

Faculdade de Engenharia da Universidade do Porto



iPortalDoc - Motor de Workflow

Sérgio André Oliveira de Lima Moreira

VERSÃO 1.1

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Automação

Orientador: Prof. Doutor Gabriel David
Co-orientadora: Eng.^a Telma Salgueiro

Junho 2017

© Sérgio André Oliveira de Lima Moreira, 2017

Resumo

Este artigo de dissertação descreve o trabalho realizado, em ambiente empresarial, na IPBrick SA com base no tema - iPortalDoc: Motor de Workflow.

O *iPortalDoc* permite o registo, classificação e tratamento de todas as comunicações (emails, conversações de *Instant Messaging*, chamadas telefónicas, faxes), bem como o tratamento processual de todos os processos de uma organização

Apesar de eficaz na modulação e tratamento de processos documentais em organizações e empresas, o iPortalDoc carecia de uma solução gráfica que o tornasse mais inteligível e fácil de customizar. Aproveitando o pedido de um dos seus clientes, para exportar os workflows do iPortalDoc para uma linguagem Business Process Management, mais intelegível para a sua análise e compreensão, a IPBrick aproveitou para agarrar a oportunidade de evoluir o seu Sistema de Gestão Documental e Workflow.

Nesta aspiração de melhoria funcional de um produto, o maior desafio foi mesmo o de automatizar a disposição de um grafo que tem de obedecer a uma lógica própria, garantindo a máxima inteligibilidade possível e precavendo erros do cliente.

Palavras-chave: BPMN, Grafo, Workflow, XML

Abstract

This dissertation describes the work accomplished, in business environment, at IPBrick SA based on the theme - *iPortalDoc: Workflow Engine*.

IPortalDoc allows the registration, classification and treatment of all communications (emails, Instant Messaging conversations, telephone calls, faxes) as well as the procedural handling of all the processes of an organization.

Although effective in modulating and processing documentary processes in organizations and companies, iPortalDoc lacked a graphical solution that made it more intelligible and easy to customize. Taking advantage of one of its clients' requests to export IPortalDoc workflows to a Business Process Management language, more intelligible to its analysis and understanding, IPBrick took the opportunity to evolve its Document Management and Workflow System. In this aspiration of functional improvement of a product, the biggest challenge was even to automate the layout of a graph that has to obey its own logic, guaranteeing the maximum possible intelligibility and guarding the client's errors.

Keywords: BPMN, Graph, Workflow, XML

Agradecimentos

À minha Mãe que me ensinou a Ser,
Ao meu Pai que me ensinou a fazer.

“The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency.”

Bill Gates

Índice

Resumo.....	iv
Abstract.....	vii
Agradecimentos <opcional>	viii
Índice.....	xii
Lista de figuras	xiviii
Lista de tabelas.....	xv
Abreviaturas e Símbolos.....	xv
Capítulo 1.....	1
1.1 Introdução.....	1
1.1.1 Grupo IPBRICK SA.....	2
1.1.2 SGDW iPortalDoc.....	2
1.2 Objetivos.....	3
Capítulo 2.....	5
2.1 - Estado da arte	5
2.1.1 - Business Process Management	5
2.1.2 - Business Process Modeling Notation.....	6
2.1.3 -WORKFLOW.....	7
2.1.4 - Tecnologias.....	9
2.1.4.1 - LINUX.....	9
2.1.4.2 - PostgreSQL.....	10
2.1.4.3 - PHP	11
2.1.4.4 - XML.....	11
2.1.4.5 - XSLT.....	12
2.1.4.6 JavaScript.....	12
Capítulo 3.....	13
Problema e Requisitos.....	13
3.1 - Motor de Workflow do SGDW iPortalDoc.....	13
3.2 - Arquitetura da Base de Dados do iPortalDoc.....	15
3.3 - Levantamento de Requisitos.....	17
3.4 - Notação escolhida e Output Pretendido.....	17
Capítulo 4	19
Proposta de Solução e Arquitetura do Sistema.....	19
4.1 Relação entre a sintaxe do BPMN e do iPortalDoc.....	19
4.2 Grafo BPMN.....	22
4.3 Arquitetura do sistema.....	24
Capítulo 5	25
Implementação.....	25
5.1 Validação dos resultados.....	25

5.2 Integração no iPortalDoc.....	27
Capítulo 6	29
Conclusão	29
6.1 Estágio na IPBRICK SA	29
6.2 Trabalho futuro	29
Referências.....	31

Lista de figuras

Figura 1 - Exemplo de workflow no iPortalDoc	3
Figura 2 - Características da solução	4
Figura 3 - Etapas da metodologia BPM	5
Figura 4 - Representação do Workflow “Férias” no iPortalDoc	14
Figura 5 - BPMN do Workflow “Férias”	14
Figura 6 - Setor da BD responsável pela informação dos workflows	15
Figura 7 - Parte do UML geral com secção “workflowtemp”	16
Figura 8 - XML Schema que traduz a linguagem BPMN	18
Figura 9 - Estado BPMN	20
Figura 10 - Exclusive Gateway BPMN	20
Figura 11 - Parallel Gateway BPMN	20
Figura 12 - End State BPMN	20
Figura 13 - Initial State BPMN	20
Figura 14 - Exemplo de um workflow do iPortalDoc com uma situação de “truncamento”	21
Figura 15 - Exemplo de um workflow BPMN gerado a partir do iPortalDoc com uma situação de “truncamento”	22
Figura 16 - Esquema de entrada e saída de transições no digrama BPMN	22
Figura 17 - Arquitetura do Sistema	24
Figura 18 - Algoritmo de pesquisa de estados inválidos	26
Figura 19 - Integração visual da Exportação BPMN no iPortalDoc	27

Lista de tabelas

Tabela 1 - Lista de requisitos.....	17
Tabela 2 - Validação da lista de requisitos	25

Abreviaturas e Símbolos

AJAX: Asynchronous JavaScript and XML
BPM – Business Process Manager
BPPI - Iniciativa de Gestão de Processos de Negócios
BPMN – Business Process Modeling Notation
I&D – Investigação e Desenvolvimento
OMG - Object Management Group
PDI – Preparação da Dissertação
SGBD – Sistema de Gestão de Base de Dados
SGDW – Sistema de Gestão Documental e Workflow
SO – Sistema Operativo
UC – Unidade Curricular
XML - eXtensible Markup Language
XPath - eXtensible Markup Language Path Language
XSLT - eXtensible Style Language Transformations
WfMS - Workflow Management System

Capítulo 1

1.1 Introdução

Ao sucesso traduzido no crescimento de uma organização está associado o problema da gestão da sua informação. Uma gestão descuidada significará a perda ou atraso no acesso a essa informação e como se diz: “Informação é Poder”. Já uma gestão cuidada, mas ineficiente, sobrecarregará os colaboradores com tarefas que não acrescentam valor à atividade da organização. Foi com o propósito de oferecer resposta a este problema que surgiu o *iPortalMais*, hoje, *iPortalDoc* e que permite o registo, classificação e tratamento de todas as comunicações (emails, conversações de *Instant Messaging*, chamadas telefônicas, faxes), bem como, o tratamento processual de todos os processos da mesma organização.

Mas, como diria *Lavoisier* - "*Na Natureza nada se cria, nada se perde, tudo se transforma*" - e o *trade-off* de se propor a sistematizar uma solução deste género cria um novo desafio: o de diminuir o impacto da implementação desta tecnologia. É nesse contexto, que foi elaborada a presente dissertação que partiu de um trabalho em ambiente empresarial realizado na IPBRICK SA e surgiu da necessidade concreta, de um dos seus clientes, em assegurar a exportação dos *workflows* do sistema *IPortalDoc* para uma linguagem de *BPM (Business Process Modeling)*.

Foi a perceção de que a satisfação desta necessidade resolveria, não só, o problema deste cliente, mas que tornaria a proposta solução de todo este produto mais consistente, possibilitando ao cliente descrever os seus processos, numa plataforma que usasse uma linguagem mais inteligível, atenuando o impacto de que falávamos, que me motivou a aceitar o desafio de refletir sobre esta temática.

Sendo a minha especialização na área da Gestão Industrial, vejo numa empresa tecnológica apostada em fornecer soluções corporativas para empresas, ao nível da gestão

documental e de processos, uma excelente oportunidade para aliar os conhecimentos organizacionais e orientados ao cliente da minha especialização a um processo de sistematização informático, indispensável no Mundo atual.

1.1.1 Grupo IPBRICK SA

A **IPBRICK SA** [1] é uma empresa fabricante e distribuidora especializada em Soluções de Comunicação Corporativas e tem, como dois grandes produtos, o *IPortalDoc* e a *IPBrick* (Sistema Operativo de Rede).

Esta teve a sua génese em Maio de 2000, numa época em que o sistema operativo Linux estava no foco das atenções, por parte de vários especialistas e organizações. A empresa surge como resposta a várias áreas de intervenção assentes em soluções Open Source. Ou seja, aplicações para a internet, projetos de redes estruturadas, serviços de Intranet e de interligação à internet. Tendo como principal objetivo proporcionar uma experiência de utilização das TI (Tecnologias de Informação) simplificada e acessível tanto para empresas como para organizações públicas.

No ano de 2001, com recurso a alguns capitais próprios, iniciou os primeiros passos no I&D e desenvolveu algumas soluções assentes em Linux para as comunicações empresariais. Surge o primeiro produto IPBrick (denominado de “iPortalDoc”) e em 2005, em concordância com a estabilização tecnológica dos seus produtos e das inúmeras solicitações de mercados externos, a empresa deu início ao seu projeto de internacionalização. É no sentido da melhoria e aperfeiçoamento deste produto às necessidades do mercado que a minha dissertação se debruçou.

1.1.2 SGDW iPortalDoc

O *iPortalDoc* [2] é um Sistema de Gestão Documental e *Workflow* (SGDW), que funciona como um serviço de valor acrescentado para Intranet, estando o mesmo integrado de forma nativa no Sistema Operativo *IPBrick*. O *iPortalDoc* utiliza, como Sistema de Gestão de Base de Dados (SGBD), o *PostgreSQL*. O *iPortalDoc* permite o registo, classificação e tratamento de todas as comunicações (emails, conversações de *Instant Messaging*, chamadas telefónicas, faxes), bem como o tratamento processual de todos os processos de uma organização.

Os sistemas de gestão documental e *workflow* caracterizam-se por serem softwares que permitem a organização e gestão de informação não estruturada (documentos) e

também a definição dos vários estados pelos quais um documento passa: publicação, aprovação, distribuição e circulação ou arquivo possibilitando o controlo dos fluxos de circulação de documentos. O motor de *workflow* utilizado é nativo do *iPortalDoc*. Este motor de *workflow* tem na sua génese o funcionamento das Máquinas de Estado, tendo posteriormente evoluído no sentido das *Redes de Petri*.

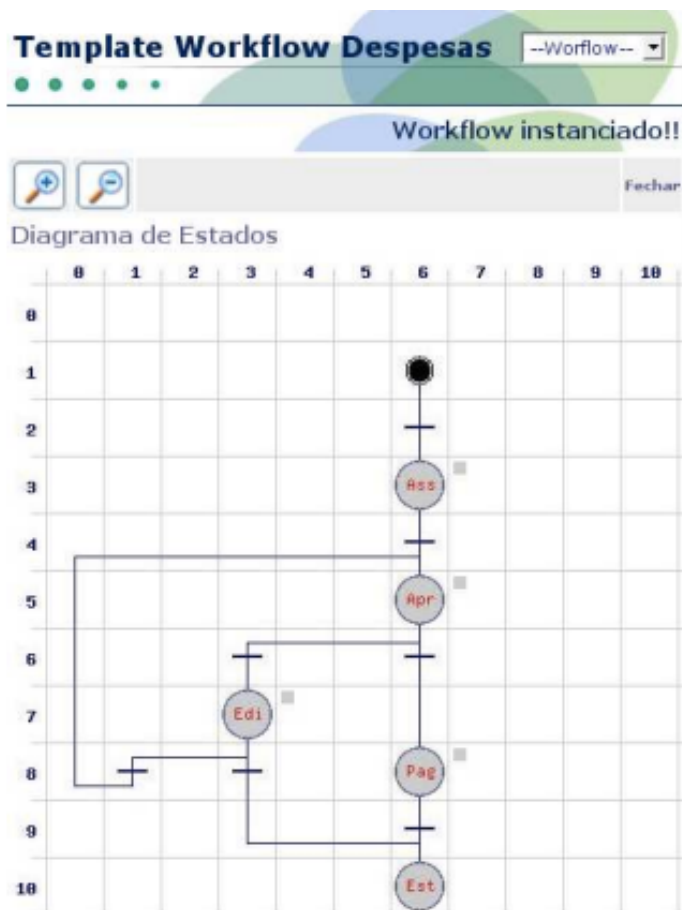


Figura 1. Exemplo de workflow no iPortalDoc

1.2 Objetivos

Na minha dissertação, tive em conta a realização de um objetivo principal: Disponibilizar uma interface no *iPortalDoc*, que permita a exportação de informação dos fluxos, para formato que possibilite a sua integração em aplicações de *Business Process Management (BPM)*, de modo a possibilitar aos clientes a análise dos processos implementados no seu sistema;

Esse objetivo vem atender a uma necessidade de quem utiliza motores de *workflow*, ou

semelhantes, para a gestão dos processos e documentos de uma empresa. A parte mais demorada e suscetível a erros de uma qualquer sistematização, informática ou conceptual, é a compreensão do problema. Propondo-se este sistema a vir traduzir os processos de uma empresa numa solução informática eficiente, não seria inteligente excluir os próprios funcionários da empresa deste processo. Estes, apesar de dotados de um conhecimento empírico e alargado sobre a sua empresa, muitas vezes, não conseguem transmitir esse mesmo conhecimento numa linguagem próxima das linguagens de programação utilizadas para conceber o programa e da mesma maneira, não conseguem avaliar a qualidade do mesmo até ele estar realmente pronto. Por outro lado, se é de sistemas eficientes que estamos a falar, estes dificilmente serão alcançados se para isso não houver uma solução que para além de conseguir traduzir as funcionalidades do *workflow* apresentado, também seja compreensível pelo funcionário da empresa. É neste propósito que se justifica a utilização de linguagem de BPM.

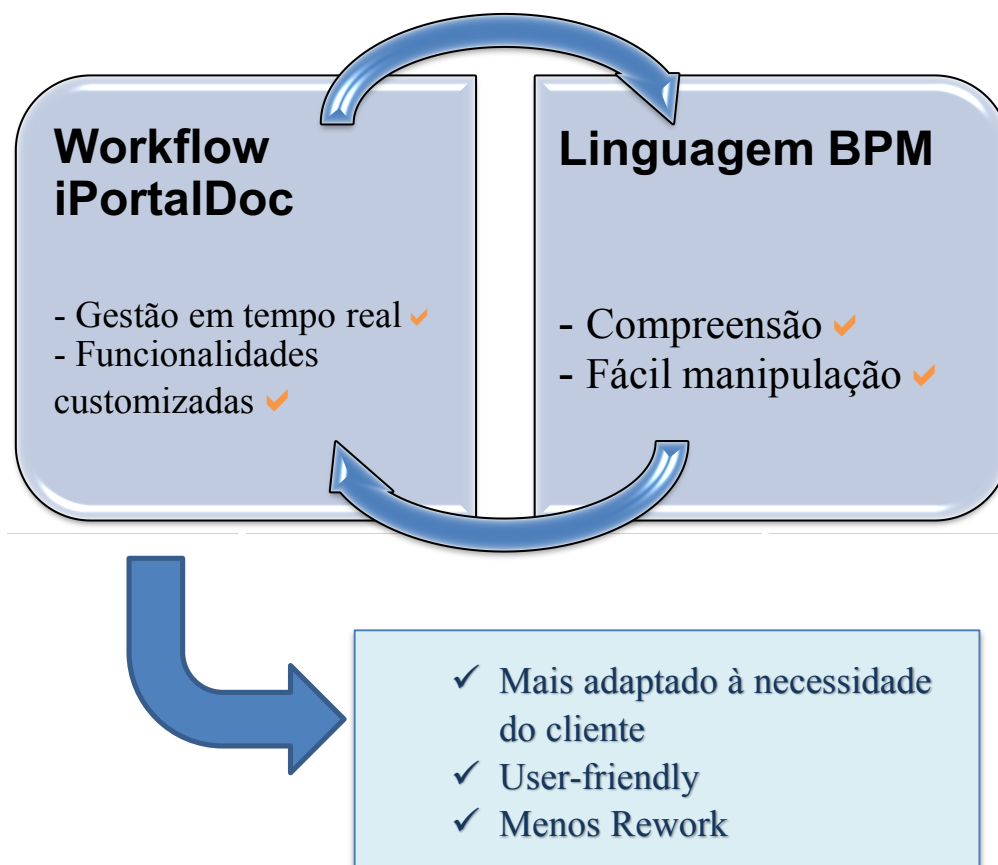


Figura 2. Características da solução

Capítulo 2

2.1 Estado da arte

Neste capítulo irá ser feita uma análise do estado da arte das várias tecnologias necessárias ao desenvolvimento do trabalho pretendido, para além de uma breve descrição de o que é o *Business Process Management* (BPM). De seguida dar-se-á foco a ferramenta que irá ser utilizada, *BPMN*, identificando algumas das suas principais características. Esta notação foi escolhida em detrimento de outras, como o UML, pela oferta de um visual mais compreensível para o usuário não especializado (*user friendly*) e por estarmos a fazer a importação de um fluxo empresarial, com um conjunto de características necessárias para o uso de *BPMN*.

2.1.1 Business Process Management

O *Business Process Management* [3] visa considerar a atual organização empresarial como um conjunto de atividades coordenadas e gerenciadas executados em uma ordem determinada para alcançar objetivos comuns definidos a nível da estratégia da empresa.

Um processo de negócio é um conjunto de atividades coordenadas executadas pelas unidades organizacionais da empresa. Estas atividades são ou tarefas manuais ou automatizados.

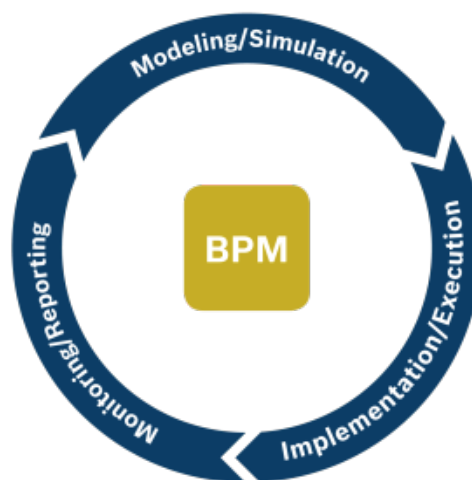


Figura3. Etapas da metodologia BPM

A iniciativa *BPM* requer pelo menos três etapas:

- A análise e desenho de processos de negócio, a fim de alcançar os objetivos estratégicos;
- A implementação e execução de processos de negócio;
- O acompanhamento dos processos de negócio e a definição de respectivas ações corretivas.

2.1.2 Business Process Modeling Notation

O que é BPMN?

O *Business Process Modeling Notation* (BPMN) [4] é uma representação gráfica usada para especificar os processos de negócios de um fluxo de trabalho. Este é usado como um padrão para modelagem de processos de negócios que fornece uma notação gráfica para especificar os processos de negócios em um *Business Process Diagram* (BPM), com base em uma técnica de fluxograma muito semelhante ao usado em diagramas de atividade *UML*. O objetivo do *BPMN* é apoiar a gestão de processos de negócios tanto para usuários técnicos e de negócios, enquanto ainda está sendo capaz de representar a semântica processo complexo. A especificação *BPMN* também fornece um mapeamento entre os gráficos da notação e as construções subjacentes das línguas de execução, particularmente *BPEL* (Business Process Execution Language).

História da BPMN

BPMN foi originalmente desenvolvido pela Iniciativa de Gestão de Processos de Negócios (BPMI), que desde então se fundiu com o *Object Management Group* (OMG). A versão mais recente do *BPMN* é *BPMN2.0*, agora apoiado por muitas ferramentas. No entanto, *BPMN1.2* era muito popular e a maioria das ferramentas ainda estão a nível *BPMN1.2*. [4]

Objetivos do BPMN

O principal objetivo do *BPMN* é fornecer uma notação padrão que é facilmente compreensível por todos os intervenientes do negócio. Essas partes interessadas incluem analistas de negócios que criam e aperfeiçoar processos, desenvolvedores técnicos responsáveis pela implementação de processos e gerentes de negócios que monitorar e gerenciar processos. Consequentemente, *BPMN* destina-se a servir como uma língua comum para preencher a lacuna de comunicação que ocorre com frequência entre o design e implementação de processos de negócios.

O *BPMN* suporta apenas os conceitos modulares que são aplicáveis aos processos de negócios. Isto significa que outros tipos de modelagem realizado por organizações para fins não-comerciais será fora do âmbito do *BPMN*. Por exemplo, a modelagem de estruturas organizacionais, avarias funcionais e *datamodels* não será uma parte de *BPMN*. [4]

2.1.3 WORKFLOW

Apesar da sua origem ser bastante antiga, os sistemas de workflow tiveram o início do seu auge de popularidade no decorrer do século passado quando a automatização dos processos fabris, que já os tinham incorporado, foi readaptada aos processos de negócio devido à necessidade de automatização das organizações.

Um workflow consiste numa sequência de passos que permite sistematizar a informação, documentação ou tarefas entre utilizadores, de acordo com uma série de regras, permitindo, assim, tornar estes processos mais simples e intuitivos aos utilizadores. A automatização de um processo identifica as diversas atividades, regras e dados usados para gerir o ciclo de vida de um workflow [5].

Um workflow é, portanto, constituído por um conjunto de:

1. Estados - diferentes elementos que definem a função de um workflow, bem como, o seu estado atual;
2. Ligações - caminhos entre elementos;
3. Regras - condições que permitem a transição entre estados;
4. Papéis e Atores - funções de pessoas ou programas que intervêm no desenrolar do workflow com tarefas pré-definidas e associadas a um dado estado;
5. Tarefas - ações que podem ser realizadas por utilizadores ou programas que, depois de completas, poderão definir o caminho do workflow (se a essa tarefa estiver associada a regra de transição para o estado seguinte).

A obtenção da informação traduzida, desta forma, numa linguagem de workflow, é conseguida através da articulação de vários sistemas que refletem as dinâmicas internas de uma organização recolhendo aqueles que possam ser os dados críticos para o seu funcionamento, por meio tecnológico (através de sensores, software, etc..) ou humano, convidando os diferentes atores a atualizarem essa informação no sistema. A utilização de bases de dados e de ferramentas que permitem colocar informação atualizada no sistema da forma mais eficiente e simples possível, para evitar a excessiva burocratização do sistema, são importantes aliados deste conceito.

De notar, que a utilização desta metodologia numa organização será, tanto ou mais eficaz, quanto a modulação dos seus workflows refletir, de facto, aqueles que são os pontos críticos da sua atividade permitindo que, através do controlo e auditoria dos seus processos, se possa embarcar num sistema de melhoria contínua que se irá refletir, antes de mais, na produtividade da empresa e na qualidade do desempenho no serviço prestado pela mesma.

Vantagens no uso de sistemas de Workflows

As vantagens no uso de sistemas de *workflows* são as seguintes: [5]

- eliminação do incomodo e do lixo dos produtos de papel;
- simplificação dos formulários previstos;
- acesso remoto;
- arquivamento e recuperação de informações simplificados;
- habilidade de rapidamente trilhar as informações submetidas;
- possibilidade de saber os responsáveis de cada tarefa do processo;
- aumento no tempo de linhas de informação;
- garantia da integridade dos processos;
- manipulação eletrónica de documentos;
- eliminação do tempo de espera entre atividades;
- integração das atividades da empresa.

Funcionalidades disponíveis no mercado

Segundo (Moro, 1998), os sistemas de *workflow* disponíveis no mercado possuem um conjunto relativamente comum de funcionalidades. As principais são:

- roteamento de trabalho (predefine a sequência em que as atividades serão executadas, podendo ser baseado em respostas e em regras);
- invocação automática de aplicativos;
- distribuição dinâmica de trabalho - determinar qual participante irá executar a tarefa;
- priorização de trabalho (a maioria dos sistemas de *workflow* permite que a prioridade de uma instância seja alterada, em geral por um usuário “administrador”);
- acompanhamento do trabalho (capacidade de acompanhar uma determinada instância de *workflow* e descobrir imediatamente seu status atual de processamento, sob a responsabilidade de quem está no momento, e quanto tempo ela está esperando na atividade atual);
- geração de dados estratégicos (através do armazenamento de certos atributos de cada instância de *workflow* executada, pode-se criar uma base de dados que reflete

a eficiência e a eficácia dos processos atualmente desempenhados pela organização).

As vantagens associadas ao uso da ferramenta de workflow mantiveram-se, num plano mais geral, a sua evolução verifica-se desde logo na eficácia e na facilidade com que a obtenção dos dados críticos é obtida, com a introdução de aplicações modernas e *user-friendly* e com uma maior capacidade para armazenar e tratar grandes quantidades de informação em tempo útil. Outra evolução, que já foi referida anteriormente, é o contexto da aplicação desta ferramenta que evolui para diferentes áreas de negócio, deixando de se limitar ao contexto fabril e estando cada vez mais presente no nosso dia-a-dia.

2.1.4 Tecnologias

Durante a minha dissertação recorrerei ao uso de diversas tecnologias para a realização da mesma. Aqui fica um resumo das principais tecnologias que serão utilizadas:

2.1.4.1 LINUX [6]

Linux é um sistema operativo: uma série de programas que permitem interagir com o computador e correr outros programas.

A parte mais importante de um sistema operativo é o *kernel*. Num sistema *GNU/Linux*, o componente do *kernel* é o *Linux*. O resto do sistema consiste noutros programas, muitos dos quais escritos por ou para o *GNU Project*. Como o *kernel* sozinho não forma um sistema operativo utilizável, costuma-se utilizar o termo “GNU/Linux” para se referir aos sistemas a que muitas pessoas vulgarmente chamam de “Linux”.

O Linux tem como modelo o sistema operativo *Unix*. Desde o início, Linux foi desenhado para ser um sistema multi-tarefa e multi-utilizador. Estes factos são suficientes para tornar Linux diferente de outros sistemas operativos bem conhecidos. Em contraste com outros sistemas operativos, ninguém é dono de Linux. Muito do seu desenvolvimento é feito por voluntários não pagos.

O desenvolvimento do que mais tarde se tornou *GNU/Linux* começou em 1984, quando a *Free Software Foundation* iniciou o desenvolvimento de um sistema operativo livre, ao estilo *Unix*, chamado *GNU*.

O *GNU Project* desenvolveu um extenso conjunto de ferramentas de software livre para utilizar com *Unix*TM e sistemas operativos do tipo *Unix* tais como o *Linux*. Estas ferramentas permitem aos utilizadores executar tarefas que vão desde o mundano (como copiar ou remover ficheiros do sistema) ao arcano (como escrever e compilar programas ou editar de forma sofisticada numa variedade de formatos de documentos).

Enquanto que muitos grupos e indivíduos contribuíram para *GNU/Linux*, o maior contribuidor individual continua a ser a *Free Software Foundation*, que criou não só a maioria das ferramentas utilizadas em *GNU/Linux*, mas também a filosofia e a comunidade que tornaram *GNU/Linux* possível.

Os utilizadores de *GNU/Linux* têm uma imensa liberdade de escolha no seu software. Por exemplo, utilizadores de *GNU/Linux* podem escolher de entre uma dúzia de *shells* de linha de comandos e vários ambientes gráficos. Esta seleção é muitas vezes confusa para os utilizadores de outros sistemas operativos, que não estão habituados a pensarem na linha de comandos ou no ambiente de trabalho em algo que possam substituir.

É também menos provável que o *Linux* bloqueie, que corra melhor mais de um programa ao mesmo tempo e que seja mais seguro que muitos sistemas operativos. Com estas vantagens, *Linux* é o sistema operativo que mais rapidamente cresce no mercado de servidores. Mais recentemente, *Linux* passou a ser popular entre os utilizadores domésticos e empresariais.

2.1.4.2 PostgreSQL

PostgreSQL [7] é um sistema de gestão de base de dados de código aberto que conta com mais de quinze anos de desenvolvimento, tornando-o numa arquitetura com resultados comprovados de grande fiabilidade garantindo a integridade e exatidão dos dados. Como se trata de um software de código aberto, pode ser usado, distribuído ou mesmo modificado de acordo com os desejos de cada utilizador tendo garantido o apoio por parte de uma grande comunidade de colaboradores. Neste momento é possível utilizar o *PostgreSQL* na maioria dos sistemas operativos. Este fornece diversos recursos como, por exemplo, consultas complexas, garantia da integridade dos dados, *triggers*, vistas, procedimentos armazenados, heranças singulares ou de múltiplas tabelas; também oferece suporte para objetos de grande dimensão como por exemplo imagens ou vídeos.

2.1.4.3 PHP

PHP [8], que significa "*PHP: Hypertext Preprocessor*", é uma linguagem de programação de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a web e pode ser mesclada dentro do código *HTML*. A sintaxe da linguagem lembra *C*, *Java* e *Perl*, e é fácil de aprender. O objetivo principal da linguagem é permitir a desenvolvedores escreverem páginas que serão geradas dinamicamente e rapidamente, mas você pode fazer muito mais do que isso com *PHP*.

2.1.4.4 XML

XML, *eXtensible Markup Language*, é considerado uma meta-markup language, uma linguagem de anotação descritiva extensível, oferecendo todas as características que tornam desejável este tipo de linguagens: independência relativamente às plataformas de software e de hardware utilizadas, longevidade, custos de manutenção reduzidos e fácil utilização. É uma linguagem que disponibiliza um formato para a descrição de dados estruturados proposta pelo *W3C*, *Worldwide Web Consortium*. O *XML* é uma metalinguagem que em qualquer momento permite acrescentar novos elementos à linguagem, o que torna esta linguagem, aberta e sem limitações do ponto de vista de extensibilidade. [10]

2.1.4.5 XSLT

Do ponto de vista de um programador, *XSLT* é uma linguagem de programação para processamento de dados *XML* - ou seja, transformar documentos *XML*. Como tal, ela suporta o seguinte:

Um pequeno conjunto de tipos flexíveis de dados: booleano, número, *strings*, conjunto de nós e objetos externos.

Um conjunto completo de operações: `<xsl: template>`, `<xsl: apply-templates>`, `<xsl: sort>`, `<xsl: output>`, e assim por diante.

Programação de controlo de fluxo: `<xsl: if>`, `<xsl: for-each>`, `<xsl: choose>`, e assim por diante.

XSLT adota a sintaxe da linguagem *XPath* para expressões. Há também *built-in* e extensão personalizada funções para expressões de construção para consultas de dados *XML* e transformações.

Uma folha de estilo *XSLT* é um documento *XML*. Isto significa que as instruções *XSLT* são expressas como elementos *XML* e uma folha de estilo *XSLT* pode transformar outras folhas de estilo *XSLT*, ou mesmo ela própria. [11]

2.1.4.6 JavaScript

JavaScript é uma linguagem de escrita leve e fácil de aprender. Esta é usada em todos os sites para responder às ações do usuário, validar formulários web, detetar o suporte ao navegador, e muito mais.

JavaScript é uma linguagem de programação web, ou seja, uma linguagem que permite ao designer do site controlar como uma página da web se comporta. Isso implica que o *JavaScript* seja crucialmente diferente de *HTML*, a linguagem que dá estrutura aos documentos web e *CSS*, o idioma que controla a aparência das páginas da web. [12]

Capítulo 3

Problema e Requisitos

No capítulo 2 foi apresentada uma abordagem aos conceitos teóricos presentes na elaboração desta dissertação, bem como das tecnologias e softwares utilizados. Neste capítulo, irão ser esmiuçados os problemas que o trabalho realizado se propôs a resolver, bem como, a lista de requisitos considerados na sua conceção. Para isso, será apresentado com mais detalhe o motor de *workflow* do *SGDW iPortalDoc*, a sua sintaxe gráfica e as limitações que lhe estão associadas.

3.1 Motor de Workflow do SGDW iPortalDoc

Como foi explicado, o Motor de *Workflow* do *iPortalDoc* modeliza os diferentes processos documentais de uma organização, automatizando-os e chamando os atores que terão de intervir nesses processos apenas na hora devida e orientando, sempre que possível, a intervenção dos mesmos.

Em baixo, podemos ver um exemplo da modelização de um processo básico e comum à generalidade das organizações: um pedido de aprovação de férias. Depois de despoletado o início deste processo, por parte de um colaborador, este é convidado a enviar uma proposta de um período de férias que pretender gozar, preenchendo, para o efeito, um formulário automático. O responsável direto, por esse colaborador, será convidado a tomar uma decisão: aprovar ou não aprovar o pedido de férias do colaborador inicial. Se aprovar, este será automaticamente notificado que o seu pedido de férias foi aceite e o sistema registará a ausência do colaborador no período indicado, concluindo depois o processo. Se não aprovar, o colaborador será convidado a escolher novamente o seu período de férias (contando com uma possível justificação que o seu responsável anexou à não aprovação deste pedido). Depois de corrigir a proposta de férias, a decisão estará novamente nas mãos do responsável.

As 2 figuras apresentadas, representam o mesmo workflow, uma representada no motor do iPortalDoc, a outra em linguagem BPMN ¹.

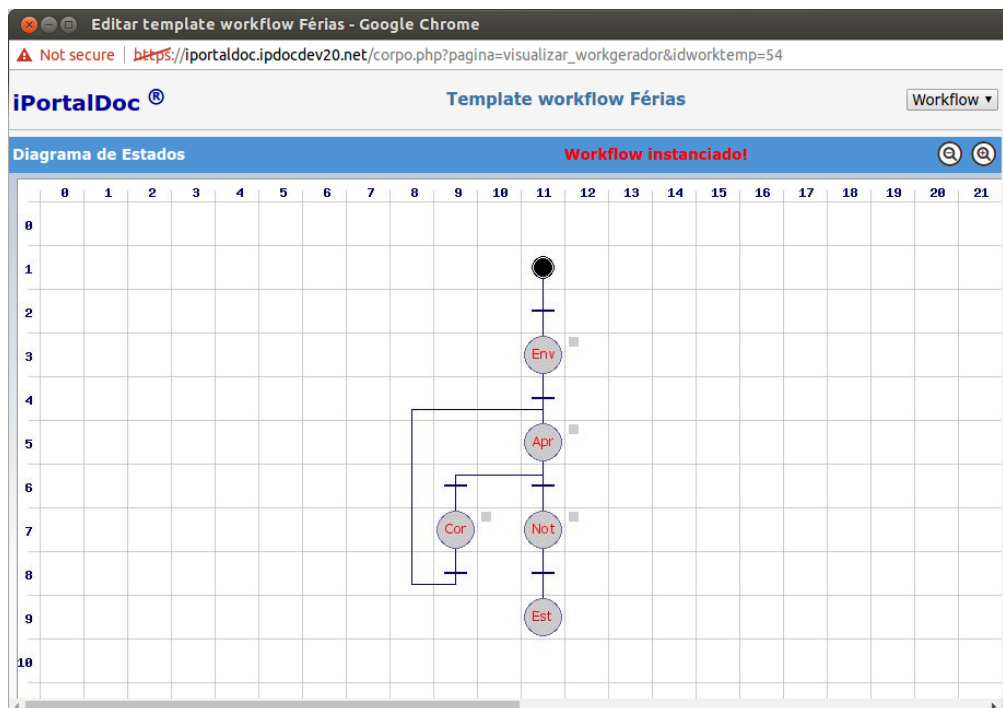


Figura 4. Representação do Workflow “Férias” no iPortalDoc

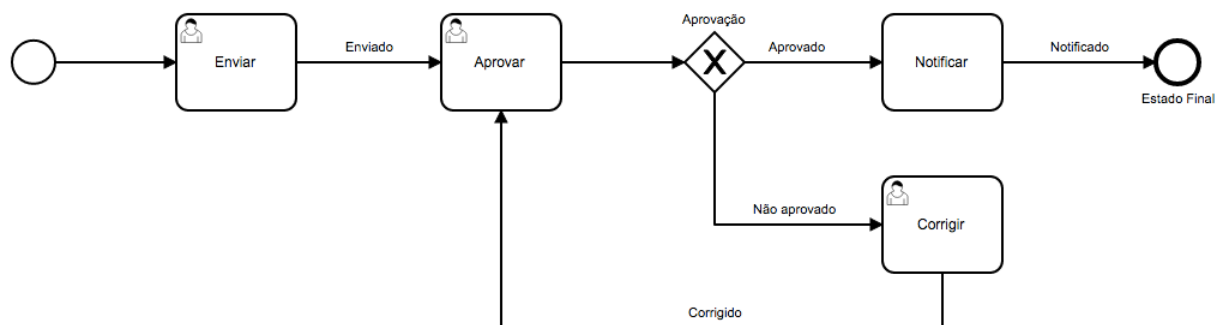


Figura5. BPMN do Workflow “Férias”

É perceptível a limitação do motor de *workflow* original em ser autoexplicativo, deixando ainda alguma margem para ambiguidades, como por exemplo, na sequência do estado “Enviar”, onde um olhar mais desatento poderia ser levado a acreditar que a saída desse estado se dividiria em duas distintas, quando na verdade o que acontece é uma transição simples de um estado para outro.

¹ Gerada através da ferramenta online bpmn.io.

A vantagem do BPMN não se esgota no aspeto visual, mas está presente, também, na manipulação do *workflow*. Um leitor desta linguagem permite arrastar blocos ou até destruir e criar outros, tal como, conexões entre eles com uma rapidez que não seria possível no motor de wf original. Pude comprovar tal facto mesmo com utilizadores mais experientes, como referirei mais à frente.

3.2 Arquitetura da Base de Dados do iPortalDoc

No setor da Base de Dados da IPBrick, responsável por armazenar a informação relativa aos *workflows* podemos encontrar 3 grupos distintos de informação: o “Workflowtemp”, o “Workflow” e o “Workflowrev”.

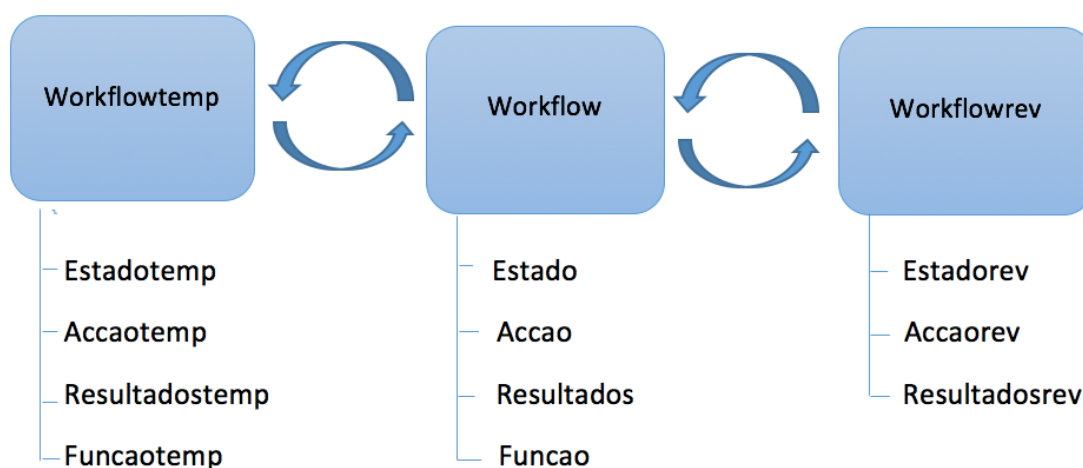


Figura6. Setor da BD responsável pela informação dos workflows

O “workflowtemp” contém informação relativa à estrutura do *workflow*, aquilo que podemos definir como o seu esqueleto. O “workflow” aproveita o esqueleto do “workflowtemp” e instancia essa estrutura numa entidade própria. O “workflowrev” é o *workflow* já instanciado depois de um conjunto de transições. De maneira, que um “workflowtemp” pode ter vários “workflows” e um “workflow” pode ter vários “workflowrevs” associados. Numa analogia simples, o “workflowtemp” seria o corpo de uma pessoa. O “workflow” seria, por exemplo, a pessoa X. De notar que tanto X, como Y podem ter associados uma mesma estrutura humana (workflowtemp). Nesta analogia, o “workflowrev” seria um dos vários X futuros, que dependeria das decisões que tomasse ou, no exemplo concreto, das condições de transição que satisfizesse, bem como da informação que lhe fosse sendo associada. Reaproveitando, o exemplo do pedido de férias mostrado anteriormente, de notar que a mesma estrutura “Férias”, pode ter várias

instancias associadas (ex. Férias RH, Férias LOG, ...). Depois de iniciada, uma destas instâncias, poderá dar origem a vários “revs”.

Acredito que depois desta explicação se conclui que a manipulação da informação será feita ao nível do “workflowtemp”, antes do *workflow* ser instanciado, pois vamos operar na estrutura do mesmo.

Ao “Workflowtemp” estão associados um “estadotemp”, uma “accaotemp” e um “resultadostemp”. Mais à frente será explicada, a relação que estes dados têm com a sintaxe da linguagem BPMN. Neste capítulo, limitar-me-ei a apresentar a constituição da base de dados do iPortalDoc, na qual me apoiei para realizar este trabalho. Na figura em baixo podemos ver a secção do UML da BD onde se inserem o Workflowtemp e as classes que lhes estão associadas:

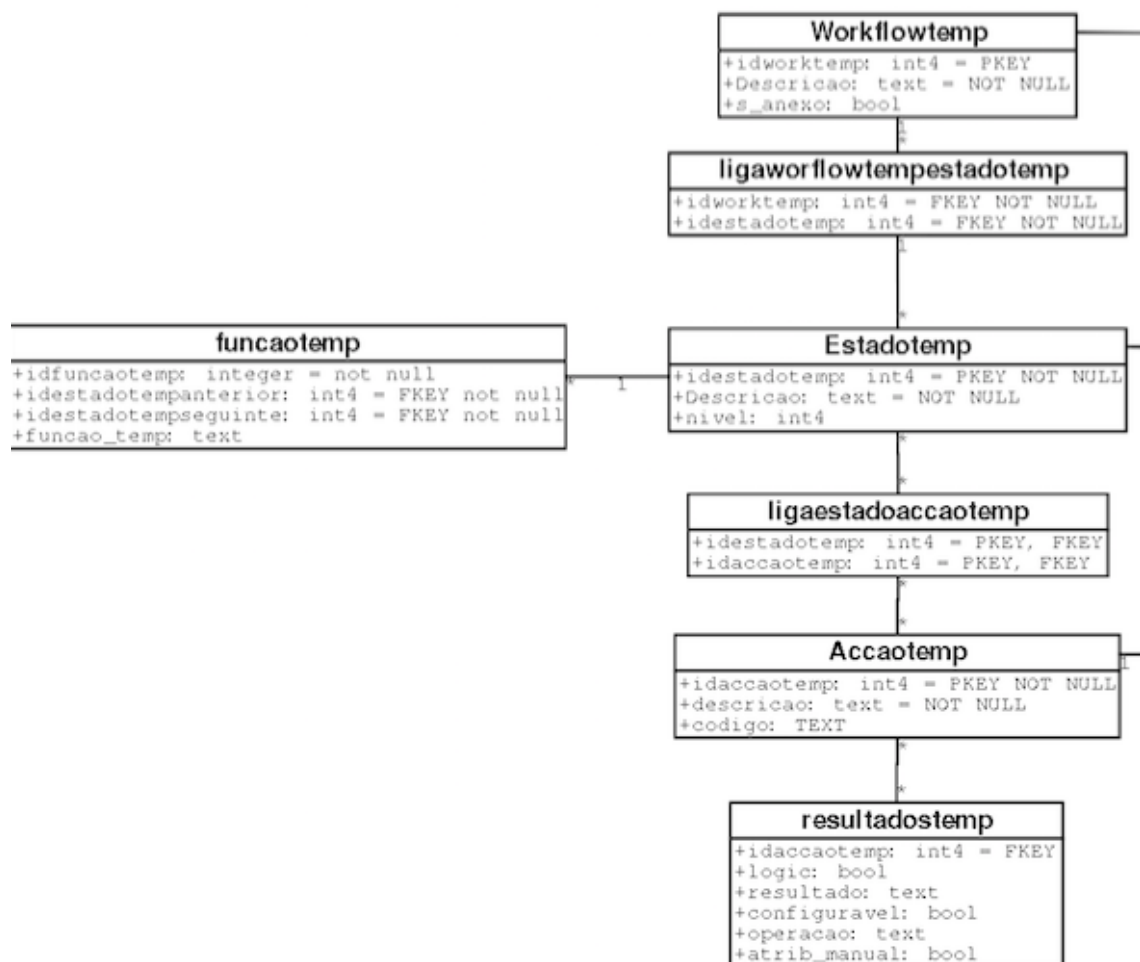


Figura7. Parte do UML geral com secção “workflowtemp”

3.3 Levantamento de Requisitos

Defini em colaboração com a IPBrick um conjunto de requisitos que servirão de ponto de partida para a elaboração deste trabalho. Com o conhecimento mais aprofundado do produto, bem como com os *inputs* que me iam chegando ao longo da realização do mesmo, fui tendo em conta outros que tentei, sempre que possível, satisfazer para customizar o produto de acordo com as expectativas da empresa e do cliente final.

Na Tabela seguinte indicarei os estabelecidos inicialmente pois foram os que marcaram estruturalmente a minha abordagem:

Requisito	
BPM01	Solução compatível com o SO Linux
BPM02	Exportar workflows recorrendo a uma das linguagens de BPM mais comuns (ex. BPMN, EPC, UML, IDEF, RAD, ...)
BPM03	Garantir um sistema user-friendly
BPM04	Se possível, utilizar as linguagens já utilizadas (PHP, XML, XSLT, JavaScript) e respeitar a organização de documentos actual.
BPM05	Prevenir erros do cliente
BPM06	Garantir um tempo de resposta rápido
BPM07	Fazer uma integração da funcionalidade coerente com o restante produto.

Tabela1. Lista de requisitos

3.4 Notação escolhida e Output Pretendido

Para resultado do ficheiro de exportação final, escolhi a notação BPMN. As razões da escolha foram a popularidade da notação, por um lado, e a inteligibilidade que oferece, a um cliente não especializado, em detrimento de outra notação igualmente popular como o UML.

Escolhida a notação percebi que o produto final teria que estar descrito num *XML Schema*, ficheiro XML com uma arquitetura própria que está representada na figura seguinte:

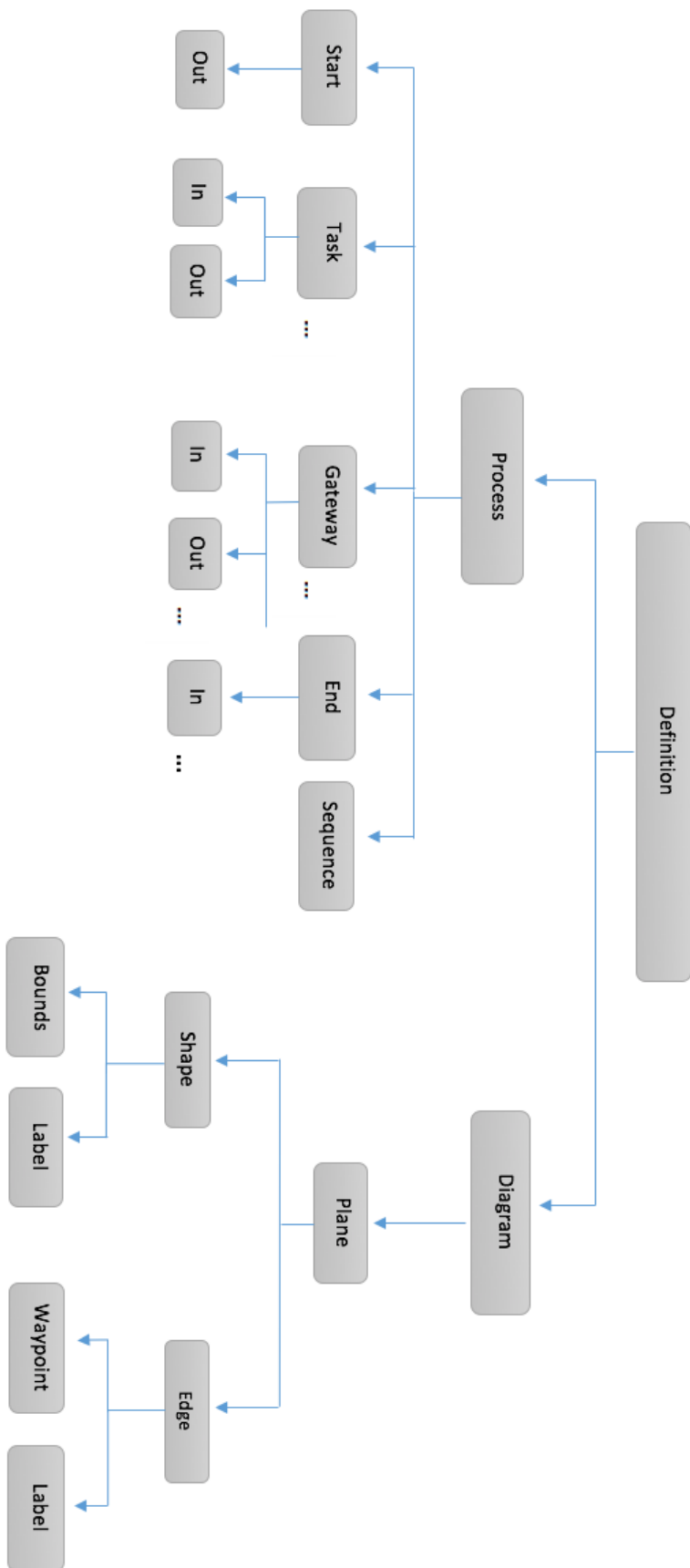


Figura8. XML Schema que traduz a linguagem BPMN

Capítulo 4

Proposta de Solução e Arquitetura do Sistema

No capítulo 3 foi contextualizado o problema que este trabalho se propôs resolver e apresentado o ponto de partida não-teórico, bem como, os requisitos que me propus a cumprir.

Neste capítulo, será apresentada a proposta de solução criada, procurando explicar a abordagem que lhe deu origem, os seus pormenores técnicos, os pontos críticos do problema e a maneira de como estes foram superados.

Na última figura, apresentou-se o *XML Schema* pretendido. Este poder-se-á dividir em 2 blocos de requisitos. Um com a informação presente nas classes da Base de Dados já apresentadas. O outro, com a informação relativa à construção do grafo BPMN. Este segundo foi sem dúvida o mais crítico na elaboração desta dissertação.

Faremos então a analogia entre a informação disponível na BD e a notação BPMN, que permitirá perceber como foi satisfeito o primeiro requisito do *XML Schema*.

De seguida será apresentada a abordagem para obter a informação espacial do grafo que permitirá satisfazer o segundo requisito. Por fim, será ainda apresentada a arquitetura global do sistema.

4.1 Relação entre a sintaxe do BPMN e do iPortalDoc

A analogia entre as duas sintaxes é, na maioria dos casos, intuitiva. A informação do *Workflowtemp* corresponde ao workflow descrito em BPMN. Com o

Estadotemp sucede o mesmo. Já quando existe mais que um *Resultadotemp*, associado à(s) ação(ações) de um estado gera-se uma ou mais Gateways.

Observe-se o seguinte exemplo:

Se ao *estadotemp* “Notificar” estiver associada uma *accotemp* de “notificação” e daqui puderem surgir dois resultados distintos, por exemplo, “notificação aceite” e “notificação rejeitada”, então este mesmo estado terá que ter como elemento seguinte uma *exclusive gateway*, que é usada sempre existe uma bifurcação associada a um processo de decisão.



Figura9. Estado BPMN

Se existirem duas, ou mais, condições de transição iguais entre um estado e outros dois, ou mais, distintos, então ao estado sucederá uma *parallel gateway* que após verificar a sua condição de transição ativa não um, mas 2 ou mais estados (os que lhe estiverem diretamente a montante)



Figura10. Exclusive Gateway BPMN

Se se verificarem, em simultâneo as duas condições anteriores, então serão gerados os dois tipos de gateway, sendo que existirão tantas *parallel gateways*, quantos grupos de transições com resultados (*resultadostemp*) que se repetem.



Figura11. Parallel Gateway BPMN

O Estado Inicial e Final têm a mesma função em ambas as sintaxes.



Figura12. End State BPMN

Sobram ainda as funções de transição, que assumem papéis semelhantes em ambas as sintaxes, mas que têm um funcionamento muito específico na solução que propus como vou ter a oportunidade de explicar mais à frente.



Figura13. Initial State BPMN

Existe, porém, mais um caso notável do sistema onde é usada uma *parallel gateway*, é o que é mostrado na figura seguinte onde se vê que uma barra horizontal mais carregada trunca a entrada no estado imediatamente seguinte, o que na prática significa que só permitirá a transição para o mesmo quando todos os estados imediatamente anteriores estiverem ativos e virem as suas condições de transição aprovadas em simultâneo.

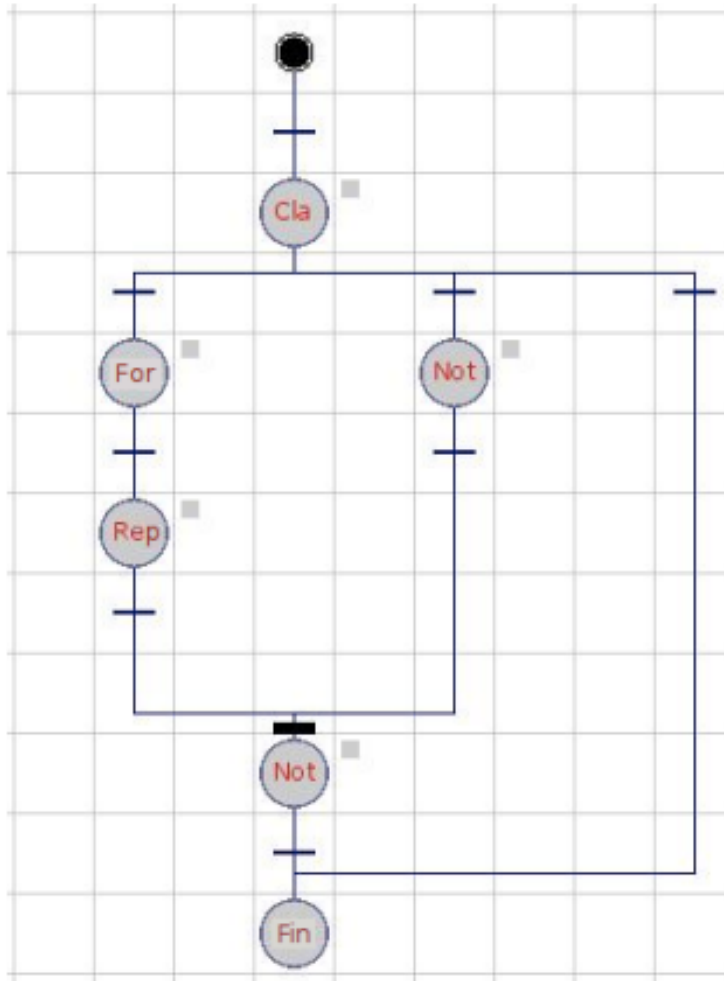


Figura14. Exemplo de um workflow do iPortalDoc com uma situação de “truncamento”

O exemplo anterior configura o caso mais complexo de conversão para BPMN, através das estruturas de *workflow* do iPortalDoc, pois como se pode verificar na imagem seguinte, requer uma *gateway parallel* inicial, devido às condições de transição para os estados “For” e “Not” serem idênticas, uma *exclusive gateway* devido à condição da transição que parte do estado “Cla” para o estado final ser diferente das outras duas (agrupadas na *parallel gateway*) e, finalmente, uma *parallel gateway* posterior representada pelo “truncamento” como acabei de explicar.

Fica o exemplo de como é traduzida graficamente essa conversão, através da figura seguinte:

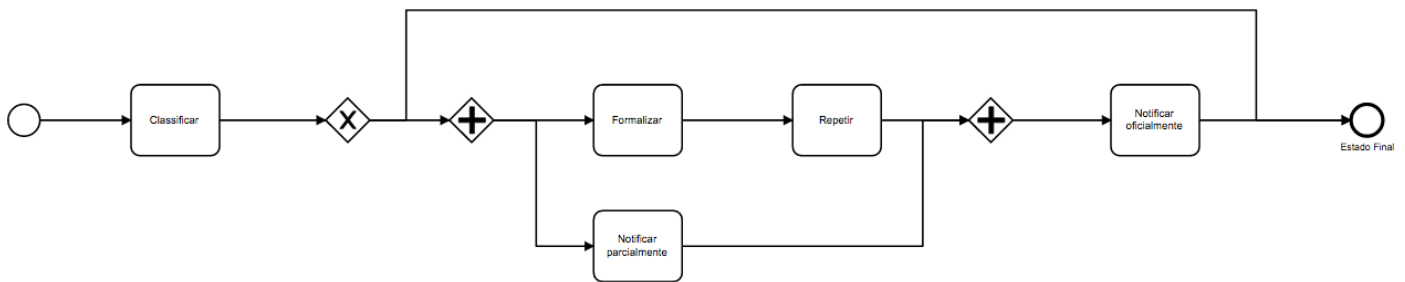


Figura15. Exemplo de um workflow BPMN gerado a partir do iPortalDoc com uma situação de “truncamento”

4.2 Grafo BPMN

Antes de falar no processo de construção do grafo BPMN começarei por apresentar todas as regras que convencionei para a construção do mesmo.

Começemos pelas regras de entrada e saída dos diferentes elementos que apresentamos na secção anterior.

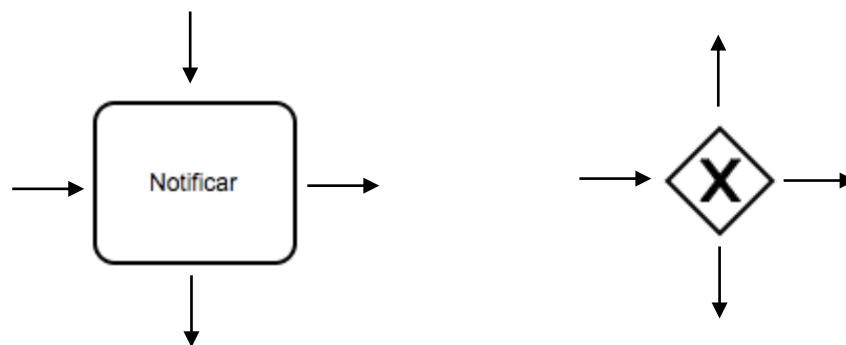


Figura16. Esquema de entrada e saída de transições no digrama BPMN

O esquema para uma *parallel gateway* é o mesmo que o convencional para uma *exclusive gateway*. Um estado inicial só tem uma saída, pela direita, e o final, uma só entrada, pela esquerda.

No processo de construção do grafo optei por não aproveitar as estruturas de workflow iniciais, presentes no iPortalDoc pois, como são construídas manualmente e, ao contrário, das que foram até agora mostradas, apresentam muitas vezes formatos incoerentes, incompletos e até desatualizados. Este processo, embora mais complexo, automatiza a criação de grafos de *workflow* e

oferece uma funcionalidade realmente inovadora ao iPortalDoc. O processo comporta 5 passos essenciais:

1. Importação de informação da Base de Dados

É nesta fase que são classificadas as situações passíveis de introdução de gateway, identificadas em cima. Sendo que existem 4 possibilidades de classificação:

- i. Estado normal
- ii. Estado + Exclusive Gateway
- iii. Estado + Parallel Gateway
- iv. Estado + Exclusive Gateway e Parallel Gateway(s)

Cada um destes casos pode ainda preceder de um caso especial de truncamento, representado com uma *parallel gateway* imediatamente antes.

2. Criação de uma estrutura simples em XML que definirá a sequência do grafo.

Esta opção foi essencial para diminuir o tempo de execução do código, em detrimento de uma solução com uma estrutura em *tree*, ou com classes e apontadores.

3. Definir o posicionamento inicial dos diferentes elementos.

Depois desta fase só é possível definir um posicionamento relativo e não absoluto para o elemento (p.ex estado "A" (x,y) -> (2,3) , estado "B" (x,y) -> (4,3)).

4. Calcular espaçamento entre estados e com isso definir posicionamentos finais ou absolutos.

(p.ex estado "A" (x,y) -> (42,50) , estado "B" (x,y) -> (106,50))

Este ponto é crítico para uma boa inteligibilidade do *workflow*. Se fosse ignorado um ajuste espacial entre os estados, em situações em que entre estes passasse um número considerável de transições, seria impercetível o seu significado. Por outro lado, uma solução que calculasse espaço excessivo entre estados para superar este problema, tornava a solução visualmente desagradável e tive em mente que neste produto, a fácil compreensão da informação mostrada por parte do cliente é essencial.

5. Descrever as transições entre estados

Nesta fase calculam-se e declaram-se os segmentos que compõe as transições entre diferentes elementos. Esta fase só foi possível depois de estabelecer uma “norma” de transição entre estados e Gateways, sendo que cada um deles tem os seus pontos de entrada e saída de transição. A transição é definida em função, portanto, do elemento inicial e do elemento final que conecta e do posicionamento do segundo em relação ao primeiro.

4.3 Arquitetura do sistema

A Arquitetura global da solução proposta é a representada a seguir:

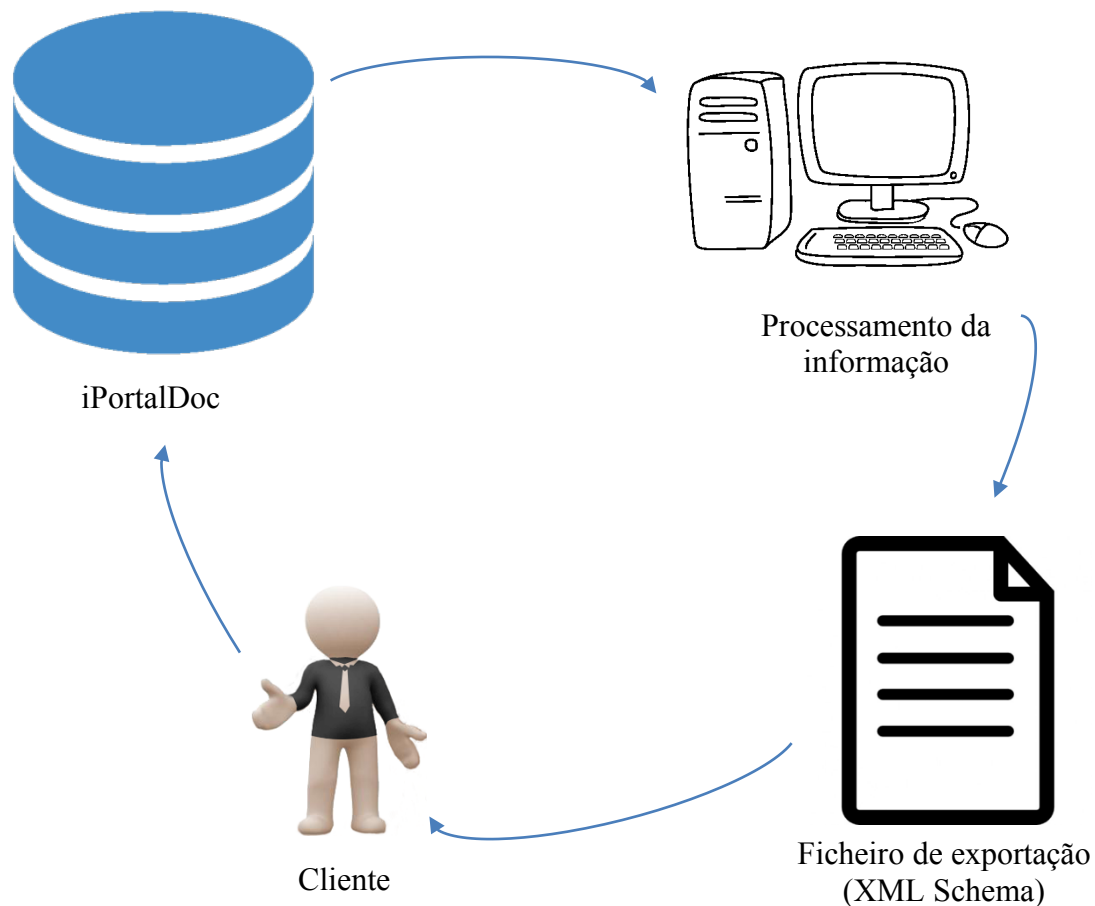


Figura17. Arquitetura do Sistema

Capítulo 5

Implementação

No capítulo 4, foi explanada a minha proposta de solução para o problema inicial, abordando a relação sintática entre as duas linguagens, a construção do grafo BPMN e a arquitetura global do sistema. Neste capítulo, concluir-se-á a apresentação do resultado do trabalho prático abordando a validação dos resultados e apresentação da funcionalidade inserida no iPortalDoc.

5.1 Validação dos resultados

Todos os requisitos iniciais foram satisfatoriamente realizados.

Requisito		
BPM01	Solução compatível com o SO Linux	Validado
BPM02	Exportar workflows recorrendo a uma das linguagens de BPM mais comuns (ex. BPMN, EPC, UML, IDEF, RAD, ...)	Validado
BPM03	Garantir um sistema user-friendly	Validado
BPM04	Se possível, utilizar as linguagens já utilizadas (PHP, XML, XSLT, JavaScript) e respeitar a organização de documentos atual.	Validado
BPM05	Prevenir erros do cliente	Validado
BPM06	Garantir um tempo de resposta rápido	Validado
BPM07	Fazer uma integração da funcionalidade coerente com o restante produto.	Validado

Tabela2. Validação da lista de requisitos

Garantir um tempo de resposta rápido em *workflows* de elevada complexidade e dimensão nem sempre foi tarefa fácil, mas o melhoramento contínuo do processo de seleção e pesquisa de informação ao longo da conceção do produto permitiu obter

respostas muito satisfatórias que não excederam mais que 5 segundos nos piores casos experimentados.

Durante a fase de verificação e avaliação dos resultados foram testados variados exemplos de *workflows* reais e usados em clientes, mas também fictícios, criados especialmente para contemplar os casos mais complexos. Foi num destes *workflows* teste, importados da base de dados de um cliente que me deparei com um verdadeiro teste à prevenção do erro do cliente. Um dos técnicos da IPBrick, responsável por modificar um dos *workflows* de um cliente e dado, também, o processo moroso que está associado a estas modificações no motor de wf original, decidi desconectar um estado inicial de outro a montante para o ligar diretamente a um dos estados finais desse workflow, deixando em aberto uma grande sequência de estados. Estes ligavam entre si e, por isso, um simples teste de verificação de existência de transição de entrada e transição de saída em todos os estados não resolveria o problema, levando a que se importasse informação desnecessária e depois esta fosse declarada de uma maneira que não estava bem especificada.

Para resolver este problema foi criado um teste prévio à importação da informação que sinaliza os estados que, começando no estado inicial, nunca são percorridos por maior que seja o número de transições sucessivas realizadas. Esta pesquisa precisa de um número máximo de iterações igual ao número total de estados e é feita aquando do seu tratamento (de cada estado).

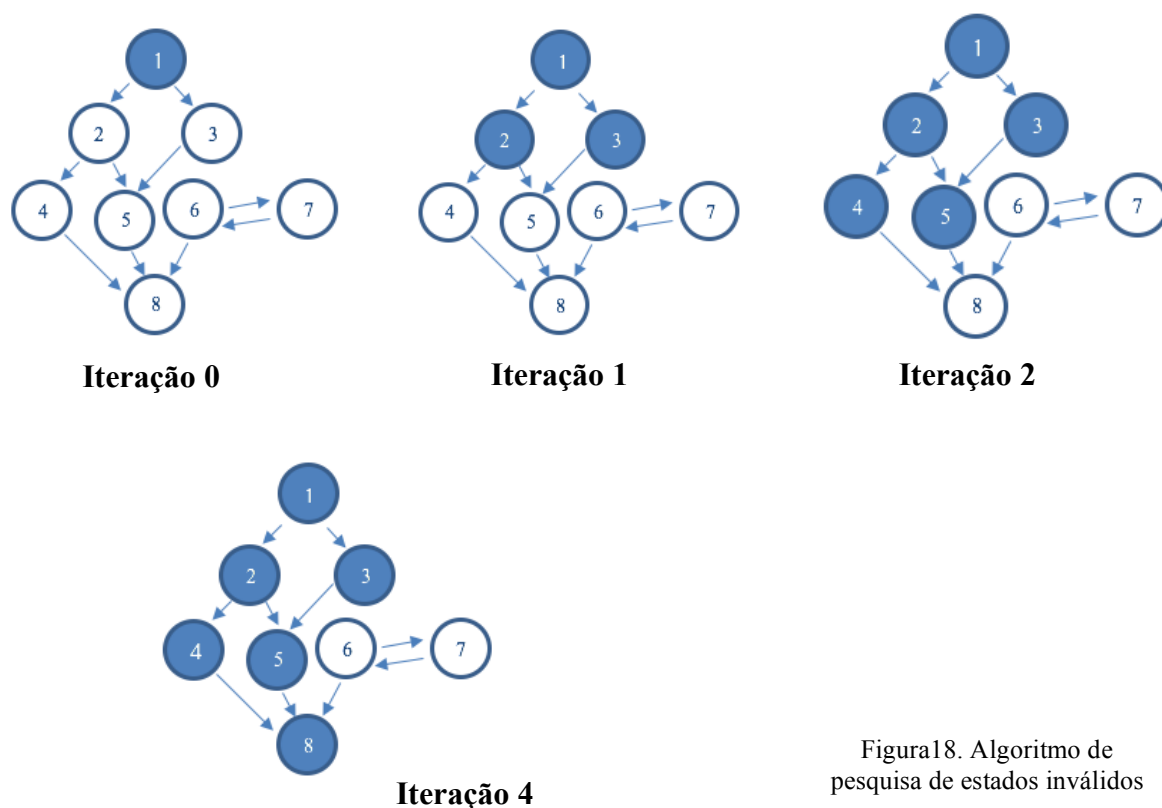


Figura18. Algoritmo de pesquisa de estados inválidos

Na imagem anterior é demonstrada a atuação do algoritmo que permite detetar os estados inválidos, transferindo, para os elementos-filho, um elemento genético de validade, todo o estado que estava considerado válido aquando do seu tratamento, sendo que o primeiro a ser tratado é o estado inicial que é considerado válido à *priori*.

5.2 Integração no iPortalDoc

Optei por fazer a integração desta funcionalidade no iPortalDoc, seguindo o critério de outras semelhantes, colocando no menú do canto superior direito da página, que permite ver o workflow original, a opção *Exportar BPMN*. Desta forma, o cliente disporá da nova funcionalidade no momento em que dela irá necessitar (quando tiver problemas de compreensão do workflow ou desejar altera-lo) da maneira que está habituado a fazer no iPortalDoc.

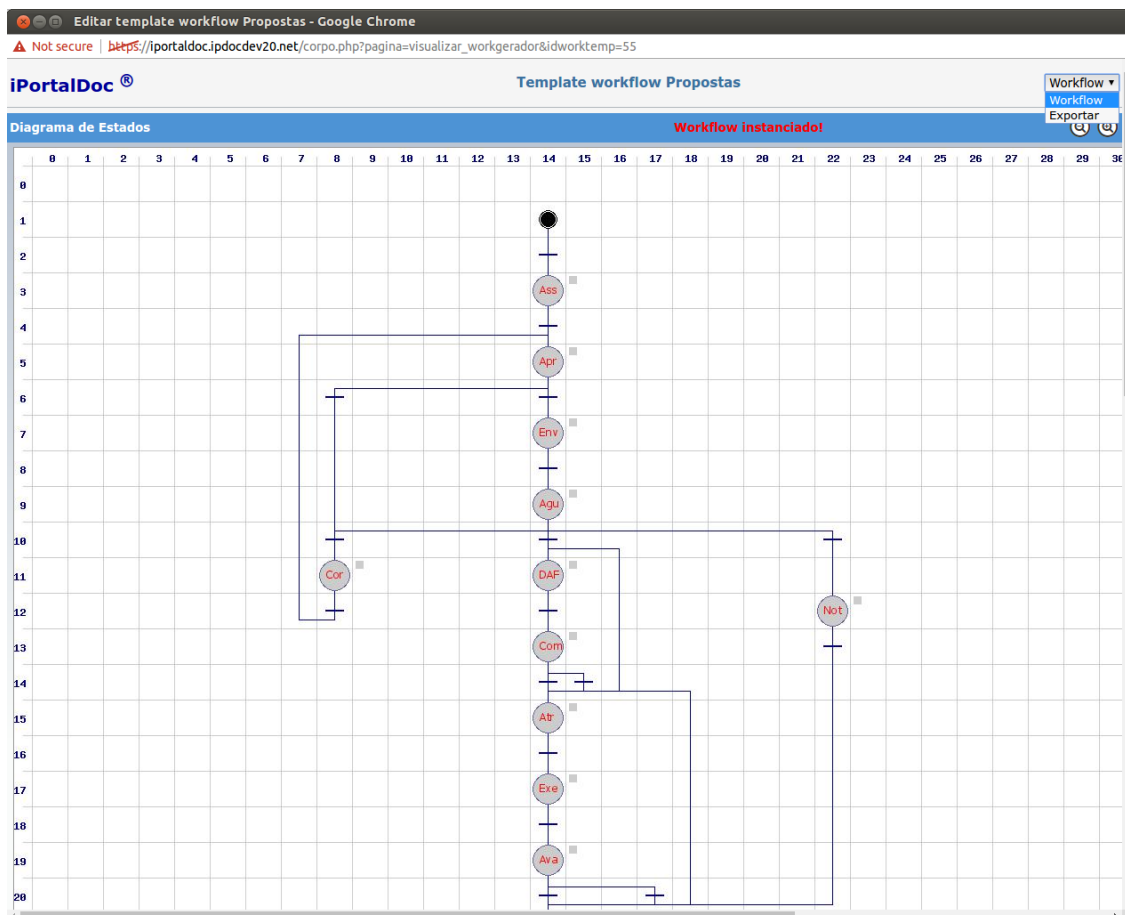


Figura 19. Integração visual da Exportação BPMN no iPortalDoc

Capítulo 6

Conclusão

Este último capítulo, pretende tecer as considerações finais sobre a presente dissertação, mas também sobre o trabalho realizado, que é experienciado de uma forma muito própria em dissertações realizadas em ambiente empresarial.

Este projeto pretendeu dotar o iPortalDoc de uma nova funcionalidade de exportação para uma linguagem de BPM. O objetivo do trabalho foi atingido, possibilitando a criação de workflows mais inteligíveis, manuseáveis e protegidos contra erros de construção de workflows. O produto foi organizado de forma modular para que, por exemplo, a criação do grafo possa ser utilizada na exportação para outras linguagens semelhantes, no futuro.

Irá ser feita uma reflexão do que ficou por fazer, bem como, serão apresentadas propostas para a continuação do desenvolvimento desta funcionalidade.

6.1 Estágio na IPBRICK SA

Na *IPBRICK* encontrei uma equipa jovem, não fosse este um projeto que começou com uma *Startup* iniciada a partir da Faculdade de Engenharia da Universidade do Porto e que hoje conta com um sistema de Gestão da I&D e Inovação certificado.

Contudo, sendo a minha especialização na área da gestão industrial não posso deixar de sugerir que se promova mais momentos de *brainstorming* na sua atividade. Dou, por exemplo, a adoção de uma rotina de *whiteboard* diário onde cada um dos elementos da equipa é chamado a partilhar o que fez no dia anterior, se cumpriu o seu planeamento e/ou que dificuldades encontrou ao longo do dia. O mesmo elemento seria convidado a apresentar o que tem planeado fazer hoje e partilhar qualquer dúvida ou problema com a equipa.

6.2 Trabalho futuro

Neste trabalho haveria espaço para melhorar, sobretudo, apostando numa integração visual dos grafos BPMN no próprio iPortalDoc, para que o cliente não necessitasse de nenhum leitor externo complementar. Poderiam ainda ser dadas indicações de ajustes automáticos no *workflow* para notificar o cliente do seu erro e tornar a experiência mais interativa.

Referências

- [1] <http://www.ipbrick.com/pt-pt/>, acessido pela última vez a 15 de Julho de 2016.
- [2] Manual de Utilizador v4.0, iPortalDoc - Gestor Documental, IPBRICK International (2012);
- [3] Briol, Patrice (2009) “BPMN, the Business Process Modeling Notation Pocket Handbook”;
- [4] BPMN. Disponível em <https://www.modeliosoft.com/en/technologies/bpmn.html>, acessido pela última vez a 2 de Julho de 2016.
- [5] Coalition, Workflow Management. Workflow Management Coalition Terminology &Glossary. 1999.
- [6] O que é GNU/Linux?. Disponível em <https://www.debian.org/intro/about>, acessido pela última vez a 15 de Julho de 2016.
- [7] PostgreSQL Global Development Group. About postgresql, 2016. Disponível em <http://www.postgresql.org/about/>, acessido a última vez a 15 de Julho de 2016.
- [8] Manual do PHP. Disponível em <https://secure.php.net/manual/preface.php>, acessido pela última vez em 15 de Junho de 2016.
- [9] Garrett, Jesse J., Ajax: A New Approach to Web Application, publicado em 18 Fevereiro de 2005, <http://www.adaptivepath.com/ideas/essays/archives/000385.php>
- [10] José Carlos Ramalho e Pedro Rangel Henriques. XML & XSL: da teoria à prática. Série Tecnologias de Informação ISBN-972-722-347-8. Editora FCA, 1st ed. edição, Outubro 2002.
- [11] XSLT. Disponível em <https://msdn.microsoft.com>, acessido pela última vez a 15 de Julho de 2016.
- [12] JavaScript Tutorial. Disponível em <http://html.net/tutorials/javascript>, acessido pela última vez a 15 de Julho de 2016.

[13] “Regras para a Apresentação de Dissertações de Cursos de Mestrado da FEUP”, Faculdade de Engenharia da universidade do Porto, Junho de 1995.

[14] Borges, José Luís Cabral de Moura; Dias, Maria Teresa Galvão; Cunha, João Bernardo de Sena Esteves Falcão e (2015) Modelação de dados em UML: uma abordagem por problemas, Lisboa: FCA - Editora de Informática

[15] Allweyer, Thomas(2010) BPMN 2.0 : introduction to the standard for business process modeling, Norderstedt : Thomas Allweyer ed.