



Universidade do Porto  
Faculdade de Engenharia

**FEUP**



João Manuel da Silva Braga

# Workflow de Processos de Negócio na Modelo Continente Hipermercados SA

**CONFIDENCIAL**

**Workflow de Processos de Negócio na  
Modelo Continente Hipermercados S.A.**

*João Manuel da Silva Braga*

**Relatório do Estágio Curricular da LEIC 2001/02**

Orientador na FEUP: Prof. Eugénio Oliveira

Orientador na Modelo Continente Hipermercados S.A.: Eng. Paulo Ribeiro e Dr. Carlos  
Latourrette



**Faculdade de Engenharia da Universidade do Porto  
Licenciatura em Engenharia Informática e Computação**

2002-09-16

**UNIVERSIDADE DO PORTO**  
Faculdade de Engenharia  
**BIBLIOTECA** M  
N.º 65323  
CDU 604(047.3)  
Data 14 1 3 2003

*Em memória da minha avó materna*

## Resumo

O presente relatório documenta o projecto “Workflow de Processos de Negócio”, realizado no âmbito do estágio curricular da licenciatura em Engenharia Informática e Computação, da Faculdade de Engenharia da Universidade do Porto, que decorreu na Modelo Continente Hipermercados, S.A..

O trabalho realizado no âmbito deste estágio inseriu-se no estudo dos conceitos relacionados com o *workflow*, identificação de melhorias no processo de entrada e saída de artigos da placa de vendas e implementação do processo de comunicação de artigos.

O trabalho efectuado dividiu-se nas seguintes partes:

- Uma primeira parte de introdução ao *workflow*, apresentando as normas existentes e caracterizando conceitos e benefícios. Ainda nesta parte foi desenvolvida uma forma normalizada para análise destes projectos, afim de facilitar posteriores análises. Foram também identificados os erros mais frequentes e as formas de os evitar.
- Numa segunda parte, foi feito um *procurement* de ferramentas, afim de descrever o estado da arte das tecnologias associadas ao *workflow*. Em mais pormenor foi feita uma análise das ferramentas da Oracle (usadas na empresa), de forma a compreender as funcionalidades existentes e quais as possibilidades de evolução.
- Noutra parte, foi criada uma metodologia para desenvolvimento de projectos nesta área, tendo-se definido as fases a existir no projecto, as regras de interface, desenho de processos e documentação. Esta metodologia encontra-se ainda enriquecida com a definição da arquitectura tipo das aplicações desenvolvidas.
- Finalmente, foi implementado um projecto intulado “Comunicação de Artigos” que serviu para validar a metodologia criada e melhorar o processo de comunicação às lojas. A implementação consistiu na análise, desenho dos processos e interface gráfica, arquitectura, especificação de testes de aceitação e acompanhamento da solução. Foi igualmente feita uma análise da possibilidade de expansão deste processo, para suportar a entrada e saída dos artigos desde o fornecedor até à placa de vendas.

Em conclusão, os objectivos foram atingidos estando neste momento a decorrer o piloto do projecto implementado. Além disso, o estágio foi rico em experiências, tais como, conhecimentos de processos administrativos na área do retalho, gestão de projectos de software na área do *workflow*, conhecimento da solução de *workflow* da Oracle e trabalho numa empresa de grande dimensão.

## **Agradecimentos**

Começo por agradecer a todos aqueles que na Modelo Continente me receberam e apoiaram ao longo do estágio. Quero agradecer ainda aos elementos do centro de competências WEB/MIS/DW, pelo apoio e companheirismo demonstrado, em especial Eng. João Piedade e Eng. Alberto Mota. Aos meus orientadores, Eng. Paulo Ribeiro e Dr. Carlos Latourrette, quero agradecer pelo excelente apoio profissional e pessoal dado ao longo desta nova experiência. Termino agradecendo a oportunidade que me foi dada, pela Modelo Continente, e que me permitiu a criação de novas experiências e de novos amigos.

Deixo um sincero agradecimento às pessoas da FEUP, que contribuíram para a minha formação, realçando os professores que souberam encaminhar-me na direcção correcta. Um obrigado especial, ao Prof. Eugénio Oliveira por toda a amizade e apoio dado ao longo do curso e do presente estágio.

Para os meus pais, fica o agradecimento pelo carinho, paciência que tiveram e pelos sacrifícios que fizeram para tornar possível chegar até aqui. Ao meu avô agradeço as longas tardes de convívio que me refrescavam as forças para continuar. Ao meu irmão agradeço a alegria e a inocência com que me brindou ao longo destes primeiros anos da sua vida. Não podia terminar sem agradecer à minha namorada pela sua constante presença e apoio ao longo destes últimos anos. Finalizo agradecendo aos meus amigos pelos bons momentos que me proporcionaram.

## Índice de Conteúdos

<b>1. Introdução</b>	<b>1</b>
1.1. Empresa	2
1.2. DSI	3
1.3. Historial do workflow na Sonae	4
1.4. Trabalho realizado	5
1.5. Organização e temas abordados no presente relatório	6
<b>2. Workflow</b>	<b>8</b>
2.1. Conceitos	8
2.1.1. O que é um workflow?	8
2.1.2. Quais as características de um sistema de gestão de workflow?	9
2.1.3. O que é uma worklist?	10
2.1.4. O que é uma notificação?	10
2.1.5. O que é delegar?	10
2.1.6. O que é devolver?	10
2.1.7. O que é uma role? E uma role temporária?	10
2.2. Benefícios	11
2.3. Forma de análise	12
2.4. Erros frequentes	14
<b>3. Tecnologias</b>	<b>17</b>
3.1. Funcionalidades mínimas	17
3.2. Estado da arte	18
3.3. Oracle Workflow 2.5	20
3.4. Oportunidades de Evolução	21
3.4.1. Oracle Workflow 2.6	21
3.4.2. Real Application Clusters (RAC)	23
3.4.3. Oracle Internet Directory (OID)	23
<b>4. Metodologia</b>	<b>24</b>
4.1. Objectivos	24
4.2. Fases de um projecto	24
4.3. Arquitectura lógica	27
4.4. Manual de qualidade	28
<b>5. Processo “Comunicação de artigos”</b>	<b>29</b>
5.1. Situação actual	29
5.2. Objectivos	29
5.3. Intervenientes	29
5.4. Funcionalidades	29
5.5. Processos	30

5.5.1. Entrada em linha .....	30
5.5.2. Saída de linha .....	31
5.5.3. Suspensão .....	31
5.6. Arquitectura Lógica .....	31
5.7. Interface com o utilizador .....	32
5.8. Benefícios esperados.....	32
5.9. Implementação .....	33
5.9.1. Plano .....	33
5.9.2. Principais dificuldades.....	34
5.10. Melhorias a introduzir.....	36
<b>6. Processos: Entrada e Saída de Artigos .....</b>	<b>38</b>
6.1. Processo.....	38
6.2. Novos intervenientes.....	39
6.3. Integração com outros sistemas .....	40
<b>7. Conclusões.....</b>	<b>41</b>
7.1. Estágio .....	41
7.2. Futuro.....	41
<b>Glossário.....</b>	<b>43</b>
<b>Referências e bibliografia .....</b>	<b>46</b>
<b>ANEXO A: Direcção de Sistemas de Informação.....</b>	<b>47</b>
<b>ANEXO B: Plano do estágio .....</b>	<b>48</b>
<b>ANEXO C: O que é a WfMC? .....</b>	<b>49</b>
<b>ANEXO D: Taylor (início do workflow).....</b>	<b>50</b>
<b>ANEXO E: Questionário de levantamento de processo.....</b>	<b>51</b>
<b>ANEXO F: LDAP .....</b>	<b>52</b>
<b>ANEXO G: “Monitor” (Ferramenta Oracle para administração).....</b>	<b>53</b>
<b>ANEXO H: Exemplos de vantagens dos eventos com o exterior.....</b>	<b>56</b>
<b>ANEXO I: Exemplo de vantagem dos eventos com o utilizador .....</b>	<b>57</b>
<b>ANEXO J: Cenários RAC .....</b>	<b>59</b>
<b>ANEXO K: Suporte da solução OID .....</b>	<b>60</b>
<b>ANEXO L: Estrutura da documentação a criar .....</b>	<b>63</b>
<b>ANEXO M: Desenho de processos.....</b>	<b>69</b>
<b>ANEXO N: Interface base para projectos Workflow .....</b>	<b>72</b>
<b>ANEXO O: Interface com utilizador (parte não genérica).....</b>	<b>78</b>
<b>ANEXO P: Análise macro de entrada e saída de artigos .....</b>	<b>81</b>

**ANEXO Q: Entrada em linha ..... 82**

## Lista de Figuras

Figura 1 : Sistema de gestão de workflow.....	9
Figura 2 : Fases da análise .....	12
Figura 3 : Exemplo de confusão entre workflow e aplicacional .....	14
Figura 4 : Eliminação de falsas tarefas.....	15
Figura 5 : Funcionalidades mínimas.....	17
Figura 6 : Arquitectura tecnológica da solução Oracle workflow 2.5 .....	20
Figura 7 : Eventos (Business Event System) .....	22
Figura 8 : Fases de um projecto.....	25
Figura 9 : Arquitectura para os projectos de workflow .....	27
Figura 10 : Suspensão de um artigo.....	31
Figura 11 : Arquitectura Lógica do projecto "Comunicação de Artigos" .....	32
Figura 12 : Alteração de uma comunicação.....	35
Figura 13 : Alteração de datas.....	35
Figura 14 : Opções no "Monitor" .....	53
Figura A- 1 : Organograma da DSI .....	47
Figura B- 1 : Plano do estágio .....	48
Figura F- 1 : Comunicação LDAP .....	52
Figura G- 1 : Opções no "Monitor" .....	53
Figura G- 2 : Gestão de férias no "Monitor" .....	53
Figura G- 3 : Definição do processo "Monitor".....	54
Figura G- 4 : Definição do processo "Builder".....	54
Figura G- 5 : Procura de processos "Monitor" .....	54
Figura G- 6 : Percurso de um processo "Monitor" .....	55
Figura G- 7 : Gestão de um processo "Monitor".....	55
Figura I- 1 : Processo de alteração de datas, sem uso de eventos .....	57
Figura I- 2 : Processo de alteração de datas, usando eventos.....	57
Figura J- 1 : RAC com acesso uniforme a disco.....	59
Figura J- 2 : RAC com acesso não uniforme a disco .....	59
Figura K- 1 : Arquitectura física do OID.....	60
Figura K- 2 : Arquitectura lógica do OID (um nó) .....	60

Figura K- 3 : Funcionamento. Componentes run-time .....	61
Figura M- 1 : Desenho de processo (instalação de telefone).....	71
Figura M- 2 : Desenho de processo (solução modular) .....	71
Figura N- 1 : Cabeçalho genérico do workflow .....	72
Figura N- 2 : Worklist genérica para o workflow.....	72
Figura N- 3 : Interface entre worklist e pesquisa de notificações .....	74
Figura N- 4 : Pesquisa notificações genérica.....	75
Figura N- 5 : Corpo de uma notificação (genérico).....	76
Figura O- 1 : Selecção de lojas.....	78
Figura O- 2 : Introdução de dados sobre o artigo (entrada) .....	78
Figura O- 3 : Introdução de dados sobre o artigo (saída) .....	79
Figura O- 4 : Introdução de dados sobre o artigo (suspensão).....	79
Figura O- 5 : Facilitador para introdução de datas .....	79
Figura O- 6 : Listagens de artigos.....	79
Figura O- 7 : Pesquisa dentro das listagens .....	80
Figura P- 1 : Entrada e Saída de artigos .....	81
Figura Q- 1 : Entrada em linha.....	82

## 1. Introdução

O estado actual da tecnologia na área do *workflow* e a necessidade de desmaterialização dos processos, conduziram a um crescendo de aplicações orientadas aos processos.

Dada esta oportunidade de melhoria dos processos do negócio, os objectivos do estágio foram os seguintes:

- Integração numa equipa de projecto;
- Adquirir conhecimentos no motor de *workflow* da *Oracle*;
- Implementação do processo "Comunicação de artigos";
- Análise e desenho do processo "Entrada e Saída de artigos";
- Desenvolver metodologia para implementação alargada de projectos de *Workflow*;
- Análise do processo de implementação de projectos nesta área;
- Conhecer os processos de negócio;
- Identificar formas de melhorar a performance da tecnologia existente.

Este estágio foi pensado de forma a ir ao encontro das situações práticas no mundo do trabalho (em especial, gestão de projectos relacionados com a implementação de melhorias na área do retalho), tais como entrevistas, pesquisa de informação (*procurement*), contacto com o fornecedor e cliente. Além disso, o estágio também tem uma componente tecnológica (de forma a ir ao encontro do curso em questão), porque prevê a participação nos projectos a nível da definição de arquitecturas, estratégias de resolução do problema e análise das ferramentas usadas (com vista a detectar possibilidades de melhoramentos).

Por tudo isto, o estágio permitiu o contacto com o mundo do trabalho, com o conceito *workflow* (não existente actualmente no curso) e com os vários processos existentes (despesas de colaborador, conferência de facturas, encomenda de imobilizado, comunicação de artigos à loja).

No fim do estágio os objectivos foram cumpridos, ficando como resultados:

- Manual de qualidade;
- Sugestões de melhora a nível da tecnologia e interface do *workflow*;
- Implementação do projecto "Comunicação de Artigos" (especificação, desenvolvimento e entrada em produção);
- Análise da possibilidade de alargamento do projecto de "Comunicação de Artigos".

Embora tenham sido atingidas estas metas a nível do manual de qualidade, é necessário garantir a continuidade e controlo desse documento de forma a melhorar ainda mais o desenvolvimento dos projectos.

## 1.1. Empresa

A SONAE foi fundada em 1959, na Maia, com o objectivo de produzir termolaminados decorativos. O seu nome aparece da designação – **SO**ciedade **NA**cional de **E**stratificados, **SARL**. Durante as primeiras duas décadas cresceu como uma pequena e média empresa, centrada no sector dos produtos derivados da madeira.

Na década de 80, a empresa iniciou um processo de crescimento mais acentuado, que foi conseguido através da diversificação do negócio. Primeiro pela aquisição de uma cadeia de supermercados e depois, lançando o primeiro hipermercado em Portugal (Continente de Matosinhos). De seguida, seguindo a natural necessidade do ramo do retalho, surgiu a SONAE Imobiliária que é orientada para a construção de centros comerciais (responsável pela construção do primeiro centro comercial regional – CascaisShopping). Em simultâneo, foi feito um investimento nas seguintes áreas: comunicações, tecnologias da informação, lazer e turismo.

Em Setembro de 1999, aparece a SONAE SGPS, SA, empresa holding, engloba os seguintes sectores: retalho, imobiliário, indústria dos derivados da madeira, turismo, telecomunicações móveis e fixas, construção, media, transportes e capital de risco.

A gestão está baseada num esquema de Holdings e Sub-Holdings, ou seja, cada Sub-Holding gere o seu negócio e a Holding consolida toda a gestão, como demonstra a figura:

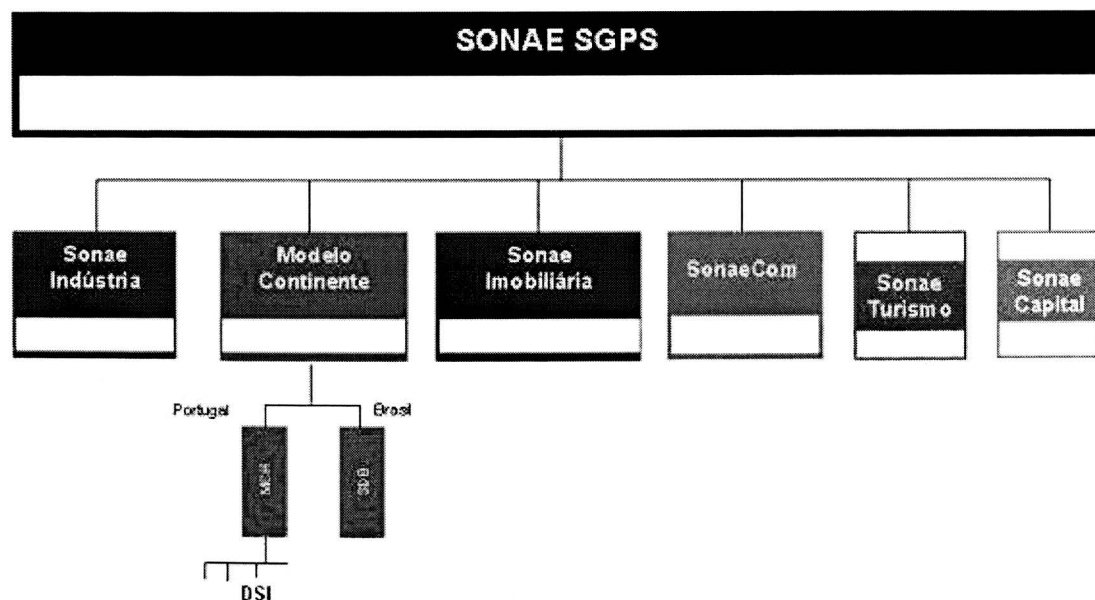


Figure 1 : Estrutura Sonae SGPS

A Sonae Indústria fica responsável por todo o sector de derivados de madeira (líderes mundiais no sector). Além disso, controla a produção dos estratificados decorativos. Controlando ainda a empresa Tafisa (painéis derivados da madeira) e a Sonae Indústria Revestimentos (bricolage, construção, mobiliário, resinas, ...).

A Sonae Imobiliária tem como objectivo a gestão dos centros comerciais, empreendimentos de lazer, investigar e criar formatos de retalho inovadores, e estudar as políticas de expansão internacional.

A Sonaecom tem como objectivo fornecer uma visão apurada das Novas Tecnologias da Informação, Telecomunicações, Internet, Multimédia e media (Público Online).

A Sonae Turismo teve a sua criação em 1994 e é fruto de uma clara aposta no sector do turismo. Tem como objectivo a gestão hoteleira, lazer e distribuição de viagens.

A Sonae Distribuição tem sob a sua alçada todas as insígnias<sup>1</sup> na área do retalho. Esta Sub-Holding detém o controlo do negócio de retalho tanto no Brasil, como em Portugal. Sendo neste último que se insere a DSI (Direcção de Sistemas de Informação).

## 1.2. DSI

A sub-holding Sonae Distribuição, por sua vez organiza-se da seguinte forma:

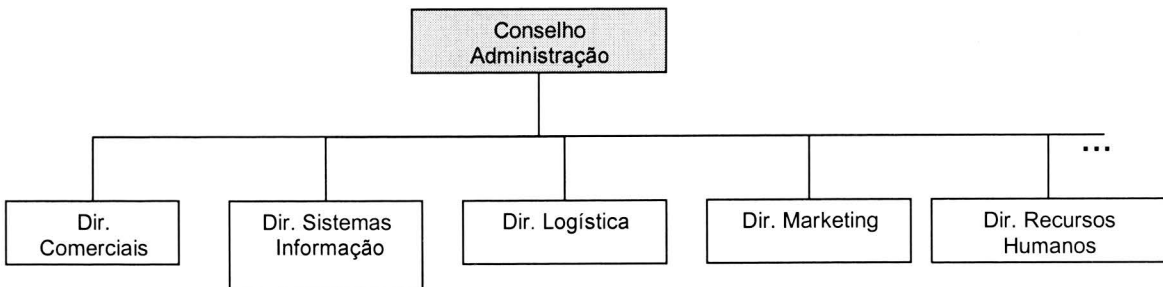


Figure 2 : Organograma Modelo Continente Hipermercados

A DSI aparece como responsável pela parte tecnológica e de análise do negócio, ou seja, é responsável pela análise dos processos com o objectivo de os tornar mais eficazes, usando para isso as tecnologias do momento. Sendo assim, cabe-lhe analisar o processo como ele é (análise As-Is<sup>2</sup>), fazer optimizações ao processo (análise To-Be<sup>3</sup>), escolher a tecnologia mais apropriada, criar a documentação necessária, gerir o desenvolvimento do projecto e garantir apoio pós-implementação.

Para que a DSI consiga responder de forma eficiente a todos os problemas que lhe são colocados, tem a seguinte estrutura (ver ANEXO A):

- **Análise de Negócio**

Esta área é responsável pela gestão da relação funcional com outras direcções, com o objectivo de gestão de contas internas. Além disso, gere alguns projectos de sistemas

- **Centros de Competência**

São responsáveis pelos projectos dentro da sua área e por garantir suporte e consultadoria nas suas tecnologias. Na DSI existem 3 centros de competência nas

<sup>1</sup> São uma espécie de marcas associadas à SONAE, na área do retalho

<sup>2</sup> Estudo do processo tal como ele se desenrola

<sup>3</sup> Criação de um novo processo, com o objectivo de melhorar o existente

seguintes áreas: SAP<sup>4</sup>, RETEK<sup>5</sup> e WEB. Este último, além de ser responsável pela infra-estrutura WEB, é responsável pelas Bases de Dados, o *Data Warehouse* e projectos de workflow. Todos os outros centros de competência, são responsáveis pelo produto que lhes deu o nome (SAP e RETEK)

- **Produção e Suporte**

Esta área é responsável pelo apoio às aplicações existentes, de forma a garantir que a aplicação continua a satisfazer os requisitos. Além disso, é responsável pela definição das políticas de produção

- **Tecnologia**

Esta área é responsável pelos servidores existentes e infra estruturas de telecomunicações e tem como objectivo determinar falhas e carências a nestes níveis

- **Arquitectura da Empresa**

Esta área é responsável por analisar a arquitectura da rede de aplicações existentes na Sonae Distribuição, de forma a garantir que estão de acordo com a estratégia prevista

- **Desenvolvimento Organizativo e Controlo**

Esta área é responsável pela definição das agendas das reuniões, gestão da documentação geral, controlo dos custos usando o PANDA<sup>6</sup> e são ainda responsáveis pelo Plano de Actividades e Recursos.

### 1.3. *Historial do workflow na Sonae*

Na Sonae a génese do *workflow* fica, em parte a dever-se à criação da Intranet (em 1999), porque só com essa plataforma é que se criou um local central de partilha de informação.

Em Novembro de 2000, começa a caminhada do *workflow*, que teve como primeiras acções a escolha da empresa fornecedora, e principalmente a escolha da tecnologia. A tecnologia escolhida foi *Oracle Workflow* em conjunto com o IIS (*Internet Information Services*) já usado na Intranet.

Em Abril de 2001, é criado o primeiro protótipo de um projecto de workflow (Serviço de conferência de facturas (SCF), versão 1), que tinha como objectivo testar a tecnologia e a adaptação da organização a esta nova situação. No fim desse mesmo ano, e devido ao sucesso do SCF, foi criada a versão 2 que teve um visual renovado. Além do visual, esta nova versão viu alterada a linguagem de programação da camada de apresentação de PSP (*Procedure Server Pages*) para ASP (*Active Server Pages*).

Em 2002 continua a caminhada, pois entrarão em produção os seguintes projectos de Workflow: Comunicação de Artigos, Direcção de Desenvolvimento de Lojas, Despesas dos Colaboradores, Sistema Central de Reparações, entre outros.

<sup>4</sup> Software de apoio à empresa na área de: recursos humanos, financeiro e contabilidade

<sup>5</sup> É um ERP especializado à área do retalho

<sup>6</sup> Aplicação que armazena os custos associados a cada projecto

Outros se seguirão sem nunca esquecer os seguintes princípios: oportunidade de melhoria nos processos actuais (controlo, produtividade e eficiência), digitalização de processos (desmaterialização - eliminação dos documentos em suporte físico e passagem a suporte digital) e satisfação do utilizador.

#### **1.4. Trabalho realizado**

O trabalho realizado no âmbito do estágio (ver ANEXO B), teve como início uma fase de integração na empresa. Essa fase de integração teve como objectivos: introduzir o negócio (retalho), dar conhecimento dos sistemas existentes, mostrar o papel da Direcção de Sistemas de Informação e apresentar os colegas de trabalho. Essa fase durou cerca de um mês e teve as seguintes acções:

- Visitas no terreno (3 dias no Continente de Gaia (Continente + Worten), Entreposto da Maia, Modelo Bonjour de Matosinhos, Novis);
- Visitas a algumas direcções (Bazar ligeiro, Têxtil, Espaço, Recursos Humanos);
- Reuniões individuais com cada elemento da DSI e incorporação na área de trabalho (Centro de Competências WEB / MIS /DW);
- Leitura de documentação que permitisse apreender o que já estava a ser feito e como. Essa leitura foi complementada através do estudo de casos de sucesso conhecidos.

Além do referido anteriormente, a integração foi enriquecida com uma formação com a *Oracle* (do qual saiu um documento intitulado “Workshop Tecnológico Sonae - Link – Oracle”), que tinha como objectivo analisar o estado actual da tecnologia e as funcionalidades das novas versões.

Foi durante a integração que começou uma das tarefas do estágio, o projecto “*Comunicação de Artigos*”. Com ele foi necessário conhecer todo o fluxo de comunicação entre as direcções comerciais e as lojas, assim como o significado dos dados associados a cada artigo. Depois de assimilada esta informação relativa ao negócio, foi necessário implementar o projecto. Para que a implementação do projecto fosse garantida com sucesso, foram realizadas as seguintes tarefas:

- Realização da especificação funcional, que foi obtida através de reuniões com o cliente;
- Desenho dos processos em *Oracle* que permitissem implementar o processo em questão, assim como a forma de resolução de perfis. Para a implementação deste último ponto, foi necessário proceder ao estudo do *OID* da *Oracle*, protocolo *LDAP* e alguns exemplos de implementações bem sucedidas;
- Desenho da maquete que exemplificasse a interface ao utilizador e a navegação pretendida;
- Especificação e validação de um protótipo desta aplicação;
- Especificação dos testes de aceitação e execução dos mesmos, de forma a garantir a qualidade da aplicação antes da sua entrega ao cliente;

- Acompanhamento, validações pontuais e documentação de passagem a produção.

Sendo assim, este projecto requereu o uso dos conhecimentos na área de arquitectura de software e gestão de projectos.

Paralelamente, foi feita a análise da possibilidade de modelar o processo de entrada e saída de artigos como um todo, mas sem ir ao pormenor, pois o objectivo era estudar a possibilidade de alargamento (ex: expansão aos fornecedores) do processo “*Comunicação de Artigos*”. Para isso, foram feitas algumas entrevistas a elementos do negócio.

Entretanto, e com o apreender do que realmente acontecia a nível do *workflow*, surgiu a necessidade de criar um modelo que permitisse regular e melhorar o desenvolvimento destes projectos. Por isso, foi iniciado o desenvolvimento do Manual de Qualidade que cobre as especificidades destes projectos usando para validação o projecto “*Comunicação de Artigos*”.

O estágio foi enriquecido com a inserção nos outros projectos de *workflow*, de forma a por um lado conhecer melhor o que está a ser feito (conhecimento do negócio e os projectos de *workflow*), e por outro lado garantir a verificação das normas de qualidade.

### **1.5. Organização e temas abordados no presente relatório**

O relatório começa com um ponto introdutório, começa com uma descrição dos objectivos do estágio, seguindo-se a contextualização do estágio na estrutura da empresa. Refere também o que já foi feito nesta área (*Workflow*) ao longo do tempo. Terminando com a descrição do trabalho realizado e uma descrição do presente relatório.

No ponto seguinte apresentam-se as particularidades do *workflow*, criando assim uma base teórica acerca do tema. Este ponto é resultado da leitura das normas existentes e da compilação de casos de estudos conhecido.

Segue-se um ponto intitulado “Tecnologias”, que discute o que deve existir nas ferramentas nesta área e compara algumas soluções existentes no mercado. De seguida, acrescenta-se a descrição da ferramenta escolhida pela empresa. Terminando com as melhorias que podem ser introduzidas na tecnologia existente actualmente.

No ponto “Metodologia”, é explicada toda a metodologia a seguir nestes projectos de forma a atingir os objectivos propostos. Para isso, descrevem-se as fases que cada projecto deve passar e qual a arquitectura tecnológica a utilizar nos desenvolvimentos. Termina com uma breve apresentação do instrumento responsável por garantir a obtenção dos objectivos propostos, ou seja, descreve o “Manual de Qualidade” criado para esta área de desenvolvimento.

De seguida, no ponto “Comunicação de artigos”, é apresentado o resultado da utilização desta metodologia num projecto concreto. Para tal, é descrito o processo de negócio associado, os problemas e os benefícios retirados da digitalização deste projecto.

Em seguida é apresentado um estudo das possibilidades de alargamento do projecto referido anteriormente, mostrando quais as novas funcionalidades a incorporar e quais possíveis ganhos.

O relatório termina, com as conclusões retiradas do estágio, referindo no que fica para fazer num futuro próximo.

Para enriquecer o presente relatório, foi criado um glossário de forma a melhorar a compreensão dos termos utilizados.

## 2. Workflow

O *workflow* aparece com a revolução industrial, mais precisamente como resultado da teoria de trabalho criada por Fredrick Winslow Taylor. Essa teoria aponta como forma de atingir maior eficiência a especialização dos intervenientes.

Ao longo de vários anos, as aplicações criadas suportavam algumas funcionalidades do *workflow*. Por evolução dos negócios foi reconhecida a necessidade de se criarem ferramentas que suportassem/englobassem todas essas funcionalidades de uma forma flexível. Sendo assim, as funcionalidades a suportar são:

- Processamento de imagem, porque em alguns casos é necessário ter documentos digitalizados;
- Gestão documental, de forma a armazenar os documentos digitalizados e fornecer, além do suporte físico, também funcionalidades de pesquisa e alteração;
- Distribuição de informação, ou seja, suportar um canal que permite encaminhar processos a intervenientes remotos;
- Suporte transaccional, para permitir interação entre várias plataformas
- Desenho de processos, de forma a suportar a modelação dos processos (fluxos, perfilagem, eventos, ...).

Devido às ferramentas de *workflow* nascerem da conjunção de várias aplicações e com propósitos diferentes, surge a necessidade de normalizar garantindo a interoperabilidade entre ferramentas. Com essa necessidade nasce a WfMC (*Workflow Management Coalition*) (ver ANEXO C).

### 2.1. Conceitos

De forma a compreender melhor as propriedades do *workflow*, há que conhecer os conceitos que se seguem. Dadas estas ferramentas terem especificidades próprias, faz com que pertençam a uma classe à parte.

#### 2.1.1. O que é um *workflow*?

Um *workflow*, tal como o próprio nome indica é um fluxo de trabalho necessário para realizar uma tarefa. Mas nem tudo pode ser interpretado como *workflow*, ou seja, para que seja um *workflow* há que satisfazer os seguintes requisitos:

- Existir uma tarefa bem definida
- Essa tarefa estar dividida em subtarefas/passos, que podem ser executadas em sequência ou em paralelo
- Serem precisas pelo menos duas pessoas para realizar essa tarefa
- Existir um objectivo comum, ou seja, o realizar da tarefa

Em conclusão, um processo só deve ser tratado como *workflow* se estiver dividido e necessitar de vários elementos para a sua realização.

Exemplos: Ordens de compra, Aprovação de pedidos de capital, Queixas, Conferência de Facturas, Entrada e Saída de artigos em produção, Cadeia de Abastecimento (*Supply Chain Management*), ...

### 2.1.2. Quais as características de um sistema de gestão de *workflow*?

Um sistema de *workflow* é uma plataforma que define, controla e executa encaminhamento de processos com base numa representação digital do processo. Por isso, deve prever a existência dos blocos apresentados na figura.

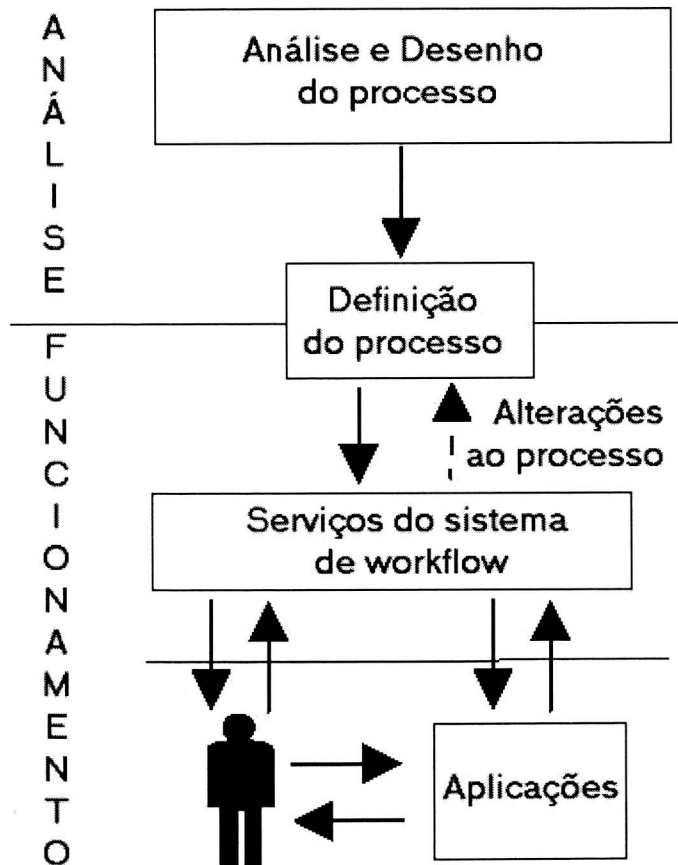


Figura 1 : Sistema de gestão de *workflow*

Sendo assim, e conforme a figura, o sistema terá de prever dois grandes blocos que são respectivamente a análise e o funcionamento.

No bloco de análise, deve existir uma ferramenta que permita modelar o processo a implementar. Essa modelação ser uma funcionalidade fácil de utilizar, de forma a ser fácil recriar a realidade por qualquer pessoa, mesmo sem conhecimentos técnicos apurados.

No bloco de funcionamento, deve existir um motor que controle o decorrer de cada processo lançado. Deve ainda existir forma de o utilizador interagir com o motor (directa ou indirectamente). De forma a facilitar a interação deverá existir uma camada que forneça algumas funcionalidades (exemplo: Aplicativos de gestão, Análise de histórico, ...) e que permita a interação com outras aplicações.

Entre estes dois blocos, deverá existir um módulo que permita criar a definição do processo, ou seja, criar a representação digital do processo modelado anteriormente, de forma a ser interpretado pelo motor.

### 2.1.3. O que é uma *worklist*?

Uma *worklist* contém as acções que são requeridas ao utilizador para que o processo continue. Sendo assim, como o workflow é orientado ao encaminhamento dos processos o ecrã principal de uma aplicação destas deve ser a *worklist*. Isto corresponde a um factor diferenciador de todas as outras aplicações e deve servir como alerta para quem vai efectuar o design da aplicação.

### 2.1.4. O que é uma notificação?

Uma notificação é o mecanismo utilizado para requerer a intervenção do utilizador. Essa notificação pode ser um mail ou uma informação via *worklist* convencional. Uma notificação é constituída por um assunto, corpo e resposta (quando necessária).

### 2.1.5. O que é delegar?

Uma delegação é um instrumento que permite a um utilizador enviar a responsabilidade de tratamento de uma notificação para outro utilizador. Esta funcionalidade torna-se útil quando o utilizador vai de férias ou não se sente capaz de tratar aquela notificação.

### 2.1.6. O que é devolver?

Uma devolução é um instrumento que permite a um utilizador enviar a notificação para o estado anterior. Por exemplo: uma factura foi conferida e está neste momento à espera de aprovação, mas o aprovador decide que não está bem conferida, então devolve ao utilizador anterior para rever a sua tarefa.

### 2.1.7. O que é uma *role*? E uma *role* temporária?

Uma *role* é o correspondente ao português “papel/função”. Sendo assim, uma *role* é um conjunto formado por utilizadores com características semelhantes, ou seja, têm privilégios e restrições iguais. Por isso, no limite uma pessoa corresponde a um papel (exemplo: *role* orientadores de estágio contém Carlos Latourrette, Eugénio Oliveira e Paulo Ribeiro e a *role* estagiário corresponde a uma só pessoa). Uma *role* permite controlar melhor os envios para utilizadores, quando os mesmos trocam muito de função (exemplo: se criarem um *role* director de loja com um mail específico, quando o director de loja mudar (outra pessoa), não existe necessidade de alterar o processo. Apenas é necessário cadastrar correctamente o utilizador).

Uma *role* temporária é uma técnica utilizada no desenvolvimento de projectos de *workflow*, que serve para evitar estar sempre a procurar as pessoas que desempenham uma função, pertencente aquela instância do processo. Por exemplo: num estágio é necessário marcar reuniões, sendo assim, no início de um estágio cria-se um *role* com todos os intervenientes de forma a se poder marcar reuniões sem nos preocuparmos com o mail de cada pessoa (espécie de *mailing list*<sup>7</sup>).

---

<sup>7</sup> Lista de endereços de correio electrónico

Concluindo, são estes conceitos que fazem com que o *workflow* seja uma aplicação diferente das restantes. Sendo que, por vezes o *workflow* deve ser enriquecido com funcionalidades que não englobam nenhum destes conceitos.

## 2.2. Benefícios

O que faz com que as aplicações de *workflow* sejam importantes para o negócio são os benefícios que permitem atingir. Esses benefícios são obtidos nas seguintes áreas:

- **Manipulação de informação**

A manipulação é melhorada devido à diminuição do papel que circula nas actividades a desempenhar e que será digitalizada facilitando o transporte, pesquisa e cálculo sobre os dados.

Além disso, a informação ficará centralizada em bases de dados em vez dos duplicados usados em papel, sendo menor a possibilidade de entrada de erros. Muitas vezes ainda, o benefício tirado reside no facto de a aplicação de *workflow* difundir a informação por vários sistemas diferentes, sendo o contrário também possível.

- **Controlo**

Nesta área, o benefício tirado reside no facto de se poder localizar o estado de cada processo ao longo do *workflow* definido, ficando a saber-se os prazos/tempos por actividade e as pessoas que realizaram as tarefas. Além disso, é garantido que os processos passam por todas as actividades, assegurando que não existem fugas ao sistema.

- **Redução de tempos**

A redução do tempo de execução de um processo, diminuiu devido ao tempo da comunicação do documento digitalizado ser menor, pela automatização de certas actividades, pela integração entre os sistemas (menor necessidade de introdução de dados) e por vezes pela diminuição das actividades a realizar e/ou paralelização das mesmas. Além disso, a especialização leva a uma maior rapidez de realização de actividades devido ao estabelecimento de rotinas.

- **Segurança**

Nesta área, os ganhos são: a garantia de não violação da informação, a garantia de quem realizou a tarefa e a garantia de não se perderem processos (o máximo que poderá acontecer é caírem em erro, podendo depois disso ser recuperados).

- **Valorização pessoal**

A um nível menos tecnológico, os ganhos advêm de se libertarem as pessoas para efectuar só as tarefas que necessitem de algum conhecimento. Além disso, consegue-se ser mais justo na avaliação do desempenho de cada um, pois é possível verificar a eficiência de cada actividade, fazendo com que as pessoas sintam que o seu trabalho será valorizado.

Em conclusão, os benefícios retirados são evidentes embora possam haver problemas levantados pela falta de formação e resistência à mudança (medo da perda de poder e de se ser controlado).

### 2.3. Forma de análise

Embora os benefícios/ganhos sejam evidentes, só serão atingidos se existir cuidado durante a fase de análise. Sendo assim, há que respeitar as fases representadas na figura.

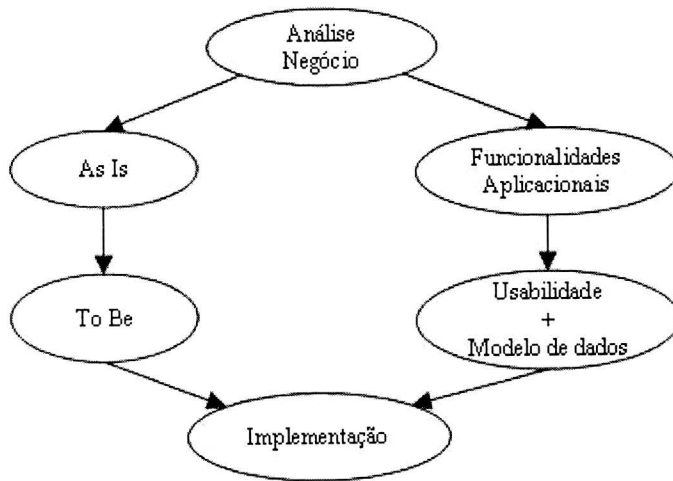


Figura 2 : Fases da análise

Em cada fase há que ter os seguintes cuidados:

- **Análise Negócio**

O objectivo é criar a possibilidade de partilha de conhecimento entre os elementos do negócio e os elementos que serão responsáveis pela implementação tecnológica.

Sendo assim, cada um dos elementos tem de ter em conta o não conhecimento de cada uma das áreas. Por isso, a pessoa do negócio terá que ser o mais precisa possível na explicação do que é necessário criar.

Quanto aos responsáveis pela implementação tecnológica, devem preparar uma apresentação da tecnologia em questão de forma a demonstrar quais as funcionalidades disponíveis (podendo assim a pessoa do negócio ficar com uma noção dos limites). Além disso, devem preparar um questionário de forma a garantir que conseguem definir todo o processo. Esse questionário deverá ter em conta que consegue definir todas as regras, entradas, saídas, eventos (relação com o exterior) e perfis de execução para cada actividade. Adicionalmente, deverá conter algumas questões que permitam realizar uma análise de ganhos de forma a se conseguir regular as expectativas do cliente e contabilizar as melhorias introduzidas. No ANEXO E, encontra-se um questionário que demonstra o que foi descrito.

É ainda aconselhável existir um terceiro elemento que faça de mediador entre os dois elementos anteriores, saiba gerir expectativas e garanta o atingir de metas. Esse terceiro elemento deverá ter alguns conhecimentos de ambas as áreas.

Desta interacção resultam dois ramos, que embora separados devem ser analisados globalmente (de forma a garantir coesão da aplicação final). Um dos ramos preocupa-se com a completa definição do processo da seguinte forma:

- **As Is**

Deste tipo de análise, devem resultar diagramas que permitam ao representante do negócio validar que o processo está entendido.

Sendo assim, os diagramas devem modelar apenas o que acontece na realidade e não introduzir nenhuma alteração. Além disso, estes diagramas devem-se abster de conter actividades/estados que sejam relativos a cuidados a ter na implementação, tais como sincronização, carregamento de dados e outros parecidos.

- **To Be**

Esta fase terá como entrada os diagramas anteriores, sendo o seu resultado um conjunto de diagramas a usar na definição do processo. O valor acrescido nesta fase é a optimização do processo, que passa por aproveitar a entrada desta nova tecnologia para implementar alterações ao processo em si. Nestes diagramas, já podem ser colocados actividades/estados que sejam relativos à implementação. Embora nos casos em que a optimização “desfigurou” o processo, não seja aconselhável introduzir esta informação, sob pena de não se entender o valor acrescido na optimização.

No outro ramo, a preocupação é centrada na interacção com utilizador (usabilidade e funcionalidades acessórias) da seguinte forma:

- **Funcionalidades aplicacionais**

Nesta fase o objectivo é definir quais as funcionalidades extra processo, avaliando os impactos no desenho do mesmo. À semelhança do que aconteceu aquando do desenho do processo, há que especificar quais os perfis de acesso e os dados necessários. É de garantir que os dados necessários ao encaminhamento do processo, apenas são controlados pelo próprio desenrolar do processo.

- **Usabilidade + Modelo de dados**

Nesta fase, o objectivo é garantir o suporte às funcionalidades tanto a nível de dados necessários com ao nível da interacção com o utilizador.

A nível dos dados há que ter em conta os seguintes pontos: garantir que não existe de informação redundante (necessário criar um equilíbrio entre o espaço ocupado em disco e a facilidade dada à introdução de dados – listagens de valores possíveis); flexibilidade do modelo de dados, permitindo assim suporte a novas funcionalidade e prevendo alterações

tais com o IVA (Imposto de Valor Acrescentado); e as necessidades de pesquisa de forma a garantir rapidez de acesso à informação.

A nível da usabilidade, há que ter em conta os seguintes pontos: acessibilidade à informação de forma intuitiva; fácil aprendizagem; prever formas de orientar o utilizador ao longo da interacção com a aplicação, de forma a melhorar cada vez mais a sua utilização; e reduzir os erros introduzidos.

- **Implementação**

Finalmente, segue-se a implementação que, caso a análise tenha sido bem conduzida, não terá problemas de maior. Apenas será necessário documentar tudo o que foi implementado e garantir que essa implementação é feita de forma modular, possibilitando melhor compreensão e reutilização das soluções, tendo sempre em conta o uso correcto da ferramenta.

## 2.4. Erros frequentes

Nos projectos de workflow há que estar atento para não se cometerem os seguintes erros:

- **Confundir *workflow* com funcionalidades aplicacionais**

Há que ter sempre presente que o *workflow* é o centro da aplicação, e sendo assim, tudo deve decorrer em função dele, de forma a garantir a passagem pelas actividades. Mas, o erro mais frequente é dar à camada aplicacional responsabilidades que são do *workflow*, tais como, determinação de destinatários e resposta às actividades e alteração de dados do processo.

Por exemplo, se for necessário notificar um conjunto de pessoas de uma direcção, a solução a adoptar deverá ser a que permite que o motor consiga resolver quais os destinatários, em detrimento de criar um repositório genérico de notificações que só em função de uma camada suplementar de parametrização (ou seja, uma camada aplicacional) faz chegar a notificação ao destinatário (ver figura seguinte)

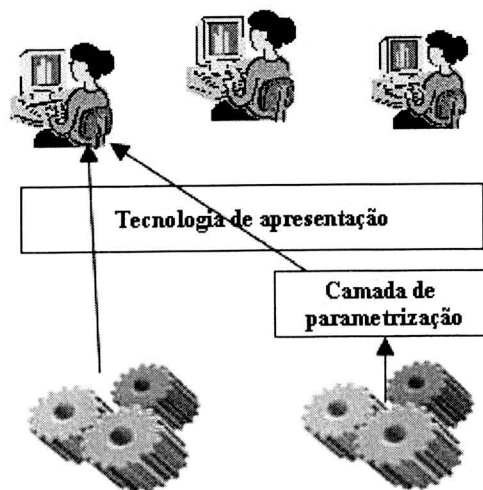


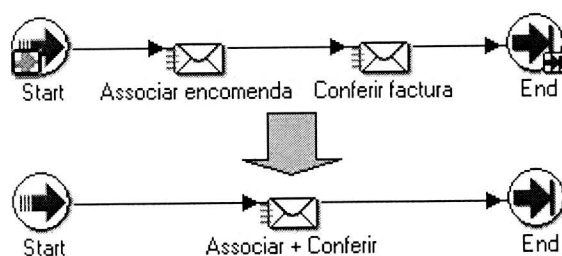
Figura 3 : Exemplo de confusão entre *workflow* e aplicacional

- **Controlo do que entra na *worklist***

Neste ponto, o erro mais frequente é o facto de aparecerem notificações que não podem ser tratadas, ou seja, quando se responde aparece novamente na *worklist*, criando assim uma falsa noção de trabalho. Noutros casos, o erro é semelhante e consiste em ter notificações sucessivas para a mesma pessoa.

Em conclusão, terá de existir uma especial atenção com a granularidade das notificações e com as falsas tarefas (que correspondem geralmente a situações de acções não obrigatórias que pela definição de *workflow*, não são suportadas).

Por exemplo, antes de conferir uma factura, é necessário associar contrato (caso exista) e só de seguida conferir. Para que não se caia neste erro, deve-se juntar numa só notificação as funcionalidades de conferência e associação de encomenda. Com esta alteração eliminamos a falsa noção de resolução da notificação (ver figura com resolução)



**Figura 4 : Eliminação de falsas tarefas**

Senão vejamos, no processo existente na parte superior da figura anterior, um utilizador recebe uma notificação para associar contrato/encomenda, mas depois de associar, em vez de ver terminada a sua intervenção no processo, recebe outra notificação para conferir, ou seja, não é garantida a usabilidade desta acção (tem demasiadas passagens de ecrãs). Enquanto que no processo abaixo desse, o utilizador desenvolve todo o seu trabalho de uma só vez. Além disso, a delegação torna-se confusa, porque se delegar a primeira notificação, quando o utilizador (receptor da delegação) tratar a notificação, ela aparecerá novamente na *worklist* do utilizador que delegou inicialmente.

- **Nos perfis, usar funções em vez de utilizadores**

Um erro que é geralmente cometido nas empresas é modelar o *workflow* de forma a detectar os destinatários pelos nomes em vez do seu cargo. O erro reside no facto de as pessoas “rodarem” de forma muito frequente pelas várias funções, fazendo com que o *workflow* esteja sempre desactualizado. De forma que isto não aconteça, dever-se-á na medida do possível definir os perfis associados a cargos, garantindo assim a actualização por mais tempo (até o cargo ser extinto).

Por exemplo, num processo qualquer é necessário notificar o director da loja de Matosinhos que hoje é o senhor António, mas no futuro será outra pessoa. A solução a implementar será criar um utilizador fictício chamado director de loja e notificar esse utilizador, garantindo assim que o *workflow* está sempre

actualizado. Quando o director de loja mudar, é só ir a esse utilizador fictício e associar a nova pessoa.

- **Não garantir modularidade (processos e aplicação)**

Outro erro que se deve evitar é não dividir a aplicação de forma modular. A falta dessa preocupação tem as seguintes consequências: inibe a possibilidade de reutilização (porque não tem as funcionalidades separadas), inibe a boa compreensão (não permite assimilar a arquitectura da aplicação em fases) e torna os processos menos flexíveis tornando impossível futuras alterações (uma alteração é sentida por toda a aplicação, ou seja, não se podem fazer alterações localizadas).

- **Simbologia incoerente**

Nos projectos de *workflow* são utilizados muitos diagramas, e por isso, deve existir a preocupação de usar sempre a mesma simbologia, de forma a facilitar a compreensão entre as várias equipas.

- **Não documentar**

Finalmente, um projecto pode correr muito bem, mas se não estiver documentado será impossível de manter (porque por vezes os diagramas são de grande dimensão). E por melhor que esteja, mais tarde ou mais cedo terá de evoluir. Além disso, serve para introduzir as novas pessoas num projecto, podendo mesmo até retirar lições do que aconteceu.

Em conclusão, deve existir um enorme cuidado em não incorrer nestes erros porque afectam a robustez da aplicação, impossibilitando a correcta manutenção e a evolução da aplicação.

### 3. Tecnologias

Neste ponto, pretende-se apresentar os aspectos tecnológicos associados ao *workflow*. Sendo assim, far-se-á uma análise das funcionalidades necessárias, comparação de algumas ferramentas disponíveis no mercado, descrição da ferramenta *Oracle Workflow* (adoptada pela Sonae Distribuição) e algumas sugestões para melhoria da tecnologia existente.

#### 3.1. Funcionalidades mínimas

Quando se pensa numa ferramenta de *workflow*, está-se a pensar não só no motor mas também em funcionalidades de desenho, entre outras.

Para criar uma base para discussão do valor de cada ferramenta, as funcionalidades mínimas a encontrar são as seguintes (como demonstra a figura):

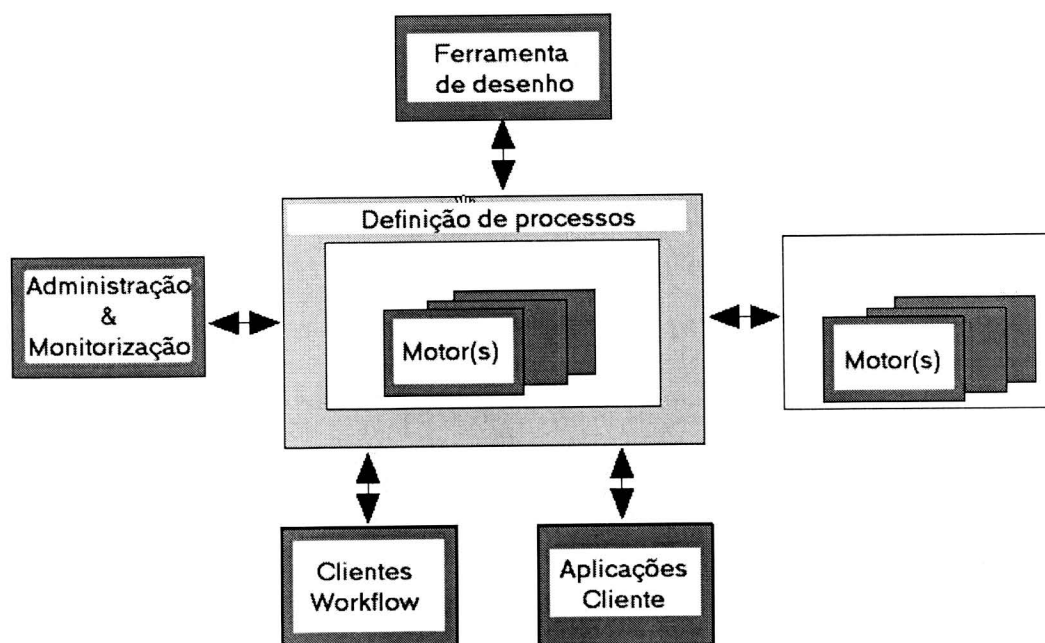


Figura 5 : Funcionalidades mínimas

- **Ferramenta de desenho do processo**

Uma das funcionalidades importantes é a existência de uma ferramenta que permita desenhar o processo. Essa ferramenta permitirá uma melhor análise visual dos processos e um ganho de tempo na programação do motor de *workflow*.

- **Ferramenta de definição do processo**

Esta funcionalidade consiste em transformar o desenho do processo numa representação compreendida pelo motor de *workflow*, e vice-versa. Só assim, a ferramenta de desenho terá utilidade.

- **Base de dados**

O motor deverá conter uma base de dados que lhe permita armazenar os dados necessários controlo do processo. A informação a armazenar é relativa a perfis, regras, histórico, estado actual e dados associados ao negócio.

- **Interligação com o exterior**

O motor de *workflow* deve prever formas de interacção com o exterior nas seguintes vertentes: com outros motores de *workflow*, com outras aplicações (*ERP*, ...) e com aplicações criadas pelo cliente.

- **Ferramenta de administração / monitorização**

Esta ferramenta servirá para efectuar a análise de cada processo a correr no motor. Sendo assim, deverá conter as seguintes funcionalidades: gestão de utilizadores e perfis, possibilidade de auditar o funcionamento, localizar, desbloquear e controlar os níveis de prioridade dos processos em execução.

Conhecendo as funcionalidades mínimas a garantir, segue-se a análise de algumas possibilidades existentes no mercado.

### 3.2. Estado da arte

Identificadas as funcionalidades mínimas, é necessário identificar quais serão os outros pontos a ter em conta. Sendo assim, teremos de considerar capacidades a nível da base de dados, integração com outras aplicações (*Word*, *Excel*, ...), existência de exemplos que garantam segurança, funcionalidades extras fornecidas, esforço de desenvolvimento previsto, plataformas possíveis para suportar o motor (*Unix* e/ou *Windows*), custo da ferramenta e a adequação com a situação da empresa.

As ferramentas escolhidas para comparação foram as seguintes: o *Oracle workflow* porque é a ferramenta adoptada pela Sonae Distribuição; o motor de *workflow* da SAP porque usamos na empresa parte desse *ERP*; *OpenCS* escolhida ao acaso entre as pertencentes da *WfMC* e *Ultimus* que é uma empresa que não pertence à *WfMC*.

Depois de definida a amostra e os factores mais importantes a considerar, o estado actual da arte em cada ponto é o seguinte:

1. Embora as funcionalidades mínimas estejam presentes na maioria das soluções existentes no mercado, as ferramentas de desenho e definição não seguem essa regra. Por exemplo, a ferramenta da *OpenCS* não suporta o desenho e a definição.
2. Em relação às bases de dados, existem 3 tipos de apostas actualmente: soluções com bases de dados proprietárias, com a vantagem do custo ser menor (existem muito poucas); soluções que apostam em uma base de dados, com a vantagem de se especializarem nessa e consequentemente criar mecanismos que garantam melhor performance (*Oracle workflow* usa exclusivamente *Oracle* e *OpenCS* usa *SQL Server*); outras soluções que usam qualquer base de dados, quer incorporando (exemplo *SAP*) quer por *ODBC* (*Ultimus*).

3. Acerca da integração com o exterior, existem as seguintes implementações no mercado: soluções que usam memória partilhada para comunicar, *RFC*<sup>8</sup>s, o uso de *XML* e criação de agentes especializados.

Em relação aos *RFC*'s (caso do *SAP*), existe a desvantagem de ser necessário desenvolver em ambas as aplicações em simultâneo, ou seja, se quisermos integrar *SAP* com *Oracle Workflow* teremos de desenvolver funções em ambas as aplicações para conseguirem comunicar.

O *XML* (caso do *Oracle Workflow*), tem a vantagem de ser aceite pela maioria das aplicações, devido a ser normalizado.

Finalmente, uma solução que está a aparecer no mercado é a criação de agentes especializados na comunicação entre ferramentas, permitindo de forma fácil integrar funcionalidades de aplicações externas (caso do *Ultimus* existem os "*Flobots*" – *Workflow Robots*).

4. Em relação a existirem exemplos de implementação, cada vez existem mais casos de estudo sobre projectos de workflow que servem para complementar a qualidade das soluções (exemplo referências: British Telecom, CERN, Hong Kong Telecom, McKesson – área de medicina, WA Water & Rivers Commission - entidade governamental, ...).
5. Em relação aos extras que podem ser encontrados no mercado, temos: ferramentas de desenho de formulários directamente para *ASP* ou outras tecnologias (caso *Ultimus*), incorporação de um gestor de perfis (todas as escolhidas têm excepto *OpenCS*) e ferramentas de monitorização enriquecidas com gráfico e relatórios automáticos (caso do *SAP* e *Ultimus*).
6. O esforço de desenvolvimento é cada vez menor, devido às ferramentas acessórias que são disponibilizadas pelas soluções de *workflow* existentes. Pegando nas ferramentas enunciadas, temos no caso do *Ultimus* uma ferramenta de desenho de interfaces com o utilizador e no *Oracle* e *SAP* temos templates de processos (desenhos pré-definidos) embora com granularidade deferente. Por isso, em cada caso ter-se-ão de analisar quais as vantagens tiradas de cada uma das ferramentas acessórias disponíveis.
7. Em relação às plataformas, existe uma larga escolha no mercado, embora poucas sejam multiplataforma (*Oracle Workflow* e *SAP* são multiplataforma, mas *Ultimus* só suporta *Windows*).
8. Sobre o custo, e tendo em conta o mercado há que ter em conta os seguintes custos: compra da solução, bases de dados, condições de evolução, esforço previsto de desenvolvimento, necessidades de formação e manutenção e criação da plataforma base (*hardware* e *software*).
9. Na adequação devemos identificar onde estão os dados principais de forma a não se perder tempo em integrações desnecessárias (por exemplo, na companhia não existe uma solução de *SAP* completa, por isso, os

<sup>8</sup> Remote Function Call – chamada de funções remotamente entre duas aplicações

ganhos não seriam muitos. Pois, o motor da *SAP* tira as maiores vantagens nesses casos).

Concluindo, não existe a ferramenta ideal, pois cada caso será um caso, ou seja, cada empresa deverá identificar o que é mais importante. Mesmo assim, será sempre uma escolha difícil porque a oferta é enorme.

### 3.3. Oracle Workflow 2.5

Dado que a ferramenta escolhida para suportar os processos na Sonae Distribuição foi o *Oracle Workflow*, há que conhecer bem qual a arquitectura de forma a se poder tirar todo o partido da ferramenta. A arquitectura tecnológica é representada na seguinte figura.

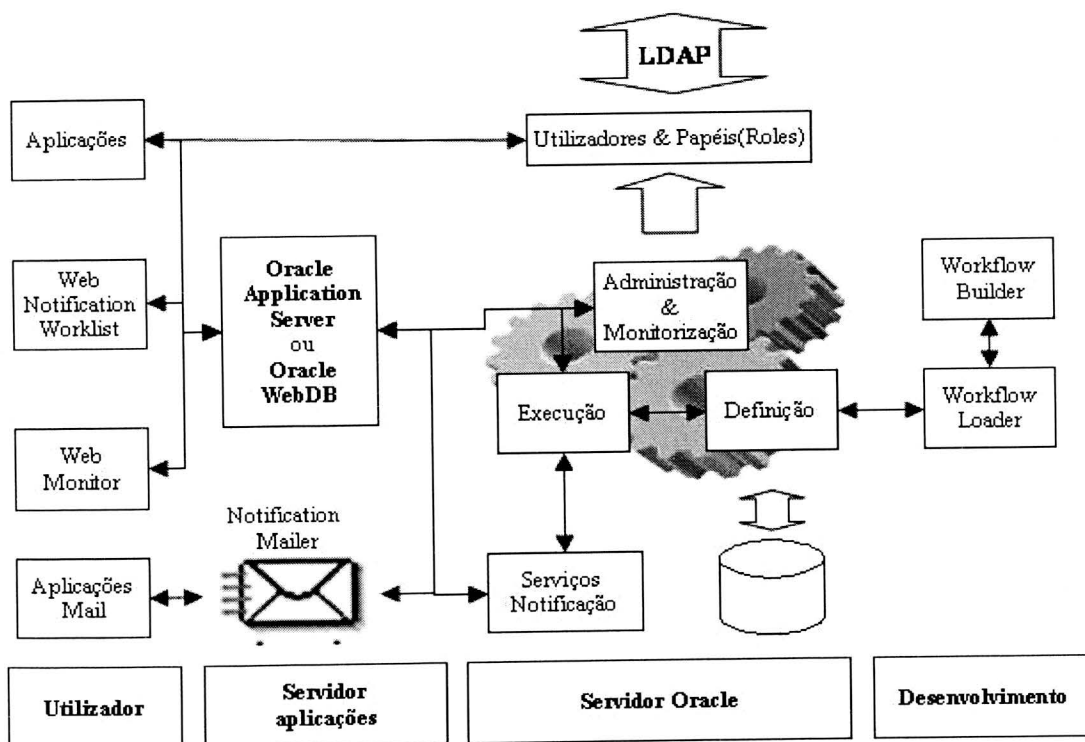


Figura 6 : Arquitectura tecnológica da solução *Oracle workflow 2.5*

As funcionalidades fornecidas encontram-se divididas em quatro grandes blocos, que são:

- **Desenvolvimento**

Neste bloco, a preocupação incide na parte de representação do processo na forma digital, ou seja, de forma a poder ser entendido pelo motor de *workflow*. Para isso, contem duas ferramentas que são o *Builder* e o *Loader*, que servem respectivamente para desenho do processo e definição do processo. Na primeira ferramenta, são fornecidas um conjunto de actividades standard que permitem criar um processo de forma mais rápida. A segunda ferramenta dialoga directamente com um módulo do motor, de forma a criar uma a versão digital do processo (definição) inteligível pelo módulo de execução

- **Servidor Oracle**

O servidor de *Oracle* é constituído por todos os módulos existentes no motor e na base de dados *Oracle* (embora só estejam representados os mais relevantes).

Os módulos correspondentes ao motor são: definição (anteriormente descrito); execução, que é responsável pela execução dos processos desenhados; e administração & monitorização que é responsável por retirar medidas, controlar e analisar o processo.

Em relação à constituição da base de dados, os módulos são os seguintes: utilizadores que guarda a informação necessária dos intervenientes nos processos, permitindo ainda importar informação ao exterior via *LDAP*<sup>9</sup>; serviços de notificação (*Notifier*) que permite o encaminhamento e tratamento das mensagens geradas ao longo do processo; finalmente o módulo da base de dados que contém todas as funcionalidades relativas à gestão de dados (gestor de transacções, gestor de acesso, *roll-back*, ...)

- **Servidor de Aplicações**

O servidor aplicações é o módulo responsável pela interligação entre o servidor e o mundo exterior. Está subdividido em dois módulos que são o *Oracle WebDB* (ou *Oracle Application Server*) e o *Notification Mailer* que corresponde respectivamente ao servidor *Web Apache* e a um *middleware* de interligação com as ferramentas de mail existentes (*MS Exchange*, *Lotus Notes*, *Unix Sendmail*, ...)

- **Utilizador**

Este módulo corresponde às funcionalidades fornecidas aos utilizadores. Funcionalidades essas que são: integração do processo na conta de mail (Aplicações de mail); *Web Monitor* que serve para administrar o processo; *Web Notification Worklist* fornece a worklist do utilizador e aplicações que permitem ao utilizador gerir os seus dados (dados pessoais e férias) e efectuar algumas pesquisas. No caso da *Oracle workflow*, estas três últimas funcionalidades são fornecidas pelo monitor, acessível no endereço do motor (endereço do motor/wfa\_html.home)

Depois de conhecer toda a estrutura da ferramenta escolhida, já podemos tirar melhor partido da mesma, porque conhecemos as funcionalidades fornecidas e quais os pontos onde podemos interagir com ela (quais as entradas e saídas). Este conhecimento vai permitir aumentar a performance das aplicações desenvolvidas.

### 3.4. Oportunidades de Evolução

Após uma análise foram detectadas algumas oportunidades de melhoria da estrutura tecnológica existente.

#### 3.4.1. Oracle Workflow 2.6

Uma das possibilidades de evolução existente neste momento, é na área da sincronização com aplicações periféricas e/ou representação de acções esporádicas do utilizador.

<sup>9</sup> Lightweight Directory Access Protocol

Entende-se como acção esporádica, algo que o utilizar pode realizar ou não realizar (Exemplo: uma factura depois de ser conferida, e enquanto não tiver sido aprovada, existe a possibilidade de o conferente rectificar algo).

O ganho em usar o *Oracle Workflow 2.6*, é o facto de suportar o tratamento de eventos que podem ser despoletados por outra aplicação ou uma acção do utilizador.

Um evento é uma ocorrência numa aplicação que é importante para o funcionamento de outros objectos pertencentes a outros/mesmo sistemas. Exemplo: existem duas aplicações, uma que monitora o disco e uma que fornece um serviço de ficheiros. Quando a aplicação de serviço de ficheiros detecta que o disco está a 90%, lança um evento de aviso à aplicação responsável pelo disco, afim de esta (que monitora o disco) proceder à realização das acções necessárias.

Sendo assim, a utilidade dos eventos na Sonae Distribuição seria a centralização de toda a interacção com o exterior, captando assim todas as mudanças no mundo exterior (em particular com o *SAP*).

Para isso, é necessário ter instalado o *Advanced Queuing* (já vem na versão 8i). A arquitectura que é necessário criar é a seguinte:

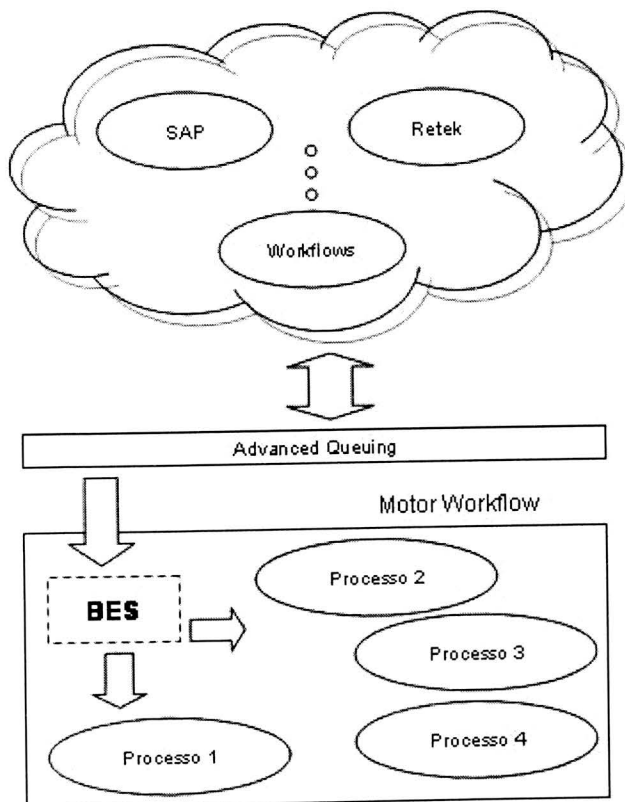


Figura 7 : Eventos (*Business Event System*)

É importante referir, que podemos restringir um evento ao nível da aplicação pois os eventos são “hubs” de mensagens que implementam uma arquitectura *Publish/Subscribe*, ou seja, quando qualquer evento corre é difundido pelos processos que subscreveram esse evento.

Outras vantagens do uso de eventos são: a facilidade com que se consegue integrar sistemas, possibilidade de integrar *EDI*, *XML* e a integração com os vários *Workflows* e/ou com o utilizador. Em anexo existe um exemplo sobre como poderá melhorar o processo de *workflow* a interagir com o exterior (ver ANEXO H). No anexo seguinte (ANEXO I), encontra-se um exemplo da melhoria obtida a nível dos processos.

#### 3.4.2. Real Application Clusters (RAC)

Dado que na Sonae Distribuição existe como suporte de base de dados o *Oracle*, torna-se necessário garantir que os tempos de resposta e a disponibilidade da informação sejam as melhores.

Sendo assim, existe a possibilidade de criação de *clusters* de hardware controlados por esta nova ferramenta da *Oracle*. As vantagens do *cluster* serão a maior rapidez de resposta às pesquisas e maior disponibilidade, acompanhadas pelas funcionalidades de controlo de *Backups*, controlador de *caches*, balanceamento, recolha de *KPI's* e controlo da arquitectura fornecidas pelo software *Oracle*.

Em Portugal, o único exemplo de uma implementação bem sucedida é na FEUP (Faculdade de Engenharia da Universidade do Porto). Em anexo estão descritos os cenários possíveis de implementação (ver ANEXO J).

#### 3.4.3. Oracle Internet Directory (OID)

Outra necessidade existente na Sonae Distribuição, é a sincronização do *Oracle workflow* com o repositório de informação dos colaboradores. Sendo assim, a vantagem retirada deste componente de base de dados, é o facto de permitir a criação e gestão de uma imagem da informação dos colaboradores. Com essa replicação tornar-se-ia a resolução de perfis mais robusta (deixa de depender do outro sistema) e mais rápida dado que se eliminariam os acesso *LDAP* ao exterior.

Em anexo, podemos encontrar qual seria a arquitectura física a implementar e informação acerca dos componentes lógicos existentes (ver ANEXO K).

## 4. Metodologia

A metodologia consiste na criação de um modelo de desenvolvimento de projectos de *workflow*, que permita melhorar a situação actual. Embora já existam linhas orientadoras para os projectos de *software*, os projectos de *workflow* merecem especial atenção pelo facto de se dividirem sempre em duas partes distintas (processo e aplicação)

### 4.1. Objectivos

Os objectivos necessários para melhorar o desenvolvimento nestes projectos são:

- **Garantir independência da tecnologia de apresentação**

Garantir que qualquer desenvolvimento é modular suficiente ao ponto de qualquer alteração na interface com o utilizador não obrigar a fazer mudanças a nível do processo/funcionalidades criadas.

- **Aumentar a reutilização das soluções**

Garantir que tudo é criado de uma forma genérica afim de ser reaproveitado no futuro. Sendo assim, aumentaremos as funcionalidades existentes e diminuímos o tempo de desenvolvimento. Para isso, será necessário ter uma visão futurista das funcionalidades a criar.

- **Criar linhas orientadoras**

É necessário prever formas de actuar quando acontece algo, permitindo assim que o tempo de resposta a uma situação seja menor, o tempo de entrosamento de um novo elemento seja menor e finalmente que todos estejam em sintonia com as soluções a criar, evitando assim que se criem várias soluções para um problema já resolvido. Em resumo, ter-se-á que criar um conjunto abrangente de situações, causas associadas e forma de resolver/responder à situação.

- **Zelar pela documentação dos projectos**

É necessário garantir que todos os desenvolvimentos se encontram bem documentados, afim de a manutenção, compreensão e a utilização serem mais fáceis.

Em conclusão, o atingir destes objectivos garante o sucesso de todo o desenvolvimento.

### 4.2. Fases de um projecto

Antes de mais, há que garantir que a criação de um projecto de *workflow* tem um processo bem definido, de forma a uniformizar o modo de análise e não se perder nenhum passo.

Um processo de *workflow* divide-se em três grandes blocos: a especificação, o desenvolvimento e o lançamento (ver figura seguinte).

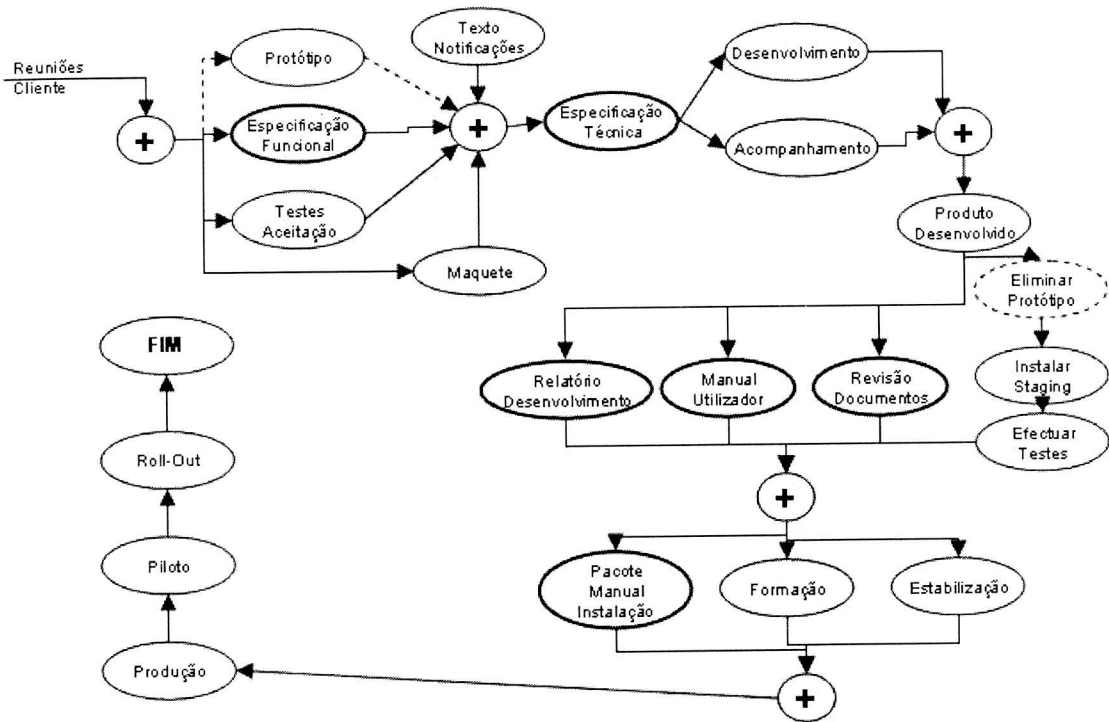


Figura 8 : Fases de um projecto

Sendo assim, na especificação teremos toda a relação com o cliente com o objectivo de especificar todos os requisitos que são necessários implementar. O método de análise processa-se de acordo com o que foi descrito no ponto 2.3, excepto no facto de existir a possibilidade de criação de um protótipo que serve para verificação da solução final. Temos ainda a especificação do texto a aparecer nas notificações, ou seja, a informação a aparecer na *worklist*.

Na fase de desenvolvimento, temos de garantir os seguintes factores:

- **Validação Técnica**

Na validação técnica, pretende-se assegurar os seguintes pontos:

- Bom uso da tecnologia, ou seja, não estamos a violar nenhuma regra associada à ferramenta. Por exemplo, criar um processo com um utilizador que pode responder às notificações em nome de outras pessoas, é uma violação à tecnologia de *workflow* porque retira o controlo ao processo.
- Todos os requisitos estão englobados na implementação e além disso que não são incompatíveis.
- Soluções encontradas são modulares e por isso reaproveitáveis no futuro, de forma garantir retorno de investimento elevado.
- Integração com o utilizador (maquete) e outras aplicações (garantir a segurança de dados).

- Não violação da integridade da nossa infraestrutura, de forma a não gastar recursos de forma massiva enquanto existem outras necessidades (Exemplo: sobrecarregar a rede aquando das difusões).

Depois de efectuadas todas estas validações, podemos prosseguir os desenvolvimentos. Sendo ainda de frisar que poderão aparecer novas situações durante a codificação da solução que deverão ser validadas, de forma a não violar nenhuma situação já fechada.

- **Usabilidade**

É ainda necessário garantir que o produto final é usável, para isso, temos de nos colocar no papel do utilizador final, de forma a garantir os seguintes factores: aprendizagem rápida, acessibilidade das funcionalidades (evitar excesso de ecrãs até chegar ao que queremos fazer) e redução da possibilidade de introdução de erros. Neste ponto, há que ter em consideração a maquete validada pelo cliente e a experiência adquirida em situações de anteriores projectos.

Sendo assim, a maquete validada pelo cliente serve de base a um estudo mais alargado a efectuar projecto a projecto, tendo em conta a gama de utilizadores e as funcionalidades mas usadas. Além de tudo isto, sem nunca esquecer que o centro terá de ser a *worklist* do utilizador, pois será a funcionalidade mais utilizada e mais importante (porque é através da *worklist* que o utilizador acrescenta valor ao processo).

- **Custos, prazos e expectativas**

Como em qualquer projecto, temos de controlar os custos finais que estamos a incorrer ao optar por certas soluções.

Garantir que a solução estará disponível na data combinada inicialmente, de forma a não criar desconfianças sobre a solução final.

E finalmente, fazer com que o cliente acompanhe também a solução de forma a não criar falsas expectativas acerca da solução final.

- **Documentação**

Como foi referido anteriormente, há que garantir que toda a implementação está descrita. Sendo assim, os documentos a produzir são: especificação técnica (descrição das decisões tomadas para codificar os requisitos), relatório de desenvolvimento (dados relativos à codificação da solução) e manual o utilizador (guiar na utilização da aplicação). Em anexo, encontra-se a estrutura destes relatórios (ver ANEXO L).

- **Qualidade do produto**

Finalmente, temos de assegurar a qualidade do que foi desenvolvido, por isso, temos de efectuar um plano de testes de forma a garantir que a aplicação vai responder de acordo com o especificado.

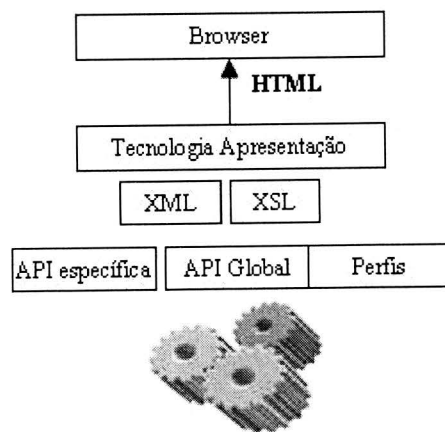
Terminada esta fase, segue-se o lançamento. O lançamento passa por uma fase de estabilização da solução (criar confiança no funcionamento da solução – complemento dos testes anteriormente efectuados), formação dos utilizadores iniciais (piloto) e

preparação da instalação final. Passadas estas três acções, a aplicação é colocada no seu ambiente final, de forma a ser efectuado um piloto com o objectivo de verificar quais os ganhos, ambientar os utilizadores e o mais importante retirar as reacções das pessoas (de forma a criarmos a estratégia para o alargamento da utilização da solução – *Roll-Out*).

Resumindo, as maiores diferenças da metodologia para implementação destas aplicações são: a fase de análise ter duas partes distintas, definição do texto das notificações, a orientação da interface com o utilizador e as especificidades da tecnologia usada.

### 4.3. Arquitectura lógica

A arquitectura que vai permitir a independência do desenvolvimento em relação à tecnologia de apresentação, é a seguinte:



**Figura 9 : Arquitectura para os projectos de *workflow***

A filosofia associada à escolha desta arquitectura, assenta em dois pontos fulcrais que são: a modularização das soluções e a utilização do *XML* (*Extensible Markup Language*) como forma de comunicação com o mundo exterior.

Com a modularização, pretende-se facilitar a compreensão do que está desenvolvido e como está desenvolvido. Além disso, facilita a reutilização do que já foi implementado. De forma a atingirmos os objectivos foram criados os seguintes blocos:

- *API* específica, que serve como repositório das funcionalidades específicas do processo em questão, ou seja, contém as funcionalidades que não poderão ser reutilizadas noutras situações
- *API* Global, que contém todas as funcionalidades genéricas implementadas até ao momento
- Perfis, faz parte da *API* Global e serve para concentrar todas as funções associadas ao tratamento de perfis

Posto isto, tudo que for desenvolvido está correctamente classificado, mas convém frisar que de nada vale esta estrutura se as *API's* não estiverem documentadas.

Quanto à parte de apresentação, será independente de qualquer tecnologia de apresentação utilizada (*PSP*, *JSP*, *ASP*, ...), graças ao facto das *API* anteriormente

referidas gerarem *XML*. Sendo assim, basta apenas criar o *XSL (Extensible StyleSheet Language)*, associado à tecnologia usada, de forma a gerar o *HTML* para poder ser interpretado pelo browser. O papel do *XML* é importante porque, é dessa forma que os dados são enviados entre as camadas de apresentação e a *API*.

#### **4.4. Manual de qualidade**

De forma a se poderem atingir todos os objectivos, foi criado um manual de qualidade associado a estes projectos. Esse manual descreve de forma exaustiva os passos a cumprir. Sendo assim, além dos dois pontos anteriormente descritos, o manual de qualidade refere os seguintes pontos:

- Normas para documentação (ver ANEXO L)
- Sintaxe a usar nos processos, de forma a facilitar a compreensão. E formas de tornar os desenhos dos processos flexíveis (ver ANEXO M)
- Normas para os textos das notificações, de forma aos elementos aparecerem sempre no mesmo local
- Cuidados com o modelo de dados
- Formas de controlo/garantia da não violação das regras estipuladas, ou seja, fazer auditorias aos projectos
- Interface com o utilizador, de forma a garantir globalização entre todos os processo digitalizados (ver ANEXO N)
- Mecanismos já criados, ou seja, uma espécie de repositório de formas de decidir perante certas situações

Sendo assim, este documento servirá como garantia da qualidade de desenvolvimento dos projectos

Em conclusão, a criação desta metodologia vai permitir melhorar os desenvolvimentos e aumentar as funcionalidades fornecidas aos utilizadores dos processos. As maiores melhorias esperadas são: a nível do reaproveitamento de soluções, rapidez e qualidade de desenvolvimento.

## 5. Processo “Comunicação de artigos”

Este processo, intitulado “Comunicação de artigos”, serviu de exemplo para validar a metodologia apresentada. Além disso, este processo, é pioneiro no historial do *workflow* na Sonae Distribuição, porque é o primeiro a modelar um processo não administrativo, ou seja, este processo é relativo ao negócio do retalho propriamente dito. Tem também a particularidade de se associar de forma mais “íntima” com a Intranet da Modelo Continente Hipermercados (*Insite*).

### 5.1. Situação actual

Actualmente o processo é feito de forma pouco estruturada, pois cada director comercial utiliza a sua própria estratégia. Há uns anos foi criada a *TALCO* (Tabela de Comunicação de Alterações de Gama), que era um conjunto de folhas em Excel que requeriam a introdução de um enorme conjunto de informação sobre os artigos. Por esta razão, nem todos os gestores comerciais utilizam este método, criando algumas formas mais simplificadas (criando uma certa entropia no processo, principalmente na comunicação às lojas). Esta situação leva que de certa forma todos os outros intervenientes (Gestores de Inventário, Direcção de Espaço e Loja) tenham também dificuldades de trabalho. Para resolver este problema resolveram criar uma outra folha de Excel que compilava tudo que estava em linha (*CARDEX*).

### 5.2. Objectivos

Além do objectivo referido anteriormente (validação da metodologia), pretende-se com a digitalização deste processo atingir os seguintes objectivos: a unificação do canal de comunicação, zelar pelo cumprimento de todas as fases (envolver todas as pessoas), aumentar a eficácia e facilitar o controlo.

### 5.3. Intervenientes

Os intervenientes neste processo são: as Direcções Comerciais que são responsáveis pela criação da comunicação; a Direcção de Espaço que é responsável por validar que o artigo terá lugar na prateleira da gôndola; a Gestão de Inventário que será responsável por garantir que o artigo estará disponível para encomenda da loja; e finalmente, as Direcções de Operação das lojas que serão responsáveis pela implementação da comunicação em causa (efectuam o trabalho no terreno).

### 5.4. Funcionalidades

As funcionalidades necessárias para complementar o processo são:

- Criação de listagens, que permitam visualizar as comunicações recentes (activas), removidas e históricas
- Exportação para *Excel*, para envio a lojas que eventualmente não tenham acesso à intranet da Sonae
- Gestão de categorias planogramadas, que corresponde a definir quais as categorias que gozam desta propriedade.

Posto isto, falta agora definir os processos que irão suportar a gestão das comunicações.

## 5.5. Processos

Os processos que são necessários digitalizar, são os correspondentes às diferentes comunicações possíveis, sendo elas: entrada em linha, saída de linha e suspensão de venda.

### 5.5.1. Entrada em linha

Uma entrada em linha corresponde a uma decisão de colocar um produto disponível para venda. O desenrolar do processo associado a uma comunicação deste tipo, depende do artigo em questão. As situações possíveis e as diferenças de fluxo são:

- **Artigo centralizado<sup>10</sup> e não planogramado<sup>11</sup>**

É necessário que o gestor de inventário coloque a data em que o produto estará disponível para encomenda. Essa data irá servir para determinar quando é que a loja terá acesso a essa informação (será na data respectiva).

- **Artigo planogramado e não centralizado**

É necessário que a direcção de espaço valide a entrada, ou seja, garanta que não haverá problemas de espaço na loja. Caso a entrada não seja validada, a direcção comercial terá a possibilidade de analisar o motivo apresentado e decidir se reenvia para nova validação (depois de alterar a comunicação) ou envia para histórico. Depois de validada, a direcção de espaço coloca a data de planograma.

Quando chegar a essa data a comunicação estará disponível para as lojas, sendo que o planograma terá de ser consultado num conteúdo existente na *intranet*.

- **Artigo centralizado e planogramado**

Envolve um conjunto maior de pessoas. Dado que este processo é a soma das duas situações anteriores, o artigo passa primeiro por uma validação pela direcção de espaço. De seguida, o gestor de inventário define a data de *stock* (disponível para encomenda), depois dessa data definida, a direcção de espaço atribui uma data de planograma. Dessas duas datas anteriormente referidas, a data de planograma tem de ser sempre maior ou igual à data de *stock*.

A comunicação ficará visível às lojas quando chegar à data de planograma.

- **Artigo não centralizado e não planogramado**

Não existirão intermediários entre a direcção comercial e as lojas. Sendo assim, a comunicação ficará imediatamente disponível para as lojas.

Em todos os casos, convém garantir que enquanto a comunicação não estiver disponível para as lojas, as datas poderão ser alteradas. No caso, de existirem ambas as datas definidas (**Artigo centralizado e planogramado**), o outro interveniente deve ter conhecimento das alterações feitas. Sendo assim, quando a data de planograma é alterada, o gestor de inventário terá de ter conhecimento dessa alteração. O mesmo se verifica

<sup>10</sup> Significa que o artigo é fornecido directamente pelo operador logístico da Sonae (Modis)

<sup>11</sup> Artigo em que é necessário definir uma localização na gôndola. Um planograma tem uma data associada, que corresponde à sua data de publicação

quando é alterada a data de *stock*, sendo neste caso dado conhecimento ao representante da direcção de espaço.

Em ANEXO M, poder-se-á observar o processo desenhado na ferramenta da *Oracle (Builder)*. Os sub-processos que aí se encontram vão ser descritos no ponto 5.9.2.

### 5.5.2. Saída de linha

Uma saída de linha, corresponde a uma decisão de retirar o produto da placa de vendas. O processo é idêntico ao anterior, excepto no facto de as lojas poderem comunicar o *stock* actual. Essa informação serve para que as direcções comerciais possam actuar em caso de necessidade de escoamento de *stock*, ou seja, reconsiderar a saída de venda até que o *stock* seja esgotado.

### 5.5.3. Suspensão

Uma suspensão corresponde a retirar imediatamente o produto de um posto de venda. Esta acção reflecte problemas associados ao artigo. Problemas esses que podem ser: falta de qualidade do artigo, problemas de negociação com o fornecedor (litígios) e outras desconfianças que existam.

Por isso, dada a gravidade associada, o processo corresponde apenas a comunicar de forma imediata todos os intervenientes em questão (de acordo com a figura seguinte).

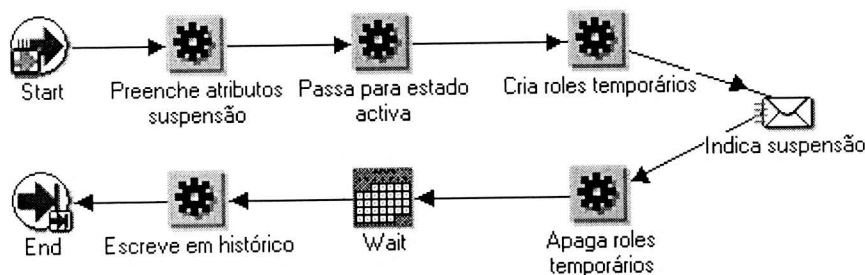


Figura 10 : Suspensão de um artigo

Notação: interação com o utilizador com resposta, acção executada pelo sistema, nó de início, nó terminal, espera um determinado tempo

### 5.6. Arquitectura Lógica

A arquitectura lógica associada a este projecto, está extremamente ligada à arquitectura da Intranet da Sonae. Por isso, toda a parte da interface com o utilizador é semelhante, ou seja, o uso do *IIS* como serviço de Internet, *ASP* como tecnologia de apresentação e a ligação à base de dados via *ODBC*.

O *Oracle Workflow* precisa ainda do *Apache* como serviço de Internet, para disponibilizar a ferramenta de monitorização e controlo.

Posto isto, a arquitectura a implementar para este projecto em específico, consiste em: criar o *DBLink* para a base de dados de negócio para efectuarmos pesquisas aos artigos; usar *LDAP* para acesso ao directório de informação de utilizadores e finalmente o acesso ao *Exchange Server* para envio de mails aos utilizadores.

Na figura seguinte, está representada toda a arquitectura lógica necessária.

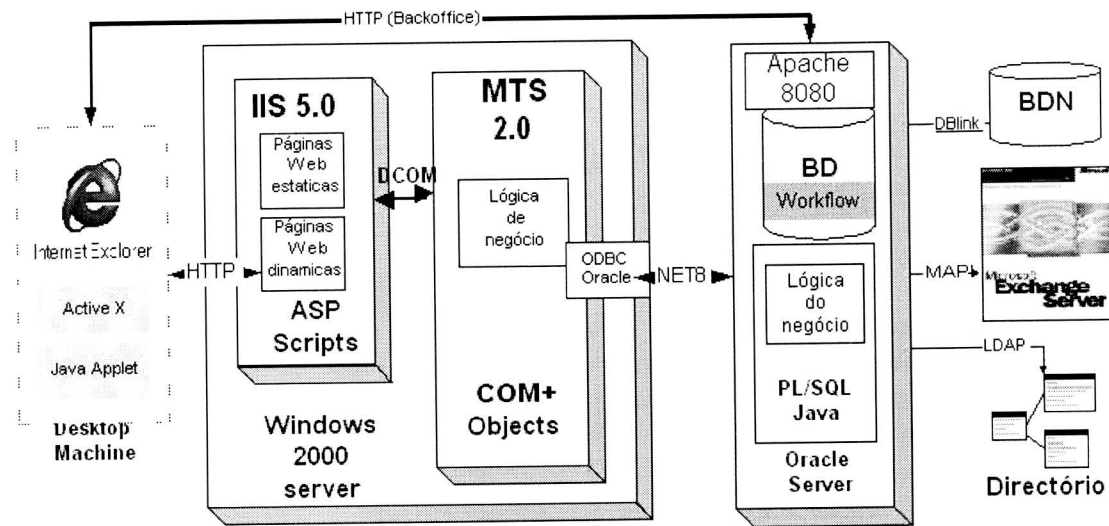


Figura 11 : Arquitectura Lógica do projecto "Comunicação de Artigos"

### 5.7. Interface com o utilizador

A interface com o utilizador consistiu em usar o que estava estipulado na metodologia, relativamente à parte de gestão da worklist, ou seja, a lista de trabalho e o tratamento das notificações.

Em relação à parte aplicacional, foram desenhadas as interfaces que se encontram descritas em anexo (ver ANEXO O). Tendo em conta os princípios enunciados na metodologia e de forma a não criar impacto em tudo que já estava definido.

### 5.8. Benefícios esperados

Os benefícios retirados desta digitalização foram os seguintes:

- **Comunicação online**

Com esta aplicação a comunicação de entradas, saídas e suspensões far-se-ão no próprio dia, sem se esperar por datas programadas. Antes da ferramenta, existiam datas programadas para comunicar a cada loja o que tinha que fazer, o que levava a um acumular do trabalho na loja, fazendo assim com que as modificações não fossem todas feitas na data prevista.

- **Eliminação de circuitos paralelos**

Outro ganho, é o facto de deixar de haver a confusão actual. Essa confusão era motivada por as lojas receberem acções por telefone, mails e até por comunicação pessoal. Com a ferramenta a fonte de comunicação fica a ser só uma e partilhada por todas as direcções.

- **Incorporação com a base de dados dos artigos**

Outro ganho, é o facto de não ser necessário introduzir a totalidade dos dados, pois os mesmos são copiados automaticamente pela aplicação (a partir da base de dados dos artigos).

- **Associar imagens**

Nesta aplicação podemos associar imagens ao artigo em questão, melhorando assim a qualidade da comunicação.

- **Sincronização e controlo**

Outro benefício é o facto de cada processo passar por todas as pessoas necessárias (sincroniza todas as pessoas relativas ao processo), o controlo do estado do processo (onde está, com quem está) e o armazenamento do histórico por forma a possibilitar posteriores análises.

- **Melhor comunicação às lojas**

Esta melhoria corresponde à unificação do canal de comunicação (evitar a chegada de informação por várias fontes) e à personalização da informação a aparecer aos intervenientes, ou seja, com esta aplicação as lojas vêem apenas o essencial e não perdem tempo à procura da informação que necessitam.

- **Personalização da informação visualizada**

Este factor foi largamente melhorado, porque, ao contrário do que existia antes, agora cada pessoa apenas vê a informação estritamente necessária para o desempenhar da sua função. Sendo assim, não perde tempo a separar o que precisa do resto da informação.

Embora seja difícil a neste ponto contabilizar de forma quantitativa qual o peso total dos ganhos (devido a ainda não se ter efectuado o *Roll-Out*), existe uma substancial melhoria prevista. Melhoria essa que irá fazer com que a curto prazo, exista a necessidade de proceder a um alargamento desta solução, de forma a suportar todo o processo de Entrada/Saída de artigos (será apresentado no capítulo seguinte).

## **5.9. Implementação**

Neste ponto, vão ser descritas as fases que foram previstas para implementação deste projecto, mostrando as diferenças em relação aos outros projectos (nova metodologia), trabalho realizado no âmbito do estágio e as principais dificuldades encontradas, assim como a resolução das mesmas.

### **5.9.1. Plano**

A implementação do projecto no presente estágio, seguiu a metodologia e consistiu no seguinte:

1. Análise do documento de especificação e efectuar a análise “*As Is*”, complementado através de reuniões com o cliente
2. Criação da maquete relativa à aplicação (ver ANEXO O)

3. Criação dos desenhos dos processos “*To-Be*” e criação de um protótipo com base nesse desenho. Neste caso específico não houve lugar a alterações ao processo inicial, sendo apenas acrescentadas as actividades necessárias à implementação
4. Definição dos textos das notificações segundo a metodologia criada, de forma a facilitar o trabalho do utilizador
5. Criação do plano de testes, para verificação da qualidade da aplicação criada
6. Validações técnicas: total cobertura dos requisitos, qual a origem dos dados, estratégias de resolução de perfis, especificação das horas a que devem executar os processos mais pesados (aviso *worklist*, tratamento dos *timeout* e sincronização *LDAP*)
7. Acompanhamento do desenvolvimento e teste da aplicação
8. Análise, auditoria e aprovação da documentação associada ao projecto
9. Preparação da entrega ao cliente e acompanhamento da sua utilização

Fora do âmbito do estágio ficou o *Roll-out*, devido à duração do estágio não ser suficiente. Sendo assim, no fim do presente estágio está a decorrer o piloto da aplicação, de forma a efectuar uma análise real dos benefícios conseguidos.

### 5.9.2. Principais dificuldades

As principais dificuldades encontradas na digitalização deste processo foram:

- **Nova plataforma de trabalho**

Foi o facto de as pessoas estarem habituadas a utilizar o mail e o *excel* para fazer este tipo de comunicações, que levou a ser associado ao processo uma funcionalidade que “convida” os utilizadores a ir ao encontro da nova ferramenta. Essa funcionalidade corresponde à pessoa receber periodicamente um mail, com um resumo das acções à espera de serem tratadas. Incluiu também um *link* para a sua *worklist*.

- **Gama de utilizadores**

A existência de utilizadores associados ao processo, com pouco contacto com as tecnologias da informação, levou a que a tivesse havido uma especial atenção na usabilidade desta aplicação, de forma a abranger o maior número de utilizadores possíveis (garantindo os princípios enunciados na metodologia acerca da usabilidade).

- **Introdução do novo modelo de desenvolvimento**

Na implementação surgiram algumas dificuldades, pois a forma de trabalho que este modelo impõe tem algumas diferenças, ao que era praticado, nomeadamente ao nível da documentação.

- **Perfilagem**

Esta situação fica a dever-se ao facto de ser necessário notificar um subconjunto de pessoas do grande universo Sonae. Sendo assim, punham-se problemas de pesquisa na estrutura de utilizadores. Esta situação foi facilmente ultrapassada com a criação de roles temporárias e notificação à função em vez de às pessoas (Ex: Notificar o director de loja e não o Senhor X).

• **Alteração de artigos**

Esta situação ocorre devido a existir a necessidade de alterar um artigo ao longo do seu percurso (processo de entrada, saída ou suspensão). Como a alteração poderia ser feita em qualquer parte do processo, a solução encontrada foi criar um processo à parte para tratar essas situações (figura seguinte).

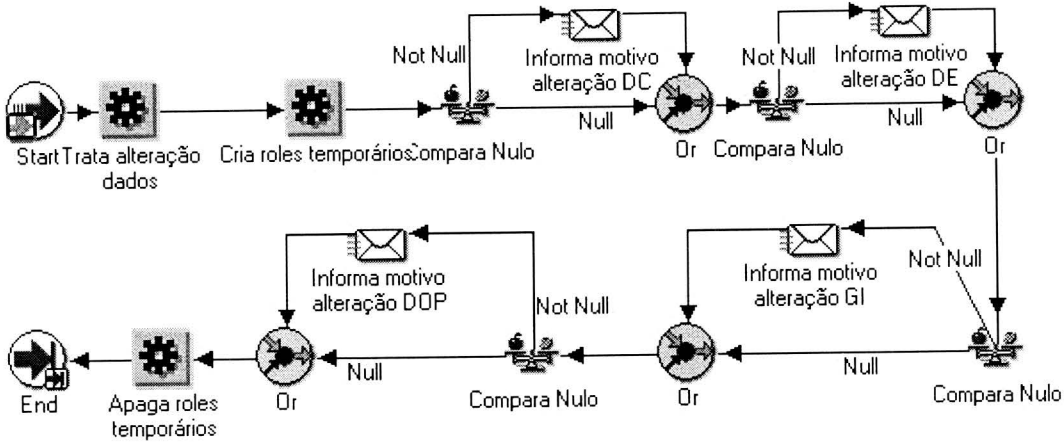


Figura 12 : Alteração de uma comunicação

Sempre que existe uma alteração, o processo inicia começando por detectar quais foram as alterações. De seguida, verifica os destinatários e de acordo com as modificações efectuadas notifica os intervenientes da comunicação.

• **Alteração das datas (planograma e/ou stock)**

Finalmente existiu mais uma situação. Situação que se encontra relacionada com o facto de haver necessidade de modelar a possibilidade de alterar as datas enquanto as comunicações não estiverem publicadas. Esta alteração é um caso particular da situação anterior, porque apenas pode ocorrer num ponto bem definido do processo (entre a colocação das datas e a publicação às lojas – conforme representado nas entradas em linha).

Daqui advêm dois problemas que são: a espera em ciclo para determinar se já chegou a data de publicação ou se o utilizador mudar a data (“*busy waiting*” – espera activa) e o facto do *workflow* servir por definição para modelar acções obrigatórias. A solução encontrada foi a seguinte:

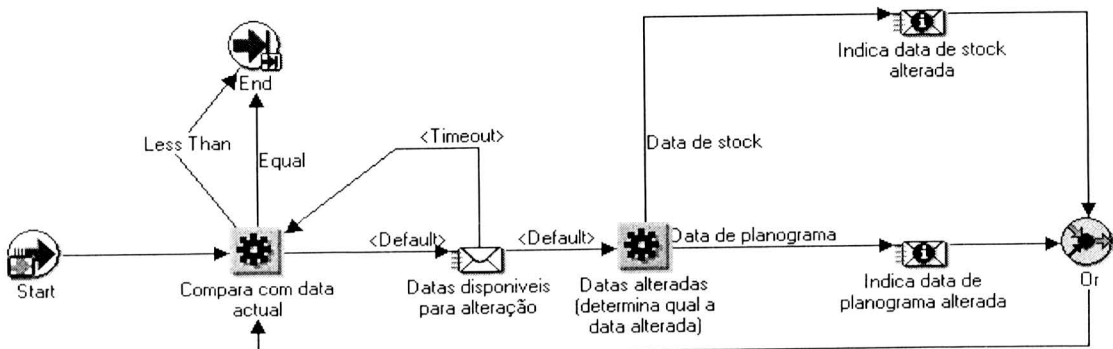


Figura 13 : Alteração de datas

Para não ocorrer a espera activa, foi criado um *time-out* de forma à comparação só ocorrer de 24 em 24 horas. Para modelar a possibilidade de alterar as datas, foram criadas notificações de baixa prioridade (para não serem notadas), pois este é um erro a não cometer, mas a tecnologia em questão não permite outra solução.

Estes problemas foram ultrapassados tendo em conta a modularidade das soluções, pois segundo o meta modelo criado, é mais importante resolver o problema de uma vez e de forma a conseguir uma elevada reutilização dessa solução (*ROI* elevado).

### **5.10. Melhorias a introduzir**

Antes de pensar em alargamentos da solução, surge a necessidade de alterar o processo “Comunicação de artigos”, pois não suporta a totalidade das acções necessárias efectuar numa entrada, saída e suspensão.

Sendo assim os melhoramentos seriam:

- **Novo fornecedor**

Quando o fornecedor ainda não interagiu com a Sonae Distribuição, e caso o artigo em questão seja centralizado (distribuído pela operadora logística da Sonae – Modis), é necessário garantir uma janela de entrega para a mercadoria. Por isso, é necessário notificar os entrepostos para definirem uma janela de entrega para esse fornecedor.

Para finalizar o processo de cadastramento, é necessário criar esse fornecedor no *SAP*, de forma a poder-se efectuar toda a contabilidade necessária com ele. Para isso, é necessário notificar o *Helpdesk* associado a estas acções.

- **Encomenda automática quando se aprova uma entrada**

Quando se decide um entrada em linha, a(s) loja(s) associada(s) a essa entrada, terão de ter o artigo em questão na data da entrada. Para isso, é necessário garantir que é feita a definição de quantidades a encomendar e conseqüentemente efectuar a encomenda ao entreposto/fornecedor.

Em relação à definição de quantidades, ter-se-á que notificar o gestor de inventário, direcção de espaço (qual o tamanho da área de venda atribuída) e a loja. No fim deste processo, estarão definidas as quantidades tendo em conta o espaço disponível, a percepção do responsável pela loja e a visão estratégica do gestor de inventário.

- **Interligação com conteúdo “Planogramas”**

Outra melhoria, é a incorporação na aplicação “Comunicação de artigos” da informação relativa à localização do artigo na gôndola. Sendo assim, centralizamos a interacção de toda a informação relativa às entradas, saídas e suspensões (eliminando a situação actual, caracterizada por duas ferramentas distintas).

- **Crítica de quantidades**

Para finalizar, seria extremamente útil acoplar este pequeno processo para permitir que as lojas e os gestores de inventário discutam qual será a melhor política de gestão de stock a utilizar e principalmente qual o stock inicial. Esta melhoria poderá ajudar a resolver os problemas de excesso de stock nas lojas, quando se efectua uma entrada.

Com estas alterações, ficam cobertas todas as acções associadas à interacção necessária quando se efectua a comunicação de uma decisão à placa de vendas.

Em conclusão, com a implementação deste processo conseguimos validar e melhorar a metodologia, a ponto de, no momento, estar a ser alargado a todos os projectos de *workflow* existentes na Modelo e Continente.

Em relação ao projecto em si, o sucesso também se deveu à grande abertura que os utilizadores finais tiveram em relação a esta mudança. Sendo assim, ficamos todos satisfeitos porque conseguimos criar um conjunto de soluções para o futuro e ao mesmo tempo satisfazer os utilizadores.

## 6. Processos: Entrada e Saída de Artigos

O processo intitulado Entrada/Saída de artigos, é um processo mais alargado que o “Comunicação de artigos”, pois pretende além da comunicação, preocupar-se com o que acontece antes, depois e durante.

Antes corresponde à tomada de decisão sobre a comunicação a efectuar. Por isso, deve ter em conta as negociações com fornecedores, as várias sugestões e o mercado.

Depois corresponde à análise do resultado da decisão tomada, por isso é importante fornecer todas as métricas necessárias para que seja possível retirar quais os benefícios. Além disso, pode ser necessário algum tratamento excepcional (acções jurídicas, planos de escoamento, ...)

A preocupação durante, refere-se à manutenção da competitividade da decisão tomada (entrada/saída do artigo), gestão dos espaços disponíveis para negociação (*Checkouts* e *topos*) e ao acompanhamento ao cliente.

Como todo o negócio em questão, depende da velocidade e qualidade da resposta dada, surge a oportunidade de uso desta tecnologia. Esta tecnologia, agiliza o processo devido a facilitar a transmissão e/ou automatizar certas acções, e garante a qualidade porque possibilita a interacção de todos os elementos necessários.

### 6.1. Processo

Depois de verificar todo o processo aquando de um entrada ou saída em linha, surgiram os seguintes processos não contemplados em comunicação de artigos:

- **Alterações aos eventos**

Um dos factores associados a uma entrada de um artigo, é o facto de pertencer ou não a um evento, ou seja, se o artigo está abrangido por alguma acção promocional. Sendo assim, torna-se necessário interagir com o conteúdo associado aos eventos promocionais (PEP), e actualizar esses dados no repositório de artigos de forma a difundir essas alterações para as lojas.

- **Gestão de Preços (Aprovação e crítica de preços)**

Um processo a melhorar é a forma de reacção aos preços da concorrência, dado que hoje em dia a comunicação é efectuada por mail e muitas vezes até cai no esquecimento, ou seja, quando o Critico de Preços altera um preço, envia um mail ao Director Comercial de forma a informá-lo da sua reacção à concorrência. Por isso, o uso desta ferramenta a iria permitir maior rapidez de reacção, mesmo envolvendo mais pessoas.

- **Escoamento de excessos**

Este processo corresponde à resolução do que fazer ao stock existente numa loja. Na realidade existem três hipóteses que são: venda a preço inferior, nova entrada na placa de vendas e transferência de stock para outra loja.

Este processo, hoje em dia é demasiado moroso devido a necessitar de muita interacção entre as lojas e os gestores de inventário. Além disso, seria uma enorme ajuda para as lojas resolverem problemas de stock.

- **Gestão de Checkouts e Topos**

Este pequeno processo tem como objectivo facilitar a interacção entre as Direcções Comerciais, Direcção de Espaço e Fornecedores, na venda destes locais especiais de vendas. Com este processo, conseguimos eliminar a perda de informação acerca do estado de cada ponto de venda, o que implica melhor aproveitamento das infra-estruturas de venda.

- **Monitorização de artigos (Stock e Vendas)**

Um processo importante seria a possibilidade de usar o *workflow*, para monitorização automática dos artigos que entraram na placa de vendas, ou seja, criar alertas em função de regras de gestão a desenvolver para cada artigo. Estes alertas permitiriam aos gestores comerciais melhorarem as vendas, aos gestores de inventário controlarem melhor os stocks e aos técnicos de *merchandizing* perceber possíveis problemas de localização/apresentação dos produtos.

- **Negociações com fornecedores**

Este processo consiste na avaliação das propostas enviadas pelos fornecedores para compra dos seus produtos. O maior ganho neste processo seria a possibilidade de se efectuarem rápidas triagens à partida e permitir criação de formas de negociação automática (espécie leilões).

- **Transferências de produtos**

Este processo refere-se à passagem de produtos entre lojas (uma das formas de escoar stock). O ganho aqui seria a rápida interacção entre todos os intervenientes de forma a se escoar o produto o mais rapidamente possível e aproveitando de espaço nos transportes entre as lojas.

- **Tratamento das suspensões**

Finalmente, outro processo crítico é a suspensão de um artigo, que dada a sua gravidade devem ser tratadas de forma rápida. Este processo pode levar a interacção com o fornecedor para devolução, ou com o serviço de reparações, ou com processos judiciais ou apenas uma suspensão temporária.

Em conclusão, todos estes processos em conjunto fariam o processo de entrada e saída de artigos simples e fiável. No ANEXO P, podemos observar os vários processos em interacção, cada nuvem corresponde a um processo específico.

## 6.2. **Novos intervenientes**

Com o alargamento será necessário prever o acesso dos seguintes utilizadores:

- **Fornecedores** – colocação das propostas
- **Clientes** – sugestões e reclamações
- **Shoppinguista** – introdução dos preços da concorrência

- **Critico de preços (Lojas)** – reacção aos preços da concorrência

Com a integração destes intervenientes ficaria completa toda a gestão da entrada e saída de artigos. Mas, para que isto aconteça é necessário permitir acessos à rede interna da Sonae Distribuição, de forma a que os intervenientes possam interagir com a aplicação.

### 6.3. **Integração com outros sistemas**

Para que se consiga garantir o bom funcionamento dos processos aqui descritos, é necessário garantir a interacção com os seguintes sistemas:

- **Planogramas** - para poder retirar qual o local onde o artigo ficará na loja
- **Catálogo técnico** – para completar a informação acerca dos artigos
- **SAP** – retirar informação sobre os fornecedores e/ou criação automática dos fornecedores
- **Retek** – para retirar e actualizar os dados referentes a um artigo
- **Gestão de preço de venda ao público (SPVP)** – para difundir as alterações a preços efectuadas (podendo até prever um interface de difusão directamente para o FrontOffice)
- **Shopping** – para retirar a informação dos preços da concorrência
- **Planeamento de Eventos Promocionais (PEP)** – para retirar e actualizar informação das promoções existentes

Com a interacção com estes sistemas garantida, o utilizador tem no *workflow* uma interface única, onde centraliza todas as suas acções. Ficando assim com um conjunto alargado de funcionalidades num só ponto de interacção, e garantindo que não precisa de perder tempo na replicação manual de informação.

Em conclusão, a comunicação de artigos à loja é uma pequena gota, no enorme e importante processo de gestão dos artigos a colocar na placa de vendas.

## 7. Conclusões

Com o término do estágio, há que fazer a avaliação do que aconteceu e do que fica para fazer, pois numa empresa desta dimensão temos de estar sempre atentos para garantir que estamos a fazer o melhor possível (não existem soluções vitalícias).

### 7.1. Estágio

O trabalho realizado foi muito rico em experiências. Isto deve-se essencialmente ao facto do ambiente de trabalho ser diferente do que é vivido na faculdade e pelo tema do estágio ser uma novidade (em relação ao plano de estudos). As novas aptidões recolhidas foram:

- Aprendizagem dos conceitos que regem o *Workflow*
- Aprendizagem e uso da solução de *Workflow* da *Oracle*
- Gestão de projectos em *Outsourcing*
- Conhecimento de alguns processos relacionados ao negócio do retalho (operacional e administrativo)

Como resultados do estágio ficam: o projecto “Comunicação de artigos”, a base do manual de qualidade para estes projectos e o estudo macro do processo de entrada e saída de artigos.

Em conclusão, os objectivos do estágio foram atingidos, embora se continuem a colocar muitos novos desafios nesta área tão promissora.

### 7.2. Futuro

Depois do que foi feito durante este estágio, ficam como acções futuras (por ordem cronológica) as seguintes:

- **“Comunicação de artigos” para Retek**

Dado que vai desaparecer a actual base de dados de artigos, torna-se necessário adaptar o *workflow* “Comunicação de artigos” de forma a continuar a importar a informação necessária. O que é necessário fazer nesta área, é articular com os responsáveis desta nova solução para tradução das consultas existentes (esforço será reduzido).

- **Criação plano de formações**

Dado que neste momento se começa a fazer o *Roll-Out* para as lojas, torna-se necessário criar uma estratégia para formar o universo de utilizadores (cada vez mais disperso e heterogéneo).

- **Abertura do workflow a agentes externos**

Um dos aspectos que será importante tratar, é o facto de quem não tem acesso à intranet da Sonae não poder usar o *workflow*. Os ganhos de abertura do *workflow* ao exterior, são a possibilidade de integração directa com fornecedores, reparadores externos, empresas da Inscó e da Star Turismo. Sendo assim, falta

fazer uma análise dos níveis de segurança a implementar para garantir o sigilo da informação.

- **Reformulação horizontal do *workflow* Sonae**

Esta acção já foi iniciada no presente estágio, existindo no âmbito do manual de qualidade um conjunto de propostas e lista de novas funcionalidades. Sendo assim, falta fazer um levantamento junto aos utilizadores para completar e enriquecer essa proposta.

- **Sincronização com o directório de perfis**

Necessário efectuar o estudo da nova tecnologia de directório que vai ser implementada na Sonae, de forma a utilizar o OID para sincronização de perfis.

- **Modelo de suporte**

Necessário evoluir o modelo de suporte, que permita um apoio de qualidade ao utilizador. As possibilidades de evolução são: completar a tabela que contém as formas de reagir a certas situações, abrir um canal de comunicação que permita a fácil interacção entre a *Novis* e o fornecedor de software e criação de funcionalidades *Workflow* para reportar automaticamente os erros.

- **Alargamento progressivo do “Comunicação de artigos”**

Necessário criar um plano de execução que permita, a curto prazo, completar o processo “Comunicação de artigos” e, a médio prazo, fazer a digitalização de todos os processos estudados, começando pelos mais críticos.

- **“Manutenção” da metodologia**

É necessário continuar a evoluir / afinar a metodologia criada no âmbito do estágio, de forma a servir de sustentação ao processo de digitalização que a Modelo Continente Hipermercados SA preconiza e que ambiciona a curto/médio prazo.

## Glossário

- **API** – Uma *Application Programming Interface* é uma peça de software que serve de intermediária entre camadas do mesmo programa ou com outros programas
- **ASN** - *Abstract Syntax Notation* é um standard que descreve um formato de mensagens para comunicação por uma rede. Divide-se em duas partes: a definição, que indica como interpretar uma sequência de dados e os dados
- **As-Is** – Estudo do processo tal como ele se desenrola. Consiste em verificar como está a decorrer o processo, sem quaisquer alterações/melhoramentos
- **BackOffice** – É o nome que se dá à parte da aplicação que contém as funcionalidades de administração
- **BES** - *Business Event System*, ver ponto 3.4.1
- **Busy Waiting** – O termo usado quando um processo está num certo ponto, a executar uma comparação (gastando para isso recursos de processamento)
- **Centralizado (artigo)** – Significa que o artigo é fornecido directamente pelo operador logístico da Sonae (Modis). Se não fosse centralizado, significava que o artigo era pedido directamente ao fornecedor
- **Checkout** – Pequena prateleira de venda de produtos, que se encontra junto às caixas de pagamento
- **Cluster** – é uma criação de grupos/blocos. Um *cluster* informático é constituído por um conjunto de processadores que estão a funcionar como se só existisse um
- **ERP** – É um software de gestão que tem como objectivo integrar todos os departamentos e funções num único sistema. Os exemplos mais conhecidos são: *SAP* e o *Baan*
- **Frederick Winslow Taylor** – ver ANEXO D
- **FrontOffice** - É o nome que se dá à parte da aplicação que contém as funcionalidades de uso geral (ou seja, não incluiu as funcionalidades de administração)
- **Gôndola** – É a designação dada a uma fila de prateleiras numa loja
- **Insígnias** – São uma espécie de marcas associadas à Sonae. Existem as seguintes: Continente, Modelo, Modelo Bonjour, Worten, Modalfa, MaxMat, SportZone, Vobis e MaxOffice. No Brasil existem: BIG, Nacional, Mercadorama e Cândia
- **KPI** – Key Performance Indicators são indicadores que servem para medir a qualidade do trabalho de algo
- **LDAP** – ver ANEXO F
- **Middleware** – ver API

- **ODBC** - Open Database Connectivity é um standard para acesso a qualquer base de dados. Sendo assim, funciona como API entre uma aplicação e uma base de dados. É através desta API que se envia o SQL para acesso às bases de dados
- **OutSourcing** – forma de desenvolvimento baseada na contratação de uma empresa externa, quando poderia ser realizado usando competências internas
- **PANDA** - Aplicação que armazena os custos associados a cada projecto. Tem como base o MS *Access* e permite o controlo orçamental e temporal de cada projecto
- **Passo** – Uma acção que não pode ser divisível e tem com objectivo contribuir para a realização de uma tarefa
- **Planograma** – É um mapa que demonstra como deve ficar o artigo numa gôndola
- **Publisher/subscribe** – este tipo de arquitectura corresponde a existir um conjunto de aplicações/pessoas que publicam informação e, um conjunto de pessoas que se inscrevem para acesso a essa informação. Por exemplo: existe um conjunto de jornais diários (publishers) e cada pessoa pode criar uma assinatura de qual/quais quer receber (subscribers)
- **RAC** - Real Application Clusters, ver ponto 3.4.2 e ANEXO I
- **RFC** – é uma forma de integração possível, que consiste em chamar funções existentes noutra aplicação (remotamente), recebendo posteriormente o resultado dessa mesma função. A vantagem é reduzir as necessidades de largura de banda para comunicar entre aplicações, porque apenas é enviado o resultado e não o universo de dados que levam a esse resultado. A desvantagem é o facto de ser necessário efectuar desenvolvimentos em ambas as aplicações
- **ROI** – *Return of Investment* é o valor que se espera obter com algo. Na área de informática, por vezes é difícil estimar esse valor, por isso, o uso desta terminologia está associado a funcionalidades que podem ser usadas em mais que uma situação
- **Role** – ver ponto 2.1.7
- **Roll-Back** – significa “desfazer”, ou seja, efectuar a operação inversa à última operação. Este termo é usado frequentemente em bases de dados, e corresponde a recuperar o último estado estável da informação
- **Roll-Out** – Disponibilização da aplicação a todos os utilizadores
- **Tarefa** – Uma actividade que só ficará completa, mediante a realização de um conjunto de passos.
- **To-Be** – Criação de um novo processo com o objectivo de melhorar o existente. Consiste em pegar no modelo As-Is e optimizá-lo
- **Topo** – pequeno espaço de exposição de artigos que se encontra nas extremidades das gôndolas
- **WfMC** - *Workflow Management Coalition* ver ANEXO C

- **SQL** – *Structured Query Language* é a linguagem usada para efectuar pesquisas e alterações a bases de dados
- **XML** – *Extensible Markup Language* serve para etiquetar a informação a apresentar. Sendo assim, não define o aspecto mas sim a estrutura dos dados.
- **XSL** - *Extensible StyleSheet Language* descreve como deve ser apresentada a informação. Sendo assim, pega numa árvore de XML, introduz-lhe a informação sobre como deve ser apresentado (Bold, em que local da página, ...) e envia para o browser que irá apresentar a página definida

## Referências e bibliografia

- 1999 European Business Process and Workflow Conference
- Application Developer's Guide - Advanced Queuing, Release 8.1.5
- Apresentação nova estrutura da DSI, Miguel Ângelo Santos
- Comunicação de Artigos – Proposta de Especificação, Carla Nora
- Dicionário de economia digital, Computerworld
- Entrevista de Carlos Latourrette, intitulada “Modelo Continente: provocar mudança digitalizando processos”
- Exemplos de implementação: Brithish Telecom, CERN, Hong Kong Telecom, McKesson (medicina), WA Water & Rivers Commission
- <http://technet.oracle.com> – Oracle
- <http://www.bpmi.org/>
- [http://www.cyberstudio.fr/gamme\\_opencs\\_uk.asp](http://www.cyberstudio.fr/gamme_opencs_uk.asp) - OpenCS
- <http://www.e-workflow.org/>
- <http://www.ideafinder.com/history/inventors/taylor.htm> - Taylor
- <http://www.sonae.pt> – página de apresentação da Sonae
- <http://www.ultimus.com> - Ultimus
- <http://www.umich.edu> – Universidade de Michigan
- <http://www.w3.org>
- <http://www.waria.com/> - WARIA
- <http://www.wfmc.org/standards/standards.htm>
- <http://www.whatis.com> – site com dicionário de termos técnicos
- Mail de Victor Martins - Oracle
- Manual de qualidade, João Braga
- Oracle Workflow Guide, release 2.6.1
- Oracle9i Real Application Clusters, Oracle
- SAP Business Workflow, por Michel Berlier – AEGIS Consulting
- Understanding LDAP, IBM
- Workflow Management Coalition - The Workflow Reference Model
- Workshop tecnológico Oracle – Link – Sonae, João Braga

# ANEXO A: Direcção de Sistemas de Informação

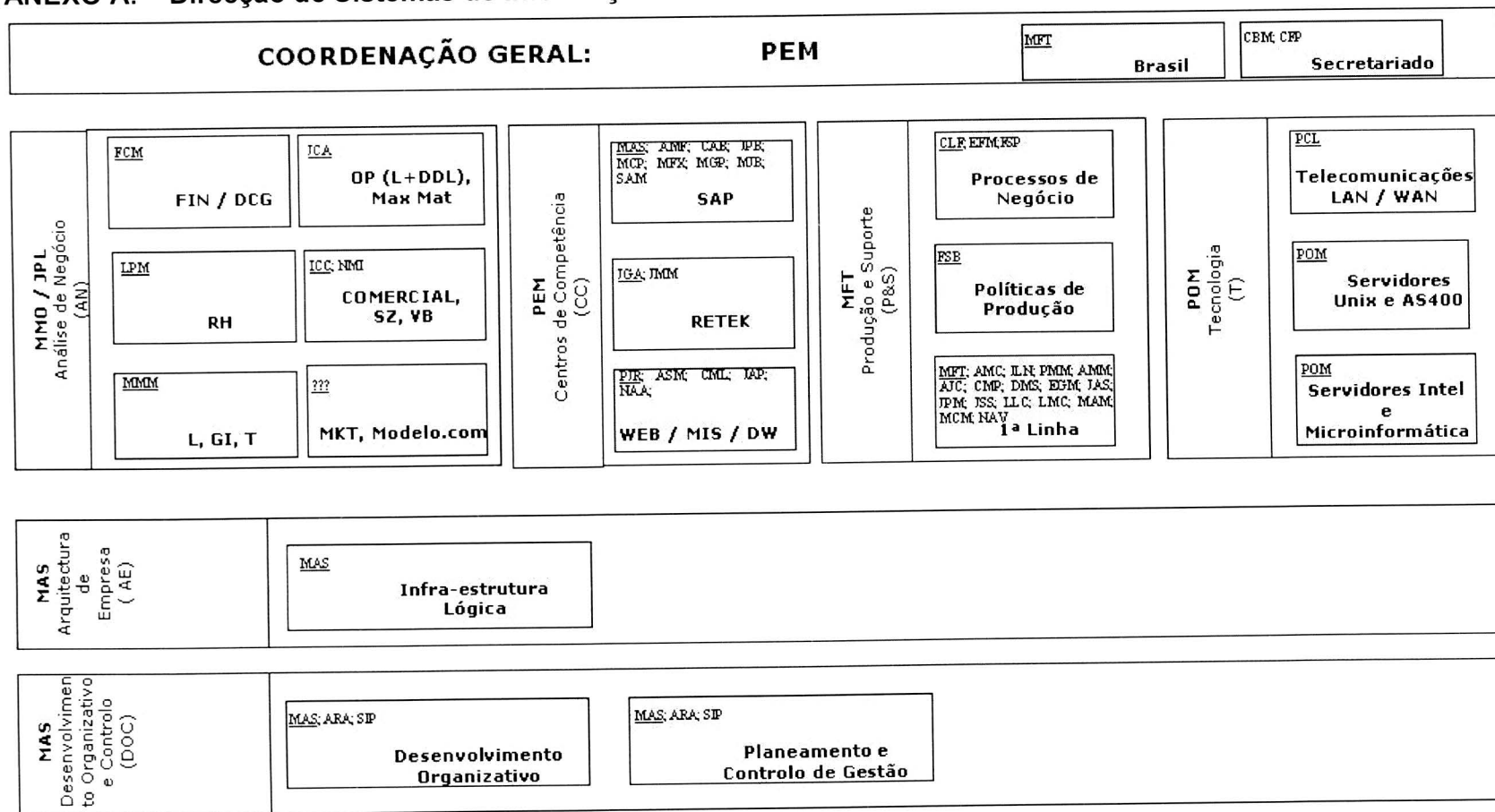


Figura A-1 : Organograma da DSI

## ANEXO B: Plano do estágio

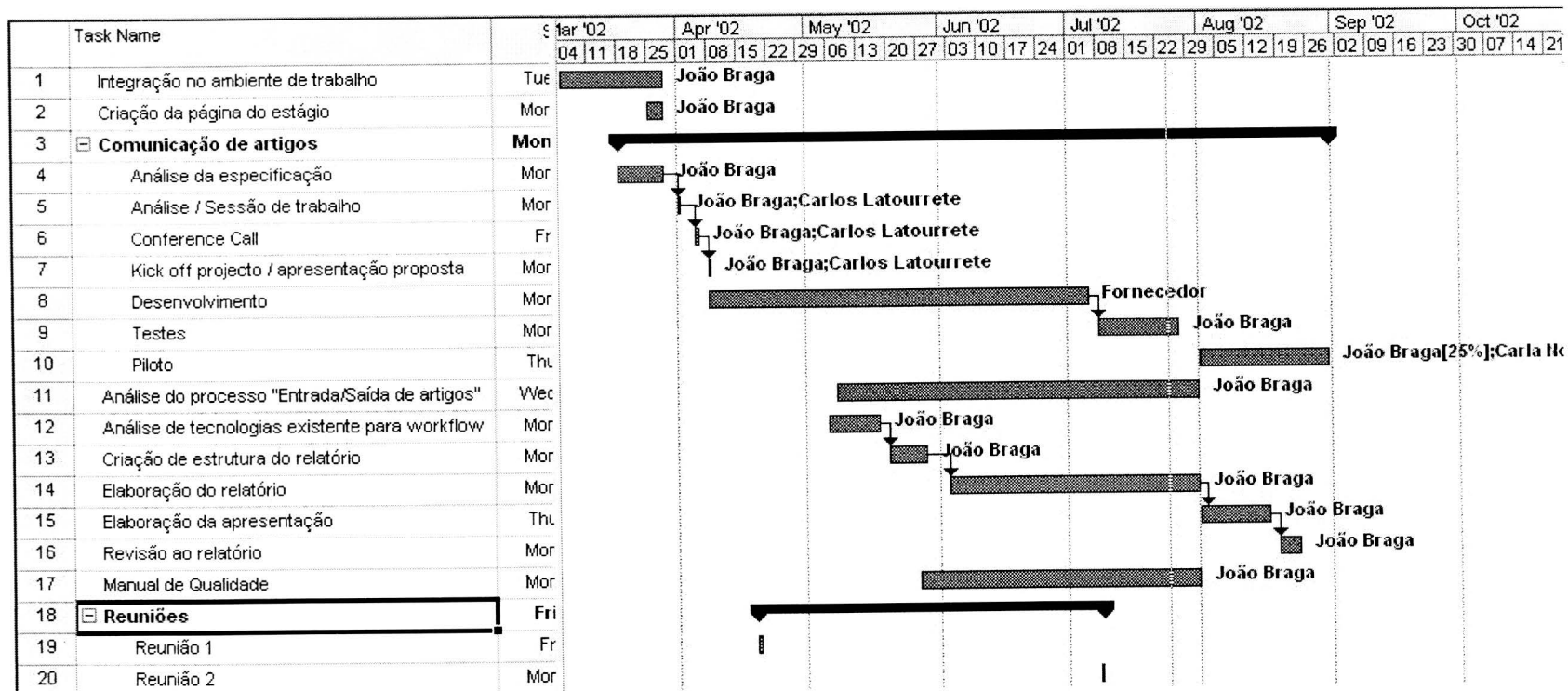


Figura B- 1 : Plano do estágio

### **ANEXO C: O que é a WfMC?**

A *Workflow Management Coalition (WfMC)* é constituída por 285 membros em todo o mundo. Foi fundada em Agosto de 1993 e é uma empresa sem fins lucrativos que tem como missão garantir:

- Satisfação dos clientes das ferramentas de *workflow*
- Diminuição do risco das ferramentas de *workflow*
- Expansão do mercado das ferramentas de *workflow*
- Garantir interoperabilidade entre produtos de *workflow* e outras aplicações

Tudo isto é conseguido, criando normas sobre as ferramentas relacionadas com o *workflow*

## **ANEXO D: Taylor (início do workflow)**

*Frederick Winslow Taylor* nasceu em 1856 em *Germantown* (hoje parte de Filadélfia) e morreu em 1915. A sua formação era engenharia industrial e durante toda a sua vida, preocupou-se com o melhoramento da produtividade nas empresas.

Dessa sua preocupação, além da automatização das tarefas executadas nessa época, ficou como principal princípio para o *workflow* o “*Taylorismo*”. O taylorismo defende como forma de melhorar a eficácia no desenrolar dos processos, a especialização dos intervenientes. Esta filosofia faz com que o tempo de execução diminua e as pessoas não precisem de ter muita formação (só relativa à sua tarefas). Mas, a especialização levou a uma ampla discussão, acerca da possível diminuição de motivação por parte dos executantes (daí terem surgido outras filosofias).

Depois desta introdução, facilmente se verifica que o *workflow* tem a sua génese nesta teoria, porque cada utilizador é chamado no processo de forma recorrente e sempre para efectuar a mesma tarefa. Quanto aos problemas que *Taylor* deparou na época, hoje não se verificam, porque com as ferramentas actuais, as pessoas deixam de efectuar as tarefas repetitivas de menor valor, para executar apenas aquelas que necessitam de um conhecimento humano.

Embora estejamos livres dos problemas que *Taylor* teve, hoje em dia, temos de nos preocupar com a formação nas tecnologias de informação (a dar aos utilizadores), definição das fronteiras de cada acção (de forma a podermos especializar as acções), a sensação de perda de poder das pessoas que estão habituadas ao papel e alguns aspectos legais associados aos documentos.

**ANEXO E: Questionário de levantamento de processo**

**Nome do processo**

**Actividade:** *(nome)*

---

**Precedências:** *(contexto no processo)*

---

**Dados necessários: (inputs e outputs)**


**Quem são os intervenientes, com que usos:** *(perfil de utilização)*

---

**Tempo dispendido:** *(análise de ganho)*

---

**Resultado:** *(qual o valor acrescentado associa à actividade)*

---

**Processo de comunicação do resultado:** *(análise de ganho)*

---

**Tempo de comunicação:** *(análise de ganho)*

---

**Interacção com outros (sistemas/pessoas):** *(necessidade de integração exterior – eventos)*

---



---

**Problemas ocorridos:** *(análise de ganho – resolução de erros)*

---



---

**Que formação é necessária:** *(análise de ganho)*

---

**Excepções:**

---



---



---

**Outros:**

---



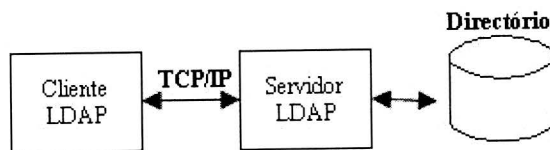
---



---

## ANEXO F: LDAP

LDAP é a sigla de *Lightweight Directory Access Protocol*, e define o protocolo e as funcionalidades para acesso a informação armazenada em directórios. Como o próprio nome indica, o LDAP é uma versão mais eficiente (“mais leve”) que o DAP, porque utiliza o TCP/IP em vez do OSI (redução de camadas), simplifica e elimina as funcionalidades já existentes e usa caracteres para representar a informação (em vez do sintaxe ASN.1 –*Abstract Syntax Notation One*).



**Figura F- 1 : Comunicação LDAP**

Um directório é uma espécie de base de dados, onde ficam armazenados os dados sob forma de uma árvore.

As principais diferenças entre um directório e uma base de dados convencional são: é otimizado para a leitura de informação (sendo a escrita muito penalizada), não suporta transacções (melhora a performance e como não há escrita muitas vezes não existem problemas de inconsistência significativos) e a linguagem de pesquisa foi simplificada e otimizada (em substituição do SQL - *Structured Query Language* -que eleva o custo das pesquisas).

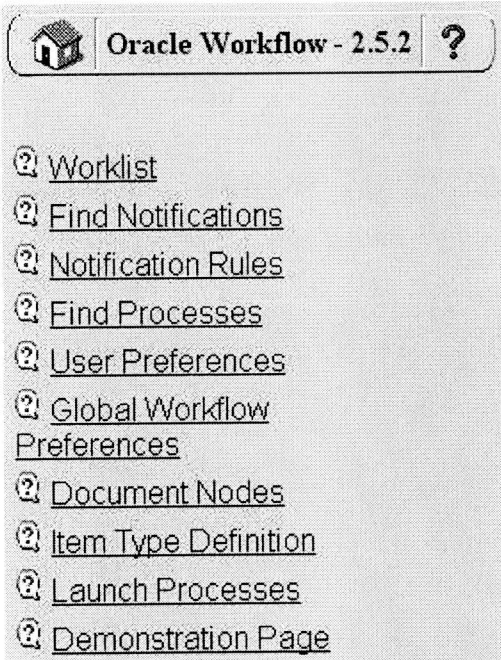
**ANEXO G: “Monitor” (Ferramenta Oracle para administração)**

A ferramenta de administração fornecida pela Oracle (“Monitor”), tem as seguintes funcionalidades:

**Figura 14 : Opções no "Monitor"**

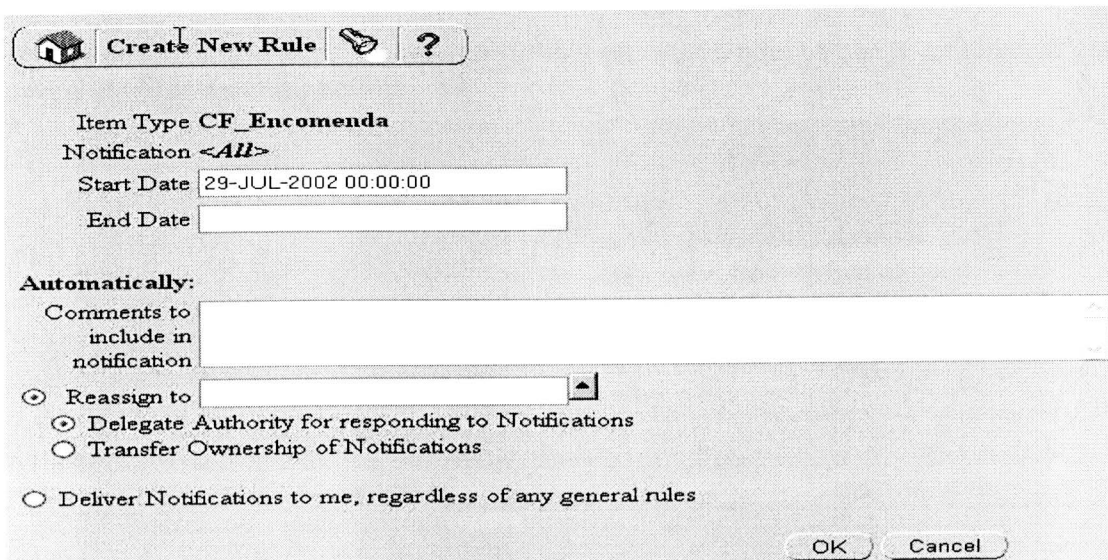
Como podemos ver nesta figura, as funcionalidades fornecidas são: a possibilidade de ver a sua lista de trabalho (worklist), procurar notificações na sua lista, procurar/gerir processos, definir regras do seu perfil, gestão de preferências do utilizador (a nível da ferramenta), gestão de documentos, ver a informação acerca da definição do processo, lançar processos e por fim existe por defeito um exemplo de workflow para se fazerem alguns testes.

Convém frisar que todas estas opções só aparecem no caso de ter perfil de administrador, porque senão não apareceriam as opções de controlo do processo.



**Figura G- 1 : Opções no "Monitor"**

Uma das opções mais utilizadas pelo utilizador comum é a possibilidade de marcar um período de tempo em que delega as suas notificações. Esta funcionalidade é útil para por exemplo marcar as férias. Sendo assim, os processos não ficarão bloqueados.



**Figura G- 2 : Gestão de férias no "Monitor"**

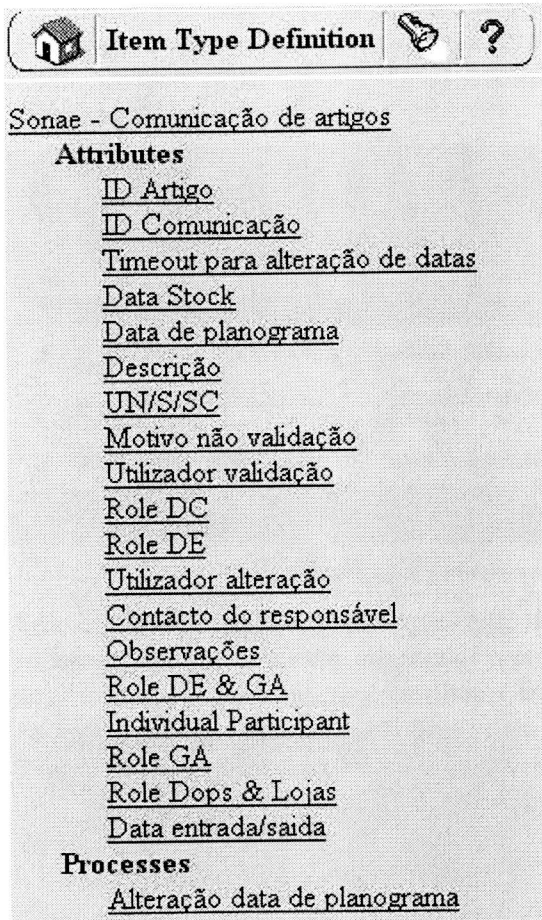


Figura G- 3 : Definição do processo "Monitor"

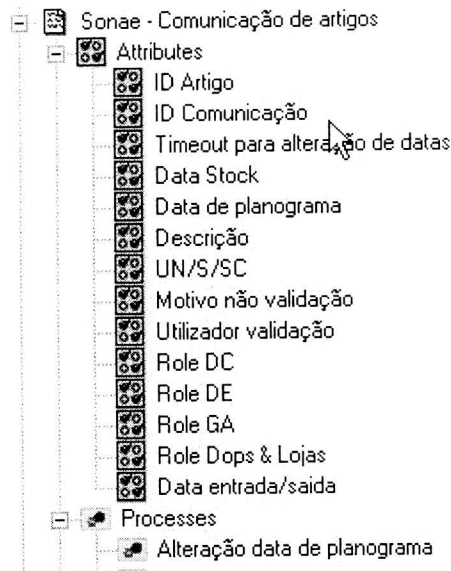


Figura G- 4 : Definição do processo "Builder"

À esquerda temos uma imagem do "Monitor" e em cima uma imagem do "Builder".

No "Monitor" podemos ainda, analisar as definições do processo, como se faz na ferramenta de desenho. Embora, não possa ser modificado directamente no "monitor", ou seja, as modificações ao processo deverão ser feitas na ferramenta de desenho do processo "Builder".

Outra funcionalidade fornecida é a procura dos processos, quer os que já terminaram que os que ainda se encontra activos.

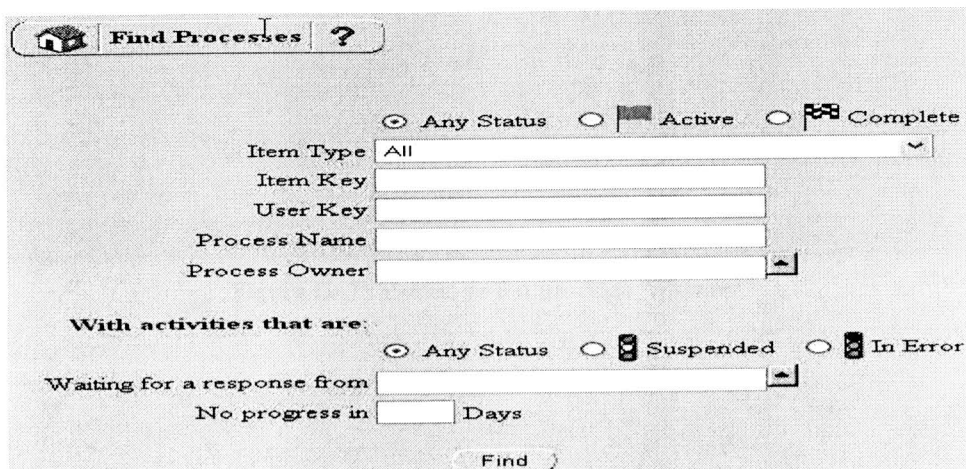


Figura G- 5 : Procura de processos "Monitor"

Depois de pesquisado o processo, pode-se fazer uma análise ao nível do percurso percorrido e os tempos associados. Esta funcionalidade permite dar aos utilizadores o estado do seu processo e dar aos administradores uma forma de verificar possíveis estrangulamentos.

Notifications List ?

Valida Factura: CF\_Facturas, CAFONSECA - 0066002054  
Started: 04-JAN-2002 ( 3 Days )

Date	Who	Activity	Started	Duration	Result
✓	LUCILIA MARIA SILVA NEVES CANDIDO	Associação de Encomenda a factura	07-JAN-2002 10:24:08	1 Minutes 35 Seconds	No
✓	LUCILIA MARIA SILVA NEVES CANDIDO	Notificação Comprador Conferência Factura já efectuada	07-JAN-2002 10:25:43	4 Minutes 23 Seconds	Yes
✓	MIGUEL ANJOS RODRIGUES	Notificação Aprovação Factura	07-JAN-2002 10:30:06	1 Hours 34 Minutes	Approve

View Diagram Advanced Options

Figura G - 6 : Percurso de um processo "Monitor"

Adicionalmente, podemos ver o processo directamente na sua representação gráfica. Aqui podemos, abortar processo, suspender o processo, delegar uma notificação, responder a uma notificação e visualizar os atributos desta instância do processo.

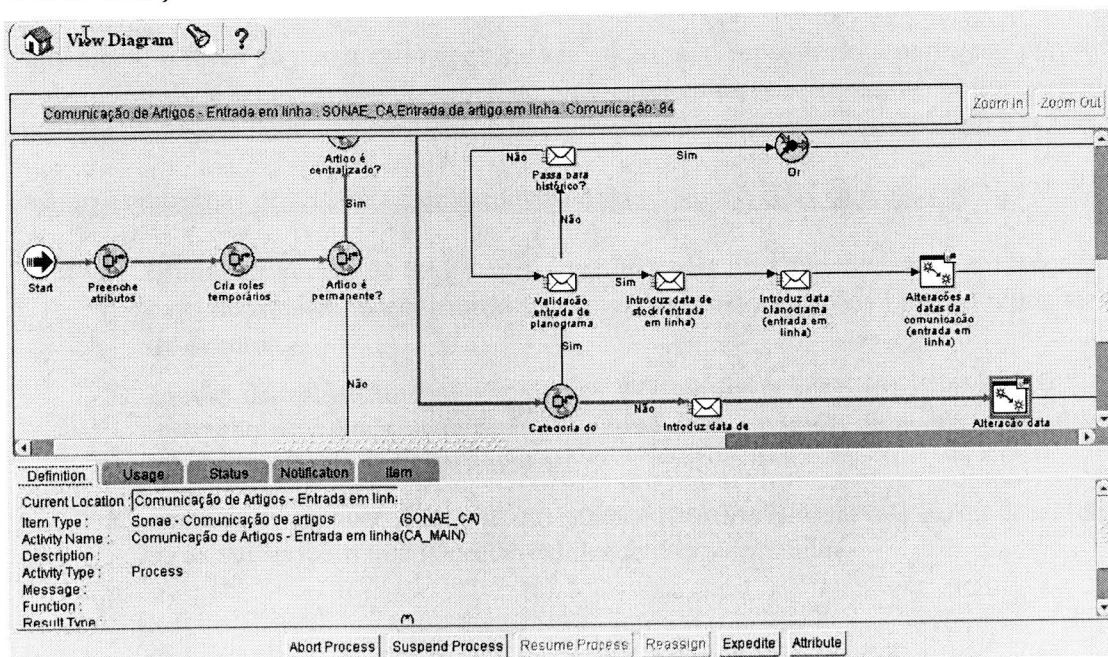


Figura G - 7 : Gestão de um processo "Monitor"

## ANEXO H: Exemplos de vantagens dos eventos com o exterior

De seguida, vamos apresentar um exemplo sobre a melhoria introduzida pelo uso de eventos na comunicação com o *ERP SAP*. O exemplo é o seguinte:

Tenho um *workflow* que precisa de receber uma "ordem" do *SAP* para continuar. Se usar eventos quais serão as vantagens, quando eu:

- Mudar de versão de *SAP*
- Mudar de aplicação (em vez de ser *SAP* passar a ser outra qualquer)
- Desistir de precisar dessa ordem

No caso de haver a necessidade de ter uma "ordem" do lado do *SAP* para iniciar um *workflow*, o facto de controlar essas acções através de eventos vai obter uma nova camada que permitirá controlar QUE eventos devem ser despoletados ("ordens" do *SAP...*), QUANDO, por QUEM, associar regras para os chamar, poderão activá-los ou não conforme as necessidades, e sobretudo, poderão com toda a tranquilidade fazer actualizações às versões das aplicações com menos percalços, já que o repositório de Eventos não será posto em causa. É evidente que tudo dependerá do nível de "customizações" existentes no nosso cenário, e da necessidade de os passar para esse repositório.

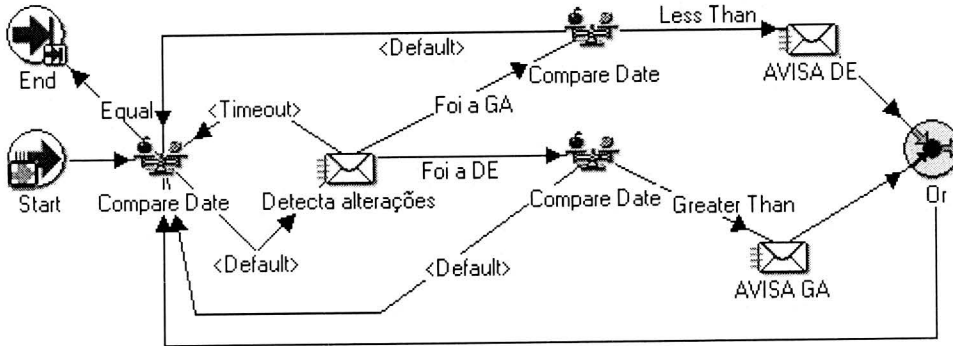
Mais ainda, o facto de poder criar regras do negócio num repositório à parte, vai permitir também reutilizá-las noutras aplicações ou ambientes que poderão existir. O que vem de encontro da necessidade manifestada de modularização e reutilização de código e regras.

Quanto às questões postas, são precisamente casos em que o *BES* vem ajudar:

1. Mudar de versão de *SAP* - serão minimizadas as dificuldades na mudança, no caso de transportarem muitas das regras e "customizações" para o repositório de eventos.
2. Mudar de aplicação (em vez de ser *SAP* passar a ser outra qualquer) - Esta operação será ainda mais fácil pelo facto do repositório de eventos permitir a tal abstracção do lado aplicacional.
3. Desistir de precisar dessa ordem - Neste caso basta desactivá-la, ou desactivar só as subscrições que não querem que sejam submetidas.

**ANEXO I: Exemplo de vantagem dos eventos com o utilizador**

Os eventos permitem por exemplo eliminar as falsas tarefas da worklist do utilizador. Senão vejamos:

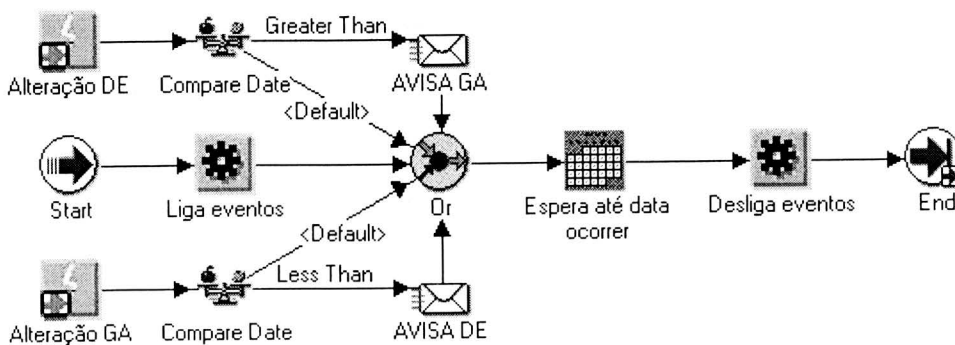


**Figura I- 1 : Processo de alteração de datas, sem uso de eventos**

Este processo, tem como atributos duas datas que podem ser alteradas por intervenientes diferentes (GA e DE), mas quando houver uma alteração tem de se notificar a outro interveniente. Além disso, a data do interveniente GA terá de ser sempre menor que a data do interveniente DE. Esta possibilidade de alteração só pode ser exercida enquanto a data do interveniente DE não ocorrer (chegar a essa data).

A solução apresentada na figura anterior (*Oracle Workflow 2.5*), começa por comparar se data já ocorreu e depois coloca uma notificação na worklist de cada interveniente (DE e GA). Essa notificação tem um *timeout* de 24 horas, para que possa estar constantemente a verificar a data (só de 24 em 24 horas é que muda a data). Quando se altera a data, procede-se à validação explicada anteriormente, e caso seja válida notifica-se o outro interveniente.

A mesma solução usando o *Oracle Workflow 2.6*, seria a seguinte:



**Figura I- 2 : Processo de alteração de datas, usando eventos**

A solução apresentada, começa por activar os eventos (nova funcionalidade) e de seguida calcula o tempo de espera até à data ocorrer. Quando qualquer utilizador alterar uma data, o evento correspondente é lançado de forma a se validar a data e recalculer o tempo de

espera. Depois de ocorrer a data desligam-se os eventos, para que mais nenhum utilizador consiga alterar nenhuma das datas.

As vantagens desta nova solução são:

- Eliminação da actividade facultativa da *worklist*, dado que as datas não precisam de ser obrigatoriamente alteradas. O ganho seria a eliminação da confusão introduzida por uma notificação que está sempre na *worklist*
- Eliminação do *timeout* sucessivo, necessário para actualização da comparação para determinar a ocorrência da data
- Outra vantagem seria o facto de o evento poder ser uma ocorrência noutra sistema, ou seja, vamos supor que a alteração da data era feita na aplicação “xpto”, bastava que a aplicação lançasse um evento para o *Oracle*

Em conclusão, o uso da nova funcionalidade do *Oracle Workflow 2.6*, permite melhorar a interacção com os vários intervenientes (aplicações ou pessoas) e diminuir a necessidade de processamento (elimina nós e transições desnecessárias).

## ANEXO J: Cenários RAC

Depois de descritas as vantagens da utilização desta solução, vamos analisar quais os cenários possíveis. Este cenários dependem apenas da arquitectura física que se optar, sendo assim os cenários possíveis são:

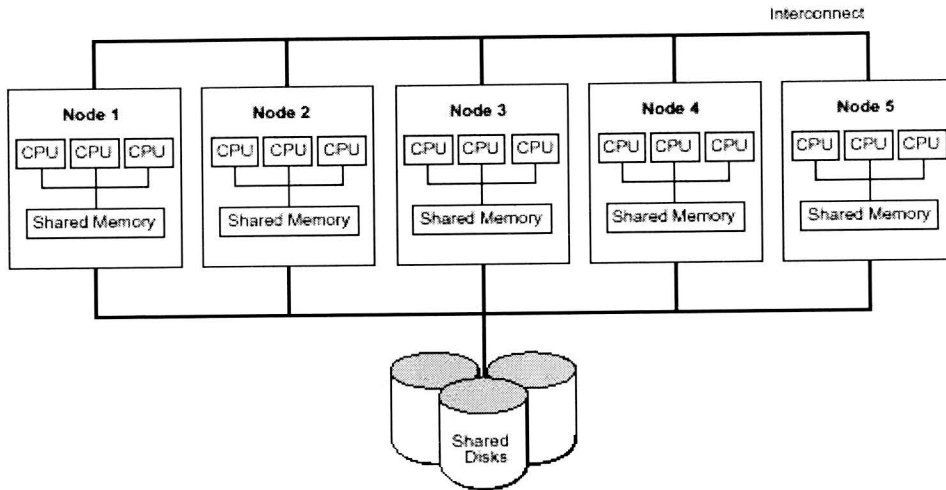


Figura J- 1 : RAC com acesso uniforme a disco

Este primeiro cenário consiste num conjunto de nós, que acedem a uma bateria de discos comum. As desvantagens estão em ter um ponto crítico de falha (o disco) e o possível congestionamento do acesso a disco (que limita o crescimento desta arquitectura). A vantagem é que todos os *CPU* acedem a disco com custo igual e o *backup* e controlo são centralizados.

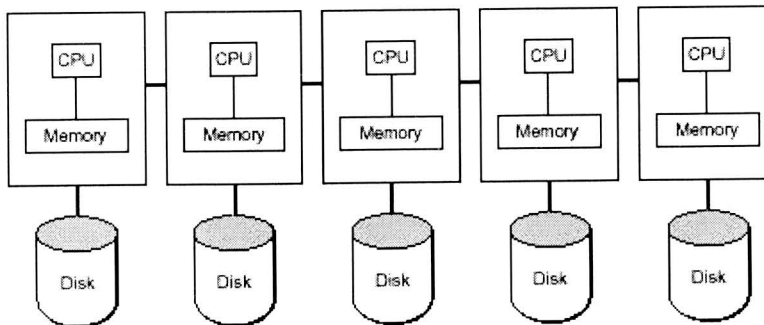


Figura J- 2 : RAC com acesso não uniforme a disco

Este segundo cenário consiste num conjunto de nós, que têm um disco independente, mas que podem aceder aos outros discos como um todo (embora com custo maior). As desvantagens são o acesso ao disco de outro nó (não pertencente ao que quer aceder) e o facto ter de existir um serviço que crie um disco virtual (não existe no RAC e tem de ser suportada a nível do sistema operativo). A vantagem é o facto desta arquitectura poder crescer livremente e o disco deixar de ser um ponto de falha.

Qualquer um destes cenários é perfeitamente viável, dependendo apenas dos risco que se pretende incorrer (desvantagens) e o custo a incorrer.

### ANEXO K: Suporte da solução *OID*

De forma a se poder replicar e manter a replicação actualizada, a arquitectura física a implementar seria a seguinte:

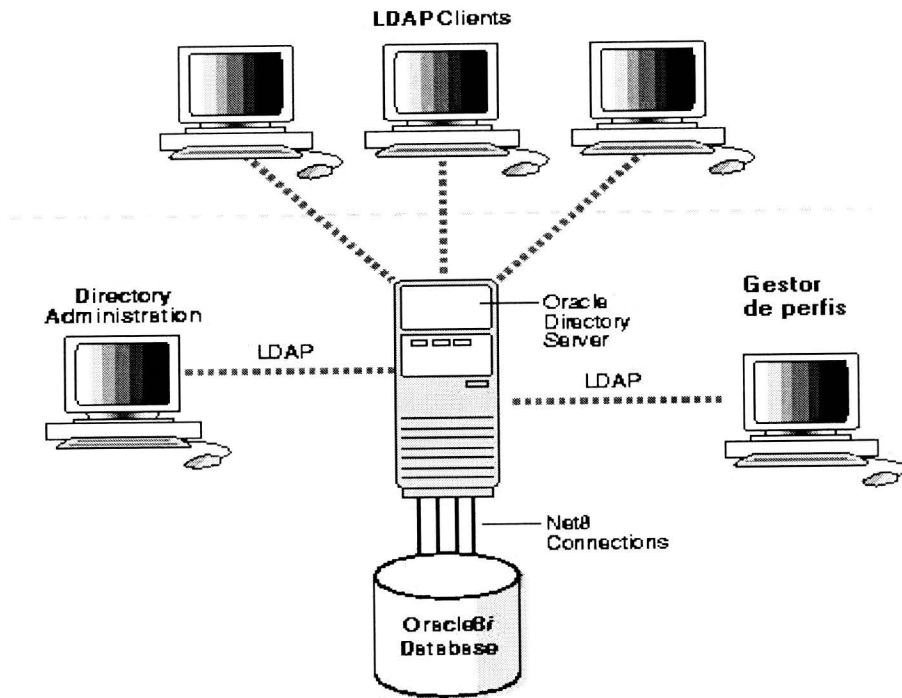


Figura K- 1 : Arquitectura física do *OID*

A arquitectura lógica (figura seguinte) que suporta esta solução tem os seguintes componentes:

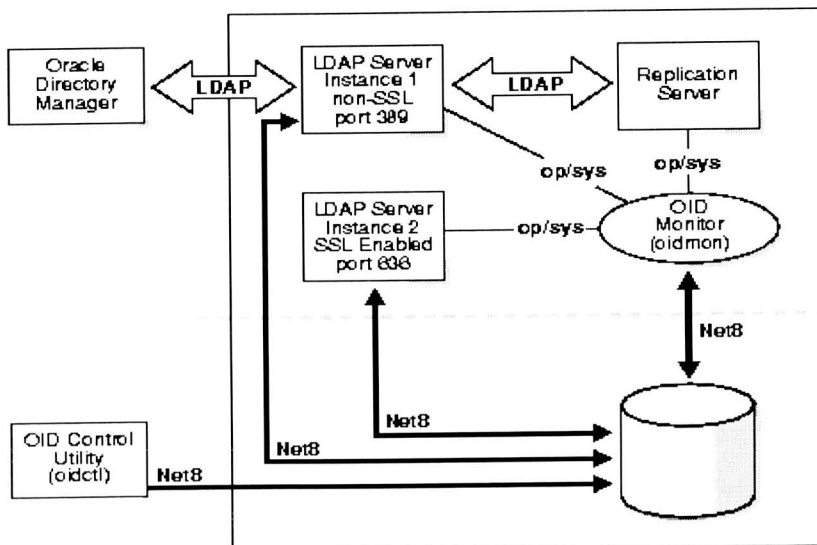


Figura K- 2 : Arquitectura lógica do *OID* (um nó)

- **LDAP Server**

Responsável por receber e responder aos pedidos. O número de instâncias activas é limitado pelo número de portos livres.

- **OID Monitor (oidmon)**

Executa os comandos necessários à gestão de todos os componentes. Inicializa, monitoriza e termina o *LDAP*; controla o *Replication Server* e interpreta os comandos enviados pelo *OID Control Utility*

- **Replication Server**

Serve para gerir as actualizações de dados entre nós, ou seja, quando se faz uma alteração num nó a afectação de todos os outros nós é controlada por este módulo. Só pode existir no máximo um *Replication Server* por nó

- **OID Control Utility**

Serve para configurar toda a estrutura de directórios e controlo de parâmetros. Em anexo pode ser encontrada uma interface de exemplo

- **Base de dados**

Base de dados para armazenar a informação acerca dos directórios. É recomendado existir uma base de dados dedicada a armazenar esta informação. A base de dados pode ser local ao nó ou residir noutro nó.

O processo desencadeado por pedido é o seguinte (figura seguinte):

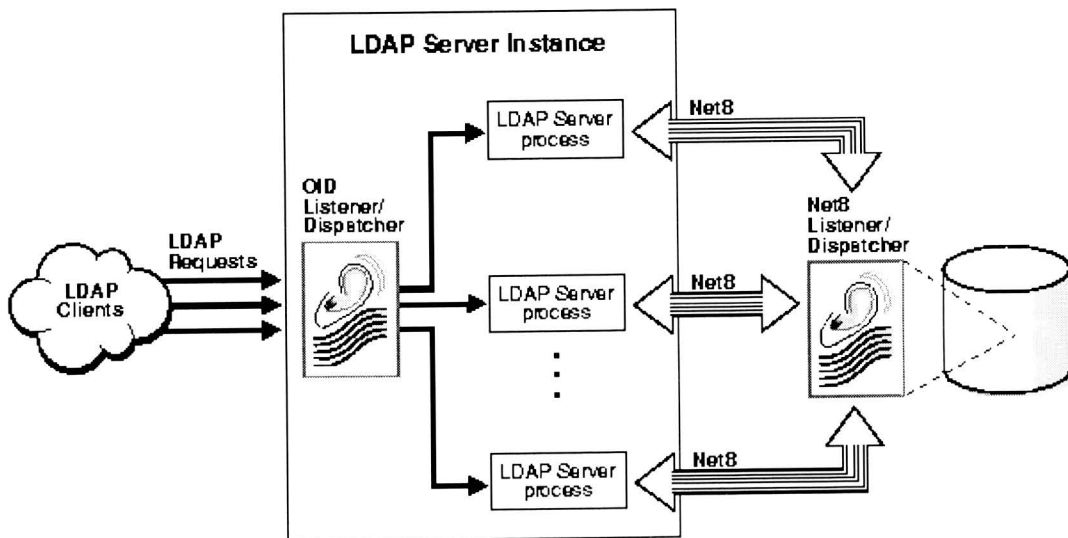


Figura K-3 : Funcionamento. Componentes *run-time*

1. Criação do pedido (*SSL*, autenticação, *query*)
2. Se o pedido tiver sido feito via *Oracle Directory Manager* o pedido é convertido de *Java Native Interface* para C API. Se o pedido for feito via linha de comandos não é necessário converter

3. Envio do pedido de conexão ao directório pretendido, usando LDAP para comunicar com o *directory server*
4. O *directory server* autentica o utilizador e faz o pedido de busca da informação
5. O pedido é convertido de *LDAP* para *OCI*<sup>12</sup>
6. A instância de Base de dados executa o pedido e a informação volta pelo percurso inverso

Concluindo, esta solução torna mais eficaz o acesso à informação relativa aos perfis e além disso, não incorre em custos suplementares porque a implementação desta solução passa por utilizar melhor as funcionalidades já existentes nas nossas bases de dados.

---

<sup>12</sup> Oracle Call Interface

## ANEXO L: Estrutura da documentação a criar

O objectivo da preocupação com a documentação, é garantir que todas as aplicações realizadas se encontram bem documentadas, de forma a facilitar a sua manutenção. Além disso, servirão para que durante o decorrer do projecto haja visibilidade do estado do mesmo e das acções e opções tomadas. Para garantir isso, cada documento deve ser independente, não sendo para isso usadas referências para outros documentos, sendo assim quando for necessário referenciar transcrever-se-á o necessário.

De seguida, serão especificados os pontos mínimos a frisar em cada documento, mas poder-se-ão colocar outros pontos desde que enriqueçam o documento.

- **Especificação funcional**

A documentação funcional é obtida com base no documento de especificação criado pela Sonae. No documento a criar é requerida uma linguagem clara, simples e diagramas realizados no *Builder*. De forma que o cliente possa verificar se foi compreendido o que ele quer com este projecto.

Sendo assim, os pontos mínimos a frisar no documento são:

### 0. Introdução ao documento

O objectivo neste ponto é apresentar o documento, tendo em conta as diferenças com versões anteriores, objectivo do mesmo e o enquadramento

### 1. Introdução

O objectivo deste ponto é introduzir o projecto em questão. Por isso, será requerida breve explicação do problema e/ou o âmbito do projecto neste problema

### 2. Requisitos do sistema (Funcionais / Não Funcionais)

O objectivo deste ponto é enumerar quais os requisitos que a aplicação deve suportar. A enumeração deverá ser feita separando os requisitos em funcionais e não funcionais. Entende-se como requisito funcional tudo que a aplicação deve suportar a nível de utilizador. Entende-se como requisito não funcional, os requisitos que terão de ser cumpridos para que a aplicação funcione (Exemplos: Quero em plataforma *Windows* com browser Xpto, Tempo máximo de espera será X minutos).

Dentro de cada um destes blocos deverá ser subdividido em *BackOffice* e *FrontOffice*. Entende-se como *BackOffice* todos os requisitos que estão associados à parte da gestão da aplicação (Exemplos: Gestão de tabelas, Políticas de acesso, ...). Entende-se por *FrontOffice* toda a parte de uso mais corrente / massivo e que não tem como objectivo gerir a aplicação, mas sim promover um conjunto de funcionalidades que suportem um certo problema.

Finalmente, por requisito deverá haver uma descrição da funcionalidade em questão, tendo em conta a enumeração dos dados necessários.

Exemplo:

1. FrontOffice

- 1.1. Funcionais
  - 1.1.1. Requisito X
- 1.2. Não Funcionais

## 2. BackOffice

.....

### 3. Utilizadores do sistema

O objectivo deste ponto é enumerar os utilizadores do sistema. Sendo assim deverá ter dois grandes blocos, que são: os utilizadores e seus atributos e as funcionalidades que podem ter acesso.

No primeiro, deverão ser descritos de forma clara quantos são os utilizadores alvo, que tipos de utilizadores existem e quais os atributos de cada um. Entende-se como utilizadores uma pessoa ou um role.

No segundo ponto estarão as afectações dos requisitos funcionais a cada utilizador. Essa apresentação poderá se feita com recurso a um tabela de afectação e/ou usando os diagramas UML correspondentes (Diagramas de casos de uso)

### 4. Integração com outros sistemas

Este ponto tem como objectivo, enumerar qual a interacção com outros sistemas já existentes. Sendo assim, deverá aparecer por sistema qual a sua interacção (fornecimento de dados, eventos gerados, ...) com a nova aplicação e vice-versa

### 5. Processo

Este ponto tem como objectivo, apresentar o processo a desenvolver e as medidas a associar. Sendo assim, deverá ter um diagrama “*As-Is*” do processo, desenhado no *builder*. Esse diagrama deverá ser seguido por uma descrição ao nível da actividade, tendo em conta os seguintes pontos: Tipo de Actividade, Descrição da actividade, Interveniente, *Timeout*, Descrição, *Inputs* e os *Output's*.

Depois de descrito o processo, deverão ser apresentados os KPI a retirar do processo, de forma a efectuar o controlo do mesmo. Em casos em que se tenha optimizado o processo, deverá prever a descrição do seu modelo “*To-Be*”

### 6. Interfaces gráficas

Este ponto tem como objectivo, apresentar quais as interfaces a desenvolver para implementar as funcionalidades especificadas. Ficando claro, o *layout* final, dados e funcionalidades a suportar por cada ecrã

### 7. Validações junto cliente

Este ponto tem como objectivo, apresentar quais foram as validações efectuadas pelo cliente e que não estavam claras na especificação por ele fornecida

### 8. Riscos

Este ponto tem como objectivo, apresentar quais são os riscos previstos. Os riscos a apresentar deverão servir para alertar sobre problemas na implementação, na

volatilidade de algum requisito (exemplo: IVA) e outros riscos que sejam significativos

## 9. Glossário

Este ponto tem como objectivo apresentar conceitos que tenham sido usados durante o documento

## 10. Anexos

Este ponto servirá como repositório de documentos, tabelas, mails, ... e outros elementos que complementem o documento em questão

- **Especificação técnica**

A documentação técnica é obtida com base no documento de especificação funcional. No documento a criar é requerida uma linguagem clara, simples e diagramas realizados no *builder*, mas o nível de detalhe deve ser mais profundo. Pois deverá ser bem descrito como fazer para se atingir o especificado, ou seja, este será um documento que servirá como guia para a criação da aplicação. Os receptores deste documento serão pessoas que têm conhecimentos mais apurados de criação de aplicações.

Sendo assim, os pontos mínimos a frisar no documento são:

### 0. Introdução ao documento

O objectivo neste ponto é apresentar o documento, tendo em conta as diferenças com versões anteriores, objectivo do mesmo e o enquadramento

### 1. Introdução

O objectivo deste ponto é introduzir o projecto em questão. Por isso, será requerida breve explicação do problema e/ou o âmbito do projecto neste problema

### 2. Arquitectura física

O objectivo deste ponto é descrever como funciona a nível físico (hardware e componentes) a aplicação a desenvolver. Sendo assim são necessários dois diagramas.

Um primeiro diagrama que retrate a Topologia a nível do hardware, necessária para suportar a aplicação (arquitectura alvo). Esta descrição deve conter um diagrama do estilo do diagrama 2 (em anexo), acompanhado por uma descrição do que está a ser suportado em cada nó.

O segundo diagrama descreverá a arquitectura dos Objectos necessários em *runtime*. Neste diagrama, serão apresentados os vários objectos (COM+ e outros) e as suas relações entre eles ou com livrarias existentes. Mais uma vez, este diagrama terá associada uma descrição

### 3. Arquitectura lógica

O objectivo deste ponto é descrever quais as camadas existentes na aplicação. A forma de descrição a adoptar será um diagrama, acompanhado por uma descrição

#### 4. Modelo de dados

O objectivo deste ponto é apresentar todo o modelo de dados a criar, sendo assim deverá conter um diagrama de classes que demonstre as tabelas e os relacionamentos a criar na base de dados. Este diagrama será seguido de uma descrição que indicará os tipos de cada campo, as restrições que tem de obedecer, valores por defeito e se o campo é obrigatório ou não

#### 5. Interfaces gráficas

O objectivo deste ponto é especificar as interfaces gráficas que vão suportar a aplicação. Este ponto será um melhoramento do ponto similar do documento de especificação funcional

#### 6. Processo

Este ponto tem como objectivo, apresentar o processo a desenvolver. Sendo assim, deverá ter um diagrama “*To-Be*” do processo, desenhado no *builder*. Este diagrama, já poderá conter nós relativos às necessidades de codificação (Exemplo: nós para iniciação de dados, nós para criação de *roles*, ...). Esse diagrama deverá ser seguido por uma descrição ao nível da actividade, tendo em conta os seguintes pontos: Tipo de Actividade, Descrição da actividade, Interveniente, *Timeout*, Descrição, *Inputs* e os *Output* 's.

Este ponto é uma evolução do ponto similar do documento de especificação funcional. O melhoramento será a introdução de nós de controlo, necessários à implementação do processo (actividades do tipo “*Wait For Flow*”, ...)

#### 7. Algoritmos importantes (caso existam)

Este ponto tem como objectivo, apresentar algoritmos importantes para a realização da aplicação. A descrição deverá vir acompanhada por um fluxograma representando a solução adoptar, assim como por uma descrição que explique de forma completa as opções tomadas e as vantagens desta solução

#### 8. Mecanismos

Este ponto tem como objectivo apresentar mecanismos que tenham de ser criados/usados para garantir o bom funcionamento da aplicação (Exemplo: não repúdio de facturas, ...)

#### 9. Testes de aceitação

Este ponto tem como objectivo apresentar quais os testes que garantem que a aplicação está conforme o especificado

#### 10. Tecnologias a utilizar

Este ponto tem como objectivo apresentar quais as tecnologias a usar na implementação da aplicação, deverão ser incluídas as versões de cada tecnologia usada

#### 11. Planeamento da fase seguinte

Este ponto tem como objectivo apresentar quais os passos seguintes (*timings*, durações, afectações, contacto do responsável, ...)

## 12. Riscos

Este ponto tem como objectivo, apresentar quais são os riscos previstos. Os riscos apresentar deverão servir para alertar sobre problemas na implementação, na volatilidade de algum requisito (exemplo: IVA) e outros riscos que sejam significativos. Este ponto será uma evolução do documento de especificação funcional, pois poderá conter algumas preocupações associadas à tecnologia a utilizar

## 13. Anexos

Este ponto servirá como repositório de documentos, tabelas, mails, ... e outros elementos que complementem o documento em questão

- **Relatório de desenvolvimento**

Este documento tem como objectivo um detalhe minucioso da implementação. Sendo assim este documento deverá navegar por todo o código, afim de se compreender o que se implementou e assim se reaproveitar.

O documento deverá ter os seguintes pontos mínimos:

### 1. Introdução

Este ponto tem como objectivo introduzir o documento e colocar algumas considerações acerca das opções tomadas (criação de pacotes, uso de livrarias, ...)

### 2. Código Gerado

Este ponto tem como objectivo detalhar cada função desenvolvida para esta aplicação ou usada pela aplicação. Cada função deve conter uma descrição do que realiza e os dados de entrada e de retorno associados. Em casos mais complexos poder-se-ão usar fluxogramas ou listagens do código associado.

Além das funções estarem divididas por pacote, deverão estar também separadas em dois blocos, que são PL/SQL e ASP. Este ponto será um especialização do que é descrito no ponto PL/SQL deste documento, porque aqui apenas são descritos os dados relativos a esta aplicação

### 3. Conclusões

Este ponto tem como objectivo descrever situações que tenham decorrido e que mereçam algum comentário suplementar (do ponto de vista do código)

### 4. Anexo

Este ponto servirá para colocação de listagens, tabelas e outros elementos que complementem o documento em questão

- **Manual do utilizador**

Corresponde à entrega de um manual de Utilização, que permita ao utilizador perceber como interagir com a aplicação. Sendo assim, deverá ter os seguintes pontos:

## **1. Introdução**

Neste ponto dever-se-á encontrar uma breve explicação de como utilizar o manual. Assim como, a forma de chegar à aplicação definida

## **2. Funcionalidades**

As funcionalidades devem ser agrupadas, consoante os dois grandes blocos a que podem pertencer. Esses blocos são o *FrontOffice* e o *BackOffice*. A descrição de cada funcionalidade, deverá vir acompanhada por imagens exemplificativas, de forma a tornar o manual mais fácil de ler e compreender

## **3. Índice Remissivo**

Com este índice pretende-se facilitar a navegação no documento, de forma a ser mais fácil localizar o que se pretende realizar.

- **Manual de instalação**

Este documento deve conter todos os passos que são necessários para instalar a aplicação a partir do pacote de instalação.

## ANEXO M: Desenho de processos

O objectivo deste ponto é garantir que o desenho é sempre feito da mesma forma, tendo em conta o uso do que já existe e a necessidade de tornar flexível o desenho. Sendo assim há que ter em conta os seguintes pontos:

### • Sintaxe

A uniformização da sintaxe a utilizar, vai fazer com que a forma do desenho seja sempre a mesma, e sendo assim, os vários intervenientes terão a possibilidade de completar e melhorar o desenho sem que se estejam sempre a perder tempo em correcções de sintaxe. Sendo assim, temos:



Este símbolo deverá ser usado, quando queremos representar a existência de uma função sem intervenção do utilizador



Este símbolo deverá ser usado, quando quisermos representar algo a realizar na base de dados sem intervenção do utilizador. Sendo assim, corresponde a uma especialização do anterior



Este símbolo deverá ser usado, quando queremos representar a intervenção de um utilizador



Este símbolo deverá ser usado, quando queremos representar uma notificação sem resposta. Sendo assim, corresponde a uma especialização do anterior



Este símbolo deverá ser usado, quando queremos representar um sub-processo. Por vezes, pode-se optar por utilizar outro símbolo, mas sem abdicar da parte superior direita desta imagem. Esta opção deve ser tomada só se, se verificar algum acréscimo de compreensão



Este símbolo deverá ser usado, quando queremos representar uma decisão relativa a perfis, ou seja, quando é necessário avaliar destinatários



Este símbolo deverá ser usado, quando queremos representar que existe uma necessidade de paragem do processo



Este símbolo deverá ser usado, quando queremos representar um ciclo com contagem



Este símbolo deverá ser usado, quando queremos representar um sub-processo vazio, ou seja, uma sub-processo que não faz nada mas que servirá de suporte a uma possível alteração futura



Este símbolo deverá ser usado, quando queremos representar comparação de dois valores



Este símbolo deverá ser usado, quando queremos representar uma decisão por algum tipo de votação



Este símbolo deverá ser usado, quando queremos representar a recepção/lançamento de um evento



Nó inicial de um processo



Nó terminal de um processo



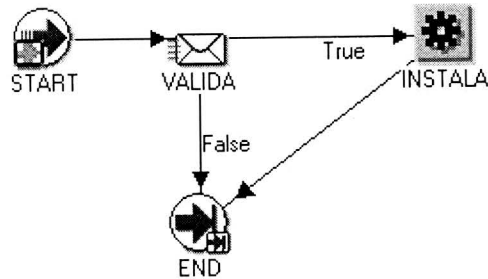
Ponto de junção de fluxo

Pode-se usar qualquer outro símbolo desde que se prove o ganho em compreensão.

## • Diagramas

Aquando do desenho é necessário ter em conta que o processo deverá ser dividido em vários sub-processos, de forma a garantir melhor compreensão e facilitar/isolar o impacto das alterações futuras no processo.

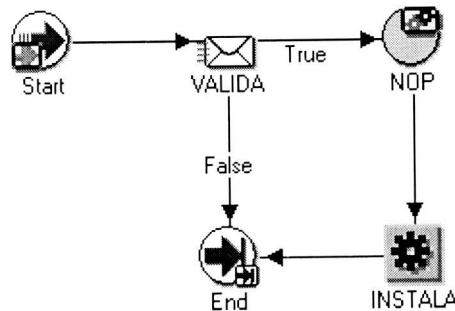
Além disso, o processo deverá ter sub-processos de *NOP* (sub-processos vazios), quando se achar que no futuro se terá necessidade de proceder a alterações. Por exemplo: Para fazer um pedido de um telefone o processo é este:



**Figura M- 1 : Desenho de processo (instalação de telefone)**

Faz-se a encomenda, de seguida alguém valida e só depois será feita a instalação.

Mas, se se prever que no futuro teremos que, antes da instalação escolher a empresa, podemos desde já preparar o Workflow para essa eventualidade. Essa preparação é feita da seguinte forma:



**Figura M- 2 : Desenho de processo (solução modular)**

Sendo assim, temos possibilidade de no futuro colocar no processo inicial as novas acções

## ANEXO N: Interface base para projectos *Workflow*

Existe uma necessidade de normalizar a interface do workflow com o utilizador. Esta normalização tem como objectivo, permitir a integração entre os vários *workflows*, ou seja, no futuro cada utilizador terá um só ponto de acesso a todos os *workflows* existentes. Podendo ou não especificar a sua linha de trabalho.

Queremos ainda permitir uma melhor integração entre a parte aplicacional dos projectos e a sua vertente de *workflow*. Sendo assim, a proposta é a seguinte

### • Cabeçalho



Figura N- 1 : Cabeçalho genérico do workflow

Nesta interface as propostas são:

- Usar um visual diferente para a caixa de combinação do menu, tornando-a menos agressiva (conforme a figura)
- No local onde está “Comunicação de artigos”, será um menu *pop-up*, que permitirá a passagem entre os vários *workflow*. O que significa que só aparecerá a worklist correspondente ao *workflow* seleccionado, assim como o menu aplicacional correspondente.
- No caso de o utilizador escolher todos os workflows, ou seja, ver a sua worklist global o menu deverá desaparecer
- No local onde está “Worklist” deverá aparecer o contexto em que o utilizador está inserido
- A linha laranja no fundo será onde aparecerá os destaques do dia (em movimento). Esses destaques serão em função do workflow seleccionado
- No local onde está “João”, será para aparecer o nome do utilizador

Feitas estas alterações, já será possível englobar todos os processos como se fosse um único processo.

### • Worklist

SEL.	PRI.	STATUS	OBSERVAÇÕES	ENVIÓ	LIMITE
<input type="checkbox"/>	Normal	Passar a histórico o artigo 12345+010101+xpto - Não validado pelo motivo falta de espaço		2002-05-22 11:00:50	2002-05-22 11:00:50
<input type="checkbox"/>	Normal	Passar a histórico o artigo 12345+010101+xpto - Não validado pelo motivo falta de espaço		2002-05-22 11:00:50	2002-05-22 11:00:50
<input type="checkbox"/>	Normal	Passar a histórico o artigo 12345+010101+xpto - Não validado pelo motivo falta de espaço		2002-05-22 11:00:50	2002-05-22 11:00:50
<input type="checkbox"/>	Normal	Passar a histórico o artigo 12345+010101+xpto - Não validado pelo motivo falta de espaço		2002-05-22 11:00:50	2002-05-22 11:00:50

Figura N- 2 : Worklist genérica para o workflow

Para a *worklist* seriam necessárias as seguintes áreas:

- **Coluna SEL.**

Esta coluna será onde o utilizador poderá seleccionar as notificações a tratar ou a delegar

- **Coluna PRI.**

Esta coluna será onde o utilizador visualizará a prioridade da notificação. Além disso, o cabeçalho terá um menu *pop-up* que permitirá filtrar a *worklist* por prioridades. O *pop-up* a aparecer deverá ter as seguintes opções: Todas, Alta, Normal, Baixa, Ordenar crescente e decrescente

- **Coluna STATUS**

Esta coluna será onde aparecerá o texto correspondente à notificação. A exemplo da coluna PRI, aparecerá um pop-up com um lista de opções, correspondentes aos tipos de notificações (Por exemplo, no caso da figura: apareceria Todos e Histórico, caso existissem mais a lista seria mais extensa

- **Coluna OBSERVAÇÕES**

Esta coluna será um local onde se pode colocar alguma informação aplicacional que se achar relevante

- **Coluna ENVIO**

Esta coluna será o local onde fica a data de envio da notificação. Caso se clique no cabeçalho deverá ser alterada a ordenação actual, ou seja, passar de crescente para decrescente e vice-versa

- **Coluna LIMITE**

Esta coluna será o local onde fica a data limite para tratar a notificação. Caso se clique no cabeçalho deverá ser alterada a ordenação actual, ou seja, passar de crescente para decrescente e vice-versa

- **Paginação**

Todas as listagens devem ser paginadas de acordo com o que já existe (figura anterior)

- **Botão GO SEL**

Este botão servirá para tratar notificações em bloco, ou seja, tratar as notificações seleccionadas. O aparecimento de um nova janela de tratamento, estará dependente de: as notificações seleccionadas serem compatíveis e existir um facilitador previsto

- **Botão DELEGAR**

Este botão permite delegar as notificações seleccionadas

- **Botão PESQUISAR NOTIFICAÇÕES**

Este botão permitirá abrir um formulário de pesquisa de notificações, numa nova frame (como representa a figura)

The screenshot displays the 'insote' application interface. At the top, there is a header with the logo 'insote' and the title 'Comunicação de artigos'. Below this, a navigation bar includes buttons for 'FECHAR', 'ATRÁS', and 'WORKLIST', along with the user name 'João'. The main content area is divided into two sections. The upper section is a table with columns for 'SEL.', 'PRI.', 'STATUS', 'OBSERVAÇÕES', 'ENVIÓ', and 'LÍMITE'. It contains three rows of data, each with a checkbox, a 'Normal' priority, and a status message: 'Passar a histórico o artigo 12345+010101+xppto - Não validado pelo motivo falta de espaço'. The 'ENVIÓ' and 'LÍMITE' columns show dates and times (2002-05-22 and 11:00:50). The lower section is a search form titled 'Pesquisar Notificações'. It includes several input fields and dropdown menus: 'STATUS' (set to 'Aberta'), 'DATAS' (with radio buttons for 'DATA DE ENVIÓ' and 'DATA LÍMITE'), 'PROCESSO' (set to 'Todos'), 'TIPO' (set to 'Todas'), 'PRIORIDADE' (set to 'Todas'), 'OBSERVAÇÕES', 'ASSUNTO', and 'NOTIFICAÇÕES DELEGADAS'. A 'LISTAR' button is located at the bottom right of the search form.

SEL.	PRI.	STATUS	OBSERVAÇÕES	ENVIÓ	LÍMITE
<input type="checkbox"/>	Normal	Passar a histórico o artigo 12345+010101+xppto - Não validado pelo motivo falta de espaço		2002-05-22 11:00:50	2002-05-22 11:00:50
<input type="checkbox"/>	Normal	Passar a histórico o artigo 12345+010101+xppto - Não validado pelo motivo falta de espaço		2002-05-22 11:00:50	2002-05-22 11:00:50
<input type="checkbox"/>	Normal	Passar a histórico o artigo 12345+010101+xppto - Não validado pelo motivo falta de espaço		2002-05-22 11:00:50	2002-05-22 11:00:50

**\* Pesquisar Notificações**

STATUS: Aberta

DATAS:  DATA DE ENVIÓ  DATA LÍMITE

PROCESSO: Todos TIPO: Todas

PRIORIDADE: Todas

OBSERVAÇÕES:

ASSUNTO:

NOTIFICAÇÕES DELEGADAS:  LISTAR

Figura N- 3 : Interface entre worklist e pesquisa de notificações

A janela de pesquisa de notificações, deverá ter os seguintes campos:

- **Status**

Permitirá seleccionar o status da notificação (Aberta, Fechada ou Todas). Por defeito deverá aparecer “Aberta”

- **Data**

Permitirá colocar datas para restringir a nível temporal as notificações

- **Processo**

Permitirá escolher em qual dos processos procurar. Por defeito deverá aparecer “Todos”

- **Tipo**

Permite seleccionar as notificações por tipo. Tal como é feito no *pop-up* de STATUS. Por defeito deverá aparecer “Todas”

- **Prioridade**

Permite seleccionar as notificações pela sua prioridade . Tal como é feito no *pop-up* de PRI. Por defeito deverá aparecer “Todas”

- **Observações**

Permitirá colocar uma restrição para a coluna de observações

- **Assunto**

Permitirá colocar uma restrição para a coluna de STATUS

- **Notificações Delegadas**

Permitirá colocar quem delegou as notificações ao utilizador actual

The image shows a search interface titled "Pesquisar Notificações". It contains several input fields and buttons:

- STATUS:** A dropdown menu with "Aberta" selected.
- DATAS:** Two date input fields. The first is labeled "DATA DE ENVIO" and the second "DATA LIMITE". A date format "aaaa-mm-dd" is shown to the right of the second field.
- PROCESSO:** A dropdown menu with "Todos" selected.
- TIPO:** A dropdown menu with "Todas" selected.
- PRIORIDADE:** A dropdown menu with "Todas" selected.
- OBSERVAÇÕES:** A text input field.
- ASSUNTO:** A text input field.
- NOTIFICAÇÕES DELEGADAS:** A text input field.
- Buttons:** "LISTA", "CANCELAR", and "PESQUISAR".

**Figura N- 4 : Pesquisa notificações genérica**

- **Tratamento das notificações**

O corpo de tratamento de uma notificação deverá estar dividido conforme a figura seguinte.

O diagrama ilustra o corpo de uma notificação genérica, dividido em três seções principais:

- Delegado:** Contém os campos "POR" e "COMENTÁRIO".
- Dados Aplicacionais:** Contém os campos "DE", "PARA" e "OBSERVAÇÕES".
- Dados a Preencher:** Contém dois campos rotulados "XPTO" com menus suspensos.

Na base da notificação, há uma barra com os botões "LIMPAR" e "OPÇÕES" e o texto "(RESPOSTAS DESTA NOTIFICAÇÃO)".

Figura N- 5 : Corpo de uma notificação (genérico)

As divisões representadas na figura são:

- **Delegado**

Neste ponto, estarão as informações relativas a delegar a notificação. Sendo assim, são apresentados os campos “POR” e “COMENTÁRIO” que significam respectivamente, quem fez a delegação e o comentário associado a essa delegação

- **Dados Aplicacionais**

Neste ponto, estarão presentes os dados a consultar que permitem tratar a notificação, dados esses que dependem de caso para caso (são suportados pela parte aplicacional). Existem ainda três dados não aplicativos, que são o “DE”, “PARA” e “OBSERVAÇÕES” que significam respectivamente quem enviou a notificação, para quem e as observações associadas

- **Dados a Preencher**

Neste ponto, estarão presentes os dados a preencher para que a notificação seja preenchida. Nesta secção estarão apenas dados aplicativos. Além dos dados, estarão presentes os seguintes botões:

- “**Limpar**”, que limpa o conteúdo das caixas de inserção presentes nesta secção
- “**Gravar**”, que grava o que foi escrito actualmente, sem enviar
- “**Respostas desta Notificação**”, que irá ter os botões necessários a esta notificação. Os botões a apresentar, dependem do retorno da notificação (definido no *builder*)

Excepções a esta interface devem ser discutidas antes de serem executadas, pois poderão ser casos que se repitam no futuro, e sendo assim é importante parar para definir uma solução geral



No caso das saídas e suspensões, não acontece o mesmo porque os dados a introduzir são em menor quantidade. (ver imagens seguintes).

Figura O- 3 : Introdução de dados sobre o artigo (saída)

Na introdução dos dados foi criado um facilitador que melhorasse a introdução das datas. Como isto foi eliminada a confusão criada pela entrada de datas em formatos diferentes e/ou datas inválidas. (ver imagem seguinte)

Figura O- 4 : Introdução de dados sobre o artigo (suspensão)

Figura O- 5 : Facilitador para introdução de datas

As listagens foram feitas para que se possa ter a percepção do estado actual do processo. Sendo assim, foram colocadas as datas relevantes ao processo. Além disso, teremos na mesma listagem a possibilidade de abortar o processo e alterar os dados.

UN/C/S/C	Código	ARTIGO	Descrição	TIPO COMUNICAÇÃO	Publicação	Alteração	DATAS	Stock	Planograma
340101	2225551	TINTA ESPECIAL BRANCA	REF.1135110020	Entrada					✓ ✕
330401	2222222	LIVRO DE COLORIR PINTA	REF:98101 \$D7	Entrada	2002-08-23				✓

Figura O- 6 : Listagens de artigos

Finalmente de forma a facilitar a navegação nos dados existentes nas listagens, criaram-se ecrãs de pesquisa que permitam seleccionar os dados a apresentar. Além disso, foi disponibilizado um símbolo para escrita de múltiplos valores na pesquisa.

**Pesquisa Avançada**

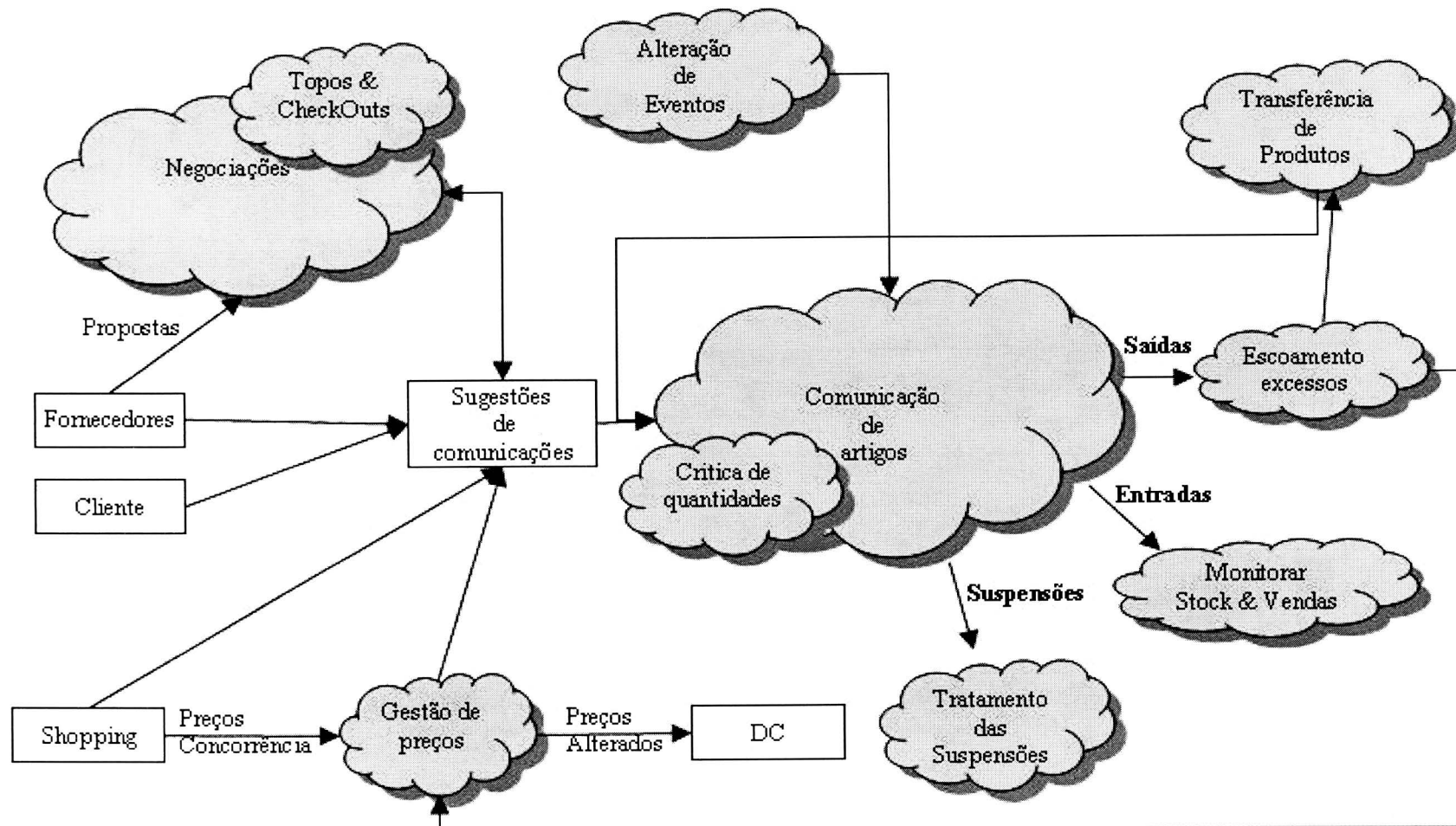
Estado	<input type="text" value="Activas"/>	Tipo de Comunicação	<input type="text" value="---"/>
Unidade de Negócio	<input type="text"/>	Data de Publicação	<input type="text"/>
Categoria	<input type="text"/>	Data de Alteração	<input type="text"/>
Sub Categoria	<input type="text"/>	Data Planograma	<input type="text"/>
Código do artigo	<input type="text"/>	Data Stock	<input type="text"/>
Descrição	<input type="text"/>	Tipo Acção	<input type="text" value="---"/>
Lojas	<input type="text"/>	Entrega Domicilio 1	<input type="text" value="---"/>
Cor	<input type="text"/>	Entrega Domicilio 2	<input type="text" value="---"/>
Fornecedor	<input type="text"/>	Sem Entrega Domicilio	<input type="text" value="---"/>
Gama Tipo	<input type="text"/>		

\* Para fazer pesquisas por estes atributos separar os valores por , ou ;

**Figura O- 7 : Pesquisa dentro das listagens**

Todos os ecrãs apresentados, pertencem a um tipo de utilizador. Todos os outros ecrãs são subconjuntos de este, ou seja, apenas são retirados alguns campos ou colocados de forma a não serem modificados

**ANEXO P: Análise macro de entrada e saída de artigos**



**Figura P- 1 : Entrada e Saída de artigos**





FACULDADE DE ENGENHARIA

UNIVERSIDADE DO PORTO

BIBLIOTECA



000066323