um estudo prévio das suas características, funcionalidades e formas de integração, para se proceder ao desenvolvimento do principal objectivo deste projecto que foi a passagem dos actuais processos de negócio

que utilizam a anterior ferramenta Workflow da Oracle, para esta nova tecnologia, adoptando todas as suas vantagens e fazendo do PSI uma das pioneiras em Portugal a integrar a tecnologia BPEL nas suas aplicações.

A consequente utilização da ferramenta Oracle BPEL Process Manager para a criação e gestão de todos os processos de negócio e de demais tecnologias e plataformas, permitiu integrar ferramentas de alto nível no principal sistema da instituição, e reduzir os custos e complexidades de desenho, integração e gestão de futuros processos de negócio no SIGARRA.

Sabendo à partida que a integração desta tecnologia seria bastante complexa, devido à utilização das mais recentes normas e tecnologias e à fraca e até por vezes inexistente divulgação de soluções, nomeadamente na passagem de processos Workflow para BPEL, este projecto foi um ponto de partida para a total transformação dos actuais processos de negócio. Deste modo, sendo o SIGARRA constituído por cerca de 22 módulos que integram a ferramenta Workflow da Oracle, este projecto assentou principalmente na transformação do módulo de “Trouble Tickets”, tanto a nível do seu processo de negócio, como a nível da própria aplicação, de modo a integrar as mais recentes tecnologias tais como a tecnologia AJAX, XML e comunicações via SOAP.

Toda a transformação deste módulo, requereu a definição e implementação da arquitectura SIBPEL de interligação entre o Sistema de Informação (SIGARRA) e a tecnologia BPEL, cujas comunicações foram estabelecidas através da disponibilização de informações dos processos e instâncias do BPEL em forma de Web Services em Java, recorrendo à API de métodos e funções Java para o BPEL, que comunicam com o Sistema de Informação, desenvolvido maioritariamente em Oracle PL/SQL, através de mensagens SOAP e XML.

Existindo mais duas tecnologias desenvolvidas pela Oracle, nomeadamente para gestão e controlo de processos BPEL, o Oracle BPEL Console e o Oracle Worklist, um grande factor de reconhecimento deste projecto foi igualmente a possibilidade de durante o período de estágio e utilizando já toda a arquitectura implementada, proceder a visualização e controlo no SIGARRA dos mesmos conteúdos e funcionalidades disponibilizadas por estas duas ferramentas, tendo sido inclusive incluídas novas características que melhor se adaptam e correspondem aos objectivos deste Sistema de Informação.

1.4 Motivação e Objectivos

O desenvolvimento deste projecto decorreu segundo os seus objectivos que foram distribuídos pelas seguintes etapas, que serão detalhadamente apresentadas ao longo deste documento:

1. **Estudo da tecnologia BPEL:** Aqui era pretendida a realização de um estudo sobre o novo standard de automatização de processos de negócio. O standard escolhido foi o WS-BPEL 2.0 OASIS Standard que define uma nova linguagem conhecida por Business Process Execution Language (BPEL) baseada em Web Services, XML, Java e SOAP. A futura integração desta linguagem no Sistema de Informação do PSI irá permitir a implementação de uma arquitectura distribuída e baseada nos conceitos da Arquitectura SOA.
2. **Transformação de processos em Oracle Workflow para BPEL:** Nesta fase era pretendida a realização de uma comparação entre os processos de negócio definidos em Oracle Workflow e a tecnologia BPEL, de modo a possibilitar a definição de possíveis mapeamentos que permitem a sua evolução para este novo standard.
3. **Especificação e implementação de uma arquitectura distribuída:** Esta fase representa o início da evolução do sistema de informação existente. Aqui deverá ser desenhada e construída uma nova arquitectura que possibilite a integração da tecnologia BPEL no Sistema de Informação.
4. **Levantamento de novos Requisitos do módulo de Trouble Tickets:** Aqui era pretendido o levantamento das actuais funcionalidades deste módulo e de novos casos de utilização junto dos seus principais utilizadores. Deste modo, isto deveria ainda levar a uma total reformulação das suas funcionalidades e interfaces Web.
5. **Evolução do Módulo de Trouble Tickets:** Esta fase pretende iniciar a evolução do primeiro módulo do SIGARRA de Workflow para BPEL. Esta evolução deverá ainda utilizar toda a arquitectura implementada e tirar o máximo partido das vantagens da tecnologia BPEL. Deverá igualmente servir de exemplo para as futuras evoluções dos restantes módulos.
6. **Criação de um sistema de administração BPEL:** Definido como um objectivo suplementar após a conclusão dos anteriores, pretende o desenvolvimento de um sistema de administração de toda a tecnologia BPEL.

1.5 Organização do Documento

Para além desta introdução, este documento é ainda composto por mais oito capítulos. No capítulo 2 é descrita com um maior detalhe a motivação e o plano de desenvolvimento do projecto. Quanto ao capítulo 3, são apresentados os conceitos e vantagens da Arquitectura SOA, explicando ainda a importância da tecnologia BPEL para a sua implementação. No capítulo 4, é efectuada uma análise do Sistema de Informação, antes da realização deste projecto e ainda na versão em Oracle Workflow. O capítulo 5 apresenta um conjunto de normas e etapas, definidas neste projecto, que devem ser seguidas para efectuar

Introdução

processo de transformação dos actuais processos de negócio em Oracle Workflow para BPEL. No capítulo 6, são especificadas todas as soluções estabelecidas para proceder à evolução do primeiro módulo do Sistema de Informação. Em relação ao capítulo 7, são apresentados os detalhes da implementação deste projecto. Por fim, são apresentados no capítulo 8, os testes efectuados ao sistema e a sua avaliação final, terminando com as respectivas conclusões e perspectivas de trabalho futuro no capítulo 9.

Introdução

Capítulo 2

Motivação do Projecto

As aplicações empresariais e sistemas de informação representam hoje em dia as bases fundamentais para o correcto processamento das operações de negócio. Havendo cada vez mais uma procura de automatização e informatização de processos de negócio, torna-se fundamental melhorar a sua eficiência e eficácia através das mais recentes tecnologias e serviços. É com base nestes princípios e fundamentos, que os objectivos principais das empresas passam por adaptar todas as suas aplicações à integração das mais recentes normas de definição de processos de negócio.

Tendo estes sido constantemente os objectivos principais do Projecto de Sistemas de Informação, surgiu então a necessidade de adaptar os seus actuais processos às mais recentes tecnologias de informação e automatização. É então deste modo que surgem os objectivos de implementação deste projecto e a definição do seu título como Utilização do Oracle BPEL Process Manager.

Como o próprio nome do projecto indica, este incidiu fundamentalmente sobre o estudo e análise de utilização e integração de uma recente aplicação empresarial, desenvolvida pela Oracle Software e denominada de Oracle BPEL Process Manager, de forma a tirar o máximo partido de todas as suas vantagens. Esta consiste numa poderosa ferramenta de configuração e administração de processos de negócio, definidos na linguagem Business Process Execution Language (BPEL).

A linguagem BPEL representa uma evolução na simplicidade e funcionalidades para o desenho dos mais complexos processos de negócio. Pode inclusive ser comparado às mais tradicionais linguagens de programação, possibilitando igualmente a definição dos seus processos de negócio em forma de algoritmos.

A principal vantagem desta nova linguagem é a sua relação com as mais recentes tecnologias de integração e adaptação de Arquitecturas Orientadas a Serviços às aplicações empresariais. Actualmente as vantagens de utilização deste tipo de arquitecturas são cada

vez mais reconhecidas como fundamentais para a integração entre diversos serviços aplicativos e empresariais, bem como na simplificação das complexidades em estabelecer ligações e trocas de informação entre diversos e diferentes sistemas e ferramentas utilizados especialmente nas grandes multi-empresas.

A tecnologia BPEL integra actualmente todas as especificações fundamentais para a realização destes objectivos, especialmente nas suas relações com a integração de Web Services e com todas as suas bases como o protocolo SOAP e as linguagens WSDL e XML. A linguagem XML é de facto a linguagem base da tecnologia BPEL, que apenas define as normas e regras de desenho das diversas actividades, ligações e configurações de um processo de negócio, sendo ferramentas como o Oracle BPEL Process Manager as responsáveis pelo seu processamento e gestão.

Devido ao interesse em usufruir de todas as vantagens que estas tecnologias proporcionam, surgiu então a especificação dos objectivos deste projecto como sendo o início da implementação da arquitectura do futuro na maior ferramenta Projecto de Sistemas de Informação, o SIGARRA.

Este módulo corresponde a um completo e eficiente sistema de gestão e disponibilização de informação para instituições de ensino, docentes, funcionários e alunos. Hoje em dia, todas as suas funcionalidades são utilizadas principalmente na gestão das diversas instituições que compõem a Universidade do Porto. Sendo a UP representada por diversos órgãos internos e externos, a necessidade de interligação entre ambos e a criação de um sistema distribuído tornaram-se cada vez questões fundamentais para o SIGARRA. Por outro lado, sendo este módulo implementado principalmente em PL/SQL e utilizando como servidor aplicativo o Oracle Enterprise Manager 10g, a futura evolução para a nova versão 11g é igualmente vista como uma prioridade.

Todos estes factos levaram a realização deste projecto. Por um lado, a especificação de integração da tecnologia BPEL e de ferramentas como o Oracle BPEL Process Manager e Oracle BPEL Console, permitem evoluir e adaptar os actuais processos de negócio definidos neste módulo em Oracle Workflow, tirando, por um lado, partido de todas as suas vantagens que permitem resolver a primeira questão de interligação entre diferentes sistemas e linguagens bem como solucionar a questão de que o Oracle Workflow deixou de ser suportado no Oracle Database 11g dando apenas lugar à tecnologia BPEL.

Deste modo, os principais objectivos definidos para este projecto consistiram principalmente, numa primeira fase, no estudo destas tecnologias e análises de soluções de integração com o módulo do SIGARRA. Foi então especificada a criação da futura arquitectura SIBPEL, de interligação do actual Sistema de Informação às funcionalidades da tecnologia BPEL e às respectivas ferramentas de gestão. Foi igualmente definida a utilização do Oracle BPEL Process Manager como servidor aplicativo de processos BPEL, cuja actual importância originou o título deste projecto.

Para uma segunda fase, foi estipulada a transformação de um dos actuais 22 módulos

que compõem o SIGARRA, de Oracle Workflow para BPEL. Este era de facto considerado o principal desafio deste projecto, visto não existir actualmente nenhuma norma ou regras divulgadas para este efeito. Deste modo, o estudo desta tecnologia na primeira fase veio ser fundamental na estipulação destas tão procuradas mas inexistentes normas de transformação e evolução, permitindo definir algumas soluções que serão apresentadas essencialmente no capítulo 5 deste documento.

O módulo seleccionado para o início desta integração foi o módulo de Trouble Tickets. Este consiste num sistema que permite reportar problemas ou falhas detectadas na instituição de ensino ou no próprio sistema, e ainda propor melhorias ou novas implementações nas funcionalidades disponibilizadas. Estando este módulo no final do projecto totalmente interligado com a nova arquitectura orientada a serviços SIBPEL, deverá servir de exemplo para a transformação e evolução de todos restantes módulos. Ainda nesta fase era pretendido o levantamento dos actuais casos de uso do Trouble Tickets, de forma a posteriormente identificar novos casos de utilização e funcionalidades que poderiam agora usufruir das vantagens desta nova arquitectura e tecnologias. As novas especificações originaram igualmente uma total remodelação a nível de interfaces, cumprindo todos os requisitos identificados pelos seus utilizadores, nomeadamente na necessidade de um melhoramento da sua rapidez, e integrando para isto as mais recentes tecnologias Web como a tecnologia AJAX.

Na fase final da realização deste projecto, visto que todos estes objectivos estavam devidamente cumpridos atempadamente, foram ainda especificados novos objectivos extra de desenvolvimento de aplicações de administração de processos BPEL no próprio Sistema de Informação do SIGARRA. Estes sistemas de administração consistem na disponibilização das principais funcionalidades existentes nas ferramentas Oracle BPEL Console e Oracle Worklist, mas de forma totalmente desenhada para utilização pelos administradores do SIGARRA e integrando inclusive algumas características inovadoras para este efeito.

2.1 Plano e Calendarização

O plano de desenvolvimento deste projecto e a respectiva calendarização, são apresentadas na figura 2.1, conforme os objectivos e etapas definidas e os prazos estipulados para a realização deste projecto final.

2.2 Tecnologias e Ferramentas Utilizadas

Todo o processo de especificação e implementação deste projecto foi pensado de modo a integrar a utilização das mais recentes tecnologias existentes actualmente. A tecnologia que possui um maior destaque neste projecto é a Business Process Execution Language

Motivação do Projecto



Figura 2.1: Diagrama de Gantt do Plano de Realização do Projecto

(BPEL), que obteve um total interesse por parte da Oracle na criação do seu novo servidor, o Oracle BPEL Process Manager, cuja utilização é aliás o título deste projecto.

A definição da tecnologia BPEL está ainda intimamente ligada à utilização de Web Services. A importância da inclusão de Web Services nas aplicações empresariais tem vindo a crescer constantemente. Desta forma, estes dois fortes aliados permitem implementar uma arquitectura orientada a serviços e tornar as aplicações em fortes sistemas distribuídos.

Para isto é ainda necessário referir a importância de tecnologias como o SOAP para a comunicação e troca de dados, o XML para a composição de mensagens SOAP e estruturas de dados e o XPATH para descodificação e leitura dos dados transmitidos pelas mensagens SOAP.

Outras tecnologias essenciais para a implementação deste projecto foram o PL/SQL para implementação de todo o motor de lógica do Sistema de Informação e das comunicações com a tecnologia BPEL, o Java na implementação dos Web Services e o Ajax e Javascript que tiveram um papel fundamental para o sucesso da reformulação das interfaces Web desenvolvidas.

Em termos de ferramentas necessárias para a utilização destas tecnologias, para além da já referida Oracle BPEL Process Manager, foram ainda utilizados o Oracle JDeveloper 10.1.3.3 para definição de processos BPEL e implementação de Web Services em Java, o Oracle SOA Suite 10g que inclui o Oracle BPM e todo o suporte para a gestão e disponibilização de Web Services, o Oracle Application Server 10g onde se inclui todo o repositório de dados utilizado pelo Sistema de Informação, o SQL Navigator para implementação do motor da lógica de negócio, o Oracle BPEL Console e o Oracle Worklist para gestão da tecnologia BPEL e o Oracle Designer para definição e criação automática de modelos relacionais de dados.

Capítulo 3

Arquitectura Orientada a Serviços

Actualmente, com a evolução das tecnologias e com o consecutivo aumento da diversidade de aplicações e áreas de negócio para as empresas, tornou-se fundamental que os sistemas de informação desenvolvidos suportem alterações rápidas e eficazes, de modo a se adaptarem facilmente as constantes alterações do mercado de negócios. Estas alterações podem ser analisadas tanto a nível dos processos de negócio, mas também nas adaptações às mais recentes tecnologias e plataformas.

Deste modo, a maioria das empresas contém um elevado número e conjunto de diferentes sistemas, aplicações, tecnologias e arquitecturas. A integração destes conjuntos num único e centralizado sistema de informação é crucial no acesso e controlo dos dados e verificações de integridade.

De forma a atingir estes objectivos, a especificação da Arquitectura Orientada a Serviços (SOA - Service Oriented Architecture) tornou-se na resposta mais eficaz existente actualmente a nível de integração, desenvolvimento e manutenção de complexos sistemas de informação. A especificação desta arquitectura resulta da evolução de diversas arquitecturas distribuídas aliados aos mais recentes métodos de integração.

A implementação de uma arquitectura SOA está intimamente ligada à integração de serviços externos, os Web Services. Estes serviços são um conjunto de definições que operam sobre trocas de mensagens entre clientes e servidores, que contém dados definidos em XML e que seguem os padrões de comunicação do SOAP. Normalmente estes serviços são definidos por um Web Service Description Language (WSDL) e disponibilizados para acesso através de protocolos de registos de serviços. Dadas as suas vantagens, os Web Services possuem potencial para transformar a Internet numa verdadeira plataforma distribuída.

Os principais conceitos da arquitectura SOA são focados na reutilização de aplicações existentes, na interoperabilidade e integração de aplicações e na composição de pro-

cessos de negócio através de serviços e funcionalidades disponibilizados por diferentes aplicações e sistemas. Um dos objectivos principais desta arquitectura é então garantir que futuras alterações e manutenções aos sistemas de informação sejam efectuadas de uma forma fácil e rápida. Para isto, esta arquitectura define todo um conjunto de conceitos, arquitecturas e frameworks, reduzindo a complexidade e apostando na integração e reutilização, de modo a garantir um menor custo de desenvolvimento, integração e de manutenção possível. Estes conceitos não se apoiam apenas numa única tecnologia, mas sim num conjunto que poderá ser agrupado do seguinte modo:

- **Serviços:** Os serviços da arquitectura SOA representam funcionalidades que apresentam ou devolvem valores de negócio, de uma forma autónoma e transparente a nível de implementação.
- **Interfaces:** As interfaces estão relacionadas com os serviços, na medida em que definem o seu conjunto de operações públicas, através da qual os seus interessados irão aceder. Representam um meio de comunicação entre o prestador de serviços e os consumidores.
- **Mensagens:** As comunicações e operações são definidas com um conjunto de mensagens que especificam os dados que devem ser trocados, de forma independente da linguagem e da plataforma em que estes se incluem.
- **Sincronização:** Os consumidores acedem aos serviços através de um barramento de serviços, que tanto poderá ser um protocolo como o SOAP, ou um Enterprise Service Bus (ESB) que actua como uma camada intermediária de comunicação. Estas comunicações podem ser síncronas ou assíncronas.
- **Minimização de Dependências:** Utilizando apenas as dependências mínimas necessárias, garante-se que apenas irá existir um pequeno número de alterações necessárias quando um serviço for modificado.
- **Registos:** As listas de registos facilitam a procura de serviços disponíveis. Funcionam como uma lista de directórios, onde são publicados os diversos serviços que serão posteriormente consultados pelos consumidores. O Universal Description, Discovery and Integration (UDDI) é um exemplo de um registo de serviços.
- **Qualidade dos Serviços:** Representam atributos que os serviços devem incluir, tais como segurança, gestão e fiabilidade nas transacções, correlações e definições.
- **Composição de Serviços em Processos de Negócio:** Esta transformação representa o conceito mais importante da arquitectura SOA. É nesta fase que se introduz a linguagem BPEL como o único e poderoso motor existente actualmente, para

definição e composição de processos de negócio baseados em serviços e nos conceitos desta arquitectura. Desta forma obtemos um suporte total para realizar rapidamente modificações e manutenções destes processos, de uma forma muito flexível. Atingindo este nível de transformação, poderemos então usufruir de todas as vantagens que esta arquitectura providencia.

Estes conceitos e as respectivas estruturas e relações, são apresentados na figura 3.1 como uma visão geral da arquitectura SOA.

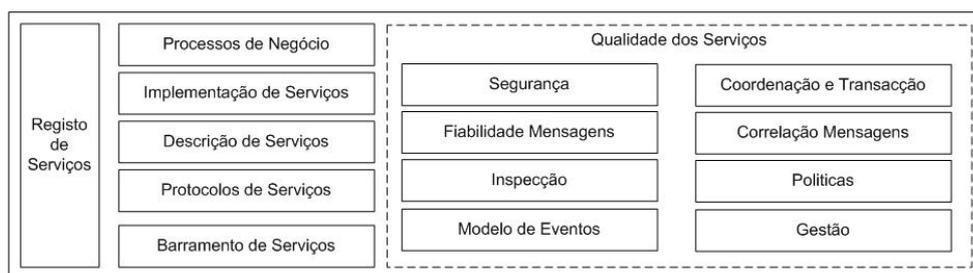


Figura 3.1: Visão geral da Arquitectura SOA

Se na figura 3.1 introduzirmos as tecnologias principais e todos os standards definidos para esta arquitectura e que foram utilizados na elaboração deste projecto, conseguimos então obter uma visão clara na figura 3.2 de como o BPEL se relaciona com os restantes níveis e de como se integra no conceito mais importante desta arquitectura.

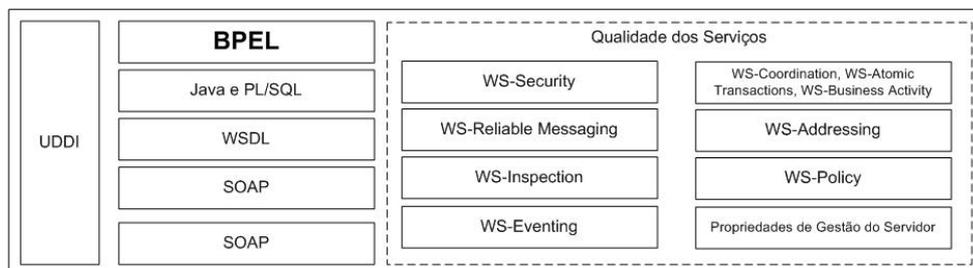


Figura 3.2: Tecnologias do Projecto integradas na Arquitectura SOA

3.1 Tecnologia BPEL

Desenvolvendo uma infra-estrutura baseada em standards e seguindo os princípios da arquitectura SOA, permite que as organizações adaptem as suas aplicações aos novos ambientes e requisitos de uma forma flexível. Permite igualmente que estas automatizem os seus processos e reajam mais rapidamente e facilmente às alterações necessárias, o que leva a uma diminuição dos seus custos e complexidades. Actualmente o BPEL é o

standard mais completo e vantajoso para automatizar processos com base nos conceitos da arquitectura SOA. Em conjunto, o BPEL e a arquitectura SOA oferecem uma plataforma eficaz na redução de custos de manutenção e oferecendo rápidas adaptações às alterações dos processos de negócio.

A tecnologia BPEL foi introduzida para completar a realização dos conceitos da arquitectura SOA. Esta especifica que os serviços devem ser agregados em novos e maiores serviços, e depois integrados nos respectivos processos de negócio. Desta forma, poder-se-á definir um processo de negócio como um conjunto de actividades através dos quais os serviços são invocados. No entanto, o próprio processo de negócio deverá ser visto para o exterior igualmente como um novo serviço e igual a todos os outros que são por ele invocados. Uma definição generalizada para os processos de negócio poderá ser a seguinte:

Um processo de negócio é um conjunto de invocações coordenadas de serviços e de actividades relacionadas, que produzem um resultado final, tanto a partir de uma simples organização ou de várias.

Para ilustrar a composição de um possível processo de negócio, apresenta-se o seguinte exemplo, onde um cliente efectua um pedido a uma Agência de Viagens, para obter o melhor preço para uma dada viagem. Esta, por sua vez consulta os serviços disponibilizados por várias companhias aéreas em forma de Web Services, sobre as suas disponibilidades e preços. Por fim, calcula o melhor preço e devolve a proposta da agência ao cliente. A composição de actividades e serviços deste possível processo de negócio é apresentada na figura 3.3.

Este constitui um simples exemplo real da integração da tecnologia BPEL na definição de processos de negócio, que podem agrupar e reutilizar serviços externos à empresa, de uma forma fácil e transparente do ponto de vista do processo de negócio. Ou seja, o processo de negócio não tem acesso à forma como os serviços retornam os valores, quer seja por acessos ao sistema, a ficheiros, à base de dados, ou de qualquer outra forma, o que interessa é apenas os valores que estes devolvem.

O desenvolvimento destes processos de negócio em BPEL implica que sejam eles próprios também divulgados e acedidos como serviços, tirando o máximo partido dos conceitos e objectivos do SOA na reutilização e diminuição da complexidade dos sistemas. Foi então desta forma que a especialização da linguagem BPEL se tornou, actualmente, no standard dominante da arquitectura SOA para a automação de processos de negócio utilizando Web Services.

O BPEL define processos de negócio como agrupamentos de serviços, operações e outros processos e expõe as suas funcionalidades como serviços.

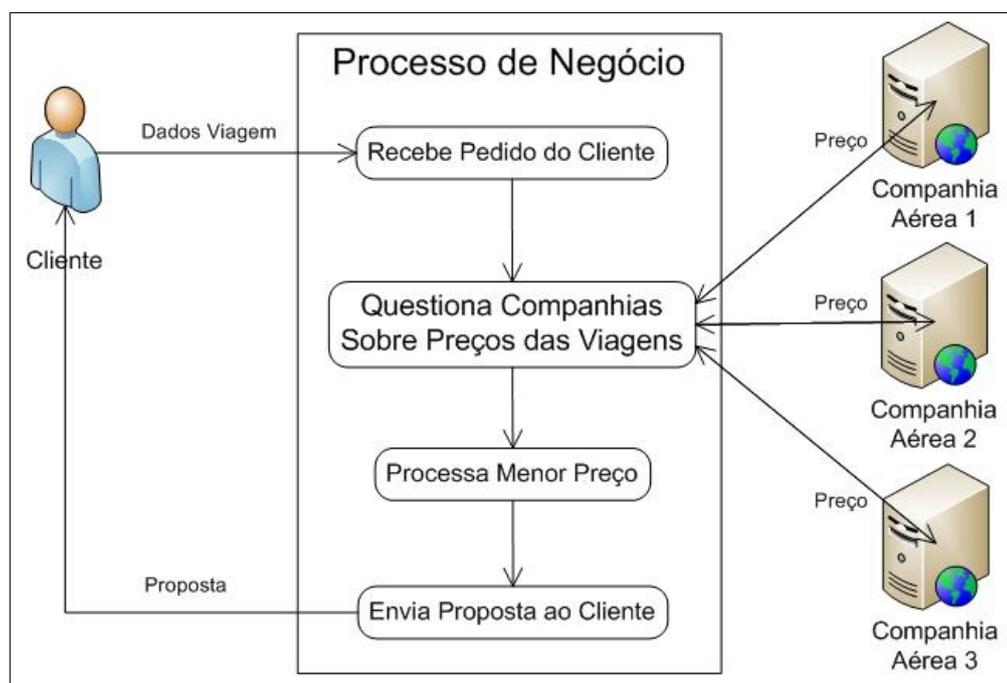


Figura 3.3: Exemplo da Agregação de Serviços num Processo de Negócio

A primeira versão BPEL4WS 1.0 foi desenvolvida em conjunto pela IBM, BEA e a Microsoft em Agosto de 2002, a partir de duas principais linguagens de workflow e definição de processos: o WSFL (Web Services Flow Language) e XLANG. Após várias modificações e melhoramentos, foi estabelecida a adopção da versão 1.1 que foi depois submetida como um standard para a OASIS (Organization for the Advancement of Structured Information Standards) onde foi formado o WS-BPEL. Através de consecutivos melhoramentos e com a integração de mais standards como o WS-HumanTask e o BPEL4People em 2007, surgiu entretanto a versão 2.0, utilizada actualmente como o Web Services Business Process Execution Language OASIS Standard WS-BPEL 2.0.

Esta norma corresponde à linguagem dominante actualmente e que tira o máximo partido e vantagens de todas as suas outras linguagens. De entre outras linguagens precedentes, incluem-se por exemplo o BPML, o WSCL, BPSS, WSCI e WS-CDL cujas funcionalidades foram integradas ao longo do tempo numa única tecnologia, o BPEL.

A sua utilização dentro das empresas auxilia igualmente a integração de sistemas isolados. Entre empresas, o BPEL facilita a integração com vários parceiros de negócio. A adopção deste standard provoca uma reestruturação, evolução e selecção dos processos mais apropriados, levando a consequente optimização de toda a organização. O BPEL tornou-se assim na chave de sucesso em ambientes tecnológicos que já utilizam Web Services ou que pretendem expor as suas funcionalidades como serviços.

3.1.1 Principais Características e Funcionalidades

A linguagem BPEL é totalmente definida em XML. Desta forma é então possível descrever e especificar processos de negócio simples e complexos. A sua composição é similar às tradicionais linguagens de programação, incluindo igualmente variáveis, estruturas, ciclos, atribuições, verificações, etc., que possibilita definir processos de forma algorítmica.

Os aspectos fundamentais desta tecnologia são de facto a simplicidade na invocação de operações definidas por Web Services e a possibilidade de definir a execução de actividades e operações tanto em sequência como em paralelo. Outras características incluem-se a gestão de excepções e erros, actividades de espera por acções, suporte para a execução de longos processos e actividades de controlo e interacção com utilizadores. Desta forma, apresentam-se de seguida as principais funcionalidades oferecidas por esta tecnologia:

- Descrição da lógica de processos de negócio através da composição de serviços.
- Definição de longos processo de negócio dadas as fortes propriedades da sua linguagem XML nativa.
- Definição das actividades e operações de forma algorítmica.
- Composição de grandes processos a partir de outros pequenos processos e serviços.
- Manipulação de operações e invocações síncronas e assíncronas, gerindo respostas que sejam recebidas após um longo período de tempo.
- Invocação e composição de serviços e actividades tanto em paralelo como em sequência.
- Manipulação de erros, compensações e eventos.
- Envio, recepção e reencaminhamento de mensagens para os processos e actividades respectivas.
- Agendar actividades e tarefas e definir a suas ordens e prioridades de execução.
- Estruturação de actividades dos processos de negócio em agrupamentos.
- Interacção dos processos de negócio com os utilizadores, através de actividades de notificação e decisão.

3.1.2 Exemplo de um Processo de Negócio

A especificação de um processo de negócio BPEL na sua linguagem nativa XML é caracterizada por quatro grupos de definições: Processo, Parceiros, Variáveis e a Lógica de Negócio. O esquema seguinte apresenta o código fonte XML de um simples processo

de negócio síncrono em BPEL, implementado num ficheiro com a extensão .bpel, que apenas recebe um parâmetro de entrada de um cliente e devolve uma resposta.

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<process name="BPELProcess1"
    targetNamespace="http://xmlns.oracle.com/BPELProcess1"
    xmlns="http://schemas.xmlsoap.org/ws/2003/03/business
        -process/"
    xmlns:client="http://xmlns.oracle.com/BPELProcess1"
    xmlns:ora="http://schemas.oracle.com/xpath/extension"
    xmlns:ldap="http://schemas.oracle.com/xpath/extension/ldap"
    xmlns:bpelx="http://schemas.oracle.com/bpel/extension"
    xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/
        03/business-process/">
<!-- PARTNERLINKS -->
    <partnerLinks>
        <partnerLink name="client"
            partnerLinkType="client:BPELProcess1"
            myRole="BPELProcess1Provider"/>
    </partnerLinks>
<!-- VARIABLES -->
    <variables>
        <variable name="inputVariable"
            messageType="client:BPELProcess1RequestMessage"/>
        <variable name="outputVariable"
            messageType="client:BPELProcess1ResponseMessage"/>
    </variables>
<!-- ORCHESTRATION LOGIC -->
    <sequence name="main">
        <receive name="receiveInput" partnerLink="client"
            portType="client:BPELProcess1" operation="process"
            variable="inputVariable" createInstance="yes"/>
        <reply name="replyOutput" partnerLink="client"
            portType="client:BPELProcess1" operation="process"
            variable="outputVariable"/>
    </sequence>
</process>
```

As primeiras definições são relativas ao processo e as extensões que este irá utilizar. As definições dos parceiros (Partner Links) inicializam as definições tanto dos clientes,

como de todos os serviços (Web Services) invocados pelo processo.

De seguida, definem-se as variáveis e os respectivos tipos, que podem estar associados a tipos de mensagens que por sua vez podem igualmente representar conjuntos de variáveis ou variáveis que não são um tipo pré definido pelo BPEL. Neste exemplo definem-se as variáveis de entrada e saída do processo, que constituem comunicações que obedecem as regras de SOAP, sendo a definição do seu tipo definida então como uma mensagem.

Por fim apresentam-se as definições das várias sequências de processamento que compõem a lógica de negócio do processo. Neste exemplo incluem-se duas actividades na lógica de negócio, um Receive que recebe os parâmetros do cliente e atribui os seus valores à variável inputVariable, e um Reply que retorna ao cliente o resultado contido em outputVariable.

Todos os projectos BPEL ficam ainda associados ao seu WSDL, implementado num ficheiro com a extensão .wsdl, onde são guardadas todas as informações sobre os serviços e parceiros externos e os tipos de mensagens trocados entre estes e o processo de negócio. Para o exemplo apresentado, uma possível definição do seu WSDL poderá ser a apresentada no seguinte esquema.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="BPELProcess1"
  targetNamespace="http://xmlns.oracle.com/BPELProcess1"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:client="http://xmlns.oracle.com/BPELProcess1"
  xmlns:plnk="http://schemas.xmlsoap.org/
    ws/2003/05/partner-link/">
<!-- TYPE DEFINITION-->
  <types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://xmlns.oracle.com/BPELProcess1"
        schemaLocation="BPELProcess1.xsd" />
    </schema>
  </types>
<!--MESSAGE TYPE DEFINITION-->
  <message name="BPELProcess1RequestMessage">
    <part name="payload"
      element="client:BPELProcess1ProcessRequest"/>
  </message>
  <message name="BPELProcess1ResponseMessage">
    <part name="payload"
```

```

        element="client:BPPELProcess1ProcessResponse"/>
    </message>
<!--PORT TYPE DEFINITION-->
    <portType name="BPPELProcess1">
        <operation name="process">
            <input message="client:BPPELProcess1RequestMessage"/>
            <output message="client:BPPELProcess1ResponseMessage"/>
        </operation>
    </portType>
<!--PARTNER LINK TYPE DEFINITION-->
    <plnk:partnerLinkType name="BPPELProcess1">
        <plnk:role name="BPPELProcess1Provider">
            <plnk:portType name="client:BPPELProcess1"/>
        </plnk:role>
    </plnk:partnerLinkType>
</definitions>

```

O WSDL de um processo BPEL é composto por quatro grupos principais. No primeiro são especificados os ficheiros com a extensão .xsd, que contêm as definições dos esquemas para os tipos de parâmetros relacionados com o processo. De seguida são definidos os tipos de mensagens usados nas comunicações com os serviços e que podem ser utilizados e manipulados pelo processo. Por fim são especificadas as propriedades das ligações e o tipo de parceiros externos incluídos no processo.

A nível de desenvolvimento gráfico de processos BPEL, existem actualmente algumas ferramentas que disponibilizam estas funcionalidades, onde podemos simplificar todo o processo de definição dos processos através de uma intuitiva interface gráfica, de modo a obtermos uma visão geral das sequências de actividades que o nosso processo implementa. Assim, para o exemplo da Agência de Viagens apresentado na Figura 3.3, é apresentada na figura 3.4 a sua implementação num processo BPEL, utilizando o Oracle JDeveloper BPEL Designer.

3.1.3 Aplicações e Servidores Disponíveis

Os servidores de BPEL consistem em ambientes de processamento onde os processos são registados e executados. Actualmente, devido a sua forte ligação com Web Services, estes servidores estão desenvolvidos principalmente em plataformas que suportam Web Services tais como o Java Enterprise Edition e o Microsoft .NET. Deste modo, apresentam-se de seguida uma lista de alguns servidores de BPEL existentes actualmente:

- Oracle BPEL Process Manager.

Arquitectura Orientada a Serviços



Figura 3.4: Exemplo da Implementação Real de um Processo BPEL

- Microsoft BizTalk.
- IBM WebSphere Business Integration Server Foundation.
- IBM alphaWorks BPWS4J.
- BEA WebLogic e AquaLogic.
- Sun SeeBeyond eInsight Business Process Manager.
- Parasoft BPEL Maestro.

Existem ainda algumas versões em software livre, tais como:

- JBoss Business Process Manager.
- ActiveBPEL Engine.
- FiveSight Process Execution Engine PXE.
- Bexee BPEL Execution Engine.
- Apache Agila ou Twister.

Em termos de aplicações gráficas para o desenvolvimento de processos BPEL, existem igualmente algumas ferramentas disponíveis, embora algumas delas venham apenas incluídas conjuntamente com os servidores.

- Oracle JDeveloper BPEL Designer incluído no Oracle BPEL Process Manager.
- Oracle BPEL Designer for Eclipse.
- IBM WebSphere Studio Application Developer Integration Edition.
- iGrafx BPEL.
- Itp Process Modeler for Microsoft Visio.
- Eclipse BPEL Designer.

Para a realização deste projecto foram utilizadas as ferramentas disponibilizadas pela Oracle Software, nomeadamente o Oracle BPEL Process Manager, cujas vantagens sobre todas as restantes aplicações serão apresentadas com um maior detalhe no tópico seguinte.

3.2 Oracle BPEL Process Manager

O Oracle BPEL Process Manager foi, de todos os servidores de BPEL disponíveis, o escolhido para a realização deste projecto. As razões da sua escolha prendem-se com o facto de este ser, actualmente, o servidor aplicacional para BPEL mais completo e eficaz do mercado. Foi desenvolvido em Java, integrando-se no Oracle Application Server com o Oracle Containers for Java (OC4J). A Oracle disponibilizou igualmente diferentes versões que permitem integrar com outras ferramentas livres tais como JBoss e pode, por exemplo, ser utilizado igualmente com os servidores aplicacionais da IBM e da SUN.

As ferramentas disponibilizadas por esta aplicação, utilizam todos os standards definidos para arquitecturas orientadas a serviços, permitindo facilmente desenhar, processar e administrar processos de negócio orientados a serviços de uma forma muito rápida e intuitiva, tanto a nível automático, como a nível de interacções humanas. De todos os seus standards suportados, incluem-se o BPEL, XML, XSLT, XPATH, JMS, JCA e Web Services, tornando-o uma solução completa e ideal para atingir o máximo de portabilidade, integração e reutilização de sistemas e serviços, permitindo facilmente integrar processos de negócio em diferentes plataformas. Para além destas características, todas estas funcionalidades assentam sobre a plataforma Oracle Application Server, cujas vantagens são bastante conhecidas e usadas pelas maiores empresas mundiais.

Actualmente as principais funcionalidades e vantagens da utilização do Oracle BPEL Process Manager são as seguintes:

- Suporte de todos os standards que possibilitam facilmente a integração de arquitecturas orientadas a serviços nos diversos ambientes e sistemas empresariais, bem como a total interoperabilidade e portabilidade das suas aplicações e processos.
- Utilização de um sistema de Dehydration para melhorar o desempenho de longos processos e a correlação de mensagens assíncronas.
- Capacidades de gerir, monitorizar e administrar erros e excepções dos processos de negócio, tanto em desenvolvimento como em execução.
- Possibilidade de facilmente transferir o processamento de um processo entre servidores. Por exemplo, um processo pode ser criado num servidor, realocado para outro aquando da ocorrência de uma excepção e ser concluído por um terceiro servidor, caso a sua carga de processamento for excessiva.
- Disponibilização de ferramentas de gestão e administração de processos de negócio em tempo real, incluindo, por exemplo, a visualização de estatísticas, detalhes dos fluxos de actividades tanto a nível gráfico como em XML, histórico de acções, etc.
- Importante suporte para um controlo de versões dos vários processos, permitindo que estes possam evoluir ao longo do tempo e que as diversas instâncias, ainda a decorrer no sistema, possam terminar o seu processamento normalmente.
- Possibilidade de integração com diversos servidores aplicativos incluindo o Oracle Application Server, WebSphere, WebLogic e JBoss. Em termos de base de dados, poder-se-á utilizar Oracle, SQL Server, DB2, etc.
- Disponibilização de uma completa API em Java, com diversas funcionalidades que permitem administrar e gerir os vários processos BPEL do servidor.
- Inclusão de diversas ferramentas e aplicações que simplificam o desenvolvimento de processos de negócio em BPEL e o consequente registo no servidor, tais como o Oracle BPEL Console, o BPEL Designer e o Worklist Application.

3.2.1 Arquitectura do Oracle BPEL Process Manager

A figura 3.5 apresenta a arquitectura base do Oracle BPEL Process Manager, onde é possível identificar os seus principais componentes e serviços integrados.

Esta visão geral da arquitectura do Oracle BPEL Process Manager apresenta os seus componentes agrupados em quatro camadas principais: Desenho de Processos BPEL, Servidor, Consola de Gestão e a Base de Dados. Nos pontos seguintes, serão apresentadas com um maior detalhe estas quatro camadas, referindo igualmente as ferramentas principais disponibilizadas no Oracle BPEL Process Manager.

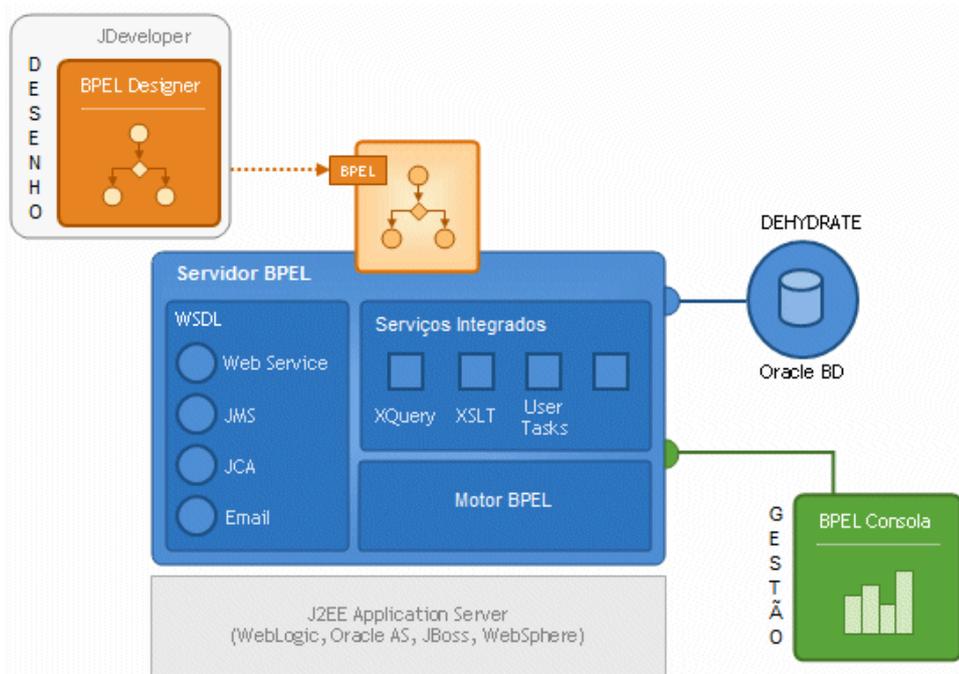


Figura 3.5: Arquitectura do Oracle BPEL Process Manager

3.2.1.1 Desenho de Processos BPEL

Com o Oracle BPEL é incluído o Oracle BPEL Process Designer, definido como um Plug-in para o Oracle JDeveloper. Este Designer apresenta uma intuitiva interface que permite facilmente a definição e desenho de processos de negócio em BPEL, em vez da escrita directa do seu código. No entanto, é igualmente possível editar directamente o código fonte destes processos e utilizar simples assistentes de criação de actividades, expressões e ligações a serviços externos, tais como a definição de tarefas para utilizadores, ligações a Web Services, Bases de Dados, etc. Estes são construídos totalmente em cima do formato XML nativo de BPEL, o que permite utilizar esta ferramenta com qualquer servidor aplicacional desta tecnologia. A figura 3.6 apresenta o aspecto do ecrã principal do Oracle BPEL Process Designer, integrado no Oracle JDeveloper 10.1.3.3.0, aquando da criação de um novo projecto BPEL.

Nesta figura é possível visualizar, ao centro, a definição do processo de negócio BPEL mais simples, que apenas recebe e envia parâmetros a um cliente, quando é invocado. Na barra de actividades, apresentada à direita, é possível identificar todas as actividades que podem compor a definição de um processo de negócio. Nas opções apresentadas em baixo, é possível alternar entre a apresentação gráfica do processo, o seu código fonte e o histórico de alterações efectuadas.

Para iniciar a definição do processo, é necessário arrastar as actividades para o desenho do processo. A criação de algumas actividades, são auxiliadas por assistentes mais

Arquitectura Orientada a Serviços

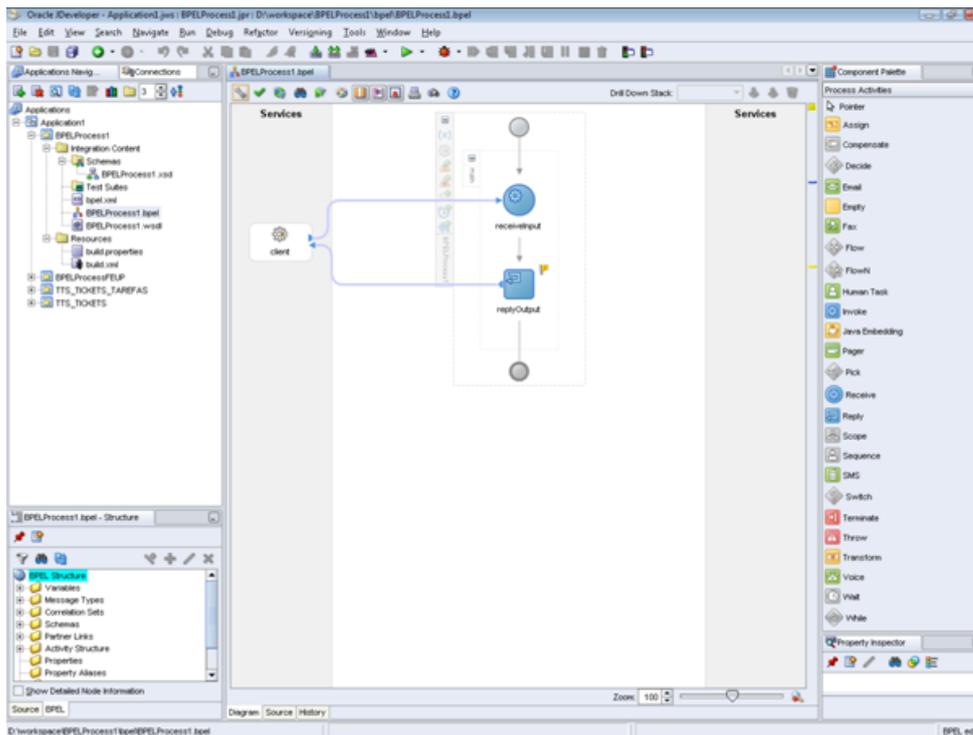


Figura 3.6: Oracle BPEL JDeveloper Designer

intuitivos que auxiliam a definição das suas propriedades principais como, por exemplo, a definição das ligações aos serviços externos.

Após a definição das suas propriedades e características, as actividades são incluídas no desenho principal do processo e automaticamente definidas no seu código fonte em XML, que será posteriormente interpretado e processado pelo servidor de BPEL.

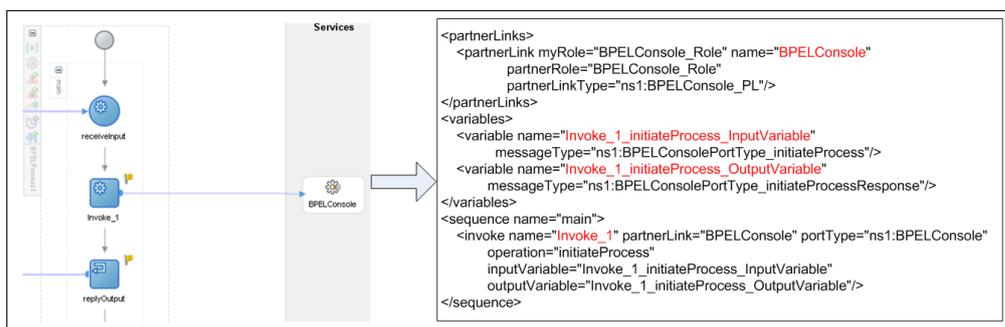


Figura 3.7: Da Interface Gráfica para Código Fonte BPEL

Para além destas vantagens referenciadas, incluem-se ainda aspectos importantes como a possibilidade de realizar testes aos processos, o registo automático com um simples clique, de um processo em desenvolvimento no servidor de BPEL e as facilidades de alteração e manutenção dos processos de negócio atingindo, deste modo, os principais

objectivos da arquitectura orientada a serviços que são a redução dos custos de desenvolvimento, integração e manutenção dos sistemas de informação. De facto, sendo uma das principais vantagens da concepção da linguagem BPEL a remoção da complexidade no desenvolvimento de sistemas, aplicações como o Oracle Designer tornam-se, hoje em dia, fundamentais para atingir totalmente estes conceitos.

3.2.1.2 Servidor BPEL

O servidor de BPEL é essencialmente composto por um núcleo de processamento BPEL, por adaptadores WSDL e por serviços de integração.

O seu núcleo central providencia a mais robusta implementação de BPEL existente actualmente. É neste núcleo que se concentra o registo e processamento dos seus processos de negócio. Este foi desenvolvido usando o Oracle Containers for Java EE (OC4J) como um servidor aplicacional J2EE, com suporte para correr em conjunto com outros servidores aplicativos tais como WebLogic, Oracle AS, JBoss e WebSphere. Sendo todo este núcleo desenvolvido em Java, uma grande vantagem na utilização do Oracle BPEL Process Manager é a disponibilização de uma API em Java, que foi utilizada na implementação das principais ferramentas de administração e configuração de processos BPEL, tais como o BPEL Console e o Worklist Application. A total disponibilização desta API foi fundamental na implementação deste projecto, visto que desta forma foi possível aceder as funcionalidades do gestor de processos BPEL através do SIGARRA.

De entre as suas principais particularidades, destacam-se ainda os adaptadores e os serviços de integração tais como XQuery, XSL e XPath, que possibilitam ligações e transformações entre diversos protocolos e plataformas. Alguns exemplos de adaptadores incluídos destacam-se os acessos a servidores de FTP e ficheiros, bases de dados, tabelas e registos.

De igual forma, as interacções humanas tais como a gestão de tarefas, de notificações e de identidade são também fornecidas como serviços BPEL integrados, permitindo a total interacção de utilizadores com fluxos de processos BPEL. Para isto, incluem-se ainda serviços de notificações, tanto a nível de tarefas e decisões humanas, como a nível de avisos e mensagens, como por exemplo o envio de emails, SMS, mensagens de voz, fax, etc. Todos estes factores estão ainda aliados a um sistema de identificação (Identity Service), que permite registar e autenticar o acesso dos utilizadores a estas funcionalidades.

Aliado a estes factores, a integração de uma poderosa Framework baseada em Web Services WSDL, que utilizando diversos protocolos e formatos de comunicação tais como SOAP, RMI (Remote Method Invocation), JMS (Java Message Service), Email, JCA (Java Connector Architecture), Sockets, HTTP GET e POST, fornece uma total conectividade entre diversos sistemas e plataformas.

3.2.1.3 Consola BPEL

Com o Oracle BPEL é igualmente disponibilizado o Oracle BPM Console. Esta consola fornece, aos administradores do sistema BPEL, uma intuitiva interface Web para gestão, administração e consulta dos processos BPEL registados no servidor. De todas as suas funcionalidades destacam-se:

- Registrar e Apagar processos de negócio em BPEL no servidor; Iniciar novas instâncias para um determinado processo BPEL;
- Visualização de instâncias de processos BPEL activas e terminadas;
- Visualizar graficamente todo o fluxo de actividades processado até um dado momento por uma instância BPEL;
- Aceder ao Audit Trail de uma instância, que apresenta, em forma de XML, todas as suas informações e detalhes de processamento das suas actividades até um dado momento;
- Aceder ao histórico de informações que são automaticamente mantidas no repositório;
- Cancelar e Apagar uma determinada instância;
- Visualizar conteúdos dos sensores definidos nos processos BPEL, para uma dada instância.

A figura 3.8 ilustra um exemplo de utilização da consola na visualização gráfica do fluxo corrente de actividades de um dado processo.

Com a aceitação do standard BPEL4People como uma nova extensão para o BPEL, de forma a suportar uma nova gama de cenários que incluem interacções dos utilizadores com os processos de negócio, surgiu todo um conjunto de actividades que incluem tarefas para utilizadores e notificações. Deste modo, para uma total integração do mundo BPEL nas aplicações empresariais, o Oracle BPEL é igualmente composto pela ferramenta Worklist Application.

O Oracle Worklist constitui uma intuitiva interface Web que possibilita aos utilizadores se autenticarem no sistema de identificação do BPEL (Identity Service) e aceder a todas as suas tarefas e notificações, estejam ou não pendentes, e ainda as tarefas que foram por eles iniciadas nos vários processos BPEL a decorrer no servidor. Esta constitui de facto, a única solução existente no mercado a disponibilizar uma solução gráfica para interacção total com as funcionalidades da extensão BPEL4People.

O seu principal objectivo, para além de apresentar as tarefas e notificações, é disponibilizar uma total interacção dos utilizadores com os vários processos de negócio. As suas funcionalidades permitem de uma forma rápida e intuitiva responder, cancelar ou apagar uma dada tarefa, influenciando o fluxo de um determinado processo de negócio. A

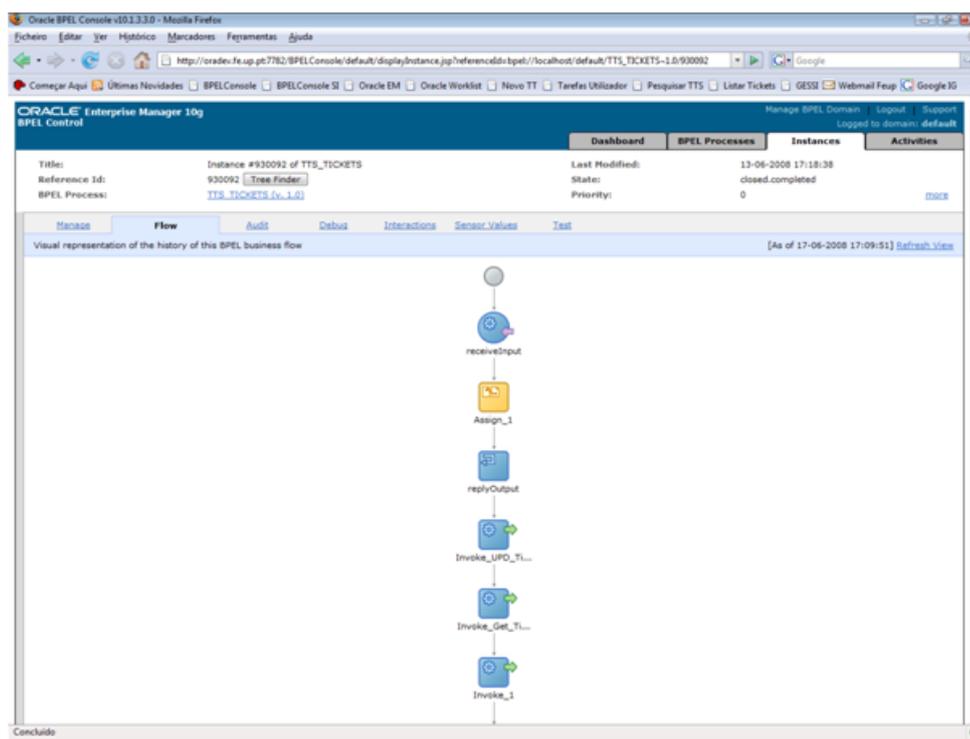


Figura 3.8: Apresentação do Oracle BPEL Console

figura 3.9 apresenta o ecrã principal desta aplicação, onde é possível identificar diversas tarefas e notificações que estão pendentes.

Ambas as ferramentas foram implementadas pela Oracle usando a sua própria API em Java, disponibilizada para a administração dos processos BPEL pelo seu servidor.

3.2.1.4 Repositório de Dados

A quarta camada de Base de Dados constitui igualmente um núcleo de informações para todos os processos BPEL. De facto, todas as acções, definições e propriedades dos processos de negócio, bem como todo o seu histórico, são guardados neste repositório. O seu modelo relacional pode ser visualizado na figura 3.10.

Uma das principais inovações deste produto no processamento de processos BPEL reside na Base de Dados. Esta inovação tem o nome de Dehydration e Hydration. Devido ao facto de que os processos podem ter um longo período de tempo para processamento, especialmente com a inclusão de tarefas para utilizadores onde é possível demorar vários dias até formalizar uma dada resposta, estes têm de ficar constantemente à espera de uma resposta para poder avançar o seu fluxo. Desta forma, a Oracle introduziu esta inovação que permite guardar todo o seu estado e processamento directamente na base de dados (Dehydration), libertando os recursos do servidor até que seja recebida uma resposta e

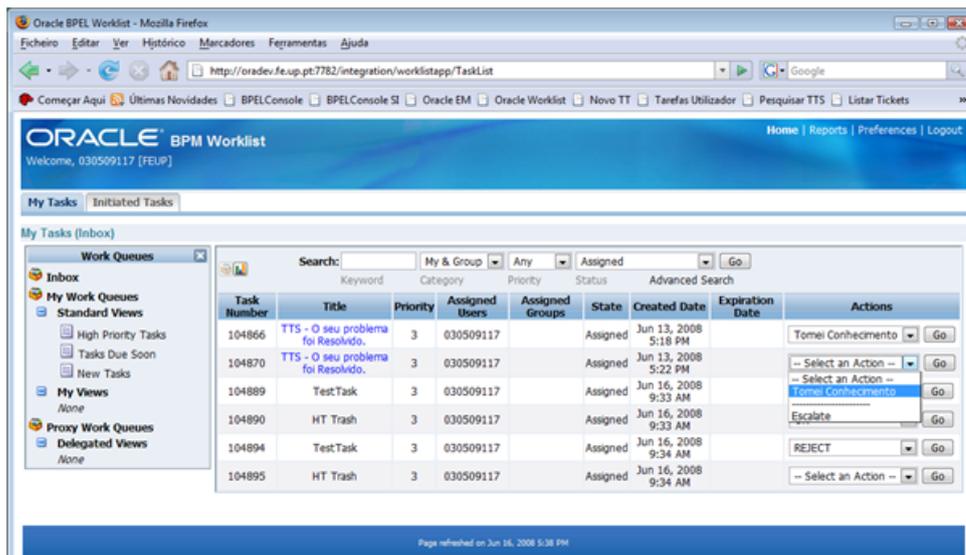


Figura 3.9: Apresentação do Oracle Worklist

todas as informações sejam restauradas da base de dados (Hydration), continuando novamente o seu normal processamento.

3.2.2 Comparação do Oracle BPEL com Outros Servidores

A decisão da escolha de utilização do Oracle BPEL Process Manager na implementação deste projecto, não se deveu apenas ao facto de todas as plataformas desenvolvidas pelo Projecto de Sistemas de Informação serem totalmente administradas e implementadas utilizando aplicações e sistemas desenvolvidos pela Oracle Software.

Na verdade o Oracle BPEL Process Manager é sem dúvida o mais completo e poderoso Servidor Aplicacional para BPEL existente actualmente. Ao longo dos anos em que a tecnologia BPEL foi sendo desenvolvida, a Oracle tem vindo a apostar bastante no seu futuro e na liderança do mercado com as suas aplicações para esta tecnologia. Mesmo concorrentes directos, que possuíam ferramentas bastante aceitáveis, foram totalmente adquiridos pela Oracle, como é o caso da BEA, que teve inclusive um papel fundamental no nascimento desta tecnologia.

Em termos de funcionalidades disponibilizadas, de simplicidades, de standards utilizados e de adaptadores, nenhum outro sistema possui actualmente comparação possível com esta ferramenta. Em termos de uma possível alternativa para utilizadores que preferem a plataforma .NET, seria a adopção do Microsoft BizTalk Server 2004, que tem vindo igualmente a apostar na integração dos conceitos da arquitectura SOA, embora sem fornecer tantas funcionalidades quantas as incluídas no Oracle BPEL Process Manager.

Por fim, para as empresas que têm preferência pela utilização de software livre, o JBoss Business Process Manager representa a melhor alternativa. A Oracle disponibiliza igualmente alguns pacotes e versões de integração para este servidor.

3.3 Resumo e Conclusões

As organizações e as pessoas envolvidas em sistemas de informação necessitam cada vez mais de trabalhar em conjunto. De igual modo, a forte evolução e desenvolvimento de novas tecnologias tem levado as organizações a adoptarem a integração da arquitectura SOA nos seus sistemas de informação, e a consequente adopção do BPEL como um standard único na utilização de Web Services. Estas medidas levam a um aumento dos benefícios e a optimizações em termos de adaptabilidade, integração, portabilidade e interoperabilidade entre diferentes sistemas, plataformas e aplicações.

Por outro lado, de forma a facilitar a implementação da tecnologia BPEL e a consequente automatização dos processos de negócio, existem actualmente ferramentas que tiram partido de todas as vantagens da conjugação de BPEL e SOA, como é o caso do Oracle BPEL Process Manager.

Deste modo, os conceitos apresentados ao longo deste capítulo constituem o resultado do estudo realizado na fase inicial deste projecto e que permitiram, posteriormente, a sua implementação e a consequente definição dos capítulos seguintes presentes neste documento.

Capítulo 4

Análise do Sistema na Versão Oracle Workflow

Este capítulo apresenta os detalhes do sistema na sua versão original antes da realização deste projecto. Pretende-se deste modo descrever o ponto de situação do sistema e do módulo dos Trouble Tickets no início do projecto, de forma a serem identificadas as principais alterações e possibilitar a especificação da proposta de evolução para o novo sistema que será apresentado nos capítulos seguintes.

A realização deste projecto incidiu sobre um módulo pertencente ao principal projecto do PSI, o sistema SIGARRA. Este é constituído por 22 módulos, onde se inclui o Trouble Tickets. Todos estes módulos correspondem a pacotes de procedimentos e funções em PL/SQL sobre um servidor aplicacional de Bases de Dados Oracle 10g, onde residem igualmente todos os modelos relacionais de dados que completam este sistema. Todas as páginas Web deste sistema são acedidas pelo nome do pacote seguido do nome do procedimento.

Deste modo, a arquitectura deste sistema caracteriza-se por um servidor aplicacional de dados, onde reside toda a informação necessária para o seu funcionamento. É principalmente devido a este facto que surge a necessidade da integração do BPEL para tornar o SIGARRA num sistema distribuído, fornecendo e acedendo a serviços de outras instituições.

O módulo de Trouble Tickets constitui uma importante ferramenta na manutenção deste Sistema. Permite reportar e solucionar problemas ou falhas que sejam detectadas, tanto a nível da instituição de ensino como a nível do próprio Sistema de Informação. Para isto, estão alocados diversos funcionários, maioritariamente pertencentes aos HelpDesks das várias unidades da instituição, cujas funções passam pela recepção, análise e tratamento dos vários problemas que são reportados. Quando necessário, são atribuídas tarefas

adicionais a utilizadores mais especializados na área do problema. Este módulo possibilita ainda uma total interacção entre os diversos utilizadores envolvidos na sua resolução, tanto a nível da troca de mensagens como a nível da disponibilização de documentos.

Contudo, desde a sua primeira implementação em 2004, este módulo já sofreu imensas alterações, principalmente na solução de diversos problemas detectados e na integração de novas funcionalidades. Deste modo, devido às necessidades de alterações rápidas para continuar o seu funcionamento, todo o módulo acabou por ficar bastante confuso, tanto a nível de implementação como de utilização, como irá ser demonstrado nos tópicos seguintes.

4.1 Processos de Negócio

Os processos de negócio utilizados pelo SIGARRA foram totalmente especificados utilizando o Oracle Workflow. Embora a maioria das suas funcionalidades estejam definidas nos pacotes de procedimentos PL/SQL, a sua interacção com o processo de negócio constitui um factor fundamental para o seu funcionamento, tanto em termos da definição de regras de execução como na sincronização das suas actividades e estados. Um processo de negócio torna-se ainda responsável pelo envio automático de notificações aos diversos utilizadores envolventes e nas alterações de dados conforme o avanço nas actividades que constituem o seu fluxo.

Em relação ao módulo dos Trouble Tickets, o seu processo de negócio é responsável por enviar notificações aos utilizadores envolventes, sobre os estados e avanços da resolução de um problema. É igualmente responsável por garantir a total integridade dos dados quando ocorram avanços ou retrocessos no seu fluxo.

No sistema em Workflow, um problema é caracterizado por três estados: Em Análise, Em Resolução ou Terminado. O primeiro estado inclui as actividades de notificação do HelpDesk responsável, que deverá analisar a sua possibilidade de resolução, ou reatribuir esta análise para um outro HelpDesk. Este passará por sua vez ao segundo estado quando for criada a primeira tarefa. Neste segundo estado são enviadas notificações aos executantes sobre as suas tarefas, e notificações de novas mensagens. Quando todas as tarefas estiverem concluídas, o problema pode passar à terceira e última fase, onde são canceladas as notificações pendentes e é enviada uma nova notificação ao utente sobre a sua conclusão. Durante o fluxo de actividades o processo de negócio é ainda responsável por verificar limites de tempo mínimos de acção excedidos, notificando os respectivos responsáveis pela unidade do problema.

A sua versão em Oracle Workflow inclui ainda as actividades responsáveis pela interacção com o sistema, como os acessos e alterações dos dados de um problema. Para visualizar este processo na sua versão Workflow, apresenta-se na figura 4.1 a sua definição real com a utilização do Oracle Workflow Builder.

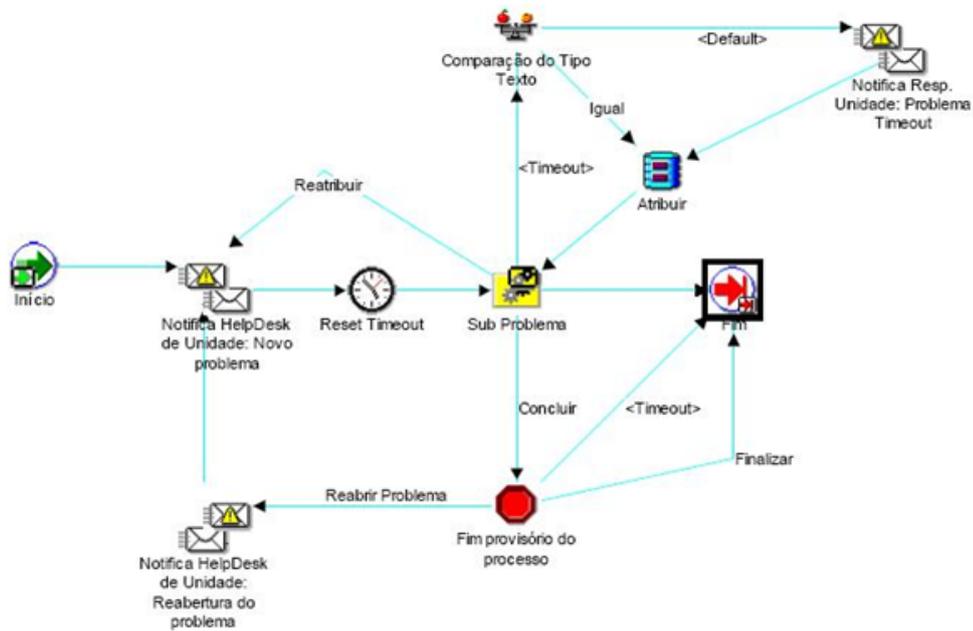


Figura 4.1: Processo de Negócio Principal do Trouble Tickets em Oracle Workflow

O processo apresentado na figura 4.1, representa o esquema principal das actividades de análise de um problema pelo HelpDesk. Neste diagrama é possível identificar algumas das actividades do Workflow que foram referenciadas no capítulo anterior em termos de mapeamento para o BPEL. É igualmente possível identificar os acessos e verificações de dados realizados no sistema, bem como as excepções por excesso do tempo limite na análise do problema.

Para visualizar o fluxo de actividades dos estados seguintes, é necessário apresentar o sub-processo agrupado no objecto Sub Problema no fluxo 4.1. Para isto, é apresentado na figura 4.2 o fluxo de actividades que compõem o Sub Problema deste processo.

Deste modo, a junção entre os dois modelos apresentados processa o fluxo de actividades deste módulo. No entanto, este fluxo decorre com algumas falhas a nível da implementação, sendo que algumas das suas actividades não possuem sincronização com o Sistema de Informação ou não são simplesmente utilizadas. A nível de notificações, apenas são enviados avisos, mesmo quando existem tarefas por resolver. Uma outra desvantagem é a não existência de interligações com outros módulos ou serviços exteriores.

De forma a superar todas estas desvantagens e otimizar este fluxo de actividades com toda as características de uma arquitectura distribuída e com a tecnologia BPEL, estas análises tornam-se, no entanto, fundamentais para obter toda a lógica de actividades e os objectivos do processo, simplificando deste modo a definição da sua nova versão em BPEL.

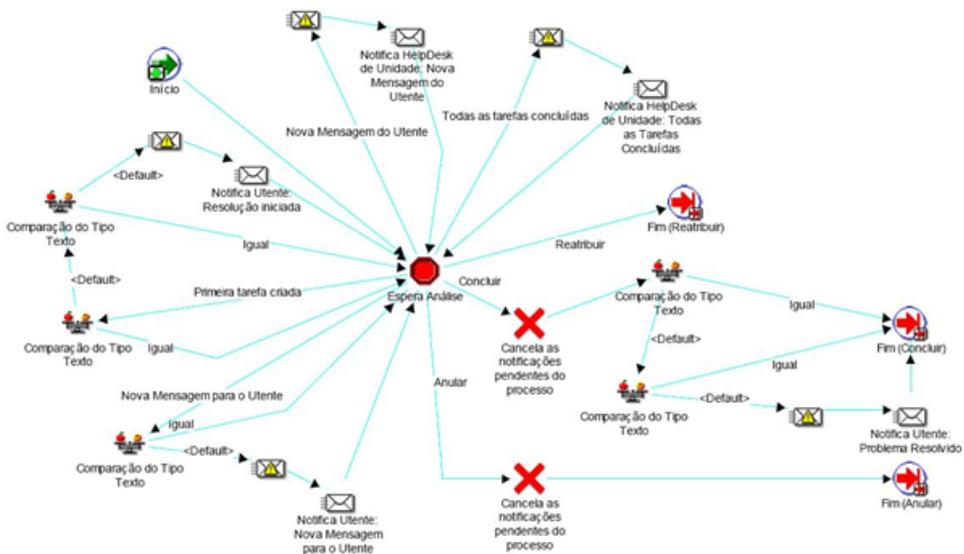


Figura 4.2: Sub-Processo de Negócio do Trouble Tickets em Oracle Workflow

4.2 Modelo Relacional de Dados

O modelo de dados original do módulo de Trouble Tickets na sua versão em Oracle Workflow, era constituído por dez tabelas, onde eram guardados os dados relativos aos problemas, tarefas, mensagens e documentos. Uma característica deste modelo era a inclusão das categorias de problemas (hipóteses), que se organizavam em três níveis: unidades, tipos de problemas e por fim os problemas. A estas unidades estavam associados HelpDesks, cujos funcionários eram responsáveis por analisar os problemas submetidos para a sua unidade.

No envio de mensagens e documentos estes poderiam ser associados ao problema ou a uma determinada tarefa. Relativamente as tarefas, existia igualmente uma tabela de precedências onde era definida a ordem de conclusão das tarefas. A figura 4.3 apresenta o modelo relacional de dados da versão original do módulo de Trouble Tickets.

4.3 Interfaces

No módulo dos Trouble Tickets, as interfaces com os utilizadores eram essencialmente caracterizadas pelo grande número de formulários apresentados para realizar simples operações. O excesso de tempo requerido para a realização destas operações era, de facto, a razão que os utilizadores e funcionários mais se queixavam.

Simple operações como a selecção da unidade, tipo de problema e problema requeriam uma actualização do formulário para apresentar os resultados coerentes. De igual forma, a escolha de utentes, salas e introdução de respostas era tudo efectuado em páginas

The figure displays four screenshots of the Oracle Workflow Trouble Tickets module forms, arranged in a 2x2 grid:

- Top Left: Criar Problema** - Form for creating a new problem. Fields include: Unidade (CICA - Centro de Informática Prof. Correia Araújo), Tipo de Problema (Sistemas de Informação), Problema (Outros), Descrição (teste para apagar), Utilizante, Outro Contacto, Sala, Tipo de Contacto (Email), Prioridade (3 - Média), Resposta Imediata, and Observações. Includes an 'Adicionar Documento' section with a 'Browse...' button and a 'Criar' button.
- Top Right: Detalhe do Problema** - Form showing details of a problem. Fields include: ID (2222), Problema (Sistemas de Informação - Outros), Descrição (teste para apagar), Data de Criação (2008-06-26 11:16), and Utilizante (Elisabete Ferreira da Silva). Includes a 'Criar Tarefa' section with fields for Unidade, Descrição, Executada por, and Observações. Includes an 'Adicionar Documento' section with a 'Browse...' button and an 'Inserir' button.
- Bottom Left: Alterar Problema** - Form for editing an existing problem. Fields include: Unidade (CICA - Centro de Informática Prof. Correia Araújo), Tipo de Problema (Sistemas de Informação), Problema (Outros), Descrição (teste para apagar), Data de Início (2008-06-26 11:16), Utilizante (Elisabete Ferreira da Silva), Sala, Tipo de Contacto (Email), Resposta Imediata (Não), Resposta, Data Prevista de Conclusão (YYYY-MM-DD), Prioridade (3 - Média), Problema na GP, and Observações.
- Bottom Right: Alterar Tarefa** - Form for editing an existing task. Fields include: Unidade (Unidade de Sistemas de Informação), Descrição (spdfpdf), Criada por (Elisabete Ferreira da Silva), Data de Criação (2008-06-26 11:17), Executada por (Elisabete Ferreira da Silva), Data de Início, Duração (Horas (Ex.: 1,5 = 1 hora e 30 minutos)), Problema na GP, and Observações.

Figura 4.4: Formulários Originais do Módulo de Trouble Tickets

4.5 Resumo e Conclusões

Observando o funcionamento do Sistema de Informação do SIGARRA e tendo em consideração o seu crescimento, nomeadamente na necessidade de desenvolver um sistema distribuído e interligado entre diversas instituições de ensino, é possível identificar que esta necessidade não era realizável com apenas as tecnologias utilizadas.

Por outro lado, dada a evolução do servidor aplicacional da Oracle para a versão 11g, todo o sistema em Workflow deixou de ser suportado nesta versão. Tudo isto levou de facto à necessidade de integração da tecnologia BPEL. Ainda relativamente ao módulo dos Trouble Tickets é igualmente evidenciado uma total complexidade na sua implementação e no seu modelo relacional para a disponibilização de simples funcionalidades. De igual forma, as suas interfaces acompanharam esta tendência e apresentam-se

de forma confusa e por vezes bastante lenta aos seus utilizadores.

No que diz respeito ao Sistema de Administração, este é na sua maioria constituído pelas ferramentas disponibilizadas pela Oracle para o efeito o que, por um lado, disponibiliza todas as funcionalidades necessárias para o efeito, mas por outro não permite um total controlo e integração com o Sistema de Informação em que está envolvido.

Capítulo 5

Transformação de Oracle Workflow para BPEL

Este capítulo introduz um conjunto de normas estabelecidas durante o desenvolvimento deste projecto que, conjuntamente com o estudo inicial realizado sobre a tecnologia BPEL, possibilitaram cumprir um dos seus objectivos principais: a evolução e transformação do módulo de Trouble Tickets, anteriormente definido em Oracle Workflow, para BPEL.

Todos os conceitos apresentados de seguida foram pensados de modo a cumprir, não só a transformação do módulo de Trouble Tickets, mas igualmente de todos os restantes módulos que compõem o SIGARRA. Estes são o resultado de uma análise comparativa entre as duas plataformas, que permitiu especificar mapeamentos de processos em Oracle Workflow para BPEL.

Numa fase inicial deste estudo, foi necessário aprofundar conhecimentos acerca da definição de processos utilizando a tecnologia BPEL. Visto que a ferramenta utilizada para o efeito foi o Oracle BPEL Designer, vários tutoriais estão disponibilizados na página da Oracle Software, desde o mais simples como o famoso "Hello World Tutorial", até aos mais complexos, de modo a permitir um estudo progressivo das utilidades e vantagens desta tecnologia. De igual forma é importante referir o fórum disponibilizado pela Oracle apenas para assuntos relativos a esta tecnologia, onde todos os dias milhares de utilizadores acedem aos seus conteúdos, permitindo uma maior facilidade na troca de informações e em obter esclarecimentos sobre eventuais dúvidas ou problemas que possam surgir.

Embora todos estes recursos sejam bastante abundantes em informações sobre esta tecnologia, a verdade é que o Oracle Workflow foi sendo esquecido, incluindo importantes definições sobre a sua evolução para BPEL, com a divulgação de que o Oracle

Workflow está obsoleto e que todos os processos devem agora ser definidos com o Oracle BPM. No entanto, foi disponibilizado um conjunto de boas práticas para os utilizadores que pretendam evoluir os seus processos para BPEL, onde são apresentadas as vantagens de utilização do BPEL e que procedimentos básicos devem ser tomados na evolução do Workflow.

Visto que não existir nenhuma conversão directa entre as duas linguagens, o processo de transformação do módulo de Trouble Tickets foi definido partindo totalmente do início. Seguindo as normas de boas práticas aconselhadas pela Oracle, o estudo realizado sobre a tecnologia BPEL foi determinante na análise de quais as funcionalidades e actividades definidas para este módulo em Workflow, para posteriormente se proceder à sua conversão para um processo BPEL que cumprisse os mesmos objectivos e funções.

Desta forma, este capítulo pretende especificar um possível mapeamento das funcionalidades mais importantes entre os processos em Oracle Workflow e o BPEL. No entanto, é importante referir que estas duas linguagens não apresentam diferenças apenas nas suas actividades e conteúdos. Todo o processo de definição dos processos de negócio diferem em ambas as tecnologias, a começar pela linguagem em que estão definidas, o Oracle Worklist em PL/SQL e o BPEL em XML, passando igualmente pelas ferramentas de administração, servidores, interacção com utilizadores, etc. Deste modo, a arquitectura SIBPEL que foi implementada neste projecto e que será apresentada nos seguintes capítulos, foi igualmente determinante e crucial na integração dos processos BPEL no Sistema de Informação do SIGARRA.

De modo a introduzir a comparação entre ambas as tecnologias, são apresentadas de seguida duas figuras que ilustram um exemplo simples de um processo definido em Oracle Worklist (Figura 5.1) e o mesmo processo definido novamente em BPEL (Figura 5.2) com o Oracle BPEL Designer.

O exemplo apresentado executa as mesmas funcionalidades tanto em Oracle Workflow como em BPEL. Ou seja, começa por receber um valor de entrada, lê um valor da base de dados e conforme o seu valor envia uma mensagem a um de dois grupos de utilizadores possíveis. De seguida actualiza a base de dados conforme a resposta do grupo à mensagem enviada, consulta novamente um outro registo na base de dados e conforme o seu valor retorna uma resposta diferente ao cliente.

Visualizando os dois esquemas com detalhe, observamos que não existe, à primeira vista, uma conversão directa entre ambas as linguagens. No entanto, os seguintes tópicos apresentam normas de conversão das funcionalidades entre ambas as tecnologias, que foram utilizadas na evolução do módulo de Trouble Tickets e que deverão ser reutilizadas igualmente nos restantes módulos.

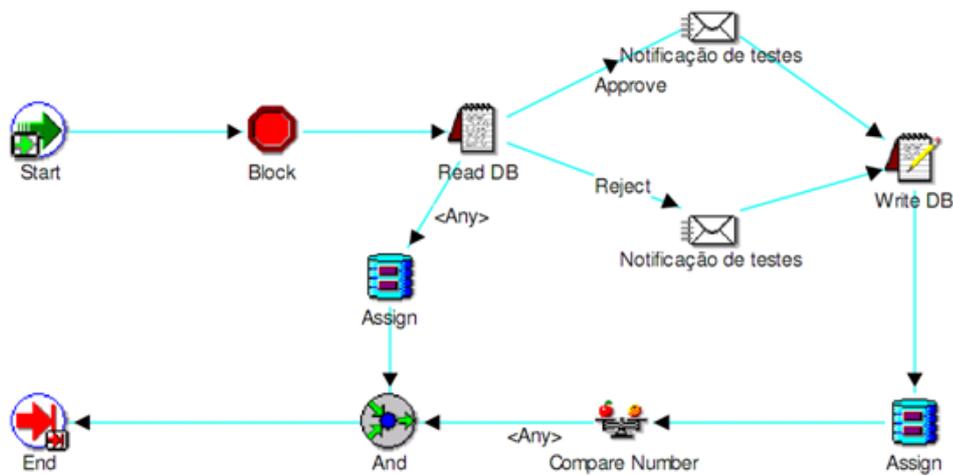


Figura 5.1: Exemplo de um Processo Definido em Oracle Workflow

5.1 Lista de Actividades BPEL

Com o nascimento da linguagem BPEL, surgiram novos conjuntos de actividades que apresentam bastantes diferenças comparativamente com o Oracle Workflow. Para proceder a uma especificação de normas de transformação, é necessário aprofundar os conhecimentos sobre as funções que cada actividade do BPEL desempenha na definição dos processos de negócio. Para isto, é apresentada na Tabela seguinte uma lista com as principais actividades do BPEL e um possível mapeamento das suas funcionalidades para as actividades do Oracle Workflow.

5.2 Normas de Conversão de Workflow para BPEL

Apresentam-se de seguida algumas normas estabelecidas ao longo da implementação deste projecto, para a evolução dos processos de negócio que compõem os módulos do SIGARRA. Estas normas representam decisões que foram especificadas para uma correcta conversão das funcionalidades de Oracle Workflow para o Oracle BPEL, permitindo igualmente ultrapassar algumas dificuldades verificadas ao longo da realização deste projecto na definição de processos em BPEL. De notar que a definição de processo difere da definição de instância. Um processo é apenas a definição de um fluxo de actividades. Uma instância é uma de muitas ocorrências de execução para o mesmo processo BPEL.

- Um processo BPEL será sempre iniciado através de uma actividade Receber, que recebe os parâmetros de invocação enviados pelo cliente.
- Caso o processo seja síncrono, o cliente ficará a espera de uma resposta, pelo que o processo deverá incluir uma actividade Responder, que envia uma resposta ao

Transformação de Oracle Workflow para BPEL

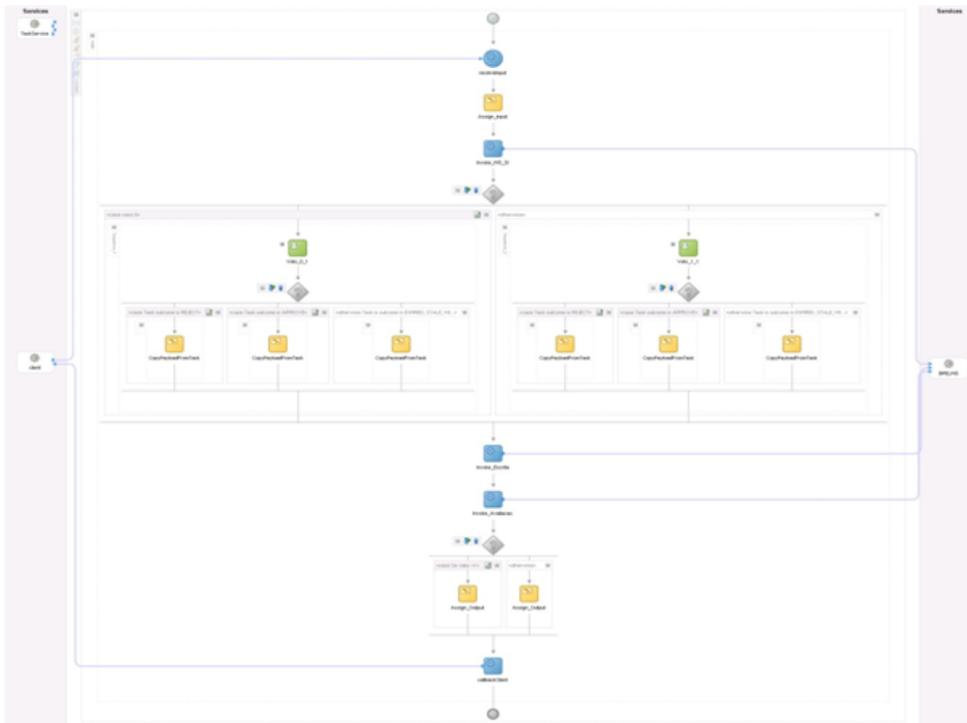


Figura 5.2: Exemplo de um processo definido em BPEL

cliente. Ter o cuidado de que o processo não demore a responder ao cliente, senão este lança uma exceção de timeout. No caso de se implementar um processo síncrono de longa duração, uma resposta deverá ser enviada ao cliente no início do fluxo, continuando depois o fluxo normalmente.

- Caso o processo não devolver nenhuma resposta, então este deverá ser definido como assíncrono.
- A definição de um processo em BPEL é implementada com actividades de forma sequencial, assim como o seu processamento.
- Todas as Notificações e Decisões devem ser implementadas com Tarefas Humanas. Estas permitem a definição dos utilizadores ou de um grupo de utilizadores que irão receber uma mensagem de decisão, onde tem de efectuar um voto, ou uma mensagem de notificação.
 - Para implementar uma Tarefa, a actividade Receber deverá estar incluída no fluxo da Tarefa, como acontece por defeito. Esta actividade irá parar todo o fluxo do processo enquanto uma resposta não for recebida. O nome destas tarefas deverá começar por "HumanTask".

- Para implementar uma Notificação, a actividade Receber deverá ser removida do fluxo de actividades da Tarefa. Desta forma, a mensagem é enviada, mas o fluxo não é interrompido. O nome destas notificações deverá começar por "Notificação".
 - Caso se pretenda que esta seja respondida por todos, deverá ser definida a propriedade "Voto por Maioria" na sua totalidade. Esta propriedade é sempre definida em percentagem relativamente ao total de utilizadores da tarefa.
 - Caso se pretenda que a tarefa seja completa quando o primeiro utilizador efectuar o seu voto, então a propriedade "Voto por Maioria" deverá conter uma percentagem de modo a corresponder a apenas 1 utilizador do total.
 - Caso se pretenda definir de um grupo dinâmico, antes do processamento da Tarefa, deverá ser definida a variável "task:systemAttributes/task:approvers" com a lista de utilizadores separados por vírgulas. De seguida, na definição da Tarefa, a lista de utilizadores deverá ser uma expressão a partir do valor desta variável. Este procedimento é necessário, pois o fluxo de actividades de uma tarefa não possui acesso às actividades exteriores.
 - Para integração e interacção de tarefas com Sistemas de Informação, deverá ser definido um utilizador sistema, em que o SI responde a estas tarefas utilizando a API de Java.
 - Por normas de interacção com as funcionalidades desenvolvidas neste projecto, todas as tarefas deverão definir o seu iniciador e o criador com o utilizador Sistema, de modo a ser possível posteriormente efectuar um controlo das mesmas.
 - Para envio de mensagens para utilizadores de um Sistema de Informação diferente do BPEL, como é o caso do SIGARRA, deverão ser igualmente definidos o task:identityContext com o nome do domínio de utilizadores e o task:OwnerUser com o utilizador que será o responsável pela tarefa.
 - Para definir uma data de expiração, deverá ser preenchida a propriedade "Expiração". No entanto, caso se pretenda Agendar uma tarefa, esta deverá antecipar a inclusão de uma actividade de Espera.
- O BPEL permite a associação de sensores a variáveis. Estes sensores reportam todas as alterações que estas variáveis sofreram. Deste modo, para obter um conjunto de estados por onde o fluxo de uma instância passou, deverá ser definido um sensor associado a uma variável global no processo com o nome EstadoProcesso, que irá sendo actualizado, através de actividades Atribuição, com a indicação do respectivo local onde o processo se encontra.

- Todos os acessos a Bases de Dados deverão agora ser implementados através de Serviços Externos. Isto permite facilmente realizar consultas, actualizações, inserções e alterações a registos da Base de Dados.
 - Com um Serviço Externo é igualmente possível aceder a Web Services externos. Para isto basta especificar a localização do ficheiro WSDL que representa o WS. Posteriormente, deverá ser incluída uma actividade Invocação, que irá invocar uma determinada operação do WS e obter a sua resposta. Para isto deverão igualmente ser definidas, na Invocação, duas variáveis, uma de chamada do WS com os seus parâmetros de entrada e uma outra com os parâmetros de saída.
 - Para realizar alterações e consultas às tabelas da base de dados, deverá ser definida uma variável com as colunas da tabela. De notar que apenas é possível aceder ou manipular um único registo em cada invocação. Para consultas e modificações de múltiplos registos, estes deverão ser implementados a partir de Web Services.
 - * Para Consultar um registo, deverão ser definidos os campos chave da variável, invocar o Serviço Externo com um Select e os campos da variável são automaticamente preenchidos.
 - * Para Inserir um registo, deverão ser preenchidos todos os campos obrigatórios da variável e invocar o Serviço Externo como um Insert.
 - * Para Apagar um registo, deverão ser preenchidos os campos chave da tabela na variável e invocar o Serviço Externo como um Delete.
 - * Para Actualizar um registo, deverão ser preenchidos os campos chave do registo e os campos que se pretendem actualizar, seguido da invocação do Serviço Externo como um Update.
- Caso se pretenda manipular um processo, para que este possa voltar atrás no seu fluxo, deverão ser implementados ciclos Repetir Enquanto ou então efectuar este tratamento através de Excepções e incluir uma actividade de Compensação, que permite definir o Agrupamento para onde o processo deverá voltar.
- No registo de novos processos no servidor de BPEL, estes ficam automaticamente registados como a versão 1.0.
 - Quando se efectuarem alterações a este processo, ao fazer um novo registo, este deverá ter uma versão acima da mais recente no servidor, de modo a continuar o processamento das instâncias a decorrerem no servidor para esse processo.
 - Caso de efectuem registos de processos iguais e com a mesma versão, todas as instâncias em processamento são automaticamente abortadas e ficam no estado

de desactualizado. O mesmo acontece com as tarefas e notificações pendentes nessas instâncias, que são igualmente canceladas.

– É igualmente possível no BPEL iniciar processos pelas versões antigas, o que não acontecia no Workflow, que apenas permite iniciar a mais recente.

- Ao abortar uma instância, todo o seu fluxo é cancelado mas as suas informações mantêm-se na base de dados como histórico. No entanto, é necessário ter em atenção que as suas tarefas e notificações pendentes mantêm-se activas.
- Ao apagar uma instância, todo o seu fluxo é cancelado e todas as suas informações são eliminadas da base de dados. O mesmo acontece para as tarefas e notificações, que neste caso, são igualmente apagadas e desaparecem da lista de notificações dos utilizadores.
- Quanto à actividade Look Up do Oracle Workflow, que permite identificar tipos de variáveis, uma grande vantagem do BPEL é que estes deixam de ser necessários, dado que o este processa automaticamente estes tipos. No Workflow esta actividade introduzia bastantes problemas e complexidades.
- A actividade Loop Counter do Oracle Workflow deverá ser implementada no BPEL a partir de uma variável contadora.
- Uma possível migração directa do histórico dos processos do Oracle Workflow para o BPEL não deverá ser considerada, dadas as grandes diferenças entre ambas as plataformas. No entanto, poderá ser especificada uma possível migração do histórico do Workflow para o Sistema de Informação, sendo posteriormente acedido pelo BPEL.

5.3 Análise Comparativa entre o BPEL e o Oracle Workflow

Sendo actualmente o Oracle Workflow considerado um produto descontinuado, este facto é por si próprio uma excelente razão para existir a necessidade da sua evolução para o seu sucessor, a tecnologia BPEL.

Para além desta nova tecnologia permitir a criação de novas actividades e a definição de novos conceitos para além dos já existentes em Workflow, a sua simplicidade na definição de processos de negócio e na ligação entre aplicações e sistemas supera todas e quaisquer características do Oracle Workflow.

Por outro lado, há ainda que realçar a correcção de diversos problemas existentes no seu antecessor. No entanto, e como em todas as tecnologias, serão sempre encontradas limitações e anomalias no seu funcionamento. Contudo, sendo esta a tecnologia mais

poderosa existente actualmente no seu contexto, existe a garantia de que será constantemente actualizada e disponibilizadas novas funcionalidades. Dadas as suas características, o BPEL possui realmente uma forte capacidade de evolução, o que não se verificava com o Oracle Workflow.

Além de todos estes factos, a tecnologia BPEL representa um standard universal, utilizado por diversas empresas mundiais que apostam bastante na sua evolução. A Oracle é actualmente a empresa que mais tem apostado no seu desenvolvimento, e a sua nova versão 11g trará muitas novidades para esta tecnologia.

5.4 Resumo e Conclusões

Através de todos os conceitos apresentados neste capítulo, é possível concluir que a conversão de processos de negócio em Oracle Workflow para BPEL não é realizado de forma directa. Para isto é necessário resumir a aplicação destes conceitos, para estabelecer as seguintes etapas de conversão que deverão ser tomadas em conta, principalmente na evolução dos restantes módulos existentes no SIGARRA:

1. Analisar o fluxo de actividades do processo de negócio definido em Oracle Workflow, identificando os seus principais objectivos e funcionalidades.
2. Repartir e redefinir todo este fluxo de modo a utilizar e reutilizar outros fluxos, aplicações, sistemas e serviços externos. Expandir o processo de forma distribuída.
3. Consultar os detalhes e explicações das funcionalidades e actividades constituintes do BPEL.
4. Seguir as normas de conversão definidas neste projecto e apresentadas neste capítulo, de forma a criar correctamente um processo de negócio em BPEL integrado no SIGARRA.

Mais do que meras regras de conversão, os conceitos apresentados neste capítulo constituem meios alternativos para a realização das funcionalidades do Workflow em BPEL e que devem ser consideradas de forma a ultrapassar os problemas e dificuldades sentidas ao longo da realização deste projecto.

Transformação de Oracle Workflow para BPEL

Tabela 5.1: Lista das Principais Actividades BPEL

Lista de Actividades BPEL		
Actividade	Descrição	Equivalência em Work-flow
Atribuição	Atribui valores ou expressões a variáveis e mensagens. Permite manipular, copiar e converter dados.	Atribuição, Comparação e Transformação
Email, SMS, Fax, Pager e Voz	Envia mensagens e notificações de texto por Email, SMS, Fax e Pager e ainda mensagens de voz para telemóveis e telefones.	Notificação (apenas Email)
Fluxos	Define múltiplos fluxos de actividades paralelas.	And
Tarefa	Envia uma Tarefa de decisão ou uma Notificação a diversos utilizadores, permitindo a sua interacção com o fluxo do processo.	Notificação (apenas para o envio de avisos)
Invocar	Permite invocar uma operação de um Serviço Externo, aceder a servidores FTP, ficheiros e bases de dados.	Function e Launch Process (apenas para aceder à Base de Dados)
Java	Inclui código Java no processo.	Function
Receber	Permite criar automaticamente os parâmetros que são retornados por um serviço externo e receber as respostas por eles enviadas após a sua invocação.	Receive Event e Block (apenas para receber valores de uma função)
Responder	Sendo todo o processo definido como um serviço, esta actividade permite retornar a resposta final da sua invocação.	-
Condição	Implementa um conjunto de condições que processam actividades caso sejam válidas. Semelhante a um Case das linguagens de programação comuns.	Comparação
Terminar	Aborta a execução do processo.	Fim
Excepção	Lança uma excepção caso detecte alguma falha ou erro, possibilitando o seu tratamento.	-
Espera	Interrompe o processamento do fluxo durante um determinado período de tempo.	Wait
Repetir	Implementa um ciclo de actividades que são repetidas enquanto a sua condição de saída não for verificada.	-
Serviço	Define um adaptador para uma ligação a um Web Service, Base de Dados, FTPs, Ficheiros, etc..	Function (apenas para ligações à Base de Dados)

Capítulo 6

Especificação de Evolução de um Módulo

Com a realização de todo o estudo e conceitos apresentados nos capítulos anteriores, foi possível especificar uma solução de implementação deste projecto no Sistema de Informação, nomeadamente a integração da tecnologia BPEL no SIGARRA.

No entanto, todo este processo teve início em Julho de 2007, com o envolvimento de dois colaboradores do PSI, na implementação do motor do Serviço de Identificação, que permite identificar os utilizadores do SIGARRA como possíveis utilizadores do BPEL.

Deste modo, para o início da elaboração deste projecto foi necessário especificar uma nova Arquitectura SIBPEL de ligação dos actuais módulos desenvolvidos no Sistema de Informação com toda a tecnologia BPEL, focado especialmente para o módulo dos Trouble Tickets que constitui o objectivo principal deste projecto. Para este processo de evolução, foram ainda definidos novos requisitos e casos de utilização para o novo Trouble Tickets de modo a que, por um lado, o seu funcionamento tirasse o máximo partido das vantagens de utilização do BPEL e que, por outro lado, fossem implementadas novas funcionalidades e integradas novas tecnologias que visassem melhorar a sua utilização e interacção com os utilizadores.

Assim, são apresentadas nos seguintes tópicos, descrições detalhadas das soluções propostas para implementação deste projecto, sendo que os aspectos relativos à sua implementação serão apenas apresentados no capítulo seguinte.

6.1 Serviço de Identificação

De modo a integrar os utilizadores do SIGARRA na tecnologia BPEL, foi essencial proceder-se a alteração directa do código fonte do seu Serviço de Identificação. Sabendo

que o BPEL contém um domínio próprio de utilizadores, foi então necessário definir um novo domínio no BPEL que incluísse todos os utilizadores definidos no modelo relacional de dados do Sistema de Informação da FEUP e os validasse como utilizadores autenticados no seu Serviço de Identificação.

Toda a implementação deste domínio e dos métodos necessários para o acesso do Serviço de Identificação à lista de utilizadores do SIGARRA, foi implementado antes do início deste projecto por um dos colaboradores do PSI, sem o qual a realização este projecto não teria sentido para a instituição.

As principais alterações efectuadas permitem a definição do domínio FEUP para a autenticação e envio de tarefas e notificações aos seus utilizadores, sendo esta lista criada e enviada por um procedimento PL/SQL. Deste modo, tornou-se ainda possível a criação de uma estrutura base para a definição de futuros domínios de outras instituições.

6.2 A Arquitectura SIBPEL

A especificação da arquitectura SIBPEL marcou, como o próprio nome indica, o início da especificação de uma solução de ligação do Sistema de Informação do SIGARRA com a tecnologia BPEL, representando igualmente o principal elo de comunicações entre as diversas tecnologias presentes neste projecto.

Sendo duas plataformas completamente diferentes, o SIGARRA em PL/SQL enquanto o BPEL é XML com administração e processamento definidos em Java, houve a necessidade de analisar e especificar uma interligação física e lógica entre ambas as plataformas. Esta arquitectura foi definida de forma a tirar partido de todas as vantagens do BPEL, conforme apresentado é na figura 6.1.

A arquitectura SIBPEL foi estudada de modo a possibilitar uma integração do novo módulo de Trouble Tickets no Sistema de Informação, bem como dos restantes módulos, interagindo totalmente com os respectivos processos de Negócio definidos em BPEL.

Para realizar o acesso às funcionalidades da tecnologia BPEL e aos conteúdos inseridos no seu modelo de dados, foram estudadas as possibilidades de integração dos seguintes acessos:

- O acesso aos dados seria efectuado e manipulado directamente pela camada de dados do Oracle BPM.
- O acesso às funcionalidades seria efectuado através da inclusão de objectos Java no próprio Sistema de Informação, utilizando a API em Java fornecida, enquanto que a comunicação seria efectuada directamente através da invocação de métodos Java a partir de PL/SQL.
- O acesso seria efectuado através da implementação de Web Services, tirando partido das funcionalidades da API de Java fornecida pelo Oracle BPEL Process Manager,

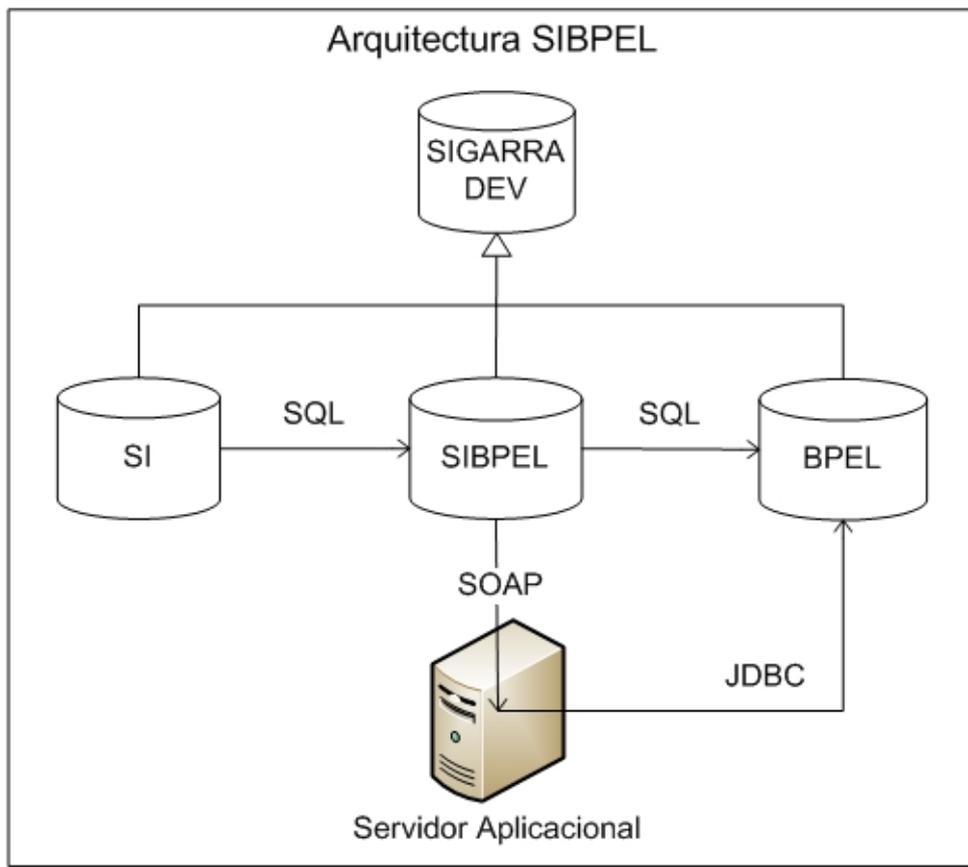


Figura 6.1: Arquitectura SIBPEL

sendo a comunicação efectuada através do envio e recepção de mensagens SOAP a partir de PL/SQL.

Sendo a API em Java, um standard de acesso às funcionalidades do BPEL, utilizada igualmente pelas ferramentas de administração integradas no Oracle BPM, de modo a simplificar possíveis alterações na implementação com a disponibilização de novas versões do BPEL, tendo em vista principalmente a evolução para a nova versão 11g, ficou especificado que a melhor solução seria a utilização desta API. No entanto a sua integração no Sistema de Informação não foi possível, dado o elevado número de classes e objectos definidos nas bibliotecas que são disponibilizadas pelo Oracle BPM, que originavam dependências recursivas entre ambas, não permitindo uma correcta compilação das principais classes.

Por outro lado, dada a implementação da tecnologia Dehydration na camada de dados do Oracle BPM, e tal como é apresentado na figura 3.10, esta camada é composta por tabelas não normalizadas e cujos dados são bastante confusos e de difícil compreensão de significados, como é o caso da utilização de diversos códigos na representação de estados.

Desta forma, a arquitectura SIBPEL é principalmente caracterizada pela implementação de diversos Web Services em Java, que comunicam através de mensagens SOAP com os procedimentos implementados em PL/SQL que compõem a lógica de negócio do Sistema de Informação do SIGARRA. Estes Web Services foram utilizados no acesso aos dados, gestão e administração das informações dos processos de negócio e respectivas instâncias registadas no servidor. No entanto, por questões de implementação e como será referido com maior relevância nos seguintes capítulos, em algumas situações houve a necessidade de aceder directamente por SQL aos dados contidos no repositório BPEL.

Para a realização destas ligações e inclusão dos componentes referenciados, a arquitectura SIBPEL é composta pelos seguintes componentes:

SIGARRA DEV Consiste num servidor aplicacional Oracle 10g de desenvolvimento de novas funcionalidades para o Sistema de Informação do SIGARRA. É neste servidor que se encontram os diversos repositórios de Bases de Dados que compõem todo este sistema e onde se incluem o SI, o BPEL e o SIBPEL.

Repositório SI Aqui encontram-se todas as tabelas, procedimentos, triggers, funções e pacotes que constituem o ambiente de desenvolvimento da instituição. É neste repositório que encontra implementado o novo modelo relacional do Trouble Tickets, bem como todos os procedimentos e funções que implementam toda a lógica de negócio e de acesso a dados por parte deste módulo. De forma a integrar e interagir com a tecnologia BPEL, este repositório contém ainda procedimentos que acedem às funcionalidades implementadas no SIBPEL através de comandos SQL.

Repositório BPEL É neste repositório que se encontram todos os dados relativos aos processos e instâncias BPEL registados no Oracle BPM. É composto principalmente por tabelas não normalizadas, cujo modelo relacional é apresentado na figura 3.10, apenas para inclusão e acesso de todas as suas informações, nomeadamente na utilização da tecnologia Dehydration. Todos os métodos e objectos definidos na API em Java acedem directamente a estes dados, através de ligações JDBC de comunicação Java com a camada de dados Oracle. No entanto, conforme referido anteriormente, são igualmente efectuados acessos a estes dados através de consultas SQL, de forma a completar todas as necessidades na implementação do Sistema de Informação e da camada de ligação deste com o BPEL.

Repositório SIBPEL Este repositório consiste no componente principal desta arquitectura. É composto por procedimentos em PL/SQL que permitem a ligação de todo o Sistema de Informação aos Web Services disponibilizados no Servidor Aplicacional e ao repositório do Oracle BPM. Desta forma é então possível aceder às funcionalidades implementadas em Java para acesso e controlo da tecnologia BPEL, bem como a outros serviços externos. Sendo os próprios processos BPEL definidos como

serviços externos, a partir desta camada torna-se então possível iniciar directamente uma nova instância para um dado processo, através da construção e envio de uma mensagem SOAP. Para isto, esta camada é constituída pelos seguintes componentes:

- **Pacote SOAP:** Este pacote contém os procedimentos e funções em PL/SQL necessárias para construção, envio e recepção de mensagens SOAP. As suas funcionalidades tiram partido da classe UTL_HTTP do PL/SQL.
- **Pacote Web Services:** Aqui são incluídos todos os procedimentos e funções em PL/SQL implementados para a invocação de serviços externos, principalmente para os Web Services implementados com a API de Java. Para isto, utiliza o pacote SOAP de forma a enviar os seus pedidos e a receber as respostas retornadas.
- **Pacote BPEL:** Este pacote define todos os procedimentos e funções em PL/SQL para o acesso, gestão e administração de instâncias BPEL. São aqui incluídas funcionalidades que permitem a construção dos parâmetros das mensagens SOAP utilizadas na invocação dos serviços, bem como de descodificação das respostas recebidas, através da tecnologia XPATH. Para isto, tira partido dos procedimentos e funções implementados nos anteriores pacotes. Todos os acessos ao repositório SIBPEL implementados neste projecto são acessos às funcionalidades contidas neste pacote.

Servidor Aplicacional Oracle SOA Suite Representa o segundo servidor aplicacional constituinte deste projecto. Aqui encontra-se todo o motor de negócio do servidor Oracle BPEL Process Manager, incluindo todas as aplicações que utilizam a sua API em Java.

Para isto, foram ainda aqui registados todos os Web Services com as funcionalidades de acesso e controlo da tecnologia BPEL representando uma base fundamental na elaboração deste projecto. Estes Web Services foram implementados segundo os métodos e classes de objectos fornecidos pela API de Java no Oracle BPM, que por sua vez consultam e manipulam os dados contidos no repositório através de conexões em JDBC. As operações disponibilizadas por estes Web Services, encontram-se agrupados no Web Service Provider e no Web Service Console, que definem respectivamente operações de acesso aos dados do BPEL e operações de administração.

Os acessos efectuados aos serviços disponibilizados neste servidor aplicacional são efectuados através da troca de mensagens SOAP. Os serviços aqui disponibilizados serão apresentados com um maior detalhe de implementação nos seguintes capítulos.

Relativamente a especificação da implementação deste projecto, este servidor aplicativo é ainda composto por um projecto de aplicação Web em Java, o Web Flow, que for forma a implementar funcionalidades extra de administração para este projecto, contém ficheiros XML e Java Server Pages (JSP) para acesso aos detalhes de actividades em fluxos de instâncias BPEL.

6.3 Especificação do Novo Trouble Tickets

O segundo objectivo principal deste projecto, para além da integração da tecnologia BPEL no Sistema de Informação, era a transformação e evolução de todo o módulo de Trouble Tickets. Esta transformação foi elaborada tanto a nível do seu processo de negócio mas também a nível da disponibilização de novas funcionalidades e do melhoramento da sua interface com os utilizadores, de forma a permitir a utilização de novas tecnologias, tais com AJAX e Javascript.

Para isto, foram estipulados os requisitos funcionais, não funcionais e casos de utilização apresentados nos seguintes pontos. De notar que apesar de alguns dos requisitos que serão apresentados estarem implementados na versão anterior, estes foram novamente especificados como requisitos de melhoramento da eficácia, rapidez e simplicidade das funcionalidades existentes.

6.3.1 Requisitos Funcionais

Após a realização de diversas reuniões, tanto com utilizadores do antigo módulo do Trouble Tickets, como com os seus utilizadores administradores, principalmente pertencentes aos HelpDesks informáticos, foram estipulados os seguintes requisitos funcionais para a nova versão deste módulo, tirando partido das vantagens da sua interligação com a tecnologia BPEL:

Permitir que todos os utilizadores possam:

- Reportar problemas de forma rápida e intuitiva.
- Consultar a lista de problemas colocados.
- Visualizar notificações e tarefas pendentes para os seus problemas.
- Reabrir um dado problema concluído.
- Adicionar documentos e mensagens para auxiliar a resolução do problema.

Permitir que os funcionários do HelpDesk possam:

- Reportar rapidamente e intuitivamente problemas no sistema que chegam via Email, telefone ou pessoalmente.

- Aceder a um conjunto de listagens rápidas de problemas que incluem os problemas por analisar ou por resolver pelo HelpDesk, bem com os problemas terminados.
- Pesquisar problemas relacionados com o seu HelpDesk.
- Analisar a resolução de um dado problema, atribuindo tarefas a diversos utilizadores do sistema.
- Redefinir a análise do problema para um outro HelpDesk.
- Disponibilizar documentos e enviar mensagens para o utente ou para outros utilizadores envolventes na resolução do problema.
- Anular ou Concluir a Resolução de um dado problema.
- Visualizar a lista de actividades processadas pelo fluxo do processo de negócio BPEL relativo a um dado problema.

Permitir que os executantes de tarefas de problemas possam:

- Anular, Concluir ou Agendar a resolução de uma determinada tarefa.
- Disponibilizar documentos e enviar mensagens para o utente ou para outros utilizadores envolventes na resolução do problema.

6.3.2 Requisitos Não Funcionais

Serão de seguida apresentados os principais requisitos não funcionais especificados para este sistema, que incluem algumas restrições ao seu funcionamento e à qualidade final do seu produto desenvolvido. Para isto, estes podem ainda ser identificados nos seguintes agrupamentos:

- **Requisitos de Usabilidade:**
 - As interfaces gráficas deverão ser intuitivas e a sua utilização deverá ser rápida e eficiente.
 - As funcionalidades e informações deverão ser disponibilizadas no menor número de páginas e iterações possível.
 - Deverão ainda ser introduzidas novas tecnologias tais como o Ajax e Javascript de modo a satisfazer os anteriores requisitos.
 - Os utilizadores frequentes do sistema deverão estar envolvidos no seu desenvolvimento, de forma a garantir a toda satisfação das suas necessidades e uma melhoria no cumprimento dos seus objectivos.

- A utilização da tecnologia BPEL e as chamadas a serviços externos deverão ser transparente para o utilizador.

- **Requisitos de Serviços:**

- A disponibilização dos serviços deverá ser generalizada, de modo a funcionar correctamente com diferentes sistemas e tecnologias.
- Deverá ainda ser considerada a inclusão de atributos de qualidade nos serviços tais como a segurança e a autenticação dos seus acessos.

- **Requisitos de Sistema:**

- O funcionamento deste projecto não deverá ficar dependente da estrutura de componentes que compõem o SIGARRA. Para isto, deverão ser eliminadas e generalizadas todas as ligações aos servidores e às funcionalidades necessárias, de forma a possibilitar a sua futura integração em diversas instituições de ensino.
- O sistema deverá permitir facilmente futuras manutenções e actualizações.

6.3.3 Casos de Utilização

Dados os requisitos funcionais e não funcionais apresentados, foram especificados os utilizadores envolventes com o sistema e os seus casos de utilização no novo módulo dos Trouble Tickets. Para isto, a figura 6.2 apresenta o diagrama geral de pacotes UML dos casos de utilização definidos para este projecto.

6.3.3.1 Actores

Os actores que interagem directamente com as funcionalidades deste módulo são utilizadores registados no Sistema de Informação do SIGARRA. Estes podem ser distinguidos pelas seguintes três categorias:

Utente Representam todos os utilizadores do SIGARRA, normalmente docentes e alunos, que identificam um determinado problema nas instalações do estabelecimento de ensino ou no próprio sistema de informação. De forma a verem o problema solucionado, estes utilizadores reportam este problema no módulo dos Trouble Tickets e auxiliaram a sua resolução, providenciando todas as informações e documentos necessários.

HelpDesk Consistem os utilizadores maioritariamente pertencentes aos funcionários dos HelpDesks informáticos do estabelecimento de ensino, que possuem funções de recepção, análise e resolução dos problemas reportados. Estes podem ainda reportar novos problemas no caso de um contacto directo com um Utente.

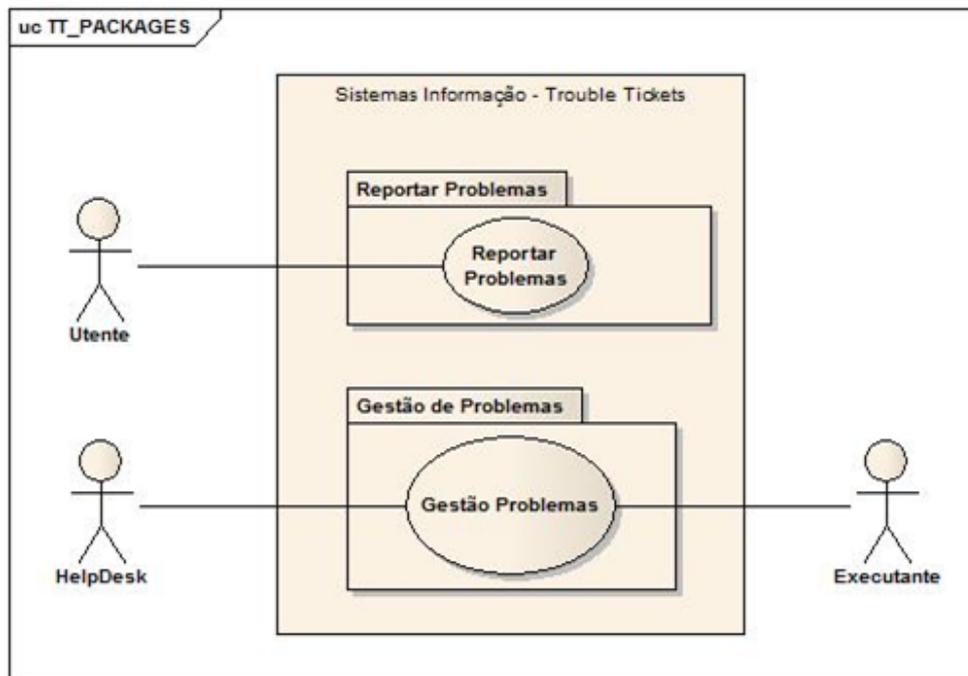


Figura 6.2: Diagrama Geral de Pacotes dos Casos de Utilização do Novo Módulo de Trouble Tickets

Executante Agrupam todos os utilizadores do sistema que possuem um papel importante no auxílio da resolução de um problema. Estes utilizadores são identificados como executantes de um problema quando lhes são atribuídas a resolução de determinadas tarefas para esse problema.

6.3.3.2 Módulo de Reportar Problemas

O pacote Reportar Problemas engloba os casos de utilização especialmente destinados aos utentes. Aqui se incluem funcionalidades que permitem reportar novos problemas e consultar os demais colocados, bem como auxiliar na sua resolução fornecendo todas as informações necessárias. A figura 6.3 apresenta o diagrama de casos de utilização UML para este pacote.

6.3.3.3 Módulo de Gestão de Problemas

O pacote Gestão de Problemas engloba os casos de utilização destinados aos funcionários do HelpDesk e aos executantes. Aqui se incluem funcionalidades que possibilitam consultar, analisar e resolver problemas e demais tarefas atribuídas. A figura 6.4 apresenta o diagrama de casos de utilização UML para este pacote.

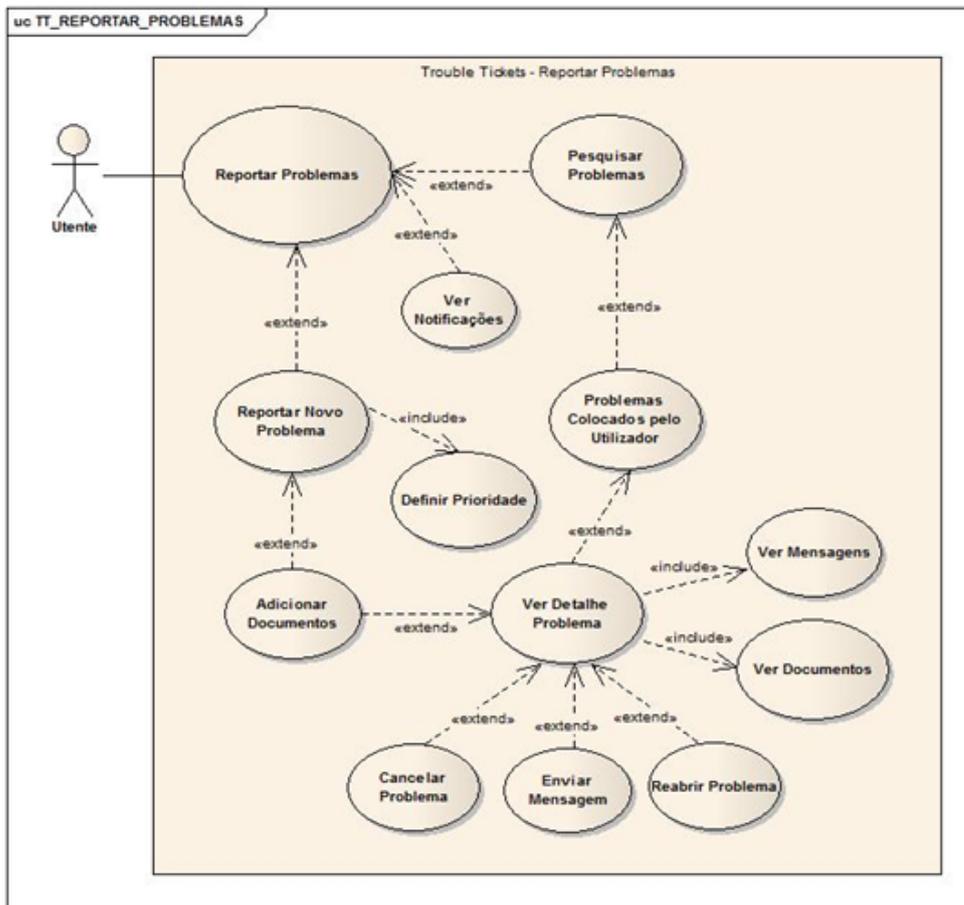


Figura 6.3: Diagrama de Casos de Utilização do Pacote Reportar Problemas

6.4 Especificação do Sistema de Administração BPEL

Com o atempado cumprimento de todos os objectivos propostos para este projecto, surgiu o interesse do desenvolvimento de um sistema de administração BPEL à semelhança do já existente para a versão Workflow.

Com a utilização do Oracle BPEL Process Manager e a disponibilização da sua API em Java com diversos métodos de acesso e controlo de funcionalidades do BPEL, este objectivo suplementar possuía um suporte fiável para ser implementado.

Dados estes factos, a sua implementação poderia ser vista como uma nova implementação das aplicações desenvolvidas pela Oracle, nomeadamente o Oracle BPEL Console e o Oracle Worklist, no próprio sistema de informação do SIGARRA. Ao contrário do que acontece na versão em Workflow, que utiliza directamente as ferramentas de administração da Oracle, a versão em BPEL poderia ter o seu próprio controlo do seu sistema totalmente a partir do SIGARRA, fornecendo as mesmas funcionalidades da consola e do gestor de tarefas da Oracle.

Especificação de Evolução de um Módulo

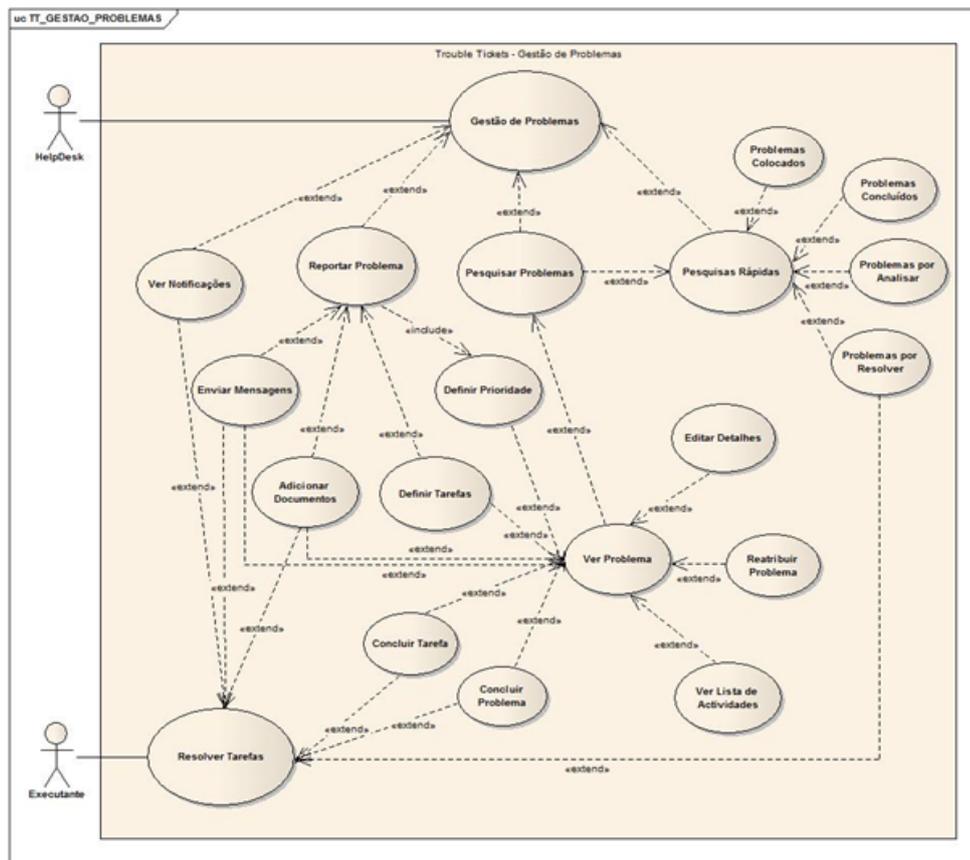


Figura 6.4: Diagrama de Casos de Utilização do Pacote Gestão de Problemas

6.4.1 Casos de Utilização

Para a especificação das funcionalidades e requisitos do sistema de administração, foram definidos inicialmente os casos de utilização que este deveria possuir, tendo em consideração os métodos e funcionalidades disponíveis na API em Java.

Observando o diagrama UML apresentado na figura 6.5, o sistema deveria ser utilizado por dois conjuntos de actores: os administradores e restantes utilizadores. Estes deveriam possuir acesso respectivamente às funcionalidades da Gestão de Processos e Instâncias BPEL (Consola BPEL) e à Gestão de Tarefas e Notificações (Gestor de Tarefas).

Quanto ao desenvolvimento da consola BPEL, este deveria disponibilizar as mesmas funcionalidades e informações que a aplicação Oracle BPEL Console, com a vantagem de ser desenvolvido no próprio sistema de informação, seleccionando deste modo, de entre todos os seus utilizadores, os que seriam administradores e possuiriam total acesso às suas funcionalidades, bem como a total definição da sua própria interface de utilização.

De igual modo, o gestor de tarefas deveria disponibilizar as mesmas funcionalidades e informações que a aplicação Oracle Worklist. A sua integração no SIGARRA deveria

Especificação de Evolução de um Módulo

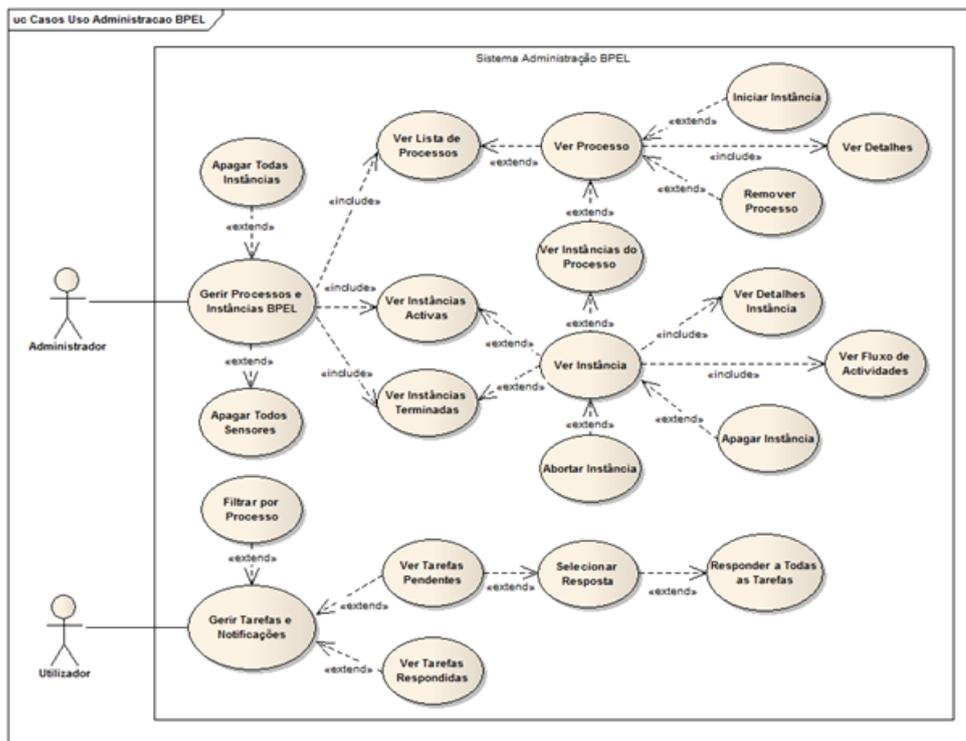


Figura 6.5: Diagrama de Casos de Utilização do Sistema de Administração BPEL

permitir automaticamente listar as tarefas e notificações do utilizador autenticado no sistema e responder a todas as tarefas pendentes de uma só vez, constituindo uma melhoria em relação à versão da Oracle que apenas permite responder a apenas uma tarefa de cada vez.

Capítulo 7

Projecto e Implementação

Após a especificação da solução proposta para a evolução do sistema, descrevem-se neste capítulo os detalhes da sua implementação, da gestão de ligações e da integração de componentes e tecnologias, de forma a cumprir os requisitos e objectivos deste projecto.

Serão apresentados nos seguintes pontos, os detalhes da arquitectura lógica e física do sistema distribuído implementado, seguido da estrutura de serviços externos desenvolvidos que possibilitaram a implementação da lógica de negócio do novo módulo de Trouble Tickets e do Sistema de Administração da tecnologia BPEL, cujos detalhes e componentes serão igualmente apresentados.

7.1 Arquitectura Lógica do Sistema

7.1.1 Decomposição Horizontal

O desenvolvimento do sistema de integração da tecnologia BPEL e de transformação do módulo dos Trouble Tickets, foi elaborado de forma corresponder as especificações da arquitectura SIBPEL, enquadrando-se em cinco camadas lógicas hierárquicas, que se diferem de acordo com o nível de abstracção em que se encontram, com o tipo de informação com que interagem e com as ligações que utilizam. A representação da arquitectura horizontal do sistema é apresentada na figura [7.1](#).

7.1.1.1 Interface para Utilizador

Esta camada engloba todos os componentes visíveis para o utilizador a que este pode aceder para interagir com o sistema, sendo deste modo, a única camada visível para o utilizador final. Aqui se encontram os três componentes de interface desenvolvidos para este projecto:

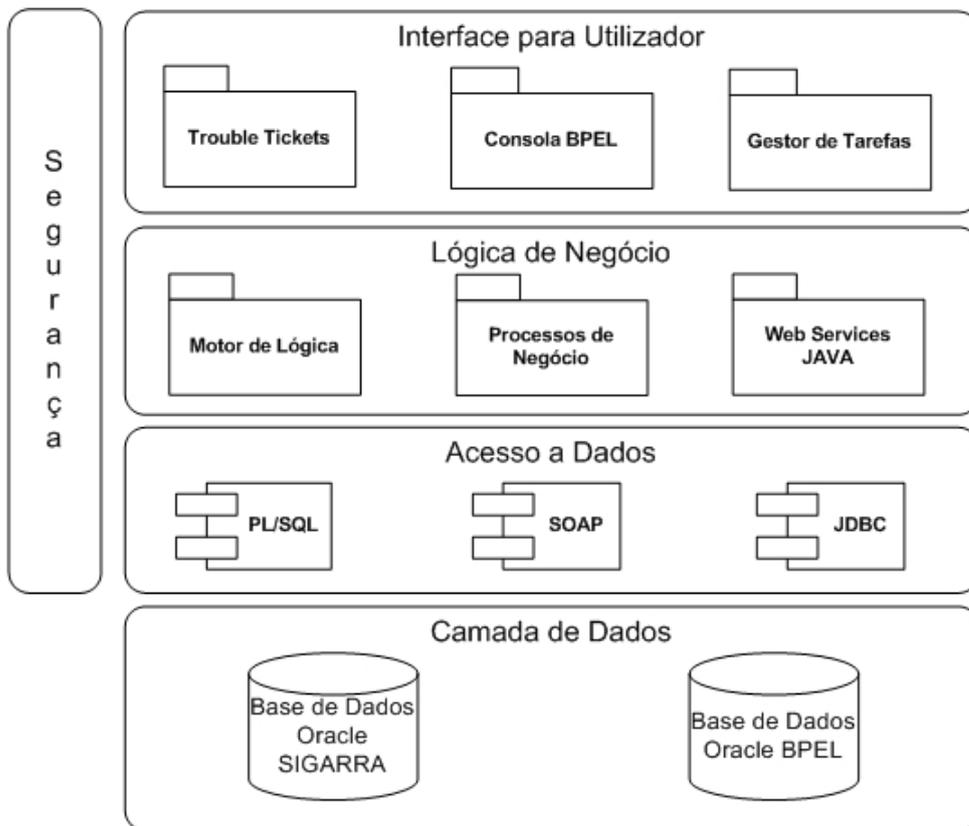


Figura 7.1: Decomposição Horizontal

- **Interfaces do módulo de Trouble Tickets:** Contém todas as páginas e formulários de utilização e interacção com as novas funcionalidades implementadas para cumprir com os requisitos propostos.
- **Interfaces da Consola BPEL:** Implementadas como um suplemento para este projecto após a conclusão de todos os objectivos do pacote de Trouble Tickets, são aqui incluídas todas as páginas e formulários para utilização pelos administradores do sistema, de modo a cumprir todos os casos de utilização especificados para o Sistema de Administração de BPEL relativos à gestão de processos e instâncias BPEL. Estas interfaces disponibilizam as mesmas funcionalidades e informações que a aplicação Oracle BPEL Console.
- **Interfaces do Gestor de Tarefas:** Implementadas igualmente como um suplemento para este projecto após a conclusão de todos os objectivos do pacote de Trouble Tickets, são aqui incluídas todas as páginas e formulários para utilização por todos os utilizadores do sistema, de modo a cumprir todos os casos de utilização especificados para o Sistema de Administração de BPEL relativos à gestão de tarefas e notificações. Estas interfaces disponibilizam as mesmas funcionalidades e

informações que a aplicação Oracle Worklist.

Toda esta camada foi desenvolvida em procedimentos PL/SQL para a Web integrados com o motor de lógica desenvolvido com a mesma tecnologia. De forma a melhorar toda a usabilidade, eficácia e rapidez destas interfaces foram igualmente integradas novas tecnologias como o Ajax e Javascript levando a uma melhoria bastante significativa de todos estes atributos. Na comunicação com a camada inferior, os seus objectivos são fornecer dados introduzidos pelos utilizadores para serem validados e processados, bem como, receber informações que serão apresentadas visualmente aos utilizadores.

7.1.1.2 Lógica de Negócio

A camada da lógica de negócio tem como objectivos disponibilizar todo o motor de negócio para os módulos do sistema de informação. Neste projecto, as suas funcionalidades aplicam regras de validação, tratamento de dados, fluxos de actividades e disponibilização de serviços. Para isto, esta camada é principalmente composta pelos seguintes componentes:

- **Motor de Lógica:** É essencialmente composto por procedimentos e funções PL/SQL que definem as regras de negócio do novo módulo de Trouble Tickets e a comunicação e interacção com a tecnologia BPEL e Serviços. Aqui se encontram implementados os pacotes de funcionalidades e regras que permitiram construir o novo módulo dos Trouble Tickets, bem como implementar os requisitos especificados pela arquitectura SIBPEL em termos de comunicação entre PL/SQL, BPEL e Web Services, criando igualmente toda a base necessária para a evolução dos restantes módulos para a tecnologia BPEL e definindo um novo conceito de arquitectura distribuída em todo o SI. As suas comunicações com os restantes componentes desta camada são efectuadas através da troca de mensagens SOAP. O acesso a dados é efectuado por comandos SQL.
- **Processos de Negócio:** Aqui se encontram todos os processos de negócio definidos em Oracle Workflow e em BPEL que complementam as funcionalidades dos módulos do SI implementados no Motor de Lógica. Foi igualmente sobre este componente, que a implementação deste projecto se apoiou bastante na evolução dos actuais processos de Workflow para BPEL. Deste modo, com a realização deste projecto, todo o processo de negócio do módulo dos Trouble Tickets foi totalmente redefinido para BPEL, estando a sua interacção com o seu motor de lógica completamente funcional e sincronizado. Os processos de negócio BPEL comunicam com os restantes componentes e com a camada de dados através de mensagens SOAP.

- **Web Services Java:** Sobre este componente foram implementados todo um conjunto de serviços que fornecem as funcionalidades necessárias aos restantes componentes desta camada. Estes foram implementados em Java, utilizando toda a API fornecida pelo Oracle BPEL Process Manager. Os seus objectivos são disponibilizar métodos de gestão da tecnologia BPEL, bem como retornar informações sobre os seus processos e instâncias. Esta disponibilização é efectuada através da troca de mensagens SOAP, sendo que o acesso aos dados do repositório BPEL é realizado por JDBC.

Com base nos componentes apresentados nesta camada, é possível identificar que este foi de facto o núcleo principal das implementações realizadas para este projecto. Os desenvolvimentos realizados sobre esta camada permitiram cumprir os objectivos de evolução do processo de negócio dos Trouble Tickets para BPEL, a criação de toda a base de comunicação entre as diversas tecnologias e a disponibilização de serviços de interacção com a tecnologia BPEL.

7.1.1.3 Acesso a Dados

Esta camada identifica os componentes necessários para interligar as duas camadas adjacentes, permitindo que a camada superior da Lógica de Negócio aceda aos dados presentes na Camada de Dados. Para isto, as comunicações entre ambas são realizadas utilizando três tecnologias diferentes:

- **SQL:** Os comandos SQL são principalmente utilizados no acesso ao repositório do SIGARRA pelo Motor de Lógica. No entanto, através da disponibilização de procedimentos e funções em PL/SQL como serviços, os processos de negócio em BPEL utilizam igualmente esta tecnologia aliada ao SOAP para realizar o acesso tanto ao repositório do SIGARRA como ao do BPEL.
- **SOAP:** Esta tecnologia é principalmente utilizada na definição de mensagens de comunicação entre os três componentes pertencentes à Lógica de Negócio. As mensagens são definidas numa estrutura XML que permite incluir os parâmetros necessários para invocar processos e serviços, bem como retornar os resultados por estes devolvidos. Quanto ao acesso a dados, esta tecnologia é essencialmente utilizada pelos Processos de Negócio em BPEL para aceder a procedimentos e funções em PL/SQL como serviços externos.
- **JDBC:** Este constitui o componente utilizado pela API em Java, usada na implementação dos Web Services da camada Lógica de Negócio, para o acesso e manipulação dos dados contidos no repositório de BPEL.

7.1.1.4 Camada de Dados

A Camada de Dados representa os dois repositórios de dados utilizados pelo sistema implementado neste projecto. Aqui se encontram todas as informações necessárias para o funcionamento dos módulos do SIGARRA e da tecnologia BPEL. Para isto, esta camada é constituída por um servidor applicacional Oracle 10g que define dois repositórios de dados, um relativo às tabelas, procedimentos e funções do SI, onde se inclui o novo modelo relacional para o módulo dos Trouble Tickets, e outro que implementa o modelo relacional apresentado na figura 3.10 com os dados relativos aos processos e instâncias definidas no servidor de BPEL.

7.1.1.5 Segurança

Em todas as camadas apresentadas são consideradas medidas de segurança para garantir o bom funcionamento do sistema e a total protecção e confidencialidade de todo o fluxo de dados gerado e de todos os serviços e funcionalidades disponibilizadas. Por exemplo, através de validações de dados e aplicação de restrições bem como da utilização de componentes relativos à qualidade dos serviços.

7.1.2 Decomposição Vertical

Nesta secção é apresentado um modelo mais funcional da arquitectura do sistema com base num diagrama de pacotes vertical, que permite visualizar as interacções entre os diversos componentes e módulos do sistema.

Visualizando o diagrama apresentado na figura 7.2, a todas as interfaces gráficas desenvolvidas está associado o respectivo motor de lógica que implementa as suas funcionalidades e aplica as regras necessárias. Quanto à lógica do novo módulo de Trouble Tickets este comunicará com o seu processo de negócio em BPEL através da geração, envio e recepção de mensagens SOAP. Por outro lado, irá igualmente efectuar acessos e manipulações de dados directamente sobre o seu modelo relacional definido na base de dados do SIGARRA, onde acede igualmente o processo BPEL.

Relativamente às interfaces de administração, estas são constituídas por dois motores de lógica que implementam a lógica da Consola BPEL e do Gestor de Tarefas. Ambos acedem às informações dos processos, instâncias e tarefas do BPEL através dos Web Services implementados para este projecto, que utilizam a API de Java fornecida de modo a aceder e controlar os dados contidos no repositório do servidor de BPEL. Os acessos a estes serviços são efectuados também através da troca de mensagens SOAP, que por sua vez acedem ao repositório de BPEL por JDBC. Os processos de negócio BPEL interagem igualmente com este repositório nomeadamente para implementar a funcionalidade Dehydration.

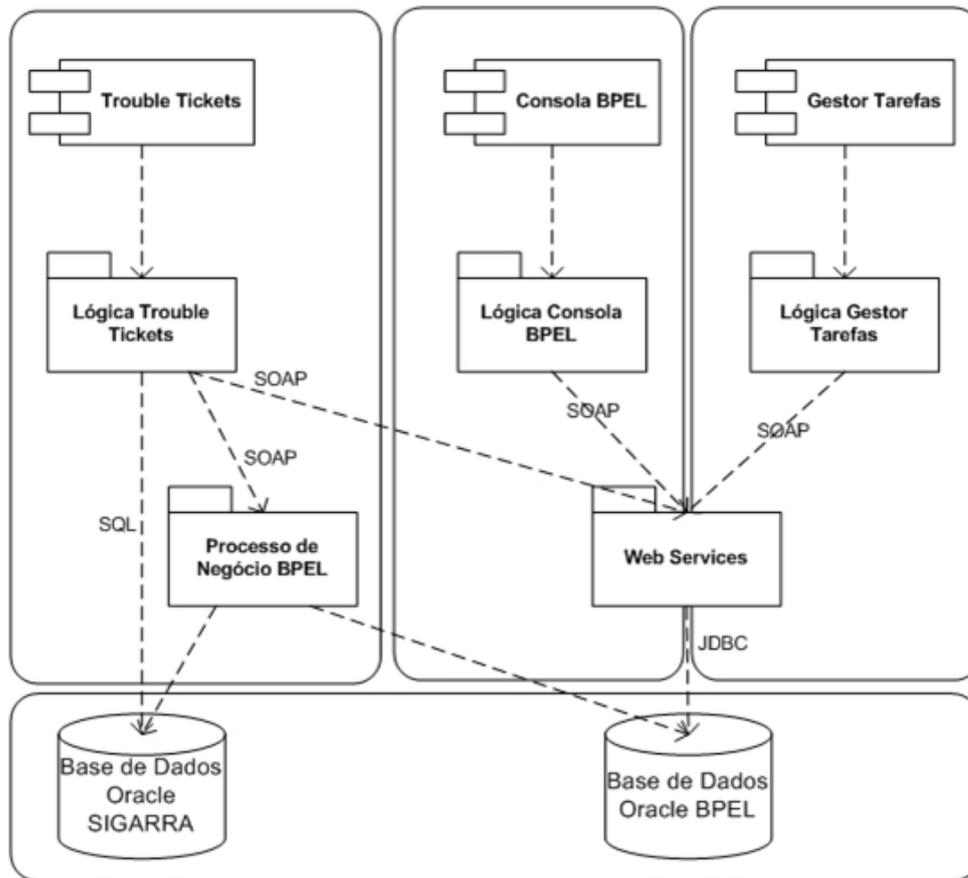


Figura 7.2: Decomposição Vertical

7.2 Arquitectura Física do Sistema

No que respeita à distribuição física dos componentes do sistema, esta permite capturar a topologia de hardware constituinte, os diversos componentes e módulos de software utilizados pelo sistema e as respectivas localizações. Para exemplificar a arquitectura física do sistema, foi elaborado um diagrama de distribuição que, conjuntamente com um diagrama de componentes UML, permite capturar toda a estrutura física da implementação, sendo este apresentado na figura 7.3.

Neste diagrama estão representados os componentes distribuídos por apenas dois servidores aplicativos. No entanto, dada a especificação da arquitectura SIBPEL, toda esta arquitectura implementada poderia facilmente estar distribuída em servidores adicionais, bem como utilizar servidores com sistemas diferentes, devido à facilidade do Oracle BPM de integração com diferentes servidores.

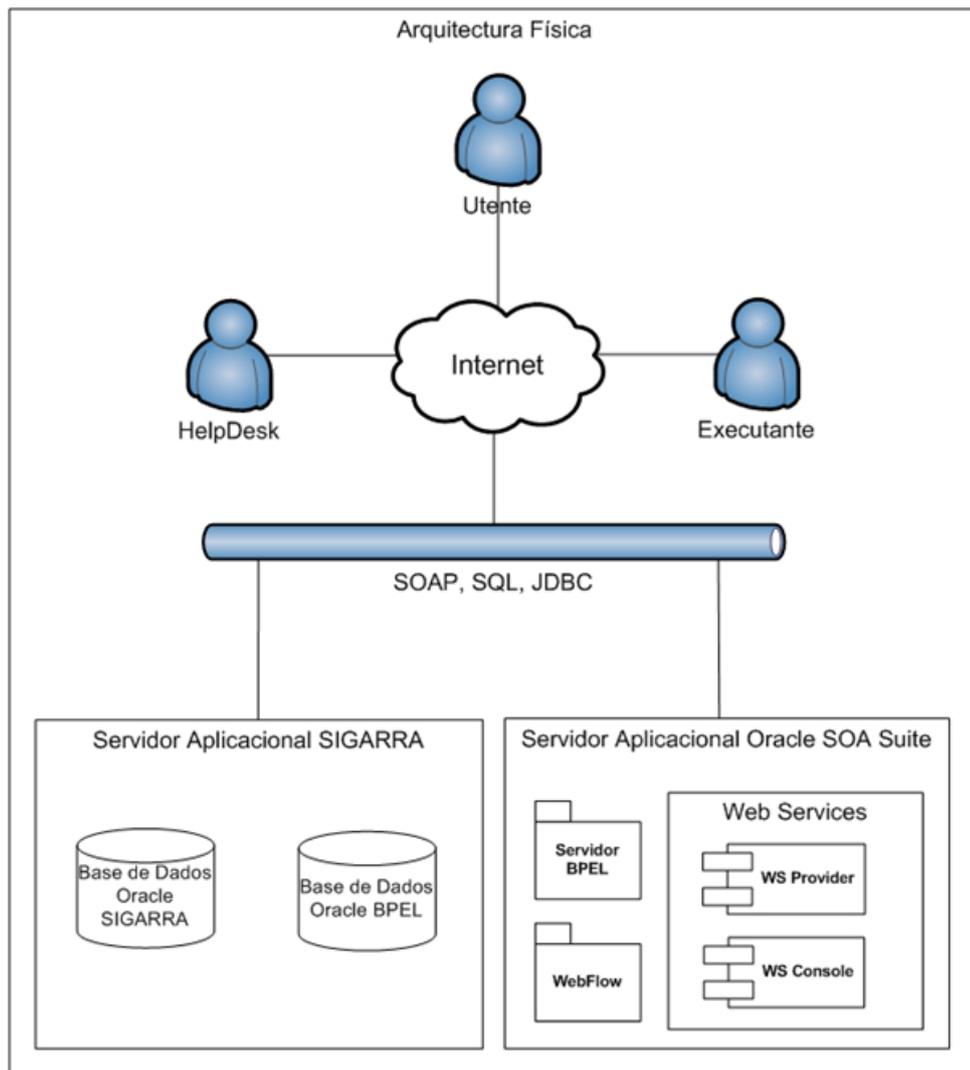


Figura 7.3: Arquitectura Física

7.3 Sistema SIBPEL

A implementação do Sistema SIBPEL neste projecto representa a criação de toda a base necessária para integração e comunicação do Sistema de Informação com a tecnologia BPEL. Para a realização deste feito, existiam inicialmente três possibilidades de comunicação:

- Através de consultas SQL directamente do repositório de BPEL.
- Incluindo os métodos JAVA disponibilizados pela API do Oracle BPM e utilizar estas classes a partir do PL/SQL.

- Implementar Web Services em JAVA que utilizam as classes e métodos desta API e comunicar com estes serviços a partir do PL/SQL através de mensagens SOAP.

A primeira solução a ser testada foi o acesso directo ao repositório de BPEL. No entanto e como já foi referido anteriormente, todo o seu esquema relacional é bastante confuso e a sua manipulação é de elevada complexidade. Contudo, no estudo realizado sobre esta solução, foram identificadas as seguintes tabelas principais que contêm informações vitais para o funcionamento do motor do servidor de BPEL e cujos registos são possíveis de serem consultados por comandos SQL:

- **Process:** Contém os detalhes principais dos processos registados no servidor, tais como o seu identificador, versão, nome e o seu estado. A partir dos processos identificados nesta tabela é possível iniciar várias instâncias para o mesmo processo.
- **Cube_Instance:** Inclui os detalhes principais relativo a todas as instâncias registadas no servidor, tanto as que estão activas como as terminadas. As instâncias estão directamente relacionadas com os processos BPEL, processando o fluxo de actividades por eles definido. Dos seus atributos destacam-se o seu identificador, criador, última data de modificação e a data de criação, versão, prioridade e o estado actual.
- **WFTask:** Esta representa a tabela mais extensa de todo o modelo relacional. Aqui são guardadas as informações relativas às tarefas enviadas pelas diversas instâncias aos respectivos utilizadores do sistema. Todo o histórico das tarefas desactualizadas é mantido na tabela WFTaskHistory que é idêntica à WFTask. Dos seus atributos destacam-se o identificador, nome, título, o identificador da instância BPEL a que pertence, criador, resposta enviada, receptores e diversas datas de modificação e criação.
- **Variable_Sensor_Value:** Contém os valores dos sensores definidos em cada instância BPEL. Estes sensores podem ser utilizados para identificar alterações ao estado ou às variáveis utilizadas numa dada instância. Os seus atributos principais são o nome do sensor, identificador do processo e da instância a que pertence o sensor e o valor respectivo do sensor. Caso um sensor possua vários valores, então são inseridos nesta tabela vários registos alterando apenas o seu atributo valor.

Para além destes factos que impediram um controlo directo e total do repositório de BPEL, esta impossibilidade permitiu recorrer à utilização da API fornecida o que possibilita evitar futuras modificações quando forem disponibilizadas novas versões, particularmente na evolução para a versão 11g, em que provavelmente muitas destas tabelas e atributos tenham sofrido grandes alterações.

Excluída esta solução de acesso ao repositório de BPEL, ficou então estabelecido o acesso por métodos da API em Java que é fornecida pelo Oracle BPM. Contudo, esta utilização poderia ser ainda efectuada de duas formas. De modo a testar a melhor solução, foi então tentada a integração das classes e objectos da API de Java no repositório do SIGARRA, que seriam depois utilizados directamente a partir de procedimentos e funções em PL/SQL. No entanto, na tentativa de incluir estas classes no repositório, ocorreu de novo um problema que impedia a sua realização. Começando inicialmente por incluir apenas as classes principais, foi-se concluído que na medida em que iam sendo incluídas novas classes de modo a eliminar dependências, estas originavam ainda mais dependências. Ou seja, muitas vezes duas classes não compilavam, pois dependiam entre si e entre diversas outras classes não disponibilizadas publicamente pelo Oracle BPM. Dado o elevado número de erros de dependências que eram geradas com o aumento das classes necessárias, conclui-se que esta solução originou a criação de dependências recursivas, pelo que a sua implementação não é igualmente possível.

Dadas as impossibilidades de implementação das duas primeiras soluções, partiu-se então para a implementação da terceira e última solução, o acesso aos Web Services que implementam a API de Java.

Para testar a possibilidade de utilização desta solução no Sistema de Informação foi implementado uma classe Java que importa as classes e métodos da API, para retornar simplesmente alguns detalhes de um processo BPEL. De forma a registar esta classe no servidor de BPEL e disponibilizar este serviço para o SI, foram testadas diversas formas de o registar como um Web Service na Lista de Registos do UDDI. Para isto, foram utilizadas funcionalidades do Eclipse e do Net Beans para este efeito. No entanto, nenhum deles conseguia registar o serviço no servidor aplicacional. Por fim, descobriu-se uma funcionalidade bastante interessante no Oracle JDeveloper, que permite compilar e registar uma classe Java como um serviço em qualquer servidor aplicacional, à semelhança do que acontece com o registo de um processo BPEL no servidor. Com este feito, estava então testada e aprovada que a disponibilização dos métodos da API em Java era possível através de Web Services, concluindo metade da implementação desta solução.

Para completar este processo, era agora necessário aceder a este Web Service a partir do Sistema de Informação, nomeadamente a partir de procedimentos e funções em PL/SQL. Para isto, foi testada a utilização do pacote UTL_HTTP disponibilizado pela Oracle. Este disponibiliza métodos que permitem aceder a dados através do protocolo Hypertext Transfer Protocol (HTTP) comunicando, desta forma, com servidores Web em HTTP. O meio de comunicação utilizado é através da construção e envio de mensagens no formato SOAP. Estas mensagens são caracterizadas por um código XML que identifica as definições de acesso aos Serviços, a respectiva operação que se pretende invocar e os parâmetros que lhes são passados. Os seus principais métodos que foram utilizados neste projecto são:

- **begin_request**: Define o início de uma ligação com o endereço do serviço, o modo de acesso e o protocolo de ligação.
- **set_header**: Permite definir os cabeçalhos da mensagem SOAP, incluindo a operação que irá ser invocada no Serviço.
- **write_text**: Envia a mensagem SOAP.
- **get_response**: Espera e recebe a resposta do serviço.
- **end_response**: Termina a conexão.

Testado o funcionamento destes métodos na comunicação com o Web Service criado, recebendo igualmente a resposta por ele retornado, foi terminado com sucesso o teste à terceira solução.

Obtendo então uma solução fiável para a criação do sistema SIBPEL, a sua implementação teve início com a definição do pacote SOAP, contendo procedimentos e funções em PL/SQL, com a capacidade de gerar automaticamente mensagens no formato SOAP, realizar comunicações com os Serviços e decodificar as respostas por eles retornado através da utilização do pacote UTL_HTTP. Para isto foram então criados os seguintes métodos:

- **New_request**: Cria a estrutura necessária para iniciar um novo pedido de ligação a um Serviço;
- **Add_parameter**: Permite adicionar um parâmetro ou uma variável à mensagem SOAP, necessária para a invocação de uma operação do Serviço;
- **Invoke**: Invoca uma operação do Web Service esperando pela sua resposta;
- **Invoke_No_Resp**: Invoca uma operação do Web Service mas não espera pela sua resposta, tornando a conclusão da comunicação mais rápida.
- **Get_return_xml**: Devolve toda a mensagem SOAP retorna pelo Web Service em formato XML. Deverá ser utilizado nos casos de retorno de vários valores ou estruturas de dados.
- **Get_return_value**: Utiliza a tecnologia XPATH para decodificar o formato XML da mensagem SOAP recebida, de forma a retornar numa string apenas o valor de saída da operação invocada. No entanto, apenas deverá ser utilizado quando apenas é devolvido um valor.

A mensagem SOAP apresentada de seguida corresponde a um exemplo de utilização deste pacote na geração desta simples mensagem, passando apenas o endereço do serviço, a operação a invocar e os respectivos parâmetros de entrada.

Projecto e Implementação

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns1="http://provider.fe.up.pt/">
    <ns1:executeBatchTaskElement>
      <ns1:String
        xmlns:ns2="http://provider.fe.up.pt/types/">
        <ns2:taskId>XYZ</ns2:taskId>
        <ns2:user>030509117</ns2:user>
        <ns2:action>OK</ns2:action>
        <ns2:realm>FEUP</ns2:realm>
      </ns1:String>
    </ns1:executeBatchTaskElement>
  </soap:Body>
</soap:Envelope>
```

Após a invocação de uma determinada operação de um serviço, este devolve sempre uma mensagem SOAP de resposta, contendo uma estrutura XML com os diversos valores por este retornados, como é apresentado de seguida.

```
<env:Envelope
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns0="http://provider.fe.up.pt/"
  xmlns:ns1="http://provider.fe.up.pt/types/"
  xmlns:ns2="http://www.oracle.com/webservices/internal/literal">
  <env:Body>
    <ns0:getProcessResponseElement>
      <ns0:result>
        <ns1:processId>TTS_TICKETS</ns1:processId>
        <ns1:wsdlInfo>
          ../TTS_TICKETS/1.1/TTS_TICKETS?wsdl
        </ns1:wsdlInfo>
        <ns1:state>Activo</ns1:state>
        <ns1:revisionTag>1.1</ns1:revisionTag>
      </ns0:result>
    </ns0:getProcessResponseElement>
  </env:Body>
</env:Envelope>
```

Obtendo esta estrutura, será apenas necessário utilizar métodos XPATH para descodificar os diversos nós que compõem a mensagem de resposta.

Após a implementação deste pacote de procedimentos que realizam a comunicação directamente com os serviços disponibilizados exteriormente, surgiu a necessidade de generalizar estes acessos e tornar toda esta implementação transparente para o Sistema de Informação. Para isto, foi implementado o pacote WEB_SERVICES, contendo procedimentos e funções onde apenas é passado o endereço do Web Service, a operação a invocar e os parâmetros de entrada. Das funcionalidades implementadas destacam-se:

- **Invoca_WebService_Cursor:** Permite invocar um Serviço definindo dinamicamente os seus parâmetros num cursor. Isto possibilita, por exemplo, invocar serviços com diversos registos obtidos através de uma consulta à base de dados. A estrutura deste cursor está definida por um conjunto nome, tipo, valor representando unicamente um parâmetro.
- **Invoca_WebService:** Invoca um Serviço com os parâmetros passados num CLOB. Esta solução é mais fiável quando o serviço recebe parâmetros definidos em vectores, o que impossibilita a sua definição num cursor;
- **Invoca_WebService_Resp_XML:** Disponibilizado igualmente na versão Cursor e CLOB, realiza as mesmas operações dos métodos anteriores, mas retorna a resposta obtida no tipo de dados XMLTYPE. Este procedimento foi implementado, devido à necessidade de retornar em diversos Serviços, grandes conteúdos de dados em XML.
- **Invoca_WebService_No_Resp:** Idêntico aos anteriores mas não espera pela recepção da mensagem de resposta do serviço, aumentando a rapidez da comunicação mas não validando a correcta invocação do serviço. Este método deverá ser apenas utilizado quando se pretende apenas realizar operações e não obter informações. No entanto deverá sempre ser efectuada uma análise por outros meios da correcta invocação deste serviço.

Com a implementação deste pacote de procedimentos e funções, todos os acessos ficam bastante mais simples e transparentes para serem disponibilizados ao Sistema de Informação e ao pacote BPEL. Para isto, é apresentado de seguida o simples código que o Sistema de Informação deverá utilizar de modo a invocar a operação executeBatchTaskElement do Web Service BPELProvider, passando os parâmetros definidos num CLOB params.

```
params := '<ns1:String_1
          xmlns:ns2="http://provider.fe.up.pt/types/">
          <ns2:taskId>XYZ</ns2:taskId>
```

Projecto e Implementação

```
<ns2:user>030509117</ns2:user>
<ns2:action>OK</ns2:action>
<ns2:realm>FEUP</ns2:realm>
</ns1:String_1>';
resposta := WEBSERVICES.INVOCA_WEBSERVICE_RESP (
    './BPEL-BPELTasks-context-root/
    BPELProviderSoapHttpPort',
    'xmlns:ns1="http://provider.fe.up.pt/"',
    'ns1:executeBatchTaskElement',
    params);
```

Para completar todo o conjunto de funcionalidades disponibilizadas neste sistema SIBPEL, foi ainda implementado o pacote BPEL. Criado inicialmente para realizar as funcionalidades do pacote WEB_SERVICE antes de este existir, este pacote acabou por ficar apenas com um método de acesso exclusivo ao BPEL, neste caso para iniciar directamente uma instância de um processo BPEL com o envio de uma mensagem SOAP, realizando a mesma funcionalidade disponibilizada pelo Invoca_WebService, focado apenas em invocar Web Services que na verdade são processos de negócio definidos em BPEL mas disponibilizados como Serviços.

No entanto o pacote BPEL é ainda constituído por diversos procedimentos que implementam a lógica de negócio da comunicação com a tecnologia BPEL, sendo principalmente responsáveis por receber informações e pedidos do SI e invocar os respectivos Serviços através do pacote WEB_SERVICES. Com a criação deste pacote, torna-se totalmente transparente para o Sistema de Informação quais os endereços dos Serviços, as operações invocadas e os esquemas dos parâmetros utilizados. Este apenas necessita de invocar uma determinada função do pacote BPEL, responsável pelo acesso a um determinado serviço, com os respectivos parâmetros passados como variáveis de entrada, e esta função trata de tudo o resto incluindo o respectivo retorno da resposta recebida.

7.4 Estrutura de Serviços

É apresentada nesta secção a lista de operações e funcionalidades implementadas segundo os métodos e classes disponibilizados pela API em Java do Oracle BPEL Process Manager.

Este conjunto de serviços divide-se em dois Web Services: o BPELProvider e o BPELConsole. O primeiro é o responsável por disponibilizar funcionalidades que permitam a interação e sincronização dos fluxos BPEL com os módulos do SIGARRA. Quanto ao segundo, pertence às implementações suplementares desenvolvidas para este projecto e disponibiliza serviços para a Consola e para o Gestor de Tarefas do Sistema de Informação. Todos estes serviços são invocados por processos BPEL e pelo novo módulo dos Trouble Tickets através da troca de mensagens XML no formato SOAP.

Tabela 7.1: Lista de Serviços do Web Service BPELProvider

Serviços BPELProvider	
Operação	Descrição
<code>deleteInstance</code>	Permite apagar de todo o sistema tudo o que seja relacionado com uma dada instância passada como parâmetro de entrada.
<code>getSensorValues</code>	Obtém os valores de todos os sensores de uma dada instância.
<code>executeBatchTask</code>	Possibilita responder a uma dada tarefa, passando o seu identificador, o utilizador que responde e o valor da resposta. Serviço de extrema importância para a sincronização das instâncias BPEL com o Sistema de Informação.
<code>getInstance</code>	Retorna todas as informações relativas a uma instância, tais como o seu nome, estado, identificador, data de criação e modificação.
<code>getUserParticipants</code>	Utilizado na integração dos utilizadores da FEUP com o BPEL. Permite retornar a lista de utilizadores dado o identificador do grupo a que pertencem. Este serviço é utilizado pelos processos BPEL, na criação de Tarefas Humanas para múltiplos utilizadores.
<code>startProcess</code>	Inicia uma nova instância BPEL da última versão registada no servidor para um dado processo. São igualmente passados, em formato XML, os parâmetros de entrada do processo.
<code>cancelTask</code>	Permite cancelar uma dada tarefa pendente para um utilizador. Para isto foi definido um utilizador sistema genérico que é o dono e o responsável por todas as tarefas e notificações enviadas no BPEL. Deste modo, é possível que este efectue um pedido para cancelar uma dada tarefa sem a intervenção dos destinatários.

Tabela 7.2: Lista de Serviços do Web Service BPELConsole utilizados pela Consola BPEL

Serviços BPELConsole	
Operação	Descrição
getProcessNames	Obtém a lista de nomes dos processos BPEL registados no servidor. Caso existam múltiplas versões do mesmo projecto, é igualmente retornada a sua versão.
deleteAllSensors	Apaga todos os registos e valores dos sensores criados por todas as instâncias.
deleteAllInstances	Apaga todas as instâncias do sistema independentemente do seu estado.
undeployProcess	Permite remover um dado processo da lista de processos activos no servidor.
getInstanceTrace	Desenvolvido para retornar o Instance Trace de uma instância. Este corresponde a uma estrutura XML contendo todas as actividades processadas num dado fluxo de uma instância e as respectivas ligações.
getAuditTrail	Retorna uma grande estrutura XML que contém os detalhes e valores de todas as actividades processadas num dado fluxo.
getTotalInstancias	Devolve o total de instâncias activas ou terminadas de um dado processo BPEL. Usado para efectuar a paginação dos resultados.
getTotalProcessos	Devolve o total de processos registados no servidor BPEL. Usado para efectuar a sua paginação.
initiateProcess	Possibilita iniciar uma nova instância para um dado processo, recebendo um conjunto de parâmetros nome e valor, que correspondem aos parâmetros de entrada de um dado processo BPEL.
getProcessForm	Gera um código html que é inserido no formulário para iniciar um processo, com o nome e tipos de todos os parâmetros de entrada desse processo. Permite posteriormente invocar o serviço <code>initiateProcess</code> passando esta lista de valores.
deleteInstance	Permite apagar de todo o sistema tudo o que seja relacionado com uma dada instância passada como parâmetro de entrada.
abortInstance	Permite cancelar o processamento de uma dada instância, mantendo todo o seu histórico no repositório.
getInstance	Retorna todas as informações relativas a uma instância, tais como o seu nome, estado, identificador, data de criação e modificação.
getProcess	Retorna todas as informações relativas a um dado processo BPEL registado no servidor, tais como o seu nome, versão, o total de instâncias activas e terminadas.
getInstanceByStatus	Devolve todas as instâncias que verifiquem o conjunto de filtros parametrizados neste Serviço. Deste modo é possível filtrar instâncias pelo seu estado e processo e ainda definir a sua paginação.

Tabela 7.3: Lista de Serviços do Web Service BPELConsole utilizados pelo Gestor de Tarefas

Serviços BPELConsole	
Operação	Descrição
<code>getTasks</code>	Permite retornar todo o conjunto de tarefas de um determinado utilizador passando o seu código. Permite ainda filtrar os resultados por processo, por instância e por nome da tarefa e ainda aplicar uma paginação aos resultados retornados.
<code>getTotalUserTasks</code>	Retorna o total de tarefas e notificações terminadas ou pendentes para um dado utilizador. Permite efectuar a paginação destes resultados.
<code>executeBatchTask</code>	Possibilita responder a um grupo de tarefas, passando os seus identificadores, o utilizador que responde e os respectivos valores das respostas.

7.5 O Novo Trouble Tickets

Após a implementação de todo o Sistema SIBPEL e de alguns Serviços essenciais, foi então possível iniciar o processo de migração do módulo de Trouble Tickets na versão em Oracle Workflow para BPEL. Toda a construção e redefinição deste novo módulo foram apoiadas em todos os conceitos apresentados nos capítulos anteriores. Deste modo, todo o seu processo de desenvolvimento e implementação foi repartido em quatro fases, que serão devidamente detalhadas nos seguintes pontos.

7.5.1 Definição do Processo de Negócio BPEL

A definição do processo de negócio BPEL para o novo módulo dos Trouble Tickets, marcou o início da evolução deste novo sistema e constitui deste modo a primeira fase de toda a sua implementação. Analisando todo o fluxo de actividades executadas na sua versão em Oracle Workflow, deu-se início ao processo de evolução de Workflow para a tecnologia BPEL. Seguindo os passos essenciais, apresentados no capítulo 5 para a realização deste mapeamento, aliado à análise dos novos requisitos e casos de utilização propostos na especificação deste novo módulo, foi então elaborado o diagrama de actividades UML que pode ser visualizado no Anexo A permitindo representar o fluxo de actividades que o novo processo de negócio deste módulo possui.

Nesta definição foram ainda introduzidas novas actividades e conceitos em todo o fluxo do processo. A sua diferença mais significativa é a repartição de todo este fluxo em dois processos distintos. No processo principal, está implementado o fluxo relativo a um Ticket, identificado como um problema no sistema em Workflow, enquanto que o novo fluxo de uma Tarefa está definido num novo processo separado do principal, que implementa as actividades necessárias para processar o envio de uma determinada tarefa para um utilizador, possibilitando a sua reutilização em futuros processos de negócio que

incluam a utilização de tarefas. Esta repartição permite tirar partido de todas as vantagens da tecnologia BPEL no cumprimento dos conceitos da Arquitectura SOA, nomeadamente na reutilização de serviços e na redução da complexidade do sistema.

O diagrama apresentado introduz ainda alguns conceitos novos, como a distinção entre tarefas (decisões) e notificações, a possibilidade de agendar Tarefas e de reabrir um Ticket. A invocação de um novo processo Tarefas poderá igualmente ser efectuada a partir de qualquer processo de negócio BPEL.

Relativamente aos estados de um problema, estes sofreram igualmente alterações. Enquanto que na versão Workflow apenas existiam três estados, nesta nova versão em BPEL o seu conjunto de estados internos foi aumentado, tanto a nível dos Tickets como das Tarefas. Para ilustrar as variações existentes nesta versão, é apresentado na figura 7.4 o seu diagrama de estados UML.

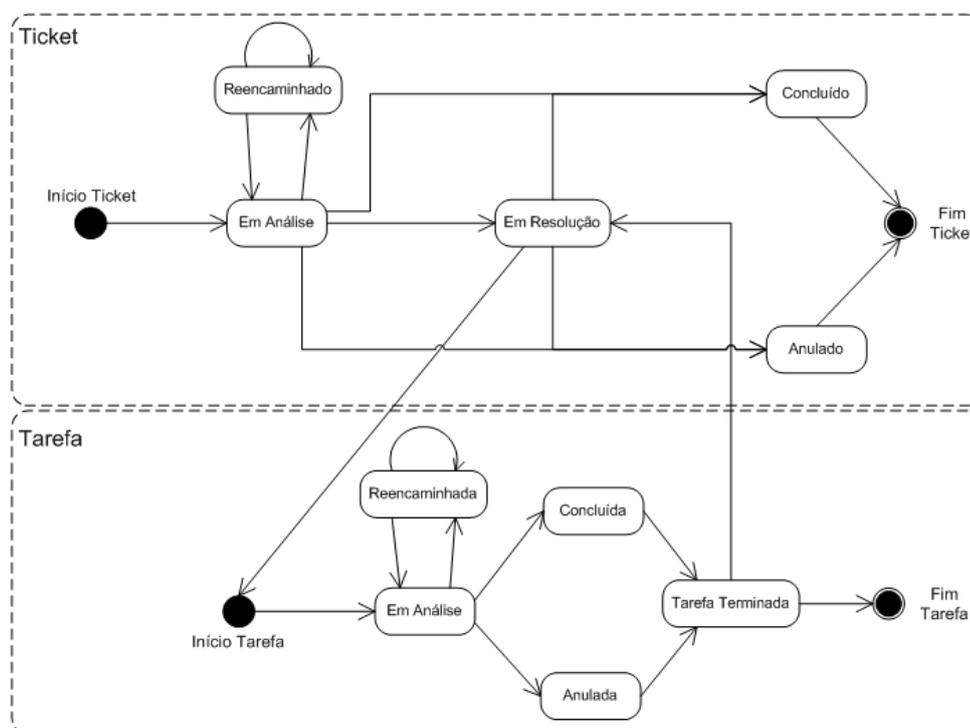


Figura 7.4: Diagrama de Estados dos Tickets e Tarefas

Como é possível identificar neste diagrama, ambos os Tickets e as Tarefas possuem cinco estados possíveis. No entanto, relativamente a um Ticket, destacam-se como os dois estados principais o Em Análise e o Em Resolução. O primeiro estado representa as actividades que são efectuadas para analisar o problema identificado e tomar as medidas necessárias para a sua resolução. Após estes passos, o Ticket passa ao segundo estado, onde são iniciadas Tarefas para os respectivos executantes. Quanto às Tarefas, estas apenas possuem um estado de análise e resolução em simultâneo.

Em ambos os fluxos é possível reencaminhar as respectivas análises para outros HelpDesks ou utilizadores e terminar o fluxo, concluindo ou anulando a sua resolução ou análise. No entanto, no último estado de uma Tarefa é verificado se todas as restantes Tarefas do Ticket estão terminadas podendo, apenas em caso afirmativo, concluir igualmente o Ticket.

Obtendo os dois diagramas apresentados e tendo uma total compreensão dos objetivos que ambos os fluxos devem implementar, aliado a um conhecimento de todas as actividades e conceitos da tecnologia BPEL, foi então possível converter este esquema para um processo BPEL. No entanto, para atingir o modelo final do processo de negócio deste módulo, foi necessário realizar várias reformulações no desenho do fluxo BPEL, de modo a torna-lo totalmente funcional, tanto a nível das suas restrições como a nível de integração com os utilizadores da Faculdade de Engenharia. De notar igualmente que todas as normas de conversão apresentadas no ponto 5.2 representam um conjunto de soluções obtidas durante a realização desta conversão, nomeadamente na resolução dos diversos contratempos que foram surgindo. No Anexo A são apresentados os dois processos de negócio em BPEL implementados para os Tickets e Tarefas, que são utilizados nesta nova versão do módulo dos Trouble Tickets.

A definição destes dois processos começou através do cumprimento de simples etapas, evoluindo para etapas mais complexas à medida que estas iam sendo cumpridas. Para descrever todo este processo de definição, apresentam-se de seguida os principais desafios que foram surgindo e que permitiram obter os dois fluxos finais apresentados e especificar todo o conjunto das normas de conversão apresentadas.

1. O desafio inicial foi tentar parar o fluxo de um processo e controlar o momento exacto em que este deveria avançar. Com as actividades do BPEL este poderia ser realizado através de um ciclo de repetição enquanto algo não aconteça, ou então através de uma tarefa para um utilizador. Foi tentada então a solução de implementação de um ciclo que verifica quando um determinado registo da base de dados é actualizado. No entanto, esta solução produz um excesso de carga no processador, pois o BPEL fica constantemente a verificar se o registo foi actualizado. Deste modo, optou-se pela definição de uma tarefa para um determinado utilizador ou serviço que é o responsável por actualizar o estado desta tarefa e avançar no processo. Graças a tecnologia Dehydration, todo o processador é libertado enquanto a tarefa não receber uma resposta.
2. O segundo desafio foi tentar encontrar um meio de apresentar textualmente qual a lista de actividades que tinham sido processadas pelo fluxo até um dado momento. Dado que a API não fornece nenhum método com estas funcionalidades, a solução encontrada foi utilizar um sensor global a todo o processo, que apresenta todas as alterações realizadas sobre uma variável Estado_Processo. Com este esquema, apenas foi necessário actualizar esta variável com uma mensagem informativa sobre

a posição actual do processo sempre que este avançava o seu fluxo. Por fim, para ter acesso a todos os passos processados no fluxo, basta apenas aceder ao relatório das alterações disponibilizadas pelo sensor.

3. Um outro desafio consistiu na tentativa de encontrar uma forma de implementar tarefas e notificações no processo. Uma tarefa representa uma decisão no processo e deve por isso esperar que seja recebida uma resposta. Por outro lado, uma notificação representa apenas um aviso que deve ser enviado ao utilizador, mas de modo a continuar o fluxo do processo. A solução encontrada foi a remoção da actividade Receber, que é automaticamente associada aquando da criação de uma Tarefa Humana. Deste modo, uma notificação é processada de igual modo que uma tarefa, mas dado que a sua resposta não influenciará o fluxo do processo, este não deverá esperar pela sua recepção. A resposta por defeito que será utilizada para uma notificação será a “Tomei Conhecimento”.
4. Um dos desafios mais importantes foi encontrar um meio de poder voltar atrás no fluxo. Dado que o BPEL não permite directamente que isto seja realizado, foram definidas duas soluções. A primeira consiste na criação de um ciclo que permite repetir todo o processamento de um conjunto de actividades quando uma determinada condição for verificada. Deste modo, esta condição poderia ser a repetição do fluxo enquanto, por exemplo, a resposta a uma tarefa for Repetir. No entanto, existe igualmente a possibilidade de realizar este objectivo através do tratamento de excepções. Para isto, uma excepção poderia ser activada, por exemplo, recebendo a mesma condição referida para o ciclo.
5. De modo a possibilitar o envio de tarefas e notificações para um Grupo de Utilizadores da FEUP, foi criado um procedimento em PL/SQL que devolve ao BPEL os utilizadores pertencentes a um dado grupo. Depois na definição da tarefa foi apenas necessário incluir a sua resposta como a definição dos utilizadores que a iriam receber.
6. Ainda relativamente a tarefas e notificações, era necessário posteriormente distinguir quando é que um utilizador deveria responder a partir do gestor de tarefas, ou a partir da interface gráfica do próprio módulo. Por exemplo, quando é enviada uma tarefa de análise de um Ticket para um HelpDesk, esta irá aparecer no gestor de tarefas. No entanto, esta deverá somente ser respondida através da própria interface do Trouble Tickets, de modo a possibilitar associar outras informações com esta resposta, como por exemplo qual é o novo HelpDesk a que se destina um Reencaminhamento. Para a realização deste objectivo, foi tentado incluir um parâmetro na própria tarefa que identifica quando é que esta poderá ou não ser respondida pelo gestor de tarefas. Contudo, devido a um bug existente na IDE do BPEL Designer

na versão 10g, este não permite adicionar correctamente estes parâmetros. Não obstante, este problema já foi resolvido na versão 11g. Dados estes factos, a solução encontrada foi realizar esta diferenciação através do nome da tarefa.

7. Um outro objectivo, de modo a implementar o novo módulo dos Trouble Tickets, seria que em diversas situações fosse o próprio sistema a responder a algumas tarefas iniciadas pelo processo, de modo a generalizar o seu controlo a diversos utilizadores que interagem com um Ticket, sem que estes recebessem tarefas ou notificações. A solução encontrada foi criar um único utilizador Sistema no BPEL, que será o responsável por responder a este tipo de tarefas.
8. Por fim, existia a necessidade de eliminar certas notificações que estavam pendentes nos utilizadores mas que já não faziam sentido no contexto actual do fluxo do processo. Visto que o BPEL não tem acesso às tarefas e notificações que enviou, a solução encontrada foi chamar um Web Service que utiliza a sua API em Java, para terminar um conjunto de tarefas relativas a uma dada instância BPEL. No entanto, surgiu mais o contratempo de que o BPEL não permite chamar serviços com parâmetros em vectores, onde iria o conjunto de notificações que seriam terminadas. Para isto, foi alterada a implementação do serviço de modo a receber num único parâmetro de texto todos os nomes das notificações separadas por vírgulas.

Com o cumprimento de todas estas etapas aliadas às soluções implementadas para superar todos os obstáculos encontrados, o novo processo BPEL dos Trouble Tickets cumpre na sua totalidade todos os objectivos propostos e possibilitou ainda a introdução de novos conceitos, que tiram o máximo partido da utilização desta tecnologia e da implementação de um sistema distribuído seguindo os conceitos e objectivos especificados pela Arquitectura SOA.

7.5.2 Detalhes da Integração do Módulo com BPEL

Neste tópico são apresentadas algumas decisões de implementação que foram tomadas de modo a obter uma total sincronização de operações entre o motor de lógica implementado em PL/SQL e a tecnologia BPEL.

Uma das primeiras decisões tomadas esteve relacionada com a invocação das respectivas instâncias BPEL para os Tickets e Tarefas. Compreendendo que havia a necessidade de ambos comunicarem mutuamente, foi então necessário guardar os identificadores das respectivas instâncias em cada Ticket e Tarefa que fossem criados, de forma a permitir aceder unicamente à instância correcta. Estes identificadores são retornados e actualizados pelo próprio processo de negócio.

Ainda relativamente à invocação de novas instâncias, visto que o BPEL permite o registo de várias versões do mesmo processo de negócio, foi também necessário utilizar

o serviço `startProcess` que passando apenas o nome do processo e os seus parâmetros de entrada, este é responsável por invocar sempre uma nova instância do processo com a versão mais recente e retornar como resposta o identificador da instância que foi iniciada.

No que respeita à sincronização e comunicação do processamento dos Tickets e das Tarefas em todo o motor de lógica deste módulo com os respectivos processos de negócio em BPEL, foram igualmente tomadas importantes decisões a nível da sua implementação.

Sabendo que todo o processamento deste módulo será controlado pelos utilizadores envolventes, que realizam as suas acções e decisões sobre a interface gráfica deste módulo, e ainda que a única forma viável de parar um processo BPEL é com a utilização de Tarefas Humanas, a todas as acções implementadas neste novo módulo foram então associadas as respectivas Tarefas Humanas, garantindo, desta forma, a sua sincronização com o respectivo processo de negócio em BPEL.

Deste modo, a cada interacção dos utilizadores na realização das suas acções sobre a interface gráfica deste módulo, corresponderá sempre uma resposta a uma tarefa cujo valor dependerá da acção tomada. Para processar esta operação foi utilizado o serviço `ExecuteBatchTask`, que responde a uma dada tarefa de um utilizador. Como a API de Java que implementa este serviço apenas permite efectuar o voto de uma tarefa passando o seu identificador, e como este não é guardado no módulo, foi necessário implementar uma solução para encontrar o identificador da tarefa associada a uma determinada acção tomada no módulo. Por exemplo, quando um utilizador pertencente ao HelpDesk quiser anular a resolução do Ticket, é necessário saber o identificador da tarefa que esse utilizador recebeu para ser autorizado a tomar esta acção. Contudo, a realização pela API desta operação de identificação, necessitava de aceder a todas as tarefas abertas para esse utilizador, verificar uma a uma qual a que pertencia à instância do Ticket pretendido e ainda qual a que correspondia à tarefa correcta.

Visto que todo este método evidenciou ser bastante lento através da API, foi implementada uma alternativa de obtenção do identificador da tarefa directamente a partir do repositório de BPEL. Deste modo, com uma simples consulta SQL à tabela `WFTASKS`, foi então possível obter o identificador correcto e responder à tarefa utilizando o Web Service. Com esta optimização, o tempo de processamento desta simples resposta passou de 32 segundos para 0.1 segundos.

7.5.3 Modelo Relacional de Dados

A nível do modelo relacional de dados deste novo módulo, este sofreu igualmente bastantes alterações comparativamente com a sua versão antiga em Workflow. Incluindo não só as necessidades de sincronização com as instâncias BPEL, foram ainda implementadas optimizações a nível das relações e dependências entre tabelas, bem como a criação de novas tabelas e registos que permitiram a implementação das novas funcionalidades

propostas. Para isto, foi definido o modelo relacional de dados apresentado no Anexo A para esta nova versão do módulo de Trouble Tickets.

Neste modelo relacional foi igualmente efectuada a repartição entre Tickets e Tarefas com os respectivos Instance_Id das suas instâncias BPEL. Por outro lado, todo o sistema de unidades, tipos de problemas e problemas existentes na versão Workflow, foi agora generalizada a uma única tabela de Problemas, que podem se relacionar entre si de forma hierárquica. A cada problema está agora associado o respectivo HelpDesk responsável pelo seu tratamento, contendo por sua vez os funcionários pertencentes.

Neste modelo relacional foi ainda incluído um novo sistema de gestão de equipas de utilizadores cuja implementação irá ser iniciada após a conclusão deste projecto.

7.5.4 Motor de Lógica

Todo o motor de lógica da versão em Workflow foi implementado em procedimentos e funções PL/SQL. No entanto, devido a diversos factores e dado o elevado número de correcções efectuadas ao seu código fonte, este apresentava-se bastante confuso, com muitas repetições de código e de estruturas, tanto a nível da validação de dados e aplicação de restrições, como também a nível de acesso e manipulação de dados.

Todos estes factores aliados à especificação dos novos requisitos, integração de novas tecnologias, optimização de toda a interface gráfica e ainda à sincronização com os novos processos BPEL, levaram à necessidade de começar o seu desenvolvimento totalmente do início, em vez de serem efectuadas mais alterações a todo um módulo já bastante devastado. Para isto, foram implementados basicamente quatro pacotes de procedimentos:

- **TTS_BPEL:** Disponibiliza todas as funcionalidades e todo o motor de lógica necessário para o funcionamento do novo módulo dos Trouble Tickets. Aqui são ainda invocadas as operações do Sistema SIBPEL de forma a integrar e sincronizar a tecnologia BPEL com este módulo.
- **TTS_UTIL:** Disponibiliza procedimentos e funções que disponibilizam, no geral, funcionalidades que complementam todas as funcionalidades do TTS_BPEL.
- **TTS_WEB:** Aqui se incluem todos os procedimentos que fornecem funcionalidades para a interface gráfica deste módulo. São aqui incluídas todas as funções Javascript utilizadas, bem como todos os procedimentos PL/SQL que são chamados por Ajax para uma actualização dinâmica, rápida e eficaz de todas as informações aqui apresentadas.
- **TTS_DM:** Gerado automaticamente por uma aplicação implementada no PSI, fornece todos os métodos necessários para realizar consultas e manipulações de dados sobre o modelo relacional deste módulo.

7.5.5 Novas Interfaces

Com a especificação do novo modelo relacional e a implementação do motor de lógica essencial para o funcionamento deste módulo, foi então possível iniciar o processo de desenho e implementação das novas interfaces Web para os utilizadores. Esta primeira etapa relativa ao processo de desenho de interacção assentou principalmente nas seguintes fases:

- **Identificação de Necessidades e Especificação de Requisitos:** Durante esta fase, foram envolvidos e ouvidos alguns utilizadores e stakeholders do sistema na versão workflow, de modo a agrupar todo um conjunto de sugestões e necessidades que estes sentissem tanto a nível visual como a nível funcional. Para isto, foi importante perceber as características e as capacidades dos utilizadores, o que estes procuram atingir e a forma como atingem actualmente. Através destas necessidades foi possível especificar os requisitos apresentados na secção 6.3.
- **Desenvolvimento de Desenhos e Versões Interactivas:** Obtendo todo o conjunto de necessidades dos utilizadores, foi possível esquematizar alguns protótipos em forma de apresentações interactivas, onde fossem agrupadas algumas soluções e ideias para o novo protótipo final. Mais uma vez, o envolvimento dos utilizadores nesta fase foi bastante importante para obter todas as suas apreciações sobre as soluções apresentadas.
- **Análise de Resultados:** Dado o vasto número de utilizadores envolvidos, as suas visões sobre este módulo e sobre as novas soluções apresentadas eram também bastante diferentes. Deste modo, os processos de prototipagem interactivos foram diversas vezes revistos de forma a satisfazer, na sua globalidade, todos os utilizadores deste sistema. Isto levou inclusive à necessidade de apresentar diferentes formas para atingir o mesmo objectivo. A avaliação realizada focou principalmente a rapidez dos utilizadores em alcançar todos os seus objectivos, os erros cometidos, a disponibilização de informações e de guias para os passos seguintes. Os resultados permitiram obter uma versão do protótipo final que foi implementado nas etapas seguintes.

Toda esta fase do processo de desenho foi fundamental para atingir o sucesso do protótipo final desenvolvido. Aqui foi ainda observada uma aplicação em Java desenvolvida externamente ao módulo, que cumpria já algumas necessidades dos funcionários dos HelpDesks na criação de novos problemas, nomeadamente em questões de rapidez. No entanto, com o esforço aplicado para o desenho deste protótipo e com os testes realizados, estava já garantido que este iria exceder todas as necessidades e funcionalidades em todos os níveis comparativamente com a aplicação externa.

Para o seu sucesso, foram ainda identificados os princípios e objectivos de desenho e de usabilidade que o protótipo final deveria satisfazer. Deste modo destacam-se como objectivos fundamentais a sua eficácia, eficiência, rapidez e a maximização da sua utilidade e da facilidade de aprendizagem. Relativamente aos princípios destacam-se ainda os seguintes:

- **Gestão da Visibilidade:** Agrupar e gerir a disposição de funcionalidades relacionadas, para que sejam devidamente identificadas todas as acções principais que o utilizador deverá tomar.
- **Feedback:** Apresentação de informações sobre o estado do sistema, acções escolhidas e guias para os futuros passos a tomar.
- **Restrições:** Introdução de restrições de forma a esconder ou desactivar certas opções que poderão levar um utilizador a cometer erros.
- **Simplicidade:** Revisão iterativa dos protótipos de forma a simplificar ao máximo os seus formulários. Estes deverão apenas conter as informações e interacções necessárias, mantendo todos os seus objectivos mas tornando-o mais claro, simples de utilizar e de alcançar os objectivos.
- **Ajudas e Documentos:** Os utilizadores poderão consultar as ajudas disponibilizadas, bem como toda a documentação elaborada em caso de dúvidas na utilização do sistema.

Obtendo deste modo, os protótipos para as novas versões da interface gráfica deste módulo, deu-se início à sua implementação utilizando e interligando todos os restantes componentes implementados para este projecto.

Os protótipos finais eram basicamente constituídos por apenas quatro páginas Web. No entanto, estas páginas eram capazes de agrupar todo o vasto e excessivo número de formulários existentes na versão anterior. Aliado à utilização de novas tecnologias, principalmente Ajax e Javascript, foi possível implementar soluções que realizavam as suas funcionalidades e apresentavam as suas informações de forma única e dinâmica. Assim sendo, descrevem-se de seguida os componentes interactivos implementados para este projecto.

7.5.5.1 Reportar Problema

Este é o formulário principal que permite iniciar um novo Ticket no sistema. Aqui é incluído um novo sistema Ajax que permite facilmente e rapidamente seleccionar hierarquicamente o tipo de problema que se pretende reportar, à semelhança do que já acontecia na versão antiga mas de forma lenta e pouco intuitiva, cuja navegação era efectuada através de recarregamentos de toda a página Web.

Todavia, através das análises elaboradas, foi verificado que um utilizador pouco usual do sistema não conhece a estrutura hierárquica dos tipos de problemas e não deve portanto ser obrigado a navegar em toda a hierarquia até encontrar o caminho correcto. Por exemplo se este identificar uma anomalia no sistema Wireless, existem à partida dois níveis principais que poderiam ser responsáveis pela sua resolução: Electrónica e Informática. No entanto, problemas relativos a redes Wireless pertencem ao agrupamento da Informática.

De forma a resolver esta questão, foi implementado um inovador sistema de pesquisa em Ajax sensível a alterações nos parâmetros de pesquisa, que permite aos utilizadores escreverem o seu problema genérico sendo imediatamente e dinamicamente apresentados todos os caminhos e opções possíveis que estão relacionadas com o texto introduzido. Com isto, para resolver o problema acima identificado, o utilizador apenas precisa de introduzir a palavra “Wireless” para que seja apresentado o caminho correcto “Informática - Redes - Wireless” (contudo ao escrever apenas “Wir” já devolve o caminho certo).

Tirando partido desta inovação, foi ainda melhorado o sistema de pesquisa de utentes (apenas visível ao HelpDesk). A versão em Workflow, abria uma nova janela onde permitia pesquisar utentes a partir do seu código, nome ou login. Eliminando a necessidade de abrir novas páginas, foi reaproveitado o mesmo sistema Ajax de pesquisa de tipos de problemas, introduzindo um sistema de pesquisa de utilizadores. Este sistema permite ao introduzir o seu nome, código, login ou correio electrónico, apresentar instantaneamente todos os utilizadores que correspondem a estes parâmetros de pesquisa. No entanto, devido ao grande número de registos relacionados com os utilizadores (cerca de 30000), tiveram de ser introduzidas algumas restrições, pois um sistema deste tipo com apresentações dinâmicas e instantâneas dos resultados requer diversas conexões e consultas ao sistema. Deste modo, apenas são processados os parâmetros com mais de quatro caracteres e que apresentem menos de 20 resultados. A figura 7.5 apresenta estes três componentes implementados na versão final do protótipo.

De forma a inovar tanto o sistema anterior como a própria aplicação externa que era utilizada pelos funcionários do HelpDesk na criação de novos problemas, foi desenvolvido um sistema dinâmico totalmente em Javascript e sem recorrer a alterações de dados no repositório enquanto o Ticket não for criado, que permite ao HelpDesk definir de imediato as suas Tarefas e inserir documentos e mensagens. O utilizador apenas poderá inserir documentos.

Relativamente à inserção de Tarefas, este foi o componente mais complexo implementado em toda esta página. Este inclui o sistema de pesquisa Ajax para pesquisar e definir utilizadores do sistema como executantes das Tarefas e ainda um sistema de verificação de todas as tarefas já especificadas para o novo Ticket, de forma a possibilitar a definição de precedências de execução. Uma Tarefa precedida implica que terá de ser resolvida antes da Tarefa que a precede. Visto que este componente permite igualmente alterar e

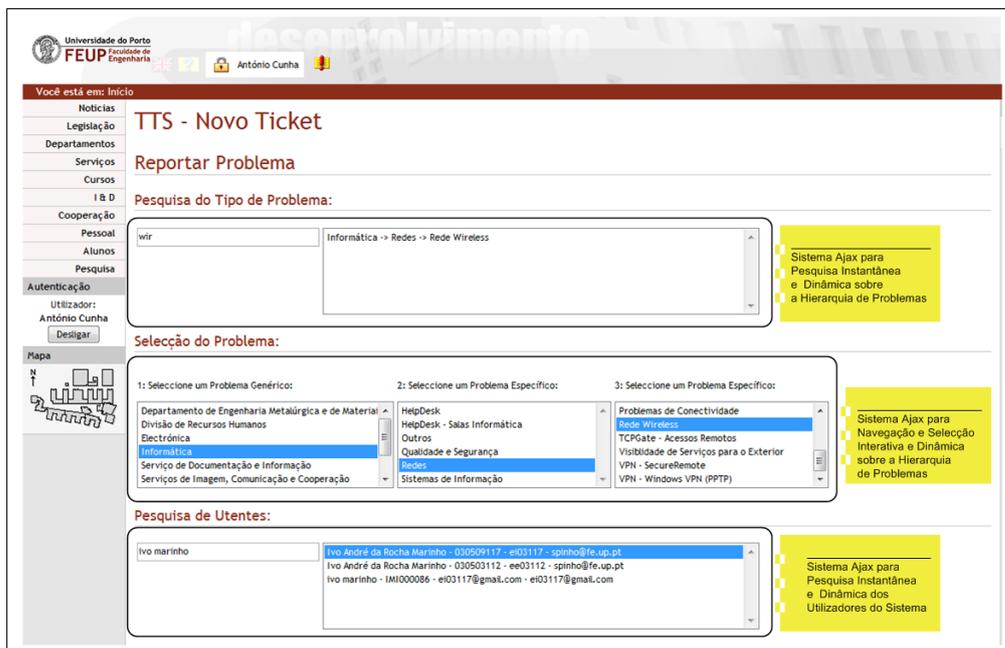


Figura 7.5: Sistemas de Pesquisa e Seleção de Problemas e Utentes

apagar uma dada Tarefa já especificada, foi necessário implementar um sistema recursivo que verifica a consistência das precedências. Por exemplo se existirem três tarefas, se a Tarefa 3 for precedida pela Tarefa 2 e esta pela Tarefa 1, ao alterar a Tarefa 1 não poderá aparecer nenhuma opção para as suas precedências, visto que esta terá de ser a primeira a ser executada para que a sua execução não entre num ciclo infinito.

Quanto à inserção de documentos e mensagens, consistem em sistema que permitem a especificação de vários elementos, contendo sempre uma nova linha onde é possível inserir o novo registo.

Dado o vasto número de funcionalidades implementadas neste formulário, estes componentes incluem ainda um sistema que permite esconder todo o seu conteúdo, de forma a apresentar e agrupar apenas as informações que o utilizador pretende visualizar. Após a sua especificação estes são posteriormente inseridos no repositório de dados e iniciadas as respectivas instâncias BPEL para as Tarefas, quando o utilizador pretender registar o novo Ticket. Estes três componentes são apresentados na figura 7.6.

7.5.5.2 Detalhes Ticket

Esta corresponde à interface de visualização e edição dos detalhes e componentes relacionados com um Ticket. Para isto, estão aqui agrupados numa só página todas as funcionalidades apresentadas na versão anterior em múltiplas páginas tais como edição de detalhes, inserção de Tarefas, inserção de mensagens e documentos, visualização da

Projecto e Implementação

Inserção de Tarefas:

ID	Executante	Precedências	Descrição	Opções
1	Ivo André da Rocha Marinho		Implementar Projecto.	Apagar Editar
2	Manuel José Mendes Pinto Machado	1	Supervisionar Projecto de Ivo Marinho.	Apagar Editar
3	Gabriel de Sousa Torcato David	1	Orientar Projecto de Ivo Marinho.	Apagar Editar

Novo: Introduza as palavras chave de pesquisa de executante. 1, 2, 3. Criar

Sistema JavaScript e Ajax para Especificação de Tarefas

Inserção de Documentos:

Localização	Mostrar
C:\Users\lei03117\Desktop\Relatorio_Ivo_Marinho.pdf	Procurar... <input checked="" type="checkbox"/>
C:\Users\lei03117\Desktop\Anexos.pdf	Procurar... <input checked="" type="checkbox"/>
	Procurar... <input type="checkbox"/>

Sistema JavaScript para Associação de Documentos ao Ticket

Inserção de Mensagens:

Data	Mensagem	Inserido por	Mostrar	Opções
03-07-2008 11:38:51	Este projecto irá ser apresentado no dia 15 de Julho de 2008.	António Cunha	<input checked="" type="checkbox"/>	
Nova Mensagem		António Cunha	<input checked="" type="checkbox"/>	Inserir

Criar Novo Problema

Sistema JavaScript para Inserção de Mensagens para o Ticket

Figura 7.6: Sistemas de Especificação de Tarefas, Mensagens e Documentos

lista de actividades do processo BPEL, finalização do Ticket e das Tarefas e introdução das observações finais.

Esta página é então composta por cinco componentes em Ajax. Sempre que um destes componentes é dinamicamente actualizado, automaticamente todos os restantes são igualmente actualizados. Isto permite uma apresentação coerente de todos os dados e uma enorme rapidez no acesso e processamento das funcionalidades.

Para a edição dos detalhes é apresentada uma opção que permite automaticamente através de Javascript, transformar as informações textuais em caixas de texto e listas de valores permitindo a sua edição. A actualização dos valores é efectuada através de Ajax o que não requer a actualização completa da página. Aqui são incluídos ainda os sistemas de pesquisa de problemas e utentes em Ajax para possibilitar o reencaminhamento do Ticket para um outro HelpDesk e efectuar uma redefinição do utente. Este sistema pode ser visualizado na figura 7.7.

Relativamente à apresentação das Tarefas, é aqui incluído o mesmo sistema de inserção do Reportar Problemas mas possibilitando que os respectivos executantes possam efectuar operações sobre uma Tarefa, como editar, reencaminhar, anular, agendar e concluir. Foi ainda implementada a atribuição de Tarefas para o conjunto de funcionários pertencentes a um dado HelpDesk seleccionado. Toda a criação destas Tarefas implica o início de uma instância BPEL para uma Tarefa, onde é processada a actividade Tarefa Humana que envia uma tarefa para o executante avisando-o que possui uma Tarefa para ser resolvida no

Projecto e Implementação

TTS - Detalhes Ticket

Detalhes Problema

ID: 551
Problema: Rede Wireless
Utente: Ivo André da Rocha Marinho
Prioridade Utente: 3 - Média
Prioridade Helpdesk: 3 - Média
Criador: António "Manuel" Sousa" Cunha
Data Criação: 03-07-2008 11:09
Data Início: 03-07-2008 11:09
Data Fim:
Estado: Em Resolução
Descrição: Ticket Exemplo para o Projecto de Ivo Marinho.

Editar Detalhes Anular Ticket

Tarefas: +
Mensagens: +
Documentos: +
Lista de Actividades: -

Data	Estado
03-07-2008 13:42	Enviada nova mensagem para o Utente.
03-07-2008 13:40	Foi criada uma nova Tarefa para o Ticket.
03-07-2008 13:40	Foi criada uma nova Tarefa para o Ticket.
03-07-2008 11:09	O Ticket foi Analisado. Ticket em Resolução.
03-07-2008 11:09	Iniciou o Processo BPEL Novo Ticket.

Detalhes Problema

ID: 551
Problema: Redes
Informática -> Redes
Informática -> Redes -> Outros
Informática -> Redes -> Acesso ao Exterior (Internet)
Informática -> Redes -> DHCP - Rede Pública - Portais
Informática -> Redes -> DHCP Estático em Gabinetes e Laboratórios
Informática -> Redes -> DNS - Interno / DNS - Externo
Utente: Ivo André da Rocha Marinho
Prioridade Utente: 3 - Média
Prioridade Helpdesk: 1 - Urgente
2 - Alta
3 - Média
4 - Baixa
5 - Muito Baixa
Criador: António "Manuel" Sousa" Cunha
Data Criação: 03-07-2008 11:09
Data Início: 03-07-2008 11:09
Data Fim:
Estado: Em Resolução
Descrição: Ticket Exemplo para o Projecto de Ivo André da Rocha Marinho. Este projecto irá ser apresentado no dia 15 de Julho de 2008.

Alterar Detalhes Anular Ticket

Figura 7.7: Apresentação do Formulário Detalhes Ticket

Ticket. Caso seja enviada para um conjunto de utilizadores, então manter-se-á pendente até que somente um dos utilizadores efectue a alteração do seu estado. Após a finalização de todas as tarefas, aparece a opção de conclusão do Ticket. Caso este seja anulado enquanto existirem tarefas por resolver, foi implementado procedimento PL/SQL que verifica todos os processos BPEL Tarefas activos e aborta o seu processamento, através do Web Service deleteInstance que apaga ainda todas as suas tarefas e notificações pendentes.

O envio de mensagens deste formulário é ainda caracterizado por uma funcionalidade em Ajax que permite filtrar instantaneamente a apresentação de mensagens pela respectiva tarefa a que está associada sem necessidade o recarregamento total da página.

Os restantes componentes consistem na inserção de documentos e na apresentação das actividades das instâncias BPEL do Ticket e de todas as Tarefas relacionadas. Para isto, são consultados e apresentados os valores dos sensores EstadoProcesso de cada instância, que contêm o registo de todas as alterações aos seus fluxos. Todos estes componentes contêm ainda a opção de esconder o seu conteúdo, à semelhança com o Reportar Problema, de modo a organizar e apresentar somente a informação necessária ao utilizador. No que diz respeito a algumas necessidades de abrir novas janelas, foi implementada uma solução que permite apresentar apenas uma nova divisão na página contendo novos formulários e dados, por exemplo na inserção das observações finais de cada Tarefa ou do Ticket e na apresentação de um calendário para seleccionar a data de agendamento de uma Tarefa.

7.5.5.3 Pesquisa de Tickets e Tarefas

Este formulário possibilita que todos os utilizadores efectuem pesquisas de Tickets e Tarefas no sistema. Para além de terem sido introduzidos novos campos de pesquisa relativamente à versão em Workflow, tais como pesquisas pelo criador e selecção de campos e formas de ordenação, foram ainda implementados sistemas em Ajax que permitem apresentar instantaneamente e sem necessidades de recarregamentos das páginas, todos os resultados da pesquisa. Estes resultados são ainda apresentados em forma de paginação, implementado igualmente em Ajax, possibilitando uma grande rapidez na sua visualização e navegação. A figura 7.8 apresenta um exemplo do funcionamento deste sistema.

The screenshot shows a search interface for tickets and tasks. At the top, there is a search form with a 'Criador' field, a large text area for 'Introduza as palavras chave de pesquisa.', and date range filters for 'Data Criação Entre' and 'Data Conclusão Entre'. A 'Pesquisar' button is located below the search fields. Below the search form, the 'Resultados' section includes filters for 'Estado' (set to 'Todos'), 'Ordenar por' (set to 'Data Criação'), and 'Ordenação' (set to 'Descendente'). The main part of the interface is a table of search results. The table has the following columns: ID, Helpdesk, Problema, Utente, Criação, Início, Conclusão, Descrição, Prioridade, Estado, and Detalhes. The results are sorted by creation date in descending order. The first row (ID 551) is highlighted, showing a ticket created on 03-07-2008 at 11:09, assigned to Ivo Marinho, with the description 'Ticket Exe...'. The table ends with a 'Seguinte >>' button and a footer indicating 'Registos 1 a 10 de 538.'

ID	Helpdesk	Problema	Utente	Criação	Início	Conclusão	Descrição	Prioridade	Estado	Detalhes
551	HelpDesk Informática	Informática -> ...	Ivo Marinho	03-07-2008 11:09	03-07-2008 11:09		Ticket Exe...	1	Em Resolução	🔍
550	HelpDesk Informática	Informática		27-06-2008 17:15	27-06-2008 18:15	27-06-2008 17:16	asfdasdfs...	8	Concluído	🔍
549	HelpDesk Informática	Informática	Ivo Marinho	26-06-2008 11:31	26-06-2008 12:33	26-06-2008 11:35	dsfsdfsadf	8	Concluído	🔍
548	HelpDesk Informática	Informática	Ivo Marinho	20-06-2008 11:00		20-06-2008 11:04	asfdasdfs...	8	Anulado	🔍
547	HelpDesk Informática	Informática	Ivo Marinho	20-06-2008 09:58		20-06-2008 09:58	dfsgdfsg	8	Em Análise	🔍
546	HelpDesk Informática	Informática	Ivo Marinho	18-06-2008 17:47	18-06-2008 18:47	18-06-2008 17:55	asfdasdf	8	Concluído	🔍
545	HelpDesk Informática	Informática	Ivo Marinho	13-06-2008 17:21		13-06-2008 17:22	hjkjkjckjkh	8	Concluído	🔍
544	HelpDesk Informática	Informática	Ivo Marinho	13-06-2008 17:17	13-06-2008 17:17	13-06-2008 17:18	sadfasdfasdf	8	Concluído	🔍
543	HelpDesk Informática	Informática	Ivo Marinho	13-06-2008 14:24	13-06-2008 15:25	13-06-2008 15:29	sdfgsdfgsdfg	8	Concluído	🔍
542	HelpDesk Informática	Informática	Ivo Marinho	13-06-2008 12:12	13-06-2008 13:12	13-06-2008 12:13	asfdasdfsdf	8	Concluído	🔍

Figura 7.8: Apresentação do Pesquisa de Tickets e Tarefas

7.5.5.4 Listagens Rápidas

Este consiste num formulário dinâmico que aproveita o funcionamento do Pesquisa de Tickets e Tarefas e de todas as suas funcionalidades para apresentar pesquisas em forma de listagens pré-estabelecidas. Para isto, é recebido um parâmetro de entrada que identifica o tipo de listagem, permitindo a apresentação imediata dos Tickets e Tarefas

sem a necessidade de preencher os campos do formulário de Pesquisa. Deste modo, estas listagens possibilitam que os utilizadores acedam rapidamente às listas dos Tickets colocados, por analisar, por concluir, Tickets com tarefas por resolver e aos Tickets do HelpDesk à que o utilizador pertence.

7.6 O Sistema de Administração BPEL

Após a conclusão atempada de todos os objectivos propostos inicialmente para este projecto, surgiu a ideia de implementar como objectivos suplementares, todo um sistema de administração da tecnologia BPEL e da sua relação com os utilizadores, semelhante às aplicações Oracle BPEL Console e Oracle Worklist, mas no próprio Sistema de Informação do SIGARRA.

Este consistia num novo desafio que permitia tirar o máximo partido de todas as implementações desenvolvidas neste projecto, para fornecer as mesmas funcionalidades disponibilizadas por estas duas aplicações, mas criando todo um ambiente próprio e integrado com todos os utilizadores do SIGARRA, o que não acontece no Oracle BPEL Console onde apenas um administrador pode aceder. Isto levaria ainda a uma total utilização do novo sistema não precisando de aceder a estas aplicações externas da Oracle.

Para a implementação dos requisitos especificados no capítulo 6, foi necessário implementar todo um conjunto de novos serviços que foram agrupados no Web Service BPELConsole. No entanto todas as interfaces e motores de lógica desenvolvidos tiram partido de todos os constituintes implementados para a arquitectura SIBPEL. Deste modo, foram desenvolvidos dois sistemas que serão apresentados detalhadamente nos seguintes tópicos.

7.6.1 Consola BPEL

A Consola BPEL consiste num sistema que apresenta o estado actual de todo o servidor de BPEL, contendo todas as informações relativas aos seus processos e instâncias. Tirando partido da API em Java que implementou os serviços do BPEL Console, foi possível implementar uma página principal muito semelhante à da aplicação da Oracle. Assim, é apresentado na figura 7.9 a página principal desta consola, referindo quais os serviços utilizados para apresentar as informações e executar as acções disponibilizadas na página.

No entanto, nem todas as funcionalidades pretendidas eram totalmente disponibilizadas pela API em Java. Com isto dos desafios mais complexos implementados em todo o projecto, foi integrado aqui na Consola BPEL. Este requisito consistia na apresentação de todo o fluxo das actividades processadas numa dada instância BPEL num dado momento, como acontece na aplicação da Oracle.

Projecto e Implementação

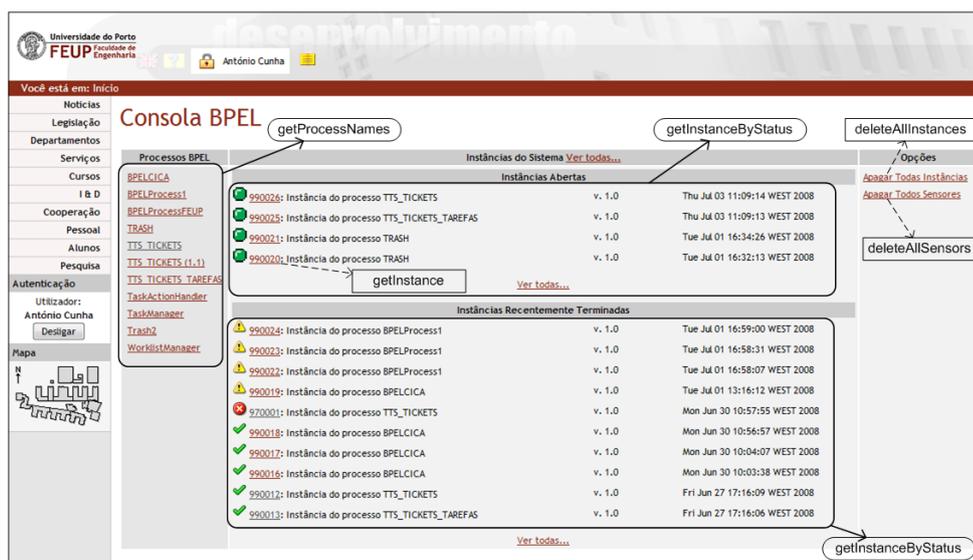


Figura 7.9: Apresentação da Consola BPEL do SIGARRA

Esta funcionalidade está implementada no Oracle BPEL Console através de 23 complexos ficheiros JavaScript, aliados ainda a ficheiros XML, CSS, imagens e Java Server Pages. Deste modo, toda a visualização do fluxo de actividades não era possível ser efectuado através de apenas serviços. Assim, a sua implementação teve início com o estudo de quais as funções JavaScript que são invocadas pelas páginas JSP para a apresentação deste fluxo.

Sabendo que os ficheiros Javascript desenham todo o fluxo através de dois grandes conjuntos XML que são passados por Ajax, a partir da API de Java por ficheiros JSP, com todas as informações sobre uma dada instância e respectivas actividades (o AuditTrail e o AuditTrace), foi implementada uma versão funcional colocando os ficheiros necessários e um exemplo destes dados XML, relativo a uma instância, directamente na base de dados. Esta versão funcional demonstrou que era possível implementar esta funcionalidade no SIGARRA, mas no entanto não era ainda dinâmico.

Para tornar todo o processamento dinâmico, houve a tentativa de incluir os ficheiros Javascript no Sistema de Informação. No entanto, surgiram diversos problemas a nível de permissões e acessos aos restantes componentes. De facto, visto que o Servidor Aplicaçional do BPEL é diferente do servidor do SIGARRA, não era possível aceder aos dados do BPEL estando os Javascript num outro Servidor. De igual modo, existiam problemas de acesso a dois ficheiros XML que definiam templates de apresentação dos dados, pois estes eram acedidos por Ajax que possui uma protecção para não permitir efectuar acessos a servidores diferentes. Ainda relativamente aos ficheiros JSPs que eram invocados, estes acediam à tecnologia BPEL com modos de autenticação próprios, o que não permitia a sua utilização a partir de aplicações externas. Quanto às informações passadas

em XML sobre a instância BPEL, houve também a tentativa de integrar esta informação no SI através de Serviços, mas dado o seu grande conteúdo, verificou-se uma limitação das funcionalidades XPATH do Oracle 10g que não permite a criação de nós superiores a 64K.

Com todas estas restrições foi implementado o WebFlow que consiste numa aplicação Web em Java, contendo todos os ficheiros XML e JSP necessários para a sua utilização. Os ficheiros JSPs aqui incluídos implementam a API de Java do BPEL para fornecer as mesmas funcionalidades que os originais, mas sem restrições de acesso. No entanto, não poderiam ser aqui incluídas todas as classes JavaScript necessárias, pois estas devem estar na localização original da Consola BPEL da Oracle de modo a aceder correctamente a todas as informações.

Contudo, com esta solução havia ainda por resolver a questão das restrições de Ajax, pois o WebFlow encontra-se no servidor aplicacional de BPEL e não no SIGARRA. Para isto, surgiu a ideia de criar uma proxy totalmente transparente, que redirecciona o endereço /bpel/ do SIGARRA para a aplicação WebFlow desenvolvida.

A implementação desta solução permitiu ultrapassar o problema do Ajax, no entanto houve ainda a necessidade de alterar todos os ficheiros JavaScript, XML e JSPs para incluir correctamente todos os caminhos dos ficheiros e das ligações. Para isto, de modo a manter ainda em funcionamento a aplicação da Oracle, foram criadas versões em duplicado com estas alterações, que são devidamente incluídas a partir do PL/SQL no procedimento que inclui todos os ficheiros necessários. Com esta solução obteve-se uma apresentação funcional de todo este fluxo, como pode ser visualizado na figura 7.10.

Um outro desafio relevante na implementação desta Consola foi a funcionalidade de conseguir iniciar novas instâncias BPEL com os respectivos parâmetros de entrada. Contudo, não existia nenhuma forma de consultar quais os parâmetros de entrada e os respectivos tipos de dados que uma dada instância possui.

A solução encontrada passou pela análise desta funcionalidade na consola Oracle, onde foi identificado o método Java que devolve um pequeno código HTML com os parâmetros correctos, incluindo nomes, inputs e tipos de dados. Deste modo, foi apenas necessário implementar e invocar o serviço `getProcessForm` para utilizar a API e devolver este código HTML, que é posteriormente integrado num formulário Web que chama um procedimento PL/SQL que recebe um conjunto de parâmetros de entrada dinâmicos. Por sua vez, este agrupa todos os parâmetros que recebeu numa mensagem SOAP, em conjuntos nome e valor, sendo enviada ao serviço `initiateProcess` que inicia uma nova instância.

7.6.2 Gestor de Tarefas

Relativamente ao Gestor de Tarefas, incluem-se aqui funcionalidades que permitem aceder e responder a todas as tarefas e notificações que são enviadas por instâncias BPEL,

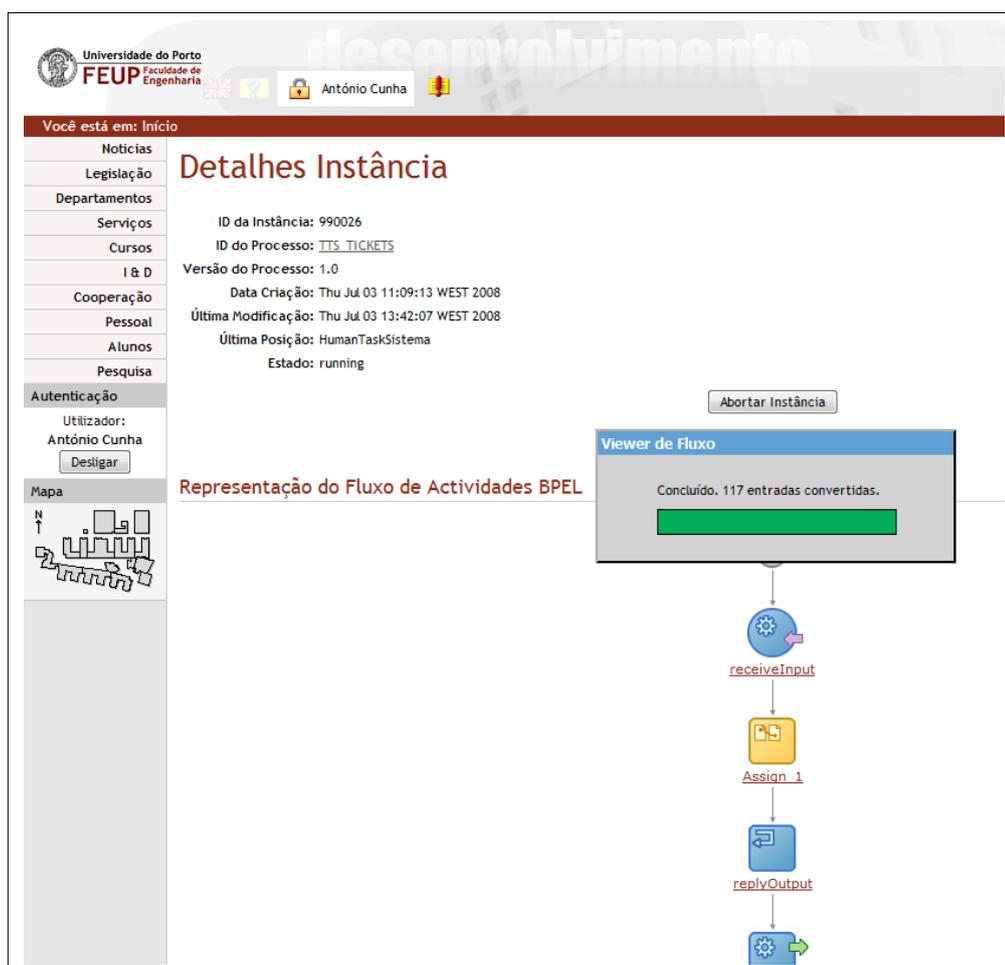


Figura 7.10: Representação do Fluxo de Actividades de uma Instância BPEL na Consola

à semelhança do que é efectuado pelo Oracle Worklist. O desenvolvimento deste sistema reaproveitou as implementações em Ajax dos restantes módulos, para disponibilizar toda uma página dinâmica e cujas informações podem ser acedidas e facilmente filtradas utilizando Ajax, o que melhora substancialmente toda a rapidez deste módulo.

No entanto, de modo a ser possível apresentar as tarefas e notificações, bem como todas as opções de resposta e todas as informações que possibilitam posteriormente enviar uma resposta, foi necessário implementar um complexo serviço que devolve uma estrutura XML com todas estas informações, sendo posteriormente processada por XPATH pelo SI, como acontece para todos os restantes serviços.

Com o funcionamento desta aplicação, foi possível integrar todas as questões de interacção dos utilizadores com o BPEL totalmente a partir do SIGARRA, possibilitando ainda controlar e definir um sistema próprio de apresentação e interacção que supera as funcionalidades do Oracle Worklist em, por exemplo, responder rapidamente a todas as tarefas e notificações de uma só vez. Na versão da Oracle, esta resposta é efectuada uma

a uma, realizando sempre recarregamentos totais da página em cada resposta. Assim, é apresentado um exemplo do funcionamento desta aplicação na figura 7.11.

The screenshot shows a web application interface for 'TTS - Tarefas Utilizador'. The page includes a navigation menu on the left with categories like 'Notícias', 'Legislação', 'Departamentos', 'Serviços', 'Cursos', 'I & D', 'Cooperação', 'Pessoal', 'Alunos', 'Pesquisa', 'Autenticação', and 'Mapa'. The main content area features a 'Filtros' section with two dropdown menus: 'Tipo de Processo' (set to 'TTS_TICKETS') and 'Estado' (set to 'Por Responder'). Below the filters is a 'Lista de Notificações' table with columns for 'Tipo', 'Titulo', 'Data Criação', 'Expira em', 'Prioridade', and 'Opções'. The table contains two entries for 'TTS_TICKETS' with titles indicating problem resolution and new messages. An 'Actualizar' button is located at the bottom of the notification list.

Tipo	Titulo	Data Criação	Expira em	Prioridade	Opções
TTS_TICKETS	TTS - A resolução do seu problema foi iniciada.	03-07-2008 11:9		3	Tomei Conhecimento
TTS_TICKETS	TTS - Tem uma nova mensagem para o seu problema.	03-07-2008 13:42		3	Tomei Conhecimento

Figura 7.11: Apresentação do Gestor de Tarefas BPEL do SIGARRA

Capítulo 8

Testes e Resultados

Este capítulo apresenta a última fase de avaliação do projecto, realizado após a implementação de todos os requisitos propostos, de forma a testar todo o seu desempenho global e verificar a satisfação dos resultados obtidos.

A fase de testes consistiu numa intensiva utilização do sistema, permitindo corrigir e otimizar alguns aspectos funcionais e visuais que não estiverem de acordo com os objectivos propostos para este projecto. Foram igualmente efectuadas avaliações ao desempenho da tecnologia BPEL e aos seus processos de negócio, comparando os resultados obtidos com a versão antiga em Oracle Workflow.

Por fim, houve ainda a necessidade de avaliar a satisfação dos futuros utilizadores deste sistema, no cumprimento de todas as suas necessidades. Deste modo, serão ainda apresentados dois exemplos de utilização, ilustrando o seu funcionamento geral e com especial importância para a sincronização entre o módulo desenvolvido e a tecnologia BPEL.

8.1 Avaliação Global do Projecto

A avaliação do projecto desenvolvido permitiu verificar o total e correcto cumprimento de todos os objectivos propostos. No entanto, dado o elevado número de novas tecnologias utilizadas e a total reformulação do módulo dos Trouble Tickets, esta fase foi essencial para testar o desempenho de todas as funcionalidades desenvolvidas. Para isto, a avaliação do sistema envolveu os seguintes componentes:

- **Processo de Negócio BPEL:** Com a definição do novo processo de negócio em BPEL para os Tickets e Tarefas, foi possível concluir que estes disponibilizam as mesmas funcionalidades que a sua versão em Oracle Workflow. No entanto, estes

possuem já uma vertente distribuída, o que permitiu verificar a optimização no controlo e acesso a certas funcionalidades implementadas do Sistema de Informação. Por outro lado, todo o processo desenvolvido é facilmente alterado pelo Designer da Oracle e novas actividades podem sempre ser incluídas. Toda a gestão e acesso a ligações externas pelo BPEL revelou ser extremamente simples e eficaz. Também no que diz respeito à sua ligação com utilizadores da FEUP, todas as tarefas e notificações especificadas foram sempre correctamente enviadas. Em relação aos testes de performance dos processos BPEL e do seu servidor, este possui uma opção de testes de desempenho, que permite a criação de múltiplos processos instantaneamente simulando uma utilização real. Aqui verificou-se um ligeiro atraso no seu processamento, no entanto o seu comportamento apresentou-se sem falhas e todas as ligações foram correctamente estabelecidas.

- **Web Services:** Os dois Web Services implementados disponibilizam funcionalidades essenciais para o correcto funcionamento de todo sistema. A sua avaliação de desempenho constituiu igualmente um teste a todos os métodos da API em Java incluída no Oracle BPM, que revelaram ser bastante completas. No entanto, visto que a sua API não se encontra devidamente documentada, foram testadas as utilizações e funcionalidades de alguns novos métodos, o que permitiu efectuar algumas alterações que optimizaram o desempenho final destes serviços. Deste modo, obteve-se um total controlo sobre a tecnologia BPEL de forma muito eficaz e eficiente.
- **Sistema SIBPEL:** Quanto à avaliação do desempenho do Sistema SIBPEL, foram criados diversos Tickets e Tarefas, de forma a testar o seu comportamento e rapidez nas ligações e sincronizações com as respectivas instâncias BPEL. Isto permitiu igualmente analisar o seu desempenho em relação a grandes trocas de informações com os serviços implementados. Os tempos de comunicação com estes serviços superaram todas as expectativas. Apenas a comunicação com o serviço `getTasks` demonstrou demorar um pouco mais que as restantes, pois a API utilizada não fornece métodos que retornem directamente todas as informações por ele disponibilizadas. Dada ainda a grande quantidade de estruturas XML transmitidas, implica um tempo de descodificação mais acentuado por parte do SI. No entanto, estes encontram-se ainda em níveis bastante aceitáveis e é apenas utilizado no Gestor de Tarefas o que não influencia directamente o comportamento dos Trouble Tickets.
- **Motor de Lógica dos Trouble Tickets:** Com a total reformulação deste módulo, obteve-se resultados bastantes superiores à sua versão em Workflow. Deste modo, foi possível optimizar e simplificar todo o modelo relacional de dados e todos os acessos a ele efectuados, permitindo obter uma melhoria significativa de todo o seu processamento. Também a sua interacção com a tecnologia BPEL, permitiu dividir e

introduzir novas funcionalidades nos dois sistemas, melhorando na sua globalidade todo o seu desempenho.

- **Sistema de Administração BPEL:** Quanto à Consola BPEL e ao Gestor de Tarefas desenvolvido no Sistema de Informação, todo o seu funcionamento revelou ser bastante eficaz dado o conjunto de dados que estão envolvidos. No que diz respeito à Consola BPEL, a disponibilização de informação dos processos demonstrou ser bastante rápida. Também as implementações mais complexas, como a apresentação do fluxo de actividades e a criação de novas instâncias, obtiveram resultados idênticos à versão original da Oracle, o que permite substituir toda a sua utilização. Quanto ao Gestor de Tarefas, toda a apresentação de tarefas e notificações pendentes assemelha-se à versão original. No entanto, no que diz respeito a utilização da funcionalidade de resposta, visto que a versão da Oracle apenas permite enviar uma resposta de cada vez, o sistema desenvolvido neste projecto permite num único acesso processar múltiplas respostas, superando significativamente a versão original no tempo total de utilização desta funcionalidade.
- **Interfaces Web:** Relativamente à interface Web tanto do módulo dos Trouble Tickets como do sistema de administração do BPEL, com a introdução da tecnologia Ajax e Javascript obteve-se resultados muito superiores às anteriores versões. De facto, só o simples caso de não ser necessário recarregar toda uma página devido a uma pequena alteração ou consulta de dados, já melhora significativamente os tempos de utilização, sendo deste modo um factor fundamental para a melhoria da qualidade do trabalho diário de vários funcionários da instituição de ensino.

Com a verificação da satisfação dos resultados obtidos nos testes e avaliações efectuadas, pode-se concluir que todas as soluções especificadas e implementadas neste projecto revelaram ser bastante eficazes e o seu comportamento superou, por diversas vezes, todas as expectativas. Deste modo, o sistema encontra-se totalmente apto para iniciar o seu ciclo de funcionamento e de evolução dos restantes módulos ainda em Workflow.

8.2 Caso de Utilização Simples

Nesta secção será apresentado um exemplo de um simples caso de utilização do módulo dos Trouble Tickets. Para isto, um utente irá criar um novo Ticket que será posteriormente analisado e resolvido por um funcionário do HelpDesk, como é demonstrado na figura 8.1. Este simples caso de utilização é composto pelas seguintes etapas:

1. Um utente verifica que o seu cartão de aluno já não funciona na faculdade e reporta esta situação criando um novo Ticket no sistema. Para isto, este pesquisa e

selecciona a categoria do problema, define a sua prioridade e introduz a respectiva descrição com todos os detalhes necessários para a sua resolução.

2. Um funcionário do Helpdesk responsável recebe o novo Ticket, analisa o problema reportado e verifica que de facto o cartão do aluno já não é válido no sistema.
3. De seguida, o funcionário inicia a sua resolução criando uma nova Tarefa para o próprio utente, dizendo que este precisa de se dirigir à secretaria para efectuar o levantamento de um novo cartão.
4. O utente recebe a Tarefa e dirige-se à secretaria a fim de obter o seu novo cartão, concluindo posteriormente a resolução da Tarefa. Caso algum interveniente do problema tenha alguma dúvida, poderá ainda introduzir uma mensagem para o Ticket ou para uma Tarefa.
5. O Helpdesk é notificado que todas as tarefas se encontram concluídas e um dos seus funcionários processa igualmente a conclusão do Ticket, introduzindo uma resposta final de que este foi concluído com sucesso.
6. Por fim, o utente pode visualizar o estado final do Ticket e caso este esteja insatisfeito, poderá ainda reabrir novamente a sua resolução.

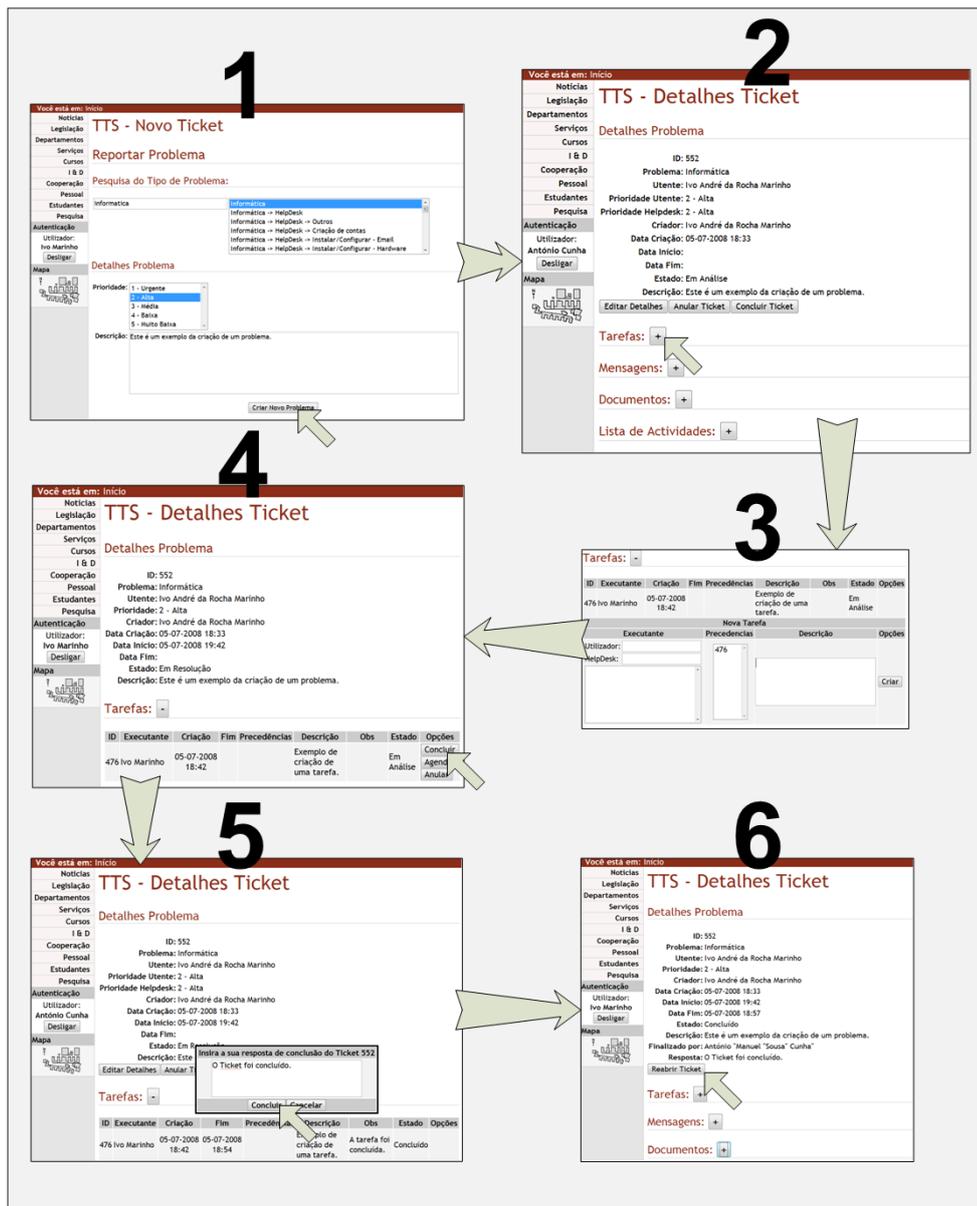


Figura 8.1: Exemplo Simples de Utilização do Sistema

8.3 Caso de Utilização Completo

Esta secção apresenta o exemplo do caso de utilização introduzido na secção anterior, mas de forma mais completa, dando especial destaque a todo o funcionamento do sistema nomeadamente na sincronização com as instâncias BPEL e na gestão das tarefas e notificações enviadas. Para isto são apresentadas na figura 8.2 as seguintes etapas do sistema desde a criação de um Ticket até a sua conclusão.

1. Um utente após identificar um problema reporta-o no módulo de Trouble Tickets.
 - (a) Quando um novo problema é criado, automaticamente é iniciada uma nova instância BPEL do processo de negócio Ticket. Os seus detalhes e fluxo podem ser visualizados pela Consola BPEL do Sistema de Informação.
 - (b) O fluxo de actividades da instância envia uma nova tarefa para todos os funcionários pertencentes ao Helpdesk responsável pela categoria do problema, informando que foi iniciado um novo Ticket e que é necessário tomar uma decisão de análise sobre o mesmo. Da mesma forma, esta irá ficar pendente no Gestor de Tarefas de todos os receptores até que um analise o problema reportado, e o fluxo da instância BPEL irá igualmente ficar a espera que esta tarefa seja actualizada.
2. Um funcionário do Helpdesk que recebeu a notificação decide efectuar a análise do problema. Deste modo, a tarefa é actualizada e removida automaticamente do Gestor de Tarefas de todos os seus receptores. A instância BPEL continua agora o seu fluxo até à tarefa seguinte, enviando uma notificação ao utente informando-o que a resolução do seu Ticket foi iniciada.
3. De seguida, o funcionário inicia a sua resolução criando uma nova Tarefa para o próprio utente.
 - (a) Sempre que uma nova Tarefa é criada para um executante, é igualmente iniciada uma nova instância BPEL do processo de negócio Tarefa. Os seus detalhes podem igualmente ser visualizados na Consola BPEL do Sistema de Informação.
 - (b) A nova instância iniciada é responsável por enviar uma tarefa ao executante, neste caso ao utente, informando-o que este deverá efectuar uma operação de resolução sobre a Tarefa criada.
4. O utente visualiza a Tarefa enviada e procede à sua conclusão, removendo-a automaticamente do Gestor de Tarefas.
 - (a) Na finalização de uma Tarefa, o seu processo de negócio BPEL é responsável por identificar se todas as Tarefas do Ticket se encontram terminadas. Em caso afirmativo, e como neste exemplo só foi criada uma Tarefa, este envia uma notificação aos funcionários do Helpdesk, informando-os deste facto e de que o Ticket se agora encontra apto para ser concluído.
 - (b) Após esta verificação e notificação, o fluxo da instância do processo BPEL da Tarefa é concluído.
5. Um funcionário do Helpdesk visualiza a notificação e procede à conclusão do Ticket.

Testes e Resultados

- (a) Quando a resolução do Ticket é terminada, a instância BPEL termina todas as suas notificações e tarefas pendentes, e envia uma última notificação ao seu utente, informando-o que a resolução do seu Ticket foi concluída.
 - (b) De seguida, o fluxo da instância do processo BPEL do Ticket é concluído.
6. Por fim, o utente visualiza a última notificação e poderá, por algum motivo, recomençar todo este processo reabrindo novamente o Ticket.

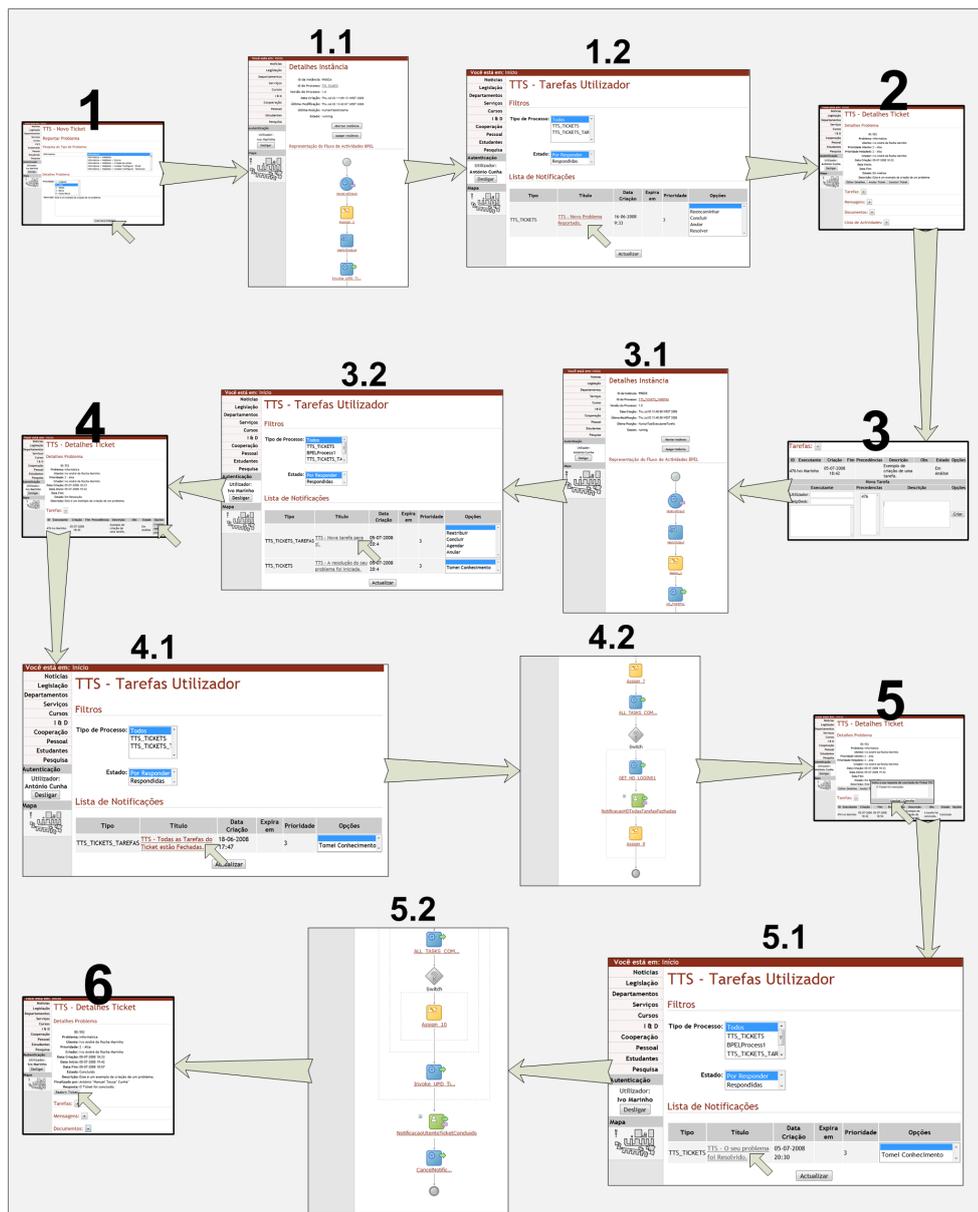


Figura 8.2: Exemplo Completo de Utilização e Funcionamento do Sistema

Testes e Resultados

Capítulo 9

Conclusões e Perspectivas de Trabalho Futuro

Este capítulo apresenta uma apreciação final e global à satisfação dos objectivos propostos para este projecto, e as respectivas direcções para uma futura evolução das soluções e funcionalidades implementadas.

9.1 Satisfação dos Objectivos

Numa avaliação global ao projecto desenvolvido, é possível concluir que todos os objectivos propostos foram cumpridos de forma adequada e atempadamente. Deste modo, isto possibilitou ainda a implementação de um importante sistema de administração para esta nova tecnologia BPEL.

Com todo o sucesso das funcionalidades e soluções implementadas, obteve-se um produto final que superou todas as expectativas. Este representa uma nova base de modernização para todo o Sistema de Informação, onde é agora possível integrar e utilizar facilmente as mais recentes tecnologias, que quando aliadas a todas as vantagens da construção de uma arquitectura distribuída seguindo os conceitos do SOA, permitirá revolucionar todas as aplicações envolvidas, como foi o caso do módulo de Trouble Tickets.

De facto, toda a reformulação deste novo módulo permitiu tirar partido de todas as vantagens da sua integração com a tecnologia BPEL. Da mesma forma, existiu ainda uma forte aposta na reformulação de todas as interfaces deste módulo, de modo a seguirem as tendências de modernização e de aproveitamento das vantagens da utilização de novas tecnologias como o Ajax e Javascript. Os resultados apresentados demonstram todas as melhorias das suas capacidades e encontram-se agora de acordo com todos os objectivos e requisitos propostos pelos seus utilizadores principais.

Contudo, há ainda que referir a importância do sucesso da implementação de toda a arquitectura SIBPEL, construída de modo a suportar a ligação e sincronização entre o Sistema de Informação e a tecnologia BPEL. Só com toda a eficácia e eficiência demonstradas pelas suas funcionalidades, foi possível obter os resultados finais tão desejados. Esta arquitectura assinalou, deste modo, o início da interligação de sistemas e serviços aliadas à utilização das mais recentes tecnologias, o que permitirá ainda simplificar todo o futuro processo de manutenção e evolução.

No entanto, todo o trabalho implementado neste projecto não teria sido possível sem a definição e especificação de todas as normas de conversão da tecnologia Oracle Workflow para BPEL. Os processos de negócio em Oracle Workflow representam ainda um dos principais componentes indispensáveis para o funcionamento de todo o sistema. Deste modo, foi essencial encontrar soluções de evolução e mapeamento das suas funcionalidades para o BPEL. Todas estas soluções apresentadas representam uma importante direcção de orientação para uma correcta evolução de sistemas para o BPEL, tornado este documento ainda mais inovador a este nível, não só pela sua introdução à uma tecnologia muito recente, mas também por referir aspectos da sua utilização e integração nunca antes apresentados.

Aliado a estes factos, inclui-se ainda a importância do desenvolvimento suplementar do sistema de administração desta tecnologia, que possibilitou uma total definição de ferramentas e aplicações próprias, sem a necessidade da utilização das aplicações da Oracle, e permitiram ainda uma total integração com os utilizadores de todo o sistema.

Deste modo conclui-se que a implementação deste projecto constitui um sistema de extrema importância para a evolução da instituição, e todos os conceitos e exemplos de transformação apresentados servirão perfeitamente de exemplo para a evolução dos restantes módulos do sistema para a tecnologia BPEL. Deste modo, será então possível cumprir os futuros objectivos do PSI, que passam pela actualização do seu servidor aplicacional para a versão 11g e pela interligação e comunicação entre as diversas instituições que compõem a Universidade do Porto.

9.2 Trabalho Futuro

Em termos de desenvolvimentos futuros, este deverá passar principalmente pela utilização dos conceitos apresentados e das implementações desenvolvidas, de forma a evoluir para BPEL os restantes módulos do Sistema de Informação, que se encontram ainda definidos em Oracle Workflow.

Desta forma, isto permitirá ainda cumprir os objectivos principais do PSI que consistem ainda na evolução do seu servidor aplicacional para a versão 11g e a sua interligação com várias instituições. Para isto, deverão ainda ser consideradas a actualização dos serviços implementados, de forma a incluir atributos de segurança e de autenticação dos

seus acessos e ainda pequenas alterações que terão de ser realizadas, para utilizar a nova API em Java da versão 11g.

Por outro lado, com a reformulação dos restantes módulos do sistema para a integração com a tecnologia BPEL, todo o seu motor de lógica deverá igualmente ser revisto, de modo a tirar partido da utilização das mais recentes tecnologias. Com isto, poderão ainda ser adicionadas novas funcionalidades ao sistema SIBPEL de forma a cumprir possíveis novos requisitos. Ainda a nível de interfaces, todos os desenvolvimentos realizados e inovações introduzidas deverão influenciar possíveis reformulações das restantes interfaces Web do SIGARRA.

Por fim, não poderia deixar de se referir futuras utilizações e integração das restantes actividades BPEL que não foram incluídas nos processos de negocio desenvolvidos neste projecto, tais como o envio de mensagens SMS, Email e Voz, bem como as novas actividades incluídas no Oracle BPEL Process Manager 11g.

Conclusões e Perspectivas de Trabalho Futuro

Referências

- [Das08] Manoj Das. Migrating to bpm from oracle work flow, Março 2008.
- [eB07] Oracle e BEA. Ws-bpm extension for people (bpm4people), version 1.0, Junho 2007.
- [eHS07] Jennifer Preece Yvonne Rogers e Helen Sharp. *Interaction Design: beyond human-computer interaction*. John Wiley & Sons, Inc., Second edition, 2007.
- [eJE07] Diane Jordan e John Evdemon. Web services business process execution language version 2.0 oasis standard, Abril 2007.
- [eMK07] Deanna Bradshaw e Mark Kennedy. Oracle bpm process manager developer's guide 10g, Janeiro 2007.
- [eMZ06] Harish Gaur e Markus Zirn. *BPEL Cookbook*. Packt Publishing, First edition, Julho 2006.
- [ePS06] Matjaz B. Juric Benny Mathew e Poornachandra Sarang. *Business Process Execution Language for Web Services*. Packt Publishing, Second edition, Janeiro 2006.
- [Ken07] Mark Kennedy. Oracle bpm process manager administrator's guide 10g, Janeiro 2007.
- [Ora04] Oracle. Oracle bpm process manager datasheet, 2004.
- [SD04] Dave Shaffer and Brian Dayton. Orchestrating web services: The case for a bpm server, Junho 2004.
- [Vas07] Yuli Vasiliev. *SOA and WS-BPEL*. Packt Publishing, First edition, Setembro 2007.

REFERÊNCIAS

Anexo A

Diagramas

São apresentados neste anexo algumas ilustrações que permitem completar a compreensão dos tópicos abordados neste documento, nomeadamente tópicos relativos ao módulo dos Trouble Tickets e da sua implementação e integração com a tecnologia BPEL.

A.1 Actividades do Processo de Negócio BPEL

O diagrama apresentado na figura [A.1](#) foi elaborado por forma a especificar e representar o fluxo de actividades que o novo processo de negócio do módulo dos Trouble Tickets possui.

A.2 Processo de Negócio BPEL de um Ticket

Nesta secção é apresentada na figura [A.2](#) o núcleo central do processo BPEL definido para um Ticket, pertencente ao módulo dos Trouble Tickets, que foi elaborado para este projecto por forma a evoluir a anterior versão em Oracle Workflow.

A.3 Processo de Negócio BPEL de uma Tarefa

Esta secção apresenta na figura [A.3](#) o processo BPEL definido para uma Tarefa, pertencente ao módulo dos Trouble Tickets, que foi elaborado para este projecto por forma a incluir o conceito de reutilização de processos. Sendo este um dos objectivos principais da Arquitectura SOA, a repartição da anterior versão em Workflow em dois processos distintos irá permitir reaproveitar as funcionalidades disponibilizadas por este processo em outros módulos.

A.4 Modelo Relacional do Novo Módulo de Trouble Tickets

Relativamente ao novo modelo relacional de dados definido e utilizado pelo novo módulo de Trouble Tickets, é apresentado na figura [A.4](#) todo o esquema de tabelas, registos e dependências existentes. É igualmente possível identificar a inclusão de registos, tais como o identificador da instância, que permitem a interligação dos Tickets e Tarefas com a tecnologia BPEL e com as respectivas instâncias criadas.

Diagramas

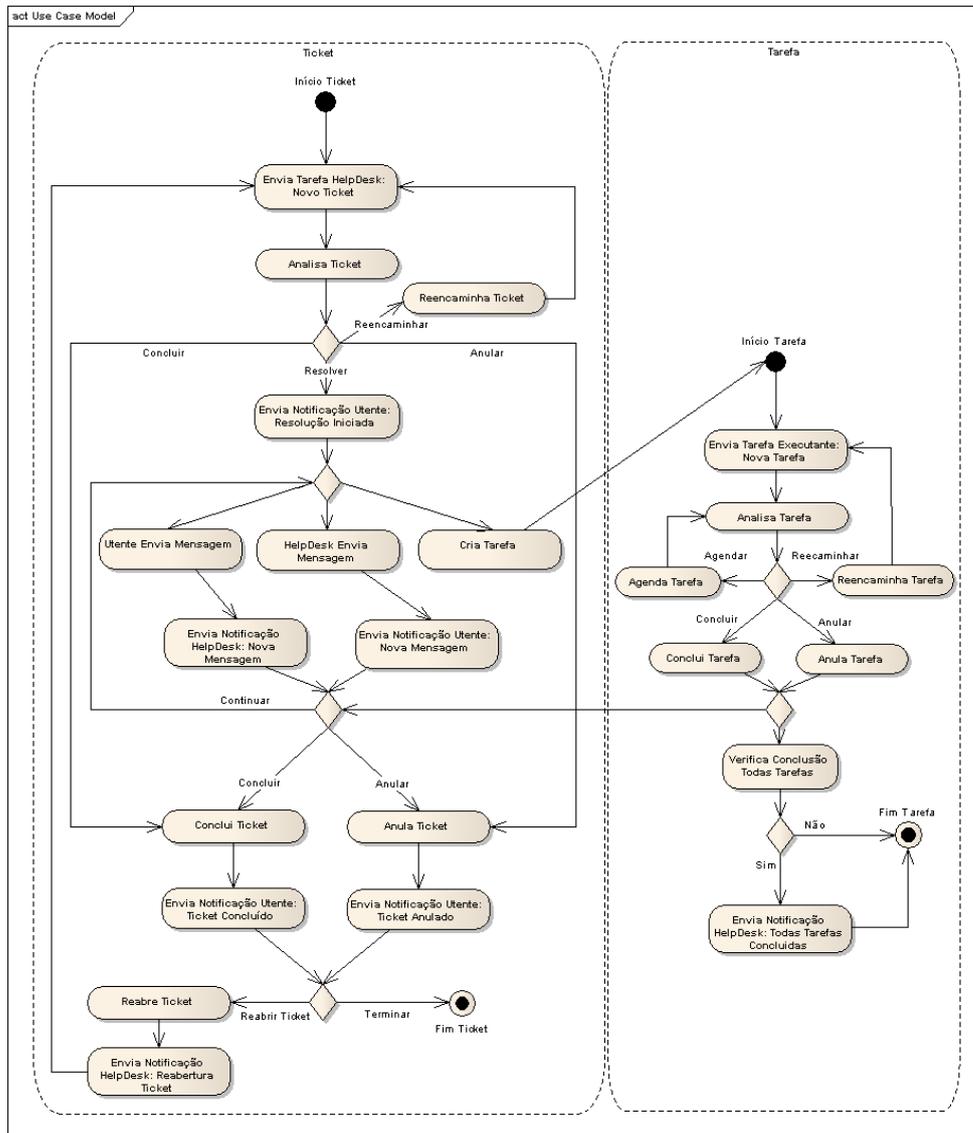


Figura A.1: Diagrama de Actividades do Novo Processo de Negócio dos Trouble Tickets

Diagramas



Figura A.2: Núcleo Central do Processo de Negócio em BPEL de um Ticket

Diagramas

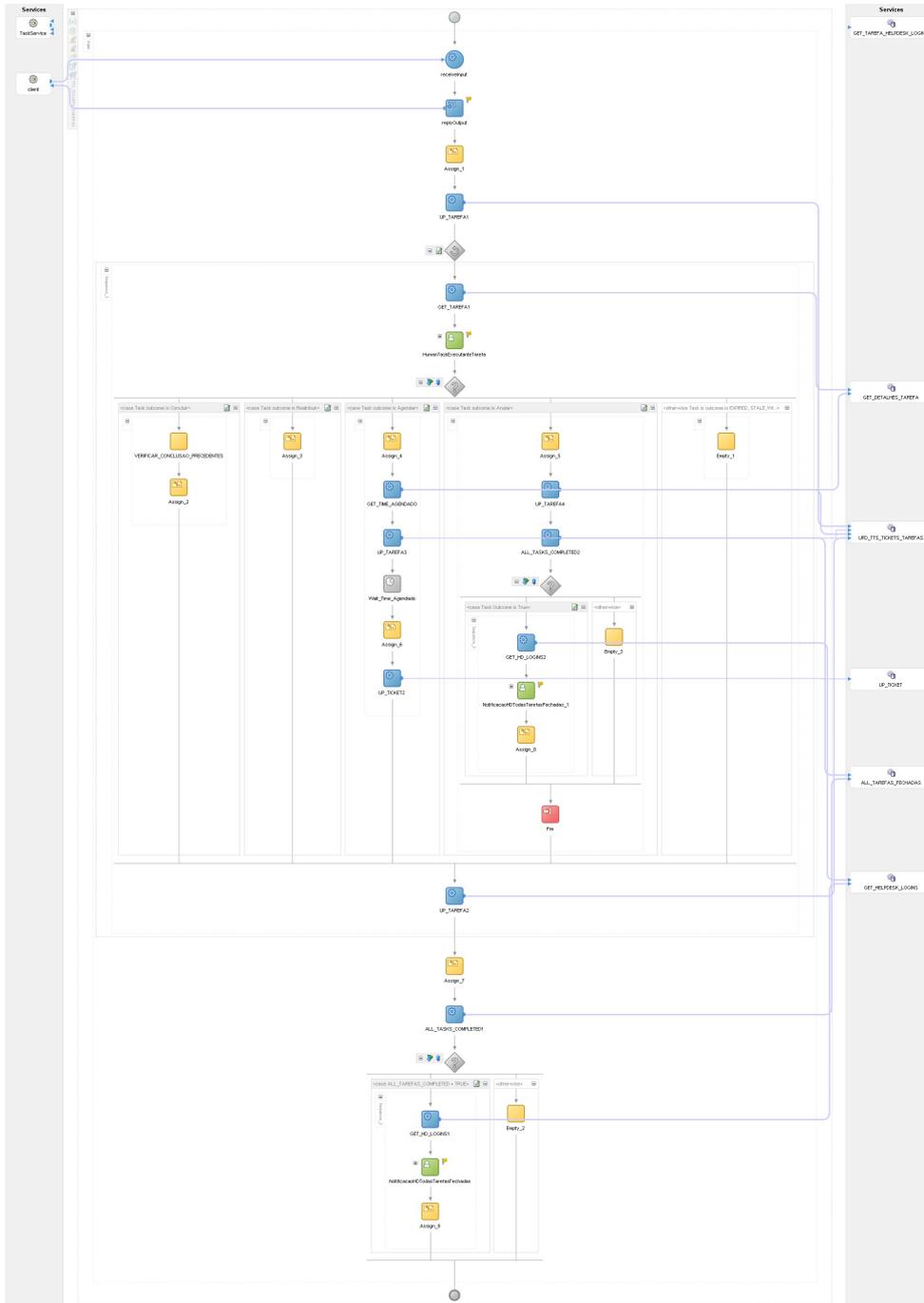


Figura A.3: Processo de Negócio em BPEL de uma Tarefa

Diagramas

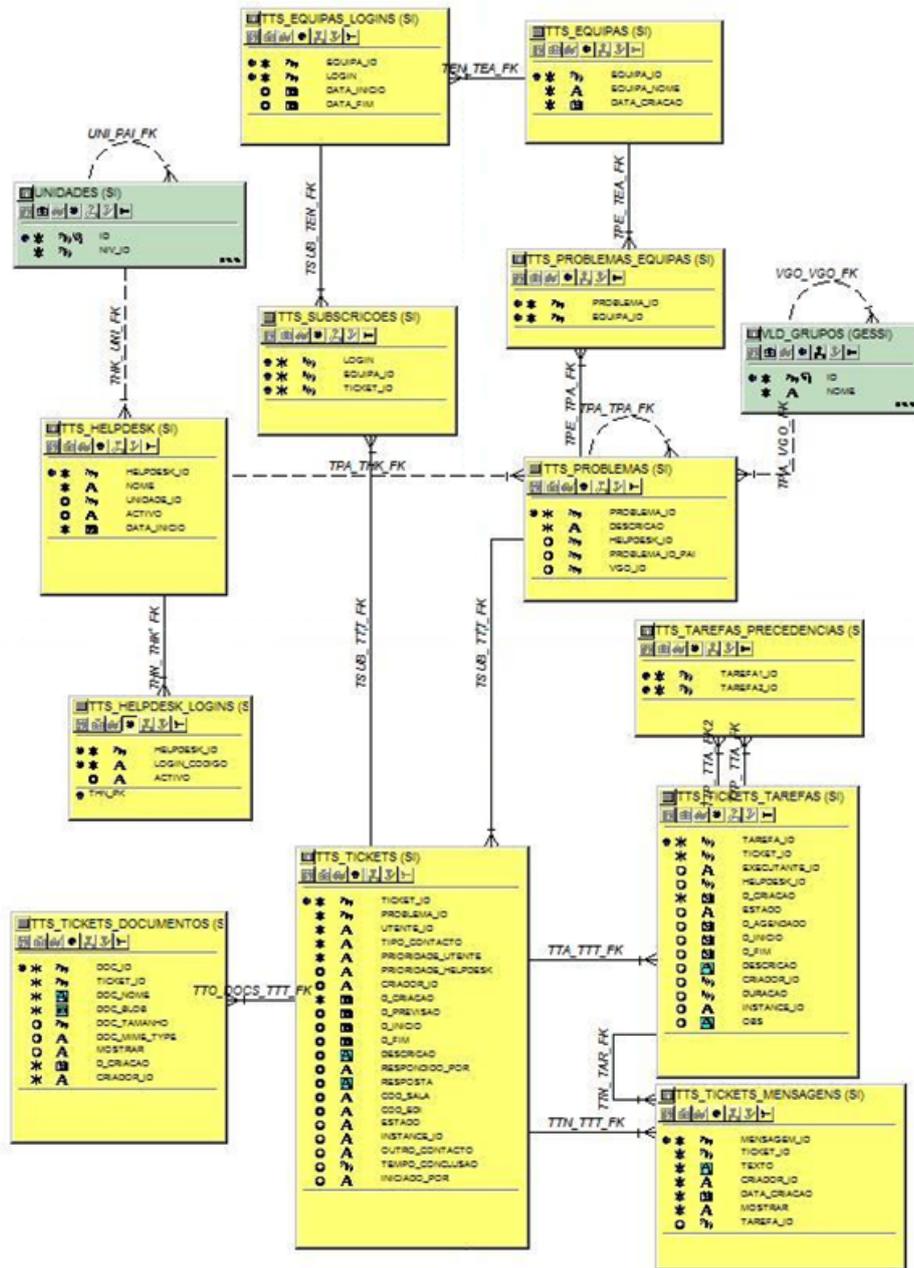


Figura A.4: O Novo Modelo Relacional do Módulo dos Trouble Tickets