

Final

Gestão Web de Livros Digitais

RELATÓRIO DE ESTÁGIO DE MESTRADO

Eduardo Nuno Ferreira Vasconcelos

MESTRADO EM ENGENHARIA INFORMÁTICA



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

junho | 2016

M
S Ges
D-R(2ex)

T/M
004
VAS Ges
+EDR (2x1)

Gestão Web de Livros Digitais

RELATÓRIO DE ESTÁGIO DE MESTRADO

UNIVERSIDADE DA MADEIRA
BIBLIOTECA

Eduardo Nuno Ferreira Vasconcelos

MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTADOR

David Sardinha Andrade de Aveiro

CO-ORIENTADOR

Hélder Gouveia Figueira Pestana

Orientadores

David Sardinha Andrade de Aveiro, PhD
*Professor da Faculdade de Ciências Exactas e da Engenharia da
Universidade da Madeira*

Hélder Gouveia Figueira Pestana, Lic.
Engenheiro Informático na Lernasnuvens – Software Solutions

Resumo

Este relatório tem como objetivo descrever o trabalho desenvolvido, as decisões tomadas e outras informações relevantes durante a realização do estágio durante o final do curso de Mestrado em Engenharia Informática, da Universidade da Madeira.

Este estágio tem como propósito a criação de uma plataforma para a empresa Lernasnuvens, Software Solutions Lda. que assistirá na produção, distribuição e marketing de livros digitais (ebooks).

O projeto terá por base a especificação DEMO, com a qual é pretendido obter informação e requisitos mais completos e próximos do que é pretendido, quando comparado a uma abordagem clássica de engenharia. Juntamente com a aplicação do DEMO será também utilizado o BPMN 2.0 e um *engine* de *workflow* que o suporte para a criação concreta da plataforma.

Por fim, tentar-se-á perceber a utilidade e o potencial da aplicação destes no desenvolvimento de *software*, assim como potenciais lacunas.

Palavras-chave

DEMO

BPMN

Publicação de livros

Engenharia de Software

Abstract

This report has as a primary objective to describe the work developed, decisions made and other relevant informations regarding the execution of the internship during the end of the master's degree in Informatics Engineering in the University of Madeira.

The internship has the primary objective of creating a platform to assist both employees and end customers of Lernasnuvens, Software Solutions Lda. in the production, distribution and marketing of digital books (ebooks).

The project will have as a base the DEMO specification, and with it it's intended to obtain more complete information and requirements that are closer to what's intended when compared to those obtained using a classic engineering approach. Along the application of DEMO, BPMN 2.0 and a workflow engine that supports it will be used for the actual development of the platform.

Lastly, an attempt will be made to understand the utility and potential of the application of DEMO and BPMN on software development as well as potential problems or faults.

Keywords

DEMO

BPMN

Book Publishing

Software engineering

Agradecimentos

Em primeiro lugar queria agradecer aos meus pais, Maria da Luz Lucas Ferreira Vasconcelos e António José Ferraz Vasconcelos, que sempre me tentaram dar as melhores condições, e no mesmo ponto, a minha avó Merita Lucas Ferreira que sempre esteve comigo e que muito fez pela minha pessoa. Não querendo deixar ninguém para trás, agradeço também a vários membros da minha família mais estendida.

Estes agradecimentos não poderiam estar completos sem mencionar amigos e ex-colegas, que de uma forma ou de outra me ajudaram (ou desajudaram); poderia ter chegado a este ponto sem vocês, mas não seria a mesma coisa. Em especial queria agradecer a quatro grandes amigos: Janine Fernandes, Daniel Garcês, Lino Oliveira e também ao Rúben Henriques.

Por fim, quero agradecer a todos os meus professores e educadores, que me acompanharam por o que foi quase a minha vida toda, dando também uma menção especial ao meu orientador, o professor David Aveiro, e co-orientador, Hélder Pestana que me assistiram durante o desenvolvimento deste projeto. Por fim, também queria agradecer a todos na LerNasNuvens, Nova Delphi e DIKE Madeira por me terem recebido e me proporcionado um ambiente tão amigável durante o desenvolvimento do projeto.

Índice

Orientadores	i
Resumo.....	iii
Palavras-chave	v
Abstract	vii
Keywords	ix
Agradecimentos.....	xi
Índice.....	xiii
Índice de figuras.....	xv
Lista de tabelas	xvii
Abreviaturas e Símbolos.....	xix
I. Introdução	1
I.1. Contexto.....	1
I.2. Motivação	1
I.3. Objetivos	2
I.4. Estrutura da tese	2
I.5. Descrição da empresa.....	3
II. Estado da Arte	5
II.1. DEMO	5
II.1.1. Modelo ontológico.....	5
II.1.2. Teoria PSI	5
II.1.3. Metodologia de modelação	7
II.2. BPMN.....	10
II.2.1. Elementos do BPD	10
II.2.2. Engines de workflow/BPMN	14
II.3. DMS.....	19
II.3.1. OpenKM.....	19
II.3.2. Alfresco	20
II.3.3. ownCloud.....	20

II.3.4. Conclusão	21
III. Especificação e Implementação.....	25
III.1. Requisitos e funcionalidades.....	25
III.2. Metodologia	26
III.3. Conversão DEMO para implementação	51
IV. Arquitetura	55
IV.1. Tecnologias e Linguagens.....	55
IV.1.1. MySQL.....	55
IV.1.2. Groovy e Java	55
IV.1.3. Apache Tomcat	55
IV.1.4. ownCloud	55
IV.1.5. JavaScript	56
IV.2. Componentes.....	56
IV.2.1. Utilizadores.....	56
IV.2.2. Produção	56
IV.2.3. Distribuição	57
IV.2.4. Marketing	57
IV.2.5. Outros	57
IV.3. Funcionalidades implementadas.....	57
IV.4. Como se utiliza a ferramenta	59
V. Testes	61
VI. Conclusões e Trabalho Futuro	65
VII. Referências.....	69
VIII. Anexos	73
VIII.1. Anexo A – Versão inicial dos diagramas DEMO	73
VIII.2. Anexo B – Modelos entidade-relação da base de dados	77
VIII.3. Anexo C – Screenshots da plataforma	82

Índice de figuras

Figura 1. Legenda do ATD	8
Figura 2. Exemplo dos tipos de transações	9
Figura 3. Exemplo de uma transação auto-ativada (T23)	9
Figura 4. Pool e duas lanes.....	11
Figura 5. Artefactos	11
Figura 6. Gateways	12
Figura 7. Atividades	13
Figura 8. Eventos	13
Figura 9. Versão final do ATD	29
Figura 10. Versão final do PSD	31
Figura 11. Versão final do OFD	33
Figura 12. Tipo de facto categoria da obra e classes associadas - Modelo OFD	34
Figura 13. Tabelas resultantes do tipo de facto e classes - Modelo relacional	34
Figura 14. Tabelas antigas para distribuição	35
Figura 15. Novas tabelas para distribuição	35
Figura 16. Tabela para registo de eventos	36
Figura 17. Tabela antiga para valores de escalão	36
Figura 18. Novas tabelas para guardar valores de escalões de preços	36
Figura 19. Tabelas com orçamentação ligada a um único serviço	37
Figura 20. Passagem a uma entrada de orçamento por livro	37
Figura 21. Tabela para registo de ações de gestão	37
Figura 22. Tabela para registo de países e regiões.....	38
Figura 23. Novas tabelas para registo das funções dos funcionários	39
Figura 24. Nova estrutura para distribuição de livros	40
Figura 25. Novas tabelas para registo de serviços de marketing ..	41
Figura 26. Tabela `book` antiga	42
Figura 27. Tabela `book` com novas alterações	42
Figura 28. Exemplo de transações	42
Figura 29. Passagem para diagrama BPD da transação T23	43
Figura 30. Exemplo de um processo para um serviço	45
Figura 31. Processo para serviços acessórios generalizado	45
Figura 32. ATD, versão após desenvolvimento	49
Figura 33. PSD, versão após desenvolvimento	50
Figura 34. OFD, versão após desenvolvimento	51

Figura 35. Versão inicial do ATD	74
Figura 36. Versão inicial do PSD	75
Figura 37. Versão inicial do OFD.....	76
Figura 38. Versão 1 da base de dados	77
Figura 39. Versão 2 da base de dados	78
Figura 40. Versão 3 da base de dados	79
Figura 41. Versão 4 da base de dados	80
Figura 42. Versão 5 da base de dados	81
Figura 43. Página do dashboard/portal	82
Figura 44. Página resumo para criação de uma nova obra	82
Figura 45. Decisão sobre resultado de um serviço	83
Figura 46. Realização de um pedido de distribuição.....	83
Figura 47. Gestão interna de um pedido de distribuição	84
Figura 48. Gestão interna sobre marketing.....	84

Lista de tabelas

Tabela 1. Quadro resumo dos vários engines de BPM	18
Tabela 2. Quadro resumo dos DMS	23
Tabela 3. Versão final da TRT	27
Tabela 4. Resumo da análise inicial da plataforma.....	62
Tabela 5. Versão inicial da TRT	73

Abreviaturas e Símbolos

API – Application Programming Interface
AR – Action Rules
ATD – Actor Transaction Diagram
BPD – Business Process Diagram
BPM – Business Process Management
BPMI – Business Process Management Initiative
BPMN – Business Process Model and Notation
C-acts – Coordination acts
CAP – Coordination-Actors-Production
C-facts – Coordination facts
CSS – Cascading Style Sheets
DEMO – Design & Engineering Methodology for Organizations
DMS – Document Management System
ePub – Electronic Publication
HTML – HyperText Markup Language
INRIA – Institut national de recherche en informatique et en automatique
ISBN – International Standard Book Number
JSP – JavaServer Pages
OCR – Optical Character Recognition
OFD – Object Fact Diagram
P-acts – Production acts
P-facts – Production facts
PIF – Performa Informa Forma
PSD – Process Structure Diagram
PSI – Performance in Social Interaction
SQL – Structured Query Language
TRT – Transaction Result Table
WebDAV – Web-based Distributed Authoring and Versioning

I. Introdução

Com este relatório pretende-se explicitar e explicar o trabalho realizado durante o estágio que se iniciou no ano escolar de 2014/15 para o curso de Mestrado em Engenharia Informática.

O trabalho começa com a aplicação de conhecimentos obtidos na unidade curricular de Engenharia Organizacional, aplicando a metodologia DEMO como forma de se melhor definir os requisitos da ferramenta e as responsabilidades de cada ator durante a execução dos diversos processos que são executados no âmbito das tarefas relevante à plataforma e à empresa. Será também utilizado um motor de *workflow* e de BPMN na criação da plataforma, existindo vários será necessário fazer uma avaliação destes, no entanto, sendo o BPMN algo novo, envolve-se aqui também a necessidade de aprendizagem, no entanto a aquisição de novos conhecimentos nunca deverá ser um impedimento mas sim uma motivação.

I.1. Contexto

O estágio curricular, previsto pelo plano de conclusão de estudos, é uma das opções para fim de curso. Uma grande atração, passa pela obtenção de experiência no mundo do trabalho, sendo o trabalho que será desenvolvido utilizado por outras pessoas como um ferramenta de suporte para a execução do seu trabalho.

I.2. Motivação

A motivação começa por uma análise e implementação de uma plataforma web para a gestão de livros digitais para a empresa *Lernasnuvens, Software Solutions, Lda*, uma *startup* na área da edição digital.

Por volta de 1450, Johannes Gutenberg criou a prensa móvel na Europa, isto seria uma revolução na criação de livros, permitindo que se produzissem milhares de páginas de por dia de trabalho, um valor muitíssimo superior quando comparando com a cópia à mão. Voltando para o presente, ocorreram já revoluções na indústria editorial devido aos livros digitais (ebook). Embora existindo uma maior adoção nos Estados Unidos da América (já tendo atingido os 18% das vendas totais de livros, em 2011), tem havido uma adoção crescente dos livros digitais tanto nos Estados Unidos da América como na Europa. Na Europa os mercados

mais desenvolvidos é o alemão e o francês. Embora os mercados de Portugal, Itália e Espanha se encontrem menos desenvolvidos, o mercado espanhol ganhou velocidade em apostas fortes no mercado pelas editoras. Em 2011 o mercado de ebooks era 1% do mercado de livros, quantia que embora possa ser mínima quando comparado com Portugal é imensamente maior: para o mesmo ano, segundo editoras nacionais o valor do mercado de ebooks era de 0.01%. [1]

É portanto é de considerar que havendo apostas por parte de editoras, com suporte, entre outras coisas, de ferramentas para tal que tal percentagem conseguirá aumentar e tornar-se relevante.

I.3. Objetivos

O objetivo consiste na criação de uma plataforma que permita a gestão, tanto por parte de funcionários como de clientes finais, da produção, distribuição e marketing de livros para a empresa.

A criação desta plataforma passa pela aplicação da metodologia DEMO e na utilização de um motor de *workflow* BPMN.

Por fim, pretende-se que a plataforma seja uma ajuda e um veículo para levar mais longe os clientes e a empresa e não um artefacto pesado que arrasta tudo e todos.

I.4. Estrutura da tese

Este relatório encontra-se repartido em 7 capítulos que descrevem o trabalho realizado durante o estágio, que são descritos de seguida:

O capítulo 2, **Estado da arte**, descreve em maior pormenor o DEMO e BPMN, fornecendo também uma descrição das maiores ferramentas utilizadas na plataforma e uma comparação entre algumas alternativas. É também descrito sucintamente plataformas semelhantes à que será desenvolvida.

No capítulo 3, **Especificação e Implementação**, é indicado os requisitos da plataforma juntamente com a descrição do processo efetuado ao longo do desenvolvimento do projeto, incluindo alterações realizadas às diversas partes deste.

Durante o capítulo 4, **Arquitetura**, são descritas as várias tecnologias e linguagens utilizadas para o desenvolvimento do projeto; são explicados os vários componentes que estão incluídos na plataforma

e é feita uma descrição, baseada nos requisitos, do que foi atingido. Por fim, é fornecido uma explicação da forma de utilização da plataforma.

O capítulo 5, **Testes e Resultados**, trata da realização de testes sobre a solução, eventuais correções, resultados obtidos e analisa os objetivos atingidos quando comparados com os objetivos iniciais.

O capítulo 6, **Conclusões e Trabalho Futuro**, fecha o relatório com um apanhado geral das conclusões obtidas ao realizar o projeto, aponta as principais dificuldades sentidas durante o desenvolvimento, fornece os primeiros passos para o trabalho futuro e encerra com um pequeno apanhado da experiência pessoal ao estagiar.

Para finalizar, o capítulo 7, **Referências**, onde são indicadas as diversas referências utilizadas para a realização deste relatório.

I.5. Descrição da empresa

A empresa Lernasnuvens - Software Solutions, Lda. que irá operar no sector da edição digital, foi criada com o intuito de explorar uma lacuna diagnosticada no mercado regional, a nível nacional de ter a vantagem de ser dos primeiros operadores nesta área, e também, porque o mercado externo a nível de países com língua lusófona apresentam grande margem de crescimento. Como tal, acredita apresentar uma proposta inovadora a nível da Região autónoma da Madeira, com largo potencial de crescimento no mercado continental português, e grandes possibilidades de afirmação nos países de língua lusófona.

Tendo a consciência que a edição digital irá vir a ocupar um lugar de destaque, embora o desaparecimento ou degradação da edição tradicional seja uma incógnita, a empresa pretende contribuir para implementação do digital no mundo editorial. A empresa pretende também promover e incrementar gradualmente o hábito de leitura através do digital como forma de esbater limitações.

O fornecimento de serviços de edição digital que acrescentam valor ao negócio de pequenas e médias editoras, assim como outras entidades que pretendam editar, e indiretamente atingindo o consumidor final através de um produto que promova o conforto na leitura e motive quem utiliza edições literárias, como forma de lazer ou desenvolvimento profissional, primando pela qualidade e resposta eficaz às necessidades dos seus clientes é a missão da Lernasnuvens.

II. Estado da Arte

II.1. DEMO

A ontologia empresarial pode ser vista como o estudo da construção e operação de uma organização de uma forma que seja independente da realização e implementação, posto de outra forma, é um modelo de alto nível de uma organização.

Gerir uma empresa tem vindo a se tornar mais complicado devido à complexidade que estas apresentam, e tal complexidade só pode ser dominada se existir uma teoria compreensiva sobre o tipo de coisas cuja complexidade pretendemos dominar e se existirem métodos e técnicas de análise baseados nessa mesma teoria. [2] [3]

A utilização do DEMO como primeiro passo no desenvolvimento do projeto, trás várias vantagens, nomeadamente: faz com que as pessoas sejam o foco central da organização; há uma análise e desenho dos processos de negócio mais orientado aos serviços e assim como uma maior transparência da organização incluindo a identificação da responsabilidade e propriedade dos dados e tarefas; reduz a complexidade dos modelos e dá uma base mais objetiva para a identificação dos requisitos. [4]

II.1.1. Modelo ontológico

Para que se possa lidar com os desafios atuais e futuros é necessário um modelo conceptual da organização que seja coerente, consistente, compreensivo, conciso e essencial.

Entrando mais em detalhe: primeiramente os pontos distintos do modelo têm de consistir num inteiro verdadeiramente lógico e integral. Requer estar livre de inconsistências ou contradições lógicas. Em terceiro lugar, todos os pontos relevantes do modelo necessitam de estar cobertos. Em quarto lugar, nenhum aspeto supérfluo deve ser considerado, de forma a ter um modelo o mais compacto possível. Por fim, necessita de ser independente da realização e implementação da organização. [2] [3]

II.1.2. Teoria PSI

A teoria PSI fornece uma explicação da construção e operação das organizações. Esta teoria é composta por vários axiomas que serão explicados de seguida.

II.1.2.1. Axioma da operação

O axioma da operação diz-nos que pessoas numa organização realizam dois tipos de atos: de produção (P-acts) e de coordenação (C-acts). Ao realizar P-acts é realizada a função da organização e com a realização dos C-acts as pessoas entram e cumprem com compromissos relativos aos atos anteriores. Um papel de ator é definido como tendo a autoridade para realizar um único tipo de P-act. Uma pessoa ao realizar papel de ator é chamado de ator. Para que P-acts possam ser realizados é necessário que exista competência, que é definida como o conhecimento coletivo, “saber fazer” e experiência necessária para realizar P-acts (normalmente relacionados com profissões). A autoridade é definida como a autorização a um sujeito por parte de uma instituição para a realização de um tipo particular de atos (e.g., o médico M no hospital H). Por fim, a responsabilidade pode ser vista como a necessidade social sentida por parte de uma pessoa para realizar os C-acts que estão autorizados, de uma forma responsável (e.g., interação entre o médico M do hospital H com um paciente deste mesmo hospital). [5]

II.1.2.2. Axioma da transação

Com o axioma da transação ganha-se o conhecimento que os atos ocorrem em padrões genéricos recorrentes denominados de transações, sendo executado por dois atores, que realizam atos alternadamente. Uma transações divide-se em três fases, começando com a fase do pedido, cuja gênese advém com um pedido de um dos atores (iniciador) e termina (se houver sucesso) com uma promessa por parte do segundo ator (executor). A fase intermédia designa-se de fase de execução, sendo a altura onde o executor realiza o P-act. A fase final, do resultado, inicia-se com uma declaração por parte do executor e termina (se houver sucesso) com a aceitação do iniciador. [5]

II.1.2.3. Axioma da composição

O axioma da composição diz-nos que as transações relacionam-se com outras em uma de duas formas: uma transação está incluída noutra ou uma transação é auto ativada. A inclusão de transações leva a uma estrutura dos processos de estilo árvore. [5]

II.1.2.4. Axioma da distinção

O axioma da distinção distingue três habilidades humanas, chamadas *forma*, *informa* e *performa*. A *forma* relaciona-se com os

aspectos da forma da comunicação e informação, posto de uma forma simples, é forma como dizemos e percebemos frases numa linguagem, a transmissão de dados ou forma de armazenamento de dados. A *informa* diz respeito com os aspectos de comunicação e informação e portanto lida-se com a comunicação e informação, abstraindo-se dos aspectos de forma, ou seja, o foco é a partilha de ideias entre pessoas, memorizar e relembrar conhecimentos e raciocínio. Por fim, a *performa* tem como responsabilidade a criação de coisas novas e originais, direta ou indiretamente através de comunicação, ou seja, entrar em compromissos, decidir ou julgar. Esta é considerada como a habilidade humana essencial para realizar negócios de qualquer tipo. [2] [4] [5] [6] [7] [8]

II.1.3. Metodologia de modelação

O primeiro passo na aplicação do DEMO é a realização da análise PIF, criando um documento que descreva, o mais detalhadamente possível, todas as atividades englobadas na organização. Nesta análise são identificados os diferentes atos e factos, sendo distinguidos em três diferentes grupos: todos os atos e factos na organização e todos os atos de coordenação o que é *performa*; idem para *informa* e para *forma*. Durante o processo de aprendizagem desta metodologia, a identificação dos diferentes tipos ocorre com a marcação utilizando cores diferentes (vermelho, verde e azul, respetivamente), no entanto outros métodos de identificação podem ser usados.

Uma vez concluída a análise PIF é possível dar início à análise CAP. Continuando com o mesmo documento mencionado previamente, e considerando apenas os itens *performa*, estes são divididos em dois grupos: de produção (P-acts e P-facts) e de coordenação (C-acts e C-facts), identificados através de losangos e círculos, respetivamente. Juntamente com esta separação, deverão ser identificados todos os atores (a amarelo). Tal como no estilo de identificação mencionado no parágrafo anterior, a marcação dos diferentes itens da análise pode ser feita de outras formas, no entanto as indicadas são as indicadas durante a aprendizagem desta metodologia.

Uma vez concluídas estas análises podemos passar à elaboração da TRT, nesta é identificado para cada transação o seu resultado assim como o iniciador e o executor; e útil a adição de um campo de identificação para os papéis de ator, onde a sua composição é 'A' seguido de um número (usualmente a transação onde este papel executa uma

transação) ou 'CA' seguido de um número (para os papéis de ator externos à organização, como clientes). Um papel de ator executa apenas uma única transação (podendo iniciar múltiplas, uma única ou nenhuma) com exceção dos papéis de ator externos à organização, pois estes são vistos como papéis de ator compostos. Por fim, os nomes atribuídos aos papéis de ator deverão estar ligados à transação que estes executam, quando possível (e.g.: para uma transação denominada 'Início de distribuição' o nome do papel correspondente seria 'Iniciador de distribuição').

Estando a TRT concluída, é possível elaborar o ATD, uma representação mais gráfica da TRT. Os elementos deste diagrama incluem os papéis de ator (elementares e compostos), a fronteira da organização, as transações e os *links* de iniciador e executor.

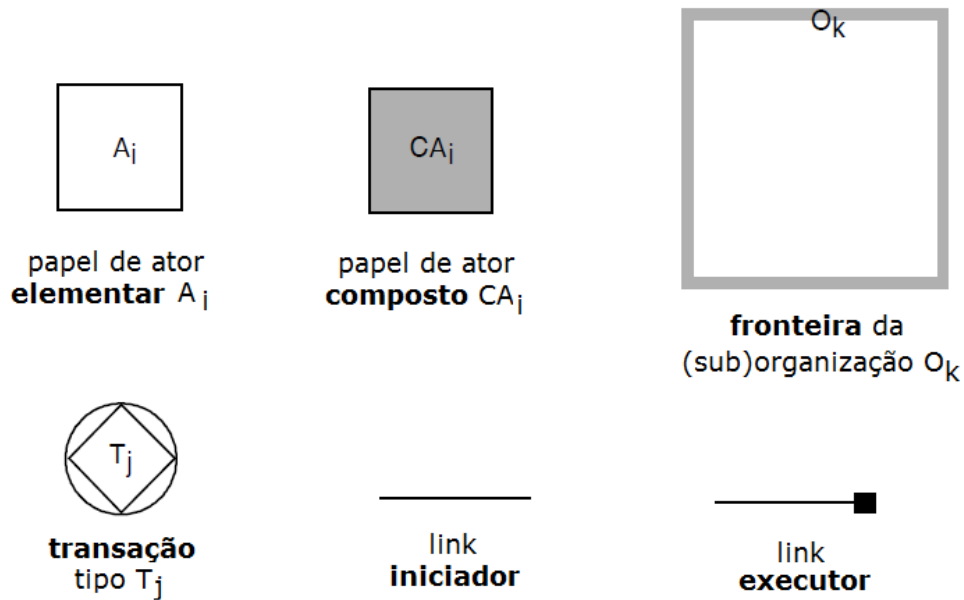


Figura 1. Legenda do ATD

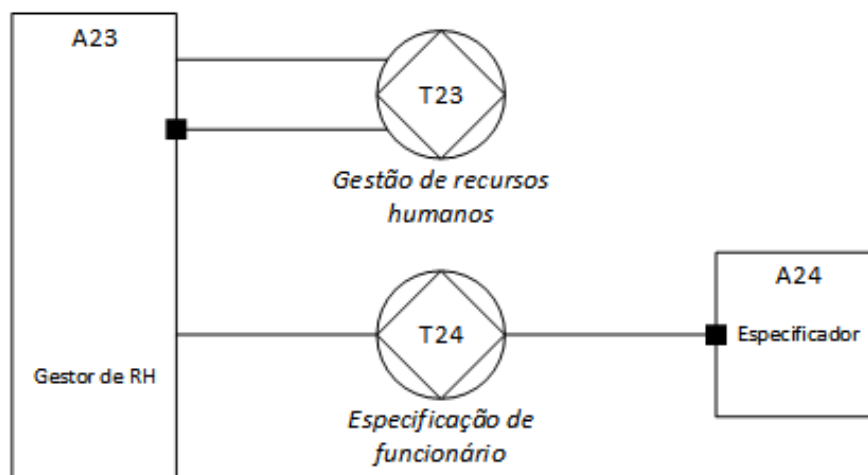


Figura 2. Exemplo dos tipos de transações

Continuando com a elaboração dos digramas, passamos agora para o PSD, onde se obtém uma visão mais detalhada dentro de cada transação. Neste é indicado, para cada etapa da transação, quais os factos de outras transações que são causadas, caso existam, (com um *link* causal, representado por uma seta com linha contínua) e quais os factos que precisa de esperar para que um certo facto possa ser realizado (através de um *link* condicional, representado por uma seta tracejada). As transações auto-ativadas são também indicadas através de uma representação semelhante à que se segue.

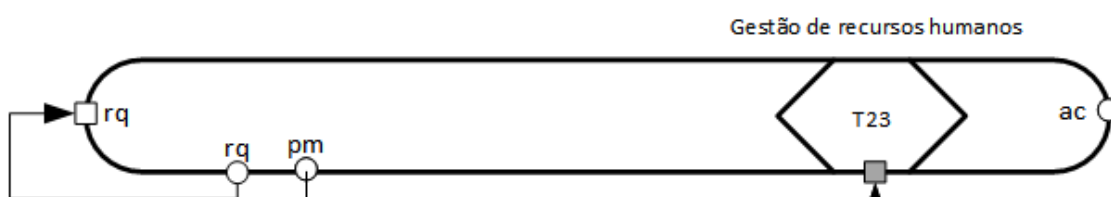


Figura 3. Exemplo de uma transação auto-ativada (T23)

Com a criação do PSD obtém-se então uma base que facilita a transformação para um sistema de *workflow* assim como uma clarificação sobre as relações entre os papéis de ator e as funções organizacionais.

Segue-se depois a especificação das AR. Nestas, para cada passo relevante de cada transação são especificadas as diversas ações que deverão ser realizadas (e.g., iniciar uma outra transação), quais as restrições que este passo impõe e qual o fluxo que se segue no caso que se verifique (ou não) o cumprimento destas restrições. No entanto é

necessário notar que as AR são mais diretrizes do que regras no sentido mais estrito, pelo que poderá ser acontecer, em certas ocasiões, ser contra produtivo seguir tais regras.

Por fim, é possível criar o OFD. Neste ocorre uma especificação dos tipos de objeto, tipos de facto e tipos de resultado (de transações) e leis de existência. Começa-se por identificar os tipos de facto relevantes, encontrados nas AR. De seguida, adiciona-se todos os tipos de resultados, listados na TRT. Por fim, são adicionadas classes de objetos quando a adição destas é relevante. O OFD acaba por se tornar no ponto inicial ideal para desenvolver e manter o dicionário de dados de uma organização. [5] [6] [7] [8]

Embora existam ainda outros modelos estes não serão descritos neste relatório uma vez que não foram utilizados para a realização da plataforma.

II.2. BPMN

O BPMN é um *standard* que permite modelar processos de negócio e serviços web, proposto pela BPMI. Na sua especificação inclui-se um diagrama de processo de negócio, BPD, que teve dois pontos como foco durante o seu desenvolvimento: facilidade de compreensão e capacidade de modelação de processos de negócio complexos.

A modelação de um processo começa com a identificação dos eventos que ocorrem no início de um processo, as tarefas realizadas e os resultados finais do fluxo de processo. Adicionalmente, um processo pode conter subprocessos, representado noutra diagrama.

É também possível a identificação dos responsáveis por cada tarefa ao inserir os elementos em áreas chamadas de *pools*. É ainda possível subdividir uma *pool* em *lanes*. [4]

II.2.1. Elementos do BPD

Existem seis categorias básicas que categorizam os elementos: participantes, artefactos, *gateways*, dados, atividades e eventos.

Os participantes incluem as *pools* e *lanes*. Uma *pool* habitualmente representa uma empresa, e, a *lane* um departamento dentro dessa empresa. No entanto, uma *pool* pode ser utilizada para representar outras coisas como uma função (algo que a empresa realiza, por exemplo: marketing), uma classe, etc... O essencial é que represente apenas uma

única ‘coisa’, desde que esta coisa venha de uma lista heterogénea de coisas. Uma *lane*, por outro lado, é uma sub-partição dentro de uma *pool* e estende-se pelo comprimento da *pool*.



Figura 4. Pool e duas lanes

Os artefactos incluem anotações de texto e agrupamentos. São, em essência, comentários que aumentam a clareza dos diagramas sem alterar a lógica interna dos diagramas.

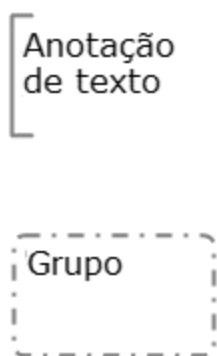


Figura 5. Artefactos

As *gateways* permitem definir separações ou fusões do fluxo de um processo e podem ser de quatro tipos: exclusivos, inclusivos, paralelos e de eventos. As primeiras, exclusivas, ao realizarem uma divisão lançam apenas uma saída; ao realizarem uma fusão apenas um dos fluxos é utilizado. As *gateways* de inclusão são utilizadas para descrever situações onde o fluxo pode divergir para um ou mais (ou todos) os caminhos; ao realizar uma fusão, a *gateway* inclusiva espera por todos os fluxos já iniciados. As *gateways* paralelas iniciam todas as tarefas que dela divergem; ao convergir esta *gateway* aguarda pelo resultado de todos os fluxos sejam completados. Por fim, as *gateways* baseadas em eventos aguardam para que o evento a que ela está afeto ocorra (esta *gateway* é mais próxima da exclusiva).

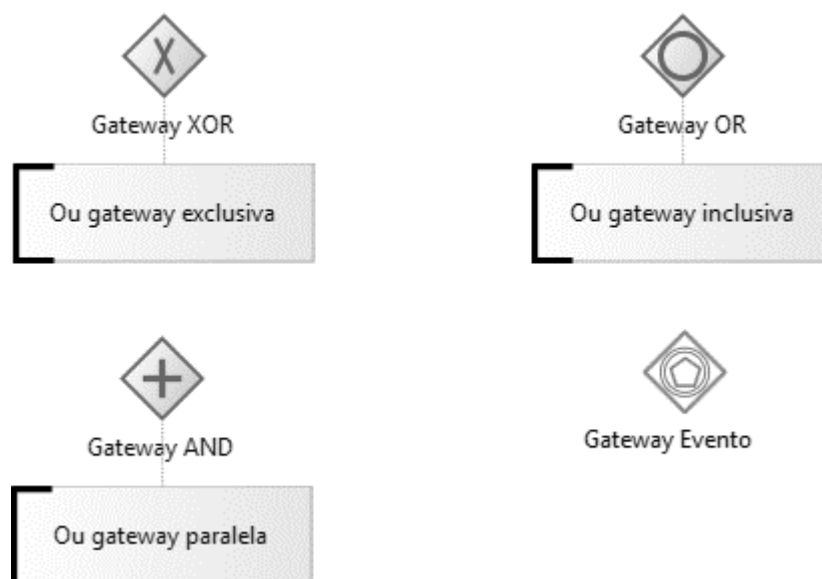


Figura 6. Gateways

Os elementos de dados servem para representar informação e documentos relativos aos processos e tarefas como também os locais de armazenamentos destes.

As atividades podem ser categorizadas nos seguintes grupos: tarefas, subprocessos, atividades de chamada, subprocesso de evento e transação.

Uma tarefa é dos elementos mais comuns no diagrama e pode ser definida por um subtipo de entre um dos seguintes: recepção de tarefa; envio de tarefa; script; serviço; manual; utilizador; regra de negócio. O envio de mensagens é utilizado para chamar serviços web assíncronos. A recepção de mensagens é uma alternativa a apanhar o evento de mensagem. Um script é executado diretamente no *engine* do processo e requer que seja escrito numa linguagem que o *engine* possa interpretar. Uma tarefa do tipo serviço é realizadas por software e embora o BPMN assuma que a função é fornecida por um serviço web, poderá ser utilizado outra implementação. As tarefas manuais são executadas por um ser humano e não afetam a realização da tarefa atribuída pelo *engine* (por exemplo: falar com clientes ao balcão). Uma tarefa de utilizador é semelhante à anterior, no sentido que é realizada por um ser humano, no entanto são atribuídas pelo *engine* e após a sua conclusão é esperado confirmação. Por fim, as tarefas de regra de negócio são aplicadas exclusivamente para aplicar regras de negócio.



Figura 7. Atividades

Por fim, os eventos servem para despoletar um processo (início), ocorrem durante o fluxo de um processo (intermédios) ou terminam o fluxo de um processo (fim). Para além destes três tipos de eventos, estes podem ainda ser categorizados como de captura (*catch*) ou de lançamento (*throw*). Eventos de captura têm um *trigger* definido e assume-se que apenas ocorrem quando este é despoletado. Por outro lado, o BPMN assume que os eventos de lançamento despoletam-se a si mesmos.

Por fim, existe uma terceira categorização para os eventos, que define especificamente o que o evento é. Entre os diversos tipos encontram-se: mensagens, timers, regras, *links*, exceções, cancelamento, início e fim. [9] [10]

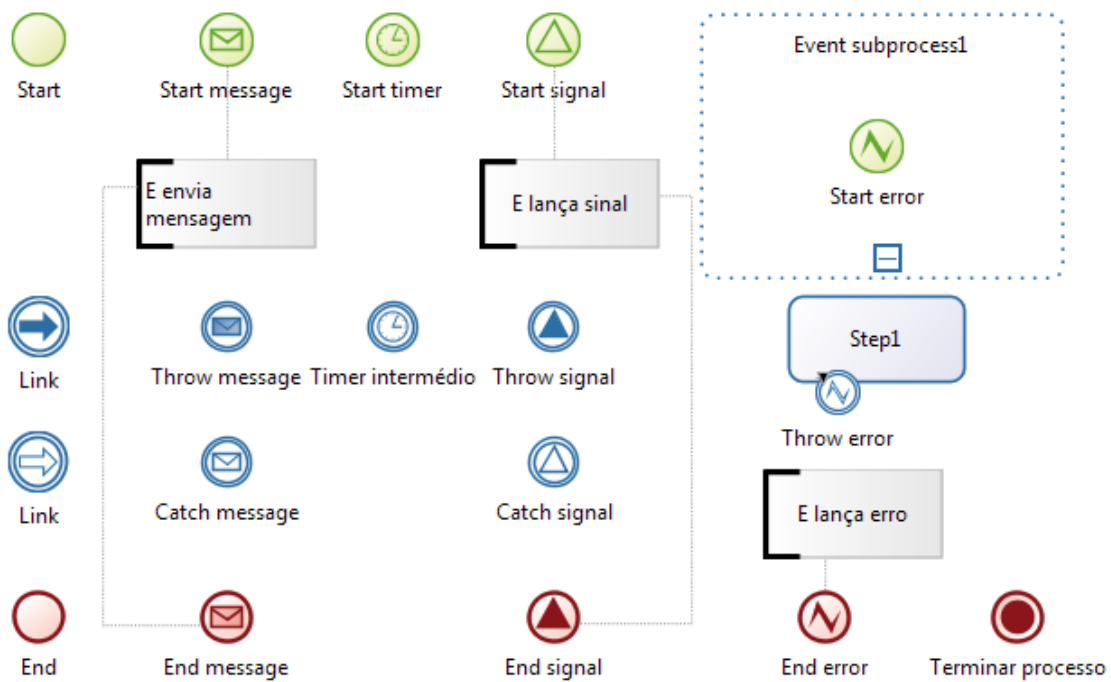


Figura 8. Eventos

II.2.2. Engines de workflow/BPMN

A existência de múltiplos *engines* traz consigo a vantagem de poder escolher o que mais se adequa às necessidades do projeto e que fornece um maior número de funcionalidades ou apoios ao desenvolvedor, por outro lado, com a existência de várias opções é necessário que se faça uma seleção de uma única opção. Descrevem-se de seguida alguns *engines*:

II.2.2.1. Activiti

É uma plataforma de *workflow* e BPM dirigida a pessoas de negócios, desenvolvedores e administradores de sistemas, desenvolvida pela Alfresco. É *open-source* distribuído sobre uma licença Apache, podendo ser executado em qualquer aplicação Java, num servidor, num *cluster* ou em *cloud*. É também de apontar que os criadores desta ferramenta pertenceram à equipa de desenvolvimento do jBPM 3. [10] É também relevante mencionar que entre as três ferramentas, o Activiti é a menos madura, tendo aparecido por último.

O Activiti é composto por 3 componentes principais, responsáveis pela modelação, *runtime* e gestão.

Ao contrário da ferramenta seguinte, o Activiti não possui integração nativa de *Drools* (um sistema de gestão de regras de negócio), pelo que é necessário realizar a integração manualmente [11] [12]

A criação dos diagramas, de forma gráfica, é possível através de duas formas: pela página do Activiti, utilizando um acesso com privilégios de administrador; com um *plugin* para o Eclipse. A utilização deste segundo método é preferencial pois permite a realização de testes antes de se implementar o processo na plataforma. É também relevante referir que os diagramas implementam a grande maioria dos elementos do BPMN 2.0.

O Activiti possui um *engine* de processos Java que executa processos BPMN 2 nativamente. Adicionalmente, é fornecido aos administradores do sistema o estado do *engine*, e a possibilidade de gerir os vários recursos e as definições dos processos, assim como a gestão das tabelas da base de dados e métricas sobre os diversos processos.

Uma vez que os dados têm uma grande importância em qualquer projeto, outro ponto a saber são as bases de dados que são suportadas.

Pelo Activiti, são as seguintes: H2, MySQL, Oracle, PostgreSQL, DB2 e Microsoft SQL Server. Continuando na sequência de *software*, é necessário também o JDK com a versão 6 ou superior.

Quanto ao *hardware*, não foi possível encontrar uma listagem específica relativa aos requisitos mínimos ou recomendados, no entanto, segundo [13] é possível realizar o *benchmark* mencionado com um mínimo de 9MB de RAM, pelo que podemos assumir que os requisitos de hardware são mínimos / desprezíveis.

II.2.2.2. jBPM

O jBPM é uma plataforma de BPM, desenvolvida pela Red Hat, sendo a primeira referência histórica no domínio de ferramentas de *workflow* e BPM *open-source*. É *open-source*, distribuído com uma licença Apache e escrito em Java. O *core* do jBPM é um *engine* de *workflow* Java, que permite a execução de processos de negócio utilizando a especificação BPMN 2.0. Entre as suas funcionalidades encontram-se dois editores, um baseado no Eclipse e outro web, para suportar a criação gráfica dos processos de negócio; *logging* do histórico para monitorização e análise dos processos.

A nível de componentes, o jBPM possui um *core engine* que permite a execução dos processos de negócio, e contém a capacidade para a realização de chamadas de API. Existem também ferramentas baseadas em web que permitem o desenho, simulação e implementação dos processos e outros artefactos relacionados. Com o sistema de gestão, é dada a capacidade aos utilizadores gerirem os seus processos e tarefas e visualizar estatísticas sobre estes. Por fim, existe também extensões do Eclipse (IDE) dirigidas aos *desenvolvedores*, que permitem criar negócio, testar e realizar debugging nos processos. [11] [12] [14]

O jBPM é, em vários pontos, uma ferramenta muito semelhante ao Activiti, sendo que a grande razão é o facto de os fundadores e criadores do Activiti terem sido desenvolvedores do jBPM. No entanto, estas ferramentas não são idênticas. [15]

A nível de requisitos, o jBPM requer, tal como o Activiti, o JDK versão 6 ou superior e necessita também do Ant 1.7 ou superior. Tal como no Activiti, os requisitos de hardware não são claros pela documentação, no entanto, segundo [16] um único *core* de processador, 1 GB de RAM e cerca de 500 MB de espaço em disco é indicado como suficiente; no

entanto é de referir que estes valores referem-se à versão 5 do jBPM que já não é a mais recente.

A nível de base de dados, é suportado DB2, Apache Derby, H2, HyperSQL, MySQL, Oracle, PostgreSQL e Microsoft SQL Server.

II.2.2.3. Bonita BPM

Inicialmente desenvolvido, em 2001, pelo Instituto Francês para a Pesquisa em Informática e na Automação (*Insituit national de recherche en informatique et en automatique*), passando a ser desenvolvida pelo Groupe Bull até em 2009, altura que passou a ser suportado pela Bonitasoft. A primeira versão suportada por esta última foi a versão 5, estando disponível desde 30 de Junho de 2009, no entanto esta versão já não se encontra suportada. Em 2013 ocorre o lançamento da versão 6.0, e mais recentemente, em Junho de 2015, foi lançada a versão 7.0, estando paralelamente suportada a versão 6.5.

Na versão 6, o Bonita BPM Engine é o motor de execução de processos do Bonita. Neste incluem-se funcionalidades que poderão ser utilizadas por desenvolvedores, como a Web REST API, o que dá uma maior capacidade a estes para realizarem operações quando fora do escopo da plataforma.

O Bonita BPM Portal é a parte do Bonita que é visível aos utilizadores finais, que o usam para visualizar tarefas e realizar acções. É também a ferramenta utilizada pelo administrador do processo para instalar, distribuir e gerir processos.

Existem quatro versões diferentes: *Community*, *Teamwork*, *Efficiency* e *Performance*, sendo que apenas a primeira é gratuita e o número de funcionalidades e suporte fornecido pela empresa aumenta à medida que se passa da *Community* para a *Performance*.

Para o desenvolvimento dos processos existe o Bonita BPM Studio, que é um ambiente gráfico para a criação destes. Contém duas grandes ferramentas: o *whiteboard*, para o desenho dos diagramas do fluxo dos processos e para a definição dos detalhes dos passos, transições, pontos de decisão e outros elementos de processo; e o *form builder*, que é usado para criar formulários utilizados nos processos da aplicação web.

Com a passagem para a versão 6 realizaram-se várias alterações, sendo que algumas serão enunciadas:

- Cumprimento mais estrito das regras do BPMN 2;
- Integração com Tomcat no Studio, onde antes não existia;
- Migração entre versões;
- Definição mais simples de conectores e variáveis de processo.

A nível de requisitos de *hardware* é exigido um mínimo de 4 núcleos de CPU, 4GB de memória e 10GB de espaço em disco. Quanto ao *software*, é suportado qualquer sistema operativo que seja compatível com ou o Oracle Java SE JRE 7 ou o OpenJDK 7. É necessário também um servidor de aplicação java compatível com a especificação Java EE 6 (sendo que a documentação lista o Tomcat 7, o JBoss 7.1.1 e o Oracle WebLogic Server 12c). Por fim, relativamente à base de dados relacionais, juntamente com a configuração para a utilização de UTF-8, são suportados o H2, PostgreSQL 9.3, Oracle Database 11Gr2, Microsoft SQL Server 2012 R2 e MySQL 5.5. [2] [12] [13] [17]

II.2.2.4. Conclusão

Uma vez que o BPMN é algo novo e devido a existência de limitações temporais por parte da empresa (com lançamento previsto no último trimestre de 2015) uma exploração a fundo dos *engines* indicados previamente não era algo que parecia ser exequível, pelo que a decisão se baseou primeiramente na leitura de avaliações já existentes. [2] [12] [18] Adicionalmente a estas leituras, foram visualizados alguns vídeos no YouTube para que se pudesse avaliar de uma forma geral quais das ferramentas seriam as mais apropriadas, tendo em conta as várias condicionantes.

Um ponto de grande importância reside na documentação relativa da ferramenta. O Bonita foi considerado como o que mais se destacava por possuir uma boa estrutura de secções e capacidade de pesquisa, sendo que o único problema a apontar é a obrigatoriedade da criação de uma conta para visualizar a documentação. A forma como a documentação do Activiti e do jBPM se encontra é muito semelhante, no entanto, têm uma exposição da informação de uma forma menos amigável ao utilizador.

Um dos problemas mais notáveis com o Activiti era a qualidade dos formulários comparativamente com as outras ferramentas: embora estes estivessem mais interligados com o *dashboard*, tinham um peso na página muito pequeno, dando a impressão que este tinha uma relevância menor.

Pondo este de parte, a decisão pendia entre o jBPM e o Bonita BPM devido à sua maior popularidade (o que resulta numa comunidade maior, onde será possível ter mais utilizadores ativos para o esclarecimento de dúvidas e resolução de problemas), no entanto o primeiro não aparentava ser amigável para utilizadores inexperientes [2], juntamente com uma avaliação mais positiva em todos os pontos para o Bonita BPM [17] levou a que decisão final fosse este último.

	<i>Activiti</i>	<i>jBPM</i>	<i>Bonita</i>
<i>Desenvolvedor</i>	Alfresco	Red Hat	Bonitasoft
<i>Licença</i>	Apache	Apache	GPL
<i>Última versão</i>	5.19.0	6.3.0	7.1.4
	5 Novembro 2015	28 Setembro 2015	3 Dezembro 2015
<i>Requisitos</i>	-	1 core 1 GB RAM 500 MB Disco	4 cores 4 GB RAM 10 GB Disco
<i>Bases de dados suportadas</i>	H2 MySQL Oracle PostgreSQL Microsoft SQL DB2	H2 MySQL Oracle PostgreSQL Microsoft SQL DB2 Apache Derby HyperSQL	H2 MySQL Oracle PostgreSQL Microsoft SQL
<i>Criação de formulários</i>	Eclipse e na web	Eclipse e na web	Adaptação do Eclipse (Bonita Studio)
<i>Importação de processos</i>	Sim	Sim	Sim
<i>Documentação</i>	Existente mas aglomerada numa única página	Extensa e com alguma organização	Organizada e com facilidade de pesquisa
<i>Tecnologia base</i>	Java	Java	Java
<i>Debugging</i>	Plugin do Eclipse	Plugin do Eclipse	A partir do Bonita Studio
<i>Business Rule Engine</i>	Integração com Drools	Integrado com Drools	Engine Próprio

Tabela 1. Quadro resumo dos vários engines de BPM

II.3. DMS

Podemos definir um sistema de gestão de documentos como um *software* que controla e organiza documentos dentro de uma organização. No âmbito do projeto existe a necessidade que se guarde os resultados dos diversos serviços, havendo também uma organização separando os diversos utilizadores e livros destes.

II.3.1. OpenKM

Desenvolvido pela OpenKM, começaram em 2005. Este DMS é escrito em Java sendo *open-source*.

Uma das razões que faz com que o OpenKM seja uma opção atraente é a existência de conectores que fazem a integração com o Bonita BPM, criados pela comunidade, o que implica que a criação de conectores seja desnecessária.

A nível de funcionalidades o OpenKM expõe uma API via SOAP e REST (e ainda um SDK para .NET) que permite integração com outras aplicações. Adicionalmente, existe, entre outras funcionalidades, a capacidade de realizar pesquisas, catalogar com códigos de barras e integração com antivírus que garantem a integridade do repositórios e a segurança dos utilizadores. Por fim, existem sistemas de estatísticas e de relatórios que dão a capacidade ao administrador de controlar melhor a aplicação.

A nível de requisitos, estes variam relativamente ao número de utilizadores, sendo que os mínimos são, para um ambiente onde existem menos de 25 utilizadores e um repositório de até 60GB, um espaço em disco entre 30 GB e 180 GB, 1 GB de memória e 1 core de processador. Para as maiores instalações, onde o número de utilizadores ultrapassa os 300 e o tamanho do repositório ultrapassa os 2TB, é recomendado uma SAN (uma rede de alto desempenho que o propósito principal é permitir que dispositivos de armazenamento comuniquem com sistemas de computadores e entre eles) no mínimo 16 GB de memória e 16 *cores* a 3.6 GHz. No entanto para este projeto o tamanho da instalação deverá ser pequena, pelo que os requisitos desta ferramenta estaria mais próxima dos valores mais baixos. Quanto ao *software*, os sistemas operativos suportados são Windows e Linux; Java versão 1.7 ou superior e Tomcat.

Quanto às versões, o OpenKM oferece três versões: *Community*, *Cloud* e *Professional*, onde apenas a primeira é gratuita. As diferenças incidem maioritariamente em suporte técnico, que se resume ao apoio no fórum da comunidade para a versão *Community* e a inexistência de funcionalidades avançadas como, por exemplo: criptografia, *add-ins* para *software* do Microsoft Office (existem apenas suporte para Word e Outlook), conversor de texto para fala, etc... [19] [20]

II.3.2. Alfresco

De forma semelhante ao OpenKM, teve o seu lançamento inicial em 2005 e desenvolvido Java. Atualmente existem três versões: *Alfresco One*, *Alfresco in the Cloud* e *Alfresco Community Edition*, sendo que apenas a última é gratuita e *open-source* e será a esta que serão feitas referências.

De forma parecida ao sistema anterior (OpenKM), o Alfresco fornece conectores para a comunicação entre ele e o Bonita BPM, no entanto, ao contrário do OpenKM, estes conectores não foram desenvolvidos pela comunidade mas fornecidos automaticamente na versão de desenvolvimento (Studio) do Bonita BPM.

As funcionalidades anunciadas, pela empresa na sua página, incluem integração com aplicações como o Microsoft Office e o Google Docs, acesso nativo ao Alfresco como uma drive, extensibilidade através de APIs comuns e *add-ons* desenvolvidos pela comunidade.

A nível de requisitos, é indicado que, para uma instalação mínima, é recomendado 1 GB de memória (sem haver referência ao número de cores de processador). É também alertado que à medida que o número de utilizadores cresce os recursos dados ao sistema terão de aumentar igualmente, chegando até aos 8 GB de memória e 8 cores de processador para até 200 utilizadores concorrentes. Quanto ao nível de espaço em disco, é apenas sugerido ter tanto espaço quanto aquele que é necessário para os ficheiros que são previstos existir. [21] [22]

II.3.3. ownCloud

A última alternativa apresentada é o ownCloud. Embora não seja apresentado como um DMS, apresenta todas as funcionalidades mais relevantes das duas outras opções que são necessárias no projecto, nomeadamente o armazenamento de documentos e a habilidade de partilha de ficheiros através de *links*.

Uma desvantagem para este sistema é a inexistência de conectores dedicados que interliguem com o Bonita BPM, o que implica que se tenha de ou utilizar conectores genéricos ou realizar o desenvolvimento de um conector dedicado para a ligação entre os sistemas.

Tal como as outras soluções, existem outras versões alternativas (e pagas) à versão Community: a *Standard Subscription* e a *Enterprise Subscription*, cujo maior ponto de diferença consiste em mais apoio.

A nível de funcionalidade o ownCloud declara que permite armazenar documentos, pastas, contactos, fotos, eventos em calendários assim como o acesso através de um dispositivo móvel. Adicionalmente, permite também sincronizar entre vários dispositivos e partilhar os diversos ficheiros e pastas a várias pessoas com ou sem *password* e tempo limite, podendo até ser partilhado publicamente para qualquer pessoa. Por fim, o ownCloud tem também a capacidade de realizar controlo de versões. Para os desenvolvedores, o ownCloud possui suporte para WebDAV e chamadas externas de API (REST)

Quanto aos requisitos, é necessário um mínimo de 128 MB de memória mas é recomendado pelo menos 512 MB, sendo que estes valores poderão ser superiores quando o número de utilizadores é maior. A nível de software é recomendado a utilização do Red Hat Enterprise Linux 7, MySQL ou MariaDB, PHP 5.4 ou superior (sendo que na versão 8.2 já é suportado PHP 7) e Apache 2.4. [23]

II.3.4. Conclusão

Inicialmente começou-se por experimentar o OpenKM pois parecia ser a escolha superior, visto possuir conectores já criados pela comunidade para o Bonita BPM e aparentar cumprir com as necessidades. No entanto, após alguns dias de testes e utilização detetou-se que existiam várias falhas nos conectores, não sendo possível manter um estado estável na plataforma (falhavam após algumas utilizações, exigindo que se reiniciasse o Tomcat). Juntamente a estes problemas, adiciona-se uma documentação fraca (existia uma wiki para o sistema, no entanto na altura encontrava-se em fase de reconstrução e possuía lacunas). Considerando todos estes pontos esta opção foi desconsiderada.

Uma vez retirado a primeira opção, foi necessário escolher um novo candidato. Entre o Alfresco e ownCloud acabou-se por escolher o último,

em grande parte pela recomendação do co-orientador que se evitasse o primeiro devido a problemas encontrados numa altura prévia por ele.

O ownCloud, como mencionado previamente não é tecnicamente um DMS, ou pelo menos não se categoriza como um, no entanto consegue cumprir com as necessidades do projecto (armazenar ficheiros, organizá-los de alguma forma e permitir a partilha aos utilizadores finais dos seus ficheiros). O problema da utilização desta plataforma passou então por encontrar uma maneira de permitir a transferência dos ficheiros de uma forma que seja o mais transparente possível aos utilizadores finais.

É também de referir que o ownCloud possui uma comunidade relativamente ativa, assim como uma documentação decente que permitiu uma instalação, sem a existência de problema algum, na máquina.

	<i>OpenKM</i>	<i>Alfresco</i>	<i>ownCloud</i>
<i>Desenvolvedor</i>	OpenKM	Alfresco Software, Inc.	OwnCloud, Inc.
<i>Licença</i>	GNU GPL	LGPL v3	AGPL v3
<i>Última versão</i>	6.3.1 22 Outubro 2015	5.0.d 23 Março 2015	8.2.1 17 Novembro 2015
<i>Requisitos recomendados</i>	30 GB Disco 1 core 1 GB RAM Java 1.7+ Tomcat 7	Sistema 64 bit 2 cores (2 GHz+) 2 GB RAM JDK 8 MySQL 5.6 Tomcat 7 LibreOffice 4.2.5	512 MB RAM Red Hat Enterprise Linux 7 MySQL/MariaDB PHP 5.4 Apache 2.4
<i>Tecnologia base</i>	Java	Java	PHP
<i>Documentação</i>	Existente mas desorganizada e com lacunas	Existente	Existente
<i>Conectores com Bonita BPM</i>	Desenvolvidos pela comunidade	Já fornecidos na instalação	Não existem conectores dedicados
<i>API</i>	CMIS Webservices REST SDK	CMIS Webservices REST	REST OCS

<i>WebDAV</i>	Sim	Sim	Sim
<i>Partilha de ficheiros</i>	Possível mas não previsto	Possível	Com ou sem password e tempo limite

Tabela 2. Quadro resumo dos DMS

III. Especificação e Implementação

III.1. Requisitos e funcionalidades

Numa fase inicial os requisitos enunciados foram os seguintes:

1. O sistema terá que permitir o registo autónomo de novos utilizadores, suportado à posterior por documentação contratual;
2. O sistema terá que permitir o registo de novas intenções de produção de obras em formato digital (ebook), devidamente preenchidas e acompanhadas dos materiais necessários (onde obrigatoriamente um título inicial terá que ser fornecido assim como o texto);
3. O sistema deverá permitir a adição de novas funcionalidades / necessidades que visam complementar a produção da obra literária em formato digital no início ou durante o processo de produção;
4. O sistema deverá permitir submeter obras literárias em formatos digitais passíveis de conversão, para os formatos atualmente suportados na distribuição digital das obras literárias;
5. O sistema deverá permitir a indicação do envio de obras literárias em formato impresso aquando da inexistência de um formato digital adequado passível de conversão;
6. O sistema terá que permitir o controlo, por parte do utilizador, das várias etapas de conversão necessárias para a produção dos diferentes formatos digitais selecionados, bem como de eventuais etapas adicionais indicadas anteriormente pelo utilizador no processo de registo da nova intenção de produção;
7. O sistema necessitará de permitir o controlo dos custos efetivos associados à produção de uma ou mais obras em formato digital;
8. O sistema deverá permitir a transferência da obra literária finalizada e disponível nos seus vários formatos;
9. O sistema deverá permitir a entrada no circuito de distribuição digital através dos canais suportados pela Lernasnuvens, mediante solicitação realizada no início, durante ou no final do processo de produção da obra literária nos seus vários formatos digitais;
10. O sistema permitirá controlar os custos e dividendos associados à distribuição da obra literária nos seus vários formatos digitais;
11. O sistema deverá possibilitar o cancelamento / saída dos canais de distribuição digitais suportados pela Lernasnuvens,

devidamente justificada e respeitando as cláusulas contratualizadas com os diferentes canais.

12. O sistema deverá permitir ao cliente a contratação de serviços de marketing para cada obra.

III.2. Metodologia

Seguindo a metodologia DEMO, o primeiro passo na realização do projeto passou pela realização de uma análise PIF e CAP para identificar quais as transações presentes.

Para este efeito foi pedido ao co-orientador, que é também o intermediário com a empresa, que criasse um documento contendo um aglomerado de possíveis ações realizáveis na plataforma, contendo o máximo de passos e processos possíveis.

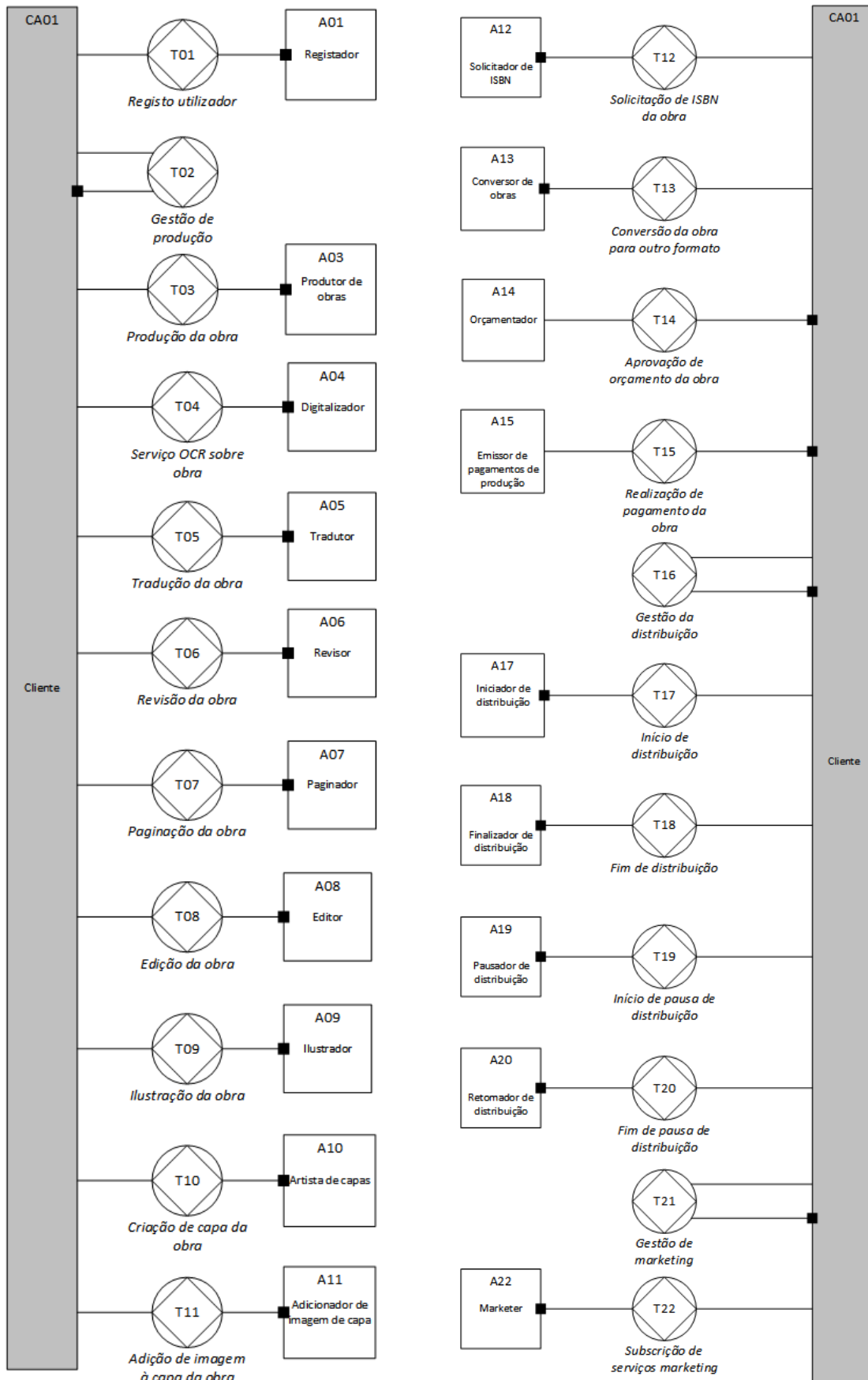
A elaboração das análises PIF e CAP passaram por algumas dificuldades, nomeadamente devido a uma falha de comunicação entre as partes envolvidas onde não ficou bem claro os requisitos do documento necessário para a realização das análises mencionadas previamente. De forma a complementar esta lacuna foram realizadas várias reuniões com o co-orientador de forma a complementar a informação obtida pelas análises, passando-se diretamente à realização da TRT.

ID Transação	Nome Transação	ID Iniciador	Nome Iniciador	ID Executor	Nome Executor
T01	Registo utilizador	CA01	Cliente	A01	Registador
T02	Gestão de produção	CA01	Cliente	CA01	Cliente
T03	Produção da obra	CA01	Cliente	A03	Produtor de obras
T04	Serviço OCR sobre obra	CA01	Cliente	A04	Digitalizador
T05	Tradução da obra	CA01	Cliente	A05	Tradutor
T06	Revisão da obra	CA01	Cliente	A06	Revisor
T07	Paginação da obra	CA01	Cliente	A07	Paginador
T08	Edição da obra	CA01	Cliente	A08	Editor
T09	Ilustração da obra	CA01	Cliente	A09	Ilustrador
T10	Criação de capa da obra	CA01	Cliente	A10	Artista de capas
T11	Adição de imagem à capa da obra	CA01	Cliente	A11	Adicionador de imagem de capa

T12	Solicitação de ISBN da obra	CA01	Cliente	A12	Solicitador de ISBN
T13	Conversão da obra para outro formato	CA01	Cliente	A13	Conversor de obras
T14	Aprovação de orçamento da obra	A14	Orçamentador	A14	Cliente
T15	Realização de pagamento da obra	A15	Emissor de pagamentos de produção	A15	Cliente
T16	Gestão da distribuição	CA01	Cliente	CA01	Cliente
T17	Início de distribuição	CA01	Cliente	A17	Iniciador de distribuição
T18	Fim de distribuição	CA01	Cliente	A18	Finalizador de distribuição
T19	Início de pausa de distribuição	CA01	Cliente	A19	Pausador de distribuição
T20	Fim de pausa de distribuição	CA01	Cliente	A20	Retomador de distribuição
T21	Gestão de marketing	CA01	Cliente	A21	Gestor de marketing
T22	Subscrição de serviços marketing	CA01	Cliente	A22	Marketer
T23	Gestão de recursos humanos	A23	Gestor de RH	A23	Gestor de RH
T24	Especificação de funcionário	A23	Gestor de RH	A24	Especificador
T25	Atribuição de funcionário	A23	Gestor de RH	A25	Atribuidor

Tabela 3. Versão final da TRT

Após iterar sobre a TRT, corrigindo falhas e preenchendo lacunas, foi obtido uma base suficientemente forte para continuar para o passo seguinte: a criação do ATD, onde se passou de uma representação da informação em forma de tabela para uma representação em forma de diagrama, transpondo a informação relativa às transações e atores, tanto iniciadores como executores para este.



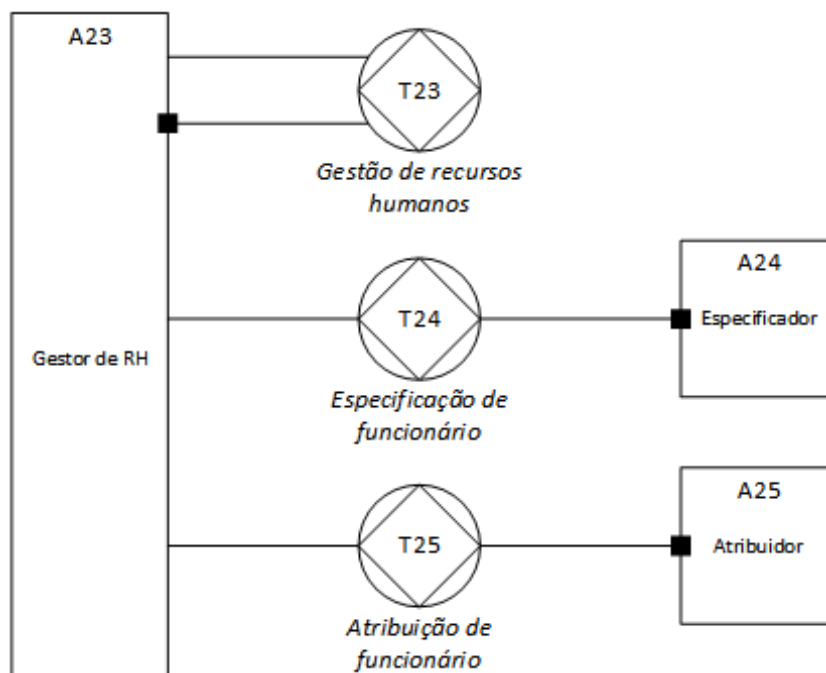
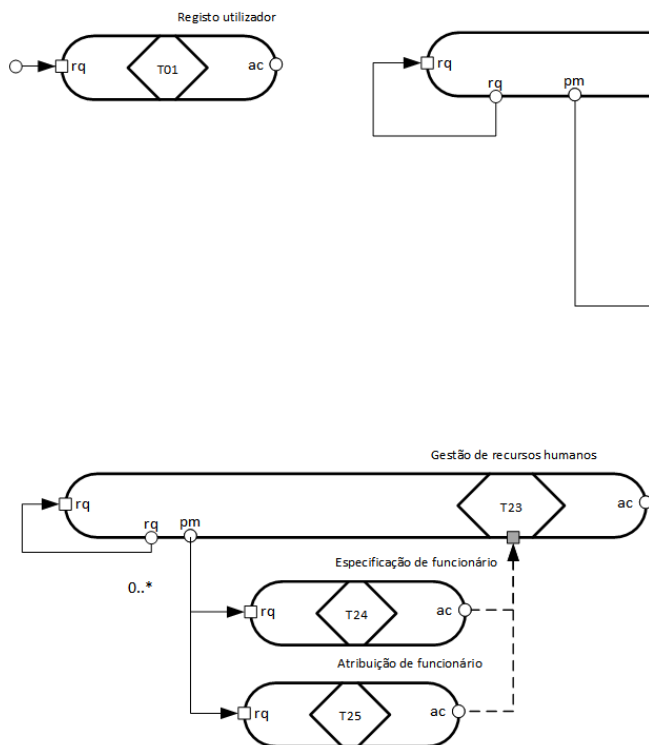
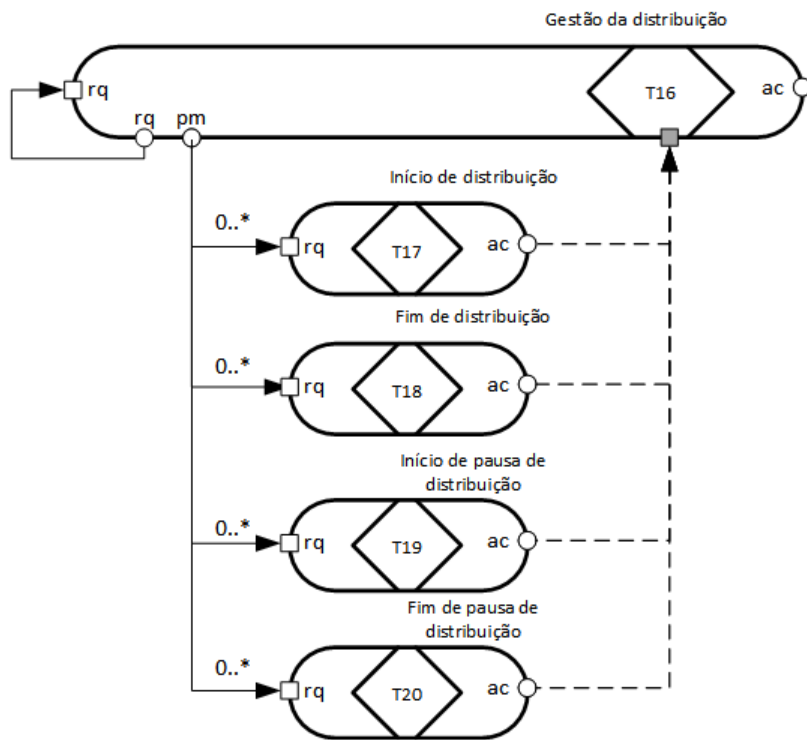
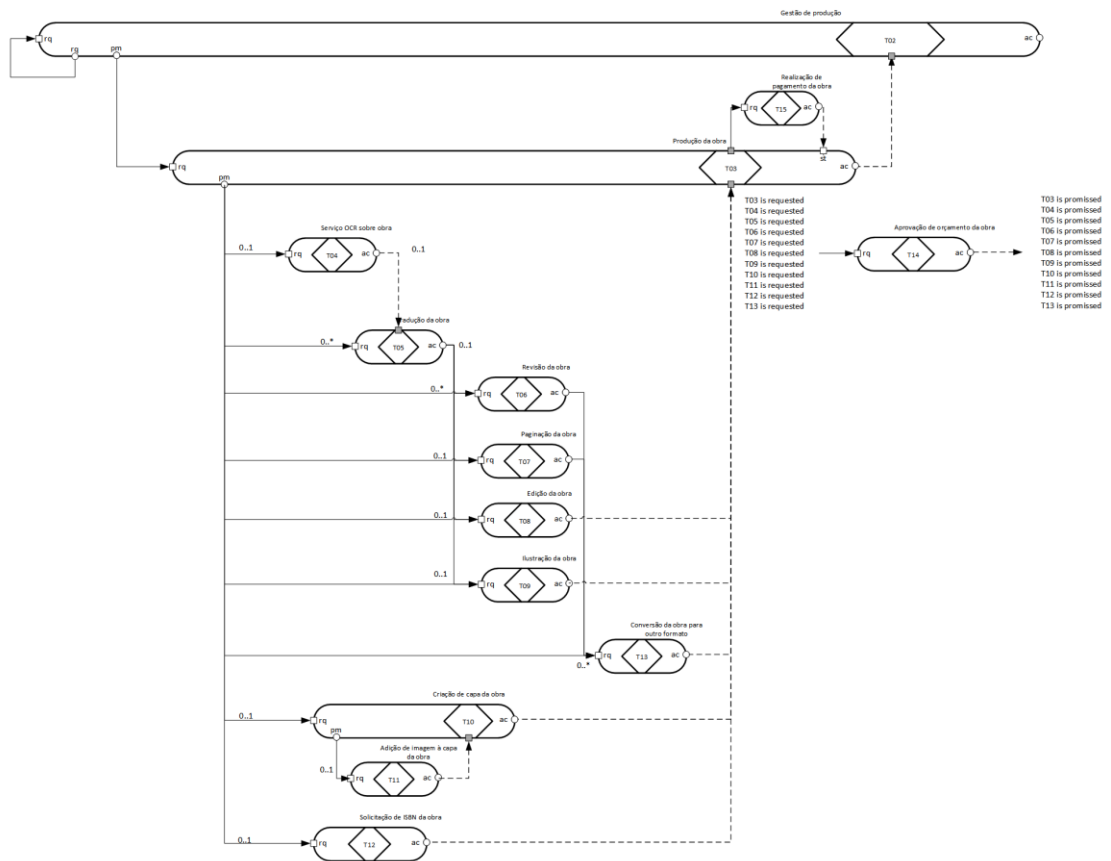


Figura 9. Versão final do ATD

Continuando com o método do DEMO continuou-se com a criação do PSD. Para a produção deste diagrama foi reutilizada novamente a informação contida no ATD, pertencente ao TRT e em informação até agora não utilizada, nomeadamente em restrições temporais e limitações a nível de unicidade, que foi obtida durante as reuniões.





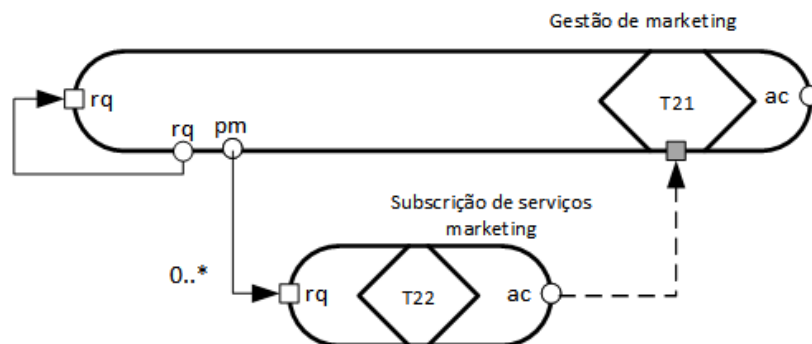
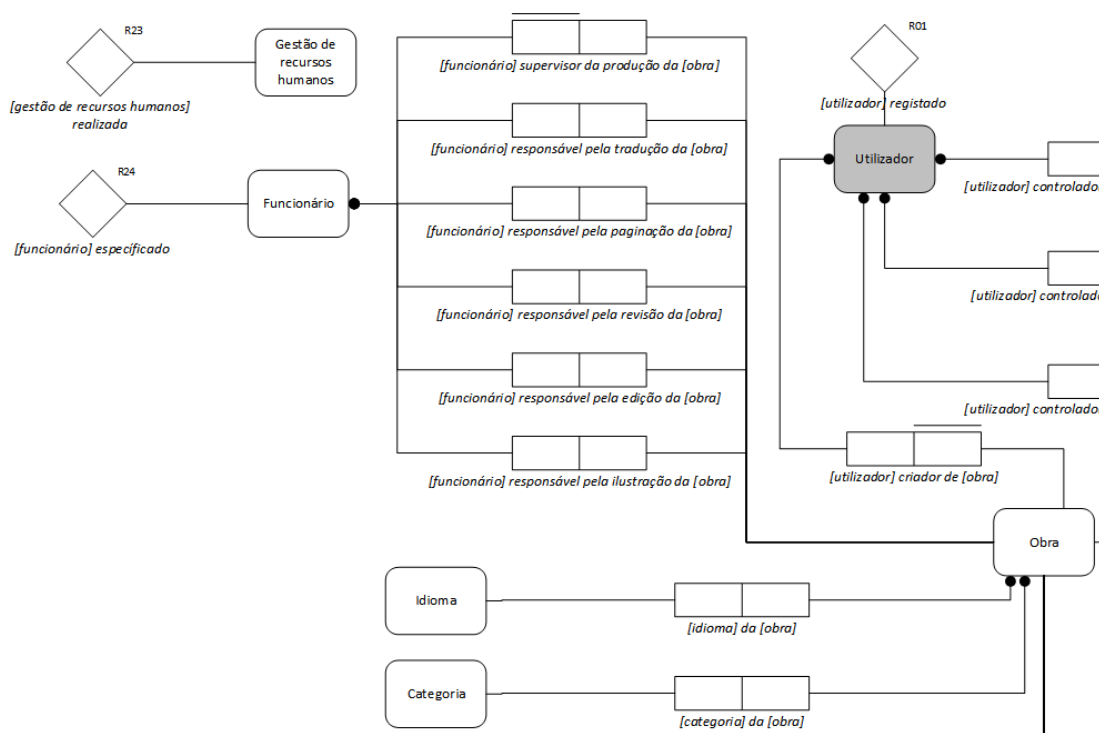
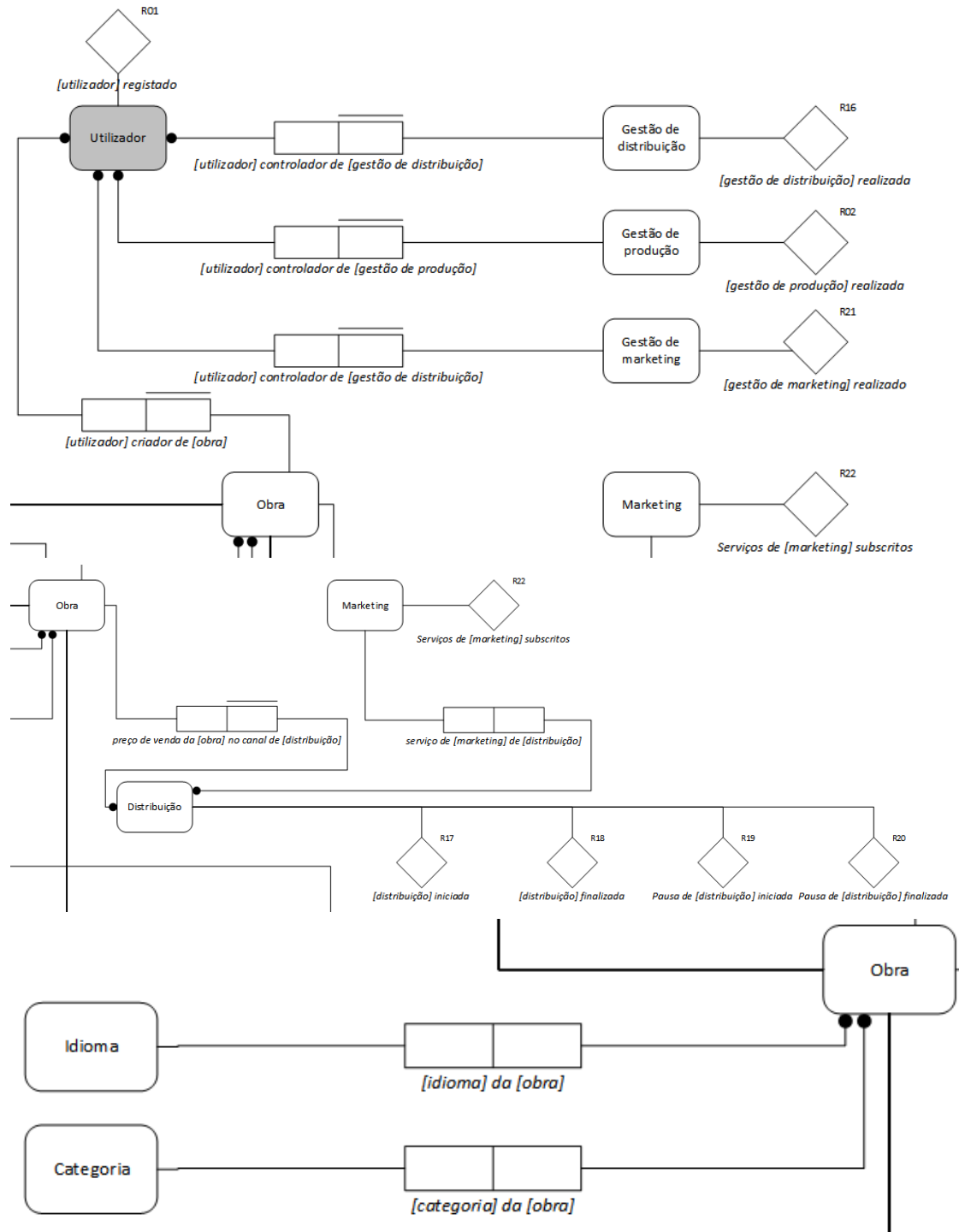


Figura 10. Versão final do PSD

Dando seguimento à metodologia continuou-se o trabalho com a criação do OFD. No entanto, após várias iterações sobre os diagramas prévios notou-se que haviam lacunas pelo que tiveram de ser corrigidas e uma vez que a informação encontrava-se interligada estas correções necessitaram de ser propagadas pelos outros modelos, o que obrigou à realização de mais iterações sobre os últimos modelos.





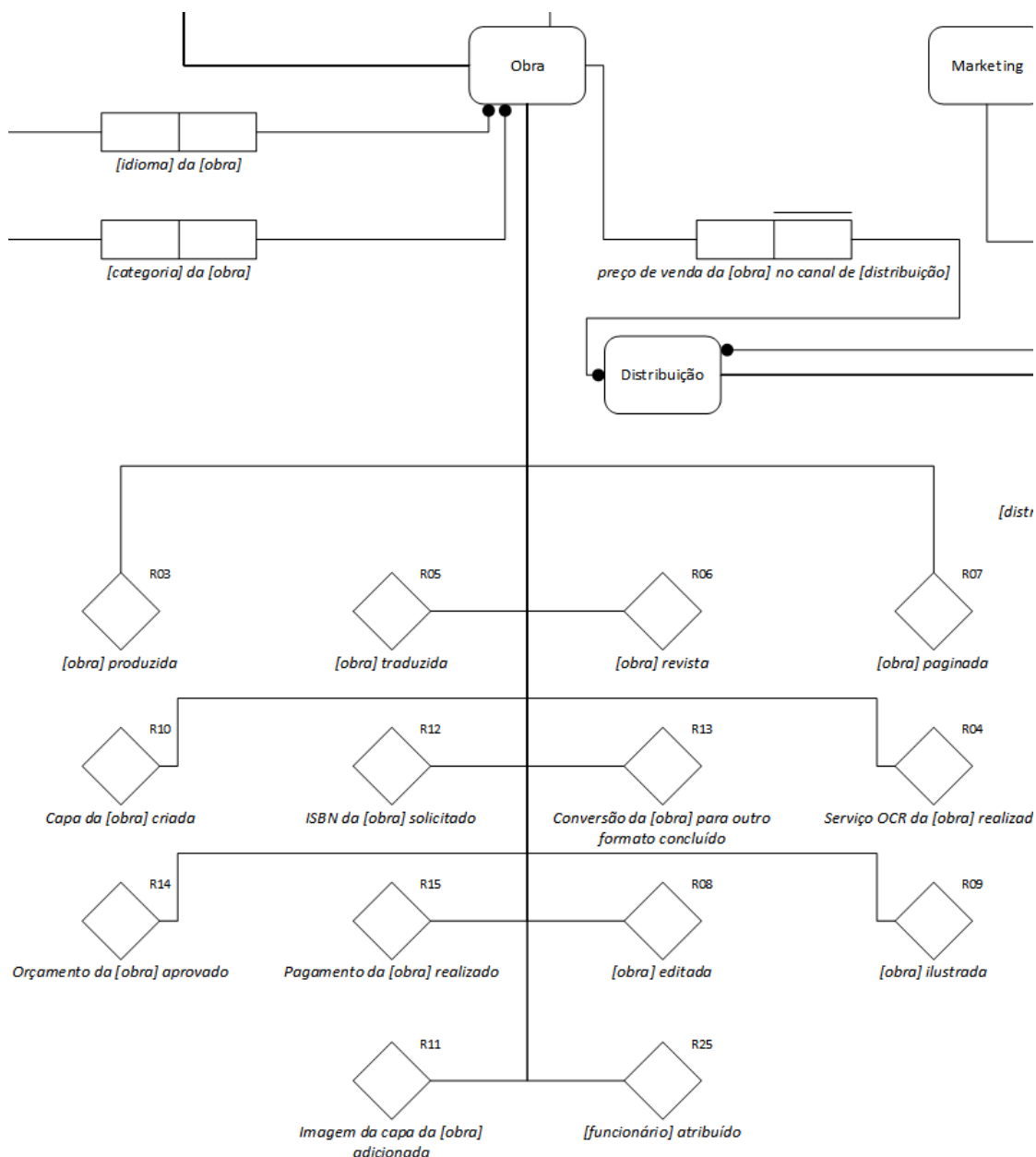


Figura 11. Versão final do OFD

Após uma verificação final dos vários modelos, juntamente com os dados obtidos de reuniões adicionais para solidificar a informação, chegaram-se aos modelos finais. Tanto os diagramas iniciais como os finais podem ser vistos na secção de anexos, no fim do relatório.

Tendo como base os diagramas criados até ao momento, que se encontram estáveis no que toca a possíveis alterações resultantes de futuras iterações, deu-se início à criação da base de dados.

Começou-se pela criação de uma tabela por cada classe existente no OFD. Continuando com a criação de tabelas, para cada tipo de resultado

associado a uma classe, onde existe mais que do um único tipo de resultado criou-se uma tabela e por fim nas relações n:m, definidas como tipos de facto no OFD criaram-se tabelas n:m no Workbench. Tendo esta base notou-se que existiam redundâncias, especialmente nos vários tipos de resultado associados com os serviços da obra, pelo que se decidiu agrupar todas estas tabelas numa que representasse os vários serviços (uma espécie de meta tabela que definia os serviços), ligada com a tabela do livro através de uma relação n:m.

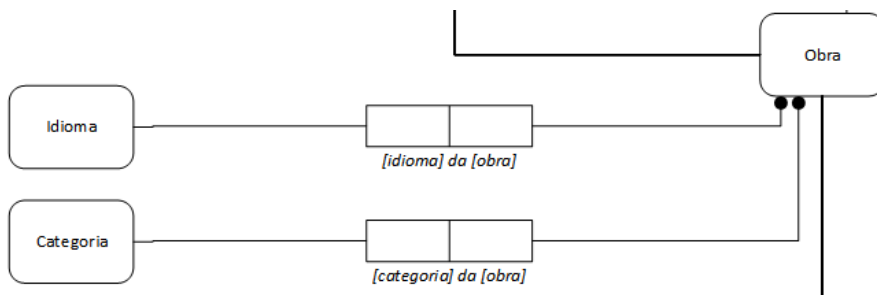


Figura 12. Tipo de facto categoria da obra e classes associadas - Modelo OFD

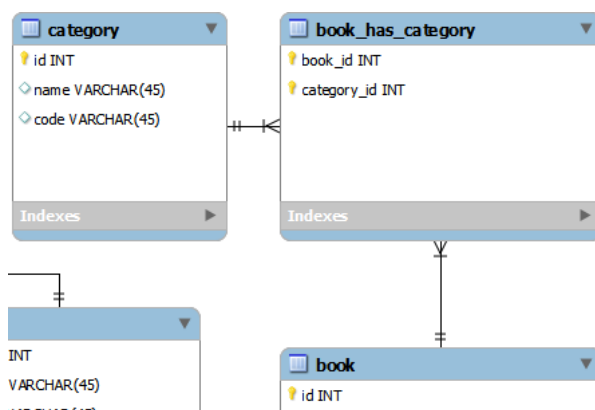


Figura 13. Tabelas resultantes do tipo de facto e classes - Modelo relacional

Depois de alguma utilização da base de dados notou-se que certas tabelas continham um número excessivo de chaves primárias, pelo que foram realizadas alterações de modo a corrigir esta situação. Uma outra alteração realizada, em grande parte devido ao DEMO, foi a criação de uma tabela para o registo dos eventos, sendo que todas as tabelas que necessitassem de registar as diferentes etapas teriam uma chave estrangeira para ligar a esta nova tabela.

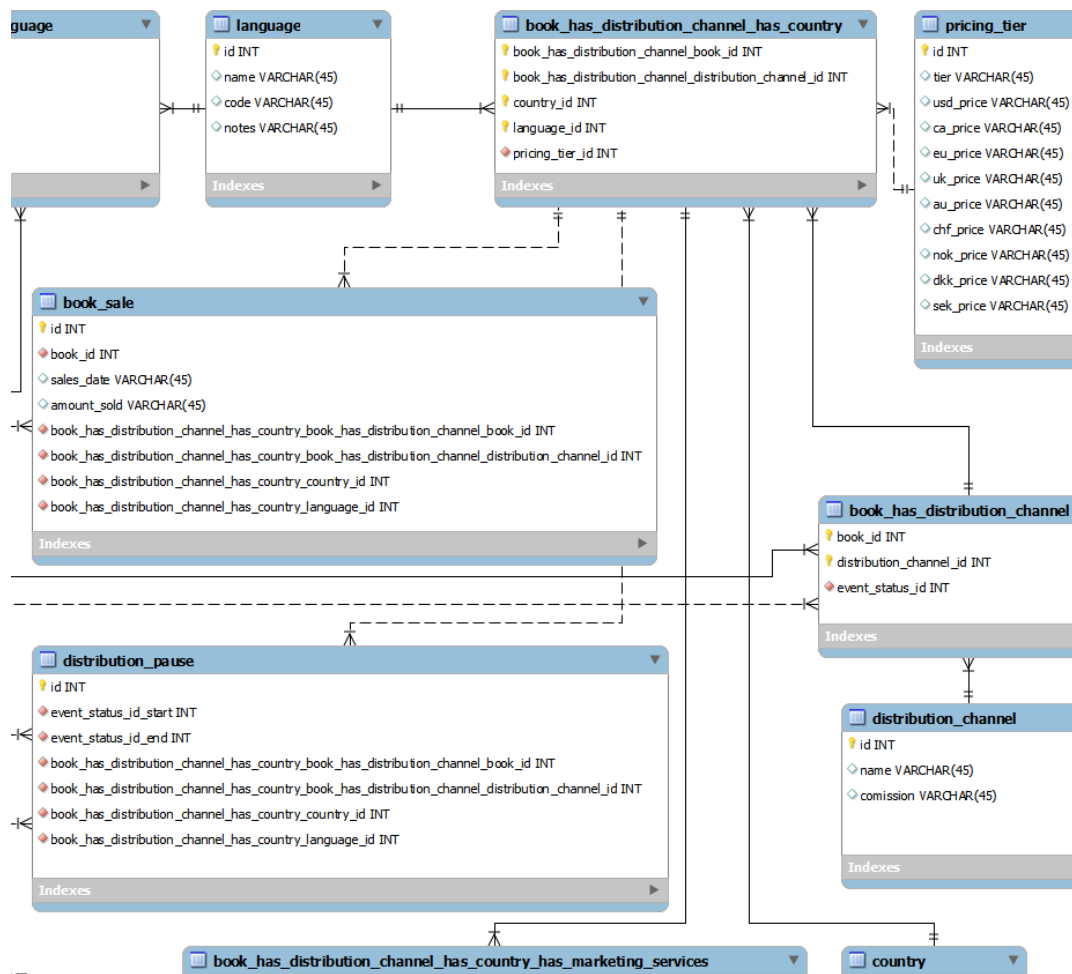


Figura 14. Tabelas antigas para distribuição

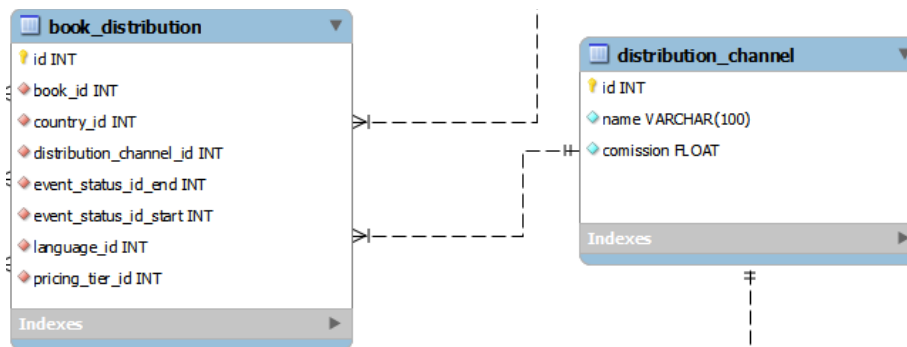


Figura 15. Novas tabelas para distribuição

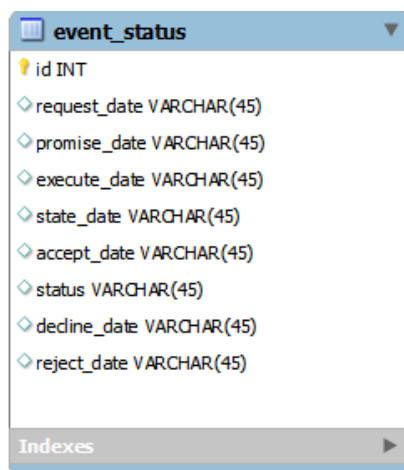


Figura 16. Tabela para registo de eventos

Alterou-se também a tabela que regista escalões de preços, criando-se três: uma para guardar as diferentes moedas, outra para os diferentes escalões e outra para o valor de cada moeda para cada escalão, isto permite uma maior escalabilidade e facilidade de modificação.

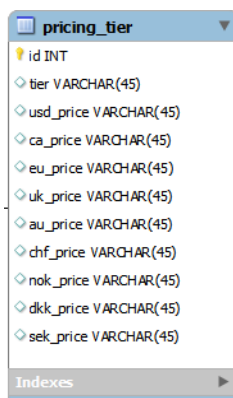


Figura 17. Tabela antiga para valores de escalão



Figura 18. Novas tabelas para guardar valores de escalões de preços

Por fim, foi removida a aceitação de orçamentação para cada serviço em específico e optou-se por um orçamento por livro para que se pudesse fornecer uma melhor experiência de utilização ao utilizador.

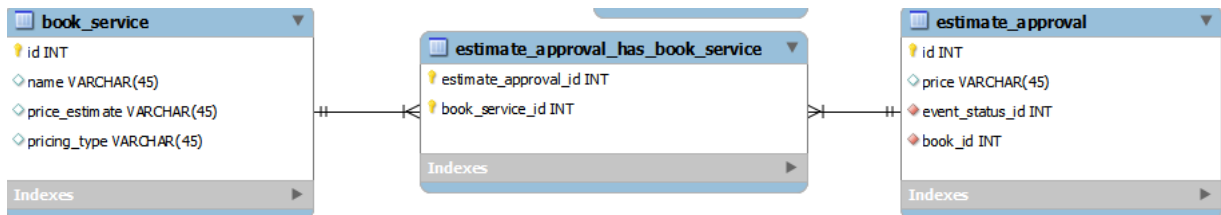


Figura 19. Tabelas com orçamentação ligada a um único serviço

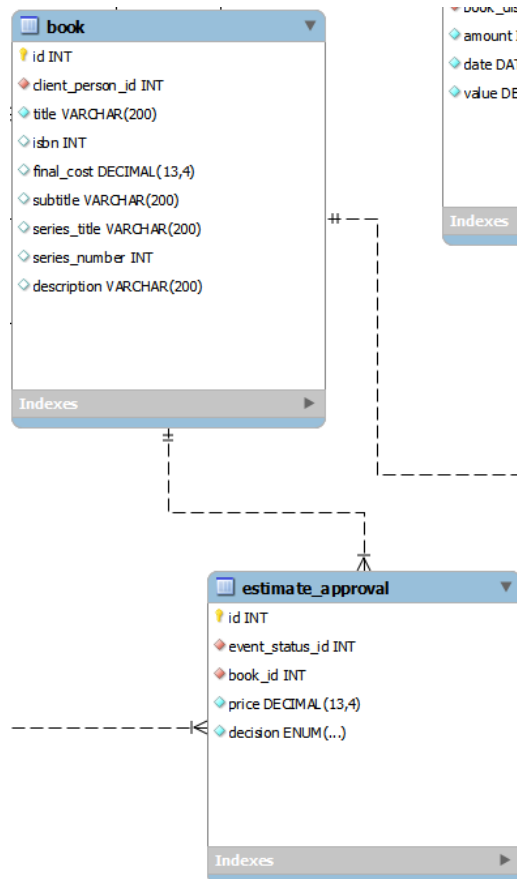


Figura 20. Passagem a uma entrada de orçamento por livro

Para que se pudesse registrar as execuções de processos de gestão (e ter a informação que indicasse que tipo de gestão foi realizada) foi criada uma tabela para o efeito (management).

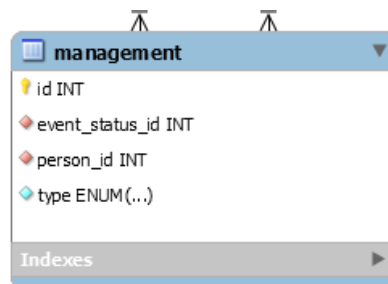


Figura 21. Tabela para registo de ações de gestão

Com o desenvolvimento do componente de distribuição e marketing foram descobertas algumas falhas e necessidades como a falta de uma tabela com todos os países, subdividido por regiões; a alteração da tabela que regista os serviços de marketing subscritos por cada distribuição (onde foi necessário incluir uma chave estrangeira para registar o início e fim de distribuição); ou a transformação da chave estrangeira que marca o fim/paragem completa da distribuição de um livro para uma que aceite ser NULL.

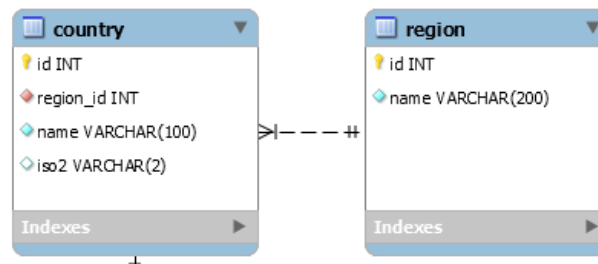


Figura 22. Tabela para registo de países e regiões

Por fim, embora tinha sido alterado para que um funcionário possa ter vários cargos organizacionais, pela definição fornecida pela empresa, vários cargos podem executar a mesma função, o que exige a existência de uma tabela que indicasse quais os cargos que possuem autoridade para realizar os diversos serviços acessórios.

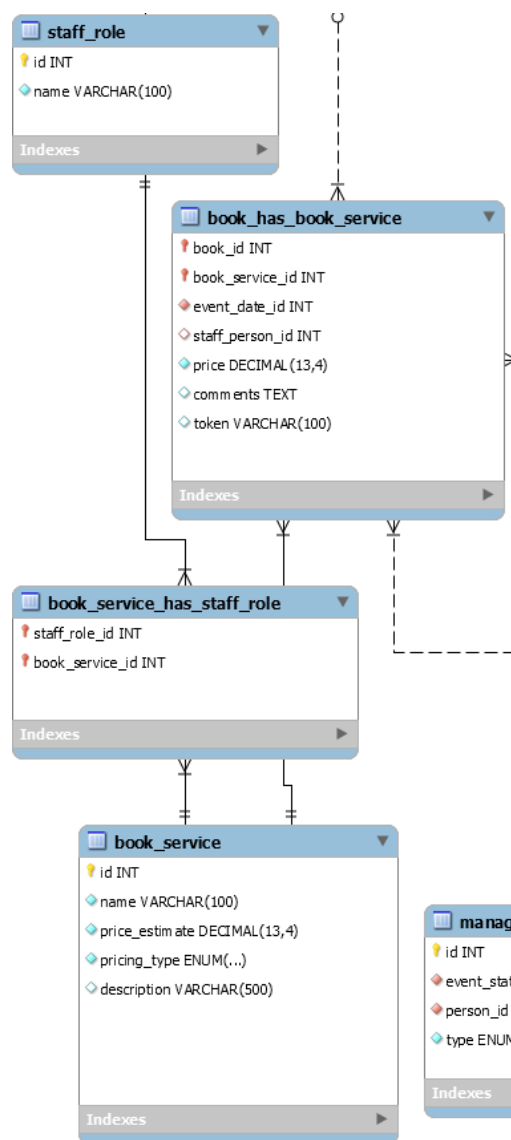


Figura 23. Novas tabelas para registo das funções dos funcionários

Numa altura posterior foram obtidas novas informações relativas ao marketing e os seus processos. Inicialmente foi assumido que os serviços de marketing estariam ligados a uma entrada de país, canal, idioma e livro: por outras palavras, para um livro que estivesse a ser distribuído em dois canais diferentes em vinte países diferentes, utilizando apenas um único idioma, um serviço teria de ser comprado 40 vezes (vinte por cada canal). No entanto foi esclarecido que tal não se iria verificar, sendo que o processo correto seria a contratação de serviços baseada apenas no livro, sendo independente do número de canais e de países.

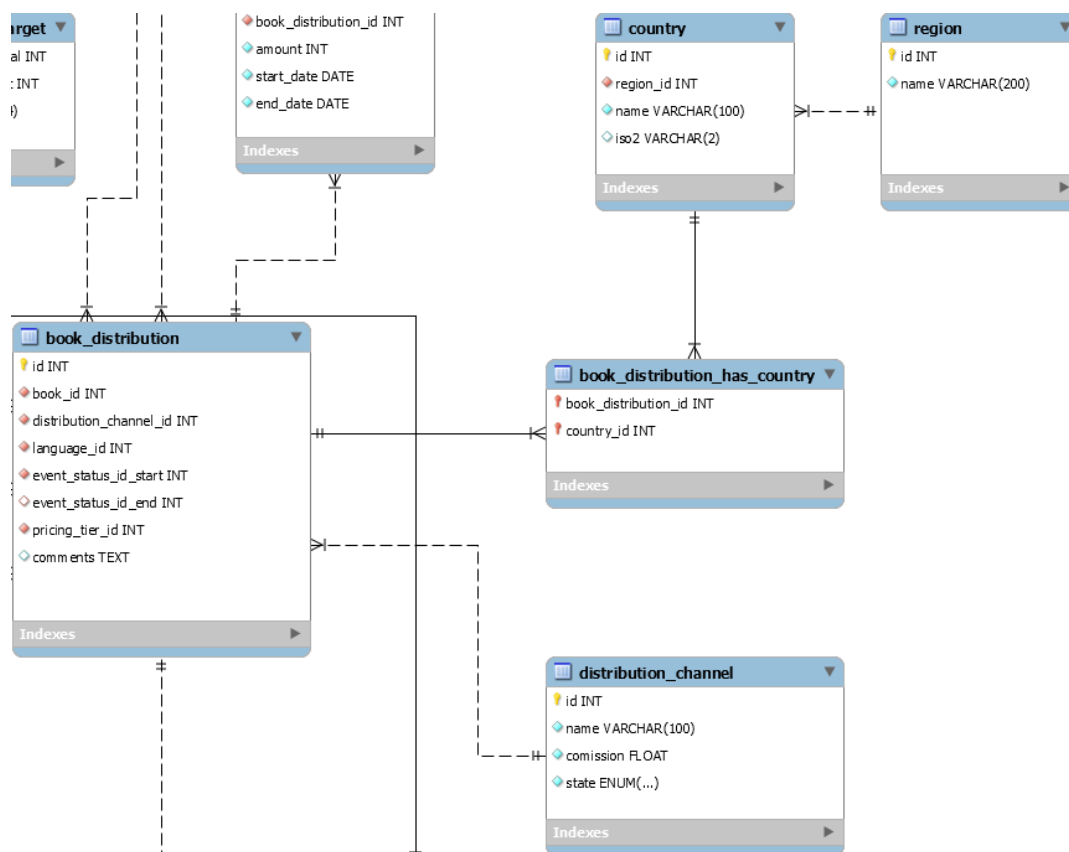


Figura 24. Nova estrutura para distribuição de livros

Devido ao tamanho destas alterações procedeu-se a novas alterações da base de dados, removendo a tabela que representava a relação n:m entre serviços de marketing e distribuições individuais, e substituiu-se esta por uma ligação n:m entre o livro e os serviços de marketing, para refletir a forma como será tratado o marketing.

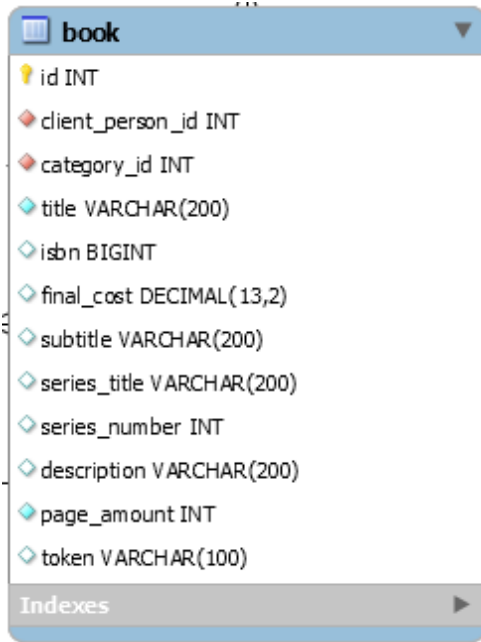


Figura 26. Tabela 'book' antiga

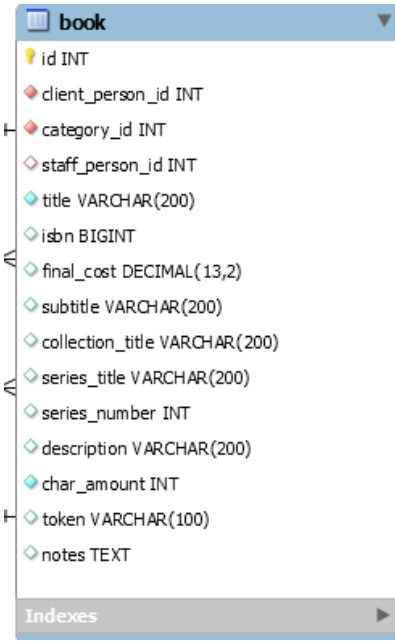


Figura 27. Tabela 'book' com novas alterações

Após as primeiras iterações sobre a base de dados achou-se pertinente começar com a elaboração dos BPD.

Seguindo as *guidelines* mencionadas em [2], começou-se por criar uma *pool* por cada transação identificada nos modelos DEMO, num total de 25. Em cada uma destas *pools*, foram adicionadas uma *lane* por cada ator distinto a ela associada, por outras palavras, para as transações auto-ativadas apenas uma *lane* é adicionada visto que o iniciador e executor são o mesmo ator, e, nos outros casos, duas *lanes* são adicionadas.

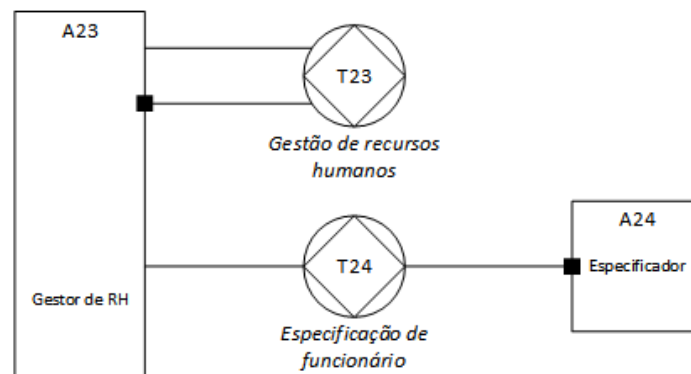


Figura 28. Exemplo de transações

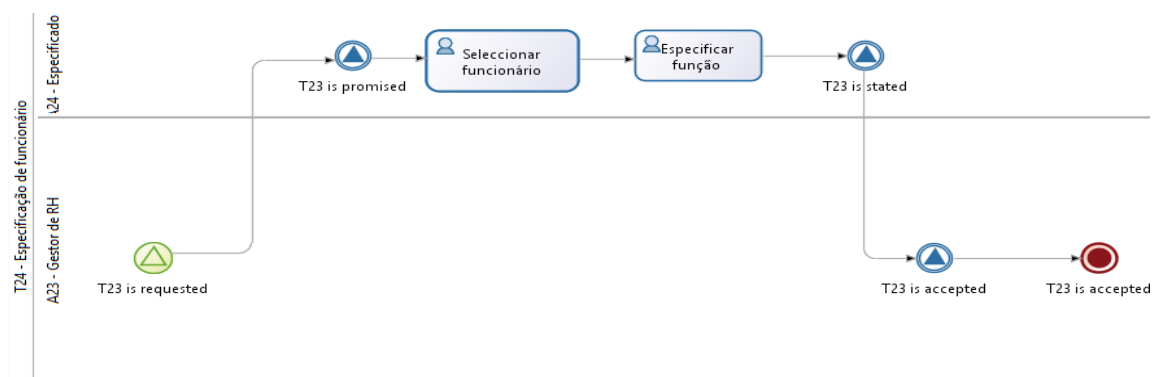


Figura 29. Passagem para diagrama BPD da transação T23

Foi adicionado também tarefas relativas à chamada de processos, sinais para a representação de C-facts e P-facts e *gateways* quando necessário para representar alterações no fluxo de sequência. No entanto é de notar que embora os diagramas criados até este ponto assistiram na construção do BPD, certos passos e ações não estavam representados (como buscas à base de dados ou atualizações de campos), sendo acrescentados quando relevante.

Para que se possa representar a passagem pelo padrão da transação (desde o *request* até ao *accept*) foram utilizados *throw signals* (representados por círculos com um triângulo no seu interior), como ilustrado na imagem acima, no entanto a adição destes nos diagramas adicionavam apenas complexidade adicional sem trazer benefícios, pelo que se removeram estes elementos dos diagramas, mantendo no entanto a realização de atualizações nos registos relevante da base de dados.

Quanto aos *gateways*, o Bonita permite um controlo simples do fluxo (que o DEMO assiste, devido à existência de processos com apenas 2 atores e uma representação simples mas completa dos processos) sem a utilização de *gateways*; acrescentando a isto várias dificuldades na utilização de *gateways*, mesmo em processos simples, acabou-se pela remoção das *gateways* em todos os processos e a utilização de transições (ligações através de setas, que permitem também a utilização de regras condicionais).

Como mencionado previamente, seguiram-se as *guidelines* descritas em [2], o que resultou na implementação de 25 *pools*. No entanto, embora em teoria estivesse correto a utilização desta metodologia implicaria a replicação de código e a realização da mesma alteração múltiplas vezes, quando pertinente. Outro problema também presente, é uma pior experiência de utilização por parte do utilizador final, uma vez que devido

à forma como o Bonita foi desenvolvido, é necessário a realização de cliques para a passagem entre tarefas pertencentes a processos diferentes, cliques estes que são mais facilmente de tornar invisíveis ao utilizador quando as tarefas pertencem ao mesmo processo.

Tendo estes aspetos referidos em consideração, procedeu-se a uma simplificação dos diagramas, sacrificando-se o seguimento estrito das regras para se obter uma maior facilidade de desenvolvimento e uma melhor experiência de utilização por parte do utilizador.

Mais especificamente, os processos relativos à realização de um serviço acessório à produção de ebook foram todos aglutinados num único processo, embora este novo processo possua um tamanho e complexidade superior quando comparado ao processo antigo de realização de serviços acessórios, quando comparado ao somatório de todos os processos os ganhos são claros. Outra alteração foi a incorporação de processos auxiliares como o pagamento e a decisão sobre o orçamento no processo pai que realiza a chamada para o início destes subprocessos; neste caso a motivação não era a redução ou simplificação dos processos mas a melhoria da experiência de utilização do utilizador, devido a maior fluidez que existe no Bonita durante a passagem de tarefas que estão contidas no mesmo processo quando comparadas com a passagem de tarefas entre processos diferentes.

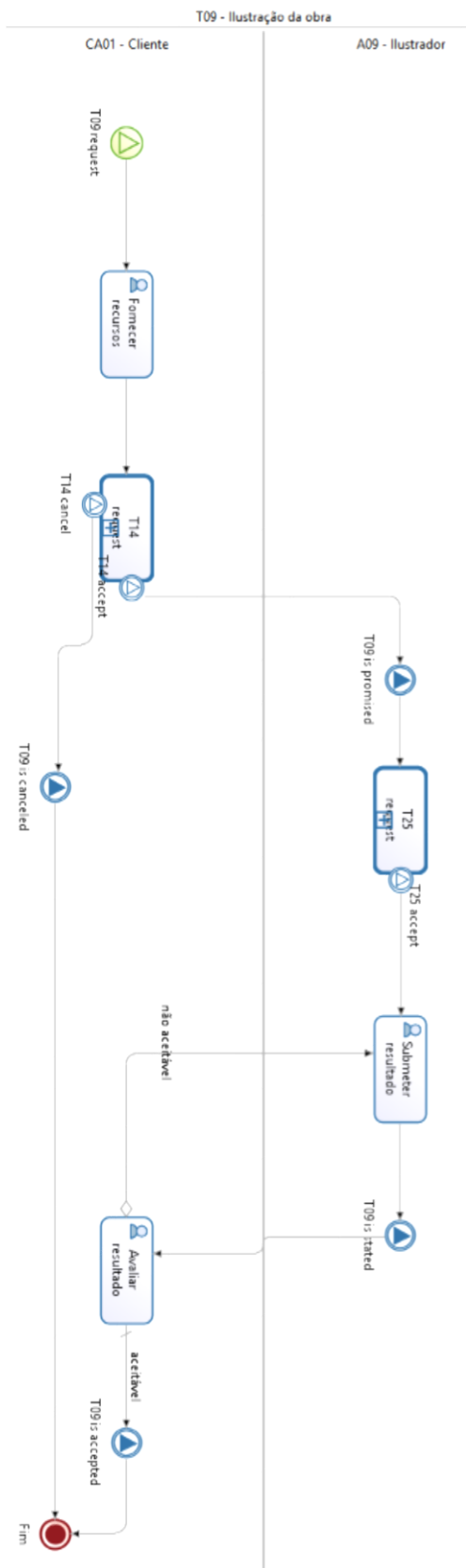


Figura 30. Exemplo de um processo para um serviço

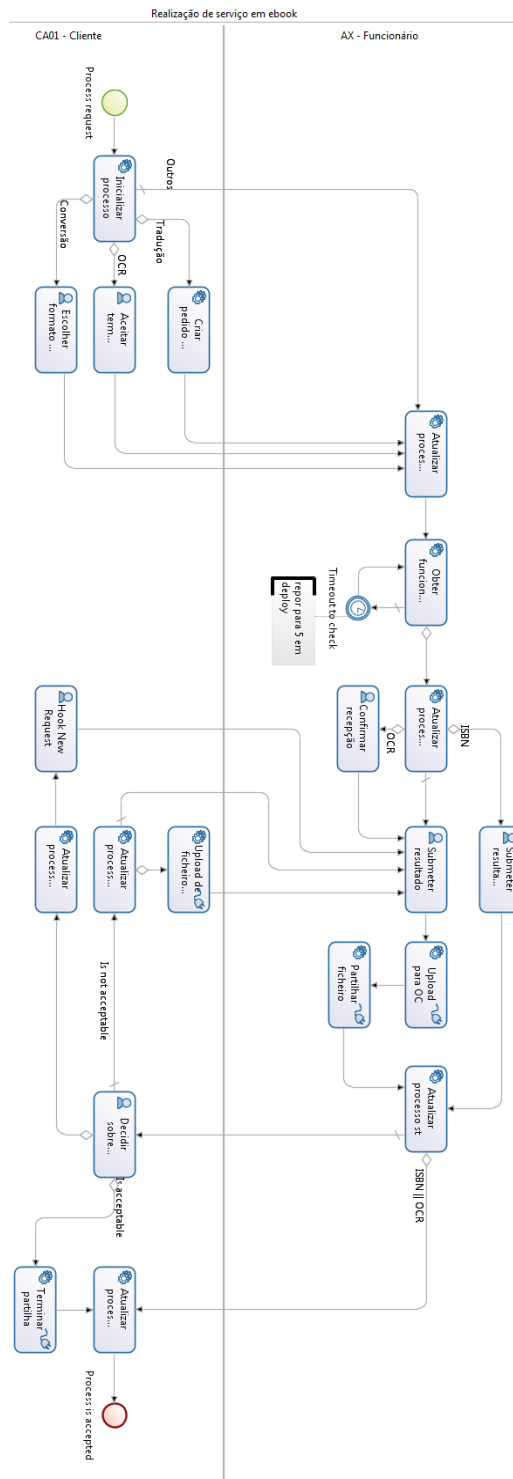


Figura 31. Processo para serviços acessórios generalizado

Uma vez que já se existiam vários processos com num estado mais estável, começou-se a considerar a migração dos processos e todos os elementos a eles associados da versão *Studio* para a versão de produção, instalado no servidor. Para este feito, foi recebido acesso a uma máquina virtual, com Debian8 como o seu sistema operativo, juntamente com 4 GB de RAM e 15 GB de disco. Embora não tendo sido realizada juntamente com a instalação do Bonita e outros ficheiros acessórios relevantes (bibliotecas e conectores extras, não incluídos na instalação base) foi também instalado um sistema de gestão de documentos para se obter uma melhor gestão de todos os documentos que poderiam eventualmente estar envolvidos nos diversos processos. Embora tenha sido experimentado inicialmente o OpenKM, acabou-se por optar pelo ownCloud. Estes sistemas, juntamente com um outro sistema que foi também foi analisado, encontram-se descritos com maior detalhe no capítulo anterior.

Chegando a um ponto onde os processos já não se encontram num estado embrionário e estando a aplicação numa situação razoavelmente estável, passou-se para um ponto de grande importância para a plataforma: o aspeto. Embora os estilos e o *look & feel* geral da aplicação possua alguma qualidade, contém algumas falhas de alguma gravidade, mais concretamente em dois grandes pontos:

A página “portal” de onde todos os utilizadores podem realizar tarefas que se encontram atribuídas a eles ou iniciar novos processos é confusa, ponto este que ficou claríssimo durante uma pequena demonstração da plataforma a uma pessoa que nunca tinha tido experiência com a plataforma mas que sentiu dificuldades em começar um processo novo assim como em retomar a execução de uma tarefa já em execução.

Continuando com as falhas, a utilização excessiva de cores que não são próprias no desenvolvimento de *software*, neste caso o vermelho, tomava um lugar principal nas cores da plataforma o que é uma situação pouco desejável visto que esta cor torna-se desconfortável para o utilizador quando exposto a esta durante períodos longos de tempo.

Considerando os pontos indicados nos dois parágrafos anteriores necessitou-se de encontrar uma solução que solucionasse estes problemas mas que ao mesmo tempo não consumisse demasiado tempo para a sua realização. Para superar o segundo ponto, a alteração das

cores da página através da modificação dos ficheiros CSS relevantes seria suficiente, tendo sido algo que acabou por ser realizado (no entanto estas alterações tiveram um escopo maior, não só dirigido às cores mas também a outros aspetos como tipo de letra, aspetos dos botões; e tendo um alvo um pouco diferente, sendo os alvos as páginas de formulários). No entanto para que se pudesse eliminar o primeiro problema recorreu-se à elaboração de uma página “portal” customizada.

Para a criação deste novo portal, começou-se pela criação de esboços simples do que seria o *layout*, sendo um ponto essencial a junção da informação das tarefas possíveis de serem realizadas, a capacidade de realizar estas mesmas tarefas e a habilidade de iniciar novo processos. Estas três funcionalidades acabaram por estar presentes todas na mesma página e lado a lado, com o objetivo de agilizar a realização de tarefas. Aquando da demonstração mencionada anteriormente, notou-se que o utilizador tentava realizar uma tarefa através do duplo clique nesta e embora não fosse uma funcionalidade existente na página do portal original, foi algo também foi considerado sendo tal funcionalidade adicionada como alternativa ao clique no botão indicativo de realização de tarefa. Outro problema também mencionado durante a demonstração foi a inability de perceber qual a obra ou autor a tarefa estava ligada, a não ser iniciado a tarefa, tendo este ponto em mente, foi imperativo a adição desta informação de algum modo na página.

Embora se utilizasse um DMS, nunca foi pretendido que o utilizador tivesse acesso direto a este, pelo que foi necessário obter mecanismos para a obtenção e envio dos ficheiros, de forma automática, para o ownCloud. Após uma pesquisa pela documentação e pelo fórum da comunidade, percebeu-se que haviam várias formas, mas de destacar duas, chamadas à API do ownCloud, para a partilha e término de partilhas de ficheiros e pastas, chamadas estas que não ofereceram nenhum problema notável para a sua integração no sistema. No entanto, para o envio dos ficheiros desde a submissão nos formulários do Bonita até o seu armazenamento no ownCloud, necessitou-se de adicionar dois passo intermédio: a criação de uma cópia temporária numa pasta dentro do servidor Tomcat e a chamada de um pequeno script para enviar esta cópia para o ownCloud e a eliminação deste ficheiro temporário dentro do servidor.

Obtendo uma versão relativamente estável e madura da plataforma foi apresentado à gerência com a participação do co-orientador. Durante

a apresentação foram feitos vários comentários e apontaram-se problemas e possíveis soluções a estes, pelo que estes foram registados e posteriormente resolvidos da melhor forma possível. Ocorreram várias apresentações, com a mesma composição de elementos presentes, tendo, no entanto, um foco diferente nos componentes em cada momento de apresentação / avaliação.

Estando o projeto numa fase quase final, foi pedido que a plataforma fosse também avaliada pela *designer* da empresa para que pudesse realizar a sua análise e sugerir modificações relevantes à plataforma de modo a melhorar e trazê-la mais próxima com a imagem existente da Lernasnuvens. Um dos maiores problemas identificados era a desconexão das cores utilizadas na plataforma (tons verdes) com a página presencial da Lernasnuvens (tons pretos/escuros com detalhes amarelos). Outras sugestões para melhorar o aspeto da plataforma incluíam também a utilização de um outro tipo de letra (Open Sans) e o redimensionamento dos diversos elementos da página.

Por fim, achou-se relevante criar novos diagramas DEMO que refletissem os processos da forma como estes foram implementados. Para esta etapa consideraram-se apenas o ATD, PSD e OFD, que se seguem.

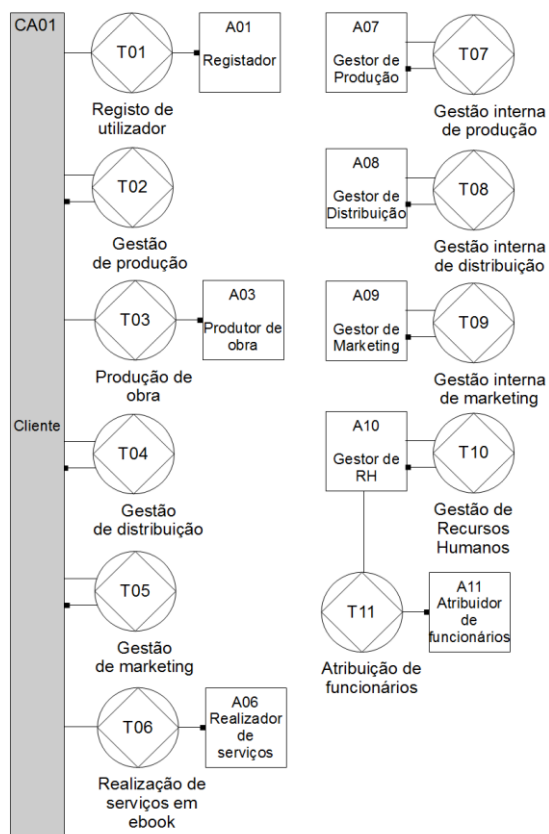


Figura 32. ATD, versão após desenvolvimento

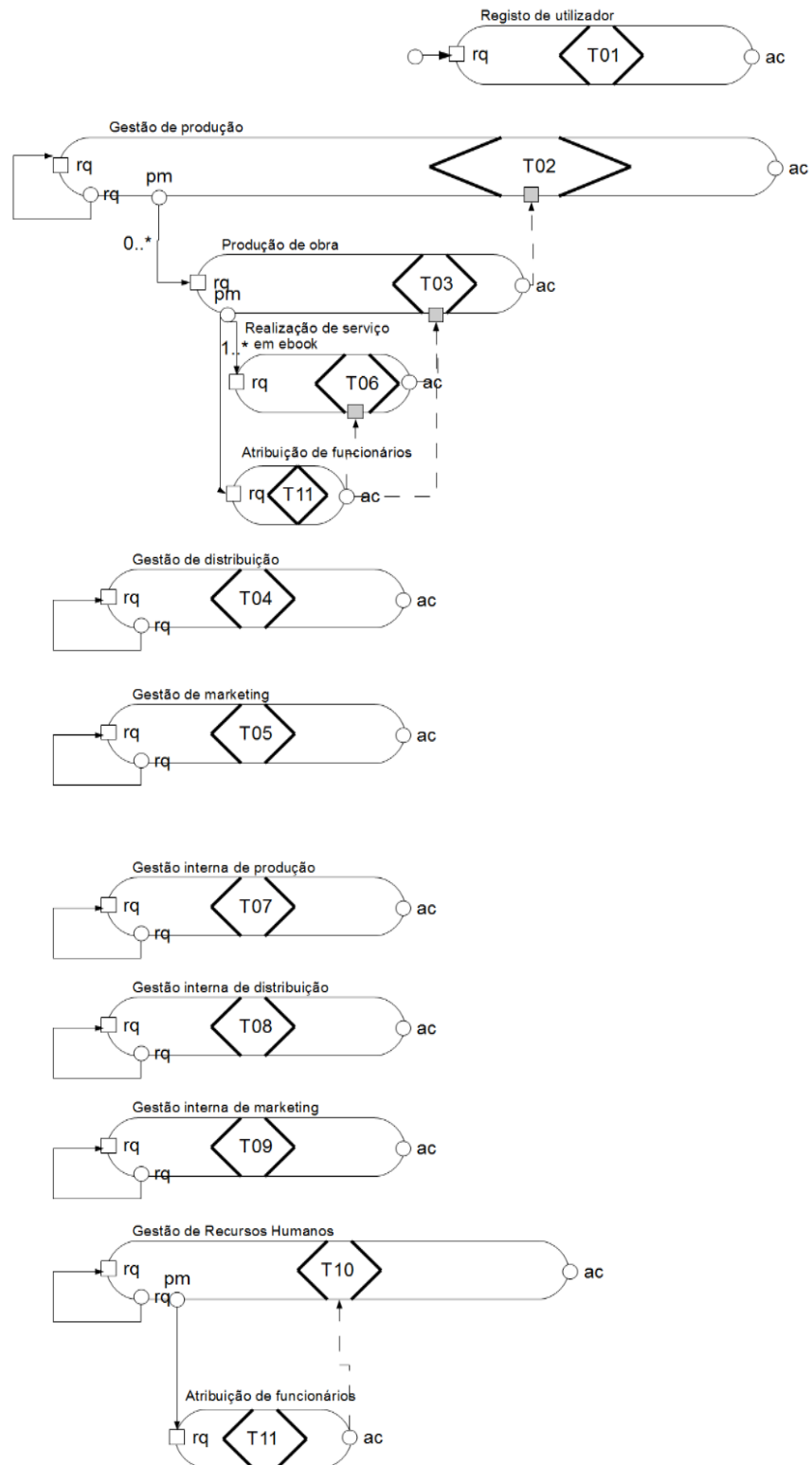


Figura 33. PSD, versão após desenvolvimento

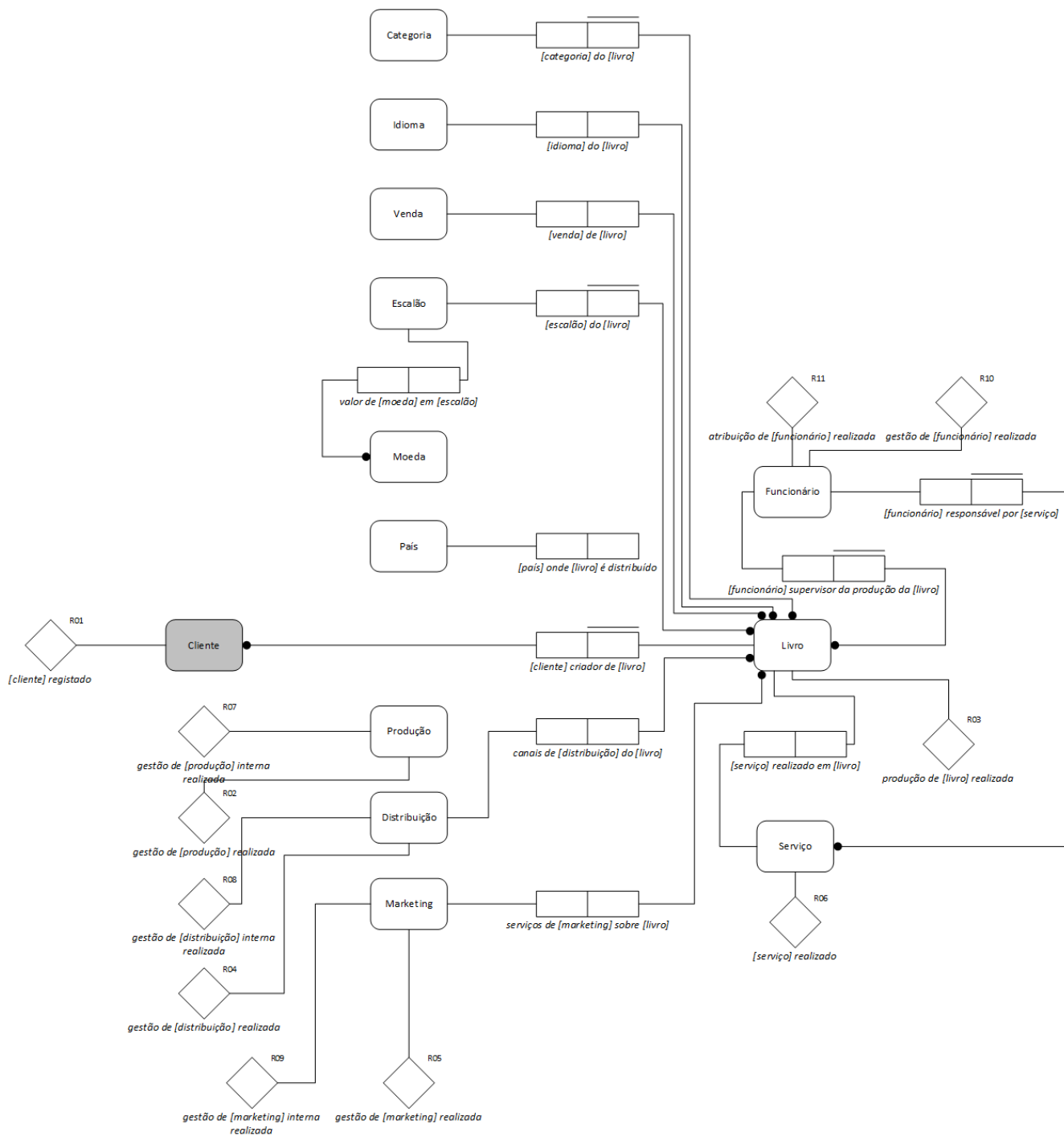


Figura 34. OFD, versão após desenvolvimento

III.3. Conversão DEMO para implementação

Idealmente, a transformação dos modelos obtidos através da aplicação do DEMO para uma base de dados que é capaz de suportar uma plataforma segue uma lista sem a necessidade de realizar atalhos ou passos que vão contra a lógica da transformação.

Em teoria, é possível obter o modelo relacional através do OFD, seguindo estes passos:

- Cada classe é uma tabela;
- Cada tipo de resultado é uma tabela quando a classe associada a este tem mais que um tipo de resultado associado. No caso de existir um número muito reduzido de tipos de resultado associado a uma classe, então este é uma propriedade da tabela da classe;
- Os tipos de facto que não possuem uma restrição de unicidade são uma tabela n:m entre as duas classes interligadas;
- Para os tipos de facto onde existe uma restrição de unicidade, a tabela da classe que não é unitária tem uma chave estrangeira da tabela da classe que possui a restrição unitária.

Relativamente à criação dos BPD:

- Cada processo identificado na TRT é uma *pool*;
- Para cada *pool* é adicionada uma *lane*, sendo que cada *lane* representa um dos papéis de ator envolvidos no processo.

Utilizaram-se estes pontos acima para se obter um ponto de partida, sendo que o esquema relacional inicial está presente nos anexos. No entanto, à medida que o projeto foi desenvolvido foi necessário realizar algumas alterações de modo a dar uma melhor experiência aos utilizadores finais.

Durante a fase inicial da criação da base de dados, optou-se por aglutinar as tabelas do esquema relacional relativas aos vários serviços acessórios da produção do livro por não se terem identificadas propriedades únicas a cada serviço, simplificando-se a base de dados.

Originalmente foi criado um processo/*pool* para cada serviço acessório da produção do livro, no entanto, e embora existem certas diferenças entre os diversos serviços, foram aglomerados todos os processos num único processo. Este novo processo, expetavelmente tem uma dimensão maior, pois necessita de ter *hooks* para tratar de casos especiais que são únicos a um ou alguns serviços.

Adicionalmente, e em grande parte devido à forma como os processos são tratados pelo Bonita, anexaram-se a alguns processos

maiores outros processos de tamanho muito reduzido para fornecer uma melhor experiência de utilizador na forma de uma redução do número de cliques necessários para avançar.

Uma vez que o DEMO tem presente as diversas fases do processo (*request, promise, etc...*) torna-se relevante registar as várias passagens entre elas e qual o processo associado. Para o efeito foi criado uma tabela com um campo para cada fase que guarda o tempo de chegada à fase e juntamente com um campo indicador do estado atual. Outras tabelas que registam de alguma forma outros processos recebem um campo que é a chave estrangeira desta nova tabela.

IV. Arquitetura

IV.1. Tecnologias e Linguagens

IV.1.1. MySQL

O MySQL é um sistema de gestão de base de dados, que utiliza como linguagem o SQL. É o *standard de facto* para aplicações web devido ao seu *engine* de *queries* de alto desempenho, grande capacidade de inserção de dados e suporte para funções especializadas na web. Outras vantagens da utilização do MySQL incluem a escalabilidade, flexibilidade, alta disponibilidade e suporte para transações. [24]

IV.1.2. Groovy e Java

O Java é uma linguagem de programação de propósito geral orientada a objetos. Desenhada para ser simples o suficiente para que muitos programadores consigam obter fluência. [25]

O Groovy é uma linguagem, também orientada a objetos, desenvolvida para a plataforma Java, podendo ser utilizada como uma linguagem de *script* para esta. A versão 1.0 foi lançada no início de 2007, sendo que em Julho de 2012 foi lançada a versão 2.0, estando a próxima (3.0) planeada para o fim de 2015. Esta linguagem é utilizada para vários propósitos no Bonita BPM, incluindo a definição de valores iniciais de variáveis, tratamento de dados após *input* dos utilizadores ou busca, alteração e inserção de dados na base de dados.

IV.1.3. Apache Tomcat

O Apache Tomcat é uma implementação *open-source* do *Java Servlet*, *JSP*, *Java Expression Language* e da tecnologia *Java WebSocket*. É desenvolvido pela Apache Software Foundation, sendo lançado pela primeira vez em 1996. [26]

IV.1.4. ownCloud

O ownCloud é um sistema de partilha e sincronização de ficheiros. Semelhante à Dropbox, tem algumas diferenças como o seu custo (gratuito) e a inexistência de limites de armazenamento (exceto devido a casos de limitações de hardware ou de quotas impostas pelo administrador do sistema). Consegue fornecer acesso aos seus dados

através de uma interface web ou por WebDAV (protocolo de transferência de documentos), e oferece extensibilidade através da sua API. [23]

IV.1.5. JavaScript

É uma linguagem de *scripting* orientada a objetos, multi-plataforma, que permite a criação de aplicações web interativas através da sua capacidade de interagir com HTML. Juntamente com o HTML e CSS é uma das três tecnologias essenciais para a produção de aplicações web. [25]

É também relevante referir o jQuery, uma biblioteca de JavaScript desenhada para simplificar o *scripting* de HTML.

IV.2. Componentes

Após a criação de todos os modelos relevantes foi possível agrupar diferentes partes da aplicação em grupos que serão descritos de seguida.

IV.2.1. Utilizadores

Uma vez que o registo de clientes e de funcionários é obrigatório é necessário que se forneçam mecanismos para o registo destes.

Da parte dos clientes, está previsto que estes se registem através da execução de um processo de pequena dimensão.

Quanto aos funcionários, estes irão ser adicionados ao sistema através da ação de um funcionário já existente (com o papel de Gestor de Recursos Humanos). Aquando do seu registo, é necessário a indicação dos papéis que este irá executar, sendo possível a adição ou remoção de um ou mais numa ocasião futura.

IV.2.2. Produção

É o componente que possui a maior dimensão entre todos os componentes e aquele de onde a maioria dos processos origina.

Da parte do cliente final, envolve o pedido de criação de novas obras (ebooks); a subscrição de serviços acessórios quer no momento da criação do pedido, quer num momento futuro; a edição de dados relativos à obra (e.g., título da obra); e a aprovação (ou rejeição) de orçamentos e resultados derivados do pedido de execução de serviços.

Da parte dos funcionários, envolve a realização dos serviços pedidos pelos clientes; a submissão dos resultados destes mesmos serviços e a atribuição de funcionários para a sua execução.

IV.2.3. Distribuição

Havendo a produção de uma obra, o possível passo seguinte é a distribuição pelos diferentes canais.

De uma forma simplificada o utilizador escolhe o canal que pretende distribuir, os países e qual o escalão de preços a utilizar. Ao realizar este pedido, um funcionário ao realizar uma gestão interna de distribuição, poderá então tratar do pedido do cliente (é de referir, que a distribuição em si, é feita sem apoio da plataforma, serve apenas como registo de todos os pedidos: por executar, completos ou cancelados) e após distribuição iniciada é indicado como completa. Este processo é quase idêntico para a paragem de distribuição.

IV.2.4. Marketing

Muito semelhante ao componente de distribuição, no entanto para a realização de serviços de marketing sobre distribuições. O processo de pedido e realização pode ser descrito lendo a descrição da realização de uma distribuição, explicado no capítulo IV.2.3. Distribuição.

IV.2.5. Outros

Para que se possam iniciar os diversos processos é necessário que exista uma página central, por defeito, existe o *Bonita BPM Portal*, no entanto este sofre de problemas de usabilidade, mais especificamente: não é fácil de perceber, para um novo utilizador, como iniciar um novo processo; as cores presentes nas páginas não são adequadas para uso prolongado devido à utilização excessiva do vermelho e existem funcionalidades e informações que são desnecessárias e apenas criam confusão aos utilizadores.

Devido a estas razões decidiu-se realizar a criação de uma página de portal, onde a continuação dos processos e execução de novos processos seja possível mas corrigindo os problemas mencionados anteriormente.

IV.3. Funcionalidades implementadas

Enuncia-se de seguida o estado de implementação de cada requisito enunciado no capítulo anterior:

1. O registo de utilizadores (clientes) é possível, no entanto, durante a fase inicial de utilização da plataforma, o registo destes será feito pelo administrador do sistema;

2. O registo de intenção de produção do ebook encontra-se implementado por completo, sendo certos campos como o título (que poderá ser alterado), o idioma, e o número de páginas dados obrigatórios. Relativamente ao texto, a situação deste será descrita em pormenor no ponto 5;

3. A adição de novas funcionalidades é possível, no entanto e como discutido dentro da organização, a adição de funcionalidades pode estar condicionada com o estado de produção do ebook, sendo que certas funcionalidades apenas poderão ser adicionadas durante a criação do pedido de produção de ebook;

4. Embora tenha sido previsto a conversão para outros formatos digitais, este serviço de momento encontra-se desativado, no entanto, uma vez que na implementação dos processos, os serviços acessórios para a produção utiliza o mesmo processo a reativação deste requererá esforço mínimo;

5. A submissão do texto da obra poderá ser feito de várias formas: a forma padrão é a submissão durante a criação do pedido de produção de ebook; uma segunda possibilidade é a contratação do serviço de OCR para a passagem de um documento em papel para digital, que responde diretamente ao requisito; uma última alternativa passa pela submissão do ficheiro em formato digital num momento posterior à criação do pedido;

6. Para cada serviço acessório, retirando casos excecionais, o utilizador pode e deverá aprovar (ou rejeitar) o resultado fornecido pelo funcionário responsável por este. O mesmo se verifica para o processo de produção de ebook. Caso o utilizador rejeite o resultado, este poderá adicionar comentários que esclareçam os problemas existentes e o funcionário deverá re-submeter um novo ficheiro contendo o resultado melhorado para aprovação do cliente;

7. Após requerer a produção de um ebook e indicar quais os serviços acessórios que são requeridos pelo cliente, é mostrado a este um orçamento baseado no número de páginas da obra e os serviços escolhidos. Este valor poderá ser alterado por um funcionário caso seja verificado inconsistências no número de páginas indicado pelo cliente e o número de páginas identificado pelo funcionário;

8. A transferência do resultado final de cada ebook é permitida tanto ao avaliar o resultado final de produção como também num momento posterior, na página de ficheiros, do novo portal;

9. O sistema permite a realização do pedido de início de distribuição. No entanto tal pedido apenas poderá ser realizado quando a produção da obra se encontrar finalizada;

10. O cliente, ao escolher os canais para distribuição recebe a indicação da percentagem de comissão cobrada por estes.

11. O cliente tem a possibilidade de remover uma das suas obras de um ou mais canais de distribuição, ou países individuais, no entanto é necessário esclarecer que o cliente realiza apenas um pedido para terminar a distribuição e esta terá que ser processada por um funcionário que poderá ou não aceitar o término da distribuição. Outra possibilidade fornecida ao cliente é a realização de uma pausa de distribuição em vez do término desta.

12. Para cada obra é possível a contratação de serviços de marketing, estes ficarão ligados à obra, estando independentes dos canais de distribuição.

IV.4. Como se utiliza a ferramenta

Descrito do ponto de vista de um cliente (e.g., uma editora) descreve-se a utilização da plataforma:

Um funcionário de uma editora (um novo cliente da plataforma) entra na página de acesso e, por ser a primeira vez, escolhe a opção de registo. Após preenchimento do formulário passa a ser um utilizador da plataforma com acesso aos processos relevantes.

Visto ser um novo funcionário e conseqüentemente não possui nenhum livro produzido, escolhe a opção de produção de obra, começa por preencher os dados básicos da obra (onde se incluem informações como o título, o ISBN do ebook, etc...) confirma estes dados e passa à seleção dos serviços acessórios à produção. Escolhe uma capa sem imagem e uma revisão da obra, continua no processo e vê o orçamento inicial da obra. Ao avaliar este orçamento nota que se esqueceu de pedir uma tradução para português (porque a obra era em inglês), escolhe a opção para rever os serviços e desta vez escolhe também a tradução. Continua e após decidir que o valor do orçamento entra dentro dos seus parâmetros de aceitável aprova o orçamento e escolhe que tipo de pagamento irá realizar. Do outro lado da plataforma é criado um pedido

de atribuição para a função de supervisor da produção desta nova obra. Após atribuição, o supervisor submete o documento contendo a fatura, após submissão, o utilizador poderá realizar o *download* deste ficheiro e realizar o pagamento. Embora o supervisor só deverá declarar a continuação da realização do serviço após confirmação do pagamento, esta confirmação poderá ser realizada a qualquer momento.

Havendo confirmação do pagamento inicial da produção, os serviços acessórios poderão começar, sendo que existe um novo pedido para atribuição de funcionários para cada um dos serviços acessórios.

Aquando da conclusão de qualquer serviço (exceto a delegação de ISBN e a realização de serviço de OCR) o cliente terá que dar a sua aprovação sobre o serviço realizado, e caso não ache satisfatório o resultado poderá recusar, fornecendo também alguns comentários a explicar o(s) problema(s).

Quando todos os serviços acessórios se encontrarem completos o supervisor poderá submeter o resultado final e o ficheiro contendo a fatura para pagamento final (caso não tenha sido realizado o pagamento na íntegra durante a fase de pagamento inicial), esta fase é semelhante ao pagamento inicial, podendo o supervisor continuar com o processo antes do pagamento ser confirmado. Por fim, o cliente confirma a qualidade do trabalho e aceita, terminando o processo de produção de obra (ou recusa, requerendo que alterações de algum tipo seja feita).

Tendo agora uma obra completamente produzida, o cliente poderá escolher distribuir a sua nova obra, iniciando para esse propósito um processo de gestão de distribuição e escolhendo a opção para iniciar nova distribuição. Escolhe o canal que lhe parece ser mais atraente tendo em conta a sua popularidade e percentagem de comissão cobrada, confirma a opção, escolhe os países para onde pretende distribuir, avança para a realização de decisões escolhendo o idioma a ser utilizado e o escalão de preços a ser praticado e termina o processo confirmando as suas escolhas. Do lado do funcionário responsável pela distribuição, ao realizar uma gestão interna de marketing nota que existem novos pedidos de início de distribuição, e obtendo a informação da obra e opções tomadas pelo cliente, toma os passos necessários para que a distribuição possa ocorrer. Uma vez estando o início da distribuição completa, o funcionário declara-a como completa.

V. Testes

Uma vez atingido um nível relativamente estável da plataforma foi fornecido ao co-orientador os dados de acesso à plataforma, com uma conta de funcionário com todos os papéis, e outra conta de cliente. Durante o momento de avaliação todos os processos já se encontravam desenvolvidos. A nível de problemas a maior parte prendeu-se em questões de estética, interface com o utilizador e textos descritivos. Descreve-se de seguida com maior detalhe:

Área	Problema
Login	O dimensionamento dos botões não é proporcional
	Os botões têm um mau posicionamento (deviam de estar lado a lado)
	Os textos utilizados nos botões não são os melhores
	Os campos de texto encontram-se demasiado pequenos
Dashboard	O tipo de letra na caixa de texto não é igual ao resto do documento
	As cores utilizadas deveriam ser mais ténues
	Não existe feedback para dados de acesso incorretos
	Os elementos encontram-se mal posicionados e é necessário repensar o seu posicionamento
Gestão de produção	Existe um bug que impede executar tarefas a partir da dashboard
	Os botões deveriam estar posicionados lado-a-lado
	O posicionamento dos campos para a produção de um ebook não está coerente à edição de dados (faltam campos e certos dados não existem)
	Os elementos da janela não aproveitam suficientemente a janela e muita da página encontra-se em branco
	Os nomes de algumas tarefas deverão ser alteradas para algo mais simples (ex.: “Decisão de orçamento” para “Orçamento”)
	O texto nos botões para avançar ou recuar no processo deverá ser alterado
Ao chegar ao fim do processo de gestão é perguntado ao utilizador se pretende continuar com a gestão ou	

Área	Problema
	terminar, no entanto, seria preferível adotar a continuação como escolha por defeito, podendo ser terminado na nova iteração.
Serviços acessórios (OCR)	É obrigatório que o utilizador confirme o envio para que o serviço possa avançar, no entanto é preferível que o processo continue sem novas interações desnecessárias com o cliente.
Serviços acessórios (todos)	Os serviços acessórios não respeitam a sequência hierárquica temporal, sendo possível realizar a paginação antes de todos os serviços que esta depende.
Gestão de RH	Não é possível interromper a criação de um novo funcionário

Tabela 4. Resumo da análise inicial da plataforma

Uma vez lido o relatório, tomou-se nota dos problemas e passos foram tomados para resolver os problemas, sendo que nenhum dos itens mencionados em si se verifica.

Paralelamente às intervenções pontuais com os orientadores foram também realizadas várias demonstrações do funcionamento e estado atual do projeto com o co-orientador e o patrão da empresa, ambos possuidores de conhecimento sobre o funcionamento dos processos da produção e distribuição de livros. Com estas demonstrações, foram apontadas falhas e várias sugestões para melhorar a plataforma. Entre os itens apontados destacam-se pontos como:

- Adição de uma vista geral, para cada uma das grandes áreas da plataforma (Produção, Distribuição e Marketing) para que se possa ter acesso a informação básica de todos os livros / clientes numa única área;
- Agilização / maior integração entre processos e entre tarefas do mesmo processo;
- Aspeto gráfico da plataforma;
- Envio de avisos via *email* para os intervenientes relevantes;
- Adição e alteração de textos (*labels* e textos descritivos) para fornecer uma maior clareza aos utilizadores finais;
- Redefinição e esclarecimento sobre papéis de utilizador que não foram definidos de uma forma muito específica numa fase prévia devido a todos os detalhes associados aos processos

ligados aos papéis não estarem completamente definidos em tal altura;

- Adição da funcionalidade de prever (baseado em valores fornecidos) o prazo espectável de conclusão da produção de um livro;

Após a realização de cada uma das demonstrações, foram realizadas alterações aos pontos indicados de modo a satisfazer as necessidades apontadas.

VI. Conclusões e Trabalho Futuro

Chegando ao final do projeto, é feita agora uma revisão do trabalho realizado. É possível se afirmar que o processo utilizado difere em vários pontos à utilização de uma abordagem clássica de engenharia, em grande parte devido à utilização do DEMO e BPMN.

A realização das análises e diagramas iniciais permitiram com que se obtivesse uma transição mais fluída dos requisitos e diagramas iniciais para o desenvolvimento, não havendo uma grande alteração nas fases finais. A nível das análises e diagramas DEMO conclui-se que os que têm um maior peso para um bom desenvolvimento de *software* são: o OFD, PSD, análise PIF e a TRT. Com o OFD temos um primeiro passo para a criação do diagrama relacional e a base de dados, que pode ser visto como o esqueleto de qualquer *software*. O PSD dá-nos a visão do fluxo dos vários processos, juntamente com as condições de espera e leis de unicidade, abstraindo-se ao mesmo tempo de informação que adiciona complexidade desnecessária levando a uma descrição completa mas curta de todo o projeto. Por fim, a análise PIF e a TRT merecem também uma menção pois são a partir destas que todos os outros diagramas derivam (direta ou indiretamente).

Quanto ao BPMN e o *engine* de BPM, reconhecem-se que existem vantagens pela sua utilização, como a maior facilidade de alterar páginas e as ligações entre estas, juntamente com a capacidade de introduzir operações no meio de qualquer processo de uma forma simples enquanto são mantidos os diversos componentes modulares. Por outro lado a utilização do *engine* traz os seus problemas: começando pela utilização do *engine* onde é introduzida uma nova camada no projeto o que leva a um maior consumo de recursos aliado a algumas restrições no desenvolvimento. Adicionalmente, e embora seja algo que não é sempre observável, existiu a necessidade de aprender a como utilizar uma ferramenta, o que fez com que o custo para utilizar tal ferramenta fosse superior. Por fim, um problema que pareceu existir, na opinião do formando, é o direcionamento do BPM como uma ferramenta que assiste os processos de uma empresa, mas quando estes estão envolvidos apenas com funcionários da empresa, ou seja, quando o cliente não está envolvido diretamente, ou está envolvido sem ter contacto direto com a ferramenta.

O projeto em si trazia, desde início, as suas dificuldades e barreiras que teriam de ser ultrapassadas. Começando pela área da produção e publicação de livros, que, para o formado foi uma área, com um escopo alargado, onde previamente não possuía conhecimento. Realizar um trabalho desta dimensão sozinho traz algumas alterações à dinâmica de trabalho comparativamente aos diversos trabalhos realizados previamente, ligados às cadeiras feitas durante o curso.

Como mencionado previamente, o *engine* utilizado foi o Bonita BPM versão 6; no entanto um dos maiores problemas com o Bonita BPM e as ferramentas de BPM em si é uma restrição relativamente grande ao nível das interfaces com o utilizador (mais especificamente, prendendo-se demasiado com uma estrutura de formulários rígida) e falhas a nível de customização uma vez que a realização de alterações nos estilos das páginas foi uma tarefa que teve um nível de dificuldade acima do esperado, não pela complexidade mas pelos obstáculos à alteração de tais componentes. No entanto, foi anunciado recentemente a versão 7 que exclama em dar uma maior liberdade aos desenvolvedores. Alternativamente, a recriação da plataforma, utilizando não um engine de BPM mas sim uma *framework* para desenvolvimento web, como por exemplo Laravel ou Yii2. Uma vez que os BPD existem, estes seriam aproveitados como a base de desenvolvimento. O problema desta abordagem seria que se perderiam certas funcionalidades que o Bonita BPM fornece como a reordenação de várias páginas de uma forma rápida.

Adicionalmente, durante o desenvolvimento do projeto, novas funcionalidades foram achadas pertinentes serem adicionadas a este que trariam ao utilizador final funcionalidades que melhorariam a experiência de utilização (como, por exemplo, processos para se obter um orçamento completo e passar diretamente para a criação de uma nova obra, saltando a aprovação de orçamento, caso o cliente já tivesse aprovado este orçamento completo, e o pedido de distribuição e serviços de marketing caso estes também tenham sido escolhidos). Estas funcionalidades extra embora pertinentes no âmbito da plataforma, saem fora do âmbito do projeto, uma vez que não se encontravam nos requisitos iniciais e a inclusão no projeto traria a necessidade de dedicar mais tempo ao desenvolvimento, tempo este que nunca é suficiente.

Para terminar, penso ser relevante dar um pequeno vislumbre da minha experiência do estágio. Foi uma experiência deveras interessante desde a realização de um trabalho a só, com requisitos que acabam por

evoluir, dando-me um primeiro gosto do que será a experiência ao trabalhar no mercado de trabalho na área de Engenharia Informática. Também achei uma experiência interessante estar inserido num meio onde existiam colaboradores cujas áreas eram muito distintas das minhas. Não tenho dúvidas que tive sorte em ser escolhido para realizar este projeto, não só pelos desafios interessantes que me foram dados mas também pelo ambiente no local de trabalho que não tenho queixas a apontar.

No geral, penso que foi uma experiência que me permitiu evoluir as minhas capacidades e aplicar os conhecimentos adquiridos durante a minha carreira académica assim como perceber que embora na teórica seja de uma certa forma, na prática certos compromissos ou alterações acabam por ser necessários para que possa realizar o trabalho de uma forma descomplicada.

VII. Referências

- [1] P. M. Carvalho, *A Importância e o Futuro do E-book no Mercado Livreiro em Portugal*, 2012.
- [2] C. A. Figueira, *DEMO models based automatic workflow process generation*, 2013.
- [3] J. L. Dietz, “Enterprise Ontology”.
- [4] D. Aveiro, *DEMO Prof 2.3 - Module 1*.
- [5] D. Aveiro, “G.O.D. (Generation, Operationalization & Discontinuation) and Control (sub)organizations: a DEMO-based approach for continuous real-time management of organizational change caused by exceptions,” 2010.
- [6] D. Aveiro, *DEMO Prof 2.3 - Module 2*.
- [7] D. Aveiro, *DEMO Prof 2.3 - Module 3*.
- [8] D. Aveiro, *DEMO Prof 2.3 - Module 4*.
- [9] M. Owen e J. Raj, “BPMN and Business Process Management,” 2003.
- [10] Alfresco, “Activiti,” [Online]. Available: <http://www.activiti.org>. [Acedido em 2015].
- [11] Master the boss, “jBPM vs Activiti: which to choose?,” 12 Janeiro 2011. [Online]. Available: <http://www.mastertheboss.com/jboss-jbpm/activiti-bpmn/jbpm-vs-activiti-which-to-choose>. [Acedido em 2015].
- [12] G. N. Mansuri, “Drools + Activiti Integration,” [Online]. Available: <http://www.attuneww.com/ebook/drools-activiti-integration.pdf>. [Acedido em 2015].
- [13] J. Barrez, “What?!? Activiti needs how much memory?,” 4 Fevereiro 2013. [Online]. Available: <http://www.jorambarrez.be/blog/2013/02/04/activiti-memory-usage/>. [Acedido em 2015].

- [14] Red Hat, Inc., “jBPM Documentation,” [Online]. Available: <http://docs.jboss.org/jbpm/v6.0/userguide/>. [Acedido em 2015].
- [15] M. Salatino, “JBPM5 VS ACTIVITI5? DUMB QUESTION?,” 19 Janeiro 2011. [Online]. Available: <http://salaboy.com/2011/01/19/jbpm5-vs-activiti5-dumb-question/>. [Acedido em 2015].
- [16] “Minimum Hardware Requirements for JBossAS ?,” 16 Junho 2009. [Online]. Available: <https://community.jboss.org/message/277811>. [Acedido em 2015].
- [17] K. Baïna e S. Baïna, “User experience-based evaluation of open source Workflow systems: The cases of Bonita, Activiti, jBPM, and Intalio”.
- [18] R. Carhuatocto, “JBPM, BONITA, INTALIO, PROCESSMAKER, ACTIVITI. QUÉ BPM SUITE USO?,” 21 Julho 2011. [Online]. Available: <https://holisticsecurity.wordpress.com/2011/07/21/jbpm-bonita-intalio-processmaker-activiti-que-bpm-suite-uso/>. [Acedido em 2015].
- [19] OpenKM, “OpenKM,” [Online]. Available: <http://www.openkm.com>. [Acedido em 2015].
- [20] Software Insider, “OpenKM Reviews and Ratings | Document Management Software,” [Online]. Available: <http://document-management.softwareinsider.com/1/269/OpenKM>. [Acedido em 2015].
- [21] Alfresco Software Ltd., “Alfresco,” [Online]. Available: <http://www.alfresco.com>. [Acedido em 2015].
- [22] Alfresco Software, Inc., “Alfresco Community Edition 5.1 | Alfresco Documentation,” [Online]. Available: http://docs.alfresco.com/community/concepts/welcome-infocenter_community.html. [Acedido em 2015].

- [23] OwnCloud, Inc., “ownCloud,” [Online]. Available: <http://www.owncloud.org>. [Acedido em 2015].
- [24] Oracle, “Top Reasons to Use MySQL,” [Online]. Available: <https://www.mysql.com/why-mysql/topreasons.html>. [Acedido em 2015].
- [25] Mozilla Foundation, “Javascript Guide - Introduction,” [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>. [Acedido em 2015].
- [26] Apache Software Foundation, “Apache Tomcat,” [Online]. Available: <http://tomcat.apache.org/>. [Acedido em 2015].
- [27] BonitaSoft, “Bonitasoft,” [Online]. Available: <http://www.bonitasoft.com>. [Acedido em 2015].
- [28] “jBPM,” [Online]. Available: <http://www.jbpm.org>. [Acedido em 2015].
- [29] E. R. de Medeiros, “BPM with Bonita Open Solution,” 2011.
- [30] J. Gosling, B. Joy, G. Steele, G. Bracha e A. Buckley, “The Java Language Specification,” 2015.

VIII. Anexos

VIII.1. Anexo A – Versão inicial dos diagramas DEMO

ID Transação	Nome Transação	ID Iniciador	Nome Iniciador	ID Executor	Nome Executor
T01	Registrar utilizador	CA01	Cliente	A01	Registrador
T02	Produzir obra digital	CA01	Cliente	A02	Produtor de obras
T03	Traduzir obra digital	CA01	Cliente	A03	Tradutor
T04	Rever obra digital	CA01	Cliente	A04	Revisor
T05	Paginar obra digital	CA01	Cliente	A05	Paginador
T06	Criar capa de obra digital	CA01	Cliente	A06	Artista de capas
T07	Criar ISBN para obra digital	CA01	Cliente	A07	Atribuidor de ISBN
T08	Adicionar funcionalidade à obra digital	CA01	Cliente	A08	Funcionalista
T09	Converter obra digital para formato adicional	CA01	Cliente	A09	Conversor de obras
T10	Gerir produção	CA01	Cliente	CA01	Cliente
T12	Distribuir obra digitalmente	CA01	Cliente	A12	Distribuidor
T13	Pausar distribuição digital	CA01	Cliente	A13	Gestor de distribuição
T14	Subscrever serviços marketing para distribuição	CA01	Cliente	A14	Marketer
T15	Gerir distribuição	CA01	Cliente	CA01	Cliente

Tabela 5. Versão inicial da TRT

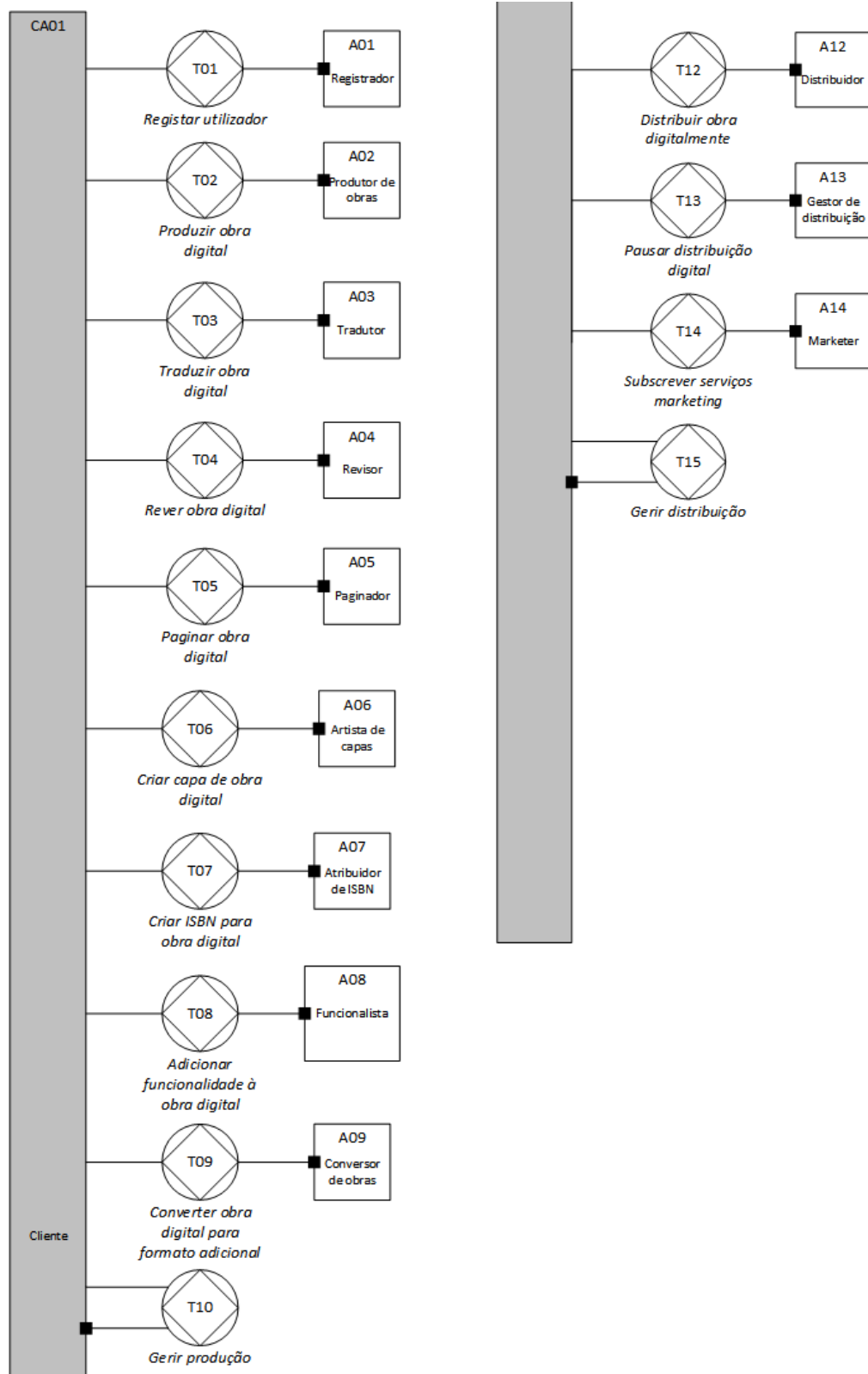


Figura 35. Versão inicial do ATD

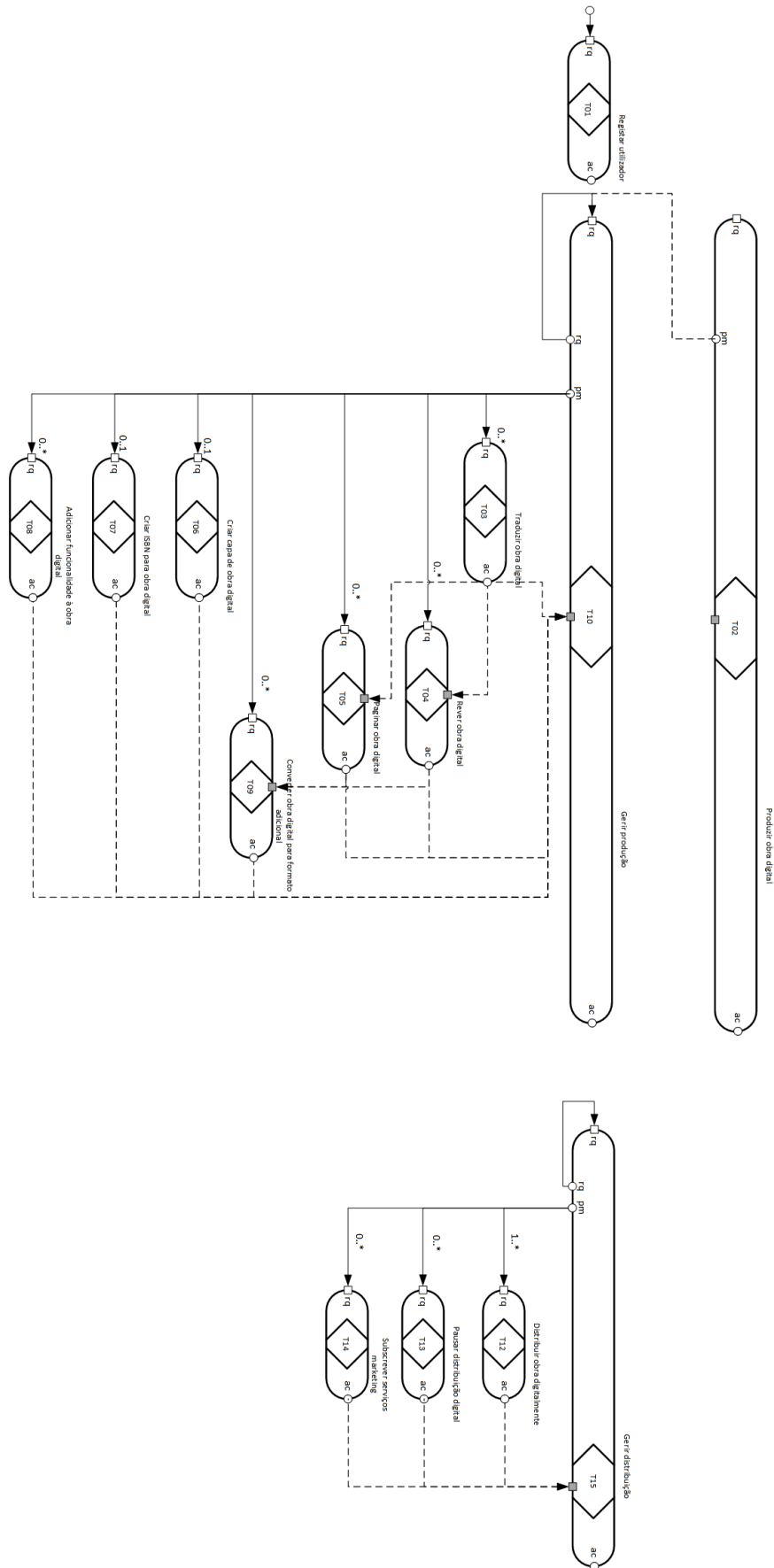


Figura 36. Versão inicial do PSD

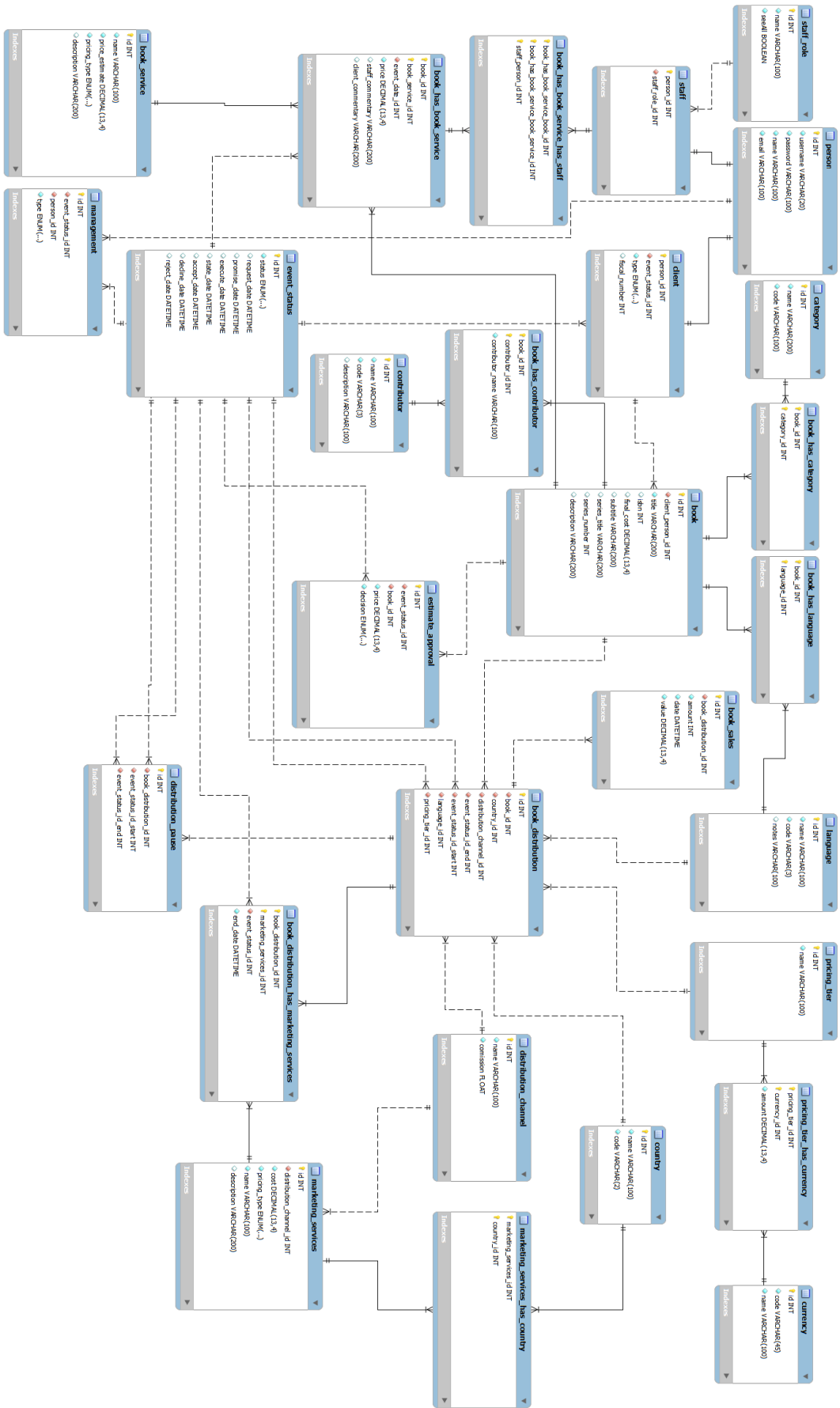


Figura 39. Versão 2 da base de dados

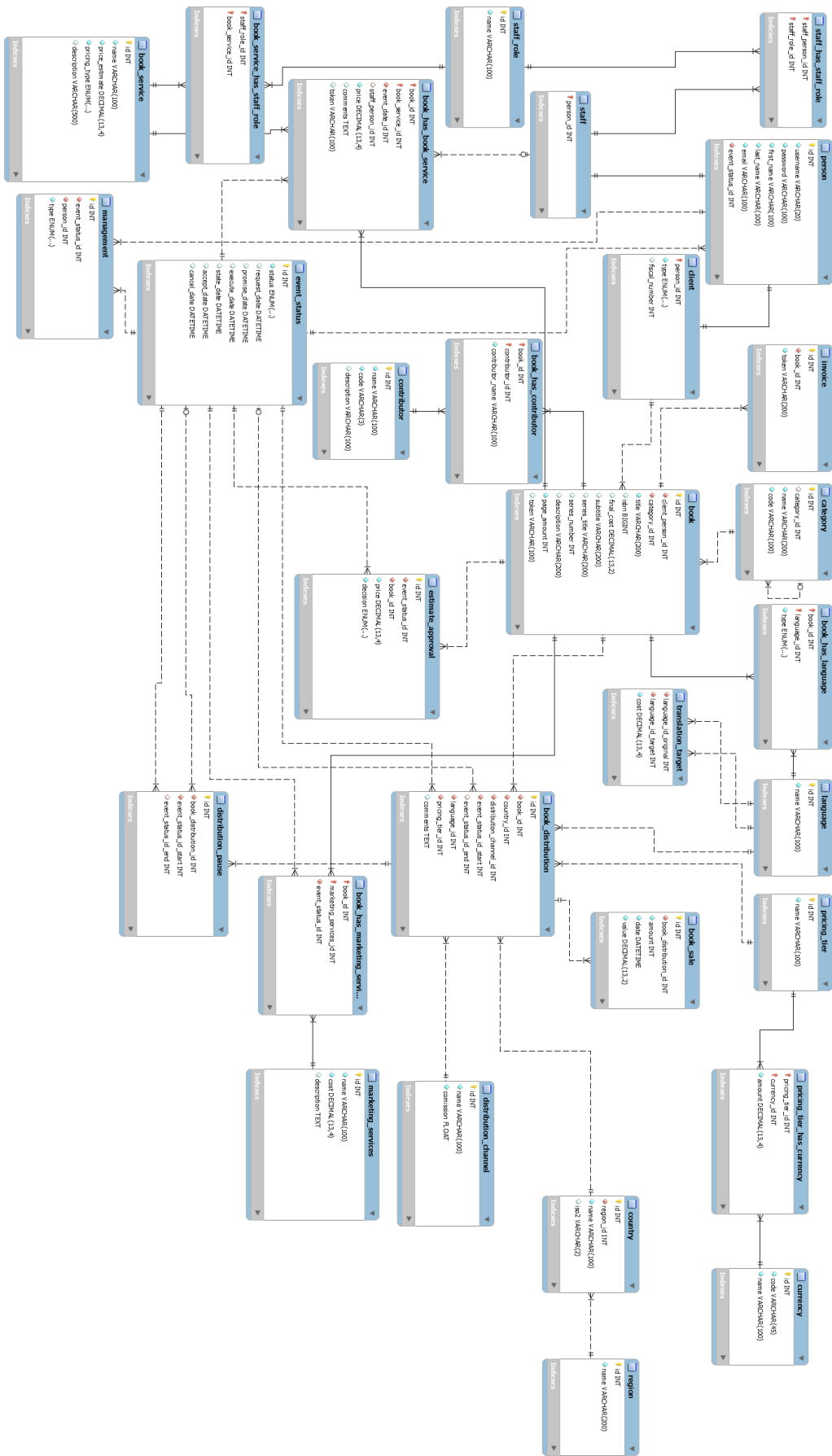


Figura 42. Versão 5 da base de dados

VIII.3. Anexo C – Screenshots da plataforma

Começar novo processo

- Gestão interna de produção
- Gestão interna de distribuição
- Gestão interna de marketing
- Gestão de recursos humanos
- Backend de valores

Tarefas por executar

Tarefas atribuídas | Tarefas pendentes

Tarefas pendentes

- Atribuir funcionário
- Atribuição de funcionários

Livro: Mestrado é vida

Serviço	Funcionário
Produção de ebook	Sem Atribuição
Revisão	Sem Atribuição
Capa sem imagem	Sem Atribuição
Paginação	Sem Atribuição
ISBN Digital	Sem Atribuição

Realizar tarefa

Figura 43. Página do dashboard/portal

Produção de obra

reg. user | Portal | Sair

Sumário

Livro: Mestrado é vida

Confirme se todos os dados foram corretamente preenchidos e se selecionou todos os serviços pretendidos.
Após confirmação, o seu processo será submetido e será contactado por um dos nossos colaboradores para confirmação e pagamento dos serviços.

Título Mestrado é vida

Subtítulo mestrado é amor

Título da Série Anexos e Demonstrações

Número da Série 1

Categoria EDUCATION

Sub categoria EDUCATION / Higher

ISBN de ebook

Idioma Português

Descrição

Número de caracteres 150000

Serviços seleccionados

Serviço	Preço (€)	Duração (h)
Produção de ebook	100.00	80
Revisão	1000.00	80
Capa sem imagem	200.00	80
Paginação	400.00	80
ISBN Digital	50.00	80
Total	1750.00	400

Método de pagamento Transferência bancária

Modalidade de pagamento 75% pré-pagamento

Se realizar o pagamento até 01/07/2016 poderá esperar que a conclusão dos trabalhos entre 05/09/2016 e 19/09/2016

CONFIRMAR | VOLTAR ATRÁS | CANCELAR

Biblioteca © 2015

Figura 44. Página resumo para criação de uma nova obra

Realização de serviço em ebook reg_user [Portal](#) [Sair](#)

Decidir Sobre Resultado
Serviço: Paginação
Livro: Tese é para fortes

Por favor descarregue os ficheiros e verifique se estão conforme o pretendido.
 Caso estejam conforme, por favor clique em confirmar resultado para prosseguir. Caso contrário, por favor diga-nos o problema e clique em Rejeitar Resultado

[DOWNLOAD DE FICHEIRO](#)

Histórico de comentários

Comentários adicionais

[ACEITAR RESULTADO](#) [RECLUSAR RESULTADO](#)

Bonitasoft © 2014

Figura 45. Decisão sobre resultado de um serviço

Gestão de distribuição reg_user [Portal](#) [Sair](#)

Confirmar Escolhas

Por favor confirme se pretende iniciar a distribuição de acordo com o selecionado

Título
 Mestrado é vida

Canais de distribuição
 Amazon
 Wook
 Lernasnuvens

Idioma
 Português

Escalação de preços

Escalação	EUR	USD	CAD	GBP	AUD	CHF	NOK	DKK	SEK
4	1,99	3,99	3,99	1,99	3,99	2,00	16,00	16,00	20,00

Países
 Espanha
 Portugal

[CONFIRMAR PEDIDO](#) [VOLTAR ATRÁS](#)

Bonitasoft © 2014

Figura 46. Realização de um pedido de distribuição

Ler Nas Níveis **Gestão interna de distribuição** staff_user Portal Sair

Ver Detalhes

Título
Mestrado é vida

Canal
Lernasnùvens

Idioma
Português

Estado
Início de distribuição pedido

APROVAR PEDIDO

RECLUSAR PEDIDO

Países alvo

Espanha
Portugal

Escalão

Escalão	EUR	USD	CAD	GBP	AUD	CHF	NOK	DKK	SEK
4	1.99	3.99	3.99	1.99	3.99	2.00	16.00	16.00	20.00

Novo comentário

ADICIONAR COMENTÁRIO

SAIR

Bonitasoft © 2014

Figura 47. Gestão interna de um pedido de distribuição

Ler Nas Níveis **Gestão interna de marketing** staff_user Portal Sair

Escolher Opção

Selecione a opção pretendida

Resumo de serviços

Livro	Serviço	Cliente	Estado	Data de pedido	Data de execução
Mestrado é vida	Website Presencial	John Doe	A ser processado	2016-06-27 13:07	Não concluído
Mestrado é vida	Press Release	John Doe	A ser realizado	2016-06-27 13:07	Não concluído
Mestrado é vida	Booktrailer	John Doe	A ser processado	2016-06-27 13:07	Não concluído

1 - 3 de 3

DECIDIR SOBRE PEDIDOS

SUBMITER FACTURAS

SAIR

Bonitasoft © 2014

Figura 48. Gestão interna sobre marketing