

Applicant Tracking System

PEDRO GONÇALO CUNHA CRUZ

novembro de 2021

Applicant Tracking System

Pedro Cruz

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Engenharia de Software**

Orientador: Dr. Paulo Maio

Júri:

Presidente:

- , -

Vogais:

- , -

Porto, 9 de novembro de 2021

Dedicatória

À Mia.

Resumo

Num contexto empresarial cada vez mais competitivo e não alheio às oscilações do mercado, a margem de erro aceitável para as empresas torna-se cada vez mais ínfima. Sendo imprescindível a adoção de ferramentas que respondam eficaz e eficientemente às mudanças de paradigma que, eventualmente, surjam. Para além disto, a constante pressão, proveniente da globalização, associada à especialização da mão de obra, leva a uma necessidade constante de mudança e inovação. Particularmente nos departamentos de recursos humanos que têm, não só de tentar prever essas mudanças, como planear uma resposta eficaz para manter a estabilidade das pessoas das organizações.

Dentro desse âmbito, o processo de recrutamento e seleção mostra-se indispensável para o sucesso das organizações, havendo a necessidade contínua de aperfeiçoar os métodos aplicados e os instrumentos tecnológicos para tal adoptados.

Este trabalho descreve o processo de análise e desenvolvimento de um novo sistema de informação para a gestão de todas as etapas, processos e sub-processos relacionados com o recrutamento e seleção de recursos humanos. Além disso, pretende-se que o sistema seja bastante configurável de modo a dar resposta às necessidades de diferentes clientes advindas, por exemplo, de estruturas organizacionais díspares e/ou de adoção de distintos métodos/fluxos de trabalho dentro da própria organização.

Por fim, o sistema desenvolvido foi, consequentemente, testado e implantado com sucesso num cliente real, mostrando ser capaz de responder aos seus anseios.

Palavras-chave: Recrutamento e Seleção, Recursos Humanos, Automatização, Informatização, Processos de Negócio

Abstract

In an increasingly competitive business environment, not oblivious to market fluctuations, the margin of error acceptable to companies becomes increasingly smaller. Therefore, it is essential to adopt tools that respond effectively and efficiently to paradigm shifts that may eventually arise. Additionally, the constant pressure from globalization, associated with the specialization of the workforce, leads to a constant need for change and innovation. Particularly in human resources departments, which must not only try to anticipate these changes, but also plan an effective response to maintain the stability of the organizations' people.

Within this scope, the recruitment and selection process is indispensable for the success of organizations, and there is a continuous need to improve the methods applied and the technological instruments adopted for this purpose.

The current paper describes the process of analysis and development of a new information system for the management of all steps, processes and sub-processes related to the recruitment and selection of human resources. In addition, it is intended that the system be highly configurable in order to meet the needs of different customers arising, for example, from disparate organizational structures and/or from the adoption of different methods/work flows within the organization itself.

Finally, the system developed was consequently tested and successfully deployed in a real client, proving to be able to respond to its expectations.

Agradecimentos

Em primeiro lugar, agradeço à CPCIT4ALL pela oportunidade, agradeço a todos os membros da equipa, em especial ao Michel Lobato e à Mariana Preto que foram quem mais me acompanhou durante este tempo.

Agradeço, também, ao meu orientador, Paulo Maio, por toda a paciência e esclarecimento de dúvidas durante a elaboração do presente documento.

Agradeço à minha família pela oportunidade proporcionada e por todo o apoio oferecido pelo caminho.

Por fim, mas não menos importante, agradeço a todos que me acompanharam nos últimos 5 anos e que, de uma forma ou de outra, tiveram influência naquilo que sou hoje. Dos momentos de convívio aos momentos de trabalho intenso, todos foram essenciais para fazer desta jornada inesquecível.

Conteúdo

Lista de Figuras	xv
Lista de Tabelas	xvii
Excertos de Código	xix
Glossário	xxi
Acrónimos	xxiii
1 Introdução	1
1.1 Contexto	1
1.2 Problema	2
1.3 Objetivos	3
1.4 Metodologia de Trabalho	4
1.5 Estrutura do Documento	4
2 Contexto	7
2.1 Domínio de Aplicação	7
2.1.1 Recrutamento e Seleção	9
2.1.2 E-Recruitment	10
2.2 Soluções Concorrentes	13
2.2.1 Manatal	13
2.2.2 TRAFFIT	13
2.2.3 Teamtailor	13
2.3 Solução Atual CPCIT4ALL	13
2.3.1 Desenho Atual	14
Vista Lógica	15
Vista de Processos	15
2.4 Comparação das Soluções	16
3 Estado da Arte	19
3.1 Enquadramento Histórico	19
3.2 Gestão de Processos de Negócio	20
3.2.1 Conceitos	20
3.2.2 Ciclo de Vida dos Processos	21

3.2.3	Representação de Processos de Negócio	24
	Diagrama de Atividade	25
	Business Process Model and Notation (BPMN)	25
	Case Management Model and Notation (CMMN)	26
	Decision Model and Notation (DMN)	27
	Comparação das Notações	27
3.3	Sistemas de Gestão de Processos de Negócio	27
3.3.1	Camunda BPM	28
3.3.2	WSO2 Business Process Server	28
3.4	Develop Framework 4 All	29
3.5	Comparação das Ferramentas	29
4	Análise de Valor	31
4.1	Processo de Inovação	31
4.1.1	Modelo de Desenvolvimento do Novo Conceito	32
	Identificação da Oportunidade	33
	Análise da Oportunidade	34
	Génese da Ideia	34
	Seleção da Ideia	35
	Definição do Conceito	37
4.2	Valor da Solução	37
4.2.1	Proposta de Valor	39
4.3	Business Model Canvas	39
5	Análise e Requisitos	41
5.1	Modelo de Domínio	41
5.1.1	Estrutura Organizacional	41
5.1.2	Domínio de RH	42
5.2	Engenharia de Requisitos	44
5.2.1	Funcionalidade	45
	Funcionalidades Transversais	50
5.2.2	Usabilidade	50
5.2.3	Confiabilidade	50
5.2.4	Desempenho	51
5.2.5	Sustentabilidade	51
5.2.6	Outras (+)	51
5.3	Notas Finais	52
6	Desenho	53
6.1	Nível 1 - Granularidade do Sistema	53
6.1.1	Vista Lógica	53
6.1.2	Vista de Processos	54
6.2	Nível 2 - Contentores ou Aplicações	55
6.2.1	Vista Lógica	55
6.2.2	Vista Processos	58
6.2.3	Vista Implantação	60

6.3	Nível 3 - Componentes de Contentores	64
6.3.1	ATS Services	64
	Vista Lógica	64
	Vista Processos	65
6.3.2	ATS Web	68
	Vista Lógica	68
	Vista de Processos	68
6.4	Nível 4	69
6.4.1	Vista de Packages	69
6.4.2	Functions	71
6.4.3	Estrutura dos Repositórios	73
7	Construção	77
7.1	Contexto Tecnológico da Empresa	77
7.2	Base de Dados	77
7.3	Interfaces dos Utilizadores	81
7.3.1	Menus	82
7.3.2	Páginas	83
	PageList	84
	PageFormulario	86
	PageFullEdit	88
	Páginas Dinâmicas	91
7.4	Processos de Negócio	92
7.4.1	Tabelas	93
7.4.2	Eventos de Tabelas	93
7.4.3	Funções de Tabelas	96
7.5	Fluxos de Trabalho	98
7.5.1	Configuração dos Fluxos	98
7.5.2	Funções dos Fluxos	100
8	Avaliação e Experimentação	105
8.1	Satisfação do Utilizador	105
8.1.1	Objetivo	105
8.1.2	Procedimento	105
8.1.3	Resultados	106
8.1.4	Análise dos Resultados	106
8.2	Fiabilidade do Sistema	107
8.2.1	Objetivos	107
8.2.2	Setup	107
8.2.3	Resultados	108
	Métrica 1	108
	Métrica 2 e 3	108
	Métrica 4	108
8.2.4	Análise dos Resultados	108
9	Conclusão	111

9.1	Objectivos Alcançados	111
9.2	Limitações e Trabalho Futuro	112
9.3	Apreciação Final	113
Bibliografia		115
A	Método AHP	119
A.1	Escala Fundamental de Saaty	119
A.2	Valores de IR Tabelados	120
A.3	Prioridades Relativas Normalizadas	120
A.4	Consistência das Prioridades Relativas	120
A.5	Matrizes de Comparação Paritária Normalizadas	121
B	Requisitos Levantados	123
B.1	Descrição dos Requisitos	123
C	Inquérito de Satisfação e Usabilidade	137

Lista de Figuras

2.1	Áreas de Atividade da Gestão de Recursos Humanos (RH) [Adaptado de (Robert L. Mathis, 1999)]	7
2.2	Sistemas Presentes no Fluxo Geral do Processo de Recrutamento e Seleção [Adaptado de (Kulkarni & Che, 2019)]	12
2.3	Vista Lógica da Solução atual	15
2.4	Diagrama de Sequência Obter Página HTML	15
2.5	Diagrama de Sequência Inserir Elemento	16
3.1	Tendências Relevantes a Business Process Management (BPM) [Adaptado de (Van der Aalst, 2004)]	19
3.2	BPM Lifecycle [Adaptado de (Dumas et al., 2013)]	23
3.3	BPM Lifecycle for Dynamic BPM [Adaptado de (Morais et al., 2014)]	24
3.4	Ecomendar Pizza Exemplo - AD UML	25
3.5	Ecomendar Pizza Exemplo - BPMN [Fonte: (Camunda, 2021a)]	26
4.1	Processo de Inovação [Adaptado de (Koen et al., 2002; Koen et al., 2001)]	32
4.2	Árvore Hierárquica de Decisão	35
4.3	Modelo Canvas	40
5.1	Modelo de Domínio das Dimensões	42
5.2	Modelo de Domínio Parcial	43
5.3	Modelo de Domínio do Fluxo de Aprovação	43
5.4	Modelo de Domínio do Fluxo de Trabalho	44
5.5	Diagrama de Casos de Uso - Utilizadores Externos	47
5.6	Diagrama de Casos de Uso - Requerente	47
5.7	Diagrama de Casos de Uso - Validador	48
5.8	Diagrama de Casos de Uso - Técnicos Abstractos	48
5.9	Diagrama de Casos de Uso - Técnicos Concretos	49
5.10	Diagrama de Casos de Uso - Master	49
6.1	Vista Lógica de Nível 1	53
6.2	Vista de Processos de Nível 1 - Publicar Anúncio	55
6.3	Vista Lógica de Nível 2	56
6.4	Vista de Processos de Nível 2 - Criar Pedido de Recrutamento	58
6.5	Vista de Processos de Nível 2 - Publicar Anúncio	59
6.6	Vista de Processos de Nível 2 - Publicar Anúncio Referência	59
6.7	Vista de Processos de Nível 2 - Realizar Candidatura	60

6.8	Vista de Implantação de Nível 2 - Alternativa 1	61
6.9	Vista de Implantação de Nível 2 - Alternativa 2	62
6.10	Vista de Implantação de Nível 2 - Alternativa 3	63
6.11	Vista de Implantação de Nível 2 - Alternativa 2	63
6.12	Vista Lógica de Nível 3 - ATS Services	64
6.13	Vista de Processos de Nível 3- ATS Services - Enviar Pedido de Recrutamento para Aprovação	66
6.14	Vista de Processos de Nível 3- ATS Services - Actuar Sobre um Pedido de Aprovação	66
6.15	Vista de Processos de Nível 3 - ATS Services - Publicar Anúncio	67
6.16	Vista Lógica de Nível 3 - ATS Web	68
6.17	Vista de Processos de Nível 3 - ATS Web	69
6.18	Diagrama de Packages - ATS Module	70
6.19	Diagrama de Packages - Package Code	70
6.20	Vista de Processos de Nível 4 - Visão Geral das Funções	71
6.21	Vista de Processos de Nível 4 - Extração de Variáveis	72
6.22	Vista de Processos de Nível 4 - Acesso à Base de Dados	72
6.23	Vista de Processos de Nível 4 - Execução do Serviço	73
6.24	Diagrama de Classes relativo aos Repositórios	74
7.1	Exemplo Directório da Base de Dados	77
7.2	Exemplo Directório de Menus	82
7.3	Menu Gerado pela <i>Framework</i>	83
7.4	Exemplo Directório de Páginas	84
7.5	Exemplo PageList - Candidato	86
7.6	Exemplo PageFormulario - Candidato	88
7.7	Exemplo PageFullEdit - Pedido de Recrutamento	90
7.8	Directório de Tabelas	93
7.9	Directório da Tabela DimensionValue	94
7.10	Directório da Tabela ATS Request	96
7.11	Lista de Perfis dos Fluxos de Trabalho	99
7.12	Configuração de um Fluxo de Trabalho	99
7.13	Lista dos Fluxos de Trabalho Configurados	100
7.14	Configuração Fluxos de Trabalho - Directórios	101
C.1	Inquérito - Página Inicial	137
C.2	Inquérito - Questões Parte 1	138
C.3	Inquérito - Questões Parte 1	139
C.4	Inquérito - Respostas Parte 1	140
C.5	Inquérito - Respostas Parte 2	141
C.6	Inquérito - Respostas Parte 3	142
C.7	Inquérito - Respostas Parte 4	142

Lista de Tabelas

2.1	Tabela de Comparação das Soluções	17
3.1	Tabela de Comparação das Ferramentas	30
4.1	Matriz de Comparação dos Critérios Selecionados	36
4.2	Vetor de Prioridades Relativas	36
4.3	Matriz Custos Associados	36
4.4	Matriz Tempo de Desenvolvimento	37
4.5	Matriz Facilidade de Implementação	37
8.1	Tabela dos Resultados do Inquérito	106
9.1	Tabela do Estado Final dos Requisitos - Parte 1	111
9.2	Tabela do Estado Final dos Requisitos - Parte 2	112
A.1	Escala Fundamental de Saaty [Fonte: (Nicola, 2020)]	119
A.2	Valores de IR para Matrizes Quadradas de Ordem n [Adaptado: (Nicola, 2020)]	120
A.3	Matriz de Comparação dos Critérios Normalizada	120
A.4	Matriz de Comparação de Custos Normalizada	121
A.5	Matriz de Comparação do Tempo de Implementação Normalizada	121
A.6	Matriz de Comparação da Facilidade de Implementação Normalizada	121

Excertos de Código

7.1	Definição Parcial da Tabela ATS Request	78
7.2	Coluna do Tipo Option - Estado do Pedido	79
7.3	Coluna do Tipo TableRef - Motivo da Contratação	79
7.4	Coluna do Tipo DimensionSet - Dimensão	80
7.5	Exemplo da Definição de uma Tabela de Domínio	80
7.6	Coluna do Tipo Dimension - Estabelecimentos de Ensino Preferenciais	81
7.7	Configuração de um Menu	82
7.8	Configuração da PageList - Visão Geral	84
7.9	Configuração da PageList - NavBar	84
7.10	Configuração da PageList - Parametros	85
7.11	Configuração da PageList - Grid	85
7.12	Configuração da PageFormulario - Visão Geral	86
7.13	Configuração da PageFormulario - Formulario	87
7.14	Configuração da PageFormulario - ActionButton	87
7.15	Configuração da PageFullEdit - Visão Geral	88
7.16	Configuração da PageFullEdit - NavBar	89
7.17	Configuração da PageFullEdit - Docks	89
7.18	Configuração da PageFullEdit - Panes	90
7.19	Página Dinâmica - HTML Geral	91
7.20	Página Dinâmica - Geração no Servidor	92
7.21	Métodos dos Eventos	94
7.22	Evento BeforeInsert do DimensionValue	94
7.23	Evento BeforeUpdate do DimensionValue	95
7.24	Evento BeforeUpdate do DimensionValue	96
7.25	Estrutura do Function.cs da tabela ATS Request	97
7.26	Implementação da função CancelRequest	97
7.27	Funções Fluxo Trabalho	100
7.28	Funções Fluxo Trabalho	101
7.29	Funções Fluxo Trabalho - BeforeStartWorkflow	102
7.30	Funções Fluxo Trabalho - Action	102
7.31	Funções Fluxo Trabalho - EndProcess	103
7.32	Funções Fluxo Trabalho - EndProcess	103

Glossário

Backlog	Uma lista prioritária de itens organizada pela equipa de desenvolvimento, contendo itens do tipo: <i>user stories</i> , mudanças a funcionalidades já implementadas e correção de erros.
bottleneck	Um ponto de congestão num sistema de produção (tal como uma linha de montagem ou uma rede informática) que ocorre quando as tarefas chegam a um ritmo maior daquele que o processo consegue acompanhar (Barone, 2020).
DF4A	Framework desenvolvida pela CPCIT4ALL para acelerar e auxiliar no desenvolvimento de aplicações Web, acrónimo do inglês Develop Framework 4 All.
E-Recruitment	Recrutamento eletrónico.
Kanban	Metodologia de gestão e desenvolvimento de projetos.
middleware	É um software que fornece serviços e capacidades comuns a aplicações fora do que é oferecido pelo sistema operativo (por exemplo, gestão de dados, mensagens, autenticação, e gestão de API) (Hat, 2021).
onboarding	É o nome dado ao processo de integração de novos funcionários numa organização, onde lhes é introduzida a cultura da própria empresa.
SaaS	Um Software as a Service é um programa que não é instalado em nenhum computador e é usado como um serviço, pela internet.

User Story	Uma explicação informal e geral de uma funcionalidade de software escrita a partir da perspectiva do utilizador final..
------------	---

Acrónimos

AD UML	Activity Diagram UML.
AHP	Analytic Hierarchy Process.
ATS	Applicant Tracking System.
BPEL	Business Process Execution Language.
BPM	Business Process Management.
BPML	Business Process Modeling Language.
BPMN	Business Process Model and Notation.
CMMN	Case Management Model and Notation.
CRM	Customer Relationship Managment.
CRUD	Create, Read, Update and Delete.
DMN	Decision Model and Notation.
FEE	Front End Engeneering.
FFE	Fuzzy Front End.
IIS	Internet Information Services.
NCD	New Concept Development.
NPD	New Product Development.
OMG	Object Managment Group.
OO	Object Oriented.
RH	Recursos Humanos.
RS	Recrutamento e Seleção.
SLA	Service Level Agreement.
UML	Unified Modeling Language.
VA	Value Analysis.
WSO2 BPS	WSO2 Business Process Server.

Capítulo 1

Introdução

O presente capítulo apresenta o contexto, o problema, o objectivo, a abordagem preconizada e, por fim, a estrutura do documento.

1.1 Contexto

O constante crescimento e evolução das tecnologias de informação tem vindo a mudar, gradualmente, a utilização que as pessoas dão aos sistemas informáticos. No contexto tecnológico atual, as empresas estão a recorrer cada vez mais à digitalização dos seus sistemas, não só para assegurar uma melhor e mais rápida análise dos dados guardados, mas também para auxiliar parcial ou totalmente a execução de determinados processos.

Tal é o caso para os departamentos de Recursos Humanos (RH) (Bell et al., 2006), que, para as organizações, mostram ser catalisadores para o sucesso. A sua função não deve ser vista como algo singular, uma vez que estes compreendem todos os processos necessários para a gestão das pessoas dentro de uma organização (Bell et al., 2006; Mukherjee et al., 2014). Como são exemplo os processos de recrutamento e seleção, gestão das férias e das ausências, pagamento da folha de salários, entre outros.

Ainda que, inicialmente, só tenham sido aplicados a grandes empresas, tem havido um crescimento constante no que concerne a adoção desse tipo de sistemas pelos departamentos de RH. Crescimento esse, que se mostra principalmente acentuado em pequenas e médias empresas (Mukherjee et al., 2014).

Isto porque, a utilização de sistemas informáticos, neste âmbito, mostra ter um impacto substancial tanto nos custos, como na eficiência e eficácia dos processos por ele abrangidos (Bell et al., 2006). Deste modo, não só os departamentos de RH têm vindo a tornar-se cada vez mais essenciais para a subsistência das organizações, como a utilização de sistemas informáticos tem vindo a acentuar ainda mais a sua contribuição.

Mais, a informatização dos sistemas no mundo empresarial trouxe consigo a necessidade de mão de obra cada vez mais especializada; aumentando, por sua vez, a necessidade já sentida de ferramentas específicas para melhorar os processos dos RH (Kulkarni & Che, 2019).

1.2 Problema

A CPCIT4ALL é uma empresa que pertence ao grupo empresarial português *Companhia Portuguesa de Computadores Informática e Sistemas S.A* (CPCis), o qual se dedica há mais de vinte anos ao sector das tecnologias de informação.

Atualmente, apesar de não haver um foco exclusivo no desenvolvimento de produtos para o setor da gestão de RH, a empresa fornece diversos módulos (e.g. recrutamento e seleção, contratação, etc.), integrados entre si, para esse mesmo setor.

No que respeita ao processo de Recrutamento e Seleção (RS), a empresa tem um módulo (presente em 10 clientes) que visa, apenas, servir como agregador de informação (neste caso, candidaturas); ignorando as necessidades mais recentes de mecanismos que contribuam para a automatização do processo.

No âmbito do RS, uma prática comum é a de não iniciar um processo sem antes haver um pedido feito pelos gestores de contratação à equipa de RH. Pedido esse que será aceite (ou não) e que dirá se o processo de RS pode, efetivamente, acontecer. Além disso, o pedido também servirá para saber quais características do processo (e.g. quantas vagas serão abertas, para que função, entre outras). Na solução atual, todo este mecanismo de pedidos não existe, ignorando, deste modo, aquilo que é uma prática corrente nos dias de hoje.

Além disso, o presente módulo não contém qualquer distinção entre o conceito de um candidato e uma candidatura. Isto implica que não haja nenhuma maneira explícita de conseguir verificar, por exemplo, quantas pessoas estão envolvidas em todos os processos. Uma vez que, se algum candidato se quiser envolver em dois processos distintos, não poderá associar as suas candidaturas a um único perfil.

Adicionalmente, e como último exemplo, a partilha de anúncios em plataformas *online* é já uma funcionalidade *standard* do mercado. Apesar disso, não há qualquer tipo de consideração por esta na solução atual.

Deste modo, após uma análise por parte da empresa, concluiu-se que a aplicação existente se encontra desatualizada. Além disso, a empresa depreendeu ser mais vantajoso o desenvolvimento de uma nova aplicação, cujos aspetos mais importantes são:

1. Estruturas Organizacionais

Uma vez que a solução deverá poder ser utilizada por qualquer organização, há a necessidade de garantir a existência de uma estrutura de dados capaz de representar qualquer estrutura organizacional.

Assim, deve-se assegurar que a estrutura a ser desenvolvida seja o mais simples possível e que a sua adopção não resulte na deterioração da *performance*.

2. Conceitos Chave do Domínio

Existem conceitos chave dentro do domínio de RS que não se encontram presentes no modelo adoptado pela aplicação actualmente utilizada.

Por exemplo, a distinção entre candidato e candidatura e a distinção entre um pedido de recrutamento e um processo de recrutamento.

3. Fluxos de Aceitação/Processo Dinâmicos

Para poder iniciar um processo de RS, é necessário fazer primeiro um pedido. Ambos (pedido e processo) irão seguir fluxos de trabalho próprios. Fluxos esses que estarão dependentes das características do pedido (e.g. cargo de quem fez o pedido) e do processo (e.g. função para qual o processo foi criado).

Na solução actual, não existe nenhum mecanismo que permita dinamizar os fluxos de trabalho. Estando os processos associados a um estado que segue a mesma lógica para todos os clientes do software.

4. Publicação de Anúncios

Com a contínua adopção de práticas associadas a *e-recruitment*, é necessário não ignorar a integração que terá de ser feita com as diversas plataformas *online* cujo propósito é a partilha de vagas de emprego.

Adicionalmente, deverá estar prevista e assegurada a publicação de vagas de emprego em sites institucionais. Isto é, sites feitos à medida de cada empresa, onde se irão encontrar todos os processos de recrutamento que ainda aceitem novas candidaturas.

1.3 Objetivos

O objetivo primordial do presente trabalho consiste na análise, desenho e desenvolvimento de uma nova aplicação de RS que satisfaça as necessidades/requisitos da CP-CIT4ALL e dos seus clientes.

Aquando do desenho da nova solução, além dos problemas identificados previamente, deve-se ter em consideração as possíveis migrações que possam existir da versão antiga para esta mais recente. Sendo que, deve ser garantido que a execução de uma migração não resulte na perda de dados relevantes.

Além disso, numa primeira instância, os requisitos enunciados visam satisfazer as necessidades de um cliente específico. No entanto, sempre que possível, generalizar-se-á o processo para permitir satisfazer diferenças existentes para outros clientes.

Considerando os tópicos expostos anteriormente, a abordagem preconizada segue a seguinte sequência:

- Investigar sobre o domínio de RH, mais especificamente do processo de RS.
- Estudo da aplicação legada;
- Estudo das diferentes abordagens, métodos, ferramentas e tecnologias adotadas para o desenvolvimento deste género de sistemas.

- Eliciar junto dos *stakeholders* do sistema os novos requisitos do sistema e sua priorização.
- Propor uma nova solução que responda às novas necessidades de negócio.
- Construção e validação do sistema de acordo com a proposta.

1.4 Metodologia de Trabalho

Para o planeamento e o desenvolvimento do projeto, seguiu-se a metodologia *Kanban*, cuja abordagem assenta numa metodologia ágil, iterativa e incremental.

Para a gestão das tarefas foi usado um produto interno da CPCIT4ALL. Este permite uma mais fácil gestão do tempo e da equipa através da definição de *tickets*. Cada *ticket* descreve uma *User Story* específica que deve ser implementada.

Além disso, a ferramenta permite ao cliente enviar diretamente *tickets* associados a um colaborador (e.g. numa situação onde um erro foi encontrado), facilitando a comunicação com o cliente. Permite, ainda, controlar o *Backlog* do projeto e manter um registo atualizado das atividades desenvolvidas ao longo do mesmo.

Numa fase inicial do desenvolvimento, o *feedback* do cliente será recebido em reuniões. Reuniões essas onde se esclarecem as dúvidas que vão surgindo ao longo da análise dos requisitos e do desenho da solução.

Contudo, numa fase posterior, onde o desenho da solução já está mais consolidado e a implementação passa a ser o foco principal, as exigências do cliente passam a ser feitas, na sua maioria, através da criação de *tickets*.

Não obstante, se surgirem dúvidas nos *tickets* criados pelo cliente, é marcada uma reunião de esclarecimento com o mesmo. Ao fim de cada reunião é elaborado um documento que detalha as especificações adicionadas/alteradas. Ficheiro esse, que é partilhado pela equipa, de modo a manter sempre todos informados das últimas decisões tomadas.

1.5 Estrutura do Documento

O relatório está dividido em cinco capítulos. No primeiro capítulo expõe-se o necessário à compreensão do documento, inclusive a estrutura do mesmo.

No segundo capítulo, apresenta-se o domínio do projeto e é feito o levantamento e a devida comparação com aplicações semelhantes à que se pretende desenvolver.

No terceiro capítulo, é elaborado o estado da arte.

No quarto capítulo é realizada a análise de valor do projeto descrito.

No quinto capítulo é feito o levantamento de requisitos.

No sexto capítulo são apresentadas as abordagens, as alternativas e o desenho do sistema, incluindo as vistas mais relevantes à sua generalização.

O sétimo capítulo foca-se na implementação da solução para o problema descrito.

No oitavo capítulo, são descritas as experiências e avaliações que foram realizadas à solução apresentada.

Por fim, o nono capítulo conclui este documento, fazendo uma pequena reflexão do trabalho desenvolvido.

Capítulo 2

Contexto

O presente capítulo começa por descrever em mais detalhe o domínio da aplicação a ser desenvolvida. Segue-se a apresentação de soluções concorrentes e uma análise geral de como a solução atual se encontra desenvolvida. Por fim, apresenta-se a comparação das soluções abordadas.

2.1 Domínio de Aplicação

Armstrong (2006) define a gestão de RH como “a abordagem estratégica e coerente adoptada para a gestão dos recursos mais valiosos de uma organização - as pessoas que lá trabalham que individualmente e colectivamente contribuem para a realização dos seus objectivos”¹.

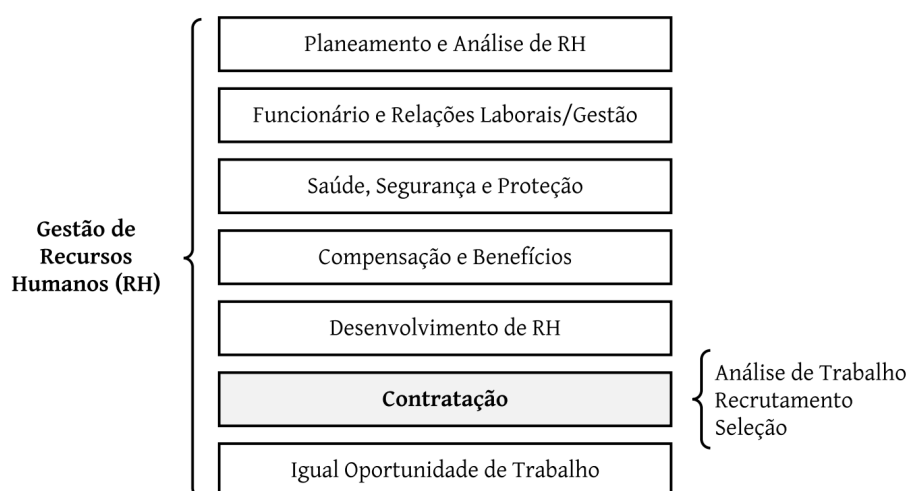


Figura 2.1: Áreas de Atividade da Gestão de RH
[Adaptado de (Robert L. Mathis, 1999)]

¹Tradução livre do autor. No original “... a strategic and coherent approach to the management of an organization’s most valued assets - the people working there who individually and collectively contribute to the achievement of its objectives.”

No entanto, esta definição não é suficiente para se conseguir identificar as funções concretas do departamento de RH. Para tal, recorre-se à divisão feita por Robert L. Mathis (1999) (Figura 2.1), onde há a divisão do contexto em múltiplas áreas de atividade, cada uma com as suas responsabilidades:

- **Planeamento e Análise de RH** - antecipar as forças que poderão influenciar a futura procura e oferta de funcionários; garantir o uso de sistemas de informação de gestão de RH adequados; analisar e avaliar a eficácia das políticas adoptadas pelos RH.
- **Funcionário e Relações Laborais/Gestão** - desenvolver, comunicar e actualizar políticas e regras de RH para garantir que os gestores e os empregados saibam o que é esperado deles; abordar os direitos dos empregados; abordar as relações sindicais/de gestão (em algumas organizações).
- **Saúde, Segurança e Protecção** - eliminar a possibilidade de acidentes e ferimentos no trabalho; promover a saúde e o exercício físico; assegurar que os colaboradores possam trabalhar num ambiente seguro.
- **Compensação e Benefícios** - desenvolver e aperfeiçoar os sistemas salariais; garantir que se recompensam as pessoas pelo seu trabalho através da remuneração, incentivos e benefícios;
- **Desenvolvimento de RH** - orientar novos funcionários; garantir a formação de aptidão profissional; encorajar o desenvolvimento dos funcionários; identificar os caminhos e actividades que os funcionários poderão seguir à medida que se desenvolvem na organização (*Career Planning*); avaliar a performance dos colaboradores da empresa (*Performance Managment*).
- **Contratação** - analisar as funções exercidas pelos funcionários da empresa e preparar as descrições e especificações das funções para recrutar novos candidatos; garantir um número adequado de indivíduos para preencher os postos de trabalho de uma empresa (Recrutamento); seleccionar as pessoas mais adequadas dentro da lista de candidatos obtida (Seleção).
- **Igual Oportunidade de Trabalho** - garantir o cumprimento das leis e regulamentos sobre igualdade de oportunidades de emprego.

Deste modo, entende-se que, além do departamento de RH se dividir em várias áreas de atuação, elas estão interligadas. A título de exemplo, uma mudança feita no *Planeamento e Análise de RH* poderá ter implicações a nível da *Contratação*. Se tal for o caso, surge a necessidade de informar a área de *Compensação e Benefícios* e a de *Desenvolvimento de RH* das alterações. A primeira, no sentido de alertar a possíveis discrepâncias salariais e a segunda no sentido de assegurar que a *Performance Managment* é feita com essa discrepância em mente.

Posto isto, o presente documento foca-se, apenas, na área afeta às contratações, uma vez que nessa se encontram tanto a responsabilidade do processo de recrutamento, como a do processo seleção de novos colaboradores.

2.1.1 Recrutamento e Seleção

Segundo Armstrong (2006), o propósito do processo de RS será o de “obter a um custo mínimo o número e a qualidade de empregados necessários para satisfazer as necessidades de recursos humanos da empresa”².

Posto isto, é necessário perceber aquilo que é a generalidade dos processos de RS. Para tal, apesar da existência de vários modelos válidos, seguiu-se a divisão feita conforme Armstrong (2006):

1. Definição de Requisitos

Nesta etapa deve ser definido o número de pessoas a contratar e os papéis a desempenhar.

Adicionalmente, poderá haver pedidos para novos postos de trabalho ou para substituições. Pedidos estes que deverão ser validados para garantir que são justificados.

Os requisitos para cada posição são definidos na forma de perfis de função e especificações pessoais. Um perfil deverá conter a informação que será utilizada para filtrar os candidatos inicialmente (e.g. competências necessárias, habilitações académicas, experiência profissional, etc.).

2. Atração de Candidatos

Nesta etapa a preocupação deverá ser identificar e avaliar quais as fontes de candidatos mais apropriadas para a partilha de cada vaga. O método mais evidente será a partilha de anúncios em diversas plataformas.

É de notar que este passo poderá nem sempre ser justificado.

3. Seleção de Candidatos

Esta última etapa é a mais extensa, sendo que pode ser, também, dividida em múltiplas etapas (que seguem, por norma, a sequência apresentada):

- (a) *Receber Formulários* - Primeiro passo será a receção dos formulários dos candidatos à vaga;
- (b) *Filtrar Candidaturas* - Utilizando os dados recebidos na fase anterior, são extraídas apenas as candidaturas que cumpram os requisitos mínimos definidos;
- (c) *Entrevistar Candidatos* - Nesta etapa ocorrem as entrevistas. Sendo que, pode haver mais que uma por candidato (e.g. entrevistas individuais, entrevistas de grupo, entre outras);

²Tradução livre do autor. No original “... to obtain at minimum cost the number and quality of employers required to satisfy the human resources needs of the company.”

- (d) *Testar/Avaliar Candidatos* - Fase onde se testam os candidatos ainda presentes no processo. Destes são exemplo: Testes psicológicos; Testes de competências em entrevistas; Testes de competências através da delegação de tarefas (num determinado intervalo de tempo);
- (e) *Referências, Qualificação e Ofertas* - Por esta altura, uma oferta pode estar esboçada e devidamente informada aos candidatos. No entanto, esta estará sempre sujeita a ficar invalidada, dependendo da informação recolhida nas referências (empresas para as quais o candidato já trabalhou).

Esta fase deve ser feita apenas a um candidato de cada vez, isto porque, caso o primeiro não aceite há candidatos de reserva; se o primeiro ficar, não há necessidade de revogar uma proposta feita a outro candidato;

- (f) *Etapa Final* - Por fim, deve-se confirmar a oferta feita e assinar contrato. Posteriormente, o novo colaborador passa, geralmente, para a fase de *Onboarding* (que já não faz parte do processo de RS).

2.1.2 E-Recruitment

A digitalização constante tem-se vindo a fazer sentir em todas as áreas empresariais, não sendo os RH uma exceção. Com a evolução da tecnologia, as abordagens adoptadas pelas equipas têm vindo a dar um uso mais importante e central aos sistemas informáticos (Lina, 2019).

Na década de 90, segundo Lengnick-Hall e Moritz (2003), os sistemas de RH cujo funcionamento se fazia pela internet começaram a ser chamados de “e-HR”. Numa fase inicial, estes sistemas digitais providenciaram a oportunidade de aceder mais rapidamente à informação necessária, em qualquer hora, em qualquer local - necessitando apenas de acesso à Internet (Lengnick-Hall & Moritz, 2003).

No entanto, Lengnick-Hall e Moritz (2003) denotaram bem que, para a adoção generalizada deste tipo de sistemas, “não é suficiente ter a tecnologia necessária.”³ Tal só se verificaria quando houvesse a disponibilidade quase universal de computadores pessoais e a cultura informática fosse mais generalizada. Só aí, é que as organizações tirariam total partido das oportunidades oferecidas pelo avanço tecnológico.

O cumprimento dessas condições permitiu que as empresas financiassem o desenvolvimento de sistemas informáticos que, gradualmente, foram mudando o paradigma sobre o qual eles próprios atuavam. Isto é, com o avanço tecnológico, estes sistemas deixaram apenas de servir como acervos de informação e passaram a incluir outros aspetos como a automatização de processos e a definição de fluxos de trabalho (Lengnick-Hall & Moritz, 2003).

No contexto atual de RS, as mudanças sentidas no mercado levaram à procura mais intensa de mão de obra especializada (Marques, 2018). Isto, acoplado com a mudança das

³Tradução livre do autor. Do inglês “It is not enough to have the requisite technology.”

práticas adoptadas por parte das equipas de RH, resultou não só numa redução substancial dos custos, como no uso generalizado de ferramentas que garantem uma maior eficiência e eficácia dos processos abrangidos (Bell et al., 2006; Marques, 2018).

Desta adoção surge o termo E-Recruitment, que se refere a:

todas as actividades de recrutamento efectuadas *online*, incluindo anúncios de emprego em *websites* empresariais e comerciais, receção virtual de candidaturas, procura de candidatos e perfis de selecção em redes sociais, avaliações e entrevistas feitas através da internet.⁴ (Marques, 2018)

Os sistemas de E-Recruitment, além de integrados com sistemas de outras atividades dos RH, são compostos por múltiplas aplicações. Kulkarni e Che (2019) dividiu as ferramentas em três tipos distintos:

1. Programa de Agregação de Empregos

Este tipo de sistema tem como objetivo “encontrar e listar as vagas disponíveis no seu site e, ao fazê-lo, atrair um número suficientemente grande de potenciais candidatos que se poderão candidatar às vagas disponíveis.”⁵

Exemplos deste tipo de software: Indeed, LinkedIn e Google.

2. Programa de Avaliação de Candidatos

O propósito deste tipo de sistema será “avaliar os candidatos usando critérios e medidas predefinidas e[, a partir destas,] escolher o melhor do conjunto de candidatos”⁶ para a vaga a preencher.

Exemplos deste tipo de software: TestDome, eSkill.

3. Programa de Rastreamento de Candidatos (ATS)

Este tipo de sistema tem como objetivo “ajudar os recrutadores a lidar com as necessidades de recrutamento da empresa e acompanhar os candidatos.”⁷

Um Applicant Tracking System (ATS) permite, tipicamente, realizar as seguintes tarefas:

- (i) Criar e anunciar vagas em aberto;
- (ii) Recolher currículos;
- (iii) Criar uma lista de candidatos;

⁴Tradução livre do autor. No original “[...] all recruitment activities done online, including job postings in corporate and commercial websites, receiving applications virtually, searching for candidates and screening profiles in social networks, and web-based assessments and interviews.”

⁵Tradução livre do autor. No original “... to find and list available jobs on their site, and by doing so, attract a sufficiently large number of potential applicants who may then apply for those job opens.”

⁶Tradução livre do autor. No original “... to assess the candidates using preset criteria and measures and pick the best from the set of candidates.”

⁷Tradução livre do autor. No original “... help recruiters handle the recruitment needs of the company and keep track of candidates.”

- (iv) Agendar entrevistas de candidatos selecionados;
- (v) Gerir o processo de entrevista;
- (vi) Fazer ofertas e facilitar o processo de integração do candidato selecionado.

Exemplos deste tipo de software: Manatal, TRAFFIT e Teamtailor.

Na Figura 2.2, apresentam-se, de um modo generalizado, as relações que cada fase mantém com as várias aplicações usadas aquando do processo de RS.

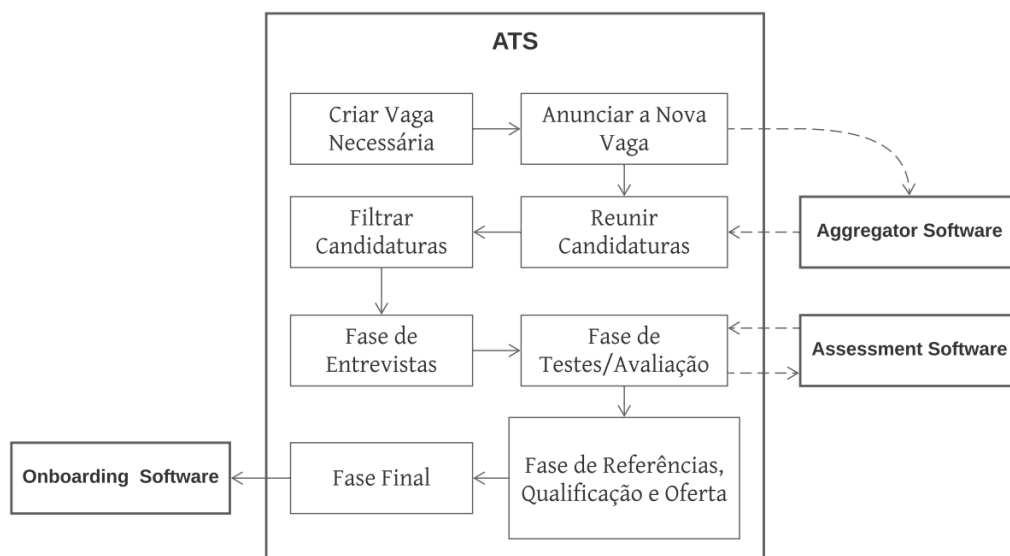


Figura 2.2: Sistemas Presentes no Fluxo Geral do Processo de Recrutamento e Seleção
[Adaptado de (Kulkarni & Che, 2019)]

Como se pode observar pela Figura 2.2, o processo de RS não se restringe à utilização de apenas uma aplicação, apesar de se reconhecer que o ATS tem um papel mais essencial na automatização dos processos.

Os ATS devem suportar a personalização de cada um dos passos apresentados na figura anterior. Adicionalmente devem, também, permitir a integração com sistemas de comunicação de modo a permitir automatizar o diálogo com os candidatos (e.g. através de emails automáticos). Estas medidas tornam todo o processo de RS mais eficiente de um ponto de vista da gestão de RH (Kulkarni & Che, 2019).

Outra das vantagens dos ATS é a centralização dos dados referentes aos processos. Dado estes que estarão acessíveis tanto para os recrutadores, como para os gestores das contratações; o que resulta numa melhor colaboração dentro das organização (Kulkarni & Che, 2019).

A solução que se pretende desenhar e implementar ao longo do presente documento consiste num ATS.

2.2 Soluções Concorrentes

Nesta subsecção são apresentadas e descritas plataformas cujo propósito é semelhante à solução a ser desenvolvida.

2.2.1 Manatal

O Manatal é um ATS baseado em SaaS (Manatal, 2021).

No que respeita as funcionalidades, permite: (1) gerir candidatos; (2) gerir processos de recrutamento; (3) gerir os eventos realizados com os candidatos; (3) criar e editar fluxos de trabalho; (4) publicar anúncios. Tem integração com plataformas de email e com plataformas de publicação de anúncios.

Esta ferramenta destaca-se pelo uso de inteligência artificial para dar recomendações. Para tal, extrai as competências necessárias de uma vaga e cruza as referências dos resultados com a lista de candidatos existentes no sistema para verificar o que melhor se enquadram na posição.

2.2.2 TRAFFIT

Traffit é, também, um ATS baseado em SaaS (Traffit, 2021).

No que respeita as funcionalidades, assegura todas as listadas na solução anterior. Além disso, também fornece um módulo CRM (Customer Relationship Managment) que permite comunicar de maneira mais eficiente com o utilizador.

2.2.3 Teamtailor

Teamtailor é, também, um ATS baseado em SaaS (Teamtailor, 2021).

No que respeita as funcionalidades, assegura todas as listadas na primeira solução. No entanto, este sistema foca mais a promoção das vagas existentes. De todos, este é o que mais promove a criação de páginas específicas para a partilha de anúncios e a sua customização.

2.3 Solução Atual CPCIT4ALL

A solução da empresa para o processo de RS faz parte de um conjunto de aplicações desenvolvidas para as várias áreas de atividade dos RH.

Tal como as soluções apresentadas anteriormente, a solução também é *cloud-based*. No entanto, esta não pode ser considerada um ATS, uma vez que as funções não são suficientes para tal.

Esta solução permite que os colaboradores (apenas) possam criar processos, candidaturas e eventos, associados entre si. Adicionalmente, permite o envio de emails para os candidatos.

No entanto, o registo das candidaturas tem de ser feito manualmente pela equipa de RH. Além disso, não há a possibilidade de definir fluxos dinâmicos. Enquanto que as soluções anterior permitiam a criação e edição dos fluxos, nesta os processos têm uma série de estados que deverão seguir.

2.3.1 Desenho Atual

A presente subsecção contém parte do desenho arquitetural do módulo atual de RS. Para a representação do desenho decidiu-se aplicar uma abordagem baseada na junção de dois modelos.

Primeiro, o Modelo C4 de Brown (Brown, 2021), criado com o propósito de facilitar a compreensão da arquitetura de software através do uso de diferentes níveis de detalhe:

- *Nível 1* - expõe o contexto do sistema no meio envolvente;
- *Nível 2* - mostra o contexto de *container* ⁸;
- *Nível 3* - representa o contexto de componentes ⁹;
- *Nível 4* - descreve o contexto de componente.

Segundo, o Modelo de vistas 4+1 de Kruchten (Kruchten, 1995), usado para descrever e demonstrar o sistema de várias perspetivas:

- *Vista Lógica* - resume as funções e relações entre os componentes do software. Esta vista pode ser representada por um diagrama de componentes;
- *Vista de Processos* - demonstra as responsabilidades e colaborações dos elementos lógicos. No contexto presente será apresentada na forma de diagrama de sequência;
- *Vista de Implantação* - implantação física de processos e componentes para nós de processamento e a rede física entre nós;
- *Vista de Cenários* - resume os casos de uso com maior importância arquitetural e os seus requisitos não funcionais;
- *Vista de Implementação* - mostra a dependência e acoplamento entre *packages* do *software* numa perspetiva de construção, bem como a enunciação e descrição de tecnologias e condições de implementação.

No contexto de apresentação da solução já existente, foram adotadas duas das cinco vistas: vista lógica e vista de processos. Estas serão aplicadas ao nível 2 do modelo C4 de Brown, de modo a ter uma ideia geral de como está desenhada a solução.

⁸Container: “um contexto ou limite no qual algum código é executado ou algum conjunto de dados é guardado. Cada contentor corresponde a uma implantação/execução distinta ou a um ambiente de execução diferente que normalmente (mas não sempre) corre no seu próprio espaço de processamento.” (Brown, 2021)

⁹Componente, no contexto do modelo: “corresponde a um grupo de funcionalidades encapsuladas numa interface bem definida.” (Brown, 2021)

Vista Lógica

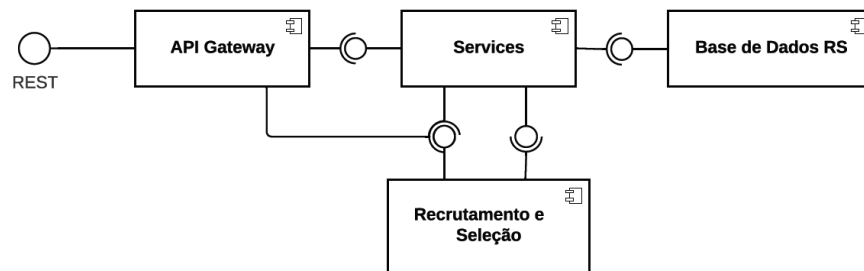


Figura 2.3: Vista Lógica da Solução atual

Na Figura 2.3 encontram-se os componentes existentes, actualmente, no módulo de recrutamento e seleção. As funcionalidades encontram-se expostas por uma API REST fornecida pelo API Gateway. Este componente está responsável por reencaminhar os pedidos para os Serviços que, quando necessário, acedem ao módulo de Recrutamento e Seleção e à respetiva base de dados.

Vista de Processos

Uma vez que a estrutura dos processos é idêntica na maioria das situações, apresentam-se, de seguida, os diagramas de sequência de como se obtêm as páginas web e como estão estruturadas as operações Create, Read, Update and Delete (CRUD), nas Figuras 2.4 e 2.5, respetivamente.

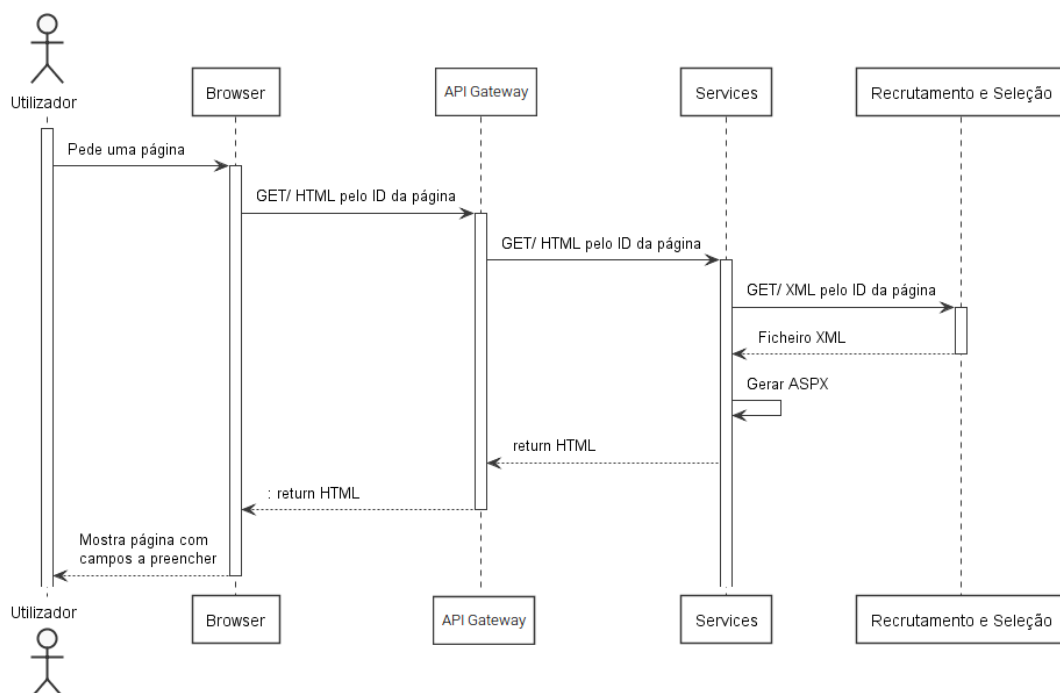


Figura 2.4: Diagrama de Sequência Obter Página HTML

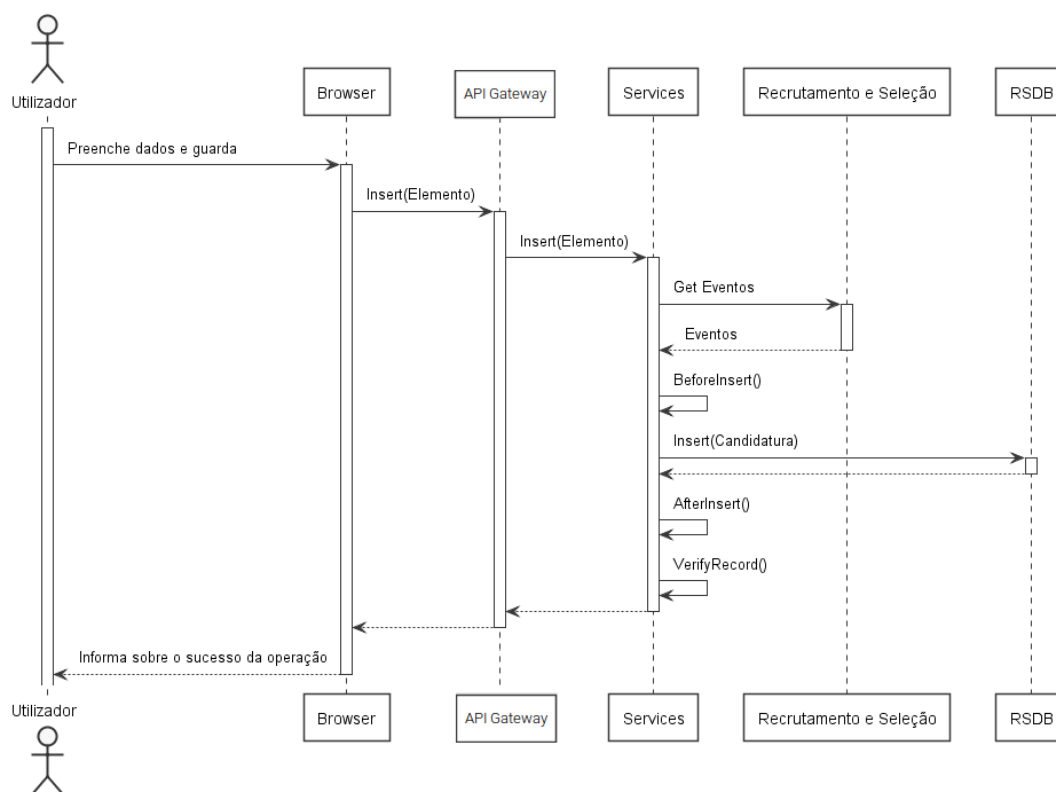


Figura 2.5: Diagrama de Sequência Inserir Elemento

A Figura 2.5 apresenta apenas uma das operações CRUD. No caso de serem outras, a lógica seguida será a mesma. O pedido segue até aos serviços e estes consultam o componente recrutamento e seleção para saber quais os eventos a despoletar.

Neste contexto os eventos referem-se ao *BeforeInsert*, *AfterInsert* e *VerifyRecord*. Os dois primeiros definem uma função a correr antes e depois da operação, respetivamente; o último serve para colocar condições que devem ser mantidas após qualquer operação, sendo que esta função é partilhada por todas as operações de uma determinada tabela. Isto é, enquanto que os eventos *Before* e *After* irão variar dependendo da operação a ser feita (Create/Update/Delete), o *VerifyRecord* é sempre a mesma função em todas as operações.

Apesar de estarem todos a correr no diagrama apresentado, se nenhum estiver definido, o sistema não irá sequer chamar a função. Tudo dependerá sob qual tabela se está a realizar a operação e quais os eventos definidos para a mesma.

Por fim, a operação *Read* não tem eventos como as restantes, correspondendo simplesmente à *query* feita à base de dados por parte dos serviços.

2.4 Comparação das Soluções

A comparação das soluções encontra-se apresentado na tabela 2.1. Por uma questão de legibilidade, os sistemas encontram-se representados pelas suas iniciais, salvo os sistemas que têm a mesma inicial (M - Manatal; Tr - Traffit; Te Teamtailor; SA - Solução

Atual; NS Nova Solução).

A escolha das propriedades a comparar foi feita tendo em conta aquilo que a empresa pretende desenvolver a nova aplicação. A gestão de candidatos, pedidos, processos e respetivos eventos são funcionalidades essenciais a qualquer ATS.

Mais, dependendo do tipo de evento a ser realizado, deve ser apresentada uma página com diferentes campos para preencher. Deve, também, permitir que se definam fluxos dinâmico, tanto para os pedidos, como para os processos.

Por fim, deve-se garantir a integração com e-mail, a integração com calendário e a publicação de anúncios, tanto em sites institucionais, como em redes sociais externas.

Tabela 2.1: Tabela de Comparação das Soluções

Propriedade	M	Tr	Te	SA	NA
Gestão de Candidatos	✓	✓	✓	✓*	✓
Gestão de Pedidos	x	x	x	x	✓
Gestão de Processos	✓	✓	✓	✓	✓
Gestão de Eventos	✓	✓	✓	✓	✓
Páginas Diferentes por Tipo de Evento	✓	✓	✓	x	✓
Fluxos de Trabalho Dinâmicos	✓	✓	✓	x	✓
Integração com Email	✓	✓	✓	✓	✓
Integração com Calendário	✓	✓	✓	x	✓
Publicação de Anúncios em Sites Institucionais	✓	✓	✓	x	✓
Publicação de Anúncios em Aplicações Third-Party	✓	✓	✓	x	✓

* O registo de uma candidatura é, sempre, feito por alguém da empresa.

Analisando a tabela 2.1, verifica-se que todos os sistemas permitem que uma candidatura possa ser registada tanto por um candidato, como por um colaborador, à exceção do atual. Além disso, todos permitem a definição da informação que aparece nas páginas dos eventos, que varia pelo tipo de evento - também não suportado pela solução atual.

Adicionalmente, a definição de fluxos de trabalho dinâmicos e a integração com sistemas de email e calendário são algo presente nos sistemas analisados. A nível da partilha dos anúncios, também todas contêm suporte (tanto em páginas próprias, como em aplicações *third-party*).

Contudo, nenhuma das soluções aborda a gestão de pedidos (pelo menos explicitamente). Sendo esta uma das funcionalidades mais importantes para o sistema que se deseja desenvolver.

Capítulo 3

Estado da Arte

Nesta secção começa-se por enquadrar a abordagem adoptada no desenvolvimento de sistemas semelhantes àquele que se pretende desenvolver. Depois, focam-se as tecnologias e as práticas adoptadas dentro dessa mesma abordagem.

3.1 Enquadramento Histórico

Na figura 3.1, apresenta-se uma possível classificação dos sistemas de informação em diferentes camadas.

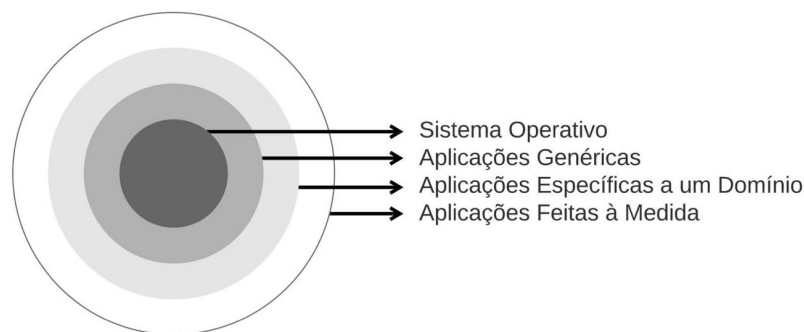


Figura 3.1: Tendências Relevantes a Business Process Management (BPM) [Adaptado de (Van der Aalst, 2004)]

Esta divisão, apresentada por Van der Aalst (2004), segue a seguinte lógica:

- a primeira camada representa os sistemas operativos - que conectam o hardware ao software;
- a segunda camada representa as aplicações genéricas - que podem ser utilizadas em vários domínios (e.g. gestor de bases de dados, editor de texto, etc);

- a terceira camada representa as aplicações específicas a um domínio (e.g. sistemas de recursos humanos);
- a quarta, e última, camada compreende apenas as aplicações feitas à medida para uma determinada organização.

No início do desenvolvimento dos sistemas informáticos, a maioria das aplicações poderiam ser alocadas na última camada, dado que não haviam muitas ferramentas genéricas que auxiliassem o desenvolvimento de aplicações. Contudo, atualmente, são menos as aplicações que são desenvolvidas à medida. Isto porque, todas as camadas têm vindo a expandir e, por isso, contêm mais funcionalidades (e.g. sistemas de gestão de base de dados que, previamente, eram feitos à medida, são agora sistemas genéricos) (Van der Aalst, 2004).

Isto teve implicações na maneira como se implementam novos sistemas. Enquanto que, antes, o foco estaria no desenvolvimento (codificação) dos vários módulos de um sistema; hoje, o ênfase está mais voltado para a integração e orquestração dos vários sistemas existentes (Van der Aalst, 2004).

Esta mudança de paradigma acompanhou a mudança de uma abordagem centrada em dados, para uma abordagem centrada em processos (Van der Aalst, 2004). Os sistemas desenvolvidos, a nível empresarial, tendem a dar mais importância à definição e execução de processo de negócio, visando a melhoria do desempenho das organizações.

Assim, entra a gestão de processos de negócio (BPM) como algo bastante relevante no contexto atual de desenvolvimento de novas aplicações.

3.2 Gestão de Processos de Negócio

BPM pode ser visto como o “conjunto de princípios, métodos e ferramentas utilizados para o desenho, análise, execução e monitorização dos processos de negócio”¹ (Dumas et al., 2013). Apesar de não ser uma definição consensual, entende-se que a maioria das definições engloba os seguintes aspetos: desenho, análise, modelação, implementação e controlo dos processos de negócio (Reijers, 2003).

No sentido de se compreender melhor os aspetos mais relevantes da BPM, dividiu-se a presente subsecção em 3 partes. Inicialmente, apresenta-se a distinção entre dois conceitos que, muitas vezes, são usados como sinónimos, apesar do seu significado não ser necessariamente o mesmo (processo de negócio e fluxo de trabalho). Após isso, apresenta-se o ciclo de vida dos processos e, por fim, apresentam-se as notações usadas para representar processos de negócio.

3.2.1 Conceitos

A distinção entre processo de negócio e fluxo de trabalho vem da aplicação de conceitos semelhantes em diferentes contextos (Georgakopoulos et al., 1995).

¹Tradução livre do autor. No original “a body of principles, methods and tools to design analyze, execute and monitor business processes.”

Por um lado, um processo de negócio

consiste numa sequência de atividades. Tem *inputs* e *outputs* distintos e serve um propósito significativo para uma organização, ou entre organizações. Uma actividade é uma etapa de um processo executada, quer por uma máquina, quer por uma pessoa... [, podendo] consistir numa ou mais tarefas.² (Stohr & Zhao, 2001)

Por outro, um fluxo de trabalho

é a automação total ou parcial de um processo de negócio, durante o qual documentos, informações ou tarefas são passados de um participante para outro, de acordo com um conjunto de regras processuais.³ (Stohr & Zhao, 2001)

Por outras palavras, os fluxos de trabalho são vistos como concretizações informáticas de processos de negócio. Isto é, referem-se a propriedades técnicas dos mesmos, tais como os dados passados entre as diferentes tarefas. Enquanto que, os processos de negócio são um conceito mais geral que não compreende, necessariamente, as necessidades informáticas para a implementação do mesmo.

3.2.2 Ciclo de Vida dos Processos

A gestão de processos de negócio implica, além da identificação de processos ainda não detetados, uma avaliação constante de todos os processos já implementados e a sua alteração conforme as necessidades encontradas ao longo do desenvolvimento (Moraes et al., 2014). Isto porque, quer por fatores externos, quer por condições detetadas aquando da utilização dos sistemas, vão sempre surgir novas necessidades.

Numa primeira fase é importante compreender quais as partes interessadas que deverão fazer parte do desenvolvimento dos processos. Isto porque, a definição dos processos deverá incorporar as necessidades de todos os *stakeholders*.

Deste modo, e a título de exemplo, apesar de a direção, à partida, conhecer os fluxos de trabalho seguidos de uma forma geral; não têm toda a informação necessária daquilo que é feito dentro de cada atividade desse processo. Para obter tal informação deve-se falar diretamente com quem está encarregue de desempenhar essas mesmas atividades.

Seguindo o modelo apresentado por Dumas et al. (2013), as partes interessadas serão divididas da seguinte forma:

- **Direção** - Dependendo da organização das empresas, poderá ter as seguintes posições: CEO, diretor de operações, CIO, diretor financeiro e o diretor de RH. Têm como responsabilidade supervisionar todas as operações;

²Tradução livre do autor. No original "... consists of a sequence of activities. It has distinct inputs and outputs and serves a meaningful purpose within an organisation or between organisations. An activity is a discrete process step performed either by a machine or human agent. An activity may consist of one or more tasks."

³Tradução livre do autor. No original "... the automation of a business process in whole or in part, during which documents, information or tasks are passed from one participant to another according to a set of procedural rules."

- *Process Owner* - A pessoa responsável por um ou mais processos. Deve definir as métricas e os objetivos afetos ao processo e deve garantir o acesso aos recursos necessários para que o processo corra sem problemas. Além disso, trabalham com os participantes do processo no sentido de retificar quaisquer dúvidas que surjam aquando da definição ou até mesmo execução do processo;
- *Participantes do Processo* - Pessoas que executam diariamente as atividades do processo. Servem como especialistas durante a descoberta e análise do mesmo;
- *Analistas*. Responsáveis por conduzir as atividades de descoberta, análise e *re-design*. Coordenam, também, a implementação e a monitorização;
- *Engenheiros de Sistemas* - Trabalham com os analistas de modo a capturar as necessidades do sistema e são responsáveis pela sua implementação, pelos seus testes e devida implantação;
- *O grupo BPM* - Não existe necessariamente em todas as empresas. Mas, seria um grupo de pessoas com experiência em BPM, cuja responsabilidade passa por manter uma cultura BPM dando suporte aos analistas e aos *process owners*.

O desenvolvimento contínuo e permanente pode ser transcrito para um modelo do ciclo de vida da gestão de processos de negócio (BPM lifecycle) (Moraes et al., 2014). Seguindo o modelo elaborado por Dumas et al. (2013), a gestão de processos pode ser fragmentada nas seguintes fases:

- **Identificação do Processo** - Nesta fase, apresenta-se um problema empresarial e identificam-se e associam-se os processos de negócio relevantes para o problema em questão.

Disto, resulta uma arquitetura de processos nova ou atualizada (visão geral dos processos e as suas relações);

- **Processo de Descoberta (ou modelação do processo ‘as-is’)**⁴ - Nesta fase, o estado dos processos são documentados “tal como estão” implementados/concebidos na altura (*as-is*), tipicamente na forma de um ou vários modelos;
- **Análise do Processo**⁵ - Nesta fase, são identificadas as preocupações associadas aos processos *as-is*, levantados na fase anterior, documentadas e, sempre que possível, quantificadas utilizando métricas de desempenho.

Disto resulta um conjunto de questões, geralmente organizadas de acordo com o seu impacto no sistema, ou de acordo com o esforço necessário para serem abordadas;

- **Redesenho do Processo** - Nesta fase, o objetivo é reconhecer quais as alterações necessárias para abordar as questões levantadas na fase anterior. Estas devem

⁴Na literatura também tem o nome de *Design do Processo*. No entanto, julga-se que o termo ‘descoberto’ seja o mais adequado, uma vez que o processo já existe implicitamente na cabeça de todos os intervenientes. O objetivo desta fase é, geralmente, descobrir o processo e não desenhá-lo (Dumas et al., 2013)

⁵Apesar de, neste contexto, a identificação das medidas de desempenho estar na fase de análise, em alguns casos é feito em paralelo com a identificação do(s) processo(s).

sempre garantir que os processos estão dentro das métricas previamente estabelecidas, sem nunca comprometer o objetivo da sua implementação.

Disto, resulta, tipicamente, um modelo de processos a ser seguido nas fases posteriores (*to-be*);

- **Implementação do processo** - Nesta fase, são executadas as alterações necessárias para passar do processo *as-is* para o processo *to-be*. A implementação das alterações implica gerir as mudanças que vão ser efetuadas na empresa a nível dos participantes do processo, e implica, ainda, o desenvolvimento e implementação dos sistemas informáticos que suportem o processo a realizar.

Disto, resulta, a implementação do novo processo pela organização;

- **Monitorização e Controlo do Processo** - Nesta fase, visto que o novo processo já está a decorrer, são recolhidos e analisados os dados relevantes para determinar o desempenho do processo para fazer a comparação com as métricas de desempenho e os seus objectivos. São identificados *bottlenecks*, erros recorrentes ou desvios relativos ao comportamento pretendido e são empreendidas acções correctivas. Novas questões podem surgir, no mesmo ou noutros processos, exigindo que o ciclo seja repetido numa base contínua.

Na Figura 3.2, encontra-se a representação gráfica do BPM lifecycle descrito.

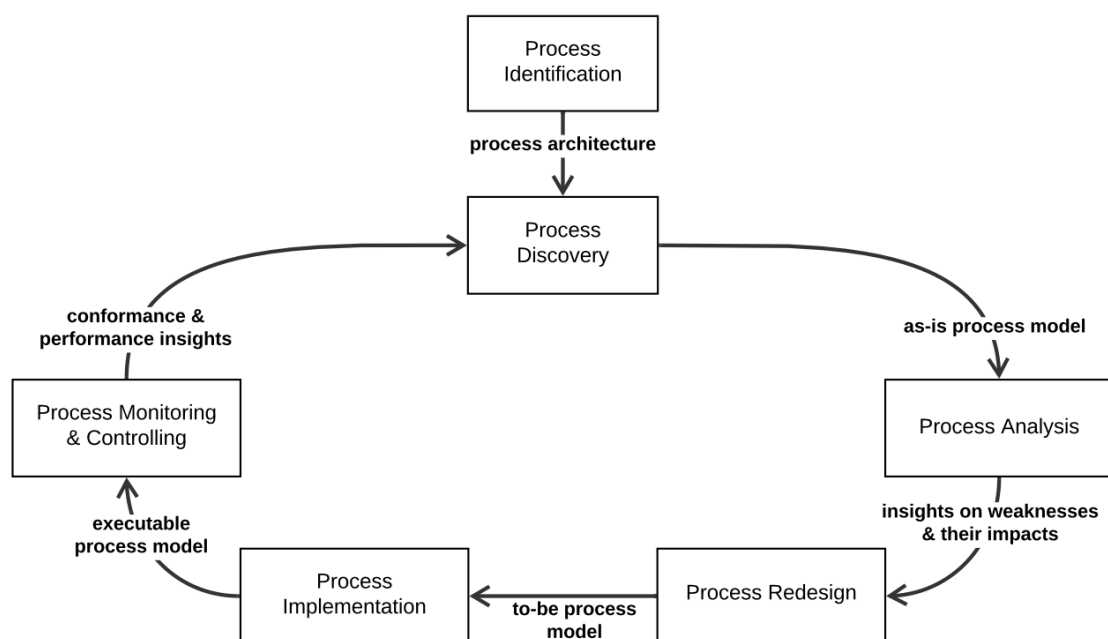


Figura 3.2: BPM Lifecycle [Adaptado de (Dumas et al., 2013)]

Este não é o único modelo proposto para representar um BPM lifecycle. Por exemplo, Morais et al. (2014) descreve o modelo apresentado como desatualizado. Considera até que “a gestão tradicional do processo carece da possibilidade de utilizar rapidamente os

conhecimentos obtidos pelos atores no decurso da execução.”⁶ Tendo isso em consideração, propôs o modelo representado na Figura 3.3.

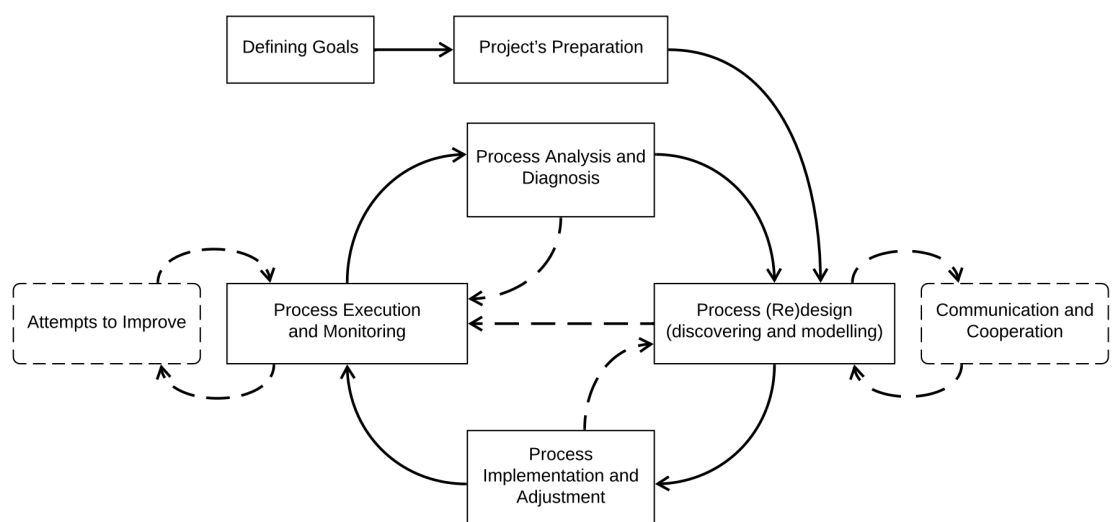


Figura 3.3: BPM Lifecycle for Dynamic BPM [Adaptado de (Morais et al., 2014)]

As etapas seguem a mesma lógica e sequência do primeiro modelo, no entanto, dão uma resposta mais rápida e, por sua vez, mais adequada quando se recorre a certos tipos de tecnologias associadas aos processos (e.g. *process mining* ou inteligência artificial) (Morais et al., 2014).

O modelo da Figura 3.3 acrescenta algumas atividades intermediárias para oferecer a possibilidade de alteração dos processos, sem ter que retomar o ciclo todo de novo. Além disso, permite regressar a fases anteriores caso se compreenda ser necessário.

3.2.3 Representação de Processos de Negócio

A modelação dos processos de negócio “é um pré-requisito para as organizações que desejem melhorar os seus processos de negócio, ou desejem envolver-se na gestão dos processos de negócio,”⁷ servindo “para descrever a forma como as organizações conduzem processos de negócios atuais ou futuros”⁸ (Indulska et al., 2009)

Tipicamente, um modelo de processos de negócio representa graficamente as atividades, os eventos e o fluxo a seguir acoplado com as operações lógicas necessárias. Informações adicionais, como, por exemplo, objectivos, riscos e métricas de desempenho, também podem estar incluídos (Indulska et al., 2009).

Desta forma, entende-se que os modelos são uma parte integral do BPM, uma vez que são instrumentos que facilitam e promovem a análise e desenho de sistemas orientados

⁶Tradução livre do autor. No original “... traditional process management lacks the possibility of quickly using knowledge obtained by the performers in the course of performance.”

⁷Tradução livre do autor. No original “is a fundamental pre-requisite for organizations wishing to engage in business process improvement or Business Process Management (BPM) initiatives.”

⁸Tradução livre do autor. No original “depict the way organizations conduct current or future business processes”

a processos (Indulska et al., 2009). Assim, nesta subsecção apresentam-se as notações/-modelos existentes para a representação dos processos de negócio.

Diagrama de Atividade

Segundo a OMG (2011b), Unified Modeling Language (UML) é uma linguagem com um âmbito muito amplo que cobre um vasto e diversificado conjunto de domínios de aplicação. Os seus conceitos podem dividir-se em 3 partes: (1) contexto estrutural; (2) contexto comportamental; (3) conceitos suplementares.

O Diagrama de Atividade UML (AD UML) é um dos diagramas desenvolvidos no contexto comportamental. É usado para descrever como as atividades são coordenadas para fornecer um serviço que pode estar em diferentes níveis de abstracção. Também é adequado para modelar a forma como um conjunto de casos de uso se coordenam para representar os fluxos de trabalho das empresas (OMG, 2011b; Visual Paradigm, 2021a).

Exemplo

A título de exemplo, e para servir de termo de comparação, é apresentado o AD UML de um processo simples. Este processo, representado na Figura 3.4, é iniciado quando um actor quer uma pizza e vão decorrendo uma série de atividades até que, ou a encomenda é cancelada, ou a encomenda corre bem e o ator acaba por conseguir comer.

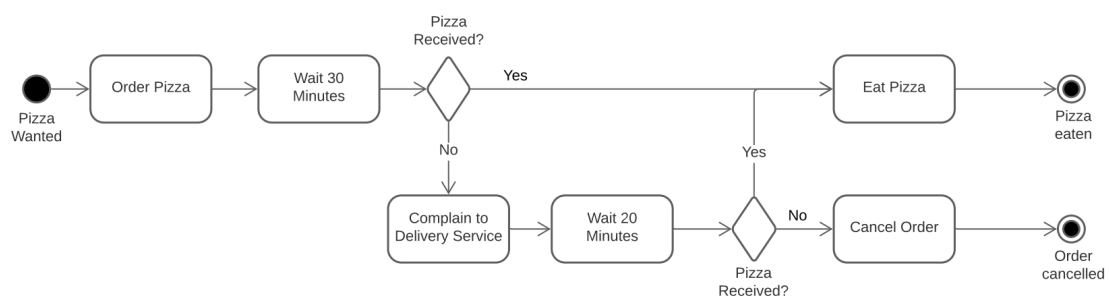


Figura 3.4: Ecomendar Pizza Exemplo - AD UML

Apesar de haver várias maneiras de representar um processo, este não poderá variar muito do que é apresentado. Sendo que, a maior alteração que se pode, eventualmente, verificar é o uso do símbolo de uma ampulheta para representar a espera em vez de ser uma actividade com o texto a especificar isso mesmo.

Business Process Model and Notation (BPMN)

O modelo BPMN, desenvolvido pelo, Object Management Group (OMG) é uma coleção de objetos gráficos que representa, maioritariamente, actividades, eventos e controlo do fluxo (Passos & Pereira, 2018).

O seu principal objetivo passa por fornecer uma notação facilmente compreensível por todos os atores envolvidos desde a concepção até à implementação dos vários processos de negócio (fluxos de trabalho). Deste modo, é possível ficar com uma ligação padrão entre o *design* e implementação dos mesmos (OMG, 2011a; Passos & Pereira, 2018).

É considerada a notação *standard* da indústria para representar processos de negócio.

Exemplo

Consideremos que o processo a ser representado na Figura 3.5 é o mesmo que foi descrito no AD UML (Figura 3.4). Apesar de ser possível organizar o processo da mesma maneira que se organizou no AD UML, decidiu-se apresentar o exemplo mais generalizado. Isto porque, pode-se apresentar partes mais específicas do BPMN. No entanto, é de reforçar a ideia de que, numa fase inicial de utilização do BPMN, a modelação será mais idêntica à apresentada na Figura 3.4.

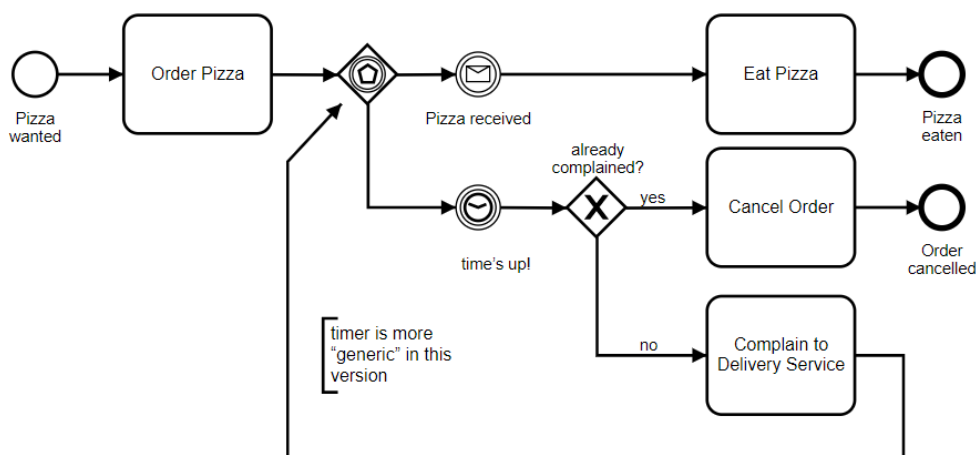


Figura 3.5: Ecomendar Pizza Exemplo - BPMN
[Fonte: (Camunda, 2021a)]

Neste exemplo, o estímulo inicial e a primeira atividade são os mesmos do exemplo anterior. Contudo, quando sai da atividade *order pizza*, o processo vai para um *event based gateway*. Isto significa que, o próximo passo a seguir dependerá de qual evento se realizar primeiro. Caso seja notificado da entrega, pode comer a pizza; caso passem os primeiros 30 minutos sem estar entregue, segue para a atividade para fazer a queixa. Após isso, volta para o mesmo *event based gateway*, onde estará sujeito aos mesmos eventos, a única diferença sendo que, caso o temporizador dispare outra vez antes da notificação, o processo seguirá para a atividade de cancelar a encomenda, pois já se realizou uma queixa.

Case Management Model and Notation (CMMN)

O CMMN, também proposto pelo OMG, é uma notação criada para complementar o BPMN (OMG, 2016; Passos & Pereira, 2018).

A notação deve ser usada em situações em que o foco principal do processo é o objetivo a cumprir, não a sequência de atividades a seguir (pois esta não é fixa). Proporciona uma maior liberdade aos atores do processo, uma vez que permite definir um conjunto de atividades que poderão, ou não, decorrer para alcançar o estado final pretendido (Passos & Pereira, 2018; Visual Paradigm, 2021b). Apesar disso, permite aplicar certas restrições

e condicionar o fluxo. Mas, em situações dessas, deve ser ponderado se a notação mais adequada não seria o BPMN.

Além disso, sendo o CMMN uma linguagem declarativa, “não se deve ler qualquer significado na posição relativa das formas gráficas [...], contrariamente ao BPMN que é [uma notação] imperativa”⁹.

Decision Model and Notation (DMN)

DMN foi, até à data, a última notação desenvolvida pelo OMG.

Segundo o OMG (2019), o DMN foi desenvolvido com o propósito de fornecer uma notação comum para a definição de decisões. Isto é, permite modelar a dependência entre critérios e definir qual o resultado da avaliação desses mesmos critérios. Permitindo que, numa situação em que se precise de avaliar múltiplos valores para saber qual a actividade seguinte a desempenhar dentro de um processo de negócio, o processo de decisão esteja descrito num diagrama próprio, facilitando a compreensão do mesmo.

Em DMN, a configuração é feita num diagrama que dita as dependências existentes para a decisão em relação com as tabelas de decisão. Esta pode ser feita independentemente, ou a par, da modelação em BPMN (Passos & Pereira, 2018). As tabelas de decisão deverão conter a lógica a seguir de acordo com a informação obtida. É uma notação de fácil compreensão para todos os atores envolvidos no processo de desenvolvimento. A notação DMN foi criada para ser utilizada com BPMN e CMMN.

Além de permitir que os processos de decisão possam ser entendidos por todas as partes envolvidas no desenvolvimento de novos sistemas, promove ainda uma maior automação dos processos, em geral (Passos & Pereira, 2018).

Comparação das Notações

Estudos (Peixoto et al., 2008; Van der Aalst, 2004), sugerem não haver uma diferença considerável na legibilidade de modelos de atividade UML e BPMN.

No entanto, no que concerne o mapeamento das linguagens de modelação de processos de negócio (BPML) para linguagens de execução de processos de negócio (BPEL), o BPMN oferece suporte, enquanto que o AD UML não define o mapeamento para qualquer BPEL.

3.3 Sistemas de Gestão de Processos de Negócio

Os sistemas (suites) de gestão de processos de negócio (BPMS) são ferramentas utilizadas para melhorar os processos de uma organização. Estas costumam possibilitar a modelação, execução, controlo e monitorização dos processos de forma automatizada.

Apesar de existirem muitos BPMS que possibilitam o desenvolvimento total de aplicações, no âmbito do projeto a desenvolver é mais relevante encontrar um sistema que seja

⁹Tradução livre do autor. No original “... one should not read any meaning into the relative positioning of graphical shapes [...], as opposite to BPMN that is imperative...”

facilmente integrado com as aplicações desenvolvidas pela empresa. Posto isto, foram analisados o Camunda BPM e o WSO2 Business Process Server.

3.3.1 Camunda BPM

Camunda BPM pode ser utilizado como um motor autónomo de processos de negócio, ou pode ser diretamente integrado em aplicações Java. Para todas as restantes linguagens de programação, a integração poderá ser feita através da API REST fornecida pelo motor autónomo (Camunda, 2021b).

É um projeto de código aberto e, como tal, não se fica dependente do fornecedor. Além disso, a empresa (Camunda) oferece uma versão empresarial, disponibilizando mais ferramentas que poderão ser utilizadas após um acordo de licenciamento com a mesma (Camunda, 2021b).

Adicionalmente, de acordo com Camunda (2021b), são disponibilizadas as seguintes ferramentas:

- Motor Camunda - motor de fluxos de trabalho desenvolvido em conformidade com os padrões ISO;
- Cawemo - permite a colaboração de vários utilizadores na definição de um processo;
- Camunda Modeler - permite aos programadores editar diagramas de processo BPMN e tabelas de decisão DMN, e distribuir fluxos de trabalho para os Motores Camunda para os executar.
- Camunda Optimize - fornece monitorização e a possibilidade de fazer relatórios.

3.3.2 WSO2 Business Process Server

O WSO2 Business Process Server (WSO2 BPS) é um produto de código aberto que serve como um *middleware*, fornecendo funcionalidades de BPM. Foi desenvolvido sobre a plataforma WSO2 Carbon e é baseado na estrutura OSGi (WSO2, 2021b).

O WSO2 Business Process Server (WSO2 BPS) fornece uma interface gráfica de utilizador que permite implementar, gerir e monitorizar processos de negócio e instâncias de processo, dentro de uma instância de servidor (WSO2, 2021a, 2021b). Permite, ainda, a gestão de *roles* que servirão como perfis de acesso às atividades de um processo.

Executa processos de negócio representados pela norma BPMN ou pela norma WS-BPEL, sendo que, para tal, contém dois motores distintos de fluxo de trabalho (WSO2, 2021b):

- Motor BPEL é alimentado pelo Apache Orchestration Director Engine (ODE);
- Motor BPMN 2.0 é alimentado pelo Activiti permitindo-lhe definir e executar modelos de processos de negócio BPMN 2.0.

A integração com aplicações externas pode ser feita através de uma REST API. Interface essa que permite a interação, tanto com os processos, como com as tarefas neles descritas (WSO2, 2021a).

3.4 Develop Framework 4 All

A *framework* DF4A foi desenvolvida por parte da CPCIT4ALL com o intuito de acelerar o desenvolvimento dos seus produtos. Está dividida em dois módulos distintos que representam diferentes responsabilidades:

- **Services**

A responsabilidade deste módulo passa por: garantir a persistência na base de dados; disponibilizar os serviços e processos comuns às várias aplicações; e implementar a lógica de negócio da aplicação a desenvolver.

Dos processos comuns temos, como exemplo: a autenticação dos utilizadores, o controlo de acessos, a gestão dos perfis de acesso, a integração com sistemas de comunicação (e.g. Email e SMS), entre outros.

- **Web**

Este módulo tem tanto a responsabilidade das *Views*, como a dos *Controllers*. É o motor responsável pela geração das páginas HTML que, neste momento, são desenvolvidas com o auxílio do Kendo UI.

Dentro dos serviços, existe um motor de fluxos de trabalho desenvolvido pela empresa. Apesar de disponibilizar uma integração simples quando acoplada com o resto da *framework*, encontra-se atrasada no que respeita o contexto atual de motores de fluxo de trabalho uma vez que não tem integração com BPMN.

No entanto, a possibilidade de configuração dos fluxos já esta assegurada pela DF4A através de páginas web.

Apesar de ter sido desenvolvida no ano de 2010, a *framework* - implementada com .NET Framework - tem tido constantes atualizações. Acompanhando, deste modo, o avanço tecnológico.

3.5 Comparação das Ferramentas

Para a comparação presente na tabela 3.1 teve-se em consideração o suporte das notações apresentadas previamente, a possibilidade da definição de permissões de acesso (dar para configurar perfis e limitar as atividades a determinados perfis) e o suporte para integrar com as aplicações da empresa.

Como se verifica pela comparação, no que respeita o desenvolvimento orientado a processos, a *framework* da empresa não seria a mais indicada pela inexistência de uma notação para representar os processos.

Tabela 3.1: Tabela de Comparação das Ferramentas

Propriedade	Camunda	WSO2 BPS	DF4ALL
Modelação BPMN	✓	✓	x
Modelação CMMN	✓*	x	x
Modelação DMN	✓	x	x
Definir Permissões	✓	✓	✓
Integração com DF4ALL	REST ou Java	REST	-

* Suporta parcialmente, não havendo planos para suportar a notação na sua totalidade.

Além disso, ambas as ferramentas têm suporte para a configuração das permissões de acesso. No entanto, o Camunda BPM contém um maior suporte de notações gráficas, enquanto que o WSO2 BPS oferece suporte para *WS BPEL* (linguagem padrão OASIS executável para especificar ações de processos de negócio com web services) o que se poderá mostrar vantajoso caso a integração seja feita por *web services*.

O Camunda suporta CMMN e DMN, ao contrário das restantes ferramentas. No entanto, o suporte ao CMMN é limitado pois a empresa notou que não era algo muito utilizado (Camunda, 2020) e apesar de se considerar uma vantagem, a experiência da empresa demonstra que não traz tanto valor quanto esperado.

Por fim, apesar de ser possível integrar o Camunda BPM com código Java, o desenvolvimento da empresa é feito noutra linguagem. Assim, ambas as ferramentas fornecem uma REST API que permite ligar às aplicações desenvolvidas pela CPCIT4ALL. Neste caso, a ferramenta da framework já se encontra integrada no sistema, sendo a única propriedade que a distingue positivamente.

Apesar de se ter utilizado a ferramenta da empresa no desenvolvimento da nova solução, o Camunda mostra ser, dentro destes critérios, a ferramenta com mais suporte em termos de notações para este tipo de sistemas e, por isso, a mais completa.

Capítulo 4

Análise de Valor

No presente capítulo será elaborada a análise de valor (VA). A VA é definida como um processo de análise sistemática ao desenho de produtos/serviços, a fim de comparar a suas funções requeridas (normalmente, por clientes) com os gastos impostos pelas mesmas (Nick & Holweg, 2000).

O objetivo deste processo é, iterativamente, tentar minimizar os custos, sem que o produto final veja o seu desempenho e/ou a sua fiabilidade prejudicados (Nick & Holweg, 2000; NPD, 2016).

Numa primeira fase apresenta-se a análise de valor do projeto recorrendo ao Novo Modelo de Desenvolvimento de Conceitos (NCD). Segue-se a definição dos conceitos de “valor”, “valor precetível” e “valor para o cliente” enquadrados no escopo do projeto.

Por fim, apresenta-se o modelo CANVAS para facilitar a visão geral do negócio e aplica-se o método Analytic Hierarchy Process (AHP), que, recorrendo a critérios qualitativos e quantitativos para avaliar as ideias geradas, facilita a compreensão e a avaliação do projeto.

4.1 Processo de Inovação

Como apresentado na figura 4.1, o processo de inovação pode ser dividido em 3 fases distintas (Koen et al., 2002; Koen et al., 2001).

1. Fuzzy Front End (FFE) ou Front End Engeneering (FEE);
2. Processo de desenvolvimento de novos produtos (NPD);
3. Comercialização.

O foco nas atividades de *Front End* precede o processo mais formal e estruturado do desenvolvimento de novos produtos. Desta forma, consegue-se aumentar o valor, a quantidade e a probabilidade de sucesso de novos conceitos altamente lucrativos, aquando da chegada a essa próxima etapa (Koen et al., 2002).

Deste modo, a primeira fase, FFE, é geralmente considerada uma das maiores oportunidades de melhoria de todo o processo de inovação (Koen et al., 2002). Contudo, verificando uma lacuna na terminologia utilizada na gestão do FFE, e desta surgir a necessidade de uma formalização, foi criado o modelo de desenvolvimento de novos conceitos (NCD) (Koen et al., 2001).

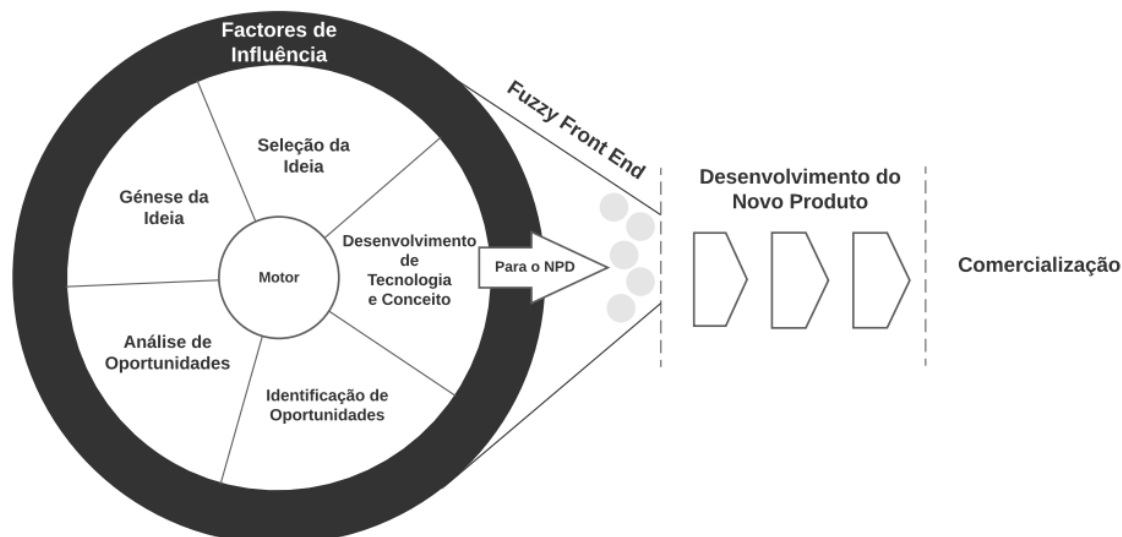


Figura 4.1: Processo de Inovação
[Adaptado de (Koen et al., 2002; Koen et al., 2001)]

Assim sendo, nesta subsecção, após uma exposição mais detalhada daquilo que constitui o NCD, encontra-se a aplicação deste mesmo modelo ao projeto apresentado.

4.1.1 Modelo de Desenvolvimento do Novo Conceito

Segundo Koen et al. (2001), e representado na figura 4.1 como a primeira etapa do processo, o New Concept Development (NCD) consiste em três secções distintas (Koen et al., 2002; Koen et al., 2001):

- *Motor* - parte central constituída pelos principais elementos que auxiliam o processo, correspondendo a atributos da organização, nomeadamente a gestão, visão, estratégia, recursos, cultura, equipas e colaboração.
- *Roda* - parte interior onde estão definidos os cinco elementos de atividade:

1. Identificação da Oportunidade¹

Elemento onde se identificam oportunidades que a organização poderá, potencialmente, perseguir. As oportunidades não se restringem a uma categoria, podendo passar por uma pequena atualização a um produto já existente, ou podendo ser a expansão para um nova área de atividade.

¹Oportunidade: um fosso empresarial ou tecnológico entre a situação actual e um futuro previsto, identificado por um indivíduo ou pela empresa, e que serve: para capturar uma vantagem competitiva; responder a uma ameaça; resolver um problema; ou melhorar uma dificuldade.

Estas oportunidades são consideradas para que haja uma eventual alocação de recursos em novas áreas rentáveis para a empresa (e.g áreas onde: o mercado esteja a crescer; a eficácia e eficiência de uma operação pode ser melhorada).

A essência deste elemento encontra-se na origem e nos métodos utilizados para identificar oportunidades que se desejam seguir.

2. Análise da Oportunidade

Elemento onde se obtêm informações adicionais sobre as oportunidades identificadas, de modo a confirmar (ou refutar) a validade do seguimento da oportunidade previamente identificada. Este elemento pode decorrer como um processo formal, ou resultar de iterações sucessivas de análise.

3. Gênese da Ideia²

Elemento onde se dá a formação, desenvolvimento e maturação de uma ideia. Alguns dos métodos utilizados incluem: contacto com clientes/utilizadores; colaboração com outras instituições; sessões de *brainstroming*; entre outros.

4. Seleção da Ideia

Elemento onde se fará a seleção de uma ideia. A seleção poderá ser a escolha de uma entre as várias alternativas identificadas e analisadas, ou pode resultar de um processo mais detalhado de escolha.

No entanto, é necessário perceber que a escolha não será tão rigorosa como noutros processos, uma vez que é necessário dar espaço para as ideias de desenvolverem e maturarem após esta fase.

5. Definição do Conceito³

Elemento final do modelo, onde se deve concluir através da construção de um caso apelativo a investimento na proposta de âmbito de negócio, ou tecnológica.

- *Aro* - parte externa que representa os fatores do meio ambiente que influenciam as restantes secções nos âmbitos: político, económico, social, tecnológico e legal.

Identificação da Oportunidade

A CPCIT4ALL atua, não exclusivamente, na área de Recursos Humanos, tendo uma série de programas (aos quais chamam de módulos) que atuam sobre diferentes responsabilidades dentro dessa mesma área. Deste modo, é possível conseguir parcerias com

²Ideia: a forma mais primordial de um novo produto ou serviço. Correspondendo, geralmente, a uma abstração daquilo que será a solução apresentada para o problema identificado pela oportunidade.

³Conceito: tem uma forma bem definida e inclui tanto uma descrição textual, como visual. Deverá ter identificadas as funcionalidades primárias e os benefícios que delas resultam para o cliente combinados com um conhecimento mais profundo da tecnologia necessária.

empresas mais facilmente, uma vez que não precisam de comprar o produto na sua totalidade.

Adicionalmente, e como é o caso aqui apresentado, um dos clientes que não tinha adoptado o módulo existente de recrutamento e seleção sugeriu o desenvolvimento de um software para esse mesmo efeito, mas com as características de um ATS (ver secção 2.1.2).

Análise da Oportunidade

Tendo em conta o contexto laboral atual, as pessoas que compõem uma empresa são bastante importantes. Ainda por cima, numa altura em que a própria filosofia adoptada por parte dos funcionários tem sofrido grandes alterações.

Antes, a relação pessoa-trabalho era algo mais definitivo, no sentido em que toda a ideia de sucesso estava relacionado com o crescimento que a pessoa poderia fazer dentro da sua empresa e a carreira que daí advinha. Contrariamente, nos dias de hoje, apesar de o sucesso ainda estar ligado à posição que uma pessoa ocupa em determinada empresa, a construção de uma carreira já não é o desejável, apenas os benefícios que advêm do emprego que se tem.

Assim, as empresas têm de se adaptar e conseguir responder a esta mudança. Podem adoptar duas estratégias opostas, mas complementares. Por um lado, podem atuar no sentido de promover uma cultura empresarial que capte e mantenha mais as pessoas; por outro, podem-se focar no desenvolvimento de um processo de recrutamento e seleção mais adequado às alterações verificadas no mercado.

Deste modo, sendo importante conseguir assegurar a rapidez e fluidez do processo de recrutamento e seleção de pessoas, surge a oportunidade de desenvolver sistemas que sejam mais adequados e acompanhem a mudança de paradigma dentro daquilo que são as práticas de recrutamento e seleção.

Adicionalmente, também é vantajoso para a própria empresa ter um módulo mais atualizado dentro deste contexto. Sendo que, dos dez clientes que utilizam módulo já existente, dois já foram apresentados com a ideia do produto que se deseja desenvolver e mostraram interesse pelo mesmo.

Génese da Ideia

No caso deste projeto, a ideia foi inicialmente lançada por um cliente da CPCIT4ALL, tanto em momentos formais e em momentos mais informais, pois tinha a necessidade de um software que permitisse dinamizar e automatizar partes do processo de recrutamento e seleção das várias empresas associadas a essa organização.

Contudo, não se pode ignorar que empresas mais pequenas não terão tantos recursos alocados para obter softwares da dimensão daquilo que se deseja desenvolver. Assim, torna-se importante perceber se é vantajoso, ou não, manter o módulo existente e desenvolver um completamente novo; ou se se deve atualizar o módulo já desenvolvido, correndo o risco de perder clientes com dimensões menores.

Seleção da Ideia

Há a possibilidade de se desenvolver uma aplicação completamente nova, ou atualizar a aplicação antiga. Apesar de ser uma escolha intuitiva, não é menos necessária a aplicação de um método de decisão que permita justificar as opções tomadas (Golden et al., 1989).

Deste modo, recorreu-se ao Analytic Hierarchy Process para poder apresentar quais os motivos que pesaram na decisão e como é que esses afetaram analiticamente a mesma.

Analytic Hierarchy Process

O Analytic Hierarchy Process (AHP) é um método de apoio à decisão multicritério (AMD) intuitivo e relativamente fácil de aplicar à formulação e análise de decisões (Golden et al., 1989). Desenvolvido por Thomas Saaty, o método permite que os responsáveis pela tomada de decisões cheguem à escolha mais lógica (Nguyen, 2014).

Para tal, a ideia principal do AHP é dividir o problema de decisão em níveis hierárquicos recorrendo a critérios qualitativos e/ou quantitativos no processo. Primeiramente, determinam-se as opções disponíveis para os decisores e os objectivos da decisão que terão de fazer. Estes objectivos e opções serão então utilizados para construir uma hierarquia analítica, que reflecte os vários factores no processo de tomada de decisão e a sua importância (Nguyen, 2014).

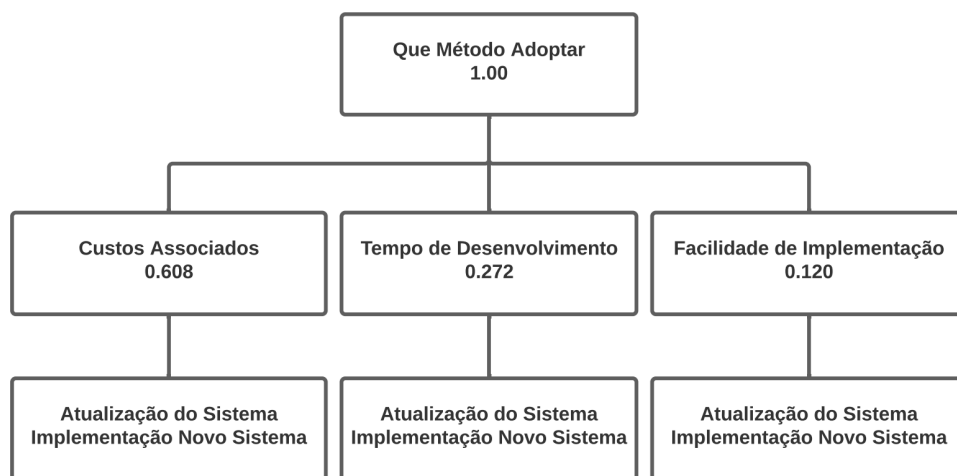


Figura 4.2: Árvore Hierárquica de Decisão

Primeiro, foi construída a árvore hierárquica de decisão da figura 4.2 composta por:

- *Objetivo* - Escolher o método a adotar;
- *Critérios* - Custos associados, tempo de desenvolvimento e facilidade de implementação;
- *Alternativas* - Atualização do sistema atual ou implementação de um novo sistema.

Na fase seguinte, foi feita a comparação entre os critérios identificados, presentes na tabela 4.1.

Tabela 4.1: Matriz de Comparação dos Critérios Seleccionados

	Custos associados	Tempo de Desenvolvimento	Facilidade de Implementação
Custos associados	1	3	4
Tempo de Desenvolvimento	1/3	1	3
Facilidade de Implementação	1/4	1/3	1

Através dos valores definidos (tabela 4.1), é possível calcular a prioridade relativa de cada critério (contas presentes no Anexo A.4).

Tabela 4.2: Vetor de Prioridades Relativas

Critério	Prioridade Relativa
Custos Associados	0.608
Tempo de Desenvolvimento	0.272
Facilidade de Implementação	0.120

Nesta etapa, é necessário avaliar a consistência das prioridades relativas, através do cálculo do rácio de consistência (RC). Estando este realizado (Anexo A.4), confirma-se que o RC é menor que 0.1, podendo dar os resultados como válidos e fiáveis.

Deste modo, podemos afirmar que os critérios, por ordem de maior a menor importância, são: custos associados; tempo de implementação; facilidade de desenvolvimento (dados da tabela 4.2).

O passo seguinte será a construção da matriz de comparação paritária para cada critério, considerando cada uma das alternativas seleccionadas.

Tabela 4.3: Matriz Custos Associados

	Atualizar Solução	Implementar Nova Solução	Prioridades
Atualizar Solução	1	1/3	1/2
Implementar Nova Solução	3	1	3/4

Tabela 4.4: Matriz Tempo de Desenvolvimento

	Atualizar Solução	Implementar Nova Solução	Prioridades
Atualizar Solução	1	1/3	1/2
Implementar Nova Solução	3	1	3/4

Tabela 4.5: Matriz Facilidade de Implementação

	Atualizar Solução	Implementar Nova Solução	Prioridades
Atualizar Solução	1	1/3	1/2
Implementar Nova Solução	3	1	3/4

Para o cálculo das prioridades apresentadas nas tabelas 4.3, 4.4 e 4.5, aplicou-se o mesmo método que anteriormente (normalização presente no Anexo A.5). A prioridade corresponde ao resultado entre a multiplicação da matriz prioridade e o vetor dos critérios:

$$\begin{bmatrix} 1/2 & 1/2 & 1/2 \\ 3/4 & 3/4 & 3/4 \end{bmatrix} * \begin{bmatrix} 0.608 \\ 0.272 \\ 0.120 \end{bmatrix} = \begin{bmatrix} 0.50 \\ \mathbf{0.75} \end{bmatrix}$$

Conclui-se, portanto, que a melhor opção será a implementação de um novo projeto, em função dos critérios apresentados.

Definição do Conceito

O objetivo do projeto passa por desenvolver um sistema que permita automatizar parcial ou totalmente tanto o processo, como as tarefas, presentes no âmbito do recrutamento e seleção de novos funcionários (um ATS, ver secção 2.1.2 para uma descrição mais detalhada do produto) que possa vir a substituir o módulo já existente de recrutamento e seleção da CPCIT4ALL.

4.2 Valor da Solução

O conceito de valor não tem nenhuma definição universal, uma vez que se entende que a percepção do valor de um produto depende de quem o analisa (Neap & Celik, 1999; NPD, 2016).

Numa perspetiva do vendedor, o valor pode ser comumente definido como a razão entre a função de um produto e o seu custo. Sendo que, a própria função pode ser desacoplada em várias características (e.g. performance e capacidade) (NPD, 2016).

$$Valor = \frac{Funcoes}{Custo} = \frac{Performance + Capacidade}{Custo}$$

Do ponto de vista de Nick e Holweg (2000), o valor pode ser dividido em 2 conceitos distintos:

$$Valor = Valor\ de\ Utilidade + Valor\ Estimado$$

O valor de utilidade estaria ligado ao valor adjacente à utilização do produto, enquanto que o valor estimado seria o valor proveniente da posse do produto. Deste modo, infere-se que há uma parte do valor mais objetiva (valor perceptível) e uma parte dependente da percepção do produto por parte do cliente (valor percepcionado).

Visto que o desenvolvimento deste projeto surgiu de uma proposta de um cliente, a empresa considerou que o valor negociado seria o suficiente para investir no desenvolvimento deste novo módulo. Além disso, na perspetiva da empresa, a criação deste módulo aumenta o valor das outras aplicações disponibilizadas para a área de RH e ainda aumenta o tamanho do mercado na qual a empresa poderá atuar.

Numa perspetiva do consumidor (valor para o cliente), segundo Woodall (2003), o valor de um produto/serviço pode ser abordado de várias perspetivas. Resumindo cada uma dessas perspetivas, consideram-se:

1. Balanço benefícios/sacrifícios;
2. Resultados do uso/experiência;
3. Atributos percecionados do produto;
4. Preço do produto/serviço;
5. Diferença do preço objetivo.

Considerando estes critérios e analisando o produto a ser desenvolvido, o valor para o cliente dependerá maioritariamente de dois factores. O primeiro será a existência de produtos da área de RH fornecidos pela CPCIT4ALL nas empresas. Apesar de não ser um fator essencial, o produto será mais valorizado por quem já usou e têm experiência na utilização dos produtos da empresa. O segundo fator será a dimensão do cliente. Empresas maiores terão uma maior necessidade em adoptar este tipo de sistemas, logo os benefícios terão um peso maior na escolha.

No entanto, na sua generalidade, o produto disponibilizado oferece às empresas não só a possibilidade de organizarem melhor o processo de recrutamento e seleção, como também a de o alterarem com mais fluidez dependendo das análises que façam com os dados recolhidos pela mesma. Além disso, a automatização de determinadas tarefas fará com que as empresas que utilizem a aplicação possam reduzir o número de pessoas necessárias para o funcionamento deste processo.

4.2.1 Proposta de Valor

Para Osterwalder e Pigneur (2003), uma proposta de valor é a visão geral dos produtos/serviços de uma empresa, que são percebidos como valiosos para o cliente.

A aplicação de modelos formais mostra-se vantajosa aquando da modelação das propostas de valor. Isto porque, permite uma compreensão mais profunda, da qual resulta uma melhor comunicação. Além disso, facilita a sua implementação e a comparação com a competição, o que, eventualmente, promove a inovação (Osterwalder & Pigneur, 2003).

Deste modo, a proposta de valor deve deixar claro quais as vantagens do produto/serviço disponibilizado, quais os clientes alvo, quais os benefícios da aquisição do mesmo e de que forma este difere das restantes ofertas no mercado. Estas características podem ser deuzidas a partir das questões : “o quê”, “quem”, “como” e “quanto” (Osterwalder & Pigneur, 2003).

Relativamente ao projeto, a proposta de valor consiste na oferta de um sistema, simples e intuitivo, que permita automatizar parcial ou totalmente tanto o processo, como as tarefas, presentes no âmbito do recrutamento e seleção de novos funcionários. O seu proveito máximo é atingido quando se usa o mesmo em conjunto com as restantes aplicações do ecossistema de RH da CPCIT4ALL.

4.3 Business Model Canvas

De acordo com Osterwalder e Pigneur (2010), “um modelo de negócio descreve a lógica de como uma organização cria, entrega e capta valor”⁴.

De modo a facilitar o processo de descrição de um novo modelo de negócio, e com vista a criar uma linguagem comum que permitisse uma melhor compreensão/colaboração de todos os envolvidos neste, foi criado o modelo Canvas.

O modelo (cuja aplicação, no âmbito deste projeto, se encontra na figura 4.3) cobre as quatro áreas principais de um negócio: clientes, oferta, infra-estruturas e viabilidade financeira; estando dividido em 9 blocos com diferentes propósitos:

- **Segmentos de Mercado** (Customer Sements) - encontram-se os diferentes grupos que as organizações visam alcançar e servir;
- **Propostas de Valor** (Value Propositions) - apresentam-se o conjunto de produtos e/ou serviços que criam valor para um segmento de clientes específico;
- **Canais** (Channels) - identificam-se os meios pelos quais as organizações alcançam os clientes para fornecer os serviços e/ou produtos;
- **Relações com Clientes** (Customer Relationships) - identificam-se o tipo de relações que a empresa terá com os segmentos de clientes para os quais atua.

⁴Tradução livre do autor. No original “A business model describes the rationale of how an organization creates, delivers, and captures value”.

Figura 4.3: Modelo Canvas

Parceiros Chave	Atividades Chave	Proposta de Valor	Relações com Clientes	Segmentos de Mercado
Trofa Saúde;	Desenvolvimento e manutenção do software;	Gestão de Processos de Recrutamento;	Assistência pessoal dedicada;	Equipas de Recursos Humanos;
	Recursos Chave	Automatização dos mesmos;	Suporte pós-venda;	
	Internet;	Redução de custos;	Canais	
	Plataformas tecnológicas;	Melhor monitorização de equipas;	Demonstrações em clientes;	
			Parceiros da empresa;	
Estrutura de Custos			Fontes de Receita	
Equipa de desenvolvimento e manutenção de software;			Venda do sistema;	
Marketing;			Serviços de desenvolvimento à medida;	

- **Fontes de Receita** (Revenue Stream) - tem-se o retorno financeiro e/ou de valor de um produto;
- **Recursos Chave** (Key Resources) - descrevem-se os recursos essenciais para oferecer e entregar os elementos previamente descritos.
- **Atividades Chave** (Key Activities) - destacam-se quais as atividades necessárias para garantir o funcionamento do modelo de negócio.
- **Parcerias Chave** (Key Partnerships) - representa a rede de fornecedores e/ou parceiros essenciais para a manter o funcionamento do projeto.
- **Estrutura de Custos** (Cost Structure) - apresentam-se os custos envolvidos na implementação da ideia.

Capítulo 5

Análise e Requisitos

O presente capítulo propõe-se, primeiramente, a construir uma base linguística e conceptual sobre o domínio do problema que potencie a compreensão dos requisitos elicitados com o cliente, apresentados posteriormente.

5.1 Modelo de Domínio

Nesta secção analisam-se os diferentes domínios com o intuito de introduzir a linguagem ubíqua utilizada aquando da definição dos requisitos. Para tal, expõe-se, inicialmente, o conceito de dimensões, seguindo-se os conceitos relativos ao domínio de RH.

5.1.1 Estrutura Organizacional

Num sistema de RS é essencial conhecer a estrutura organizacional seguida pela entidade que usufrui do mesmo, uma vez que, não só os fluxos de trabalho nele presentes dependerão dessa mesma estrutura, como os próprios pedidos de recrutamento serão feitos para funções definidas nessa estrutura.

Além disso, as soluções desenvolvidas pela CPCIT4ALL adoptam uma arquitectura *multi-tenant*, que, apesar do estar associada à redução de custos e à simplificação da manutenção do sistema, quando comparado com aplicações *single-tenant*, exigem um maior esforço no que respeita a configuração das várias instâncias.

Posto isto, identifica-se a necessidade de adoptar de uma abstracção que permita representar qualquer estrutura organizacional no sistema. Para tal, criou-se o conceito de dimensões, representado na Figura 5.1.

A representação das estruturas organizacionais é feita com base em 3 definições:

- **Dimension Setup** - corresponde ao conjunto de *Dimension Items* e à sua respectiva configuração. Estes são utilizados para definir uma estrutura organizacional em abstracto, onde a cada empresa corresponderá um *Dimension Setup* próprio que pode ser identificado através do *tenant* da mesma.

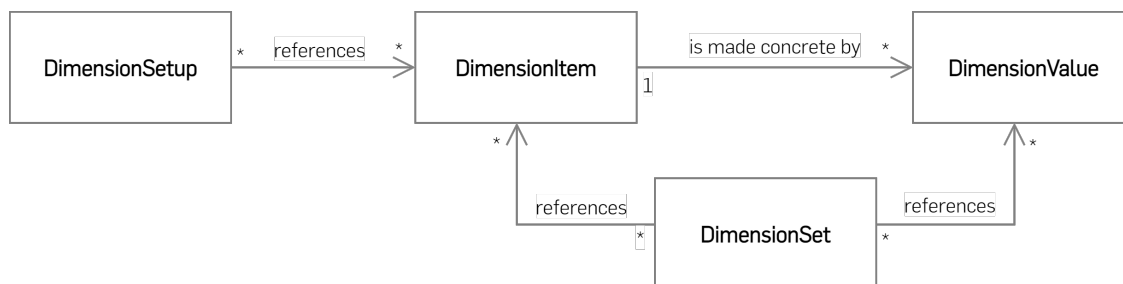


Figura 5.1: Modelo de Domínio das Dimensões

- **Dimension Item** - servirá para definir um conceito utilizado para representar a organização e, quando configurado num *Setup*, as relações que existem entre esses vários conceitos (e.g. dentro de um *Setup*, pode existir o *Item* Função que poderá ser configurado para depender do outro *Item* Departamento).
- **Dimension Value** - representa a concretização de uma das abstrações representadas pelos *Items* (e.g. Departamento de RH seria um *Value* do *Item* Departamento).

Os três conceitos enunciados são suficientes para poder representar e identificar qualquer posição na estrutura organizacional. Contudo, a consulta de toda a informação de uma posição específica exige que se consulte múltiplas vezes a base de dados.

O conceito de *Dimension Set* é criado para colmatar esta restrição. Um *Set* corresponderá à lista dos pares *Item-Value* que identificam todos os níveis da posição concreta. Por exemplo, estando uma organização dividida em departamentos, cada um com um conjunto de funções específicas, os *Sets* das funções irão conter dois pares *Item-Value* - *Departamento-DepartamentoExemplo* e *Função-FunçãoExemplo*.

Deste modo, quando se cria um *Value* processa-se o *Set* para o mesmo e elimina-se a necessidade de, no futuro, ter de voltar a processar essa informação.

Em suma, para cada *Setup* existirá uma configuração de *Items* que, por sua vez, terão um conjunto de *Values* que representam a informação concreta da organização. Além disso, existem os *Sets* para obter, de uma forma rápida e simples, a informação detalhada de uma posição específica dentro da estrutura.

5.1.2 Domínio de RH

Para compreender as necessidades do sistema, é importante conhecer, ainda que por alto, alguns conceitos e relações existente no domínio de RH.

Partindo da análise da Figura 5.2, um colaborador da empresa é responsável por determinadas funções e pode realizar pedidos de recrutamento para essas mesmas. As funções, neste contexto, serão os *Sets* apresentados no sub-capítulo anterior.

O pedido realizado origina uma ou mais vagas que serão ponderadas, individualmente, a cada passo do fluxo de aprovação do mesmo. No fim desse fluxo, não só o pedido se transforma num processo, se devidamente autorizado, como todas as vagas do pedido que tenham sido aprovadas são transformadas em vagas do processo de recrutamento.

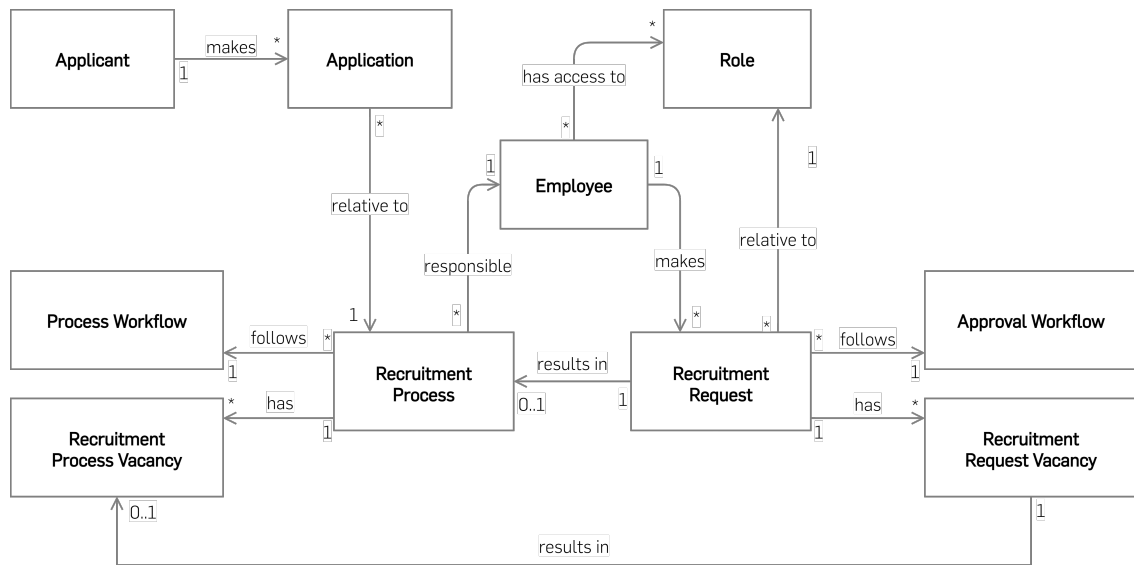


Figura 5.2: Modelo de Domínio Parcial

Para se inscreverem num dos processos, os candidatos realizam a sua candidatura, sendo esta uma cópia dos dados do candidato no momento da inscrição no processo.

Os processos estão associados a um fluxo de trabalho que será seguido por todas as suas candidaturas. Estes são dados como terminados quando todas as suas vagas ficam preenchidas, ou se, por qualquer motivo, o processo for encerrado manualmente.

O fluxo de aprovação dos pedidos e o fluxo de trabalho dos processos apesar de serem conceptualmente idênticos, têm algumas características distintas entre eles, sendo necessário abordar o domínio dos mesmos com um pouco mais de detalhe.

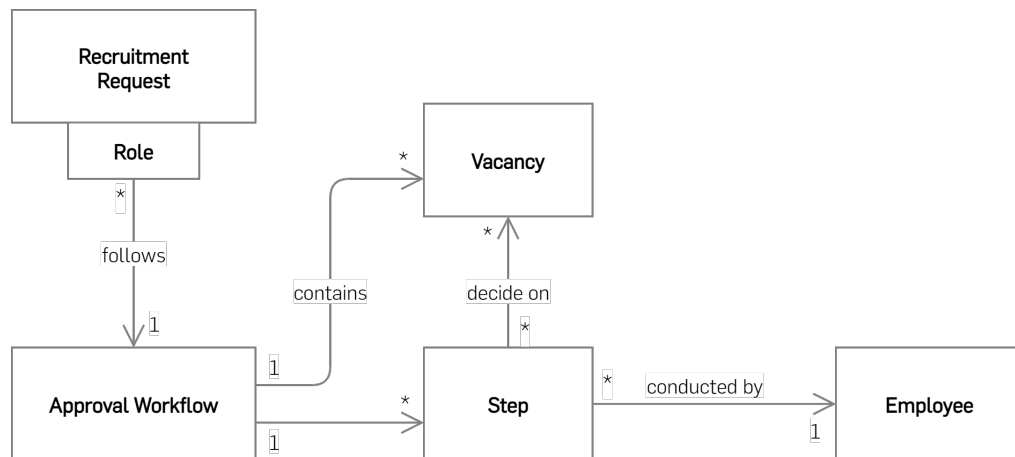


Figura 5.3: Modelo de Domínio do Fluxo de Aprovação

Os pedidos de recrutamento seguem um fluxo de aprovação específico de acordo com a função para o qual foram criados. Como se vê na Figura 5.3, os fluxos de aprovação são divididos em etapas, em que determinados empregados darão o seu parecer sobre as vagas e outras características do pedido (e.g. as condições contratuais delimitadas, os requisitos mínimos para o cargo, etc.).

Apesar de não estar representado, é possível dentro de uma etapa fazer um pedido de esclarecimento. Isto consiste em pedir a um interveniente que tenha actuado previamente no mesmo fluxo que justifique determinada acção e/ou comentário realizado.

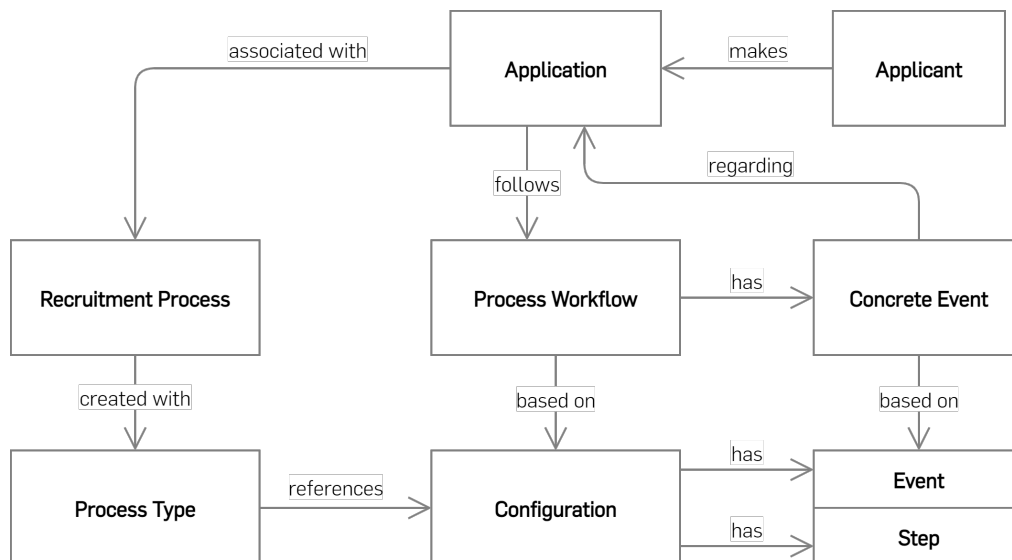


Figura 5.4: Modelo de Domínio do Fluxo de Trabalho

No que respeita os fluxos de trabalho seguidos pelas candidaturas de um processo de recrutamento, como se pode ver pela Figura 5.4, estes seguem uma configuração feita pelo tipo de processo associado.

A estratégia da selecção do tipo de processo vai variar entre as organizações. Há empresas que terão um tipo apenas, ao passo que, para outras empresas, o tipo de processo estará, também, dependente da função para o qual é feito.

Independentemente disso, um tipo de processo estará associado a uma configuração que representa o fluxo que cada candidatura seguirá por pré-definição. Cada configuração agrupa etapas aos seus respectivos eventos, mantendo a informação de qual ordem a seguir.

No entanto, a associação do fluxo de trabalho com a configuração nunca é feita directamente, pois deve haver a possibilidade de um colaborador alterar o fluxo para uma dada candidatura, se assim pretender, sem ter em conta as configurações existentes.

Deste modo, quando se dá por realizado um evento concreto pode-se: indicar o evento seguinte e este será utilizado para criar o novo evento concreto; ou, consultando a configuração do tipo de processo, o sistema saberá determinar qual o evento que se deve ter por base.

5.2 Engenharia de Requisitos

No presente sub-capítulo, são apresentados os requisitos funcionais, adoptando um modelo de caso de usos, e não funcionais, recorrendo ao modelo FURPS+ para a categorização de requisitos de qualidade e/ou que não representem casos de use *per si*.

O nome deste último modelo é um acrónimo que se separa em: Funcionalidade, Usabilidade, Confiabilidade (*Reliability*), Desempenho (*Performance*) e Sustentabilidade (*Supportability*). O símbolo “+” representa outros atributos a considerar, tais como restrições de implementação, restrições de interface, entre outras (Carlos Eduardo Vazquez, 2016; Larman, 2005).

5.2.1 Funcionalidade

Segundo Carlos Eduardo Vazquez (2016), os requisitos funcionais “descrevem o comportamento que o software deve ter em termos das tarefas ou serviços” oferecidos aos utilizadores.

A sua especificação, no presente sub-capítulo, passa pela identificação dos processos elementares de negócio. Segue-se a apresentação dos diversos actores do sistema e, por fim, da sua relação com os processos, o que permite identificar todos os casos de uso presentes na solução.

Além disso, em seguida, também são identificadas todas as funcionalidades transversais aos casos de uso, como é exemplo a autenticação, a comunicação via email, entre outros.

As seguintes listas representam os processos elementares do negócio que, mais à frente, serão associados aos actores. Estão agrupados pelo contexto dos mesmos, sendo que qualquer referência feita ao título de uma das listas corresponde à enumeração de todos os itens da mesma.

Pedidos de Recrutamento (PDR)

- PDR01 - Criar pedido de recrutamento.
- PDR02 - Listar e filtrar pedidos de recrutamento.
- PDR03 - Enviar pedidos de recrutamento para o fluxo de aprovação.
- PDR04 - Cancelar um pedido de recrutamento.
- PDR05 - Anexar documentos a um pedido de recrutamento.

Aprovação de Pedidos (AP)

- AP01 - Listar e filtrar pedidos de aprovação.
- AP02 - Actuar sobre um pedido de aprovação.
- AP03 - Reencaminhar pedido para outro utilizador.
- AP04 - Fazer um pedido de esclarecimento sobre um pedido de aprovação.
- AP05 - Responder a um pedido de esclarecimento.
- AP06 - Criar perfis de aprovação.
- AP07 - Associar perfis de aprovação à determinados fluxos de trabalho.
- AP08 - Configurar o fluxo de trabalho de aprovação.

Processos de Recrutamento (PCR)

- PCR01 - Criar um processo de recrutamento.
- PCR02 - Listar e filtrar os processos aos quais tem acesso.
- PCR03 - Colocar um processo em *StandBy*.
- PCR04 - Encerrar um processo.
- PCR05 - Pedir alteração do responsável de um processo.
- PCR06 - Alterar o técnico associado a um processo de recrutamento.
- PCR07 - Actuar sobre um pedido de alteração de técnico.

Gestão de Candidatos (GC)

- GC01 – Registrar um candidato.
- GC02 – Associar um candidato a um processo (criar uma candidatura).
- GC03 – Listar e filtrar a lista de candidatos.
- GC04 – Convidar candidato.
- GC05 – Avaliar um candidato.

Gestão de Eventos (GE)

- GE01 - Configurar os passos e eventos de um determinado processo.
- GE02 - Configurar os dados pedidos num determinado evento.
- GE03 - Realizar um evento.

Partilha de Anúncios (PA)

- PA01 - Criar e editar o *template* de um anúncio.
- PA02 - Publicar um anúncio.
- PA03 - Republicar um anúncio.
- PA04 - Remover um anúncio publicado.

Candidatos (C)

- C01 - Registrar perfil.
- C02 - Candidatar-se a um anúncio.
- C03 - Fazer uma candidatura espontânea.
- C04 - Listar anúncios.
- C05 - Eliminar registo.

É de notar que existe um grupo de funcionalidades, correspondente à parametrização do sistema, que, dado ser de uma extensão considerável, apenas se descreve no Anexo B. Além disso, ainda no anexo indicado, também estão descritos, em mais detalhe, todos os processos identificados.

No que respeita a identificação dos actores do sistema pode-se fazer a distinção entre os utilizadores externos e os utilizadores internos, sendo que se entende por utilizador interno todos os colaboradores da(s) empresa(s) que utilizam a solução.

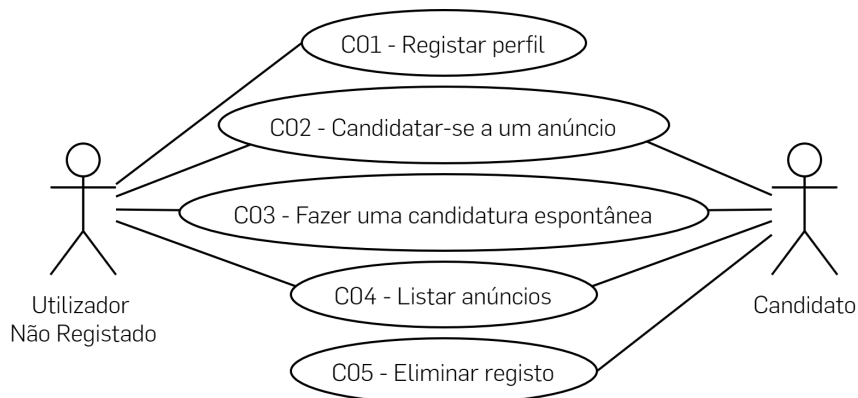


Figura 5.5: Diagrama de Casos de Uso - Utilizadores Externos

Como se verifica na Figura 5.5, identificam-se dois utilizadores externos: *Utilizador Não registado* e *Candidato*. Ambos podem aceder aos anúncios disponibilizados e candidatar-se aos que julgarem apropriados, ou realizar uma candidatura espontânea. No entanto, enquanto que para o *Utilizador Não Registado* é possível registar o seu perfil e tornar-se um *Candidato*, para este último, tendo o perfil já registado, existe a possibilidade de eliminar o seu perfil e os respectivos dados (em conformidade com o RGPD).

No que concerne os utilizadores internos, haverá a distinção entre 6 níveis, sendo *Requerente* e o *Validador* os dois primeiros. Estes representam os colaboradores que não pertencem ao departamento de RH da empresa.

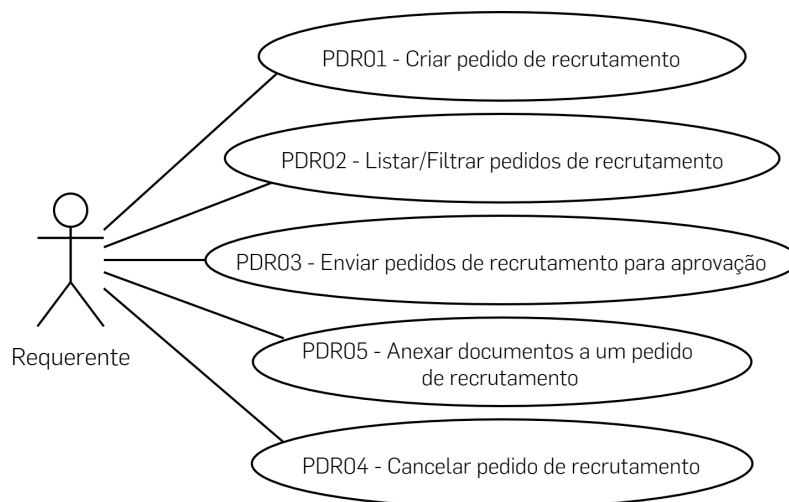


Figura 5.6: Diagrama de Casos de Uso - Requerente

Segundo a Figura 5.6, o *Requerente* terá como única responsabilidade fazer pedidos de recrutamento ao sistema, enquanto que, segundo a Figura 5.7, o *Validador*, além dos pedidos de recrutamento, irá ser interveniente nos fluxos de aprovação dos pedidos feitos pelos *Requerentes*.

Assim, tal como podemos ver na Figura 5.7, além de herdarem as funcionalidades dos pedidos de recrutamento, têm acesso a algumas das funcionalidades associadas à aprovação de pedidos.

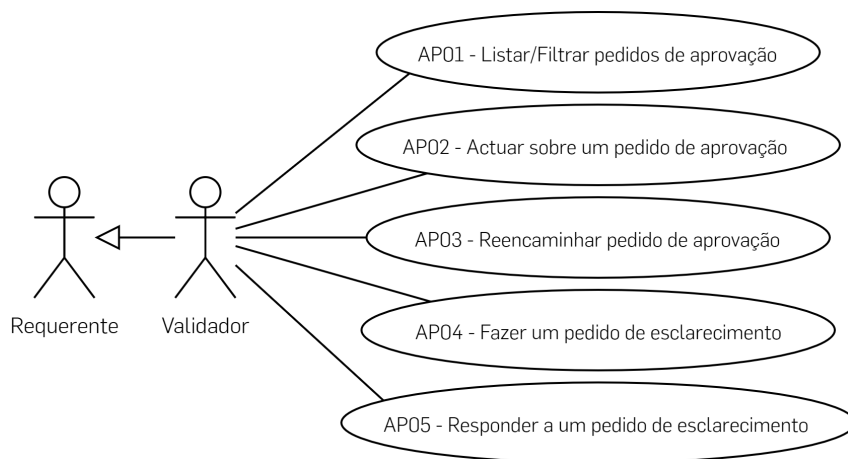


Figura 5.7: Diagrama de Casos de Uso - Validador

Os próximos actores do sistema têm acesso às mesmas funcionalidades que um *Validador* e mais algumas, dependendo do tipo de técnico que é. Para evitar ter diagramas ilegíveis, decidiu-se representar o actor *Técnico* que servirá como uma abstracção da base que é herdada por todos os tipos de técnicos.

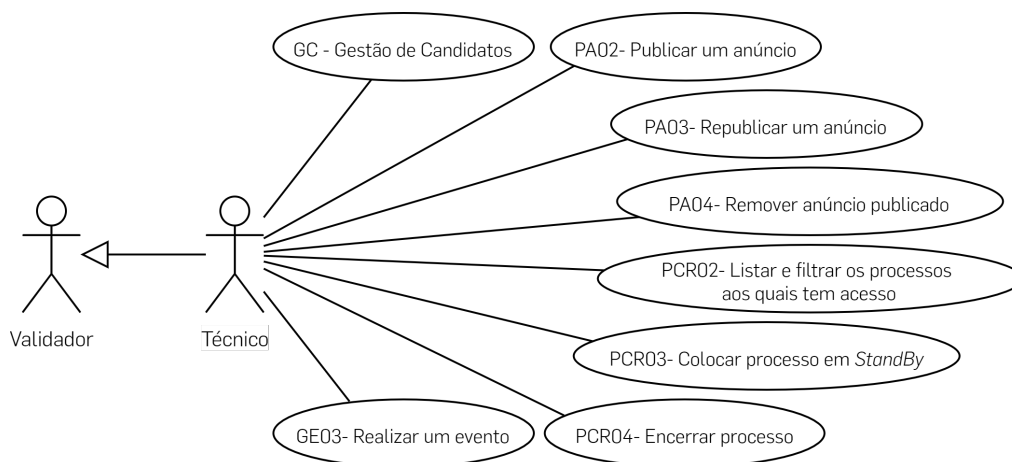


Figura 5.8: Diagrama de Casos de Uso - Técnicos Abstractos

Como se verifica pela Figura 5.8, os *Técnicos* têm acesso à gestão de candidatos e também lhes compete a publicação, edição ou remoção dos anúncios gerados para os vários processos. Uma vez que vão estar associados a processos específicos, podem ainda listar os mesmos e actuar sobre eles, colocando em *StandBy* ou encerrando-os.

Por fim, conseguem realizar eventos dentro dos processos aos quais estão afectos (e.g. realizar uma entrevista ou um teste).

Passando da abstracção para os actores em concreto que irão herdar todas as funcionalidades do *Técnico*, temos o *Técnico de Unidades*, o *Técnico de Sede* e o *Técnico de RH*.

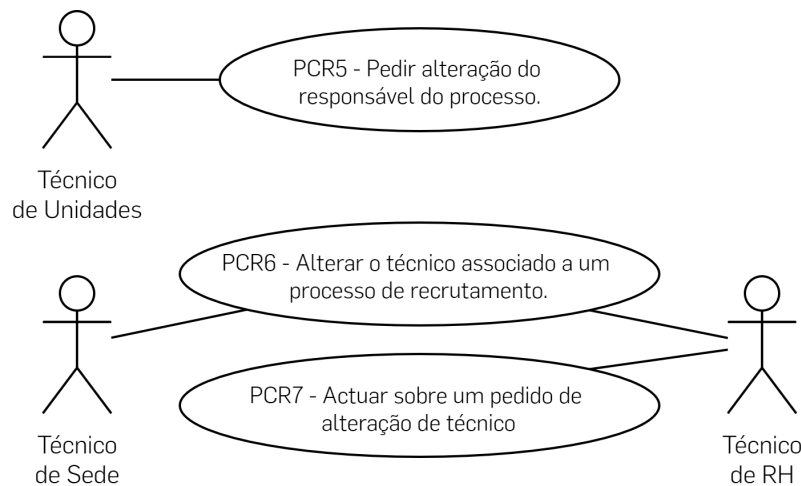


Figura 5.9: Diagrama de Casos de Uso - Técnicos Concretos

Como se verifica na Figura 5.9, a principal diferença entre os técnicos é referente à gestão dos responsáveis dos processos. Enquanto que o *Técnico de Unidades* pode propor a mudança de técnico para um processo - proposta essa cuja aprovação deve ser feita por um *Técnico de RH*; o *Técnico de Sede* e o *Técnico de RH* poderão alterar directamente o técnico responsável de um pedido.

Por fim, como se verifica na Figura 5.10, os processos elementares ainda não referidos estarão associados ao *Master*.

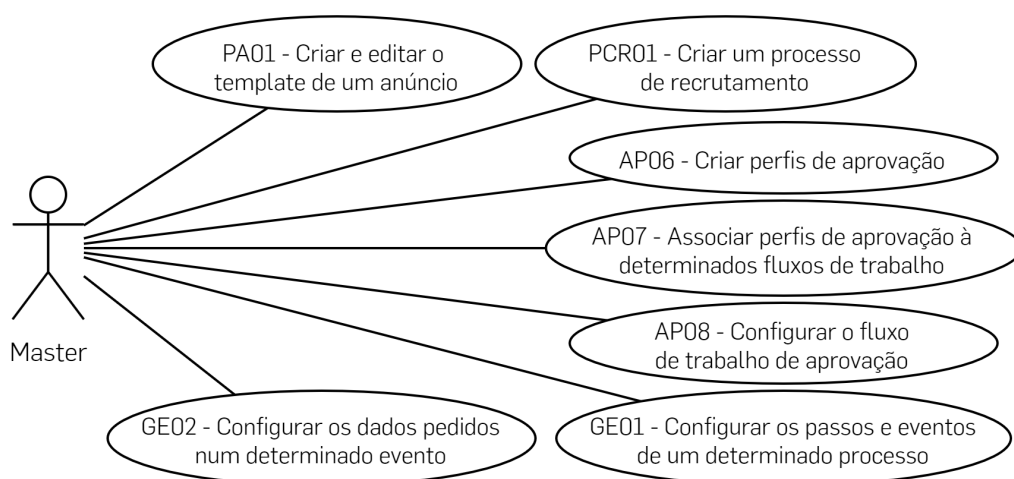


Figura 5.10: Diagrama de Casos de Uso - Master

O *Master* pode editar os *templates* dos anúncios, criar processos de recrutamento sem necessitar de fazer um pedido, criar e configurar totalmente os fluxos de aprovação e

os perfis a eles associados. Pode, ainda, configurar os fluxos de trabalho relativos aos processos e ajustar os detalhes apresentados e exigidos quando da realização de eventos.

Além disso, o *Master* tem acesso a todos os processos que os técnicos podem executar. Ainda, existe uma lista de parametrizações, presentes no Anexo B, que, pela sua extensão, não se encontra representada no diagrama, mas cujo único actor que tem acesso às mesmas é o *Master*.

Funcionalidades Transversais

Além das funcionalidades apresentadas, existe, também, um conjunto de funcionalidades que irá ser transversal ao sistema. Dentro dessas, foram identificadas funcionalidades relativas a: autenticação, *logs* e comunicação via email.

No que respeita a autenticação, deve haver mecanismos que assegurem a mesma, sendo que a autenticação será obrigatória para os processos realizados pelos colaboradores da empresa. A autenticação relativa aos candidatos deverá ser feita em separado, e não é obrigatória para todos os processos associados aos mesmos.

Além disso, o sistema deve estar preparado com um sistema de *logs* e deve conseguir assegurar a comunicação via email.

5.2.2 Usabilidade

A usabilidade refere-se à facilidade de uso do software e inclui factores humanos, estética, consistência na interface do utilizador, entre outros. No projecto actual existem os seguintes requisitos:

- Deve haver três interfaces do utilizador distintas:
 - Portal dos candidatos para os *Utilizadores Não Registados* e os *Candidatos*;
 - Portal do colaboradores de loja para os *Requerentes* e os *Validadores*;
 - Portal de RH para os *Técnicos*.
- As interfaces devem ser simples, intuitivas e eficientes;

5.2.3 Confiabilidade

A confiabilidade refere-se à integridade, conformidade e interoperabilidade do software. Inclui aspectos de frequência e gravidade de falha, possibilidade de recuperação de falhas, previsibilidade, exactidão e tempo médio entre falhas. No âmbito deste projecto tem-se os seguintes:

- O produto final deve ser tolerante a falhas;
- O lado do servidor deve garantir alta disponibilidade, seguindo um Service Level Agreement (SLA) de *two nines* (99%)¹;

¹O sistema pode, no máximo, estar em baixo durante: 14 minutos por dia; ou 3 dias de 15 horas num ano.

- O sistema deve estar acessível vinte e quatro horas por dia, sete dias por semana;
- As falhas de um componente do lado do servidor não devem ser propagadas para outros componentes;

5.2.4 Desempenho

O desempenho refere-se à velocidade, eficiência, taxa de transferência, tempo de resposta e uso de recursos. No contexto deste projecto tem-se:

- O tempo médio de resposta dos *endpoints* deve ser inferior a 1 segundo;
- Deve poder processar até 10 pedidos por segundo nos *endpoints* dos serviços;
- O tempo médio do carregamento das páginas da interface do utilizador deve ser inferior a 5 segundos;
- Deve suportar múltiplos acessos concorrentes sem comprometer a eficiência dos utilizadores.

5.2.5 Sustentabilidade

A sustentabilidade refere-se à extensibilidade, adaptabilidade, manutenção, compatibilidade, escalabilidade e testabilidade da solução. Posto isto, o sistema deve:

- Ter a capacidade de expansão e permitir a adição de novos módulos, sempre que necessário;
- Ser desenvolvido de forma genérica para que alterações futuras possam ser feitas sem impactar as funcionalidades já desenvolvidas. Adoptando, sempre, bons princípios de manutenção, adaptabilidade e configuração, que permitam a sua evolução.

5.2.6 Outras (+)

Dentro das restantes restrições possíveis, foram apenas identificadas restrições de design e de implementação.

Restrições de Design

As restrições de design são aquelas que restringem algo na arquitectura do sistema a ser desenvolvido. Assim, tem-se:

- A solução deverá ser desenvolvida seguindo uma arquitectura *multi-tenant*;
- Os dados devem ser persistidos numa base de dados relacional.

Restrições de Implementação

As restrições de implementação são aquelas que delimitam a construção do sistema. Assim, tem-se:

- O sistema deve ser desenvolvido utilizando a *framework* usada na CPCIT4ALL.

5.3 Notas Finais

Apesar de terem sido identificados, nem todos os requisitos serão contemplados na fase de implementação, seja por questões temporais, seja pela distribuição de trabalho feita por parte dos gestores do projecto.

Cada um destes requisitos irá resultar num ou mais *tickets*, dependendo da complexidade, e serão atribuídos a um dos membros da equipa.

Capítulo 6

Desenho

O presente capítulo consiste na exposição do desenho proposto como solução aos requisitos apresentados, partindo de um nível mais abstracto - o arquitetural - para um nível mais detalhado, deixando sempre margem para decisões aquando da implementação.

Para tal, a abordagem adoptada consiste na utilização dos modelos previamente apresentados (Secção 2.3.1): Modelo C4 de Brown e Modelo de Vistas 4+1 de Krutchen.

Neste contexto, no decorrer do presente capítulo são apresentados, nos 4 níveis diferentes, todas as vistas relevantes para a compreensão da solução desenvolvida.

6.1 Nível 1 - Granularidade do Sistema

A presente secção foca-se no desenho mais generalizado do sistema.

6.1.1 Vista Lógica

Na Figura 6.1 apresenta-se a vista lógica de alto nível da solução.

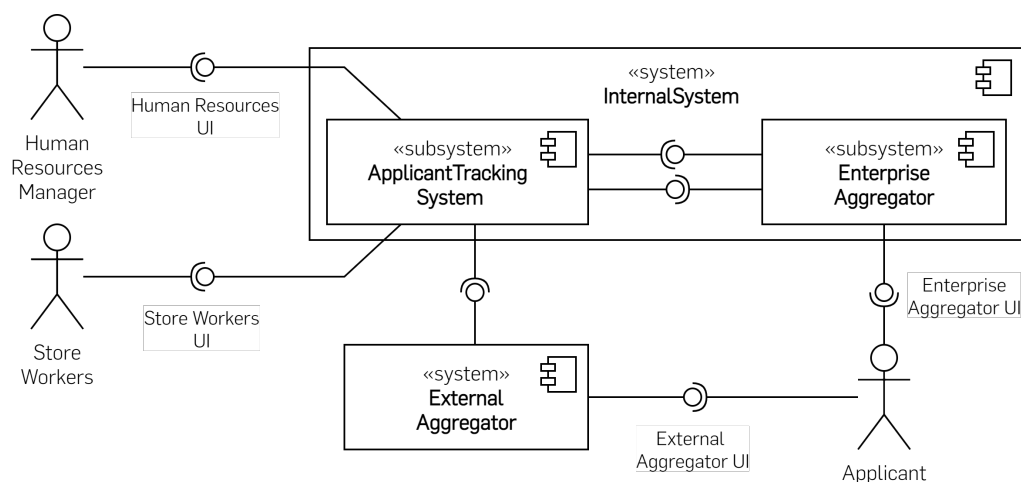


Figura 6.1: Vista Lógica de Nível 1

Por uma questão de legibilidade, no presente diagrama não se utilizam os actores apresentados nos requisitos.

No entanto, é possível fazer a associação entre eles: o *Human Resource Manager* representa todos os *técnicos* e *masters*; o *Store Workers* representa os *requerentes* e os *validadores*; por fim, o *Applicant* compreende os *candidatos* e os *utilizadores não registados*.

Ademais, na Figura 6.1 podem-se diferenciar dois sistemas:

- **Internal System** - sistema responsável pelo acervo da informação, coordenação dos vários pedidos/processos/candidaturas existentes, pela partilha de anúncios e a recolha de candidaturas;
- **External Aggregator** - sistema cujo propósito consiste na gestão de anúncios em sistemas externos (e.g. gerir os anúncios do LinkedIn, Indeed, entre outros).

Considerando a excessiva responsabilidade do **Internal System**, subdividiu-se o mesmo em dois subsistemas. Ficando este em conformidade com o conhecimento apresentado no enquadramento do projecto (Figura 2.2).

O primeiro, **Applicant Tracking System**, é responsável pelo acervo da informação relativa aos pedidos, processos, toda a informação relevante para a coordenação dos mesmos e pela informação de todas as candidaturas realizadas.

O segundo, **Enterprise Aggregator**, tem como responsabilidade a gestão dos anúncios no *site* institucional e a gestão das candidaturas realizadas nesse mesmo *site*.

Além disso, apesar de estar representado apenas um **External Aggregator**, a solução final poderá contar com mais sistemas idênticos.

Por fim, identificam-se três interfaces que devem ser desenvolvidas internamente:

- **HumanResourcesUI** - Portal utilizado pelos *técnicos* e *masters*;
- **StoreWorkersUI** - Portal utilizado pelos *requerentes* e *validadores*;
- **EnterpriseAggregatorUI** - Portal utilizado pelos *utilizadores não registados* ou por *candidatos* já inscritos.

No entanto, o presente trabalho só irá abordar as interfaces relativas ao *Applicant Tracking System*, ainda que no desenho apresentado se tenha considerado a interface do *Enterprise Aggregator*.

6.1.2 Vista de Processos

No que respeita vista de processo de alto nível, não só é importante saber quais os sistemas que pertencem à solução geral, como também conhecer como estes comunicam.

Deste modo, na secção actual apresenta-se o fluxo de um dos poucos processos de negócio que requer ambos os sistemas apresentados; isto porque, todos os outros fluxos serão a simples comunicação entre o actor e o sistema a ser consultado.

Assim, na Figura 6.2 ilustra-se a sequência de passos adoptada para a publicação de um ou mais anúncios.

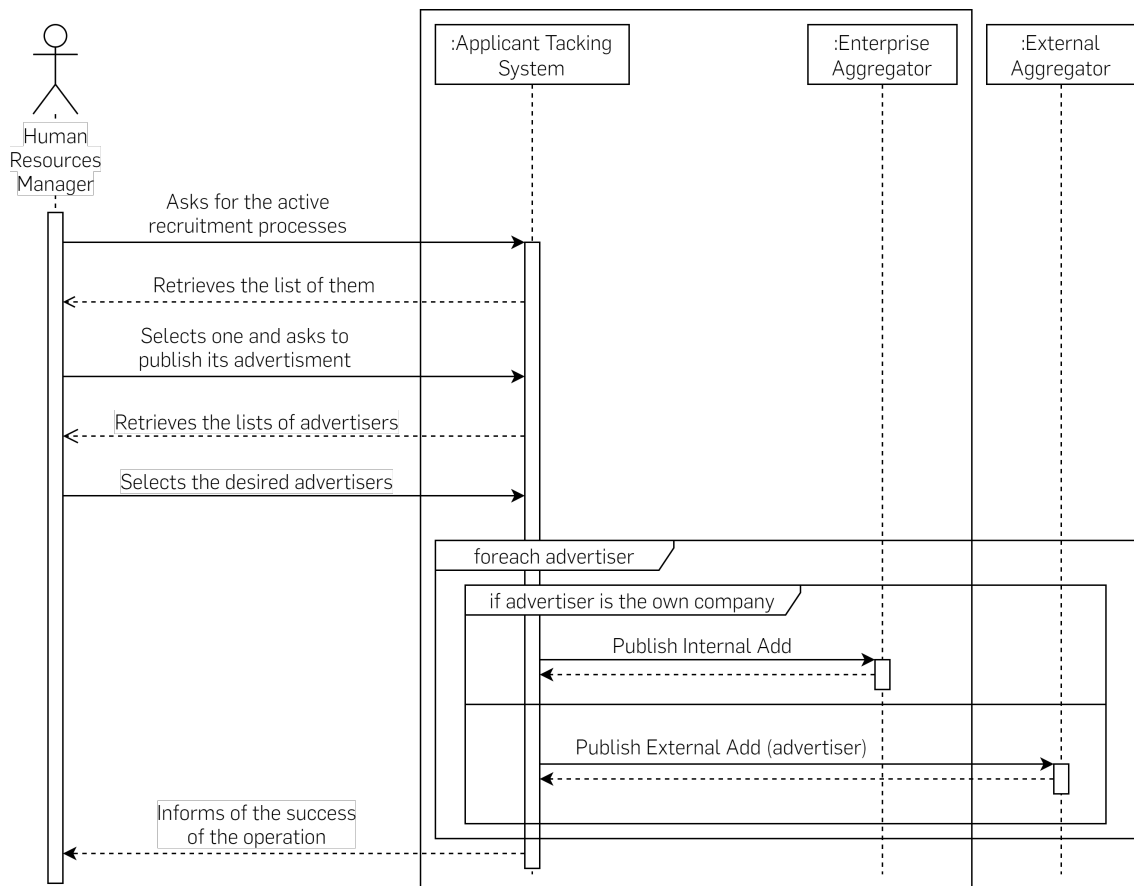


Figura 6.2: Vista de Processos de Nível 1 - Publicar Anúncio

Como se verifica no diagrama, o *Applicant Tracking System* recorrerá aos agregadores para que a publicação dos anúncios seja devidamente realizada.

6.2 Nível 2 - Contentores ou Aplicações

Esta secção foca-se no desenho dos subsistemas do *Internal System*, com o intuito de identificar os vários *containers* e a relação entre eles.

6.2.1 Vista Lógica

Nesta secção, estão representados simultaneamente os dois subsistemas do **Internal System**, o *ApplicantTrackingSystem* e *Enterprise Aggregator*, para enfatizar a forma como se preconiza a interoperabilidade entre os mesmos.

Além disso, também o *External Aggregator* lá aparece representado, uma vez que deve, igualmente, ser considerado na questão da interoperabilidade do sistema, uma vez que a publicação de anúncios também é da sua responsabilidade.

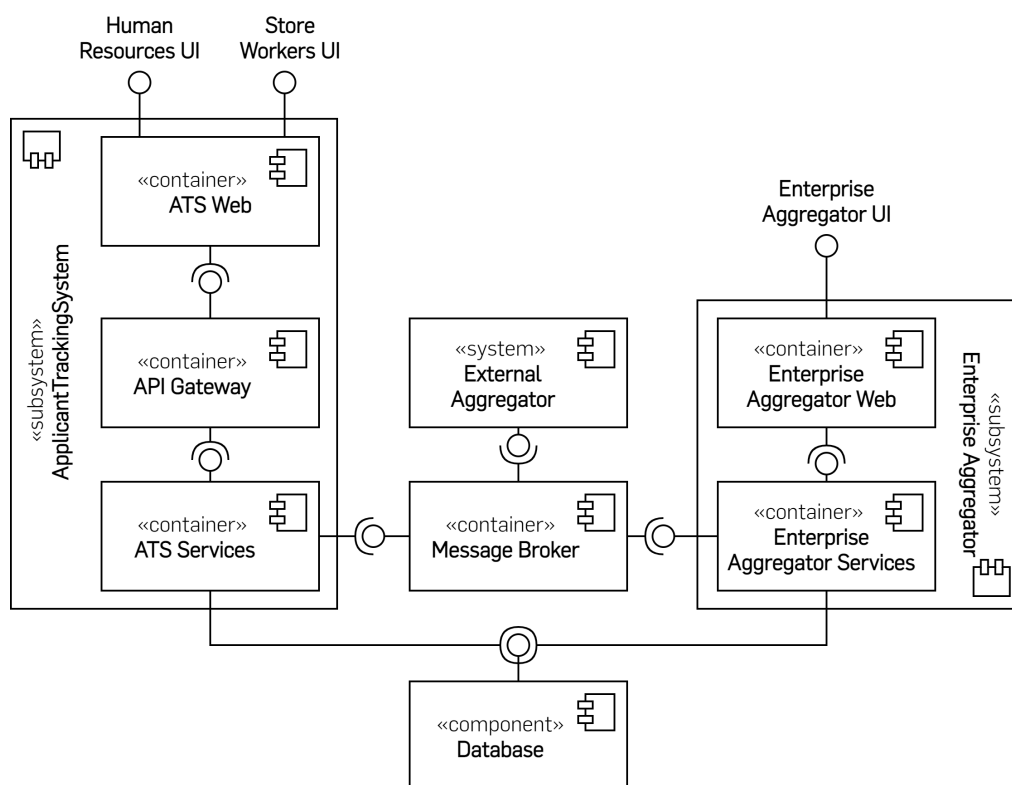


Figura 6.3: Vista Lógica de Nível 2

Como se verifica na Figura 6.3, a *Database* vai ser partilhada entre todos os sistemas e o *Applicant Tracking System* é separado em três *containers*, apresentados na Figura 6.3.

1. **ATS Web** - responsável por fornecer as duas interfaces gráficas identificadas para uso dos utilizadores internos - **Human Resources UI** e **Store Workers UI**;
2. **ATS Services** - responsável pela execução dos processos necessários à realização dos serviços requisitados por parte do **ATS Web**;
3. **API Gateway** - responsável pela coordenação dos pedidos feitos aos serviços e por definir a interface que será utilizada para comunicar com o sistema de gestão de pedidos, processos e anúncios.

Uma vez que a publicação de anúncios vai ser feita pelo **ATS Services**, é necessário haver algum tipo de comunicação com os agregadores. Tendo em conta que todos os sistemas onde se irão publicar os anúncios, à excepção do empresarial, são sistemas externos, optou-se por utilizar um **Message Broker**.

Deste modo, apesar do envio do anúncio ser despoletado pelo **Applicant Tracking System**, é da responsabilidade do **Message Broker** assegurar que esse mesmo anúncio chegue devidamente ao(s) receptor(es) necessário(s).

Uma vez que, nem o **Applicant Tracking System**, nem o **Enterprise Aggregator** necessitam de saber como as mensagens são encaminhadas entre si, optou-se por colocar o **Message Broker** fora destes componentes. Assim, garante-se uma melhor separação de responsabilidades e um maior desacoplamento.

Além disso, garante-se que, em caso de falhas nos sistema externos (e.g. uma situação de *downtime*), a publicação eventual dos anúncios é assegurada e, ainda, que os agregadores não necessitam de comunicar directamente com os serviços para os informar da devida publicação.

Relativamente ao **Enterprise Aggregator**, está dividido em dois *containers*:

1. **Enterprise Aggregator Web** - responsável por fornecer a interface gráfica aos utilizadores externos à organização;
2. **Enterprise Aggregator Services** - responsável pela execução dos serviços relativos à gestão de candidaturas e à publicação de anúncios no *site* institucional.

Ao contrário do **Applicant Tracking System**, não se adicionou um **API Gateway** uma vez que esta componente *web* só disponibiliza uma interface e a quantidade de serviços disponibilizada pelo agregador é significativamente menor que a do primeiro componente.

Por fim, apesar do **External Aggregator** não ser aprofundado nos níveis seguintes, é necessário não ignorar o seu propósito. Este será o componente responsável pela comunicação com as plataformas externas de publicação de anúncios.

Deste modo, prevê-se a necessidade de receber uma notificação aquando da devida publicação de um anúncio, para poder alterar o seu estado no sistema. A realização de candidaturas nessas mesmas plataformas estão sujeitas a esta mesma questão, sendo necessário receber a informação da candidatura para a poder registar no sistema interno.

Estes constrangimentos podem ser resolvidos de três maneiras:

1. Dando, ao **External Aggregator**, acesso à base de dados partilhada;
2. Expondo serviços na **API Gateway** para receber directamente as notificações;
3. Recorrendo ao **Message Broker**.

A duas primeiras opções são, possivelmente, as que garantem uma mais rápida actualização da informação aquando do uso do sistema. No entanto, ambas diminuem a coesão e aumentam o acoplamento do sistema como um todo. Enquanto que na primeira se aumenta o número de componentes que acedem à mesma base de dados, no segundo passa-se parte da responsabilidade do **External Aggregator** para o **Applicant Tracking System**.

A última opção, apesar de estar dependente do **Message Broker**, mantém a coesão do sistema. Isto porque, se a comunicação com as plataformas externas for realizada apenas por um dos componentes, torna-se mais fácil expandir as funcionalidades de partilha de anúncios e recolha de candidaturas, uma vez que apenas esse componente é que teria de ser actualizado.

Como o desenvolvimento do **External Aggregator** não faz parte dos objectivos do presente documento, nenhuma decisão definitiva foi feita para esta situação. Todavia, para a elaboração do desenho actual, considerou-se que o recurso ao **Message Broker** seria a melhor opção. Assim, ficam também expostas as considerações que foram feitas, pois tiveram um impacto no desenho da presente solução.

6.2.2 Vista Processos

Concretiza-se a vista de processos de nível 2 recorrendo a 3 casos de uso. Os dois primeiros referentes ao **Applicant Tracking System** (*criar pedido de recrutamento e publicar anúncio*) e o último referente ao **Enterprise Aggregator** (*realizar candidatura*).

Uma vez que todos os outros casos de uso seguem fluxos semelhantes, utilizando mensagens apropriadas a cada um, evidentemente, fica-se a conhecer a generalidade dos fluxos que a solução irá seguir.

Criar Pedido de Recrutamento

Todos os processos iniciados pelos colaboradores das empresas (*Human Resources Managers* e *Store Workers*) seguem um fluxo semelhante ao apresentado na Figura 6.4. Neste exemplo, um dos colaboradores inicia o processo de criar um pedido de recrutamento.

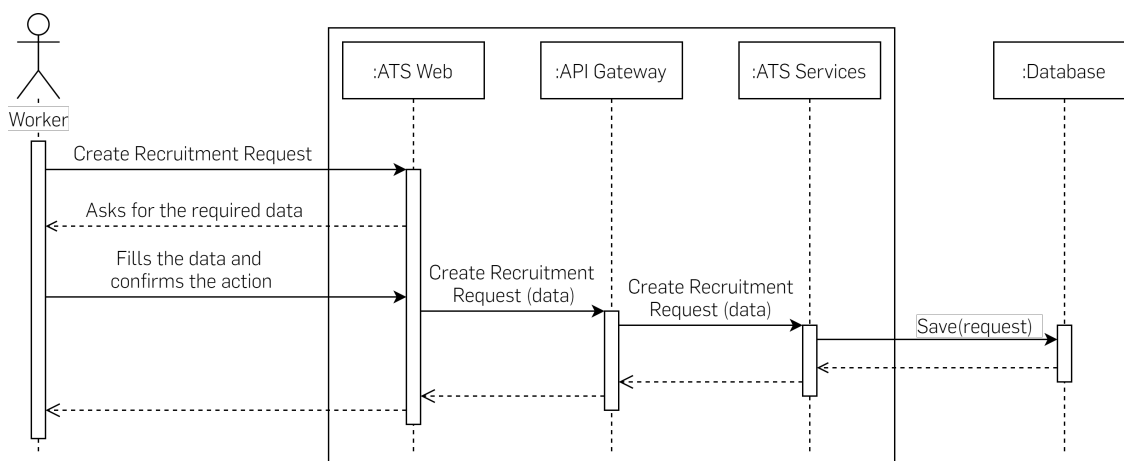


Figura 6.4: Vista de Processos de Nível 2 - Criar Pedido de Recrutamento

Utilizando as interfaces de utilizador disponibilizadas pelo **ATS Web**, os colaboradores vão acedendo às suas possíveis acções. Sempre que alguma acção seja accionada, o **ATS Web** recorre à **API Gateway** para coordenar o(s) serviço(s) necessários a executar no **ATS Services**.

Relativamente aos casos de uso que seguirão um fluxo semelhante, tem-se aqueles relativos aos pedidos de recrutamento (PDR01 a PDR05), à aprovação de pedidos (AP01 a AP08), aos processos de recrutamento (PCR01 a PCR07), à gestão de candidatos (GC01 a GC05), à gestão de eventos (GE01 a G303) e, ainda, tem-se o PA01 relativo à publicação de anúncios.

Publicar Anúncio

Sendo uma acção iniciada por um colaborador da empresa, a sequência de passos seguida é semelhante à apresentada anteriormente, tal como se pode ver na Figura 6.5.

O colaborador de RH começa por seleccionar o processo para o qual tenciona publicar um anúncio, seguindo-se a escolha de quais os anunciantes pretende que publiquem

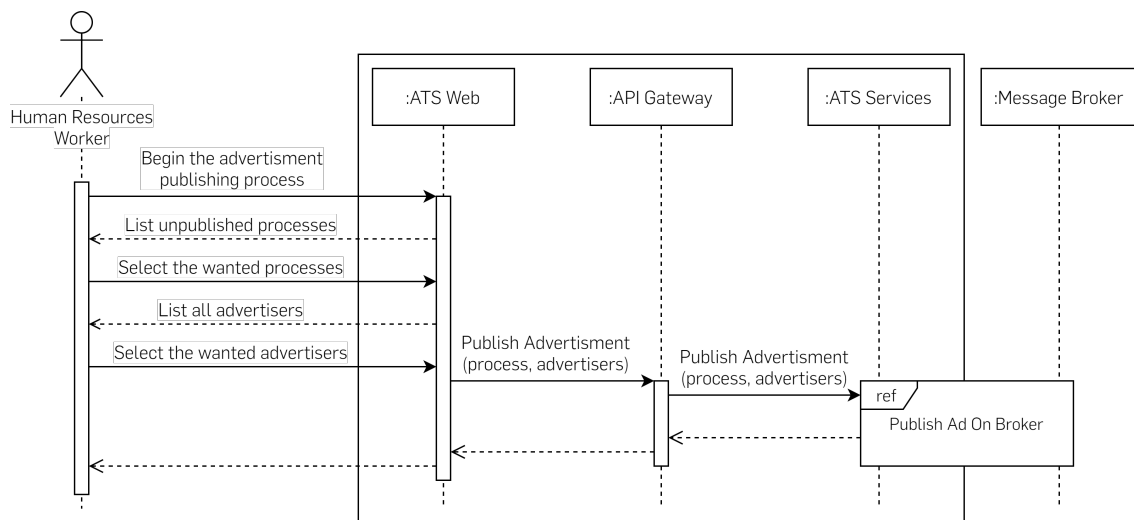


Figura 6.5: Vista de Processos de Nível 2 - Publicar Anúncio

esse mesmo anúncio. Estando todos seleccionados, o **ATS web** recorre à **API Gateway** que coordena o processo com o **ATS Services**.

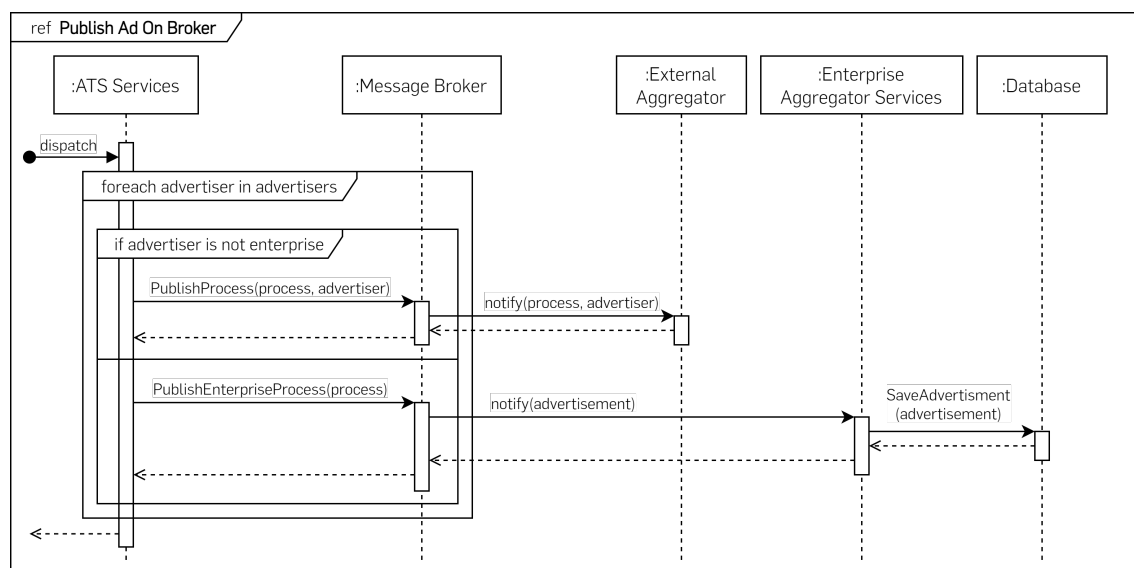


Figura 6.6: Vista de Processos de Nível 2 - Publicar Anúncio Referência

No entanto, como a publicação do anúncio em si não é da responsabilidade dos serviços, estes recorrem ao **Message Broker** para que os dados cheguem devidamente aos componentes responsáveis pela sua publicação.

Na Figura 6.6 expande-se a referência feita na Figura 6.5. Como se pode observar, o processo irá ser publicado na devida *queue* do **Message Broker**, que fará chegar a mensagem aos respectivos subscritores.

A publicação por parte do **Enterprise Aggregator** é feita pelo **Enterprise Aggregator Services** que persiste os dados da operação na base de dados partilhada com o **Applicant Tracking System**. Deste modo, não é necessário notificar o sistema da mudança de

estado do anúncio.

Dois outros casos de uso relativos à publicação de anúncios (PA02 e PA03) seguem, também, uma abordagem semelhante à adoptada por este.

Realizar Candidatura

A Figura 6.7 apresenta o fluxo seguido pelo **Enterprise Aggregator** aquando da realização de uma candidatura.

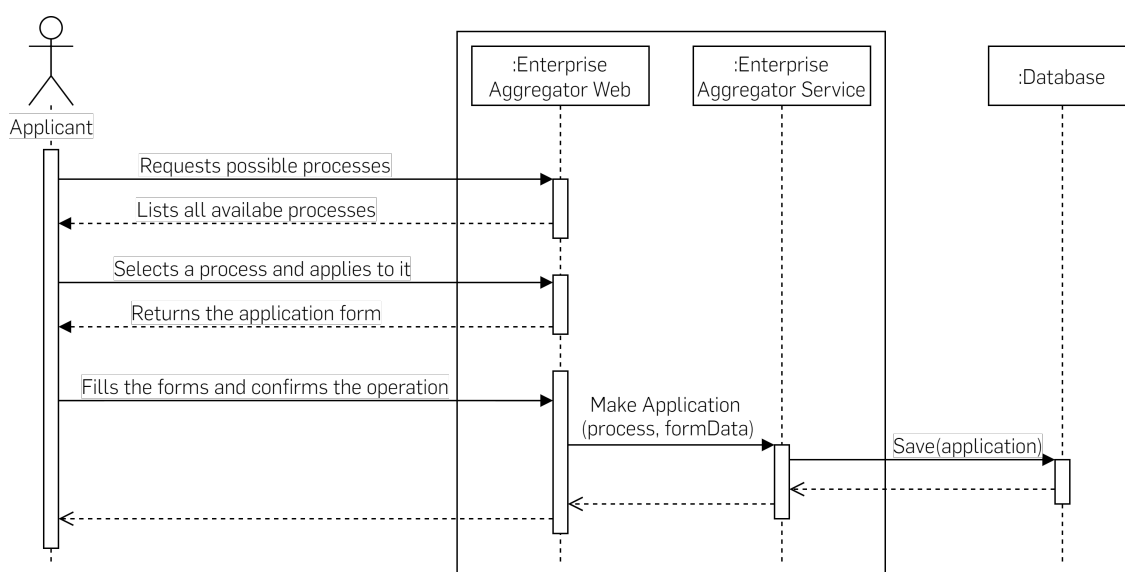


Figura 6.7: Vista de Processos de Nível 2 - Realizar Candidatura

Tal como nos exemplos anteriores, existe um componente *web* que comunica com o utilizador e que recorre aos *services* para executar as operações necessárias. Contudo, este componente *web* não recorre a um *gateway* para coordenar os serviços, uma vez que o **External Aggregator** terá serviços mais simples. Além disso, como os dados são guardados numa base de dados partilhada, os utilizadores do **Applicant Tracking System** têm logo acesso à informação da candidatura.

O fluxo apresentado irá, também, ser adoptado pelos restantes casos de uso relativos aos candidatos (C01 - C05).

6.2.3 Vista Implantação

No que respeita vista de implantação, existem múltiplas possibilidades. Por isso mesmo, nesta secção, apresentam-se as alternativas consideradas, das mais para as menos distribuídas.

Antes de tudo, é necessário referir que os *containers* que servirão para fornecer interfaces de utilizadores estarão obrigatoriamente dentro de um servidor Internet Information Services (IIS), devido à ferramenta interna utilizada para o desenvolvimento dos sistemas da empresa.

Primeira Alternativa

A alternativa mais evidente, ilustrada na Figura 6.8, seria a de implantar isoladamente todos os *containers*, identificados na vista lógica de nível 2.

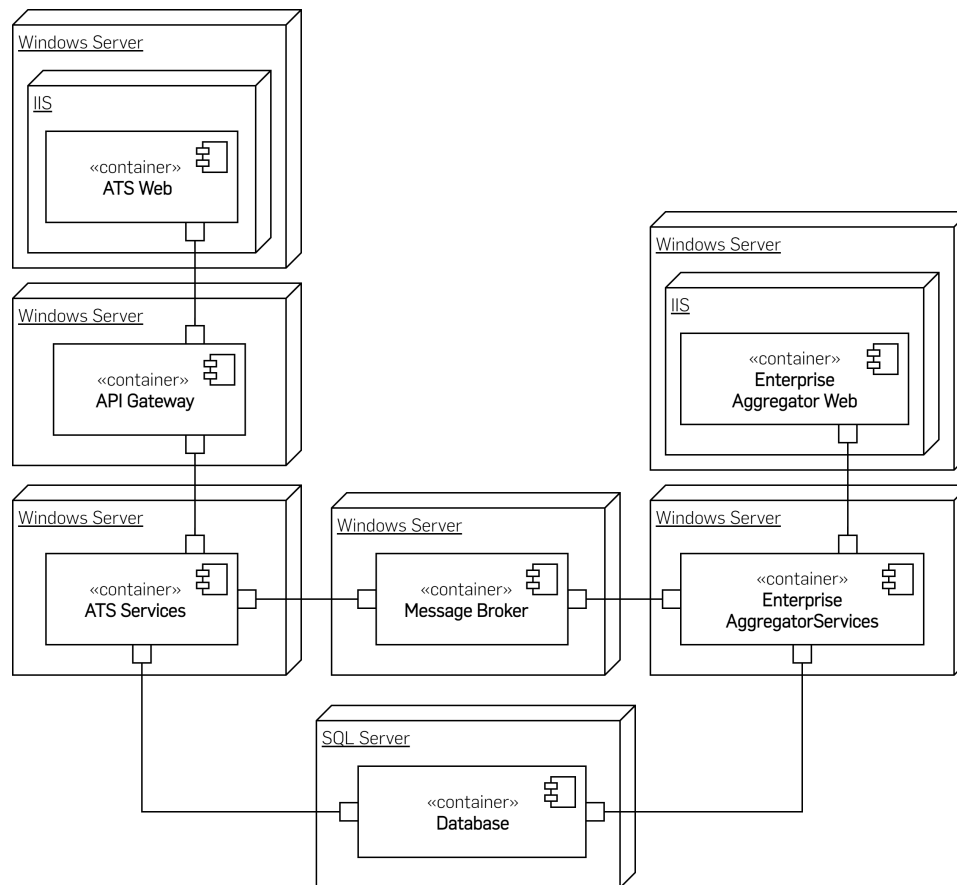


Figura 6.8: Vista de Implantação de Nível 2 - Alternativa 1

A grande vantagem desta alternativa é o desacoplamento das implantações, sendo possível alterar um dos *containers* e implantá-lo sem afectar os restantes.

Contudo, a presente abordagem não só implica mais custos, relativos à manutenção dos múltiplos servidores, como também exige uma maior preocupação com a segurança, considerando que a comunicação entre os vários *containers* - *via internet* - está mais susceptível a interferências.

A separação física dos *containers* referentes ao **Enterprise Aggregator** pode justificar os custos implicados, uma vez que o *Enterprise Aggregator Web* tem de ser exposto às redes públicas e, deste modo, por questões de segurança, os serviços permaneceriam ocultos da rede pública.

Contudo, o mesmo não acontece para o **Applicant Tracking System**, visto que fazem parte de um sistema que será disponibilizado apenas na *intranet* da empresa, não havendo tantos problemas a nível de segurança quando comparado com o agregador. Assim, a separação física dos seus *containers* só se justifica se se verificar que o sistema está sobrecarregado de pedidos.

Segunda Alternativa

Considerando o exposto anteriormente, propõe-se uma segunda alternativa, representada na Figura 6.9.

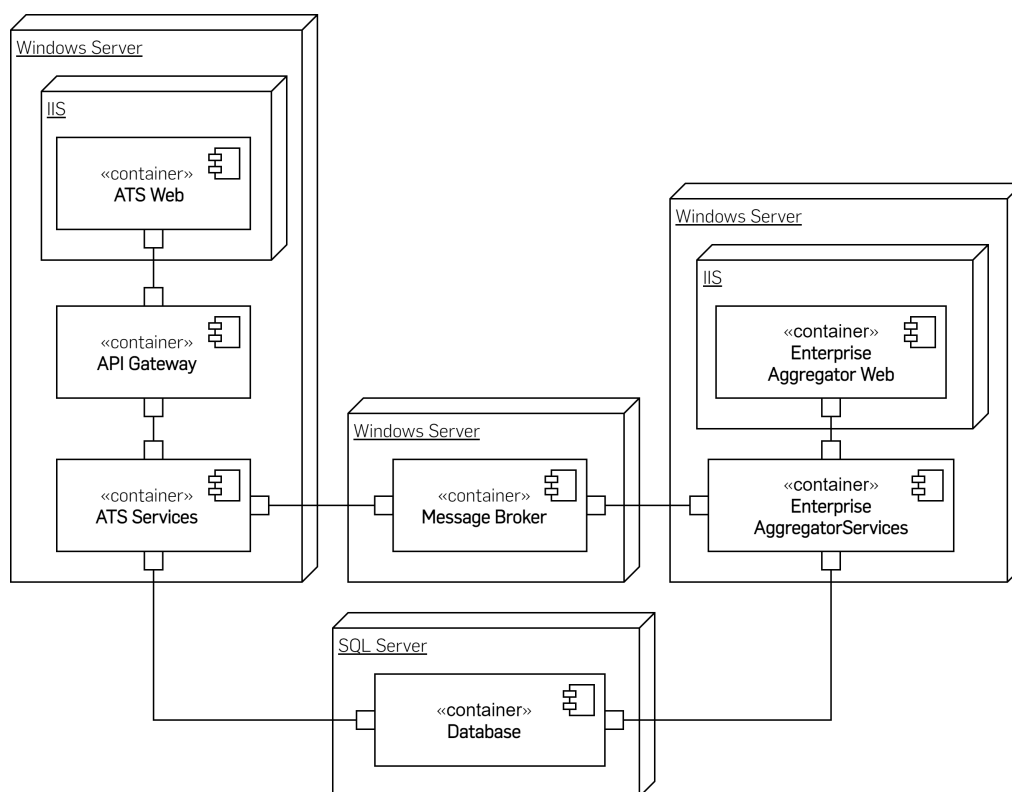


Figura 6.9: Vista de Implantação de Nível 2 - Alternativa 2

Contrariamente ao que se verifica na primeira alternativa, para esta opção agregaram-se os *containers* pelos seus respectivos subsistemas. No entanto, garante-se, do mesmo modo, o isolamento dos meios que asseguram a interoperabilidade dos mesmos.

A implantação por subsistemas reduz o número de ligações feitas entre servidores, assegurando, deste modo, que o esforço necessário para garantir a manutenção do sistema é menor do que o da primeira alternativa.

Além disso, a agregação por subsistemas não causa constrangimentos no que concerne a actualização dos mesmos. Apesar de não ser tão dividido, como a primeira opção, a alternativa proposta permite que os subsistemas e os respectivos meios de integração possam ser implantados separadamente.

Terceira Alternativa

Além das alternativas sugeridas, identificam-se mais algumas abordagens não tão díspares do já apresentado. A título de exemplo, na Figura 6.10 sugere-se uma abordagem semelhante à alternativa anterior, diferenciando, apenas, no modo em que se implantam os meios que garantem a interoperabilidade do sistema.

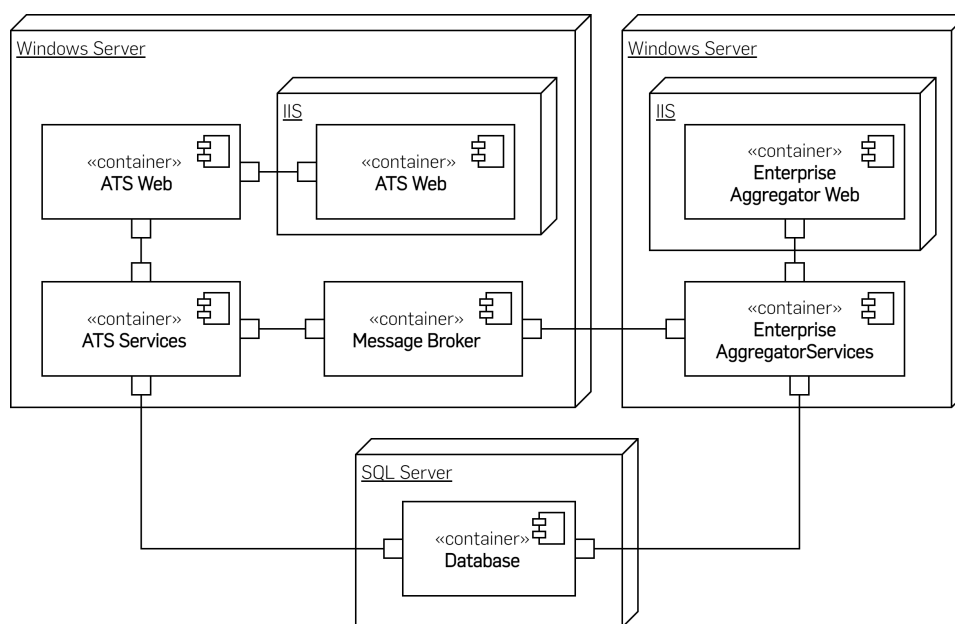


Figura 6.10: Vista de Implantação de Nível 2 - Alternativa 3

O **Message Broker** pode estar presente em qualquer um dos subsistemas, mas, considerando a questão da revelação do agregador para a rede pública, é preferível agrupar o mesmo ao subsistema do **Applicant Tracking System**.

Solução da Empresa

A escolha da empresa para a implantação do sistema foi ponderada com o cliente. Sendo que, numa fase inicial, onde a solução será utilizada para acompanhar os processos de recrutamento que estão a ser realizados na solução antiga do cliente, a implantação deverá ser o mais simples possível.

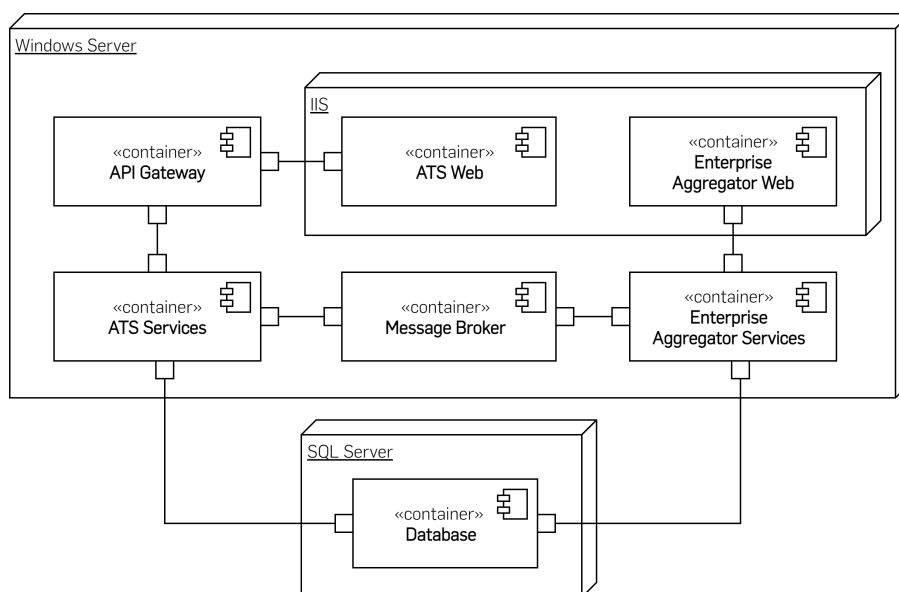


Figura 6.11: Vista de Implantação de Nível 2 - Alternativa 2

Assim sendo, apesar das suas vantagens, nenhuma das alternativas apresentadas até agora vai de encontro com as exigências do cliente onde a implantação irá ser feita inicialmente.

Numa fase inicial, a solução pode ser toda implantada num *Windows Server* e num *SQL Server*, como se pode observar na Figura 6.11.

Tanto os *containers* do subsistema responsável pela gestão de todo o processo de recrutamento e selecção, como os do subsistema responsável pela gestão de anúncios e candidaturas ficam no mesmo servidor.

Como já é prática da empresa, os componentes encarregues das interfaces dos utilizadores são postos dentro do IIS e os componentes dos serviços em si são expostos pelo servidor como *Windows Services*. Com esta disposição não só a configuração é mais rápida por parte da empresa, como os custos são mais baixos para o cliente.

6.3 Nível 3 - Componentes de Contentores

Enquanto que os níveis anteriores puderam ser descritos de forma genérica sem haver qualquer tipo de restrição imposta pelas ferramentas utilizadas, neste nível os componentes a pormenorizar devem de subscrever à estrutura utilizada nos projectos desenvolvidos pela empresa.

Nas próximas secções detalham-se apenas o *ATS Services* e o *ATS Web*.

6.3.1 ATS Services

A presente secção foca-se no desenho do *ATS Services*.

Vista Lógica

Na Figura 6.12 apresenta-se-se o diagrama de componentes do *container*.

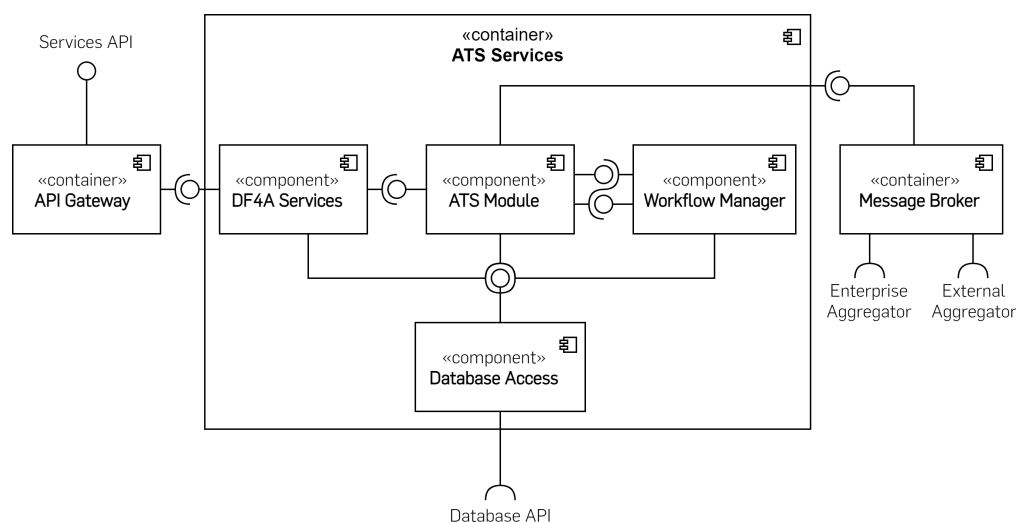


Figura 6.12: Vista Lógica de Nível 3 - ATS Services

Tendo em consideração as práticas adotadas pela empresa, o *ATS Services* foi dividido em 4 componentes distintos:

- **DF4A Services** - Módulo da empresa que contém a implementação das operações genéricas do sistema (operações CRUD) e que redirecciona as chamadas para os módulos internos necessários, neste caso o **ATS Module**.

O sistema está preparado para receber mensagens REST e SOAP, contudo, para a solução a ser desenvolvida utilizou-se REST.

- **ATS Module** - Módulo que contém a implementação dos processos elementares de negócio executados para sistema de RS.
- **Workflow Manager** - Módulo da empresa responsável pela gestão dos fluxos de trabalho existentes no sistema de RS.
- **Database Access** - Camada utilizada pelas soluções da empresa cuja responsabilidade incide no acesso à base de dados.

Vista Processos

Os fluxos seguidos no presente nível têm algumas imposições devido à utilização da *framework* interna da empresa. Esta obriga a que o componente **Services** receba todos os pedidos a processar, com os dados necessários para tal.

Caso sejam operações CRUD, esse mesmo componente saberá como os executar (semelhante ao apresentado na secção 2.3.1); caso sejam operações mais complexas, os serviços recorrem ao **ATS Module**.

No sistema a ser desenvolvido, dentro das operações executadas pelo **ATS Module** existem três possíveis situações. Na primeira, as operações recorrem apenas a esse componente para as executar, sem necessitarem de comunicar com qualquer outro elemento. Nas restantes, o **ATS Module** necessita de comunicar com o **Workflow Manager** ou com o **Message Broker**.

Nesta secção, apresentam-se fluxos das duas últimas situações e, para cada uma delas, identificam-se os casos de uso que seguirão um fluxo semelhante aos dos exemplos apresentados.

Recorrendo ao Workflow Manager

Dos casos de uso identificados que recorrem ao **Workflow Manager**, vão ser apresentados os fluxos de dois: *Enviar Pedido de Recrutamento para Aprovação* e *Actuar Sobre um Pedido de Aprovação*.

Como se pode ver pelas Figuras 6.13 e 6.14, em ambos os casos as chamadas vindas do **API Gateway** são recebidas pelos serviços e reencaminhadas para o seu módulo respectivo.

No primeiro (Figura 6.13) começa-se por validar a informação do pedido de recrutamento que será enviado para aceitação e, caso se encontre dentro das normas estabelecidas, regista-se na base de dados a alteração do estado do pedido. Logo após, o **ATS Module** chama o **Workflow Manager** para que este possa iniciar o fluxo de aprovação.

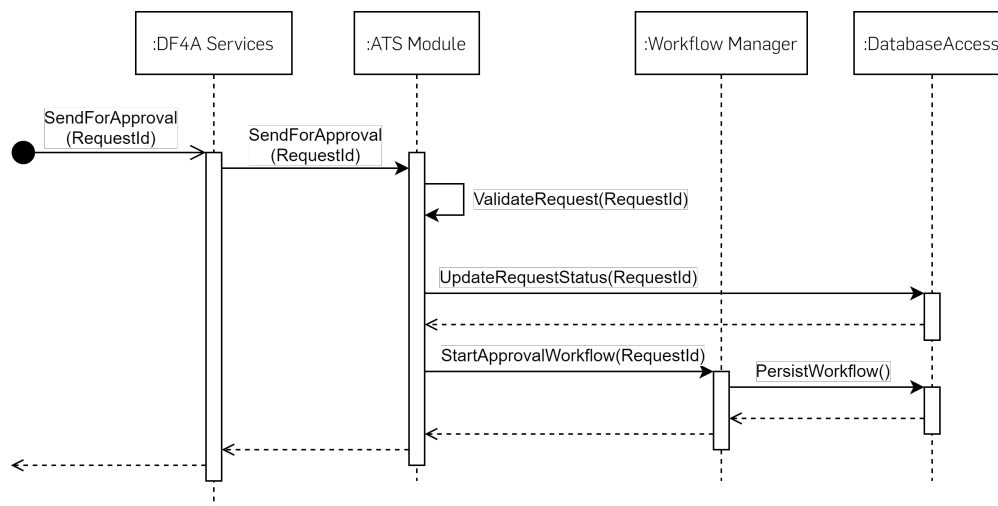


Figura 6.13: Vista de Processos de Nível 3- ATS Services -
Enviar Pedido de Recrutamento para Aprovação

É de notar que, nesta primeira situação, tanto o **ATS Module**, como o **Workflow Manager**, recorrerem ao **Database Access**, no entanto, não é algo que aconteça em todos os casos de uso semelhantes.

A título de exemplo, no segundo caso de uso, representado na Figura 6.14, apenas o **Workflow Manager** invoca o **Database Access**.

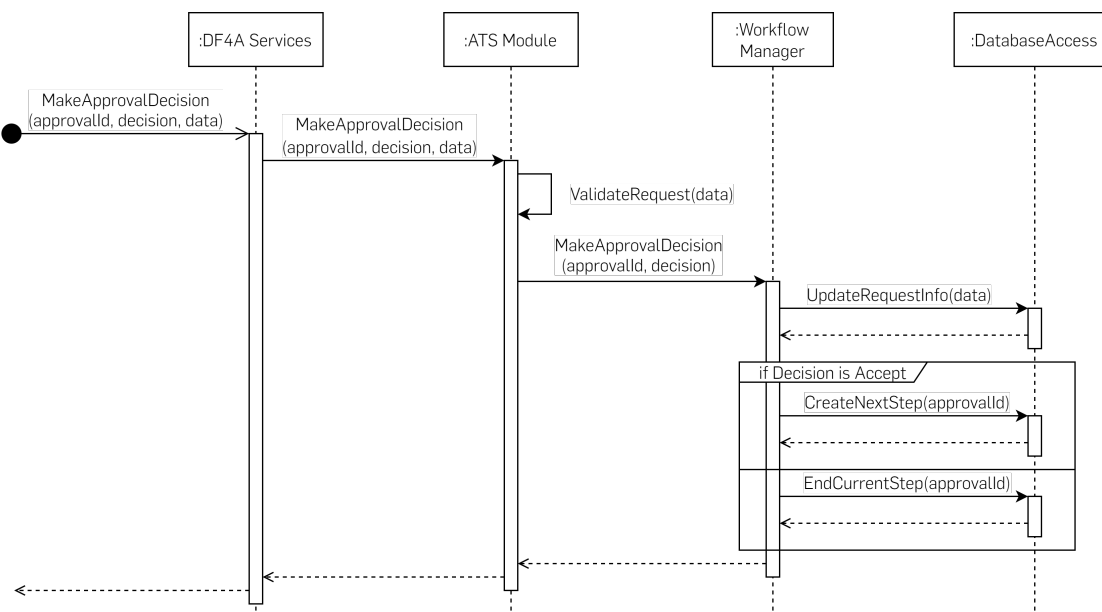


Figura 6.14: Vista de Processos de Nível 3- ATS Services -
Actuar Sobre um Pedido de Aprovação

Como se verifica na Figura 6.14, para o segundo caso de uso, o **ATS Module** valida os dados do pedido que foram alterados e chama o **Workflow Manager** para que o fluxo possa continuar. Este vai chamar o **DatabaseAccess** para guardar as alterações feitas ao pedido de recrutamento e, dependendo da decisão do colaborador sobre a aprovação do pedido, o fluxo pode ser terminado, ou pode-se passar para a próxima etapa do mesmo.

Relativamente aos casos de uso que seguirão um fluxo idêntico aos apresentados nesta secção tem-se: AP02 a AP05, PCR04, PCR05, PCR07, GC02 e GE03.

Recorrendo ao Message Broker

No que concerne os processos do **Applicant Traking System** que necessitam de recorrer ao **Message Boker**, identificam-se, apenas, aqueles relativos à publicação de anúncios (PA02 - PA04).

Na Figura 6.15, ilustra-se o fluxo seguido na publicação de um anúncio (PA02).

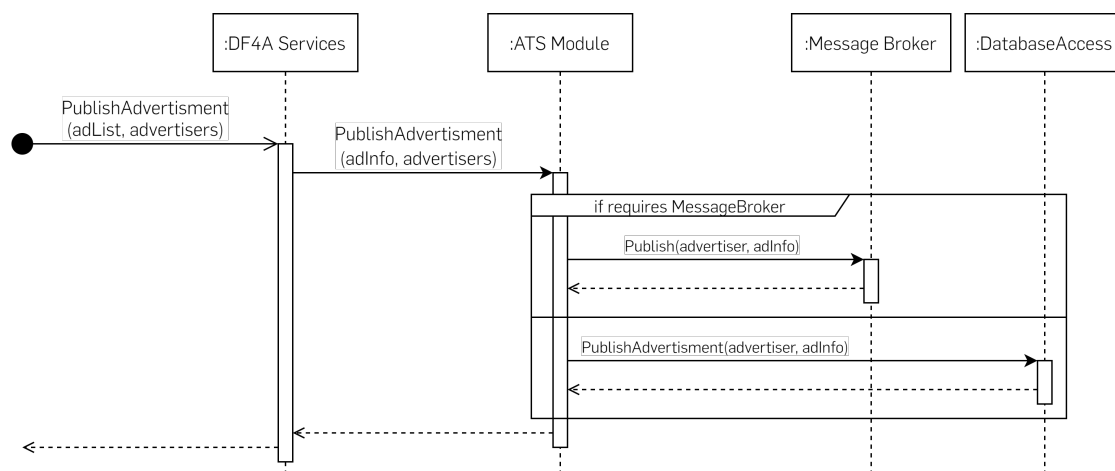


Figura 6.15: Vista de Processos de Nível 3 - ATS Services - Publicar Anúncio

Como se pode ver, tal como os processos apresentados anteriormente, o pedido passa pelo componente de serviços. Nesta situação, ele é devidamente processado pelo **ATS Module** que recorrerá ao **Message Broker** para comunicar com o **External Aggregator**.

Há duas opções para a publicação:

1. Recorre-se ao **Message Broker** para a publicação dos dados que serão consumidos pelos devidos subcritores subcritores (**External Aggregator** e **Enterprise Aggregator**) cuja responsabilidade se resume na publicação dos anúncios;
2. Recorre-se ao **ATS Module** para a publicação dos anúncios no site interno (**Enterprise Aggregator**) e o **Message Broker** serve para fazer chegar os dados ao **External Aggregator** cuja responsabilidade é a publicação de anúncios em sites externos.

No diagrama apresenta-se a segunda opção, isto porque foi a escolhida pela empresa para a implementação inicial dos anúncios internos, uma vez que é a mais rápida de implementar.

Contudo, como a separação de responsabilidades mostra ser vantajosa a longo prazo, dever-se-á fazer uma migração para primeira metodologia mal seja possível.

6.3.2 ATS Web

A presente secção foca-se no desenho do *ATS Web*.

Vista Lógica

Na Figura 6.16 apresenta-se-se o diagrama de componentes do *container* a ser explorado.

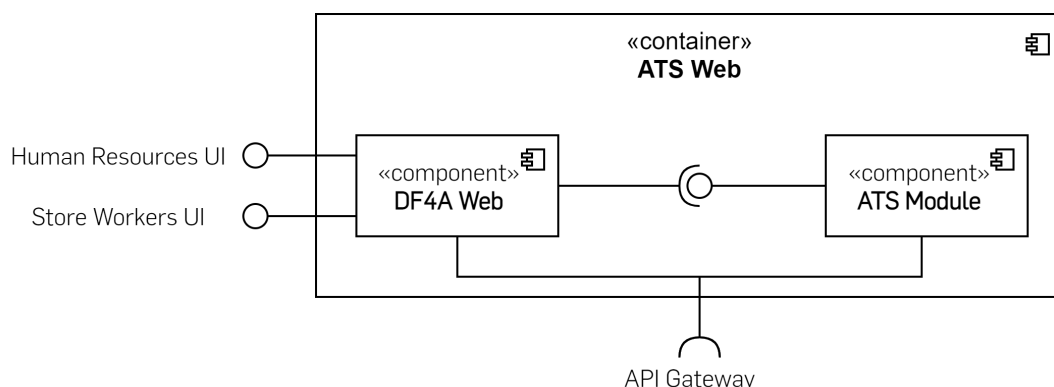


Figura 6.16: Vista Lógica de Nível 3 - ATS Web

Tendo em consideração as práticas habituais da empresa, o *container* aqui descrito está dividido nos seguintes componentes:

- **DF4A Web** - Módulo da empresa que contém a implementação genérica para gerar as páginas *web* a partir dos ficheiros de configuração, presentes no módulo específica de cada aplicação;
- **ATS Module** - Módulo que contém as configurações das páginas *web* que deverão ser geradas para esta aplicação.

É de referir, também, que as duas interfaces fornecidas pela *DF4A Web*, representadas na Figura 6.16, são, na verdade, uma única interface REST. No entanto, representam-se essas as duas para que seja compreensível que esse componente estará responsável pelos pedidos das duas interfaces.

Vista de Processos

O fluxo seguido neste *container*, idêntico ao fluxo empregue pela solução actual na geração de páginas (secção 2.3.1), encontra-se ilustrado na Figura 6.17.

Como se verifica pela análise diagrama apresentado, o *DF4A Web* recebe o pedido e, recorrendo ao *ATS Module*, consulta a configuração da página requisitada. Com essa informação, o componente é capaz de gerar a página HTML e retorná-la ao utilizador.

Note-se que, na Figura 6.17, o actor apresentado é o *employee*, uma vez que qualquer colaborador da empresa tem acesso a este processo.

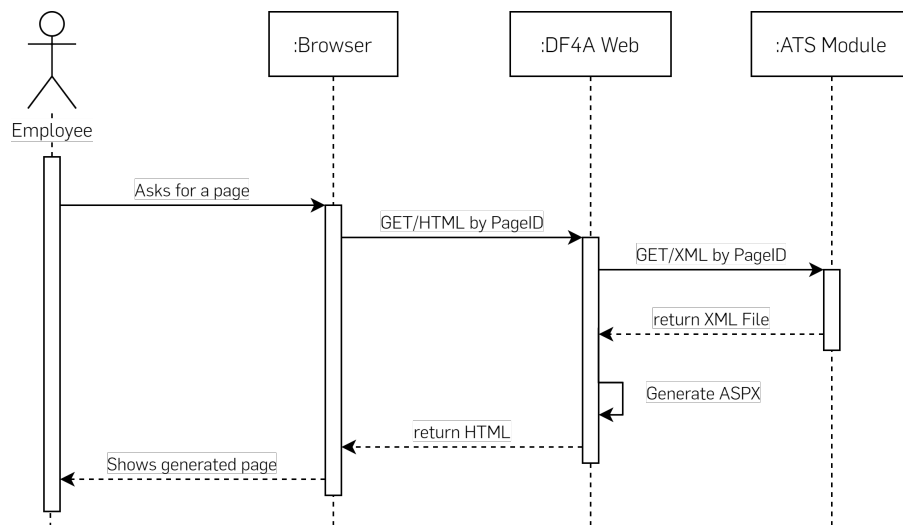


Figura 6.17: Vista de Processos de Nível 3 - ATS Web

6.4 Nível 4

Considerando as práticas actualmente adotadas pela empresa, identificou-se uma oportunidade de melhoria no que respeita arquitetura utilizada na implementação das funções ¹. Apesar de se usar uma linguagem Object Oriented (OO), o desenvolvimento segue o paradigma procedimental, não se tirando o máximo proveito das tecnologias utilizadas.

Após uma análise à implementação habitual, observa-se que além de haver bastante duplicação e uma maior desorganização de código, tem-se que o mesmo é mais difícil de ser testado e, por isso mesmo, de se verificar os requisitos levantados com o cliente.

Deste modo, foram propostas algumas alterações a nível estrutural para tirar um maior proveito das tecnologias utilizadas.

6.4.1 Vista de Packages

A utilização da framework tem algumas implicações a nível estrutural. Efectivamente, tanto o *ATS Web* como o *ATS Services* têm um módulo chamado *ATS Module* que deve seguir a estrutura apresentada na Figura 6.18.

É possível não implementar a totalidade da estrutura, por exemplo, o *ATS Services* é agnóstico relativamente aos ficheiros de configuração das páginas e menus. Todavia, adotando as práticas da empresa, a implementação dos dois *containers* foi desenvolvida num único projeto, sendo que ambos seguem esta estrutura.

Resumidamente, os ficheiros de configuração que a *framework* irá utilizar estarão dentro da *package xml*. Para além desta, existirá a *tables* que terá dependência do *code*. Esta última é onde se vai localizar o código em si, sendo que todas as outras *packages* conterão apenas ficheiros de configuração.

¹Conceito relacionado com a implementação, detalhado na secção 7.4.3.

- **Persistence** - Contém a implementação actual do acesso à base de dados;
- **Dto** - Contém os Dtos usado pelo *controller*.
- **Model** - Contém o modelo usado para responder aos requisitos do sistema.

Note-se que, a infraestrutura não está directamente relacionada com a *Persistence*, apenas com a *Factories* e a *PersistenceInterface*. Deste modo, se houver a necessidade de mudar a implementação da camada de persistência, cria-se a nova implementação e, quando se quiser começar a utilizar a mesma, basta apenas mudar a instanciação dentro das *Factories*; não sendo necessário mudar a implementação dos serviços em si.

6.4.2 Functions

Functions é o nome que se dá às classes responsáveis por receberem os pedidos feitos ao *ATS Module* do *ATS Services*, validar os dados dos mesmos e executar os serviços necessários para responder ao pedido.

Após uma análise ao código desenvolvido pela empresa, identificaram-se pontos comuns entre as *functions*; sendo que, na Figura 6.20 representa-se, de uma forma geral, o fluxo seguido pelas funções desenvolvidas pela empresa

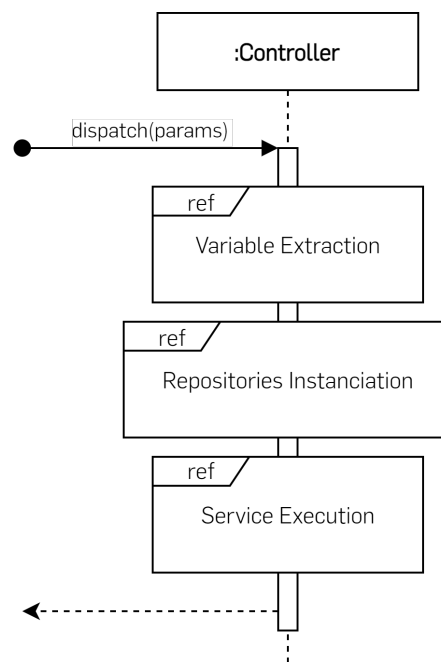


Figura 6.20: Vista de Processos de Nível 4 - Visão Geral das Funções

Como a Figura 6.20 evidencia, há três etapas na execução de uma função.

Primeiramente, são extraídos e validados todos os valores passados para a função. Para tal, deve haver um módulo responsável pelo tratamento destes mesmos dados. Assim, não só se garante uniformidade na solução toda, como também passa a ser possível testar se a extracção dos dados é devidamente feita em todas as circunstâncias.

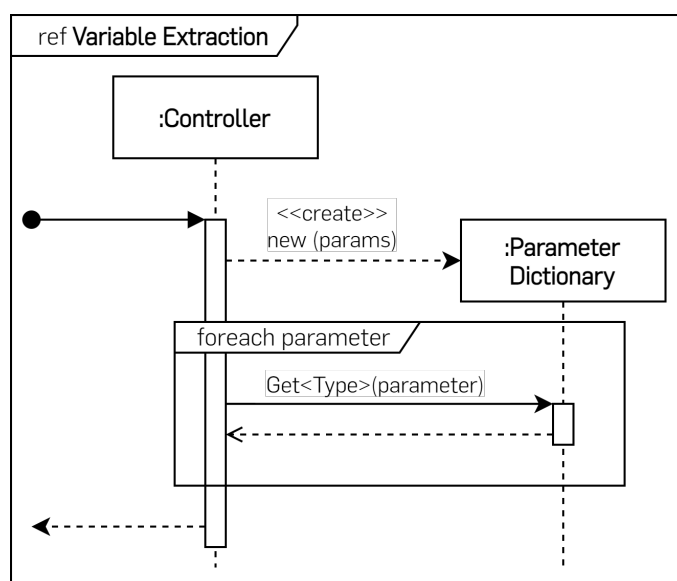


Figura 6.21: Vista de Processos de Nível 4 - Extracção de Variáveis

O módulo criado com essa responsabilidade em mente é o **ParameterDictionary**, como se verifica na Figura 6.21. Este não só valida se os dados recebidos estão em conformidade com aquilo que a aplicação necessita, como fornece uma interface mais adequada para extrair os dados que se utilizam mais à frente.

Na etapa seguinte instanciam-se as classes responsáveis pelo acesso à base de dados.

A comunicação com a mesma era feita através de duas implementações distintas, que foram abstraídas com o intuito de minimizar o desenvolvimento futuro e melhorar a manutenção dos produtos desenvolvidos (secção 6.4.3 para uma exposição mais detalhada do que foi feito).

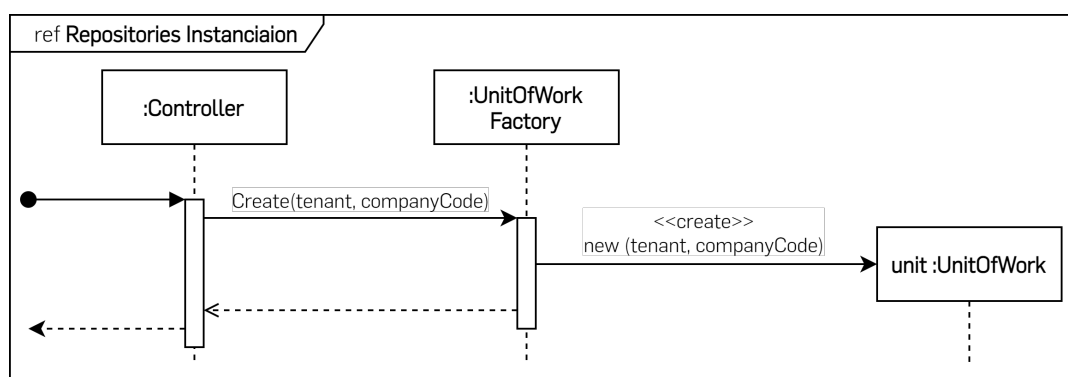


Figura 6.22: Vista de Processos de Nível 4 - Acesso à Base de Dados

Como se vê na Figura 6.22, o acesso à base dados, actualmente, é feito recorrendo a uma *UnitOfWork* criada com base no *tenant* e no *company code* passado para a fábrica.

Por fim, na terceira e última etapa, ilustrada na Figura 6.23, cria-se a transacção necessária e executa-se o processo de negócio indicado.

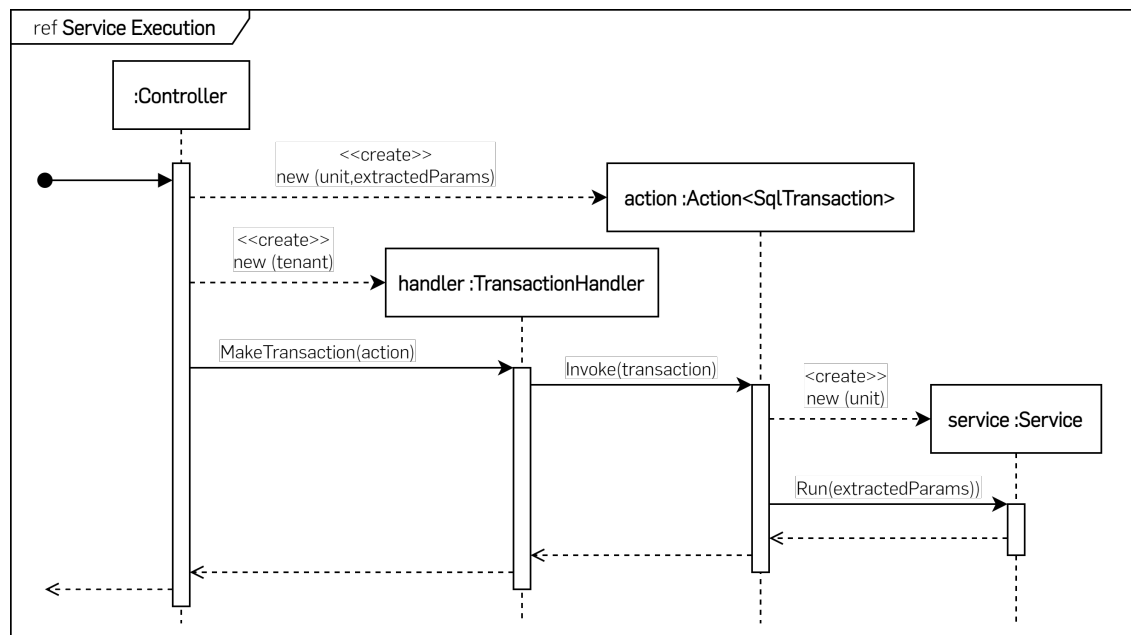


Figura 6.23: Vista de Processos de Nível 4 - Execução do Serviço

Dependendo do processo a executar, pode ser necessário criar uma transacção para garantir que não há conflitos na persistência e na consulta relativamente à base de dados. Para isso, extraiu-se a responsabilidade para o **TransactionHandler**, que além de criar a ligação, também sabe como resolver todas as excepções que forem lançadas.

Como mudar as *functions* para componentes não estáticos era uma impossibilidade, optou-se por criar o conceito de **Service** que, no presente contexto, seriam as classes onde se implementariam as funcionalidades necessárias.

Deste modo, além de o sistema de tornar mais modular, torna-se mais fácil testar os serviços.

Resumidamente, como se pode ver na Figura 6.23, cada *Controller* criará uma *Action* que será passada para o *TransactionHandler* e que será executada com a transacção criada por este último. Cada *Action* instanciará o(s) seu(s) serviço(s) e chamará os métodos necessários. A chamada do método *Run* serve apenas como uma generalização.

6.4.3 Estrutura dos Repositórios

A consulta e persistência dos dados no *ATS Module* do *ATS Service* passaram a ser responsabilidade dos repositórios desenvolvidos, que seguem o desenho apresentado na Figura 6.24.

A análise da figura apresentada deve ser feita de baixo (classes utilizadas actualmente pela empresa) para cima (abstracções criadas para a implementação de repositórios). No entanto, a camada de baixo continua a ser referida como a última.

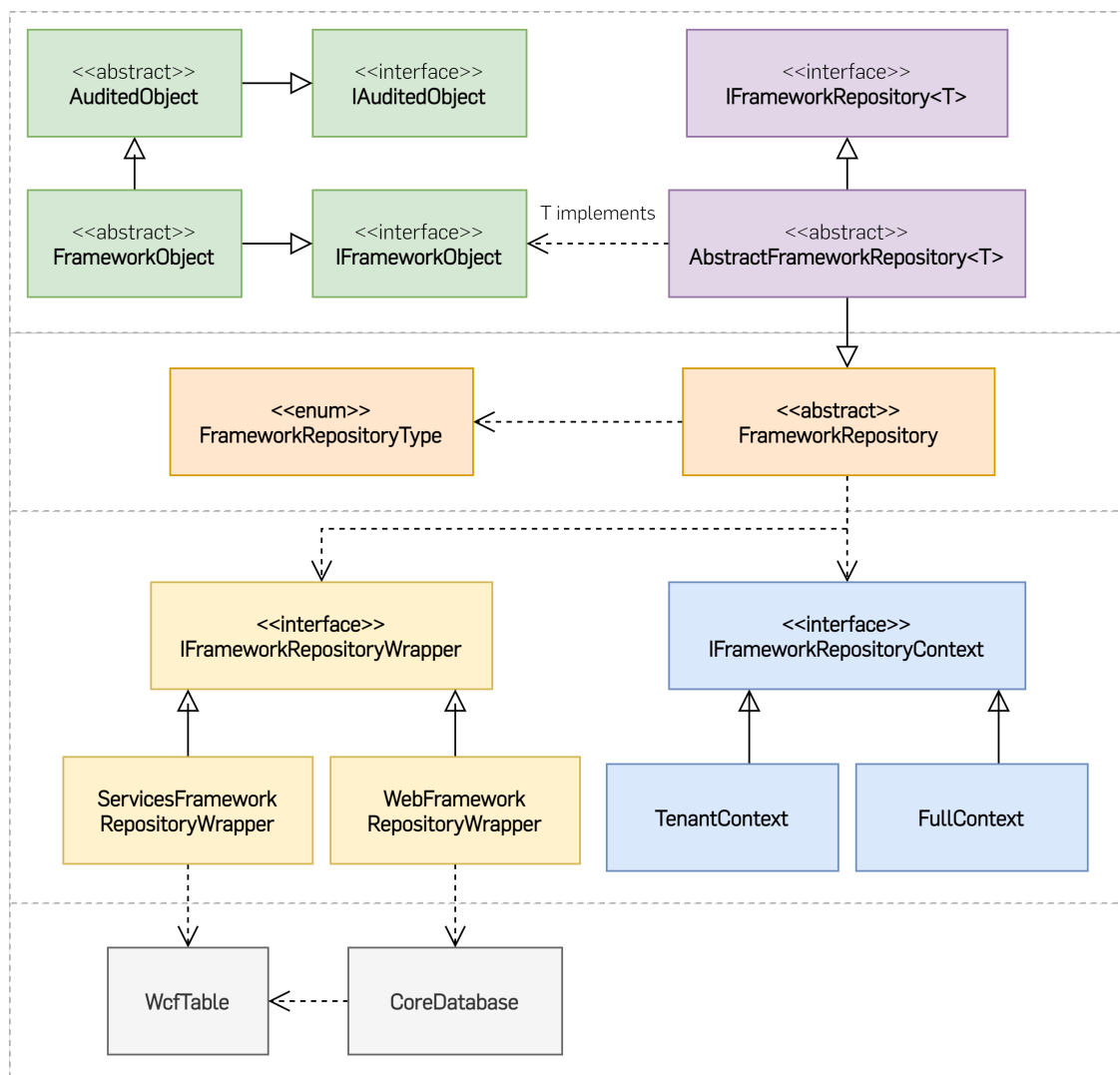


Figura 6.24: Diagrama de Classes relativo aos Repositórios

Assim, na última camada encontram-se duas implementações distintas de acesso à base de dados que são usadas actualmente. A implementação utilizada vai depender do componente (Services ou Web) - **WcfTable** ou **CoreDatabase**, respectivamente - que esteja a consultar ou persistir os dados.

Subindo uma camada, a **IFrameworkRepositoryWrapper** serve como interface comum entre as duas implementações da empresa e é usada pelo **FrameworkRepository**. Deste modo, além do repositório poder ser desenvolvido de uma forma genérica; caso surjam contextos em que o acesso será feito por outros meios, será mais fácil estender as funcionalidades sem ter de alterar a implementação do repositório em si (*open/closed principle* em conjunto com o princípio da substituição de Liskov).

Ainda na mesma camada, a **IFrameworkRepositoryContext** será a interface do contexto da base de dados. Nesta situação, entende-se por contexto a informação do *Tenant* (sempre obrigatório) e todos os atributos comuns que devam ser mapeados obrigatoriamente entre as tabelas.

Para melhor compreender este conceito, na realidade da empresa existem muitos clientes que partilham uma base de dados e, por sua vez, um *Tenant* (e.g. um grupo de empresas que deseje obter uma solução comum). No entanto, não deixa de ser necessário fazer a distinção dos dados por empresa. Sendo que, para tal, haverá um conjunto de tabelas que terão um código para identificar a qual empresa os dados pertencem.

Sabendo que esses dados estarão sempre presentes nas operações à base de dados, passa-se o contexto a cada repositório e deixa de ser necessário de fazer alterações às mesmas caso surja um outro cliente que tenha um contexto diferente.

Assim, dentro da realidade da empresa surge a distinção entre **TenantContext** e **FullContext** (sem nenhum atributo e com um código de empresa, respectivamente).

Na segunda camada, tem-se o **FrameworkRepository** cuja implementação será feita com base na **IFrameworkRepositoryWrapper** e na **IFrameworkRepositoryContext** que será completamente alheia ao mapeamento dos dados recebidos da base de dados para as devidas classes.

Além disso, depende também do **FrameworkRepositoryType** que lhe for passado; ficando, também, com a responsabilidade de instanciar o **IFrameworkRepositoryWrapper** devido. No entanto, seria vantajoso passar essa responsabilidade a um componente intermédio (e.g. a uma *Factory*), de modo a isolar totalmente a implementação desta mesma classe, que ficaria fechada para modificações.

Na primeira camada, apresenta-se a abstracção das classes que vão ser usadas para guardar e manipular os dados que vêm dos repositórios e apresentam-se as abstracções que vão ser utilizadas para desenvolver os repositórios em si.

As interfaces dos **Object** são as interfaces que existem na realidade da empresa e, como a implementação dos objectos a serem desenvolvidos tem uma parte comum, decidiu-se criar as classes abstractas que são as que devem ser estendidas pelos objectos que se querem criar.

A **AbstractFrameworkRepository** e a devida interface servem para, em conjunto com os **Objetscs**, fazer o mapeamento dos vários métodos genéricos implementados no **FrameworkRepository** para as classes devidas. Deste modo, separam-se as acções genéricas de consulta do próprio mapeamento para a devida classe.

Considerando o que foi exposto na secção de *packages* no presente nível de desenho, a primeira camada da Figura 6.24 será implementada no *package* responsável pela definição da interface utilizada para comunicar com a camada de persistência (*PersistenceInterface*), enquanto que as restantes camadas serão implementadas dentro da *package* responsável pela codificação do acesso à base de dados (*Persitence*).

Sumariamente, as cores utilizadas na Figura 6.24 servem para poder agrupar os conceitos por responsabilidades:

- **Cinzento** - Implementações distintas para aceder à base de dados;
- **Amarelo** - Abstracção do acesso à base de dados, de modo a haver uma interface comum entre as múltiplas implementações;

- **Azul** - Abstracção do contexto utilizado na base de dados. Pode ser uma base de dados que utiliza apenas o *Tenant*, ou pode este ter de estar acompanhado de informações adicionais;
- **Laranja** - Abstracção dos repositórios que acedem à base de dados utilizando as interfaces comuns criadas sem ter qualquer conhecimento da necessidade de mapear os dados;
- **Verde** - Abstracções base implementadas/estendidas pelas classes que serão persistidas pelos repositórios;
- **Violeta** - Abstracções que deverão ser implementadas/estendidas pelos repositórios que se desenvolvem. Esta implementação já tem em consideração o mapeamento dos dados recebidos para as devidas classes.

Capítulo 7

Construção

O presente capítulo tem como intuito a apresentação da construção da solução para o problema respectivo. Apenas as partes da solução final desenvolvidas pelo aluno são abordadas.

7.1 Contexto Tecnológico da Empresa

O desenvolvimento da aplicação está assente na *framework* interna da empresa, como mencionado em capítulos anteriores. A DF4A permite desenvolver desde as páginas HTML até à estrutura da base de dados.

Posto isto, os componentes são definidos através de um *Schema* de XML específico, sendo relevante demonstrar como cada um é criado.

Nas secções seguintes apresentam-se esses mesmos *Schemas* e explica-se como foram utilizados aquando da construção da solução. Posteriormente, apresentam-se outros conceitos da *framework* relevantes para a implementação do projecto.

7.2 Base de Dados

A *framework* permite, no que respeita a configuração da base de dados, não só definir as tabelas e as colunas, como também configurar as propriedades das mesmas.

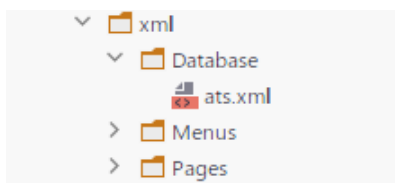


Figura 7.1: Exemplo Directório da Base de Dados

Como se verifica na Figura 7.1, o ficheiro de configuração das tabelas da base de dados deve ser posto dentro do directório *Database* que, por sua vez, se encontra no directório *xml*.

Para o sistema actual, o nome dado ao ficheiro foi **ats**, acrónimo de *Applicant Tracking System*.

Para se conhecer o *Schema* utilizado na definição da base de dados apresenta-se, no Excerto 7.1, parte da configuração de uma das tabelas do sistema **ATS Request**.

```

1 <Table Name="ATS Request">
2   <Column Name="#Code" Counter="ATS_REQUEST|Created At" />
3   <Column Name="#Company Code" />
4
5   <Column Name="Description" Null="true"
6     Label="pt-PT=Título do Pedido;"
7     Size="4000" />
8
9   <Column Name="DesiredStartDate" Null="true"
10    Type="Date"
11    Label="pt-PT=Data de Início Pretendida;" />
12
13  <Column Name="EndDate" Null="true"
14    Type="datetime"
15    Label="pt-PT=Data Finalização do Processo;" />
16
17  <Column Name="BaseSalary" Null="true"
18    Type="decimal"
19    Label="pt-PT=Vencimento Base;" />
20
21  <Column Name="Confidentiality"
22    Type="bit"
23    Label="pt-PT=Confidencialidade;" />
24
25  <Column Name="Vacancies" Null="true"
26    Type="int"
27    Label="pt-PT=Número de Vagas;" />
28
29  <CColumn Name="ProcessStatus"
30    Label="pt-PT=Estado do Processo"
31    Expression="select Status
32      from [#TENANT#ATS Process]
33      where [Request Code] = mt.Code" />
34
35  <Query>
36    <Add Name="GetRequestByManager"
37      Params="Username"
38      ParamsTypes="string">
39      <![CDATA[
40        <!-- Query statement should be put here -->
41      ]]>
42    </Add>
43  </Query>
44
45  <Constraint>
46    <add Type="PK" Columns="Code,Company Code" />
47  </Constraint>
48 </Table>

```

Excerto de Código 7.1: Definição Parcial da Tabela ATS Request

O elemento raiz do ficheiro será a *Database tag* que conterá uma ou mais *Tables*. Para definir o nome das tabelas utiliza-se o atributo *Name* e, para cada coluna que se desejar, deve-se definir uma *Column tag*. Estas últimas, tal como a *Table*, terão um atributo *Name* para definir seu nome.

Como é bastante comum as tabelas definirem colunas com formatos e nomes iguais, como é o caso do *Code*, *Company Code* e *Entry No*, há um conjunto de configurações pré-estabelecidas que se podem usar como base. Para utilizar essas predefinições é necessário dar o nome da configuração que se quer aplicar precedido de um cardinal (#) (linhas 2 e 3).

Por omissão, as colunas são criadas com o tipo **nvarchar** (linha 5), usado para guardar *strings*. Contudo, é possível indicar o tipo de dados a persistir para cada uma delas, através do atributo *Type*. Além do valor por omissão, pode ser um dos seguintes:

1. **Date**, representando uma data (linha 10);
2. **datetime**, representando uma data e hora (linha 14);
3. **decimal**, representando um número com casas decimais (linha 18);
4. **bit**, representando um booleano (linha 22);
5. **int**, representando um número inteiro (linha 26).

Existem, ainda, outros tipos de dados que, por serem mais complexos, são apresentados separadamente, de seguida.

Primeiro existem as **Option** que são utilizadas para persistir listas que não vão ser alteradas durante a execução dos programas. Como exemplo, no Excerto 7.2 mostra-se a configuração da coluna do estado dentro da tabela dos pedidos de recrutamento.

```
1 <Column Name="Status" DefaultValue="0"
2     Label="pt-PT=Estado;en-EN=Status;es-ES=Estado"
3     Type="Option"
4     Options="pt-PT=EmPreenchimento,EmAprovação,
5         Aprovado,Rejeitado,Cancelado"
6     OptionCodes="0,1,2,3,4" />
```

Excerto de Código 7.2: Coluna do Tipo Option - Estado do Pedido

Sendo os estados do pedido algo já definido, a utilização das *Option* é o mais adequado. Para configurar adequadamente, as várias opções devem estar listadas dentro do atributo *Options*, separadas por vírgulas. A cada opção estará associado um número que deve ser definido no atributo *OptionCode*, seguindo a mesma lógica e sequência do valor do atributo *Options*.

Em segundo, existe o **TableRef** utilizado para referenciar algum valor de outra tabela.

```
1 <Column Name="HiringReason"
2     TableRef="ATS Hiring Reason"
3     TableRefColumns="Entry No,Description"
4     TableRefMapColumns="Company Code=Company Code"
5     Label="pt-PT=Motivo da Contratação" />
```

Excerto de Código 7.3: Coluna do Tipo TableRef - Motivo da Contratação

A título de exemplo, no Excerto 7.3 mostra-se a configuração de uma coluna cujo valor será o código que identifica o motivo da contratação guardado noutra tabela. Como se verifica, é preciso indicar a coluna a referenciar (*TableRef*), a coluna que serve para identificar o valor a referenciar e a que guarda esse mesmo valor, separados por uma vírgula, (*TableRefColumns*) e, por fim, indicar as condições que se devem garantir para a referência estar devidamente feita (*TableRefMapColumns*).

No exemplo acima faz-se referência à tabela **ATS Hiring Reason**, onde se pretende guardar o **Entry No** para poder obter a **Description** dessa mesma entrada da tabela. Além disso, as únicas referências válidas são aquelas que têm o mesmo **Company Code** que o pedido de recrutamento (tabela onde a coluna está inserida).

Em terceiro tem-se o **DimensionSet**, criado, inicialmente, para ser aplicado apenas neste projecto, onde se iriam desenvolver, em paralelo, os mecanismos responsáveis pela gestão da estrutura das dimensões.

Não se usou um simples **TableRef**, pois estas configurações têm implicações nas operações CRUD realizadas nos elementos das tabelas. Uma vez que o *Dimension Set* necessita de um tratamento especial nessas operações, optou-se por criar este novo tipo.

```

1 <Column Name="Dimension"
2     Type="DimensionSet"
3     DimensionCode="ATS"
4     Label="pt-PT=Dimensão" Width="150px"/>

```

Excerto de Código 7.4: Coluna do Tipo DimensionSet - Dimensão

Para configurar a referência a um *Dimension Set*, como se verifica no Excerto 7.4, deve-se indicar qual o *Dimension Code* (código associado ao *Dimension Setup* a ser consultado). Este tipo de configuração está numa fase de desenvolvimento inicial, não se devendo dar grande foco ao mesmo, sendo que está sujeito a bastantes mudanças (e.g. não ser necessário configurar o *Dimension Code* pois este estará associado ao *tenant* utilizado).

Por fim, existem os **Domain**, que são, essencialmente, uma referência a uma tabela de domínio. Estas tabelas servem para, numa só estrutura de dados, se guardarem múltiplas listas de valores agrupadas pelo seu domínio (**Domain Code**).

```

1 <Table Name="ATS Domain">
2   <Column Name="Company Code"
3       Label="pt-PT=Empresa;en-EN=Company;"
4       Size="50" />
5
6   <Column Name="Domain Code"
7       Label="pt-PT=Domínio;en-EN=Domain;"
8       Size="50" />
9
10  <Column Name="Code"
11      Label="pt-PT=Código;en-EN=Code;"
12      Size="50" />
13
14  <Column Name="Description"
15      Label="pt-PT=Descrição;en-EN=Description;"
16      Size="250" />

```

```

17 <Column Name="Flag Edit"
18     Label="pt-PT=Permite edição;en-EN=Changes allowed;"
19     Type="bit" DefaultValue="1" />
20
21
22 <Column Name="Flag Archive"
23     Label="pt-PT=Arquivo;en-EN=Archive;"
24     Type="bit" DefaultValue="0" />
25
26 <Constraint>
27     <add Type="PK" Columns="Company Code,Domain Code,Code" />
28 </Constraint>
29 </Table>

```

Excerto de Código 7.5: Exemplo da Definição de uma Tabela de Domínio

No Excerto 7.5 encontra-se a configuração habitual de uma tabela de domínio. Como se pode ver, este tipo de tabelas guarda pares código-descrição associados a um código de domínio. Esta estrutura é utilizada, maioritariamente, para representar valores que serão configurados por e para cada cliente.

```

1 <Column Name="PreferredInstitutions" Null="true"
2     Type="domain"
3     DomainTable="ATS Domain"
4     Label="pt-PT=Estabelecimentos de Ensino Preferenciais"/>

```

Excerto de Código 7.6: Coluna do Tipo Dimension - Estabelecimentos de Ensino Preferenciais

Tendo como exemplo o Excerto 7.6, é preciso identificar a tabela de domínio onde serão persistidos os dados e, por omissão, o *Domain Code* associado será o nome da coluna desse tipo. Porventura, se for necessário alterar, é possível sobrepor essa configuração adicionando o atributo *DomainCode* à *Column tag* e indicar qual o valor que deve lá estar.

Uma coluna como *Domain* tem a capacidade de persistir múltiplos dados, para tal o que fica lá guardado são os *Codes* seleccionados, separados por *pipes* (|) se houver mais do que um seleccionado.

No exemplo apresentado, tem-se a coluna dos pedidos de recrutamento que guarda quais as instituições de ensino que devem ser priorizadas na escolha dos candidatos. Deste modo, não é necessário estar a criar uma tabela só para persistir esses valores, utilizando-se a tabela de domínio.

7.3 Interfaces dos Utilizadores

Além de configurar a base de dados, a *framework* permite criar menus e páginas para criar a interface do utilizador. Esta pode, posteriormente, ser implantada como uma solução única, encarregue da interface e dos serviços, ou pode-se optar por implantar duas soluções separadamente, ficando cada uma responsável por um dos dois.

Independentemente da implantação dos mesmos, o desenvolvimento é feito na mesma solução. Nesta subsecção, as configurações apresentadas focam-se no **Human Resources UI**, uma vez que, as outras interfaces foram, maioritariamente, desenvolvidas por outros membros da equipa.

7.3.1 Menus

A *framework* da empresa permite criar os menus através de ficheiros xml que sigam os *Schemas* apropriados. Para a sua configuração é necessário saber qual o identificador interno da aplicação, neste caso é o 111.

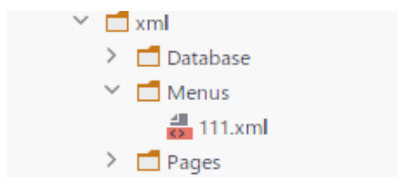


Figura 7.2: Exemplo Directório de Menus

Como se verifica na Figura 7.2, o ficheiro de configuração deve estar dentro pasta apropriada, e o seu nome deverá ser o identificador da aplicação. No Excerto 7.7 apresenta-se, parcialmente, o ficheiro mencionado.

```

1 <Menus xmlns="http://tempuri.org/WebMenuSchema.xsd">
2   <Menu Code="111" Title="pt-PT=ATS;en-EN=ATS;" Application_Code="111">
3
4     <Option ID="111009" Url=""
5       Title="pt-PT=Pedidos" Tooltip="pt-PT=Pedidos"
6       Icon="fal fa-file-alt" ObjectID="">
7   </Option>
8   <Option ID="111002" Parent="111009"
9     Url="/framework/PageList.aspx?IDPagina=111002.0"
10    Title="pt-PT=Pedidos de Recrutamento;en-EN=Recruitment Requests"
11    Tooltip="pt-PT=Pedidos de Recrutamento;en-EN=Recruitment Requests"
12    Icon="fal fa-user-plus" ObjectID="111002.0">
13  </Option>
14
15  <Option ID="111008" Url="/framework/PageList.aspx?IDPagina=111008.0"
16    Title="pt-PT=Lista de Candidatos;en-EN=Applicant List;"
17    Tooltip="pt-PT=Lista de Candidatos;en-EN=Applicant List;"
18    Icon="fal fa-address-book" ObjectID="111008.0">
19  </Option>
20 </Menu>
21 </Menus>

```

Excerto de Código 7.7: Configuração de um Menu

No Excerto 7.7 apresenta-se parte da configuração do menu. Pelo apresentado pode-se inferir que o menu, ao ser apresentado, terá duas opções (Pedidos e Lista de Candidatos) e que a primeira poderá ser expandida para apresentar uma terceira opção (Pedidos de Recrutamento).

A *tag* base dos menus será *Menus* e deverá ter dentro de si uma outra *tag* - *Menu*. Os atributos do *Menu* servirão para identificar a qual aplicação ao qual ele pertence. Além disso, dentro dessa *tag* estarão descritas todas as *Options* que o menu deve oferecer.

Cada *Option* deverá ter um identificador próprio, um título e, opcionalmente, uma *tool-tip*. Dependendo do tipo de *Option*, pode-se, ou não, definir o *Url* - valor que indica qual página abrir. Como por vezes é necessário agregar muitas opções dentro de uma só, necessitando de expandir para ver quais as opções dentro dessa mesma, existe o conceito de *Parent*.

Seguindo o Excerto 7.7, a primeira *Option* não tem nenhum *Url* definido pois não vai servir para redireccionar para nenhuma página. Como se pode ver na *Option* definida logo a seguir, a primeira serve como *Parent* desta última. Isto indica que, para o utilizador conseguir aceder à segunda opção deve, primeiramente, expandir os **Pedidos**.

Além do mais, é possível mudar o ícone que aparece em cada opção do menu e deve-se definir um *ObjectId*. Este identificador será depois associado aos perfis de utilizador que têm permissão para ver e aceder a essa opção do menu.

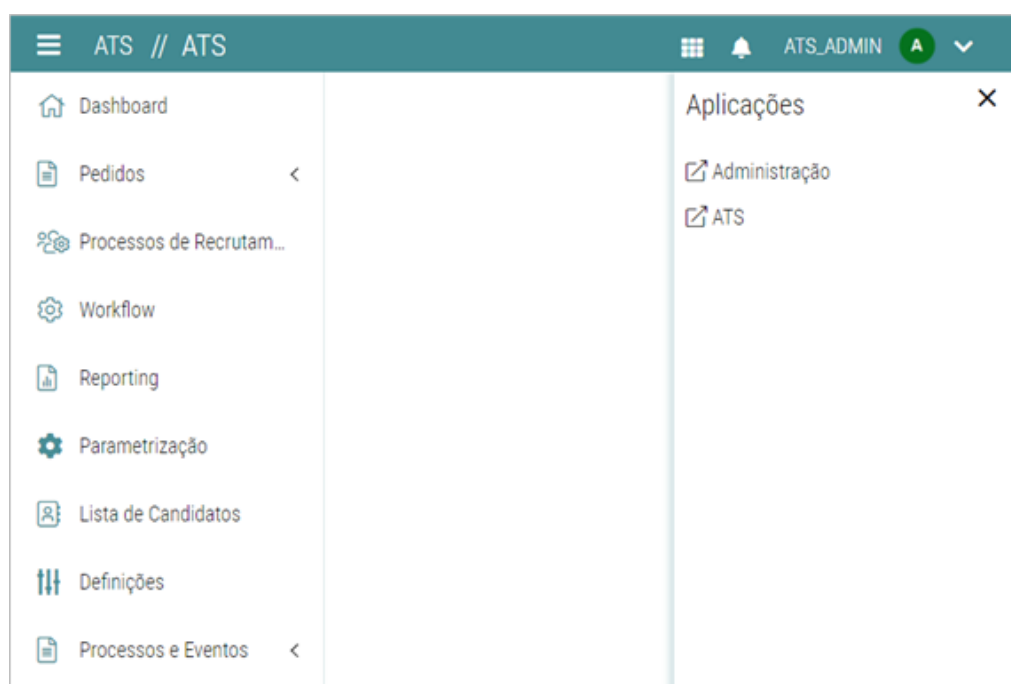


Figura 7.3: Menu Gerado pela *Framework*

O menu gerado pelo ficheiro total pode ser visto na Figura 7.3. Como se pode ver, existo uma opção de **Lista de candidatos** e o item **Pedidos** que pode ser expandido para ver as suas opções.

7.3.2 Páginas

Com a *framework* da empresa também se podem configurar as páginas que são apresentadas ao utilizador. Tal como nas configurações anteriores, recorre-se a ficheiros xml com *Schemas* já definidos.

Para criar as páginas deve-se conhecer o identificador da aplicação, pois este vai ser o nome dado ao directório onde os ficheiros xml devem ser colocados, como demonstrado na Figura 7.4.

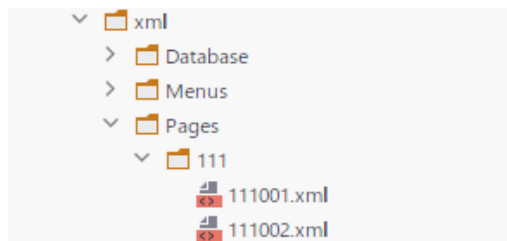


Figura 7.4: Exemplo Directório de Páginas

Cada ficheiro xml representa uma página e deverá ter um nome que começa com o identificador da aplicação e preenche o restante com números até perfazer um total de seis algarismos. Para haver uma nomenclatura uniforme entre aplicações, o normal é começar a preencher com o valor 1 e indo incrementado à medida que mais páginas vão sendo feitas.

No que respeita a criação de páginas com a *framework*, existem três categorias: *Page List*, *PageFormulario* e *PageFullEdit*.

Além disso, é possível criar o que a empresa chama de páginas dinâmicas (*Web Forms*) e fazer referência às mesmas através das configurações. Para compreender melhor cada uma delas, é apresentado um exemplo de cada categoria.

PageList

Uma *PageList* consiste numa página que lista os valores de uma determinada tabela.

```

1 <Pagina IDPagina="111008.0" Title="pt-PT=Lista de Candidato">
2   <Table Name="ATS Applicant" DataKeyNames="Company Code"/>
3
4   <NavBar> /* ... */ </NavBar>
5   <Parametros> /* ... */ </Parametros>
6   <Grid> /* ... */ </Grid>
7
8 </Pagina>
```

Excerto de Código 7.8: Configuração da PageList - Visão Geral

Os primeiros dados a ser definidos devem ser o identificador da página (*IDPagina*), o título da mesma (*Title*); seguidos da identificação da tabela que vai ter os seus valores listados (linha 2). Além disso, através do Excerto 7.8, observa-se que a configuração destas páginas está dividida em três propriedades: a *NavBar*, os *Parametros* e a *Grid*.

A primeira propriedade - *NavBar* - configura-se as acções que devem aparecer por cima da listagem.

```

1 <NavBar>
2   <add Type="Refresh" />
3   <add Text="pt-PT=Ações">
```

```

4      <add Type="New" Text="pt-PT=Novo Candidato"
5          Url="/framework/PageFormulario.aspx?IDPagina=111008.01"
6          WindowType="PopUp" WindowSize="1200" />
7      <add Type="New" Text="pt-PT=Convidar Candidato"
8          Url="/framework/PageFormulario.aspx?IDPagina=111008.02"
9          WindowType="PopUp" WindowSize="1200" />
10     </add>
11 </NavBar>

```

Excerto de Código 7.9: Configuração da PageList - NavBar

Como se verifica no Excerto 7.9, não é só possível adicionar botões com determinadas acções (neste caso, apenas redireccionam para outras páginas), como também se podem agrupar esses mesmos botões num item só, que abrirá um menu *dropdown* com os botões que foram definidos lá dentro.

A segunda propriedade - *Parametros* - serve para definir quais os filtros que podem, ou devem, ser aplicados à lista que se vai mostrar.

```

1 <Parametros>
2   <Formulario>
3     <Hidden>
4       <Add ID="2" Name="Company Code" DefaultValue="Sessao.CompanyCode"/>
5     </Hidden>
6   </Formulario>
7 </Parametros>

```

Excerto de Código 7.10: Configuração da PageList - Parametros

Para definir os parâmetros deve-se sempre colocar a *tag Formulario* e, dependendo se queremos mostrar ao utilizador, ou se abre uma *Area tag*, ou uma *Hidden tag*. No caso do Excerto 7.10, haverá apenas um filtro que o utilizador não pode alterar: o código da empresa dos valores listados deverá ser o mesmo que o código da empresa do sistema.

Por fim, a terceira propriedade - *Grid* - serve para estruturar quais os dados que deverão aparecer na lista.

```

1 <Grid>
2   <Column Name="Name"/>
3   <Column Name="NIF"/>
4   <Field Name="Invite" Width="100px" Type="Icon" Icon="fal fa-envelope"
5       IDPagina="111008.02" WindowSize="800"
6       Url="/framework/PageFormulario.aspx"
7       Label="pt-PT=Convidar"/>
8   <Field Name="LinkProcess" Width="100px" Type="Icon" Icon="fal fa-link"
9       IDPagina="111008.03" WindowSize="800"
10      Url="/framework/PageFormulario.aspx"
11      Label="pt-PT=Associar Processo"/>
12 </Grid>

```

Excerto de Código 7.11: Configuração da PageList - Grid

Como se pode ver no Excerto 7.11, é possível adicionar *Columns* ou *Fields*. As colunas que se adicionarem devem existir na tabela cujos valores vão ser listados. Os *Fields*

representam ícones que poderão ter determinadas acções, desde redireccionar o utilizador para outra página, até, por exemplo, apagar o registo representado na linha onde esse mesmo *Field* está.

No exemplo actual, cada linha apresentará o nome e o NIF dos candidatos e, além disso, terá dois ícones distintos: o primeiro - um envelope - abrirá uma página onde se poderá convidar o candidato a participar num processo; o segundo - ícone de um link - abrirá uma página onde se associa directamente o candidato a um processo.

Nome	NIF	Convidar	Associar Processo	...
Pedro Cruz	123123321			
Mariana Preto	123123123			

Figura 7.5: Exemplo PageList - Candidato

N Figura 7.7 encontra-se a página gerada através das configurações apresentadas.

PageFormulario

Uma *PageFormulário* serve para mostrar um formulário, como o nome indica, com os valores de uma determinada entrada da tabela definida. O formulário pode ser só para visualizar os dados ou pode permitir a edição dos mesmos.

```

1 <Pagina IDPagina="111008.01" Title="pt-PT=Novo Candidato">
2   <Table Name="ATS Applicant" DataKeyNames="Entry No,Company Code"/>
3   <Edit Title="pt-PT=Editar Candidato"/>
4
5   <Formulario> /* ... */ </Formulario>
6   <ActionButton> /* ... */ </ActionButton>
7 </Pagina>

```

Excerto de Código 7.12: Configuração da PageFormulario - Visão Geral

Tal como na página anterior, os primeiros dados a ser definidos são o identificador da página (*IDPagina*), o título da mesma (*Title*); seguidos da identificação da tabela que vai ter os seus valores listados (linha 2).

Adicionalmente, se for para permitir a edição dos dados no formulário, o que se verifica no Excerto 7.12, deve-se utilizar a *Edit tag*.

Se os valores recebidos pelos *DataKeyNames* conseguirem identificar apenas um registo da tabela, é aplicado o título da *Edit tag*, se não, aplicam o título definido na *tag Pagina*.

Tendo a configuração base feita, fica por definir as duas principais propriedades destas páginas: *Formulario* e *ActionButton*.

A primeira propriedade - Formulário - segue uma estrutura idêntica ao formulário dentro da tag *Parametros* da *PageList*.

```

1 <Formulario>
2   <Area Cols="3" Rows="2" Flow="H">
3     <Title ID="28" Label="pt-PT=Mobilidade;en-EN=;es-ES="/>
4     <Empty ID="29"/>
5     <Empty ID="30"/>
6     <Column ID="31" Name="GeographicMobility" CheckBoxes="true"/>
7     <Column ID="32" Name="NationalMobility"/>
8     <Column ID="33" Name="InternationalMobility"/>
9   </Area>
10  <Hidden>
11    <Add ID="21" Name="Company Code" DefaultValue="Sessao.CompanyCode"/>
12  </Hidden>
13 </Formulario>

```

Excerto de Código 7.13: Configuração da PageFormulario - Formulário

Como se verifica no Excerto 7.13, os parâmetros poderão ter uma ou mais tags *Area* que representarão um conjunto de dados a apresentar no formulário. Como a sua construção é feita à base de tabelas, pode-se definir quantas colunas e quantas linhas terá (*Rows* e *Columns*, respectivamente), e, ainda, se pode definir a orientação (*Flow*) seguida na geração das páginas (no horizontal preenche primeiro cada linha, enquanto que no vertical preenche primeiro cada coluna).

A título de exemplo, no excerto apresentado há uma área com três colunas e duas linhas que segue um *Flow* horizontal (H). Assim, a primeira linha será composta por uma célula com o título definido na tag *Title* e por duas células vazias (tag *Empty*). A segunda linha já apresentará as três células os dados existentes nas colunas identificadas.

Além disso, é possível adicionar campos invisíveis no formulário. Estes apesar de escondidos, servem como um filtro, se estiver no modo edição, ou como valor por omissão, se estiver no modo de criação e tiver definido um *DefaultValue*.

A segunda propriedade que se deve configurar nestas páginas são os *ActionButton*. Esta área servirá para colocar um ou mais botões, cada um com a sua função.

```

1 <ActionButton>
2   <Add Text="pt-PT=Gravar;en-EN=Save" Action="save"/>
3 </ActionButton>

```

Excerto de Código 7.14: Configuração da PageFormulario - ActionButton

A título de exemplo, no Excerto 7.14 existe apenas o botão de gravar. No entanto, podem-se adicionar tantas tags *Add* quantas necessárias e essas podem ser botões que redireccionam para outras páginas, ou que activem determinadas funções.

Na Figura 7.6 encontra-se a página gerada através das configurações.

Figura 7.6: Exemplo PageFormulario - Candidato

PageFullEdit

Uma *PageFullEdit* consiste numa página onde se apresentam vários *Docks*. Estes podem ser do tipo *Dynamic*, criando um formulário não editável, ou do tipo *Grid*, gerando uma lista consoante a configuração. respectivamente

```

1 <Pagina IDPagina="111002.02" Title="pt-PT=Pedido de Recrutamento">
2   <Table Name="ATS Request" DataKeyNames="Company Code,Code" />
3
4   <NavBar> /* ... */ </NavBar>
5   <Docks> /* ... */ </Docks>
6   <Panels Open="true"> /* ... */ </Panels>
7
8 </Pagina>
```

Excerto de Código 7.15: Configuração da PageFullEdit - Visão Geral

Tal como as páginas anteriores, analisando o Excerto 7.15, define-se, primeiramente, o identificador da página (*IDPagina*), o título da mesma (*Title*) e a tabela sobre a qual vai ser construída a página (linha 2).

A configuração destas páginas tem por base três propriedades: *NavBar*, *Docks* e *Panes*.

```

1 <NavBar>
2   <add Type="Anexos" Text="pt-PT=Anexos;en-EN=Attachments;"
3     Url="/framework/PageAnexos.aspx"
4     IDPagina="0.5" Method="PopUp" DockToUpdate="Anexos"
5     Parametros="Internal Key,ATS Request:{Code},TableName,ATS Request" />
6   <add Type="Medium" Name="SendForApproval"
7     Text="pt-PT=Enviar para Aprovação"
8     Icon="fal fa-paper-plane"
9     DynamicCode="ATS Request.SendForApproval"
10    VerifyColumn="Status" DisplayOnValue="0"
11    Parametros="Code,{Code},Vacancies,{Vacancies}" />
12 </NavBar>

```

Excerto de Código 7.16: Configuração da PageFullEdit - NavBar

O *Schema* da *NavBar* é igual ao apresentado para a *PageList*. Analisando Excerto 7.16, observa-se que haverá dois botões no início da página.

O primeiro servirá para redireccionar para uma página que permite adicionar anexos relacionados com o pedido que se abriu, visto que a página está a ser gerada sobre a tabela de pedidos. Para tal, recorre-se a uma página existente da *framework* - a *PageAnexos*.

O segundo não abrirá nenhuma página, em vez disso, chamará a função *SendForApproval* que está associada à tabela *ATS Request* (mais sobre estas funções na secção 7.4.3). Adicionalmente, esta opção só irá aparecer se o *Status* do pedido estiver com o valor 0 (*VerifyColumn* e *DisplayOnValue*).

A propriedade seguinte são os *Docks* e é nesta parte da configuração que se identificam as várias páginas que se irão juntar numa só.

```

1 <Docks>
2   <add ID="IdentificationArea" Title="pt-PT=Identificação do Processo;"
3     IDPagina="111002.031" Type="Dynamic"
4     AlwaysOpen="true">
5     <Command>
6       <Add Type="Edit" Text="pt-PT=Editar;en-EN=Edit;"
7         VerifyColumn="Status" DisplayOnValue="0"
8         IDPagina="111002.03" WindowType="Pop" WindowSize="1000,800"
9         Refresh="IdentificationArea"
10        Parametros="Code,{Code}" />
11     </Command>
12   </add>
13
14   <add ID="AcademicQualificationsArea" Title="pt-PT=Habilitações"
15     IDPagina="111002.08" Type="Grid" Open="false"
16     Parametros="Company Code,{Company Code},Request Code,{Code}">
17     <Command>
18       <Add Name="New" Text="pt-PT=Inserir Habilitação;en-EN=Insert Member;"
19         VerifyColumn="Status" DisplayOnValue="0"
20         IDPagina="111002.081" WindowType="Pop" WindowSize="600,600"
21         Refresh="AcademicQualificationsArea"
22         Parametros="Company Code,{Company Code},Request Code,{Code},Entry No," />
23     </Command>

```

```

24 </add>
25 </Docks>

```

Excerto de Código 7.17: Configuração da PageFullEdit - Docks

No Excerto 7.17 apresenta-se parte da configuração realizada para esta página. Para demonstrar os dois casos, está configurado um *Dock* do tipo *Dynamic*, que deve, obrigatoriamente, apontar para uma *PageFormulario* através do atributo *IDPagina*, e outro do tipo *Grid* que, por sua vez, referencia uma *PageList*.

Além disso, cada *Dock* pode ter botões com determinadas acções, usando a *tag Add* dentro da *tag Command*, como se verifica nas linhas 6 e 18. Neste excerto, ambas redireccionam para uma outra página mas é possível invocar outro tipo de acções com os botões, tal como na *NavBar*.

A terceira e última propriedade - *Pane* - serve para configurar o que aparece na lado esquerdo da página.

```

1 <Panes Open="true" Width="200px">
2   <add ID="Anexos" Title="pt-PT=Anexos;en-EN=Attachements"
3     Parametros="Internal Key,ATS Request:{Code}" />
4 </Panes>

```

Excerto de Código 7.18: Configuração da PageFullEdit - Panes

Considerando o que a empresa desenvolveu até agora, existem dois tipos de *Pane* que se utilizam recorrentemente. A primeira é uma caixa de comentários e a segunda é uma zona de anexos. Para a página a ser detalhada, como se vê no Excerto 7.18, configurou-se a segunda opção.

Figura 7.7: Exemplo PageFullEdit - Pedido de Recrutamento

Na Figura 7.7 encontra-se a página gerada pelos ficheiros de configuração completos.

Páginas Dinâmicas

Quando as páginas da *framework* não são suficientes para alguma interface, recorre-se às páginas dinâmicas. Estas podem ser desenvolvidas de raiz ou em conjunto com a DF4A, estendendo alguns dos componentes já criados.

Contudo, este tipo de geração de páginas estava a ser alterado durante a implementação da versão inicial desta solução. Isto significa que, as páginas desenvolvidas nesta altura são implementadas com duas abordagens diferentes aplicadas ao mesmo tempo. Há componentes a serem gerados tanto no lado do servidor, como no lado do cliente.

```

1 <body>
2   <form id="form1" runat="server">
3     <%: cpcit.web.framework.HtmlHelper.RenderBoundles() %>
4
5     <telerik:RadScriptManager runat="server"></telerik:RadScriptManager>
6     <telerik:RadAjaxManager runat="server" />
7
8     <cpcit:PageWrapper runat="server" Titulo="Formulario">
9       <HtmlBody>
10        <cpcit:PageMenuNav runat="server" ID="navbar" />
11        <div>
12          <div id="content-buttons" style="visibility: hidden;">
13            <input id="code" type="text" placeholder="Codigo" />
14            <input id="description" type="text" placeholder="Descrição" />
15            <input id="update_btn" type="button" value="Atualizar" onclick="
refreshGrid()" />
16            <input type="button" value="Novo" onclick="openPageAddValueByItem()" />
17          </div>
18          <div id="content-config">
19          </div>
20        </div>
21
22        <script id="treeview-template" type="text/kendo-ui-template">
23          #: item.description #
24          # if (item.type == 1 && item.itemType == 1) { #
25            <i class="fal fa-edit side-icons" aria-hidden="true"
26              onclick="refreshGridToPresentTreeItemValues('#: item.id#', ' #:
item.itemId#', '111012.02', filters)">
27              </i>
28            # } #
29          </script>
30
31          <script type="text/javascript" src="index.js"></script>
32          <script type="text/javascript"> ... </script>
33
34        </HtmlBody>
35      </cpcit:PageWrapper>
36    </form>
37  </body>

```

Excerto de Código 7.19: Página Dinâmica - HTML Geral

No Excerto 7.19 tem-se a página que é responsável por apresentar a página geral de editar as dimensões. Esta página terá um menu em cima com os diversos *Items* e cada um deles

deve levar para uma página onde listam os *Values* desses items.

```

1 public partial class Index : PageFoundation
2 {
3     protected override void Page_Init(object sender, EventArgs e)
4     {
5         base.Page_Init(sender, e);
6
7         navbar.Items.Add(new PageMenuNavItem()
8         {
9             Text = $"Árvore",
10            JS = $"buildTree()" id="tree-btn"
11        });
12
13        foreach (DimensionItem item in GetItems())
14        {
15            navbar.Items.Add(new PageMenuNavItem()
16            {
17                Text = $"{item.description}",
18                JS = $"createGridFor({item.id}, '{item.description}')"
19            });
20        }
21    }
22
23    internal DimensionItem[] GetItems() { ... }
24 }

```

Excerto de Código 7.20: Página Dinâmica - Geração no Servidor

A geração do menu com os *Items* é feita em *server-side*, como se verifica no Excerto 7.20. Repare-se que a página implementada estende uma implementação da *framework*, a *PageFoundation*, e, além disso, ao ser inicializada preenche a *navbar* declarada no HTML (*cpcit:PageMenuNav*).

A primeira opção do menu será uma página para mostrar as dimensões numa árvore, seguindo-se as opções dos *Items*. Como se pode ver no Excerto 7.20, os vários botões do menu irão apresentar páginas geradas pelo código escrito em *javascript*.

Esse código foi desenvolvido com recurso ao *KendoUI* - uma biblioteca de componentes - que é a tecnologia que a empresa está a utilizar para desenvolver a nova versão da geração de páginas.

7.4 Processos de Negócio

Como já mencionado previamente, a *framework* obriga a adopção de uma estrutura que, além do que foi referido até agora, também guia a especificação da lógica de negócio.

Cada processo de negócio (e.g. submissão de um pedido de recrutamento), quando implementado com a *framework*, está necessariamente associado a uma tabela da base de dados (e.g. *ATS Request*). Ou seja, a *framework* adota uma abordagem *data-centric* em que a cada tabela capta um principal conceito de negócio envolvido no respetivo processo.

Assim, no desenvolvimento destes deve-se considerar também a estrutura de base de dados utilizada, sendo esta uma limitação da DF4A.

7.4.1 Tabelas

A estrutura de directórios seguida para se definirem as funções deve ser semelhante ao apresentado na Figura 7.14.

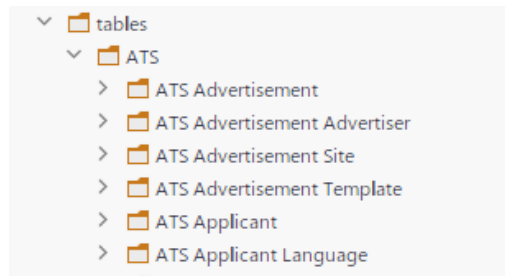


Figura 7.8: Directório de Tabelas

Dentro do directório *tables* são colocados os directórios que correspondem às aplicações desenvolvidas (no exemplo apenas existe *oats*) e, para cada uma delas, devem ser criados os directórios associados às tabelas de cada aplicação.

Para cada tabela é possível configurar os vários eventos e adicionar as funções necessárias.

7.4.2 Eventos de Tabelas

A *framework* implementa as operações CRUD de forma genérica, como já foi referido na secção 2.3.1. Além disso, é possível alterar as acções tomadas aquando da realização dessas mesmas operações para uma ou mais tabelas.

Para tal, recorre-se à definição de eventos que, no presente contexto, se entendem como as seguintes operações:

- **BeforeInsert** - executado antes de uma inserção;
- **AfterInsert** - executado após uma inserção;
- **BeforeUpdate** - executado antes de uma actualização;
- **AfterUpdate** - executado após uma actualização;
- **BeforeDelete** - executado antes de eliminar um registo;
- **AfterDelete** - executado após eliminar um registo;
- **VerifyRecord** - executado após todas as operações CRUD.

Como se verifica na Figura 7.9, é necessário criar um ficheiro *Events.cs* no directório da tabela em questão - neste caso, *Dimension Value*.



Figura 7.9: Directório da Tabela DimensionValue

Para cada evento que se deseja configurar, cria-se uma classe dentro desse ficheiro, com o seu respectivo métodos. No Excerto 7.21, apresentam-se os métodos que devem ser implementados para cada evento.

```

1 public static class BeforeInsert {
2     public static string Run(string Tenant, object[] Params, SqlCommand SQLCommand)
3     { ... }
4 }
5
6 public static class AfterInsert {
7     public static string Run(string Tenant, object[] Params, SqlCommand SQLCommand)
8     { ... }
9 }
10
11 public static class BeforeUpdate {
12     public static string Run(string Tenant, object[] Params, SqlCommand SQLCommand)
13     { ... }
14 }
15
16 public static class AfterUpdate {
17     public static string Run(string Tenant, object[] Params, SqlCommand SQLCommand)
18     { ... }
19 }
20
21 public class BeforeDelete {
22     public static void Run(string Tenant, SqlCommand sqlCommand)
23     { ... }
24 }
25
26 public static class AfterDelete {
27     public static string Run(string Tenant, object[] Params, SqlCommand SQLCommand)
28     { ... }
29 }
30
31 public static class VerifyRecord {
32     public static void Run(string Tenant, Boolean inserting, DataRow row,
33         SqlCommand sqlCommand)
34     { ... }
35 }

```

Excerto de Código 7.21: Métodos dos Eventos

Continuando com o exemplo da tabela identificada na Figura 7.9, apresentam-se, de seguida, dois eventos configurados para a tabela *Dimension Value*.

```

1 public static class BeforeInsert
2 {
3     public static string Run(string tenant, object[] @params, SqlCommand command)
4     {

```

```

5     var parameters = new ParameterDictionary(@params);
6
7     var code = parameters.Get<string>("Code");
8     var itemEntryNo = parameters.Get<int>("Dimension Item Entry No");
9     var sourceEntryNo = parameters.Get<string>("Source Entry No");
10    var parentEntryNo = parameters.Get<int>("Parent Entry No");
11
12    UnitOfWork uow = UnitOfWorkFactory.CreateServiceUnitOfWork(tenant);
13
14    if (CodeIsAlreadyBeingUsed(code, itemEntryNo, sourceEntryNo, uow,
15        command.Transaction))
16        throw new Exception("Código repetido para duas dimensões diferentes");
17
18    if (IsDuplicatedValue(code, itemEntryNo, sourceEntryNo, parentEntryNo, uow,
19        command.Transaction))
20        throw new Exception("A relação que tentou inserir já existe");
21
22    return string.Empty;
23 }
24 }

```

Excerto de Código 7.22: Evento BeforeInsert do DimensionValue

No Excerto 7.22 apresenta-se a implementação do *BeforeInsert*. Inicialmente, usa-se o **ParameterDictionary** para obter as variáveis passadas por parâmetro, seguindo-se a instanciação de uma **UnitOfWork**, que servirá para aceder à base de dados.

Neste evento, é necessário garantir que o código do *Value* não está a ser duplicado. A primeira condição verifica que o *Value* não está a ser utilizado com outro *sourceEntryNo*, enquanto que o segundo verifica que o valor a ser inserido não é um valor duplicado. Caso um deles se verifique, é lançada uma excepção, fazendo *rollback* da transacção.

```

1 public static class BeforeUpdate
2 {
3     public static string Run(string tenant, object[] @params, SqlCommand command)
4     {
5         var parameters = new ParameterDictionary(@params);
6
7         var entryNo = parameters.Get<int>("Original_Entry No");
8         var isToArchive = parameters.GetOrDefault<bool>("Flag Archive");
9         var identity = parameters.GetOrDefault<string>("Identity.Name");
10
11         if (isToArchive)
12             ArchiveChildren(tenant, command.Transaction, entryNo, identity);
13
14         return string.Empty;
15     }
16 }

```

Excerto de Código 7.23: Evento BeforeUpdate do DimensionValue

No excerto 7.23 apresenta-se o *BeforeUpdate*.

Contextualizando, o acto de arquivar um valor é indicativo que se tentou apagá-lo, mas que este já foi usado num *Dimension Set*. Nestas situações, para não haver dados perdidos, arquiva-se esse valor e tenta-se apagar os filhos do mesmo.

Posto isto, se a *flag* estiver activa o código apresentado no Excerto 7.24 é executado.

```

1 private static void ArchiveChildren(string tenant, SqlTransaction transaction,
2   int entryNo, string identity)
3 {
4     UnitOfWork unit = UnitOfWorkFactory
5       .CreateServiceUnitOfWork(tenant);
6
7     var deleteValueService = new DeleteDimensionValueService(unit);
8
9     foreach (DimensionValue child in unit
10      .GetDimensionValueRepository()
11      .GetByParentEntryNo(entryNo, transaction))
12     {
13         deleteValueService.DeleteValue(child, identity, transaction);
14     }
15 }

```

Excerto de Código 7.24: Evento BeforeUpdate do DimensionValue

Como o processo de eliminação de um *Value* não é uma simples operação CRUD, criou-se uma classe responsável pelo devido tratamento dos dados - **DeleteDimensionValueService**.

Deste modo, mais uma vez, recorre-se ao **ParameterDictionary** para obter os valores passados e instancia-se uma **UnitOfWork** para o acesso à base de dados. Com esta última obtêm-se os filhos do *Value* a ser arquivado e delega-se a função de os eliminar ao **DeleteDimensionValueService** previamente instanciado.

No fim da operação, todos os *Value* filhos do que foi actualizado estarão eliminados, caso não estejam associados a um *Set*, ou estarão com *Archive Flag* active.

7.4.3 Funções de Tabelas

Além dos eventos, existe a definição de funções, ou *dynamic codes*, dentro das *tables*. Estas são executadas através de chamadas REST e podem ser directamente configuradas nas páginas da *framework* (e.g. Secção 7.3.2, Excerto 7.16, linha 9).

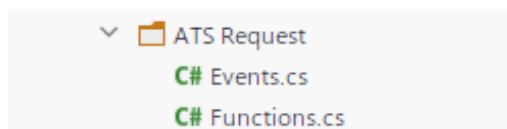


Figura 7.10: Directório da Tabela ATS Request

Normalmente criar-se-ia um ficheiro para cada função. Contudo, como foram aplicadas as sugestões apresentadas na secção 6.4.2, os ficheiros diminuíram consideravelmente o seu tamanho e, deste modo, passou-se a utilizar apenas um ficheiro *Functions.cs* dentro dos directórios de cada tabela (Figura 7.14).

A título de exemplo, no excerto 7.25 apresentam-se os métodos implementados para a tabela **ATS Request**, de modo a que se possa entender como é que estes devem ser definidos.

```

1 namespace tables.ATS.ATS_Request {
2     public static class CancelRequest {
3         public static string Run(object[] @params)
4         { ... }
5     }
6
7     public static class CloneRequest {
8         public static string Run(object[] @params)
9         { ... }
10    }
11
12    public static class GetRequestInfo {
13        public static string Run(object[] @params)
14        { ... }
15    }
16
17    public static class SendForApproval {
18        public static string Run(object[] @params)
19        { ... }
20    }
21 }

```

Excerto de Código 7.25: Estrutura do Function.cs da tabela ATS Request

Mais uma vez, a implementação dos mesmos foi realizada seguindo as sugestões propostas, como se verifica no excerto 7.26 onde se encontra implementação da função **CancelRequest**.

```

1 public static class CancelRequest {
2     public static string Run(object[] @params) {
3         var parameters = new ParameterDictionary(@params);
4
5         var tenant = parameters.Get<string>("Tenant");
6         var companyCode = parameters.Get<string>("CompanyCode");
7         var requestCode = parameters.Get<string>("Code");
8         var identity = parameters.Get<string>("Identity.Name");
9
10        var uow = UnitOfWorkFactory.CreateServiceUnitOfWork(tenant, companyCode);
11
12        var handler = new TransactionHandler(tenant);
13
14        handler.MakeTransaction((transaction) => {
15
16            var requestService = new RequestService(uow);
17            var workflowService = new ApprovalWfService(uow);
18
19            var request = requestService.GetRequest(requestCode, transaction);
20            var hasWorkflowToCancel = !string.IsNullOrEmpty(request.WFRequestEntryNo);
21
22            requestService.CancelRequest(request, identity, transaction);
23
24            if (hasWorkflowToCancel)
25                workflowService.CancelRequestWorkflow(tenant, request, identity);
26
27            transaction.Commit();
28        });
29    }
30 }

```

```
29     return string.Empty;  
30 }  
31 }  
32 }
```

Excerto de Código 7.26: Implementação da função `CancelRequest`

Analisando o excerto apresentado percebe-se que, tal como nos exemplos dos eventos, os métodos começam por criar um **ParameterDictionary** para extrair as devidas variáveis. Após isso, instanciam-se a **UnitOfWork** e, neste caso, o **TransactionHandler** que deve ser usado quando há operações de escrita na base de dados.

O **TransactionHandler** chamará o seu método *MakeTransaction* que recebe a **Action** por parâmetro e a executa logo após ter criado uma transacção.

Neste cenário, a **Action** instancia os dois serviços aos quais esta função recorre para realizar as operações (*RequestService* e *ApprovalWfService*). O pedido é obtido, devidamente cancelado e, caso já se tenha mandado para aprovação, cancela-se o seu fluxo de aprovação.

Por fim, dá-se *commit* da transacção e, não sendo necessário enviar nenhuma informação de resposta, o método retorna uma *string* vazia.

7.5 Fluxos de Trabalho

Para a implementação de fluxos de trabalho também se recorreu à *DF4A*. Nesta secção apresenta-se, inicialmente, o modo como são configurados os fluxos de trabalho e, posteriormente, mostra-se quais as implicações a nível da implementação.

7.5.1 Configuração dos Fluxos

Antes de criar um fluxo de trabalho é necessário criar todos os perfis de aprovação que farão parte dos mesmos. Esta funcionalidade já vem desenvolvida de raiz na *framework* interna, sendo apenas necessário configurar o sistema para apresentar a página que permite fazer esta configuração.

Para criar um perfil é necessário indicar um código e uma descrição apenas. Como se verifica na Figura 7.11, estão criados todos os perfis indicados pelo cliente.

De seguida, devem-se criar os vários fluxos de trabalho que se desejam implementar, sendo que para cada se deve indicar a descrição e a tabela sobre o qual ele vai actuar. Por exemplo, como se vê na Figura 7.12, os fluxos de aprovação deverão referenciar a tabela de pedidos de recrutamento (*ATS Table*).

Também se pode verificar na Figura 7.12 que é possível configurar os passos do fluxo. Para criar um passo deve-se indicar o *step* - a ordem pela qual vão ser executados -, uma descrição, qual o perfil ou o utilizador que irá actuar no passo, quais as acções disponíveis nesse mesmo passo (e.g aprovar, cancelar), e quais as páginas a usar para mostrar a informação ao utilizador.

Perfil Workflow			
Atualizar Novo Exportar			
	Code	Description	...
	ADMIN_A	Adminstrador de Área	
	ADMIN_H	Adminstrador Hospitalar	
	ADMIN_N	Adminstrador de Negócio	
	AREA_M	Área Manager	
	EQ_F	Equipa de Fornecedores	
	PCA	PCA	
	R_M	Retail Manager	
	RH_C	RH Central	
	RH_U	RH Unidade	
	RP_S	Responsável de Serviço	
	S_M	Store Manager	

Figura 7.11: Lista de Perfis dos Fluxos de Trabalho

PROCESSO WORKFLOW

Processo workflow

Editar

Description

Fluxo de Aprovação

Short Description

Fluxo de Aprovação

Datatable

ATS Request

☐ Validação processo em massa

Código Report

URL Insert

Circuito de aprovação

Novo

	Step	Descrição	...
	1	Tomar Conhecimento de Pedido	
	2	Validar Pedido	
	3	Tomar Conhecimento de Pedido	
	4	Validar Pedido	
	5	Validar Pedido	

Figura 7.12: Configuração de um Fluxo de Trabalho

Para o desenvolvimento inicial, foram identificados os utilizadores que iriam testar inicialmente o sistema e os fluxos foram configurados de acordo com esses utilizadores.

Processo workflow			
Actualizar Novo Exportar			
Description	Datatable	...	
Fluxo de Aprovação de Mobilidades	ATS Mobility Request		
Fluxo Aprovações Unidades Hospitalares	ATS Recruitment Request		
Fluxo de Aprovação Prestadores	ATS Recruitment Request		
Fluxo de Aprovações ACE	ATS Recruitment Request		
Fluxo de Aprovações VNC, VNCl e Ciberbit	ATS Recruitment Request		
Fluxo de Aprovações Cofemel e Vilanova – Sede	ATS Recruitment Request		
Fluxo Aprovações Cofemel, Cofemel Espanha, Cofemel Luxemburgo, Vilanova e Retail – Equipa de Gestão	ATS Recruitment Request		
Fluxo de Aprovações – Cofemel, Cofemel Espanha, Cofemel Luxemburgo, Vilanova e Retail – Staff	ATS Recruitment Request		
Fluxo de Aprovações – Planicare	ATS Recruitment Request		

Figura 7.13: Lista dos Fluxos de Trabalho Configurados

No fim da configuração ficou-se com os fluxos identificados pelo cliente, como mostra a Figura 7.13.

7.5.2 Funções dos Fluxos

Para iniciar os fluxo de trabalho da *framework*, colocaram-se botões na *navbar* da página que apresenta os pedidos de recrutamento que chama a função *SendRequestForApproval*.

```

1 private static void StartWorkflow(string tenant, string companyCode,
2     string requestCode, int wfEntryNo,
3     string identity) {
4
5     var wfParams = new Hashtable {
6         ["Code"] = requestCode,
7         ["Description"] = "Pedido recrutamento",
8         ["Owner Username"] = identity,
9         ["Identity.Name"] = identity,
10        ["External Code"] = $"ATS Request:{requestCode}",
11        ["WF Request Type Entry No"] = wfEntryNo
12    };
13
14    var wfParamsObj = PageHelper.HashtableToObject(wfParams);
15    CoreWorkflow.StartWorkflowByEntryNo(tenant, companyCode, wfEntryNo, wfParamsObj);
16 }

```

Excerto de Código 7.27: Funções Fluxo Trabalho

Após validar que os dados do pedido estão devidamente preenchidos, e de identificar qual dos fluxos deve ser seguido, corre o método apresentado no Excerto 7.27.

Com os dados devidamente preenchidos, a framework identifica sobre qual tabela o fluxo de trabalho funcionará. Sabendo qual é, a DF4A vai ao directório com o nome da tabela que se encontra dentro do directório *workflow* dentro do directório gertal das *tables* (Figura 7.14).

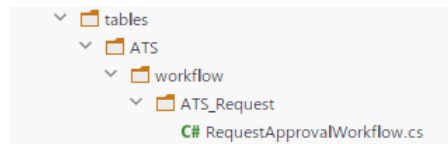


Figura 7.14: Configuração Fluxos de Trabalho - Directórios

O nome dado ao ficheiro é indiferente, apenas deve ter declaradas as classes apresentadas no Excerto 7.28.

```
1 public static class BeforeStartWorkflow {
2     public static object[] Run(string tenant, string companyCode,
3                               int requestTypeEntryNo, Hashtable @params,
4                               SqlTransaction transaction) {
5         ...
6     }
7 }
8
9 public static class Action {
10     public static void Run(RequestTaskObject requestTaskObject) {
11         ...
12     }
13 }
14
15 public static class AbortProcess {
16     public static void Run(RequestTaskObject requestTaskObject) {
17         ...
18     }
19 }
20
21 public static class EndProcess {
22     public static void Run(RequestTaskObject requestTaskObject) {
23         ...
24     }
25 }
```

Excerto de Código 7.28: Funções Fluxo Trabalho

Seguindo a sequência deste último excerto, ao se iniciar o fluxo de trabalho a primeira classe é chamada. A realização de cada passo do fluxo resulta numa chamada à segunda classe. Por fim, se o fluxo for cancelado executa-se a terceira, se chegar ao fim e tiver sido aprovado em todos os passos, executar-se-á a última classe.

No caso dos pedidos de aprovação, como se vê no Excerto 7.29, quando se dá o início do fluxo, o sistema procura quais os validadores que deverão ser notificados da criação do fluxo.

```

1 var parameters = new ParameterDictionary(@params);
2
3 var code = parameters.Get<string>("Code");
4 var identity = parameters.Get<string>("Identity.Name");
5 var wfReqEntryNo = parameters.Get<string>("IWF Request Entry No");
6
7 var dimensionEntryNoRequest = GetRequest(tenant, companyCode, transaction, code);
8 var dtWfRequestTask = GetWorkflowRequestTasks(tenant, transaction, wfReqEntryNo);
9
10 foreach (DataRow drReqTask in dtWfRequestTask.Rows) {
11     var validators = GetPossibleValidators(tenant, transaction,
12         dimensionEntryNoRequest, drReqTask);
13
14     if (validators.Rows.Count <= 0)
15         throw new Exception("Não existem validadores");
16
17     var usersToNotifyBuilder = new StringBuilder();
18
19     for (var i = 0; i < validators.Rows.Count; i++) {
20         if (usersToNotifyBuilder.ToString()
21             .Contains(validators.Rows[i]["Username"].ToString()))
22             continue;
23
24         var prefix = i == 0 ? "" : ",";
25         usersToNotifyBuilder.Append($"{prefix}{validators.Rows[i]["Username"]}");
26     }
27
28     var usersToNotify = usersToNotifyBuilder.ToString();
29     UpdateWfRequestTask(tenant, transaction, usersToNotify, drReqTask, identity);
30 }
31 return new object[0];

```

Excerto de Código 7.29: Funções Fluxo Trabalho - BeforeStartWorkflow

No Excerto 7.30 pode-se ver quais os tipos de *action* podem ser realizados e qual o seu respectivo código. É nesta classe que se codificam as validações e todos os efeitos secundários que se deseja ver no sistema, aquando da realização de determinada acção.

```

1 public static class Action {
2     public static void Run(RequestTaskObject requestTaskObject) {
3         switch (requestTaskObject.Action) {
4             case "V": ValidateAction(requestTaskObject); break; // + Aprovado
5             case "R": AbortAction(requestTaskObject); break; // + Rejeitado
6
7             case "D": throw new NotSupportedException(); // + Retorna ao Início
8             case "A": throw new NotSupportedException(); // + Cancelado
9             case "F": throw new NotSupportedException(); // + Encaminhado
10
11             default: throw new NotSupportedException("Invalid action.");
12         }
13     }
14 }

```

Excerto de Código 7.30: Funções Fluxo Trabalho - Action

No exemplo mostrado, sempre que há uma aprovação - isto é, aprovam, o pedido e este passe para o próximo passo - o pedido passa por um processo de validação, garantindo-se que não houve nenhuma alteração que pudesse levar a mais à frente no fluxo.

Se, por algum motivo, algum dos utilizadores rejeitar o pedido, o fluxo de trabalho é terminado e o Excerto 7.31 executado. Neste caso, o pedido deve ser actualizado para o estado do mesmo esatr em conformidade com a rejeição.

```

1 public static class AbortProcess {
2     public static void Run(RequestTaskObject requestTaskObject) {
3         Hashtable parameters = PageHelper.ParamsToHashtable(requestTaskObject.
4             ParamsFromUI);
5
6         var tenant = requestTaskObject.Tenant;
7         var companyCode = requestTaskObject.CompanyCode;
8
9         var identity = parameters["Identity.Name"].ToString();
10        var code = requestTaskObject.ExternalCode.Split(':')[1];
11
12        var uow = new UnitOfWork(FrameworkRepositoryType.Services, requestTaskObject.
13            Tenant, companyCode);
14
15        var requestService = new RequestService(uow);
16
17        Request request = requestService.GetRequest(code, requestTaskObject.Transaction
18            );
19
20        if (request.Status != RequestStatus.Rejeitado) {
21            requestService.CancelRequest(request, identity, requestTaskObject.
22                Transaction);
23        }
24    }
25 }

```

Excerto de Código 7.31: Funções Fluxo Trabalho - EndProcess

Por fim, se o pedido tiver sido aceite por todos os intervenientes do fluxo de aprovação, deve-se, como apresentado no Excerto 7.32, iniciar o processo de aprovação.

```

1 public static class EndProcess {
2     public static void Run(RequestTaskObject requestTaskObject) {
3         var parameter = new ParameterDictionary(requestTaskObject.ParamsFromUI);
4
5         var identity = parameter.Get<string>("Identity.Name");
6         var cCode = requestTaskObject.CompanyCode;
7         var code = requestTaskObject.ExternalCode.Split(':')[1];
8
9         UnitOfWork uow = UnitOfWorkFactory
10             .CreateServiceUnitOfWork(requestTaskObject.Tenant, cCode);
11
12        new ApprovalWfOnEndService(uow, identity, requestTaskObject.Transaction)
13            .CreateProcessFromAcceptedRequest(code);
14    }
15 }

```

Excerto de Código 7.32: Funções Fluxo Trabalho - EndProcess

Capítulo 8

Avaliação e Experimentação

No presente capítulo apresentam-se os factores de avaliação definidos para validar a solução final e descrevem-se as abordagens adotadas para os abordar.

8.1 Satisfação do Utilizador

O primeiro factor é relativo à satisfação do utilizador e, na presente secção, encontram-se os dados recolhidos e analisados para validar o mesmo.

8.1.1 Objetivo

O principal objetivo passa por verificar a satisfação global dos utilizadores que vão utilizar a solução desenvolvida.

8.1.2 Procedimento

Para responder ao objetivo descrito, elaborou-se um inquérito que foi fornecido a alguns utilizadores do sistema de modo a que seja possível saber qual a sua opinião do novo processo implementado.

Para responder ao inquérito, seleccionaram-se todos os colaboradores que tiveram presentes nas reuniões de definição do produto, nas demonstrações do mesmo e que, de um modo limitado, já utilizaram, durante um mês, a solução desenvolvida. Isto inclui 1 administrador, 4 colaboradores dos recursos humanos com diferentes responsabilidades e 2 técnicos de informática do cliente.

Tendo em conta que os grupos de entrevistados terão diferentes funções na empresa, podendo alguns não ter conhecimentos técnicos, é de uma maior relevância garantir que as perguntas sejam generalizadas, mas sem comprometer a fiabilidade do inquérito.

Um mês após a implantação da primeira versão do sistema no ambiente de qualidade fornecido pelo cliente, realizou-se o inquérito. Este consiste em 7 perguntas, avaliadas numa escala de 0 a 5 (sendo 5 o valor desejável) para que seja possível analisar quantitativamente a satisfação geral.

No sentido de testar a satisfação inclui-se a hipótese das pessoas envolvidas no processo estarem mais satisfeitas com o novo sistema. A hipótese nula corresponde à pontuação média aceitável para cada questão do formulário.

Este valor foi discutido com o supervisor do projecto e pelos gestores responsáveis. Definiu-se que, para cada questão pode haver, no máximo, 25% de respostas abaixo de 3. Sendo que o inquérito foi respondido por 9 indivíduos, não deverá haver nenhuma pergunta com 2 ou mais respostas abaixo de 4.

Contudo, uma excepção foi aberta relativamente à segunda questão. Isto porque foi definido com o cliente que o design da aplicação será melhorado significativamente quando o sistema estiver utilizável para gerir os processos. Assim, apesar de se recolher a informação pois é um dado relevante, deve-se contar que a avaliação da mesma estará abaixo daquilo que se espera.

Como já foi mencionado, as respostas serão numa escala de 0 a 5 e comparar-se-á a média de cada pergunta à pontuação média determinada. A comparação permite concluir se as pessoas envolvidas no processo estão satisfeitas com o novo produto.

8.1.3 Resultados

Na Tabela 8.1 podem-se ver as perguntas associadas ao número de respostas que obtiveram na escala definida (0 a 5).

Tabela 8.1: Tabela dos Resultados do Inquérito

Pergunta	1	2	3	4	5
A utilização do sistema como um todo é intuitiva.	0	0	0	7	2
O design aplicado ao sistema é o indicado.	0	0	5	4	0
A navegação entre páginas e a realização das operações são intuitivos.	0	0	1	6	2
A solução aparenta ser estável (poucas falhas a nível de bugs).	0	0	0	5	4
O tempo de resposta da aplicação é o indicado.	0	0	1	3	5
Qual a sua avaliação da solução inicial de 0 a 5?	0	0	1	6	2
Considera que a aquisição do sistema será uma mais valia?	0	0	2	3	4

8.1.4 Análise dos Resultados

Analisando as repostas recolhidas pelo questionário desenvolvido, é possível perceber o nível de satisfação dos utilizadores e compreender quais os pontos que precisam de melhorias. No Anexo C é possível ver o questionário e a distribuição das respostas de cada questão do mesmo.

Com os resultados tem-se:

- Apenas a questão relativa ao *design* visual não correspondeu à expectativa da empresa, o que era esperado.
- Apesar do descontentamento geral com o aspecto da aplicação, a solução mostra-se intuitiva;
- A questão relativa aos tempos de resposta é a com uma pontuação mais alta;
- Nenhum utilizador considerou que não fosse uma mais valia a aquisição da solução, somente 2 mostraram-se relativamente neutros ao assunto.

Verificando os resultados, depreende-se que os objectivos gerais relativos à satisfação e usabilidade do sistema foram alcançados. Apesar de poder haver melhorias na aparência da solução, esta mostrou-se intuitiva, rápida e, de um modo geral, uma melhoria e uma mais valia para a empresa.

8.2 Fiabilidade do Sistema

O segundo factor é relativo à fiabilidade do sistema e, na presente secção, encontram-se os dados recolhidos e analisados para validar o mesmo.

8.2.1 Objetivos

Para assegurar a fiabilidade do sistema, definiram-se métricas que devem ser cumpridas:

- **Métrica 1** - O lado do servidor deve garantir alta disponibilidade, seguindo um SLA de *two nines* (99%);
- **Métrica 2** - Tempo médio de resposta dos *endpoints* deve ser inferior a 1 segundo;
- **Métrica 3** - Tempo médio do carregamento das páginas da interface do utilizador deve ser inferior a 5 segundos.
- **Métrica 4** - Deve poder processar até 10 pedidos por segundo nos *endpoints* dos serviços;

A hipótese nula será o cumprimento dos valores previamente definidos.

8.2.2 Setup

O sistema foi implantado num único servidor (*windows server*) e a base de dados encontra-se num *SQL Server*, ambos fornecidos e mantidos pelo cliente. Este é o ambiente de qualidade, que está a servir para os utilizadores finais testarem o sistema antes de o implantar em ambiente de produção.

Os dados apresentados foram recolhidos durante o primeiro mês após da implantação no servidor de qualidade.

8.2.3 Resultados

Métrica 1

De forma a responder à métrica, utilizou-se o Uptime Robot, que disponibiliza o serviço de monitorização de serviços, recolhendo os dados de *downtime* e *uptime* dos mesmos.

Relativamente aos dados recolhidos pela ferramenta, o *uptime* durante o mês que se monitorizou foi de 99.533%, estando dentro daquilo que se espera de um SLA de *two nines*.

Métrica 2 e 3

No que respeita a métrica 2, seleccionaram-se os três serviços que requerem mais processamento: criar uma dimensão aquando da configuração da estrutura organizacional, realizar um pedido de recrutamento e enviar um pedido para aprovação.

Para conseguir fazer este tipo de análise, foi usado o serviço Statful, que é uma ferramenta que agrega os dados e permite melhor analisá-los. Os serviços foram sendo chamados ao longo do mês, numa totalidade de 20 chamadas para cada um. O tempo de resposta manteve-se entre os 500 e os 800 milissegundos, nunca excedendo o limite máximo imposto.

No que respeita a métrica 3, seleccionou-se a página de edição completa do pedido e a de configuração da estrutura empresarial, visto serem as que contêm mais componentes.

Recorrendo à mesma ferramenta que na métrica 2, pôde-se observar que o carregamento das páginas cumpre o requisito proposto. No entanto, os valores vão oscilando entre os 700 e os 1200 milissegundos, sendo o carregamento das páginas um pouco instável.

Métrica 4

Respondendo à métrica 4, realizou-se um teste recorrendo a *multithreads* que enviaram, simultaneamente, pedidos para o *endpoint* dos serviços responsável pela criação de um pedido.

Ao longo do mês após a implantação testou-se esta métrica múltiplas vezes. Sendo o requisito de 10 pedidos, e correspondendo este ao limite imposto pela versão grátis do Statful, foram realizados 10 pedidos em simultâneo.

O sistema mostrou-se capaz de responder aos mesmos em todos as execuções do teste.

8.2.4 Análise dos Resultados

O sistema encontra-se dentro de todas as métricas definidas e, por isso mesmo, pode-se dizer que é fiável.

No entanto, existem algumas limitações na experimentação que se fez. A título de exemplo, para as métricas 2 e 3 seleccionaram-se os serviços e as páginas que requeriam maior

processamento. No entanto, não se avaliaram individualmente todos os serviços disponibilizados para se saber quais eram, efectivamente, os mais demorados.

Capítulo 9

Conclusão

O presente capítulo tem como conteúdo as conclusões do projecto, incluindo os objetivos alcançados, limitações e trabalho futuro e apreciação final.

9.1 Objectivos Alcançados

Relativamente aos objetivos definidos para o presente trabalho, todos foram atingidos.

Efetivamente, houve um estudo intenso sobre o domínio de RH e de tudo o que ele compreende. A análise da aplicação legada junto com a investigação das possíveis abordagens e tecnologias que se poderiam utilizar para o desenvolvimento de uma nova solução foi bem sucedida.

No que respeita os objetivos relacionados com a solução final em concreto, tem-se que o levantamento de requisitos foi devidamente realizado e que a construção proposta mostrou-se capaz de responder às necessidades do cliente.

Nas Tabelas 9.1 e 9.2 pode-se ver quais o estado dos vários processos da solução.

Tabela 9.1: Tabela do Estado Final dos Requisitos - Parte 1

Requisito	Resultado
PDR01	✓
PDR02	✓
PDR03	✓
PDR04	✓
PDR05	✓
GC01	✓
GC02	✓
GC03	✓
GC04	x
GC05	✓

Tabela 9.2: Tabela do Estado Final dos Requisitos - Parte 2

Requisito	Resultado
AP01	✓
AP02	✓
AP03	✓
AP04	x
AP05	x
AP06	✓
AP07	✓
AP08	✓
PCR01	✓
PCR02	✓
PCR03	✓
PCR04	✓
PCR05	x
PCR06	✓
PCR07	x
GE01	✓
GE02	✓
GE03	✓
PAR01	✓
PAR02	x
PAR03	x
PAR04	x
C01	x
C02	x
C03	x
C04	x
C05	x

Tal como se verifica em ambas as tabelas, a maioria dos requisitos elicitados foram realizados. Sendo que, os que ficaram por implementar não são impedimento para que o sistema seja disponibilizado ao cliente.

9.2 Limitações e Trabalho Futuro

A obrigatoriedade da utilização da *framework* interna pode ser vista como uma limitação apenas no início do desenvolvimento. Após se estar familiarizado com a mesma, todo o processo de desenvolvimento acelera.

No entanto, foi preciso actualizar alguns dos mecanismos da *framework* e, como esse desenvolvimento é feito apenas por dois membros da empresa, houve mais uma situação

de bloqueio do desenvolvimento.

A primeira grande limitação do trabalho desenvolvido, no que respeita a implementação da solução, foi a dimensão e a organização da equipa. Tendo em conta a dimensão do projecto, considera-se que o número de elementos a trabalhar no projecto era reduzido e que afectou negativamente o desenvolvimento. Mais, apesar de se aplicar uma metodologia ágil aquando do desenvolvimento da solução, acabou por existir pouca organização entre os membros da equipa e a distribuição de trabalho sofreu negativamente.

Além disso, apesar de a ferramenta usada pela empresa permitir o desenvolvimento em *branches*, não é uma prática aplicável na empresa. Isso implicou esperas alongadas para implementação de alguns serviços. Adicionando a inexistência de mecanismos de CI/CD, tem-se um impacto ainda maior no tempo necessário para o desenvolvimento.

Relativamente ao trabalho futuro, tem-se a implementação dos casos de uso que ficaram por completar. Mais, considerando o que foi analisado da DF4A, seria relevante para a empresa investigar a possibilidade de desenvolver a sua *framework* recorrendo a técnicas de engenharia de domínio. Isto porque, uma grande parte da mesma é a leitura dos ficheiros e, com recurso a essas ferramentas, aplicando essas metodologias seria mais fácil para a empresa desenvolver, manter e actualizar os ficheiros de configuração que usam.

Ademais, e relacionado com esta última proposta, seria interessante pôr a implementação dos repositórios a ser feita automaticamente, partindo dos ficheiros de configuração que se fazem para a base de dados.

9.3 Apreciação Final

De um modo geral, o desenvolvimento deste projecto fica com um balanço positivo, pois não só permitiu aprofundar os conhecimentos num domínio completamente desconhecido (Recursos Humanos), como permitiu fazer o levantamento de requisitos directamente com o cliente.

Apesar de não ter sido concluído na sua totalidade no tempo estimado, a solução construída ficou pronta a ser testada pelos utilizadores finais.

Bibliografia

- Armstrong, M. (2006). *A Handbook of Human Resource Managment Practice*. Kogan Page.
- Barone, A. (2020). *Bottleneck*. <https://www.investopedia.com/terms/b/bottleneck.asp>
- Bell, B. S., Lee, S.-W. & Yeung, S. K. (2006). The impact of e-HR on professional competence in HRM: Implications for the development of HR professionals. *Human Resource Management*, 45(3), 295–308. <https://doi.org/https://doi.org/10.1002/hrm.20113>
- Brown, S. (2021). *The C4 model for visualising software architecture*. <https://c4model.com/>
- Camunda. (2020). *How CMMN never lived up to its potential*. <https://camunda.com/blog/2020/08/how-cmmn-never-lived-up-to-its-potential/>
- Camunda. (2021a). *BPMN 2.0 Symbol Reference*. <https://camunda.com/bpmn/examples/>
- Camunda. (2021b). *Deliver Workflow Automation To Your Customers*. <https://camunda.com/solutions/add-workflow-software/>
- Carlos Eduardo Vazquez, G. S. S. (2016). *Engenharia de Requisitos: Software Orientado ao Negócio* (1ª ed.). BRASPORT.
- Dumas, M., La Rosa, M., Mendling, J. & Reijers, H. A. (2013). *Fundamentals of business process management* (2013ª ed.). Springer.
- Georgakopoulos, D., Hornick, M. & Sheth, A. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3(2), 119–153. <https://doi.org/10.1007/BF01277643>
- Golden, B. L., Wasil, E. A., Harker, P. T., Golden, P. B. L., Wasil, A. P. E. A. & Harker, A. P. P. T. (1989). *The Analytic Hierarchy Process: Applications and Studies* (1ª ed.). Springer-Verlag Berlin Heidelberg.
- Hat, R. (2021). *What is middleware?* <https://www.redhat.com/en/topics/middleware/what-is-middleware>
- Indulska, M., Green, P., Recker, J. & Rosemann, M. (2009). Business Process Modeling: Perceived Benefits. Em A. H. F. Laender, S. Castano, U. Dayal, F. Casati & J. P. M. de Oliveira (Eds.), *Conceptual Modeling - ER 2009* (pp. 458–471). Springer Berlin Heidelberg.
- Koen, P., Ajamian, G., Boyce, S., Clamen, A., Fisher, E. S., Fountoulakis, S., Johnson, A., Puri, P. & Seibert, R. (2002). 1 Fuzzy Front End : Effective Methods , Tools , and Techniques. https://www.academia.edu/4878240/1_Fuzzy_Front_End_Effective_Methods_Tools_and_Techniques

- Koen, P., Ajamian, G., Burkart, R., Clamen, A., Davidson, J., D'Amore, R., Elkins, C., Herald, K., Incorvia, M., Johnson, A., Karol, R., Seibert, R., Slavejko, A. & Wagner, K. (2001). Providing Clarity and A Common Language to the "Fuzzy Front End". *Research-Technology Management*, 44(2), 46–55. <https://doi.org/10.1080/08956308.2001.11671418>
- Kruchten, P. B. (1995). The 4+1 View Model of architecture. *IEEE Software*, 12(6), 42–50. <https://doi.org/10.1109/52.469759>
- Kulkarni, S. B. & Che, X. (2019). Intelligent software tools for recruiting. *Journal of International Technology and Information Management*, 28(2), 2–16.
- Larman, C. (2005). *Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design and Iterative Development*. Prentice Hall PTR.
- Lengnick-Hall, M. L. & Moritz, S. (2003). The impact of e-HR on the human resource management function. *Journal of Labor Research*, 24(3), 365–379. <https://doi.org/10.1007/s12122-003-1001-6>
- Lina, M. (2019). Human Resource Information System (HRIS): An Important Element of Modern Organization. *Global Disclosure of Economics and Business*, 8, 61–66. <https://doi.org/10.18034/gdeb.v8i2.98>
- Manatal. (2021). *AI Recruitment Software*. <https://www.manatal.com/>
- Marques, L. P. R. (2018). The benefits of e-recruitment: Web 1.0 & Web 2.0.
- Morais, R., Kazan, S., Pádua, S. & Costa, A. (2014). An analysis of BPM lifecycles: From a literature review to a framework proposal. *Business Process Management Journal*, 20. <https://doi.org/10.1108/BPMJ-03-2013-0035>
- Mukherjee, A. N., Bhattacharyya, S. & Bera, R. (2014). Role of Information Technology in Human Resource Management of SME: A Study on the Use of Applicant Tracking System. *IBMRD's Journal of Management & Research*, 3(1). <http://www.ibmrjournal.com/index.php/ibmrd/article/view/46706>
- Neap, H. S. & Celik, T. (1999). Value of a Product: A Definition. *International Journal of Value-Based Management*, 12(2), 181–191. <https://doi.org/10.1023/A:1007718715162>
- Nguyen, G. H. (2014). The Analytic Hierarchy Process: A Mathematical Model for Decision Making Problems. <https://openworks.wooster.edu/independentstudy/6054/>
- Nick, R. & Holweg, M. (2000). *Value Analysis Value Engineering* (rel. téc.). http://www.ivma.org.au/wp-content/uploads/2019/05/14.-Value_Analysis_Value_Engineering_-_Rich_and_Holweg_2000.pdf
- Nicola, S. (2020). Método de Análise Hierárquica (Aula).
- NPD. (2016). *Value Analysis and Function Analysis System Technique*. <https://www.npd-solutions.com/va.html>
- OMG. (2011a). *Business Process Model and Notation (BPMN)* (rel. téc.). <https://www.omg.org/spec/BPMN/2.0/PDF>
- OMG. (2011b). *OMG Unified Modeling Language™ (OMG UML), Superstructure* (rel. téc.). <https://www.omg.org/spec/UML/2.4.1/Superstructure/PDF>
- OMG. (2016). *Case Management Model and Notation (CMMN)* (rel. téc.). <https://www.omg.org/spec/CMMN/1.1/PDF>
- OMG. (2019). *Decision Model and Notation (DMN)* (rel. téc.). <https://www.omg.org/spec/DMN/1.3/PDF>

- Osterwalder, A. & Pigneur, Y. (2003). Modeling value propositions in e-Business. *ACM International Conference Proceeding Series*, 50, 429–436. <https://doi.org/10.1145/948005.948061>
- Osterwalder, A. & Pigneur, Y. (2010). *Business model generation: A handbook for visionaries, game changers, and challengers* (1ª ed.). Wiley.
- Passos, N. & Pereira, J. L. (2018). Business Process Modeling: how CMMN and DMN complement BPMN.
- Peixoto, D., A. Batista, V., Atayde, A., Borges, E., Resende, R., Isaías, C. & Pádua, P. (2008). A Comparison of BPMN and UML 2.0 Activity Diagrams.
- Reijers, H. A. (2003). *Design and control of workflow processes: Business process management for the service industry*. Springer.
- Robert L. Mathis, J. H. J. (1999). *Human Resource Management*, 9th edition (9th).
- Stohr, E. A. & Zhao, J. L. (2001). Workflow Automation: Overview and Research Issues. *Information Systems Frontiers*, 3(3), 281–296. <https://doi.org/10.1023/A:1011457324641>
- Teamtaylor. (2021). *The recruitment & employer branding ATS*. <https://www.teamtaylor.com/en/>
- Traffit. (2021). *What is Traffit and what is it used for?* <https://knowledge.traffit.com/en/articles/3365338-what-is-traffit-and-what-is-it-used-for>
- Van der Aalst, W. M. P. (2004). Business Process Management Demystified: A Tutorial on Models, Systems and Standards for Workflow Management. Em J. Desel, W. Reisig & G. Rozenberg (Eds.), *Lectures on Concurrency and Petri Nets: Advances in Petri Nets* (pp. 1–65). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-27755-2_1
- Visual Paradigm. (2021a). *What is Activity Diagram?* <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>
- Visual Paradigm. (2021b). *What is Case Management Model and Notation (CMMN)*. <https://www.visual-paradigm.com/guide/cmmn/what-is-cmmn/>
- Woodall, T. (2003). Conceptualising 'Value for the Customer': An Attributional, Structural and Dispositional Analysis. *Academy of Marketing Science Review*, 12.
- WSO2. (2021a). *Business Process Server*. <https://wso2.com/products/business-process-server/>
- WSO2. (2021b). *Introducing WSO2 Business Process Server*. <https://docs.wso2.com/display/BPS360/Introducing+WSO2+Business+Process+Server>

Apêndice A

Método AHP

A.1 Escala Fundamental de Saaty

Tabela A.1: Escala Fundamental de Saaty [Fonte: (Nicola, 2020)]

Nível Importância	Definição	Explicação
1	Igual Importância	As duas atividades contribuem igualmente para o objetivo.
3	Fraca Importância	A experiência e o julgamento favorecem levemente uma atividade em relação à outra.
5	Forte Importância	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra.
7	Muito Forte Importância	Uma atividade é muito fortemente favorecida em relação à outra.
9	Importância Absoluta	A evidência favorece uma atividade em relação a outra com mais alto grau de certeza.
2, 4, 6, 8	Valores Intermediários	Quando se procura uma condição de compromisso entre duas definições.

A.2 Valores de IR Tabelados

Tabela A.2: Valores de IR para Matrizes Quadradas de Ordem n [Adaptado: (Nicola, 2020)]

1	2	3	4	5	6	7	8	9	10
0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.51

A.3 Prioridades Relativas Normalizadas

Por uma questão de legibilidade, substitui-se:

- C - Custos Associados;
- T - Tempo de Desenvolvimento;
- F - Facilidade de Implementação;

Tabela A.3: Matriz de Comparação dos Critérios Normalizada

	C	T	F
C	12/19	9/13	1/2
T	4/19	3/13	3/8
F	3/19	1/13	1/8

A.4 Consistência das Prioridades Relativas

O vetor de prioridades (VP) é calculado fazendo a média de cada linha da matriz das prioridades relativas normalizada (MN) (Anexo A.3). Assim, após os cálculos:

$$MN = \begin{bmatrix} 12/19 & 9/13 & 1/2 \\ 4/19 & 3/13 & 3/8 \\ 3/19 & 1/13 & 1/8 \end{bmatrix} \quad VP = \begin{bmatrix} 0.608 \\ 0.272 \\ 0.120 \end{bmatrix} \quad (A.1)$$

Tendo o VP, é necessário avaliar a consistência das prioridades relativas. Para tal, deve-se calcular o rácio de consistência (RC), cujo valor deverá ficar abaixo de 0.1 para que se possa confiar nos valores do método AHP. Posto isto, é necessário calcular:

$$M * VP = \lambda_{max} * VP \Leftrightarrow \begin{bmatrix} 0.63231 \\ 0.23577 \\ 0.13192 \end{bmatrix} = \lambda_{max} * \begin{bmatrix} 0.608 \\ 0.272 \\ 0.120 \end{bmatrix} \quad (A.2)$$

$$\Leftrightarrow \lambda_{max} = \left(\frac{0.63231}{0.608} + \frac{0.23577}{0.272} + \frac{0.13192}{0.120} \right) / 3 = 3,0061 \quad (A.3)$$

Para o cálculo do RC:

$$RC = \frac{IC}{IR} \quad IC = \frac{\lambda_{max} - 3}{3 - 1} = 0,00305 \quad (A.4)$$

Como a matriz é uma matriz quadrada de ordem 3, temos que $IR = 0,58$ (Anexo A.2).

$$RC = \frac{0,00305}{0,58} \simeq 0,0053 \quad (A.5)$$

A.5 Matrizes de Comparação Paritária Normalizadas

Tabela A.4: Matriz de Comparação de Custos Normalizada

	Atualizar Solução	Implementar Nova Solução
Atualizar Solução	1/4	3/4
Implementar Nova Solução	3/4	3/4

Tabela A.5: Matriz de Comparação do Tempo de Implementação Normalizada

	Atualizar Solução	Implementar Nova Solução
Atualizar Solução	1/4	3/4
Implementar Nova Solução	3/4	3/4

Tabela A.6: Matriz de Comparação da Facilidade de Implementação Normalizada

	Atualizar Solução	Implementar Nova Solução
Atualizar Solução	1/4	3/4
Implementar Nova Solução	3/4	3/4

Apêndice B

Requisitos Levantados

O texto presente nas seguinte secção corresponde a excertos dos documentos desenvolvidos para a empresa ao longo do projecto. O texto apresentado é uma cópia exacta do que está nos documentos da empresa.

Note-se que nem todos os casos de uso estão detalhados e que as nomenclaturas utilizadas podem não estar em conformidade com os nomes utilizados no relatório. Isto porque, não havendo um glossário para a empresa, muitos dos termos acabam por ter vários nomes.

Contudo, para o relatório, procedeu-se à uniformização dos nomes.

B.1 Descrição dos Requisitos

A descrição apresentada foi

Pedidos de Recrutamento

PDR01 - Criar pedido de recrutamento

Como *Técnico* ou *Master*, quero poder criar um pedido de recrutamento, para mais tarde o enviar para aprovação.

Sequência de acções:

1. Criação de raiz;
 - (a) Clicar no botão de “Novo Pedido”;
 - (b) Preencher os dados necessários;
 - (c) Guardar os dados;
 - (d) É reencaminhado para a página de edição de pedido.
2. Duplicação de um pedido já existente;
 - (a) Listar os pedidos;

- (b) Seleccionar o pedido pretendido;
- (c) Clicar no botão de “Duplicar”;
- (d) É reencaminhado para a página de edição de pedido.

Para se poder registar um pedido, os seguintes campos são obrigatórios:

- Área de Negócio
- Empresa
- Departamento
- Função
- Local de Trabalho
- Data de Início de Funções Pretendida (por defeito, 15 dias depois da data de criação do peido)
- Número de Vagas
- Motivo de Contratação
- Confidencialidade

PDR02 - Listar e filtrar pedidos de recrutamento

Como *Técnico* ou *Master*, quero poder listar e filtrar pedidos de recrutamento.

Sequência de acções:

1. Aceder à opção de menu “Pedidos de Recrutamento”.

Os pedidos de recrutamento apresentados a cada utilizador estão dependentes das permissões de acesso do utilizador. Adicionalmente, deve ser possível filtrar as listas pelos seguintes critérios:

- Código do Pedido
- Estado
- Área
- Empresa
- Departamento
- Função

PDR03 - Enviar pedidos de recrutamento para o fluxo de aprovação

Como *Técnico* ou *Master*, quero poder enviar para aprovação um pedido criado e devidamente preenchido.

Sequência de acções:

1. Listar os pedidos de recrutamento;
2. Seleccionar o pedido pretendido;
3. Clicar no botão “Enviar Para Aprovação”.

PDR04 - Cancelar um pedido de recrutamento

Como *Técnico* ou *Master*, quero poder cancelar um pedido. Isto só é permitido aos pedidos que se encontrem “Em Preenchimento” ou “Em Aprovação”.

Sequência de acções:

1. Listar os pedidos de recrutamento;
2. Seleccionar o pedido pretendido;
3. Clicar no botão de “Cancelar Pedido”.

PDR05 - Anexar documentos a um pedido de recrutamento

Como *Técnico* ou *Master*, quero poder anexar documentos a um pedido de recrutamento realizado.

Sequência de acções:

1. Listar os pedidos de recrutamento;
2. Seleccionar o pedido pretendido;
3. Seleccionar a opção de anexar ficheiros.
4. Escolher os ficheiros
5. Guardar as alterações.

Pedidos de Aprovação

AP01 - Listar e filtrar pedidos de aprovação.

Como *Técnico* ou *Master*, quero poder listar os pedidos de aprovação que tenho para validar/rejeitar.

Sequência de acções:

1. Clicar na opção do menu “Pedidos de Aprovação”.

AP02 - Actuar sobre um pedido de aprovação.

Como *Técnico* ou *Master*, quero poder aprovar, rejeitar ou colocar em “stand by” um pedido de aprovação.

Sequência de acções:

1. Clicar na opção do menu “Pedidos Realizados”;

2. Para o pedido pretendido, clicar no botão de “lupa”;
3. Seleccionar qual das opções quer realizar (Aprovar, Rejeitar ou Colocar em “stand by”).

AP03 - Reencaminhar pedido para outro utilizador

Como *Técnico* ou *Master*, quero poder reencaminhar um pedido de aprovação para outro utilizador.

Sequência de acções:

1. Aceder à opção de menu do “Pedidos de Aprovação”;
2. Clicar no botão de “Reencaminhar” do pedido pretendido;
3. Escolher qual o utilizador para que se quer reencaminhar.

Se for um pedido de substituição apenas o Técnico III pode realizar esta acção. Para os outros tipos de pedidos todos os utilizadores podem reencaminhar.

AP04 - Fazer um pedido de esclarecimento sobre um pedido de aprovação.

Como *Validador*, quero poder fazer um pedido de esclarecimento a um utilizador que tenha participado no fluxo de aprovação de um pedido de recrutamento.

Sequência de acções:

1. Aceder à lista de notificações;
2. Seleccionar o pedido pretendido no botão de “lupa”;
3. Seleccionar a opção “Pedir Esclarecimentos”;
4. Seleccionar a pessoa a que pretendem pedir esclarecimento (apenas pessoas que já tenham actuado no fluxo);
5. Escrever o que pretende esclarecer.
6. Enviar pedido de esclarecimento.

AP05 - Responder a um pedido de esclarecimento.

Como *Validador*, quero poder responder a um pedido de esclarecimento.

Sequência de acções:

1. Aceder à lista de notificações;;
2. Seleccionar o pedido pretendido no botão de “lupa”;
3. Aqui são possíveis duas opções:
 - (a) Escrever o necessário;
 - (b) Reencaminhar para outra pessoa:

- i. Seleccionar a pessoa a que pretendem pedir mais esclarecimentos (apenas pessoas que já tenham actuado no fluxo)
4. Confirmar;

AP06 - Criar perfis de aprovação.

Como *Master*, quero poder criar perfis de aprovação.

Sequência de acções:

1. Aceder ao módulo “Administração”;
2. Aceder à opção de menu “Workflows”;
3. Aceder à sub-opção “Perfil de Workflow”;
4. Clicar no botão “Novo”;
5. Preencher dados necessários;
6. Guardar alterações

Critérios de aceitação::

- Código
- Descrição

AP07 - Associar perfis de aprovação a determinados fluxos de trabalho.

Como *Master*, quero poder associar perfis de aprovação a determinados fluxos de trabalho.

1. Aceder ao módulo “Administração”;
2. Aceder à opção de menu “Workflows”;
3. Aceder à sub-opção “Processos de Workflow”;
4. Clicar no botão “Editar” do fluxo pretendido;
5. Seleccionar a etapa à qual se quer associar;
6. Detalhar o perfil nessa etapa.

AP08 - Configurar o fluxo de trabalho de aprovação.

Como *Master*, quero poder configurar um fluxo de aprovação.

Sequência de acções:

1. Aceder ao módulo “Administração”;
2. Aceder à opção de menu “Workflows”;
3. Aceder à sub-opção “Processos de Workflow”;

4. Clicar no botão “Editar” do fluxo pretendido;
5. Criar Etapas necessárias.

Processo de Recrutamento

PCR01 - Criar um processo de recrutamento.

Como *Master*, quero poder criar um processo de recrutamento sem necessidade de aprovação.

Sequência de acções:

1. Clicar no botão de “Novo Processo”;
2. Preencher os dados necessários.
3. Guardar os dados.
4. É reencaminhado para a página de edição de processo.

Critérios de aceitação::

Para se poder registar um processo, os seguintes campos são obrigatórios:

- Área de Negócio
- Empresa
- Departamento
- Função
- Local de Trabalho
- Data de Início de Funções Pretendida (por defeito, o dia de hoje mais 15)
- Número de Vagas
- Motivo de Contratação
- Confidencialidade

PCR02 - Listar e filtrar os processos aos quais tem acesso.

Como *Técnico I/Técnico II/Técnico III/Master*, quero poder ver (listar) os vários processos.

Sequência de acções:

1. Clicar na opção do menu “Processos”.

Filtros:

- Código do Pedido.
- Código do Processo.
- Estado do Processo.

PCR03 - Colocar um processo em *StandBy*.

Como *Técnico I/Técnico II/Técnico III/Master*, quero poder alterar o estado de um processo para “StandBy”. Sequência de acções

1. Listar processos;
2. Seleccionar o processo em questão;
3. Clicar no botão “StandBy”.

PCR04 - Encerrar um processo.

Como *Técnico I/Técnico II/Técnico III/Master*, quero poder encerrar um processo. Esta opção é útil nos casos em que um processo é criado sem necessidade de aprovação com o intuito de avaliar o mercado, então é criado sem vagas. Nestes casos é necessário terminar o processo manualmente.

Sequência de acções:

1. Aceder à opção de menu “Processos”;
2. Clicar no botão de “Editar” do processo pretendido;
3. É reencaminhado para a página de edição de um processo;
4. Clicar no botão “Encerrar”.

PCR05 - Pedir alteração do responsável de um processo.

Como *Técnico I*, quero poder atribuir um técnico como novo responsável de um processo.

Sequência de acções:

1. Clicar no botão de “Editar” presente na lista de processos.
2. É reencaminhado para a página de edição do processo.
3. Clicar o botão “Sugerir Atribuição”.
4. Seleccionar o novo técnico.
5. Guardar alteração.

PCR06 - Alterar o técnico associado a um processo de recrutamento.

Como *Técnico II/Técnico III/Master*, quero poder alterar o técnico associado a um processo de recrutamento.

Sequência de acções:

1. Clicar no botão de “Editar” presente na lista de processos;
2. É reencaminhado para a página de edição do processo;

3. Clicar no botão de “Editar” da área “Identificação do Processo”;
4. Selecciona qual o novo técnico;
5. Guarda alteração.

PCR07 - Actuar sobre um pedido de alteração de técnico.

Como *Técnico III/Master* quero poder actuar sobre o pedido de alteração do técnico.

Sequência de acções:

1. Seleccionar o item do menu com os pedidos;
2. Seleccionar o pedido específico;
3. Fazer a decisão.

Gestão de Candidatos

GC01 – Registrar um candidato.

Como *Técnico I/Técnico II/Técnico III/Master*, quero poder criar um candidato. Sequência de acções:

1. Clicar no botão de “Novo Candidato”;
2. Preencher os dados necessários.
3. Guardar os dados.
4. É reencaminhado para a página de lista de Candidatos.

Critérios de aceitação:

Para se poder registar um candidato, os seguintes campos são obrigatórios:

- Email pessoal
- Nome

GC02 – Associar um candidato a um processo (criar uma candidatura).

Como *Técnico I/Técnico II/Técnico III/Master*, quero poder associar um candidato a um processo.

Sequência de acções:

Clicar no botão de “Associar” presente na lista de candidatos.

1. É reencaminhado para uma página com a lista de processos.
2. Selecionar o processo e a fase pretendidos.
3. Guardar a alteração.

GC03 – Listar e filtrar a lista de candidatos.

Como *Técnico I/Técnico II/Técnico III/Master*, quero poder ver (listar) os vários candidatos existentes.

Sequência de acções:

1. Aceder à opção de menu “Candidatos”;

Filtros:

- Área Funcional
- Habilitações Literárias
- Função
- Pesquisar dentro de CVs por palavras-chave
- Área Geográfica
- Idade
- Expectativa Salarial
- Disponibilidade
- Número de Pedido ou Processo
- Estado de Candidato
- Situação Atual

GC04 – Convidar candidato.

Como *Técnico I/Técnico II/Técnico III/Master*, quero poder convidar candidatos.

Sequência de acções:

1. Aceder à opção de menu “Candidatos”;
2. Clicar na opção de “Convidar” do candidato pretendido;
 - (a) É enviado um email a convidar o candidato para preencher o seu perfil.

GC05 – Avaliar um candidato.

Como *Técnico I/Técnico II/Técnico III/Master*, quero poder avaliar candidatos.

Gestão de Eventos

GE01 - Configurar os passos e eventos de um determinado processo.

Por detalhar...

GE02 - Configurar os dados pedidos num determinado evento.

Por detalhar...

GE03 - Realizar um evento.

Por detalhar...

Publicação de Anúncios**PA01 - Criar e editar o *template* de um anúncio.**

Por detalhar...

PA02 - Publicar um anúncio.

Por detalhar...

PA03 - Republicar um anúncio.

Por detalhar...

PA04 - Remover um anúncio publicado.

Por detalhar...

Candidatos**C01 - Registrar perfil.**

Como utilizador não registado, quero poder registar um novo perfil, para que possa, posteriormente, iniciar sessão com os meus dados.

Sequência de acções:

1. Clicar no botão de registar;
2. Preencher os campos;
3. Guardar os dados.
 - (a) Após guardar os dados, o sistema irá enviar um email de confirmação, de modo a validar totalmente o registo. Este email consiste num email standart de confirmação, informando o utilizador que acabou de se registar neste site, com um link para confirmar a criação da conta.
4. É reencaminhado, como utilizador externo, para a página de edição de perfil.

Critérios de aceitação:

Para se poder registar um perfil, os seguintes campos são obrigatórios:

- Nome;
- Data de Nascimento;
- Género;

- Nacionalidade;
- NIF;
- Email;
- Palavra-Passe;
- Telemóvel;
- Permissão RGPD;
- Áreas Pretendidas.

C02 - Candidatar-se a um anúncio.

Como utilizador interno/externo, quero poder candidatar-me a um anúncio, para entrar no processo de recrutamento para a(s) vaga(s) disponível(eis).

Sequência de acções:

1. Acede à página da informação do anúncio pretendido.
 - (a) Caso já se tenha candidatado, não é possível repetir a operação. Sendo impossível realizar as operações seguintes – botão de “candidatar” estará inactivo.
2. Clica no botão “candidatar”.
3. É reencaminhado para a página de candidatura.
4. Preenche a primeira parte do formulário (os dados do perfil, podendo editá-los).
 - (a) Se editar os dados, estes serão actualizados no perfil do candidato.
5. Preenche a segunda parte do formulário. Esta é específica para cada candidatura (*killer questions* e outros dados).
6. Clica no botão “Candidatar”.
 - (a) Envia um email a confirmar o registo da candidatura.
7. É reencaminhado para uma página de sucesso.

C03 - Fazer uma candidatura espontânea.

Como actor do sistema, quero poder registar uma candidatura espontânea.

Sequência de acções:

1. Clicar no botão de “Candidaturas Espontâneas”;
2. Preencher os campos;
3. Guardar os dados.
4. Após guardar os dados, o sistema irá enviar:

- (a) um email de confirmação, de modo a validar a candidatura realizada;
 - (b) um email a perguntar se não deseja criar um perfil no sistema para futuras ofertas que possam vir a existir, caso seja um utilizador não registado (os emails enviados podem ser um só).
5. É reencaminhado para uma página a indicar o sucesso da realização da candidatura.

Critérios de aceitação:

Para se poder registar uma candidatura espontânea, os seguintes campos são obrigatórios:

- Nome;
- Data de Nascimento;
- Género;
- Nacionalidade;
- NIF;
- Email;
- Palavra-Passe;
- Telemóvel;
- Permissão RGPD;
- Áreas Pretendidas.

C04 - Listar anúncios.

Como utilizador não registado ou utilizador externo, quero poder ver (listar) os vários anúncios externos, para me poder candidatar aos processos.

Sequência de acções:

O utilizador pode chegar a esta página através de dois métodos:

1. Pelo link nos sites das empresas.
2. Clicar no botão de “Anúncios Externos” presente na barra de navegação do site, acessível em qualquer página.

Filtros:

É possível filtrar os anúncios apresentados por:

- Área Funcional;
- Distrito;
- Regime de Trabalho (Full-Time e Part-Time).

C05 - Eliminar registo.

Como *Candidato* desejo poder eliminar uma candidatura realizada.

Sequência de acções;

1. Seleccionar opção "Minhas candidaturas";
2. Seleccionar candidatura que se deseja eliminar;
3. Seleccionar opção de eliminar.

Parametrizações

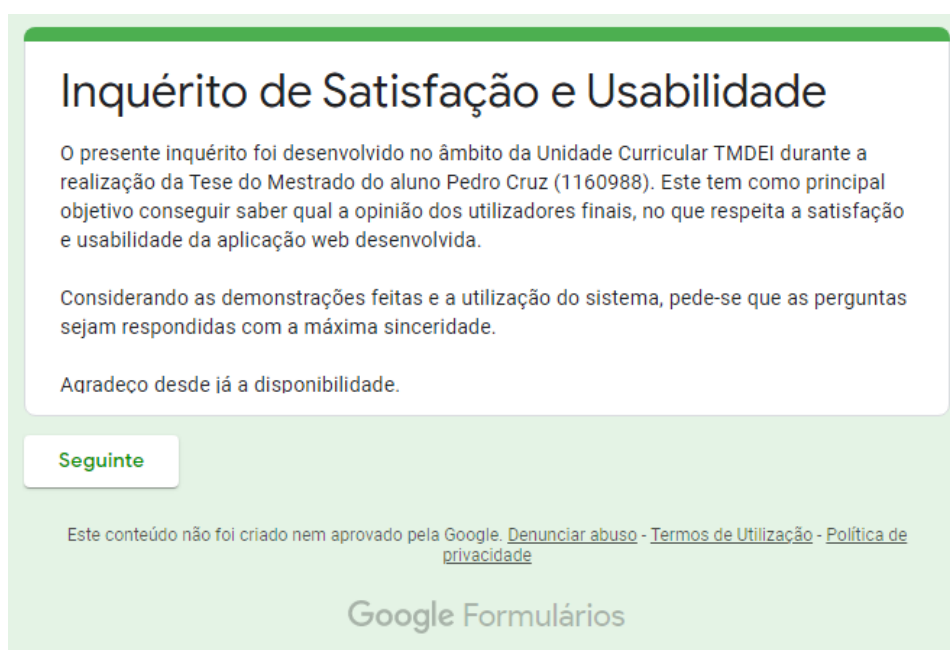
Este conjunto de processos elementares é exposto de uma maneira distinta. Como são todos as mesmas operações (listar, adicionar, editar e eliminar) e estão todos na mesma página, listam-se em baixo todas as parametrizações que devem existir:

- PP-01 – Listar as parametrizações;
- PP-02 – Adicionar áreas de formação;
- PP-03 – Editar áreas de formação;
- PP-04 – Eliminar áreas de formação;
- PP-05 – Adicionar cargas horárias;
- PP-06 – Editar cargas horárias;
- PP-07 – Eliminar cargas horárias;
- PP-08 – Adicionar condições contratuais;
- PP-09 – Editar condições contratuais;
- PP-10 – Eliminar condições contratuais;
- PP-11 – Adicionar conhecimentos informáticos;
- PP-12 – Editar conhecimentos informáticos;
- PP-13 – Eliminar conhecimentos informáticos;
- PP-14 – Adicionar equipamentos;
- PP-15 – Editar equipamentos;
- PP-16 – Eliminar equipamentos;
- PP-17 – Adicionar (níveis de) experiência profissional;
- PP-18 – Editar (níveis de) experiência profissional;
- PP-19 – Eliminar (níveis de) experiência profissional;
- PP-20 – Adicionar horas semanais;
- PP-21 – Editar horas semanais;

- PP-22 – Eliminar horas semanais;
- PP-23 – Adicionar idiomas;
- PP-24 – Editar idiomas;
- PP-25 – Eliminar idiomas;
- PP-26 – Adicionar instituições de ensino;
- PP-27 – Editar instituições de ensino;
- PP-28 – Eliminar instituições de ensino;
- PP-29 – Adicionar motivos de contratação;
- PP-30 – Editar motivos de contratação;
- PP-31 – Eliminar motivos de contratação;
- PP-32 – Adicionar níveis dos idiomas;
- PP-33 – Editar níveis dos idiomas;
- PP-34 – Eliminar níveis dos idiomas;
- PP-35 – Adicionar qualificações acadêmicas;
- PP-36 – Editar qualificações acadêmicas;
- PP-37 – Eliminar qualificações acadêmicas;
- PP-38 – Adicionar tipos de contrato;
- PP-39 – Editar tipos de contrato;
- PP-40 – Eliminar tipos de contrato;
- PP-41 – Adicionar tipos de contrato de mobilidade;
- PP-42 – Editar tipos de contrato de mobilidade;
- PP-43 – Eliminar tipos de contrato de mobilidade;
- PP-44 – Adicionar tipos de equipamento;
- PP-45 – Editar tipos de equipamento;
- PP-46 – Eliminar tipos de equipamento

Apêndice C

Inquérito de Satisfação e Usabilidade



Inquérito de Satisfação e Usabilidade

O presente inquérito foi desenvolvido no âmbito da Unidade Curricular TMDEI durante a realização da Tese do Mestrado do aluno Pedro Cruz (1160988). Este tem como principal objetivo conseguir saber qual a opinião dos utilizadores finais, no que respeita a satisfação e usabilidade da aplicação web desenvolvida.

Considerando as demonstrações feitas e a utilização do sistema, pede-se que as perguntas sejam respondidas com a máxima sinceridade.

Agradeço desde já a disponibilidade.

Seguinte

Este conteúdo não foi criado nem aprovado pela Google. [Denunciar abuso](#) - [Termos de Utilização](#) - [Política de privacidade](#)

Google Formulários

Figura C.1: Inquérito - Página Inicial

Inquérito de Satisfação e Usabilidade

Inquérito de Satisfação e Usabilidade

A utilização do sistema como um todo é intuitiva.

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

O design aplicado ao sistema é o indicado.

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

A navegação entre páginas e a realização das operações são intuitivos.

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

Figura C.2: Inquérito - Questões Parte 1

A solução aparente ser estável (poucas falhas a nível de bugs).

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

O tempo de resposta da aplicação é o indicado.

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

Qual a sua avaliação da solução inicial de 0 a 5.

1 2 3 4 5

Muito Insatisfeito ☐ ☐ ☐ ☐ ☐ Muito Satisfeito

Considera que a aquisição do sistema será uma mais valia?

1 2 3 4 5

Discordo Totalmente ☐ ☐ ☐ ☐ ☐ Concordo Totalmente

Anterior

Submeter

Limpar formulário

Figura C.3: Inquérito - Questões Parte 1



Figura C.4: Inquérito - Respostas Parte 1

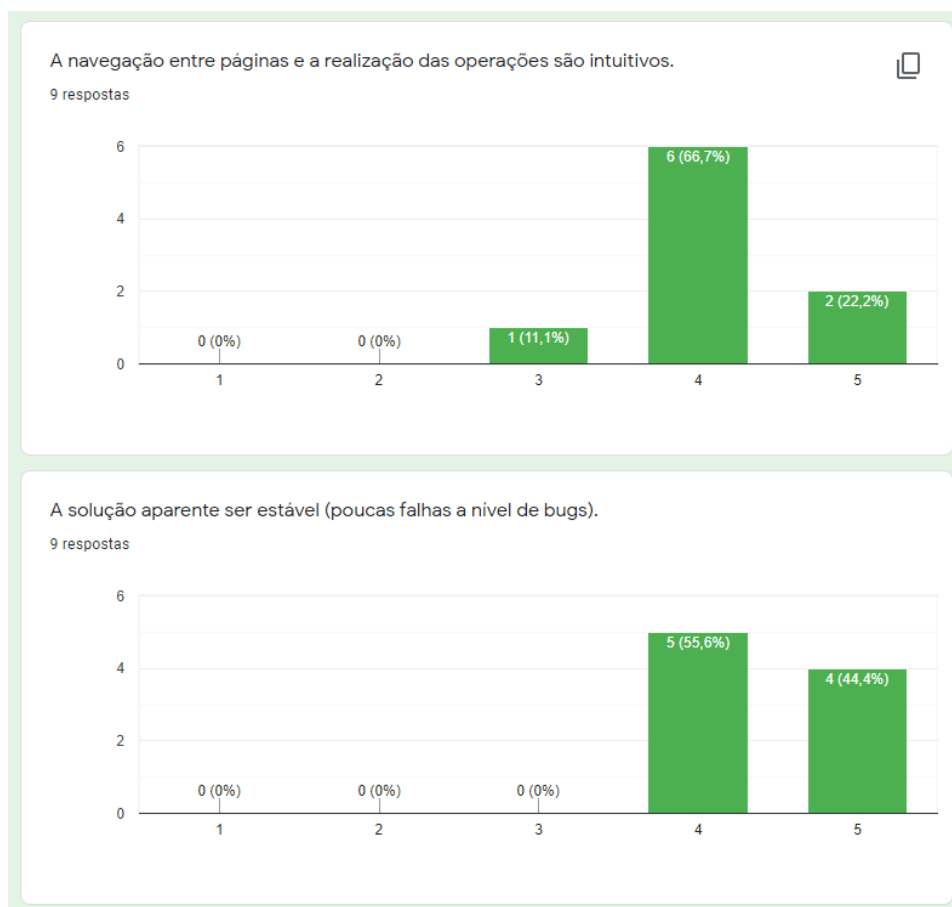


Figura C.5: Inquérito - Respostas Parte 2

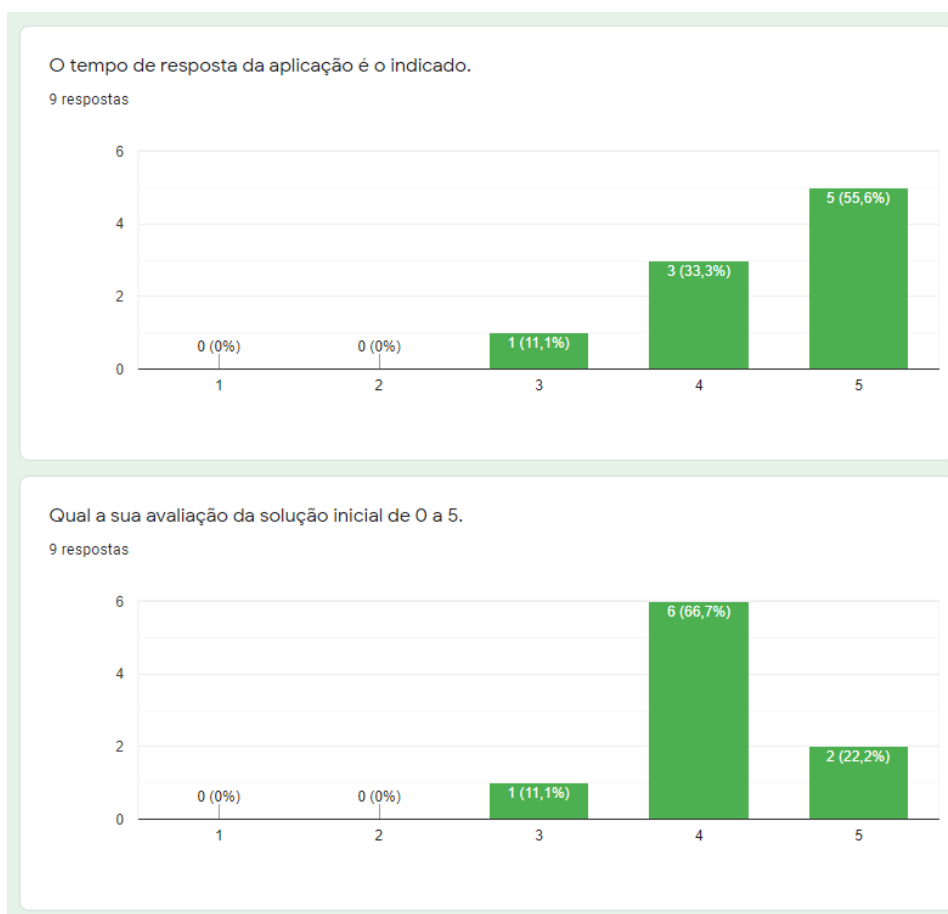


Figura C.6: Inquérito - Respostas Parte 3

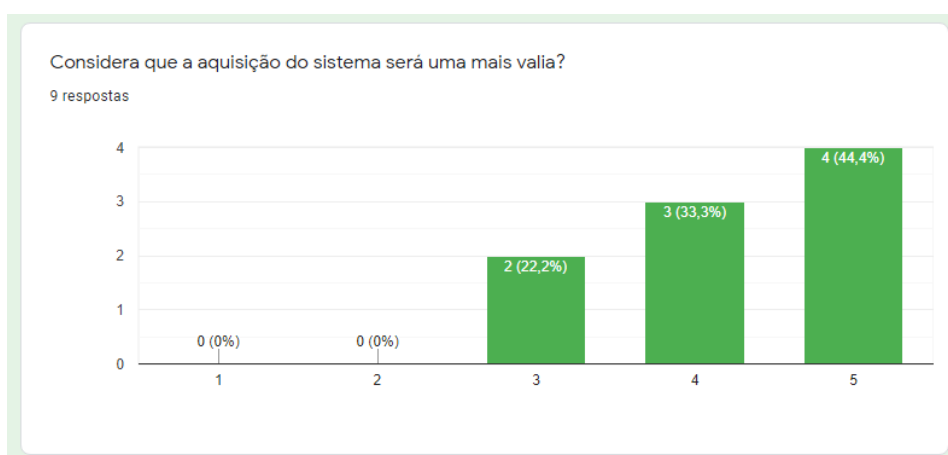


Figura C.7: Inquérito - Respostas Parte 4