



NOVA

IMS

Information
Management
School

MGI

Mestrado em Gestão de Informação

Master Program in Information Management

**Sistema de Informação de suporte ao
Recrutamento e Seleção de Candidatos na área
de consultoria**

Gonçalo Galvão Cruz Ribeiro

Trabalho de Projeto apresentado como requisito parcial para
obtenção do grau de Mestre em Gestão de Informação

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

SISTEMA DE INFORMAÇÃO DE SUPORTE AO RECRUTAMENTO E SELEÇÃO DE CANDIDATOS NA ÁREA DE CONSULTORIA

por

Gonçalo Galvão Cruz Ribeiro

Trabalho de Projeto apresentado como requisito parcial para a obtenção do grau de Mestre em Gestão de Informação, Especialização em Gestão de Sistemas e Tecnologias de Informação.

Orientador: Prof. Dr^o Vítor Manuel Pereira Duarte dos Santos, NOVA IMS

Orientadora: Prof. Dr^a Sara Ribeiro, NOVA IMS

Setembro 2020

AGRADECIMENTOS

O maior agradecimento faço-o às pessoas mais importantes para mim. À minha família, aos meus avós, aos meus amigos e a todos que contribuíram a vários níveis para que este projeto fosse alcançado. Agradeço à Professora Dr^a Sara Ribeiro e ao Professor Dr^o Vítor Manuel Pereira Duarte dos Santos pela orientação e motivação dada ao longo de todo o percurso e por todas as suas sugestões e conselhos que contribuíram para o bom desenvolvimento do trabalho.

RESUMO

O recrutamento e a seleção de candidatos é uma das principais tarefas na gestão de recursos humanos. Nos dias de hoje o recrutamento e seleção na área de consultoria é um grande desafio para qualquer organização, sendo investido grande quantidade de tempo e dinheiro na procura de recursos, que por vezes são escassos e não superam as expectativas.

Este projeto tem como objetivo construir um sistema de informação com base na identificação e análise dos principais processos no recrutamento e seleção para uma organização na área de consultoria. Através deste sistema será possível ter uma plataforma que permitirá uma gestão mais eficaz e a automatização de processos com técnicas inovadoras, na qual será possível adicionar e realizar entrevistas interativas dentro do sistema. No futuro, este sistema contribuirá para o desenvolvimento dentro da organização em estudo e poderá servir para outras empresas interessadas.

PALAVRAS-CHAVE

Sistema de Informação; Gestão de Recursos Humanos; Recrutamento; Seleção; Desenvolvimento de Software.

ABSTRACT

The recruitment and selection of candidates is one of the main tasks in human resource management. Nowadays, such roles in the consulting field are significantly challenging for any organization due to a considerable amount of money and time invested in finding resources, which sometimes are scarce and do not exceed expectations.

This project aims to build an information system based on both the identification and analysis of the leading recruitment and selection processes within an organization in the consulting market. Through this system it will be possible to have a platform that will allow a more effective management and automation of processes with innovative techniques, in which it will be possible to add and perform interactive interviews within the system. Therefore, this system will contribute to the development of the respective organization and it might serve other interested companies.

KEYWORDS

Information System; Human Resource Management; Recruitment; Selection; Software Development.

ÍNDICE

1. Introdução	12
1.1. Contextualização.....	13
1.2. Objetivos.....	14
1.3. Resultados Esperados.....	15
2. Plano de Trabalho	16
2.1. Fases do Projeto / Etapas	16
2.2. Ferramentas / Recursos.....	16
2.2.1. UML.....	16
2.2.2. SQL.....	17
2.2.3. Java	18
2.3. Cronograma.....	18
3. Enquadramento Teórico.....	19
3.1. Gestão de Recursos Humanos.....	19
3.2. Organizações	25
3.3. Consultoria	26
3.3.1. Conceito	26
3.3.2. Consultoria Tecnológica em Organizações.....	27
3.4. Sistemas de Informação	28
3.5. Desenvolvimento de Software	32
3.5.1. Conceito	32
3.5.2. Metodologias de Desenvolvimento de Software.....	33
Metodologias Tradicionais	34
Metodologias Ágeis	42
4. Desenvolvimento do Projeto	52
4.1. Metodologia de Desenvolvimento.....	52

4.2. Levantamento e Análise de Requisitos Funcionais	53
4.2.1. Requisitos Funcionais.....	54
4.2.2. Diagramas	56
Diagrama Caso de Uso	56
Diagrama de Atividade	62
Diagrama de Estado.....	63
4.3. Desenho e Análise de Sistema	64
4.3.1. Diagrama de Classe.....	64
4.3.2. Modelo Lógico de Dados	65
4.3.3. Desenho de Interfaces	66
4.4. Desenvolvimento Do Sistema	67
4.4.1. Arquitetura	67
4.4.2. Modelo Físico de Dados	73
4.4.3. Programação do Sistema.....	74
4.4.4. Testes	95
5. Considerações Finais	98
5.1. Síntese	98
5.2. Contribuições do Projeto	98
5.3. Limitações e Trabalho Futuro	99
Bibliografia / Referências Bibliograficas	100
Anexos.....	104

ÍNDICE DE FIGURAS

Figura 1 - Calendarização prevista do projeto (Fonte: Elaboração própria)	18
Figura 2 - Processo de ocupação de um cargo (Fonte: Camara et al., 2010)	22
Figura 3 – Processo de Recrutamento e Seleção (Fonte: Camara et al., 2010)	24
Figura 4 - Funções do Sistema de Informação (Adaptado de Laudon e Laudon (2012)).....	30
Figura 5 - As três dimensões dos Sistemas de Informação (Adaptado de Laudon & Laudon (2012)).....	30
Figura 6 – Representação do Modelo em Cascata (Fonte: Pressman, 2010)	35
Figura 7 - Modelos de Processo Incrementais (Fonte: Pressman, 2010).....	36
Figura 8 - Modelo de Prototipagem (Fonte: Pressman, 2010)	38
Figura 9 - Modelo Em Espiral (Fonte: https://www.gratispng.com/png-b8wnan/)	39
Figura 10 - Modelo Iterativo (Fonte: Rational Unified Process, 1998)	41
Figura 11 – Ciclo no processo XP (Fonte: Pressman, 2010)	44
Figura 12 - Processo de Desenvolvimento Scrum. (Fonte: Scrum Portugal)	46
Figura 13 - Fases de desenvolvimento de software adaptável (Fonte: Pressman, 2010)	47
Figura 14 - A família de metodologias Crystal (Fonte: http://www.devx.com/architect/ article/32836/0/page/2)	49
Figura 15 - Modelo do processo DSDM de Desenvolvimento de Software (Fonte: Processdata-DSDM.png).	50
Figura 16 - Esquema geral de requisitos	54
Figura 17 - Descrição de Atores no sistema	56
Figura 18 - Caso de Uso do SI (Fonte: Elaboração própria)	57
Figura 19 - Caso de Uso relativo ao Processo de Recrutamento (Fonte: Elaboração própria)	58
Figura 20 - Caso de Uso relativo ao Processo de Candidaturas (Fonte: Elaboração própria)	59
Figura 21 - Caso de Uso relativo ao Processo de Seleção (Fonte: Elaboração própria)	60
Figura 22 - Caso de Uso relativo ao Processo de Contratação (Fonte: Elaboração própria) .	61

Figura 23 - Diagrama de Atividade do SI (Fonte: Elaboração própria)	62
Figura 24 - Diagrama de Estado do SI (Fonte: Elaboração própria)	63
Figura 25 - Diagrama de Classes do SI (Fonte: Elaboração própria).....	64
Figura 26 - Modelo Lógico de Dados do SI (Fonte: Elaboração própria).	65
Figura 27 – Esquema com a organização das páginas e as suas funcionalidades (Fonte: Elaboração própria)	66
Figura 28 - Diagrama de Sequência com a Arquitetura MVC (Fonte: Elaboração própria). ..	67
Figura 29 - Arquitetura Oracle ADF (Fonte: Ronald, 2011)	68
Figura 30 – Exemplo da Entity Object – Projetos.....	69
Figura 31 – Exemplo do VO - Tabela Lista de Candidatos	69
Figura 32 – Exemplo do AppModule relativo aos Candidatos.....	70
Figura 33 – Exemplo dos Data Controls utilizados na aplicação.....	70
Figura 34 - Exemplo das Managed Beans utilizadas no desenvolvimento da aplicação	71
Figura 35 – Exemplo da Interface do JDeveloper - Oracle ADF com a página para adicionar uma nova candidatura	72
Figura 36 - Modelo Físico de Dados do SI (Fonte: Elaboração própria).....	73
Figura 37 - Componente de conexão com a BD do sistema.....	74
Figura 38 - Data Controls e objetos filhos listados utilizados na aplicação.....	75
Figura 39 – Painel com JSF's, Managed Beans (p. ex. Projeto.java), Page Definition (p.ex. Novo Projeto) and Task Flow (adfc-config) utilizado no SI.....	77
Figura 40 - Task Flow da Aplicação.....	78
Figura 41 - Ecrã de Login	79
Figura 42 – Lógica para verificar se o Username e Password existem na BD	79
Figura 43 - Ecrã de Homepage	80
Figura 44 - Ecrã de Projetos.....	81
Figura 45 - Código XML gerado com a tabela Projetos.....	81
Figura 46 - Ecrã de Criação de Novo Projeto	82
Figura 47 - Ecrã Processos de Recrutamento	82

Figura 48 - Page Definition da página Processo de Recrutamento	83
Figura 49 - Ecrãs Criação de Novo Processo de Recrutamento e Competências Processo de Recrutamento	83
Figura 50 - Lógica para o mapeamento de elemento “Nome Processo” entre a JSF e a classe Java Processos de Recrutamento e excerto de código para inserir novo registo.....	84
Figura 51 - Ecrã Detalhes Processo de Recrutamento	85
Figura 52 - Ecrã de Candidaturas	85
Figura 53 - Ecrã Criação de Candidatura e Adiciona Competências Candidato.....	86
Figura 54 - Ecrã Detalhes de Candidatura	87
Figura 55 - Exemplo de código para filtrar as competências de um determinado candidato na página Detalhes Candidatura	88
Figura 56 - Popup Agendar Entrevista	88
Figura 57 - Ecrã Entrevistas	89
Figura 58 – VO com query de SQL para Entrevistas Ativas	89
Figura 59 - Ecrã Detalhes / Avaliar Entrevista	90
Figura 60 - Lógica para Adicionar Perguntas e Respostas na Entrevista	91
Figura 61 - Ecrã Contratação	92
Figura 62 - Ecrã Novo Pedido de Contratação.....	92
Figura 63 – Interface do JDeveloper retirado da JSF utilizada no ecrã Novo Pedido de Contratação	93
Figura 64 - Ecrã Pedido de Contratação.....	94
Figura 65 - Código com lógica para Aprovar um contrato na classe Contratos.....	94

ÍNDICE DE TABELAS

Tabela 1 - Tabela com Requisitos Funcionais do SI	54
Tabela 2 - Caso de Uso e Fluxo de Eventos relativos ao Processo de Recrutamento	58
Tabela 3 - Caso de Uso e Fluxo de Eventos relativos ao Processo de Candidaturas	59
Tabela 4 - Caso de Uso e Fluxo de Eventos relativos ao Processo de Seleção	60
Tabela 5 - Caso de Uso e Fluxo de Eventos relativos ao Processo de Contratação	61
Tabela 6 - Caderno de Testes.....	95

LISTA DE SIGLAS E ABREVIATURAS

ADF	Application Development Framework
API	Application Programming Interface
ASD	Adaptive Software Development
BD	Base de Dados
CDU	Casos de Uso
DDL	Data Definition Language
DML	Data Manipulation Language
DSDM	Dynamic Systems Development Management
GRH	Gestão de Recursos Humanos
JEE	Java Enterprise
JSF	JavaServer Faces
MVC	Model-View-Controller
RSC	Recrutamento e Seleção de Candidatos
RH	Recursos Humanos
RUP	Rational Unified Process
SI	Sistema de Informação
SQL	Structured Query Language
TI	Tecnologias de Informação
UML	Unified Modeling Language
VO	View Object
XP	Extreme Programming
XML	Extensible Markup Language

1. INTRODUÇÃO

O aumento na procura de talento para a área de consultoria numa empresa implica a contratação certa dos seus colaboradores. Nos últimos anos muitos estudos apontam para os preços elevados na contratação de novos colaboradores, uma vez que quando uma pessoa é contratada, é necessária formação adequada. A formação acaba por ser um grande investimento para uma organização (Fatemi, 2018), daí a importância de uma escolha acertada.

Nos dias de hoje, grande parte do sucesso de uma consultora depende dos processos de recrutamento e a sua taxa de retenção subsequente. Estima-se que em cada 100 contratações, 13 falham ou as pessoas abandonam a organização durante o período de experiência (Vieira, 2019).

A importância do recrutamento e seleção de candidatos (RSC) para uma organização torna-se clara, e a sua boa utilização transforma todos os custos em retornos, permitindo uma maior produtividade para a organização que, por sua vez, atinge os seus objetivos estratégicos.

A era digital transformou a forma como é feito o RSC. Os processos como enviar o currículo, marcar a entrevista ou fazer um teste abrem novas portas para ferramentas capazes de levar as empresas a um maior nível de assertividade nas suas contratações.

Uma pesquisa realizada pelo LinkedIn em 2018, com 9000 gestores de contratações em 39 países apontou quatro tendências para os processos de seleção nos próximos anos: diversidade, novas formas de entrevista, dados e inteligência artificial (LinkedIn, 2018). Este estudo relevou ainda que cada vez mais é fundamental as empresas analisarem os seus dados, de forma a verem quais são os profissionais que estão a crescer e são mais adequados, se os índices de rotação dentro da organização estão corretos e o porquê das várias entradas e saídas de colaboradores na organização.

Na consultora objeto de estudo, todos os processos de RSC são manuais, não acompanhando a era digital a que assistimos, estando assim, de certa maneira, desatualizados. Desta forma surgiu a ideia de criar um sistema de informação (SI) que permite recrutar e selecionar candidatos de forma a acelerar estes processos e ter mais controlo sobre os mesmos. Deste modo foi realizado um estudo e análise da empresa em questão, permitindo assim identificar de forma específica os requisitos base para a seleção e recrutamento de candidatos.

Este projeto tem o seu foco na construção de um SI com base em todas as variáveis provenientes do processo de RSC da empresa em estudo, que permitirá ter toda a informação relativa aos candidatos centralizada.

Propõe-se assim colmatar as necessidades de gestão no RSC de uma consultora, tendo o seu foco no desenvolvimento de um SI inovador para todos os processos identificados, de modo que exista uma melhoria dos mesmos, com o objetivo de criar uma plataforma central para a organização gerir todo o seu RSC.

1.1. CONTEXTUALIZAÇÃO

Cada vez mais o RSC compreende um conjunto de processos que necessitam de ser controlados por uma organização. No caso da consultora em estudo não existe, atualmente, um sistema capaz de centralizar a informação, existindo assim a necessidade de um sistema feito à medida da mesma, que responda às necessidades de maneira a centralizar todo este tipo de gestão de informação interna.

É perceptível que a organização tem um sistema interno de gestão de RSC desatualizado, no qual toda a informação é guardada de forma desorganizada numa pasta de computador, tornando os processos manuais. Isto faz com que só a pessoa que tem acesso ao computador tenha também acesso às informações, o que não se justifica.

Para a organização, a construção deste projeto é um ponto importante para o seu desenvolvimento interno, proporcionando uma ferramenta capaz de acelerar e melhorar todos os seus processos de RSC. Este projeto vem colmatar um ponto crítico existente na organização, pois a falta deste sistema traz graves problemas para a organização, afetando o seu desempenho, o que pode ser facilmente resolvido através da criação um SI personalizado às necessidades da organização.

Espera-se, com este projeto, alterar o processo de recrutamento, diminuindo os problemas associados. Este vai permitir a agilização de todo o processo, passando a informação necessária ao recrutamento a estar centralizada, vai também facilitar a triagem de *curricula vitae*, manter atualizado o histórico de candidatos, a empresa vai poder ter uma plataforma para registar as entrevistas e adicionar um conjunto de perguntas por área de departamento.

Assim, as principais vantagens serão manter todas as etapas do processo controladas, respeitar prazos e garantir a qualidade no final de todo o processo. Economicamente, a empresa será beneficiada na medida em que as tarefas de recrutamento ficarão facilitadas e mais organizadas, existindo logo à partida uma triagem dos candidatos adequados ao trabalho. Isto evita a seleção e investimento desadequado que poderia resultar num futuro despedimento, com prejuízo para a empresa.

Apesar de no mercado já existirem sistemas capazes de gerir os Recursos Humanos (RH), nomeadamente o recrutamento e seleção, este sistema tem um grande aspeto que o torna único, o facto de ir de encontro às necessidades específicas da empresa em estudo.

O SI a desenvolver apresenta vantagens como o seu preço, uma vez que a empresa não pagará pelo seu desenvolvimento. A potencialidade do crescimento deste sistema no futuro e a sua personalização, com a escolha do formato dos ecrãs constituem duas outras vantagens para a empresa. Apresenta também algumas desvantagens, como o facto de o projeto não estar preparado para um grande nível de complexidade, precisar de manutenção (Sommerville, 2011) e o facto da tecnologia estar sempre em mudança, podendo a tecnologia em uso tornar-se obsoleta e ser necessário alterar tudo.

1.2. OBJETIVOS

O presente projeto tem como objetivo a conceção e desenvolvimento de um SI capaz de assegurar a gestão do RSC dentro da consultora em estudo. Será proposto um sistema eficaz e inovador para proporcionar uma melhor gestão dentro dos RH da mesma.

Para cumprir os objetivos declarados a execução do trabalho foi dividida em várias tarefas, sendo elas:

- Análise da empresa de consultoria em estudo;
- Pesquisa de requisitos funcionais;
- Análise de requisitos funcionais;
- Planeamento e desenho do SI;
- Desenvolvimento do SI;
 - Construção de uma base de dados (BD) capaz de suportar o funcionamento do software;
 - Construção de toda a interface do sistema;
- Testes e Implementação

1.3. RESULTADOS ESPERADOS

No final deste projeto é esperada a criação de um SI, que terá um papel importante na consultora em estudo, pois vai permitir obter uma visão da situação atual da mesma, encontrando assim pontos a melhorar em relação ao RSC, proporcionando o seu crescimento e desenvolvimento no mercado, cada vez mais competitivo.

Este sistema será capaz de automatizar e criar novos mecanismos mais dinâmicos no RSC, promovendo a melhoria dos mesmos, com o objetivo de inovação e uma melhor performance para a organização. Desta forma será possível reduzir ou eliminar os problemas associados ao RSC.

Espera-se ainda que este projeto sirva como exemplo para outras organizações, e possa ser reutilizado de maneira a facilitar o trabalho dos RH no RSC, tornando-o mais fácil e direto, eficaz e eficiente, melhorando assim a sua performance.

2. PLANO DE TRABALHO

2.1. FASES DO PROJETO / ETAPAS

O presente projeto é dividido em cinco grandes etapas:

- **Identificação, Análise e Definição da Necessidade/Problema**

Nesta primeira etapa, é identificada uma necessidade, e é feito todo o estudo e análise para a definição da mesma.

- **Planeamento do Trabalho**

Nesta etapa é feito todo o planeamento necessário de maneira a organizar e planejar a execução do projeto.

- **Enquadramento Teórico**

O objetivo desta etapa é perceber e estudar as áreas subjacentes ao projeto, de modo a que ao longo do desenvolvimento do projeto sejam tomadas as melhores decisões.

- **Desenvolvimento do Projeto**

Na etapa do desenvolvimento do projeto é implementada toda a parte prática do projeto.

- **Considerações Finais**

Por fim nesta última etapa pretende-se analisar e perceber todos os resultados, contribuições, limitações e conclusões do projeto.

2.2. FERRAMENTAS / RECURSOS

2.2.1. UML

Unified Modeling Language (UML) é uma linguagem padrão para construir esquemas de software. UML pode ser utilizado para visualização, especificação, construção e documentação de artefactos (Booch et al., 1999).

A linguagem UML é apropriada para a modelação de sistemas, permite definir o design e a estrutura do projeto. É uma linguagem padronizada, muito expressiva, abordando todas as visões necessárias para desenvolver e implantar sistemas.

Neste trabalho vai ser utilizada a notação UML para a modelação do sistema, a ferramenta Lucidchart será utilizada para criar todos os diagramas de UML de uma forma simples e intuitiva com base na análise, desenho e levantamento de requisitos. Existem vários diagramas importantes para representar o sistema:

- **Diagrama de Atividade:** Este simplifica a interpretação da lógica dentro do sistema, permitindo perceber a sequência de ações existentes e as suas ligações.
- **Diagrama de CDU (Casos de Uso):** É utilizado para descrever a interação entre os atores e o sistema, descrevendo as ações, relações e dependências com base nos requisitos do mesmo.
- **Diagrama de Classe:** Permite definir as diferentes classes que o sistema contém, definindo deste modo todos os seus atributos para cada classe;
- **Diagrama de Estado:** Permite representar a evolução de estados dentro de um processo. Este diagrama permite a interpretação do comportamento dos estados dentro do sistema.
- **Diagrama Relacional:** Permite ver as relações entre as várias classes do sistema, permitindo assim uma visão dos fluxos que existem na BD. É possível identificar as entidades informacionais do SI e as suas propriedades/características.

2.2.2. SQL

Structured Query Language (SQL) é a linguagem padrão para a gestão e desenvolvimento de BD relacionais, permite gerir, aceder e comunicar com a BD. No desenvolvimento da BD relacional, será utilizada a versão gratuita da Oracle *Database 18c Express Edition*. A ferramenta Oracle SQL *Developer*, também irá ser utilizada para complementar o desenvolvimento da mesma, através de comandos de SQL.

As principais tarefas do SQL são: criar, apagar, editar e ler dados de uma BD. No desenvolvimento do projeto a linguagem SQL servirá para a construção do modelo de BD relacional, sendo que numa fase inicial será construído o Data Definition Language (DDL) de maneira a definir a estrutura da BD e de seguida o Data Manipulation Language (DML) de forma a manipular os dados consoante as necessidades do sistema através de instruções de comandos SQL (Fehily, 2014).

2.2.3. Java

O projeto irá ser devolvido com base na linguagem de programação Java. Será utilizada a ferramenta Oracle JDeveloper 12c através da *framework* Oracle Advanced Development Framework (ADF). Esta ferramenta é gratuita e simplifica bastante o desenvolvimento da parte de programação do projeto. Este permite um desenvolvimento mais eficaz e interativo, tendo por base a linguagem de programação de Java.

Java é uma linguagem de programação orientada a objetos, permite criar aplicações para todo o tipo de ambientes, é constituída por classes e objetos. Cada classe contém métodos, variáveis e constantes. A programação em Java é intuitiva e eficiente. A linguagem de programação Java vai permitir codificar toda a lógica durante o desenvolvimento do projeto (Schildt, 2007).

O desenvolvimento em Oracle ADF terá como servidor aplicacional o Oracle WebLogic, que está presente na ferramenta JDeveloper 12c, permitindo assim testar a aplicação e a preparar a sua instalação no futuro.

2.3. CRONOGRAMA

Na Figura 1 apresenta-se a calendarização prevista do projeto.

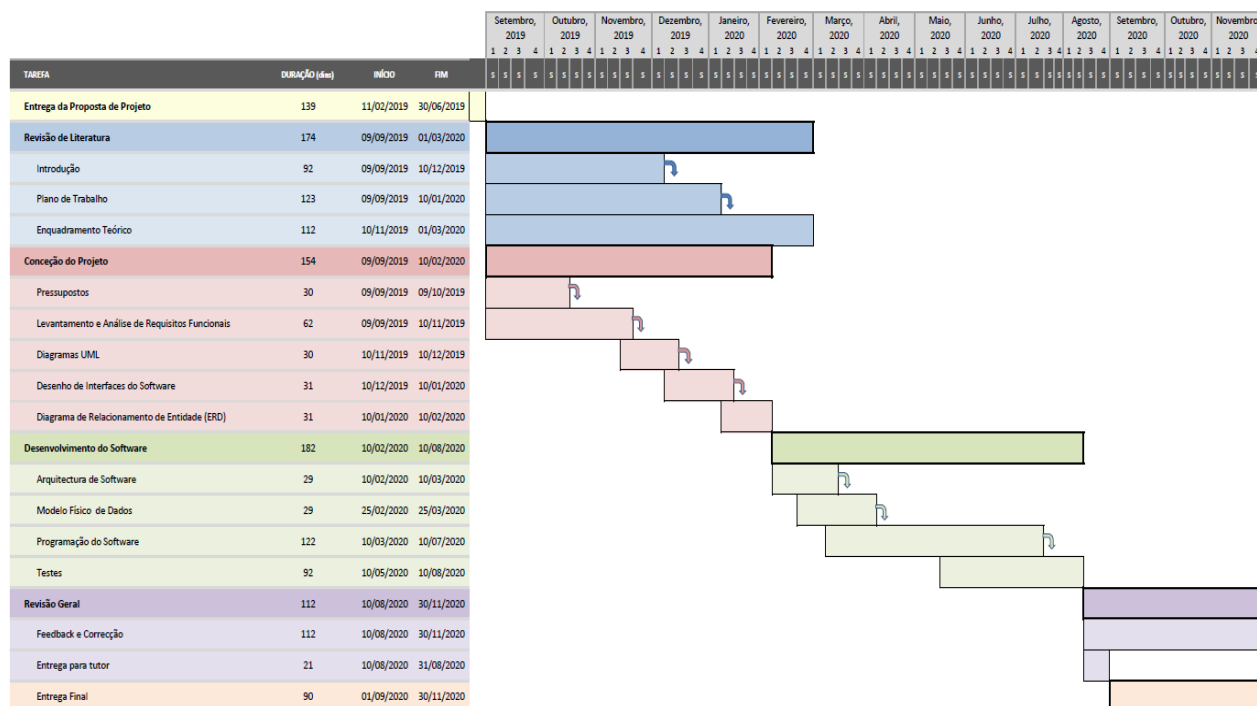


Figura 1 - Calendarização prevista do projeto (Fonte: Elaboração própria)

3. ENQUADRAMENTO TEÓRICO

O presente capítulo explica as principais áreas estudadas no desenvolvimento deste projeto, permitindo assim saber o estado de arte de cada uma das áreas e justificar as decisões que foram tomadas ao longo do mesmo.

3.1. GESTÃO DE RECURSOS HUMANOS

A gestão de RH consiste no planeamento, organização, desenvolvimento, coordenação e controlo de técnicas capazes de promover o desempenho eficiente das pessoas, ao mesmo tempo que a organização constitui o meio que permite às pessoas que colaboram com esta alcançar os objetivos individuais relacionados direta ou indiretamente com o trabalho, sendo o seu objetivo último a construção de organizações melhores, mais rápidas, pró-ativas e competitivas (Chiavenato, 1995).

A globalização, a desregulação e as rápidas mudanças tecnológicas que se fazem sentir na atualidade trouxeram novas formas de pensar e ver o mundo, incluindo no seio organizacional, exigindo a criação de uma nova forma de planeamento estratégico nas empresas, de forma a promover a eficácia e eficiência organizacional e é assim que nasce o conceito de Gestão de Recursos Humanos (GRH).

O ser humano é um recurso presente nas organizações, sendo os GRH uma forma de gerir as pessoas que trabalham numa empresa, assegurar os seus direitos e deveres e promover formas destas trabalharem de forma segura e produtiva.

Os RH incidem sobre diferentes áreas numa empresa, todas elas com a sua importância. Vários autores definem que as áreas de RH mais recorrentes são recrutamento e seleção (Armstrong & Taylor, 2014; Camara et al., 2010; Chiavenato, 1995), formação e desenvolvimento de competências (Armstrong & Taylor, 2014; Camara et al., 2010; Chiavenato, 1995), acolhimento e integração (Chiavenato, 1995), avaliação de desempenho (Armstrong & Taylor, 2014; Camara et al., 2010), planeamento (Camara et al., 2010), gestão de carreiras e retenção do pessoal (Armstrong & Taylor, 2014), apesar de existirem muitas outras funções que completam todas as práticas.

A avaliação do desempenho dos funcionários consiste no desenvolvimento da estratégia que permita avaliar a prestação dos indivíduos e verificar se esta se encontra de acordo com os objetivos estratégicos da empresa. A avaliação de desempenho pode ser útil a vários níveis uma vez que permite verificar se o recrutamento foi eficaz, analisar o contributo do funcionário para o alcance dos objetivos da empresa, verificar se o sistema de remuneração é eficiente, identificar necessidades de formação e identificar o potencial ou não nos funcionários (Camara et al., 2010). Deste modo e com

base neste *feedback* a avaliação de desempenho pode apoiar na determinação e desenvolvimento de uma política apropriada às necessidades da organização (Chiavenato, 1995).

A formação e desenvolvimento de competências é uma das ferramentas mais importantes para a gestão eficiente (Camara et al., 2010). As organizações sofrem mudanças constantemente, e para isso é necessário que os seus colaboradores acompanhem essas mesmas mudanças, sendo a formação e desenvolvimento de novas competências um caminho a seguir (Chiavenato, 1995). Atualmente existe muita competitividade no mercado de trabalho, nomeadamente na área das Tecnologias de Informação (TI), logo as organizações têm de assegurar a sua sobrevivência, e quanto mais qualificada for a mão-de-obra maior será a resposta a novos desafios (Camara et al., 2010).

A gestão de carreiras e retenção de pessoal é um plano que permite a evolução e desenvolvimento dos colaboradores, com o objetivo destes puderem continuar a contribuir de forma positiva para o alcance dos seus próprios objetivos bem como os da empresa, é papel dos RH assegurar que isto aconteça (Camara et al., 2010). Na área da TI aposta-se bastante na formação dos seus colaboradores, variando o intervalo de tempo que dura esta formação (Costa et al., 2016) o que, inequivocamente, traz custos à empresa, logo é necessário assegurar a qualidade da formação bem como garantir que os formandos fiquem o máximo tempo na organização que os formou. Muitas organizações despendem muito dinheiro para o treino de um novo candidato, por vezes estas contratações falham, sendo gasto um grande investimento em tempo e dinheiro.

O acolhimento e integração é um processo que, tal como o nome indica, cria formas do colaborador se sentir parte integrante e importante da empresa em que se insere. O objetivo principal é acolher, orientar e integrar o candidato na organização. No processo de integração é importante que o colaborador obtenha não só as competências desejadas, mas também motivação e desejo de continuar na empresa (Chiavenato, 1995). É importante referir que o processo de integração e acolhimento está dependente de tempo, adaptação e aprendizagem.

O planeamento ou a chamada gestão administrativa de RH pode ser caracterizada como planeamento adequado dos recursos humanos, que assenta no desenvolvimento de competências que permitam responder às mudanças que ocorrem no contexto organizacional. O planeamento consiste na previsão daquelas que possam vir a ser necessidades futuras de uma organização. Envolve um alto nível de conhecimento organizacional e do mercado que o rodeia, tanto ao nível do presente como o futuro.

A GRH necessita de planeamento, desenvolvimento, coordenação e práticas que promovam o desempenho eficiente e assertivo, na medida em que favorecem e apoiam a estratégia de negócio da organização e não trazem custos de estrutura desnecessários (Camara et al., 2010).

Os RH são assim um pilar para o sucesso de uma organização. Oliveira et al. (2014) afirmam que devido às organizações procurarem cada vez mais melhorar os seus resultados para assim vingar no mercado, a gestão de pessoas torna-se cada vez mais importante como estratégia da organização, para assim poder ser possível o desenvolvimento de métodos que promovam o alcance de objetivos organizacionais.

No caso da organização em estudo, o principal foco no desenvolvimento do projeto é a área do recrutamento e seleção.

Recrutamento e Seleção

Os RH têm um papel fundamental no RSC, pois quando estamos perante uma tarefa muito complexa, o desempenho colaborativo depende das habilidades e características das pessoas selecionadas (Bedwell et al., 2012), devendo estar de acordo com os objetivos organizacionais associado a um menor tempo de aprendizagem e com o maior nível de execução (Chiavenato, 1995; Chraif, 2013). É papel dos RH, durante o recrutamento identificar quais os candidatos que se adequam ou não aos objetivos da empresa, evitando assim trazer custos desnecessários à mesma.

O recrutamento é composto por técnicas e procedimentos que visam atrair candidatos potencialmente qualificados e capazes de ocupar cargos dentro da organização, em número suficiente para que a mesma disponha dos RH necessários à prossecução dos seus objetivos.

A Figura 2 apresenta o processo de ocupação de um cargo (Camara et al., 2010).

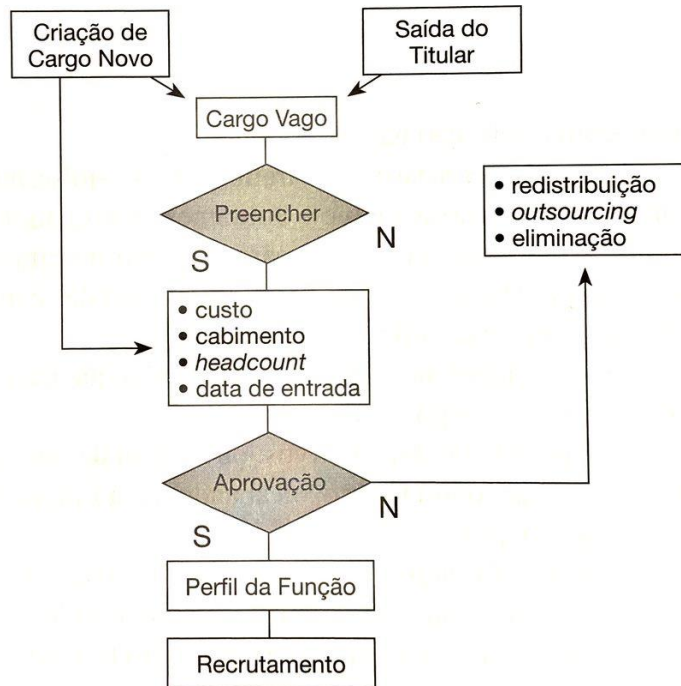


Figura 2 - Processo de ocupação de um cargo (Fonte: Camara et al., 2010)

Numa empresa a entrada e saída de pessoas é constante, e quando existe um lugar vago, é papel dos RH encontrar o candidato ideal para o ocupar. Como indicado na Figura 2, primeiramente é possível ver se o lugar surgiu porque realmente é necessário acrescentar alguém ou porque uma outra pessoa saiu do projeto. Depois segue-se a decisão se é mesmo necessário preencher esse lugar. Caso a resposta seja “não” as tarefas daquele colaborador têm de ser desenvolvidas por outro recurso já existente, onde se redistribui ou reorganiza a informação, ou caso se justifique elimina-se as funções que o recurso que saiu desempenhava. Caso seja mesmo necessário contratar outra pessoa, é necessário decidir se o fará interna ou externamente, verificando os custos e as vantagens de cada um dos casos. Por fim, os RH iniciam o processo de recrutamento, onde o SI desenvolvido para este projeto poderá dar resposta, pois através dele segue um pedido de recrutamento onde se irá definir que tipo de perfil é necessário para assumir aquele cargo.

Como mencionado anteriormente existem duas formas de recrutamento, o interno e o externo (Camara et al., 2010; Chiavenato, 1995; Michel, 2007). O Recrutamento Interno, onde se procuram possíveis candidatos dentro da própria organização, tem várias vantagens como ser mais económico e mais rápido e desvantagens como o conflito de interesses. Este processo exige que os RH tenham uma boa coordenação com os outros departamentos da empresa.

O Recrutamento Externo realiza-se através do recurso ao arquivo de candidaturas espontâneas, apresentação de candidatos por parte dos funcionários da empresa, recrutamento online / redes sociais, centro de emprego, entre outros. Quando se fala em candidatura espontânea significa que a pessoa envia o seu *curriculum vitae* em algum momento para quando houver alguma vaga para a sua área os RH possam entrar em contacto com a pessoa.

A seleção de candidatos consiste na seleção das pessoas cujos objetivos e características melhor se adequam às da empresa. Armstrong & Taylor (2014) definem a seleção de candidatos como um processo em que se compara as exigências da função com aquilo que o candidato consegue oferecer, as suas características e competências. As técnicas de seleção para a escolha de um candidato são as entrevistas de seleção, provas de conhecimentos, gerais ou específicas, testes psicométricos, testes de personalidade e testes de simulação (Chiavenato, 1995).

A entrevista de seleção consiste na interação interpessoal entre um candidato e um ou mais entrevistadores para a ocupação de uma vaga de emprego, na qual se procura estruturar e identificar os conhecimentos, competências e comportamentos do candidato que vão sobressair sobre o cargo, influenciando o seu sucesso (Armstrong & Taylor, 2014). O entrevistador não se pode limitar a avaliar o candidato de acordo com as exigências do cargo e da empresa, tem também de ver aspetos que o qualifiquem como um bom candidato, ou seja, ter em atenção aspetos como as suas características pessoais. A forma como o candidato se apresenta na empresa, a sua aparência, cuidado, a forma como fala, as empatias criadas não interferem no cargo que a pessoa vai desempenhar, mas afetam a seleção do entrevistador, mesmo que inconscientemente.

As entrevistas podem ser agrupadas consoante a forma como são realizadas, atendendo ao grau de estruturação e ao número de intervenientes. Destacamos assim quatro dos principais tipos de entrevistas: entrevista não estruturada realizada sem qualquer tipo de planeamento ou guião inicial, entrevista estruturada planeada com antecedência e obedece a um número de perguntas previamente definido, existindo um guião, por forma a adequar-se às características próprias da função em causa obtendo bons níveis de validade preditiva (Robertson & Smith, 2001), entrevistas em grupo que permitem avaliar os candidatos num contexto que requer um elevado grau de interação social e teste a sua expressão oral, e as entrevistas semi-estruturadas onde existe um guião de entrevista inicial, mas que com o decorrer da conversa pode sofrer alterações, acrescentando ou retirando perguntas que façam sentido.

As entrevistas de seleção devem disponibilizar um sinal claro sobre o seu resultado. Na avaliação do candidato devem ser registados os aspetos mais importantes da entrevista e o resultado da mesma.

A Figura 3 permite sintetizar e demonstrar o encadeamento de todo o processo de RSC, permitindo perceber como se pode realizar o recrutamento e seleção, tendo este processo várias etapas (Camara et al., 2010).

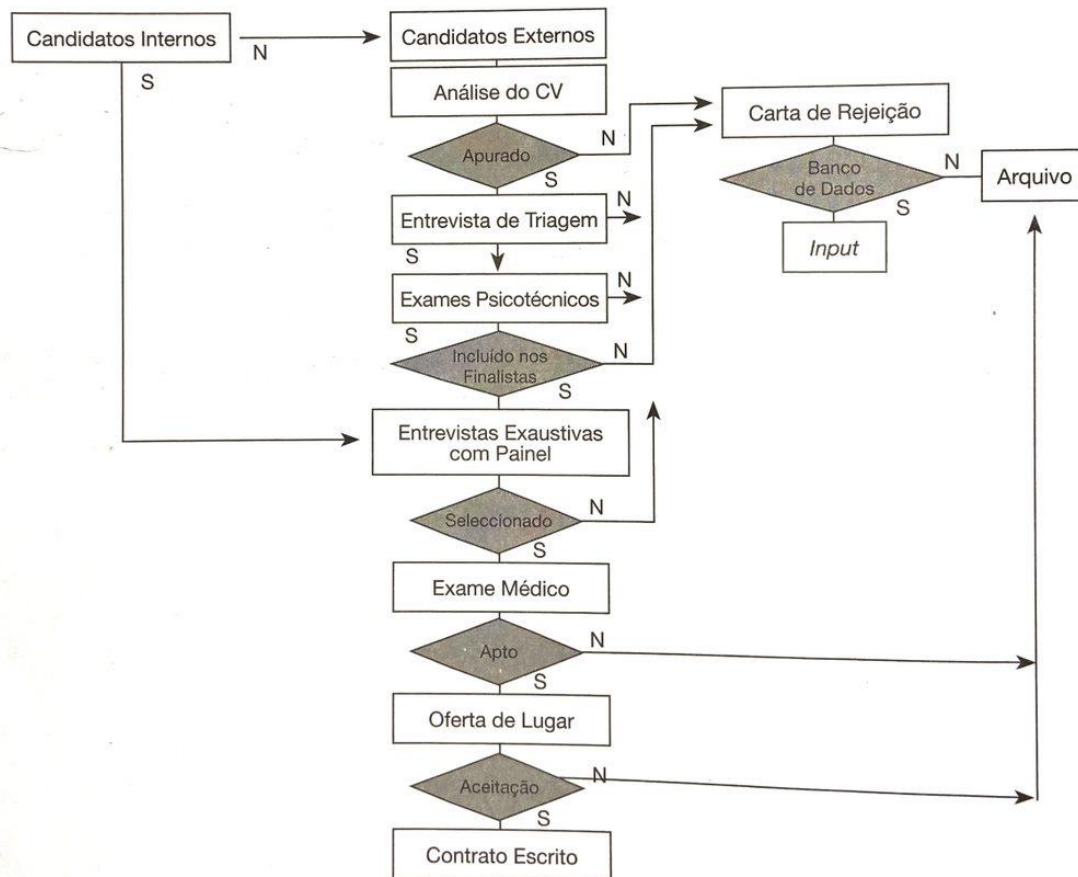


Figura 3 – Processo de Recrutamento e Seleção (Fonte: Camara et al., 2010)

A concretização destes passos permite que a organização tenha uma correta adequação das pessoas aos cargos, consequentemente trazendo outras vantagens à empresa e colaboradores.

3.2. ORGANIZAÇÕES

Segundo Karlsson (2014), uma organização pode ser entendida como um sistema composto por vários indivíduos que juntos formam comunidades organizacionais, local abstrato onde o conhecimento é criado, desenvolvido e transformado. Bilhim (2006) descreve ainda que uma organização é uma entidade social conscientemente coordenada e que possui fronteiras delimitadas, funcionando continuamente com o objetivo de realizar objetivos comuns através de uma divisão de tarefas, onde existe uma hierarquia de autoridade e responsabilidade.

Estas definições realçam a parte formal de uma empresa, como a sua estrutura hierárquica, os papéis, os recursos, o atingir de objetivos, e também valoriza a sua dimensão informal em que as interações e cooperações entre as pessoas são suportadas e sustentadas pela cultura organizacional. A cultura organizacional descreve o ambiente em que a pessoa trabalha, ambiente esse que tem uma forte influência na sua forma de pensar, agir e de experienciar o trabalho (Warrick, 2017).

As organizações são deste ponto de vista analisadas através das suas características económicas, políticas, dos seus processos, estrutura e liderança. Todas estas características têm impacto no funcionamento da organização (Laudon & Loudon, 2012).

Ao longo dos últimos anos a globalização e a crescente pressão dos mercados em acompanhar as evoluções tecnológicas e económicas levam a que as organizações tenham que se adaptar a um novo mundo empresarial, caracterizado por intensa competitividade e instabilidade.

As organizações nos dias de hoje necessitam de ter uma estratégia muito bem definida e alinhada com as suas áreas centrais e com os seus objetivos em que, por vezes, quando necessitam de recorrer a áreas que não são as suas, são levadas a depender de fontes externas.

Estas fontes externas de conhecimento tornam-se fundamentais para a empresa, pois apresentam-se como solução para problemas que a organização não consegue resolver por si só. Este processo está a crescer cada vez mais e a ser visto como a estratégia e o caminho para a evolução empresarial (Canback, 1998).

3.3. CONSULTORIA

3.3.1. Conceito

A crescente pressão dos mercados, caracterizada por uma enorme competitividade e instabilidade, leva as organizações a centrarem-se na sua área de especialidade e a recorrer a fontes externas para colmatar as áreas que não dominam. Surge assim o papel da consultoria.

Consultoria pode ser definida como um serviço contratado e prestado a organizações, que é caracterizada pelo uso intensivo de conhecimento, para assim ajudar a resolver problemas e a criar soluções através de pessoas especialmente treinadas e qualificadas (Greiner & Metzger, 1983). Surge com o aparecimento de um problema para uma organização, na qual dentro da própria não é possível resolvê-lo, o que faz com que surja a necessidade de recorrer a fontes externas.

Deste modo ocorre assim um contrato de serviço de apoio, de maneira a disponibilizar às organizações, pessoas qualificadas e especializadas em determinada área e prontas a identificar, analisar e a criar soluções (Greiner & Metzger, 1983).

Oliveira et al. (2014) aponta que a consultoria irá incidir em cima de um diagnóstico, sendo este realizado pelo consultor em conjunto com os colaboradores da organização. Depois de realizado o diagnóstico serão encontrados os problemas o que originará um contrato entre as organizações interessadas no qual o papel do consultor é apontar as melhores maneiras de resolver os mesmos, através dos seus funcionários altamente especializados, tanto para identificar como para resolver os problemas (Oliveira, 2014 sit in Schuster & Friedrich, 2017).

É de salientar que o consultor não tem papel decisório, é sim um mediador entre o problema e a solução, a decisão final cabe sempre ao cliente.

No que toca ao tipo de contrato este dependerá de alguns fatores, como o tamanho da empresa, o tipo de problema que necessita ser resolvido, o nível financeiro da empresa e a abertura para receber novas ideias vindas de fora (Jacintho, 2004 sit in Schuster & Friedrich, 2017).

Deste modo é possível concluir que a consultoria é um dos pontos principais do processo de tomada de decisão, sendo que o cliente terá, através da consultoria, condições para visualizar e projetar o que cada decisão trará no futuro (Crocco & Guttman, 2005 sit in Schuster & Friedrich, 2017), o que acaba por ser um caminho de sucesso para as organizações alcançarem os seus objetivos estratégicos.

3.3.2. Consultoria Tecnológica em Organizações

Existem várias áreas dentro da consultoria, de acordo com a FEACO (Fédération Européenne des Associations de Conseils en Organisation, 2019), uma delas é a consultoria de TI, sendo o foco do presente projeto (Currenti et al., 2019).

Nos dias de hoje, cada vez mais as organizações necessitam de inovar e ter sistemas sofisticados, mais eficientes e robustos capazes de acompanhar a evolução tecnológica.

As empresas têm grande parte do seu orçamento centrado na tecnologia, que tem crescido de forma exponencial, pois utilizam sistemas muito dispendiosos e têm custos de mudança muito elevados, o que faz com que por vezes esta mudança não surja (Laudon e Laudon, 2012).

A principal razão para isto ocorrer é o facto dos custos de software e de serviços de consultoria serem bastante elevados. Com a evolução atual do mercado tecnológico, as empresas precisam de atualizar o seu conhecimento constantemente, em busca de soluções mais eficientes e inovadoras.

As organizações tendem cada vez mais a alocar a entidades externas parte do desenvolvimento, controlo e acompanhamento dos seus sistemas e procuram respostas para os seus problemas através de soluções tecnológicas.

Por tudo isso, as organizações recorrem cada vez mais a entidades externas, ou seja, consultoras especializadas na área de TI que se ocupam da implementação e manutenção destes SI.

Estas organizações, ao recorrerem a consultoria em TI conseguem ter libertação de recursos, permitindo a organização ter foco na sua principal área e permite que exista uma melhor gestão dos recursos, sendo que os custos fixos passam a custos variáveis, sendo assim apenas utilizados quando necessário.

3.4. SISTEMAS DE INFORMAÇÃO

É primordial definir o termo Sistemas de Informação (SI), sendo que diferentes autores apresentam diferentes definições, é então necessário selecionar aquela que mais se adequa ao que se pretende estudar.

É importante referir que o estudo de um SI é um campo multidisciplinar, ou seja, pode abranger campos desde a estratégia, à gestão, às atividades operacionais envolvidas na recolha, ao processamento de dados, à distribuição e utilização da informação e respetivas tecnologias nas organizações. (Serrano et al., 2004)

Começando por Sommerville (2011), este autor define um SI como um sistema com três principais capacidades, a de organizar, a de armazenar e a de estruturar os dados com a finalidade de responder às necessidades e possíveis problemas das organizações. O seu principal objetivo é organizar e fornecer acesso a uma BD, tendo como problemas a segurança, usabilidade, privacidade e manutenção da integridade dos dados.

Laudon & Laudon (2012) definem SI como um conjunto de componentes inter-relacionados que recolhem, processam, armazenam e distribuem informação com o objetivo de apoiar a tomada de decisão, coordenação, controlo, análise e visualização numa organização.

O'Brien (1993), descreve os SI como um conjunto de pessoas, procedimentos e recursos envolvidos na recolha, no processamento e disponibilização de informação na organização.

Kroenke (2008) define um SI composto por cinco componentes, hardware, software, dados, procedimentos e pessoas. Com base nas componentes apresentadas por Kroenke, podemos definir:

- **Software**

O software consiste nas instruções detalhadas pré-programadas que controlam e coordenam os componentes de hardware do computador num SI, possibilitando ao utilizador interagir com o SI, para analisar, processar e organizar os dados gerados.

- **Hardware**

O hardware é o equipamento físico usado para atividades de entrada, processamento e saída num SI, ao seja máquinas e servidores onde o SI é executado.

- **Pessoas**

Um dos componentes mais importantes e que mais influencia o sucesso de um SI (Kroenke, 2008). Todos os sistemas precisam de pessoas para a sua utilização e execução de modo a alcançar os objetivos da organização.

- **Procedimentos / Redes**

Meios e processos de comunicação que permitem a partilha e conexão de informações dentro do SI.

- **Dados**

Conjunto de valores ou ocorrências que o SI permite aceder e atualizar. Após o processamento dos dados surge a Informação que terá de ser vista como um recurso extremamente importante nas organizações, tão importante como o capital ou as pessoas, visto que sem Informação adequada não podem sobreviver, pelo que este recurso deverá ser gerido de forma a tirar o maior proveito possível (Serrano et al., 2004). Os dados têm recebido grande relevância nos últimos anos, são considerados um dos bens mais preciosos para uma organização na medida em que ao envolvermos pessoas, os dados passam a ser informações. The Economist (2017), afirma que os dados são o novo petróleo do Século XXI, sendo este o ponto de partida para um termo que cada vez mais tem tido impacto e se tem tornado o estudo de muitos outros autores na atualidade. O bom processamento dos dados gera uma grande importância na informação que circula dentro das organizações, sendo um dos recursos cuja gestão e o seu aproveitamento mais influencia o sucesso de uma organização (Varajão, 1998).

Após análise das diferentes definições, centramo-nos naquela que mais se adequa ao projeto a realizar, proposta por Laudon & Laudon (2012). Pode dizer-se que os SI têm uma perspetiva técnica, da qual faz parte o seu hardware, software e dados para serem processados e, uma componente empresarial, onde se podem incluir as pessoas e os procedimentos. Existe uma grande importância no que toca à interação da empresa com o meio envolvente, onde o objetivo desta interação é reunir informação para que o SI possa solucionar problemas que possam surgir desta mesma interação (Laudon & Laudon, 2012).

Numa perspetiva técnica, Laudon & Laudon (2012) descrevem três atividades principais para gerar informação dentro de uma organização e no seu meio envolvente, que são o *input*, o processamento e o *output* (Figura 4).

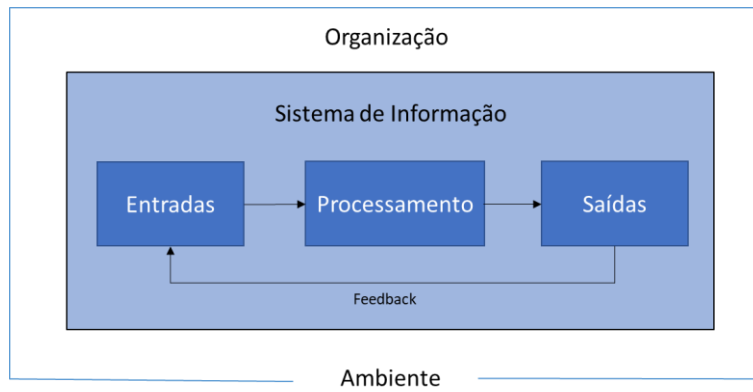


Figura 4 - Funções do Sistema de Informação (Adaptado de Laudon e Laudon (2012))

Como se pode analisar na Figura 4, são três as atividades que geram a informação que uma organização necessita: o *input* (Entradas) é onde os dados são inseridos, depois ocorre o processamento da informação, onde se realiza a conversão dos dados para uma forma que dê significado às organizações e, finalmente, o *output* (Saídas) que transmite a informação já trabalhada aos elementos que estão envolvidos em todo o processo, como os clientes, fornecedores, colaboradores e entidades reguladoras da organização, ou seja, são disponibilizadas às pessoas que as utilizam direta ou indiretamente. Por fim, o *feedback* servirá para avaliar a qualidade da informação processada para a organização.

Numa perspetiva empresarial, para melhor compreender o conceito de SI, é importante identificar as três dimensões que o estruturam numa organização, que são: a gestão, a tecnologia e a organização. (Laudon & Laudon, 2012).



Figura 5 - As três dimensões dos Sistemas de Informação (Adaptado de Laudon & Laudon (2012))

Como se pode observar na Figura 5, a dimensão organizacional dos SI está envolvida nos padrões e cultura de uma organização, da qual fazem parte as pessoas, hierarquia, processos de negócios, cultura e política (Laudon & Laudon, 2012).

Atualmente existe uma grande interdependência, que tende a aumentar, entre a organização e os SI (Laudon & Laudon, 2012). A evolução do hardware, software e redes, incluindo a Internet, têm ajudado as organizações no desempenho das suas funções.

Na realidade, hoje em dia, os SI afetam uma parte muito maior da própria organização, tais como produtos, objetivos e estrutura organizacional. Quando existem mudanças organizacionais a nível estratégico isto exige que a nível de SI existam também mudanças, vemos assim a sua interdependência.

Por outro lado, na dimensão da tecnologia estão presentes o hardware, o software, o armazenamento e redes. As organizações tiram proveito da tecnologia para poder realizar as suas funções necessárias no dia-a-dia. Nesta dimensão está presente a infraestrutura tecnológica da organização, a qual fornece a plataforma onde são construídos os SI.

De outro modo, a dimensão da gestão tem como objetivo a liderança, estratégia e o próprio comportamento na gestão para responder aos desafios de negócio existentes nas organizações.

Assim, os SI fornecem ferramentas e informações necessárias aos gestores para realizar o seu trabalho, permitindo tomar decisões, coordenar, monitorizar e alocar o seu trabalho. Os SI ajudam a impulsionar as operações diárias e as estratégias implementadas na própria gestão.

Atualmente, nas organizações, os SI são algo essencial, facilitando e assegurando negócios, aumentando assim a produtividade da empresa e assegurando a sua sobrevivência no seio das concorrentes. É através de um SI eficaz que será possível para a empresa alcançar uma maior qualidade e baixar os custos de produtos e serviços, finalizando com um aumento do valor acrescentado para os seus clientes (Serrano et al., 2004). Assim sendo, é indispensável que uma empresa que queira obter sucesso organizacional, tenha um SI capaz de auxiliar gestores e outros funcionários a analisar o estado atual da empresa, observando os pontos positivos e a melhorar, para assim poder inovar e, por exemplo, criar novos produtos.

3.5. DESENVOLVIMENTO DE SOFTWARE

3.5.1. Conceito

A Engenharia de Software é uma área da engenharia onde se fornece padrões para o desenvolvimento de softwares, existindo três elementos fundamentais, os métodos, as ferramentas e os procedimentos, que servem de base para a implementação dos mesmos, fazendo-o de forma produtiva e com qualidade. Esta área está relacionada com os vários aspetos do desenvolvimento de software, acompanhando todas as fases do processo (Sommerville, 2011).

Por sua vez o próprio desenvolvimento de software incide num processo cujo objetivo é a criação de um programa de alta qualidade. Este processo é caracterizado por um conjunto de tarefas/atividades previamente ordenadas e planeadas, que originarão um produto final, denominado software (Pressman, 2010; Sommerville, 2011). Software é definido como um conjunto de programas, informações e dados. No desenvolvimento de software existe um conceito importante que suporta e apoia o seu desenvolvimento, este conceito é designado de *framework*. Este consiste numa estrutura genérica composta por um conjunto de classes, objetos e componentes, que são integrados, reutilizáveis e adaptáveis permitindo reutilizar alguns dos seus componentes e conter uma base na construção de uma aplicação (Sommerville, 2011).

Segundo Pressman (2010), para o desenvolvimento de software é necessário executar certas atividades básicas para que assim os objetivos sejam atingidos, sendo elas:

1. Especificação de Requisitos

- a. Análise de Requisitos: Levantamento das necessidades do sistema.
- b. Especificação de Sistema: Descrição funcional do sistema.
- c. Engenharia de Sistema: Criação de uma solução geral para o problema.

2. Projeto de Sistema

- a. Arquitetura de Sistema: Definição da Arquitetura para o software.
- b. Interfaces: Definição das Interfaces para o software.
- c. Desenho do Sistema: Definição dos componentes que o sistema vai traduzir, muitas vezes através de linguagem UML.

3. Programação (Implementação)

- a. Codificação: a implementação do sistema, através da tradução dos requisitos em código.

4. **Verificação e Integração** (Validação)

- a. Teste: a realização de testes para verificar a existência de erros e saber se o sistema responde as necessidades.
- b. Manutenção e Evolução.

Estas atividades básicas permitem um desenvolvimento com uma estrutura padronizada, ou seja, têm certas etapas a serem seguidas e cumpridas, no entanto não existe uma sequência obrigatória.

As mesmas estão presentes em qualquer elaboração de um software, seja ele de que natureza for, independentemente da área em que será aplicado, do quão complexo é e da metodologia a utilizar (Pressman, 2010).

Deste modo um projeto de desenvolvimento de software tem como pontos assentes uma sequência clara e lógica de tarefas que necessitam de forte planeamento e controlo. Num contexto organizacional o surgimento destes projetos advém da necessidade de criar e implementar um software através das próprias oportunidades de negócios e da necessidade da organização em integrar processos e funções para desta forma melhorar o seu controle, coordenação e melhorar também a capacidade de reposta, permitindo dessa forma que a informação passe livremente entre todos os departamentos da empresa (Laudon & Laudon, 2012).

Desde modo é necessário ter um controlo na gestão deste tipo de projetos, sendo um dos principais objetivos a entrega de um produto que, satisfaça as necessidades dos clientes, cumprindo certo prazo e tendo em conta os custos associados.

Podemos concluir que o desenvolvimento de software é uma atividade criativa e complexa, cujo desenvolvimento é influenciado por vários fatores internos e externos que podem colocar em risco o cumprimento de objetivos. Assim surge a necessidade de utilizar uma metodologia de desenvolvimento de software para mitigar riscos e alcançar o sucesso desejado.

3.5.2. Metodologias de Desenvolvimento de Software

Uma metodologia de desenvolvimento de software pode ser definida como uma estrutura padronizada de desenvolvimento, com certas etapas, onde estão presentes métodos, ferramentas e procedimentos que servirão de base para o programador contruir o software com qualidade e ter controlo em todo o processo.

É importante ressaltar que diferentes tipos de software necessitam de uma metodologia que esteja de acordo com as suas características, por exemplo a construção de um sistema pequeno é diferente

de um sistema grande, desde a formação da equipa, o número de tarefas, o planeamento e o tempo que demora a realizar.

No que toca à metodologia de desenvolvimento de software, no momento de escolher a metodologia certa para o tipo de projeto, torna-se por vezes complicado escolher a mais adequada tendo em conta as características. Deste modo perceber e conhecer as metodologias de desenvolvimento de software é essencial, para se poder escolher a melhor opção para as características do projeto.

Existem dois tipos principais de metodologias de desenvolvimento de software, as Metodologias Tradicionais e as Metodologias Ágeis, que se detalham de seguida.

Metodologias Tradicionais

As denominadas metodologias tradicionais, muitas vezes caracterizadas como sendo pesadas, foram bastante utilizadas no passado num contexto de desenvolvimento de software, que é bastante diferente do que conhecemos nos dias de hoje. Antigamente, o custo de fazer alterações e correções num software era elevadíssimo, pois o acesso aos computadores era limitado e não existiam ferramentas modernizadas de apoio ao desenvolvimento do software. Desde modo existia a necessidade de planear e documentar tudo previamente à sua implementação.

O primeiro modelo de metodologia tradicional a surgir foi o modelo em cascata, este modelo serviu de base e referência para a criação de modelos que o sucederam (Pressman, 2010), tendo estes a mesma finalidade e o mesmo ciclo de vida que o modelo em cascata, sendo que cada um destes modelos tem os seus próprios princípios e regras. A estes modelos deu-se o nome de Metodologias Tradicionais.

Estas metodologias têm como principal característica a previsibilidade dos requisitos necessários ao desenvolvimento do sistema, o que torna necessário identificar claramente os requisitos do sistema, sendo um processo bastante rigoroso (Sommerville, 2011).

Existindo várias metodologias, as metodologias tradicionais que se seguem são as mais relevantes para o estudo.

Modelo em Cascata

O método de desenvolvimento do modelo em cascata, às vezes chamado de ciclo de vida clássico, foi criado em 1970 por Winston Walker Royce (Pressman, 2010).

A principal característica é a sua abordagem sequencial e linear na qual para passar para uma fase seguinte é necessário terminar a fase anterior (Larman, 2002; Engholm, 2010). Assim, uma fase só deve ser iniciada após a conclusão daquela que a precede (Pressman, 2010). O resultado que surge de cada etapa é a validação de um ou mais documentos subscritos (Sommerville, 2011). A Figura 6 apresenta o Modelo em Cascata.

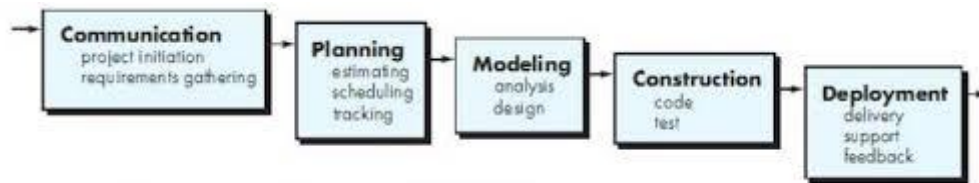


Figura 6 – Representação do Modelo em Cascata (Fonte: Pressman, 2010)

Como podemos verificar na Figura 6 esta metodologia sugere uma abordagem sequencial sistemática ao desenvolvimento de software que começa com a especificação de requisitos do cliente e progride através do planeamento, modelagem, construção e implementação (Deployment), culminando no suporte contínuo ao software concluído (Pressman, 2010).

Este modelo tem vantagens em casos em que o projeto é bastante pequeno e está muito bem definido, devido a ter um desenvolvimento muito rápido, e a diminuir a complexidade de gestão do projeto, e conseguir ter um desenvolvimento bastante rápido, mas tal não acontece para projetos médios e grandes onde as especificações de requisitos estão em constante alteração.

Atualmente os projetos mudam muito no decorrer do seu desenvolvimento, existindo a dificuldade para o cliente colocar todos os requisitos com clareza.

É importante ter em conta que existem alguns problemas nos projetos reais, como o facto dos mesmos não seguirem um fluxo sequencial, existindo também a própria dificuldade sentida pelo cliente em colocar claramente todos os requisitos, o que faz com que quase nunca seja possível expor todas as suas necessidades, fazendo com que possam existir atrasos no seu desenvolvimento, o que se pode tornar desastroso (Pressman, 2010). Ainda para mais no mundo atual em que os requisitos estão em constante mudança e não estão esclarecidos consistentemente.

Deste modo podemos concluir que o modelo em cascata tem dificuldade de acomodar a incerteza natural que existe no início de muitos projetos (Pressman, 2010) e é um modelo que não está preparado para lidar com mudanças (Engholm, 2010).

Modelos de Processo Incrementais

O modelo de processo incremental apresenta elementos do modelo em cascata, em que os requisitos iniciais estão claramente definidos, mas o esforço no seu desenvolvimento impede um processo completamente linear, ou seja, combina elementos dos processos lineares e paralelos (Pressman, 2010).

Este modelo permite que o seu próprio desenvolvimento se divida em etapas chamadas incrementos, nas quais cada etapa reproduzirá incrementalmente o sistema até chegar à sua versão final, semelhante ao que acontece no modelo evolutivo em espiral (Pressman, 2010). O modelo incremental diferencia-se da prototipagem pois este tem como objetivo apresentar um produto operacional a cada incremento.

A Figura 7 mostra como em cada incremento é realizado todo o ciclo de desenvolvimento de software, desde a comunicação, planeamento, modelação (análise e design), construção (código, testes), e *deployment*/implementação (entrega e *feedback*).

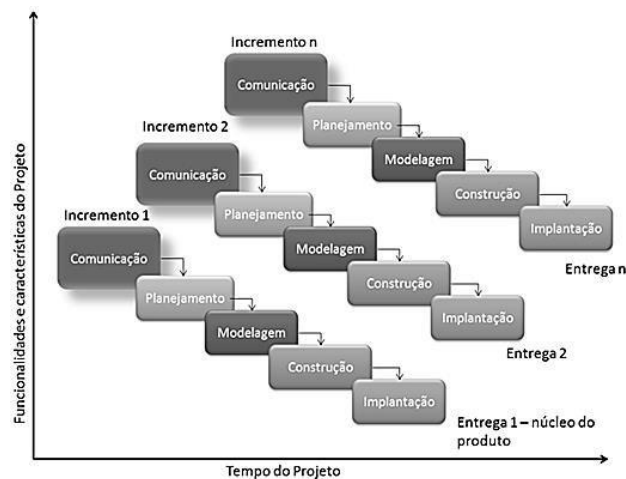


Figura 7 - Modelos de Processo Incrementais (Fonte: Pressman, 2010)

O desenvolvimento de software leva por vezes bastante tempo no seu desenvolvimento, e deste modo pode ser mais fácil e eficaz dividir o trabalho em partes mais pequenas, através de incrementos, e/ou através de várias iterações. Iterações são definidas como fluxos de trabalho e incrementos são crescimentos do próprio produto. Uma iteração resulta num incremento para o software. Um dos princípios do processo incremental e iterativo é que ao longo das iterações a

equipa responsável consiga aperfeiçoar e alargar a sua qualidade, detalhe e o propósito do sistema envolvido.

As suas vantagens passam pela maior possibilidade de avaliar riscos e pontos críticos, podendo assim mitigar e controlar. Caso aconteçam, estes riscos acabam por ser minimizados e reduzidos pois fazem parte apenas de uma iteração, sendo assim reduzidos os custos. O trabalho pode ser realizado por vários intervenientes, através da interação e partilha de conhecimento inerentes.

Modelos de Processo Evolutivos

Modelos evolutivos são iterativos. Estes são caracterizados de uma maneira que permite desenvolver versões cada vez mais completas do software.

Segundo Pressman (2010), existem dois tipos de modelos de processos evolutivos, a prototipagem e o modelo espiral.

- Prototipagem

O modelo de Prototipagem tem o seu surgimento após o modelo em cascata. É um modelo de processo evolutivo que permite apresentar versões cada vez mais próximas do resultado desejado final do software (Pressman, 2010).

Este modelo é usado quando os clientes não têm a certeza dos requisitos pretendidos no software a desenvolver, deste modo só são definidos os requisitos gerais, não existindo o detalhe necessário (Pressman, 2010). Assim, é criado um protótipo que vai auxiliar na interpretação dos requisitos de forma a que o cliente possa entender melhor o pretendido.

Para além de um modelo de desenvolvimento pode ser usado como uma técnica que pode ser implementada no contexto de qualquer modelo de desenvolvimento de software (Figura 8).

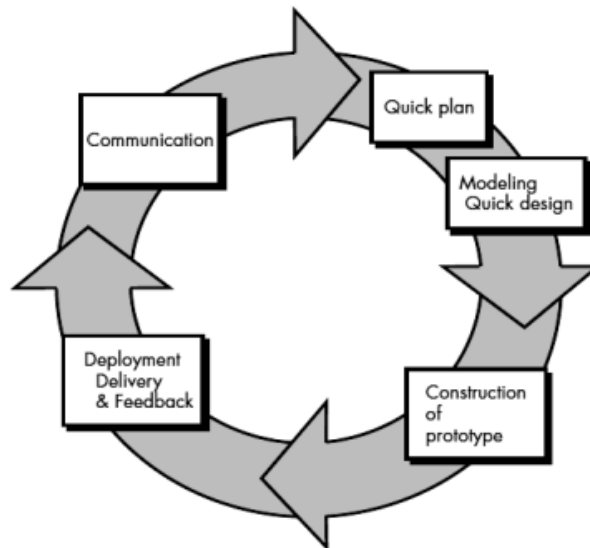


Figura 8 - Modelo de Prototipagem (Fonte: Pressman, 2010)

A prototipagem inicia o seu processo com a comunicação, na qual se definem os requisitos necessários ou pretendidos. É feito rapidamente um planeamento para o protótipo e de seguida produz-se a modelação necessária e as interfaces. O desenvolvimento do protótipo é feito, permitindo o desenvolvimento do software para que o futuro seja um processo mais rápido, e com maior definição.

Na maioria das vezes o protótipo tem como propósito a identificação dos requisitos de software, e servirá como mecanismo para avaliar a usabilidade do sistema, pois numa primeira versão existirão sempre alguns erros a detetar que serão corrigidos mais tarde (Laudon & Laudon, 2012; Pressman, 2010).

Uma das desvantagens da prototipagem é o facto de o cliente poder achar que um protótipo é o produto final e que responde a todos os seus objetivos, o que pode comprometer a qualidade. Apresenta as seguintes vantagens: a compreensão do software e a viabilidade do mesmo, mesmo quando os requisitos iniciais não estão explícitos, o facto de ser possível, através de um pequeno investimento, ter uma versão daquilo que será o sistema no futuro, através da produção de um protótipo pode reduzir-se os custos das fases que se seguirem, e por fim, a viabilidade do sistema também é alcançada através da construção de protótipos (Pressman, 2010).

Pressman (2010) aponta ainda a prototipagem como um ponto eficaz para o software, o segredo está em definir as regras logo no início.

- Espiral

O modelo em espiral foi proposto por Barry Boehm. É um modelo de processo evolutivo de software que combina a interatividade da prototipagem com o controlo e sistematização da cascata, acrescentando o elemento do risco. Com este modelo é possível desenvolver versões de forma rápida e cada vez mais completas do software (Pressman, 2010).

Este modelo vê o desenvolvimento de um sistema informático como um processo contínuo e interativo, na qual é representado através de uma espiral que expressa a realização de várias versões consecutivas desse sistema, de maneira a efetuar o seu aperfeiçoamento do produto final (Serrano, Caldeira & Guerreiro, 2004).

Apresenta-se na Figura 9 as 4 atividades principais deste modelo: a determinação dos objetivos, a identificação e análise de riscos, onde são avaliadas diferentes alternativas para a implementação do produto, o desenvolvimento e a verificação; e, finalmente, o planeamento, onde o projeto é revisto e depois tomada a decisão se o projeto avança para uma iteração seguinte ou não.

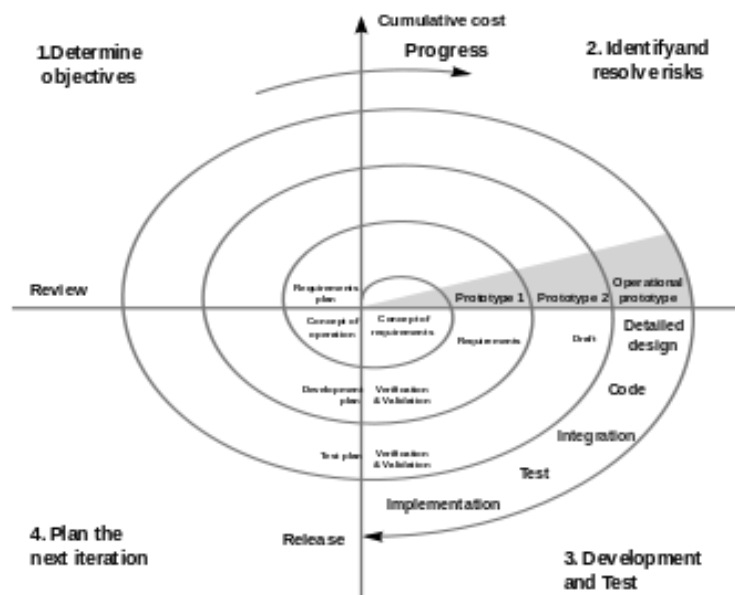


Figura 9 - Modelo Em Espiral (Fonte: <https://www.gratispng.com/png-b8wnan/>)

Uma das principais características deste modelo é o seu foco num processo conduzido por risco (Boehm & Turner, 2003).

Como desvantagens, este modelo apresenta a dificuldade no controlo, coordenação e junção de todas as partes do software, devido a por vezes ser desenvolvido em paralelo entre múltiplas partes

do projeto, cada uma sendo abordada de modo diferenciado, assim é necessário o uso de técnicas específicas para estimar e sincronizar as mesmas.

O modelo espiral tem como vantagem a sua abordagem ao risco, onde permite a percepção e reação aos riscos em cada etapa evolutiva. Outra das principais vantagens, é a evolução sistemática que ao longo de cada iteração permite obter versões cada vez mais completas do software.

RUP

A metodologia de desenvolvimento de software Rational Unified Process (RUP, Rational Software Corporation, 1998) é uma metodologia bastante conhecida, tendo sido adaptada para projetos de qualquer tamanho. Utiliza como boas práticas o desenvolvimento iterativo e o envolvimento do cliente como parte da equipa, permitindo uma boa estruturação do projeto com a diminuição dos riscos, garantindo uma maior qualidade na entrega final.

RUP é um processo de engenharia de software que trabalha através de disciplinas e melhores práticas, aumentando a produtividade no desenvolvimento de software. Esta metodologia foi criada pela Rational Software Corporation, posteriormente adquirida pela IBM. Pode ser aplicada a qualquer tipo de projeto devido a ser bastante customizável (Vasco et al., 2011). Permite uma melhor produtividade individual e colaboração entre a equipa de modo a criar sistemas de alta qualidade que alinham as necessidades de negócio com as tecnologias de informação eficazmente.

O RUP utiliza UML para denotar a sua modelação e algumas das suas principais características são a utilização de diagramas de CDU como base para o processo de desenvolvimento seguir um fluxo de ações de acordo com o *workflow*. Assim, os CDU são especificados, projetados e no fim, são a fonte a partir da qual dirigem o processo de desenvolvimento. Outra das suas características é a sua orientação para a arquitetura, que acaba refletida nos CDU criados. A arquitetura é uma visão do projeto como um todo, que torna visível as características mais importantes do mesmo. RUP tem como característica o seu desenvolvimento iterativo e incremental, por isso é prático que o trabalho seja dividido em pequenos ciclos ou mini-projetos. Cada mini-projeto é uma iteração que resulta em um incremento. Iterações referem-se a passos no *workflow*, e incrementos a evoluções do produto.

RUP é composto por quatro fases (Figura 10): Conceção (visão global do sistema, eliminando riscos e definição do âmbito do projeto), Elaboração (especificação das funcionalidades e desenho da arquitetura), Construção (implementação e teste do sistema) e Transição (distribuição e formação sobre o sistema). Este modelo iterativo é estruturado em duas dimensões.

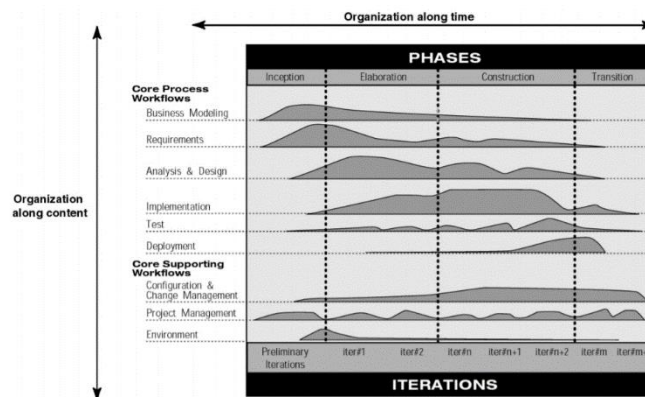


Figura 10 - Modelo Iterativo Fonte: Rational Unified Process, 1998)

As disciplinas são as áreas de conhecimento utilizadas pelo RUP, que no caso da Figura 10, são compostas pelos vários fluxos de processo e suporte.

Relativamente aos fluxos de processo, a Modelação de Negócios tem como principal objetivo avaliar a estrutura e a dinâmica da organização, enumerando os problemas principais. As atividades desta etapa incluem a definição de objetivos de modelação de negócio, avaliação atual do negócio, alinhar os objetivos e desenvolver e estruturar os modelos de CDU para permitir ter uma visão geral da funcionalidade do sistema (Moreira, 2011).

Seguidamente, realiza-se a definição dos requisitos que são estabelecidos com base nas necessidades do sistema. Os requisitos são descritos e devem servir para planear o conteúdo podendo assim definir as interfaces e as delimitações do sistema.

Estando os requisitos bem detalhados e identificados, na Análise e Projeto realiza-se o mapeamento dos requisitos, permitindo a sua transformação para a implementação do sistema. Algumas das suas atividades são a especificação de uma arquitetura robusta e a adaptação para ambiente de implementação. A Implementação define a elaboração dos módulos à execução do código, implementa as classes e objetos em termos de componentes, testa os componentes desenvolvidos como unidades e integra os resultados produzidos (Moreira, 2011).

Os Testes permitem a investigação do código, a fim de encontrar defeitos, *bugs*, permitindo a sua correção e validando todas as funcionalidades implementadas, para aumentar a qualidade do código.

Por fim, o *Deployment* que gera a versão final do produto e permite a formação e suporte aos utilizadores deste novo sistema, vindo de encontro ao plano de aceitação da organização.

Os fluxos de suporte ao desenvolvimento são a configuração e gestão de mudanças que avalia os custos e o impacto nas mudanças no sistema, a Gestão de Projeto que fornece a estrutura para gerir, planear e monitorizar o projeto e o Ambiente que é focado na adaptação ao processo organizacional.

Metodologias Ágeis

As Metodologias Ágeis são definidas por vários autores, tendo as próprias definições alguns aspetos em comum. São definidas por Boehm (2003) como métodos, práticas e técnicas que permitem o desenvolvimento ágil de projetos que tem como objetivo aumentar a satisfação do cliente.

Larman (2002) afirmou que um processo ágil está preparado para responder a mudanças que possam surgir, é então leve e adaptativo. Pode-se definir agilidade como a capacidade de equilibrar a flexibilidade com a estabilidade (Highsmith, 2002).

Trata-se de um processo iterativo onde os vários membros envolvidos no projeto colaboram entre si, proporcionando transparência, e tendo a característica de que estas equipas auto-organizadas, aceitam mudanças que possam surgir (Conboy, 2009).

Os métodos ágeis têm sido uma solução bastante popular e utilizada, pois são vistos como solução para problemas que o método tradicional não consegue resolver. Uma das características do método ágil que o distingue do método tradicional é o facto de este não ser sequencial, ou seja, para começar uma tarefa não é necessário ter terminado a tarefa anterior.

As metodologias ágeis surgiram nos anos 90, mas oficialmente só em 2001, ano onde tiveram os maiores impactos. Foi neste ano que um grupo de profissionais de TI sentiu necessidade de desenvolver uma nova forma de desenvolvimento de software que lhes resolvesse certos problemas. Após se reunirem criaram os princípios e os valores que regem o manifesto ágil, sendo os valores os seguintes (Sommerville, 2011):

- Indivíduos e interações mais do que processos e ferramentas;
- Software funcional mais do que documentação abrangente;
- Colaboração com o cliente mais do que negociação contratual;
- Responder à mudança mais do que seguir um plano.

Os 12 princípios são os seguintes (Agile Manifesto, 2001; Fowler and Highsmith, 2001):

- Prioridade em satisfazer o cliente através da entrega rápida e contínua de software;
- Aceitar alterações de requisitos, mesmo numa fase tardia do ciclo de desenvolvimento;
- Fornecer frequentemente software funcional, com preferência em períodos curtos, numa escala de semanas até meses;
- O cliente e a equipa de desenvolvimento devem trabalhar juntos, diariamente, durante o decorrer do projeto;
- Construir projetos com base em indivíduos motivados proporcionando o ambiente e o apoio de que necessitam;
- O método mais eficiente e eficaz de passar informação para dentro de uma equipa de desenvolvimento é através de conversa pessoal e direta;
- Software funcional é a medida principal de progresso;
- Processos ágeis promovem um ambiente sustentável;
- A atenção permanente à excelência técnica e um bom desenho da solução aumentam a agilidade;
- Simplicidade: a arte de maximizar a quantidade de trabalho que não é feito;
- As melhores arquiteturas, requisitos e designs emergem de equipas auto-organizadas;
- A equipa reflete regularmente sobre o modo de se tornar mais eficaz, fazendo os ajustes e adaptações necessárias.

Nas metodologias ágeis existem diferentes métodos de aplicações para o desenvolvimento de software. Segundo Pressman e Sommerville as metodologias Extreme Programming (XP), Scrum, Adaptative Software Development (ASD), Crystal, Lean e Kanban são as que mais se destacam dentro das metodologias ágeis, sendo também as mais relevantes para o projeto.

Extreme Programming (XP)

Defendida por Beck e Fowler (2000) e estando dentro das metodologias ágeis, XP é uma metodologia que serve para o desenvolvimento de projetos na área de TI baseados em requisitos básicos que sofrem constantes e rápidas modificações. O objetivo do XP é desenvolver softwares mais eficientes

e simples de forma mais rápida e económica que no final obtenham a máxima satisfação por parte do cliente. O seu desenvolvimento é conduzido por quatro valores: comunicação, coragem, simplicidade e *feedback* (Beck & Fowler, 2000).

A comunicação tem um papel crucial no desenvolvimento do software, na medida em que proporciona a criação de colaboração informal entre o cliente e os *developer's*, para assim ser mais fácil atingir a satisfação do cliente (Pressman, 2010).

A simplicidade, tal com o nome indica, visa desenvolver o trabalho de forma simples e acessível evitando colocar informações desnecessárias. O *feedback* permite que erros ou incoerências sejam detetados antes da chegada ao produto final, permitindo também ao cliente sugerir novas informações e formas a quem está a desenvolver. Por fim, e não menos importante, a coragem é fundamental para conseguir a implementação dos três anteriores valores.

Uma das características do XP é o foco nas necessidades imediatas, em vez de olhar para necessidades futuras. A metodologia XP usa uma perspetiva orientada a objetos e tem foco em resolver e abordar a necessidade do presente, englobando um conjunto de regras e práticas que se traduzem no decorrer de quatro atividades: planeamento, design, codificação e teste. A Figura 11 demonstra a produção do incremento do sistema do processo XP.

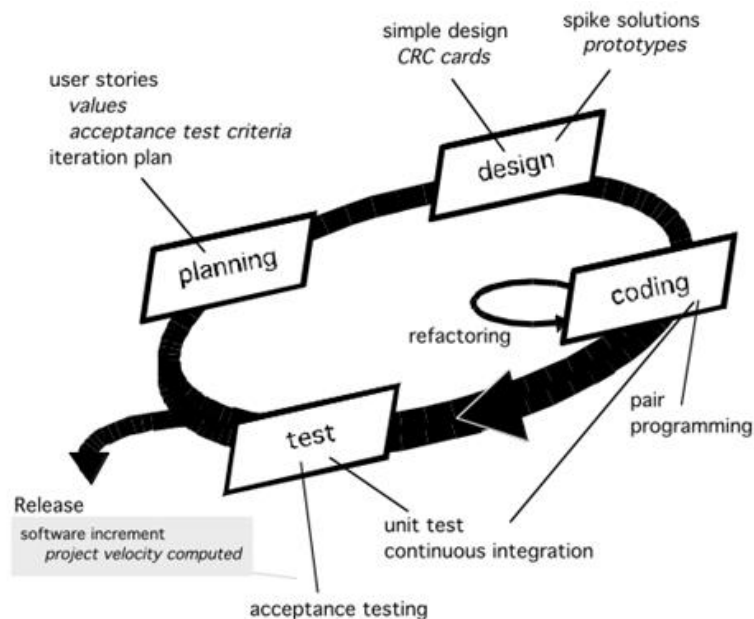


Figura 11 – Ciclo no processo XP (Fonte: Pressman, 2010)

As fases presentes no Ciclo do Processo XP, apresentadas na Figura 11, são descritas com mais detalhe de seguida.

1. Planeamento

A fase do planeamento consiste em definir a melhor forma de realizar o pretendido assim como definir as prioridades, deste modo planeamento é onde são definidos os conjuntos de atividades para a definição de requisitos, com o objetivo de que os membros do projeto compreendam o contexto, e tenham uma ideia ampla das necessidades e funcionalidades que serão implementadas primeiro e as que apenas serão realizadas mais tarde (Pressman, 2010).

2. Design

A fase do design é caracterizada pela sua simplicidade, na qual a metodologia assenta em designs simples que dêem resposta a necessidades presentes e não futuras. Este tipo de design tem como objetivo a sua fácil implementação.

3. Codificação

A codificação consiste na implementação do software, para tal XP defende uma prática característica chamada de programação em pares. Tal como o nome indica, a programação em pares consiste no desenvolvimento de software por duas pessoas através de um só computador onde podem alternar os papéis, ou seja, enquanto um está a escrever o código o outro analisa a estratégia a usar. Tendo como vantagens serem identificados erros mais facilmente, aprendem um com o outro e o trabalho é realizado mais rapidamente.

4. Teste

A fase de testes tem como objetivo perceber se a solução está de encontro aos requerimentos do projeto, deste modo serão executados testes aos recursos e às funcionalidades do software.

A principal ferramenta para testar o software utilizado pela XP é o *Unit Test* que permite que cada especificidade implementada seja testada ao longo da sua implementação.

Para concluir, o XP permite obter várias versões do produto final através de interações curtas para assim o cliente ir analisando como está a decorrer o seu desenvolvimento e poder fazer comentários e criar opiniões para assim melhorar o produto final. O projeto realizado por XP é flexível à mudança o que tem como vantagem a diminuição dos seus custos associados.

SCRUM

Scrum é das metodologias ágeis de gestão mais utilizadas para criar e desenvolver vários tipos de sistemas, prova disso é que a maioria dos programadores de software utilizam esta metodologia de forma preferencial para o seu trabalho. É considerado um processo iterativo e incremental (Sommerville, 2011).

Scrum é um processo usado preferencialmente para projetos orientados a objetos e que se focam nas pessoas e onde existam ambientes onde podem ocorrer mudanças bruscas (Schwaber & Beedle, 2001).

Dentro de cada atividade da estrutura, as tarefas de trabalho ocorrem dentro de um padrão de processo denominado sprint.

Os projetos numa metodologia Scrum são divididos em ciclos, que ocorrem normalmente mensalmente, designados como Sprint. Por sua vez um sprint é um conjunto de unidades de trabalho (formadas por 7 indivíduos) necessárias para alcançar os requisitos preestabelecidos, definidos no início de cada sprint, e que devem ser realizadas num tempo não superior a 30 dias.

As sprints são adaptadas ao problema e são definidos e modificados no imediato pela equipe de Scrum. O processo do desenvolvimento do sprint é realizado dentro do ciclo de vida do Scrum. O fluxo geral do processo Scrum é definido na Figura 12.

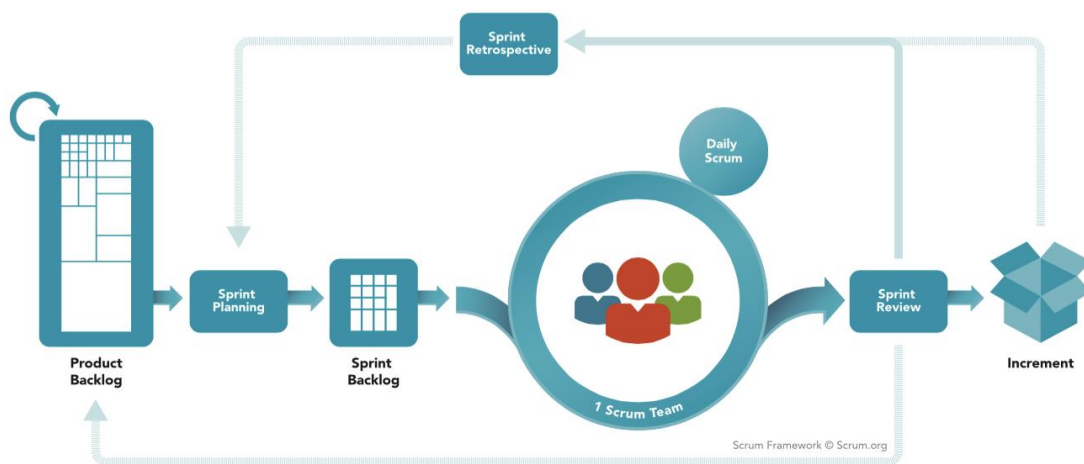


Figura 12 - Processo de Desenvolvimento Scrum. (Fonte: Scrum Portugal)

Com base na Figura 12, *backlog* consiste numa lista de requisitos por ordem de prioridade conforme o valor comercial que criem ao cliente. Existem dois tipos de *backlog*, o *product backlog* onde estão

presentes os requisitos gerais e os *sprints backlog* onde está a lista de requisitos que a equipa se compromete a resolver.

Um *team leader*, denominado *scrum master* é responsável por liderar a equipa e coordenar todas as reuniões da mesma, reuniões estas onde se ajuda a resolver problemas e também a evitá-los.

No final de cada *sprint* é feita uma demonstração das funcionalidades implementadas. No decorrer de cada novo ciclo estas funcionalidades consoante o resultado podem ser adicionadas, apagadas ou alteradas ao *backlog* do produto. Sendo novamente analisado e planeado o que seja executado durante o novo ciclo de *sprint* através da definição do *sprint backlog*.

Adaptative Software Development (ASD)

James Highsmith propôs o método ASD, como uma técnica para criar sistemas e softwares complexos.

Esta metodologia ágil tem foco nas pessoas, incentivando a colaboração entre as mesmas, sendo possível através dela chegar a resultados que não seriam possíveis de obter caso se trabalhasse de forma individual, pelo menos com a brevidade desejada. Este método é utilizado para projetos em que tendem a ocorrer muitas mudanças e de forma repentina no dia-a-dia.

Highsmith aponta que o desenvolvimento do software é feito de acordo com um ciclo de vida, que se divide em 3 fases, sendo elas: especulação, colaboração e aprendizagem (Pressman, 2010), como se pode ver na Figura 13.

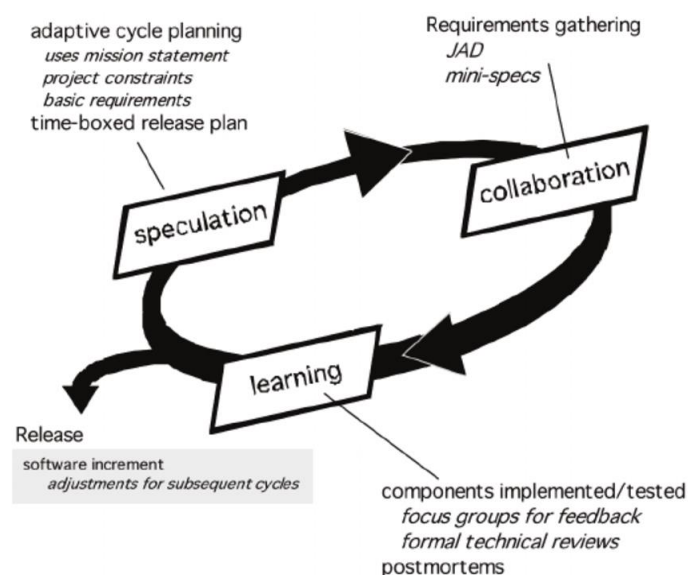


Figura 13 - Fases de desenvolvimento de software adaptável (Fonte: Pressman, 2010)

Definindo agora cada uma das fases, especulação é quando um projeto é iniciado e é feito o planeamento do ciclo (Pressman, 2010), ou seja, onde é realizado o calendário de projeto, onde está inscrito a data de entrega ou descrições dos utilizadores; definidos os requerimentos de cada interação do projeto e declaração da missão, onde o cliente diz o que pretende do projeto.

A seguinte fase é a colaboração onde se procura que as equipas de trabalho colaborem entre si para chegarem a objetivos que desta forma se tornam mais fáceis e claros, aqui é importante que haja bastante comunicação entre todos. Através da colaboração é possível alcançar melhores e mais eficientes resultados ao invés de quando esta não existe (Pressman, 2010).

Segue-se a aprendizagem, onde ao longo do ciclo adaptativo o objetivo é que os desenvolvedores adquiram conhecimentos, para desta forma completar o ciclo. Existem três formas de aprendizagem: grupos focais; análises técnicas e *post-mortems* (Pressman, 2010).

Quando as equipas utilizam este método ágil, onde devem existir equipas auto-organizadas, colaboração interpessoal e aprendizagem individual e em equipa, tem mais probabilidade de alcançar o sucesso (Pressman, 2010).

Crystal

Um outro método de desenvolvimento de software ágil, denominado Crystal, foi desenvolvido por Alistair Cockburn e James Highsmith. É uma metodologia que se foca nas pessoas e nas suas interações, acreditando que as habilidades e talentos e habilidades das pessoas têm um importante e grande impacto nos projetos.

Os autores defendem este método como diferente de outros métodos ágeis, como o XP, devido a este método ser uma família de metodologias, em que se adaptam as características do projeto a desenvolver (Pressman, 2010), e não uma única metodologia (Cockburn, 2004).

Cockburn (2004) agrupou as diversas metodologias da família Crystal por diferentes cores e dureza, tal como se faz para classificar os minerais. Em que as cores correspondem ao número de indivíduos presentes no projeto (Figura 14).

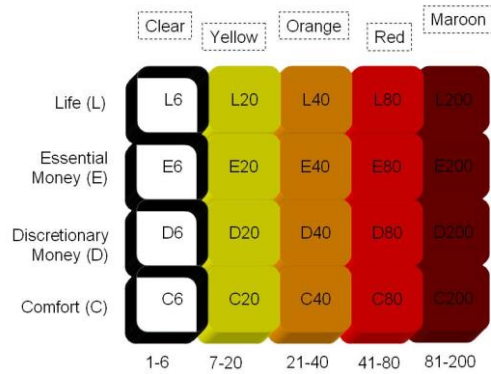


Figura 14 - A família de metodologias Crystal (Fonte: <http://www.devx.com/architect/article/32836/0/page/2>)

Para concluir qual a abordagem que mais se adequa a cada projeto, Cockburn apresenta três dimensões: o tamanho da equipa; qual a prioridade do projeto e criticidade. Os métodos de Crystal estão focados em: Pessoas, Interação, Comunidade, Habilidades, Talentos, Comunicações.

Lean Software Development

O desenvolvimento de software Lean é adaptado pelos princípios da manufatura Lean originária do Japão para o mundo da engenharia de software. É um conjunto de princípios para fornecer software que se baseia nos princípios da manufatura industrial, que tem como objetivo o alcance de aumento do crescimento rentável (aumentando a qualidade e reduzindo custos), do ritmo de inovação e na satisfação dos clientes.

Este método tem como objetivo reduzir os desperdícios e resíduos, que segundo Poppendieck e Poppendieck se pode fazer de acordo com os seguintes princípios: construir qualidade, criar conhecimento, adiar compromisso, entregar mais rápido, respeitar pessoas e otimizar o todo. (Poppendieck e Poppendieck, 2003, cit in Pressman, 2010). São considerados resíduos o trabalho parcialmente concluído, processos extras, recursos extras, troca de tarefas e defeitos existentes, ou seja, tudo o que não está a ser utilizado da melhor forma e onde é possível tirar o máximo proveito, é considerado um resíduo.

É papel dos gestores e líderes assegurarem a melhor forma de reduzir estes desperdícios, como identificar quais as tarefas prioritárias; identificar e eliminar aquilo que atrasa os processos; fornecer suporte técnico para que as equipas possam realizar as entregas de forma contínua e com o máximo de qualidade; eliminar ruídos na comunicação entre pessoas.

A abordagem Lean proporcionou uma maior eficiência no desempenho das empresas e uma redução no desperdício, o que é bastante positivo. Atualmente esta abordagem é utilizada na maioria das indústrias e em vários setores organizacionais.

Dynamic Systems Development Method

Atualmente as exigências do mercado levam a que num curto espaço de tempo se entregue um software final, logo o processo deve ser ágil, para assim responder a estas necessidade de qualidade e tempo.

Dynamic Systems Development Method (DSDM) é um dos métodos ágeis, neste caso um processo de software interativo, que fornece uma forma de ver, através de desenho, o sistema para que possa ser contruído e mantido quando existem prazos curtos de entrega. O seu desenvolvimento tem como características a utilização da prototipagem incremental, que é um método de desenvolvimento rápido. Neste método é apenas necessário algum trabalho no incremento que está a ser realizado para se passar para o próximo, os detalhes de cada podem ser analisados no fim, quando se conhecerem mais requisitos e forem sugeridas alterações.

DSDM possui quatro ciclos iterativos para o seu ciclo de vida, processados por duas atividades adicionais do ciclo de vida, como se pode ver na Figura 15:

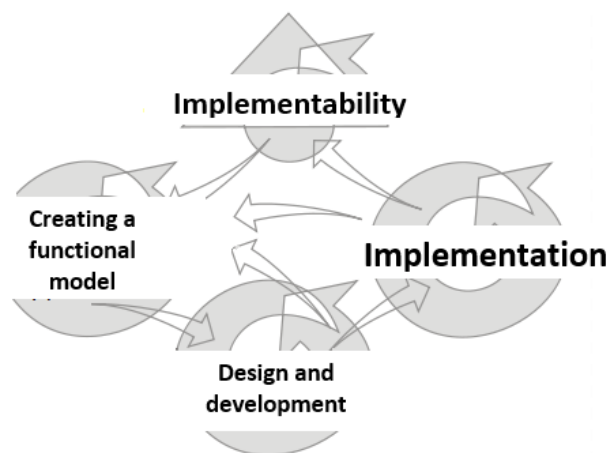


Figura 15 - Modelo do processo DSDM de Desenvolvimento de Software Fonte: Processdata-DSDM.png).

Analisando cada um dos ciclos, a viabilidade tem como característica avaliar a viabilidade com base nos requisitos funcionais e as suas informações, por sua vez a criação do modelo funcional é onde são construídos os protótipos que permitem aos clientes analisar a sua funcionalidade. Um outro

ciclo é o desejo e desenvolvimento, onde se averigua se o que foi feito no ciclo anterior está de acordo com as exigências do cliente. Por fim a implementação, tal como o nome indica é onde se implementa o software final no ambiente operacional.

Kanban

Kanban é um outro método ágil, que tem origem japonesa e é atualmente considerado o método mais simples e eficaz para gerir projetos. Permite ajudar as organizações a visualizar, planear e organizar o seu trabalho através da criação de um quadro ou tabela com os vários elementos existentes para o desenvolvimento do software.

Este método requer a existência de comunicação e transparência nos membros da equipa, de modo a haver progresso nas tarefas e no projeto. Os membros poderão consultar a qualquer momento o status do projeto, bem como aquilo que já foi desenvolvido e o que está por desenvolver.

David J. Anderson (2010) apresenta seis princípios base do Kanban sendo eles: visualização do trabalho através de um quadro que representa os passos e tarefas para o desenvolvimentos do trabalho; limitar o trabalho em desenvolvimento de maneira a ser mais eficaz tendo a ideia de base “mais fazendo menos”, ou seja, não dar mais tarefas à equipa sem esta ter terminado as anteriores tarefas; gerir o fluxo; ter uma melhor colaboração; criar critérios através de políticas de gestão explícitas.

Este método tem vantagens, como conseguir ver todas as tarefas que existem num projeto e o seu estado de execução (completo, em progresso, em teste e terminado). É também possível definir as tarefas mais importantes a realizar em certa fase do desenvolvimento, permitindo assim ter uma ferramenta para gerir o projeto.

4. DESENVOLVIMENTO DO PROJETO

4.1. METODOLOGIA DE DESENVOLVIMENTO

A decisão da metodologia de desenvolvimento a ser utilizada no projeto passou por uma análise das metodologias tradicionais e ágeis mais conhecidas no mercado. Foi assim necessário avaliar o tipo de projeto e as suas particulares características.

O desenvolvimento deste projeto é composto por apenas um programador que irá ter contacto com o cliente. Sendo apenas um programador, implementar uma metodologia ágil no seu todo pode tornar-se algo um pouco complexo e desnecessário, sendo que os processos ágeis são bastante orientados ao trabalho em equipa. Deste modo com base nas características estudadas nas diferentes metodologias descritas na seção das metodologias de software, tomou-se a decisão de retirar algumas das principais características nas metodologias ágeis, que mais se adequam e permitem sustentar este projeto, sendo elas:

- A realização de um desenvolvimento orientado aos testes (Código, Teste, Código, Teste);
- O Desenvolvimento é dividido em pequenos *sprints*/iterações bem definidas;
- Ter bastante contacto com o Cliente;
- Criar uma lista de itens que são necessários concluir (*Product Backlog*);
- Criar e visualizar um quadro com as tarefas e itens (Retrospectivas);
- Priorizar conforme necessidade.

Com base nestes pontos, foi decidido utilizar a metodologia Kanban, mais especificamente a Personal Kanban. O Personal Kanban é uma abordagem simples para gerir e organizar o trabalho, que ajuda a pessoa a aumentar a produtividade, tendo sempre presente as prioridades e o planeamento do que deve ser feito no presente e no futuro.

As características do Personal Kanban permitem:

- Visualizar o trabalho num quadro de forma a facilitar o seu planeamento, focar as prioridades e entregar na hora certa;
- A análise dos resultados do trabalho com métricas de desempenho que suportam o seu aprimoramento;
- A implementação do método Kanban permite economizar tempo na medida em que combate a tendência para fazer várias tarefas ao mesmo tempo, tentando limitar as tarefas em andamento (*Work in Progress*).

Assim sendo a utilização destas características selecionadas dos processos ágeis, juntamente com o Personal Kanban permitirá encontrar um processo que funcione e permita adaptar a novas circunstâncias. Permitindo assim agregar valor ao cliente regularmente, independentemente de o software estar ou não finalizado. Os processos serão iterativos e o trabalho será feito ao longo de várias iterações. Os testes de unidade serão seguidos, como nas metodologias ágeis, fornecendo uma indicação sobre se o software está a funcionar corretamente.

Esta escolha permitirá construir todo o sistema e executar todos os testes com frequência e fornecer pedaços de código de trabalho no final de cada iteração, tornando assim todo o processo ágil.

Como se pode ver a metodologia utilizada não será baseada apenas em algo único e específico, mas sim num conjunto de características metodológicas que se complementam entre si, possível através da pesquisa previamente realizada sobre as várias metodologias.

4.2. LEVANTAMENTO E ANÁLISE DE REQUISITOS FUNCIONAIS

O levantamento e análise de requisitos é um passo fundamental na construção de um SI. Os requisitos de software permitem que existam condições para o utilizador resolver algum problema ou até mesmo atingir certo objetivo. Os mesmos são caracterizados como as necessidades, exigências e desejos que o cliente pretende para o sistema. Este passo é algo bastante importante pois qualquer que seja o projeto que tenha requisitos bem definidos isso proporcionará o desenvolvimento de um sistema mais claro, coeso e mais próximo de alcançar a satisfação do cliente dentro do prazo de tempo previsto.

O levantamento e a análise iniciaram-se através da identificação dos problemas existentes, da identificação dos principais elementos envolventes ao sistema e na procura de soluções para os mesmos. Foram identificados um conjunto base de requisitos necessários para compor este SI, através de reuniões e entrevistas com o departamento de RH:

- A necessidade de ocupação de um cargo que resultará num processo de recrutamento;
- Um processo de recrutamento surge da necessidade de recursos para um determinado projeto;
- A fase inicial do recrutamento resultará numa lista de candidaturas para um processo de recrutamento, existindo duas formas de o realizar:
 - a. Recrutamento Interno: Verifica se existem colaboradores internos disponíveis e se não existirem, é realizado um recrutamento externo.

b. Recrutamento externo: é realizado um Processo Geral de Candidaturas através de anúncios e recomendações ou opta-se por um Processo de Subcontratação realizado externamente;

- A Candidatura tem um processo de avaliação através de entrevistas;
- Um Processo de Avaliação com sucesso resulta num Processo de Contratação;
- Um Processo de Contratação aceite e aprovado resulta numa contratação e por sua vez resulta em menos um candidato necessário para a continuação de um processo de recrutamento.

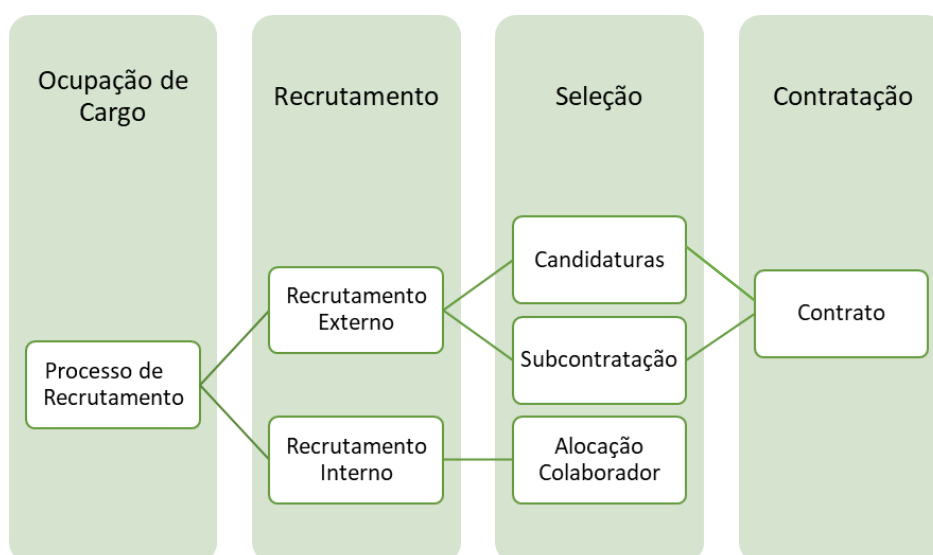


Figura 16 - Esquema geral de requisitos

4.2.1. Requisitos Funcionais

Um conjunto de requisitos funcionais são especificados através da tabela que se segue e encontram-se em notação UML, para posteriormente se construir o Diagrama de Classes e o Modelo de Dados.

Tabela 1 - Tabela com Requisitos Funcionais do SI

Código	Nome	Descrição
UC-01	Efetuar Login	Necessidade de Autenticação de Utilizadores dentro da Aplicação.
UC-02	Criar Projetos	Necessidade de se poder criar novos projetos. - É necessário

		<p>preencher os seguintes dados:</p> <p>Nome do Projeto; Responsável; Localização; Principais objetivos; Data de Início; Data de Fim.</p>
UC-03	Editar Projetos	Necessidade de opção para alteração de dados relativos a projetos já existentes.
UC-04	Ver Projetos	Necessidade de ver e consultar os projetos existentes.
UC-05	Criar Processos de Recrutamento	Um processo de recrutamento parte sempre da necessidade de recursos para um determinado projeto, sendo assim necessária a criação de um processo de recrutamento.
UC-06	Ver Processos de Recrutamento	Necessidade de ver e consultar os Processos de Recrutamento existentes.
UC-07	Terminar Processos de Recrutamento	Necessidade de opção para terminar Processos de Recrutamento, de maneira a que os mesmos possam ser concluídos.
UC-08	Criar Candidaturas	Necessidade de criar uma Candidatura relativa a um Candidato. A Candidatura está associada a um Processo de Recrutamento.
UC-09	Ver Candidaturas	Necessidade de ver e consultar Candidaturas existentes.
UC-10	Avaliar Candidaturas	Necessidade de Aprovar/Rejeitar candidaturas que existam.
UC-11	Editar Candidaturas	Necessidade de opção para alteração de dados nas candidaturas, em certos campos relativos ao candidato.
UC-12	Marcar Entrevistas	Necessidade de marcação de entrevistas.
UC-13	Ver Entrevistas	Necessidade de ver entrevistas realizadas e por realizar.
UC-14	Avaliar Entrevistas	Necessidade de avaliar entrevistas (Aprovar/Rejeitar).
UC-15	Reagendar Entrevistas	Necessidade de Reagendar entrevistas.
UC-16	Criar Pedido de Contratação	Necessidade de criar Pedido de Contratação.

UC-17	Ver Pedido de Contratação	Necessidade de ver Pedidos de Contratação.
UC-18	Avaliar Pedido de Contratação	Necessidade de Aprovar/ Rejeitar Pedido de Contratação.

4.2.2. Diagramas

Diagrama Caso de Uso

O caso de uso tem a função de descrever as funcionalidades que o sistema irá necessitar e contém a respectiva descrição do mesmo. Para tal foram definidos um conjunto de atores que estão interrelacionados com as funcionalidades a ser desenvolvidas, a Figura 17 descreve os diferentes atores existentes no sistema.

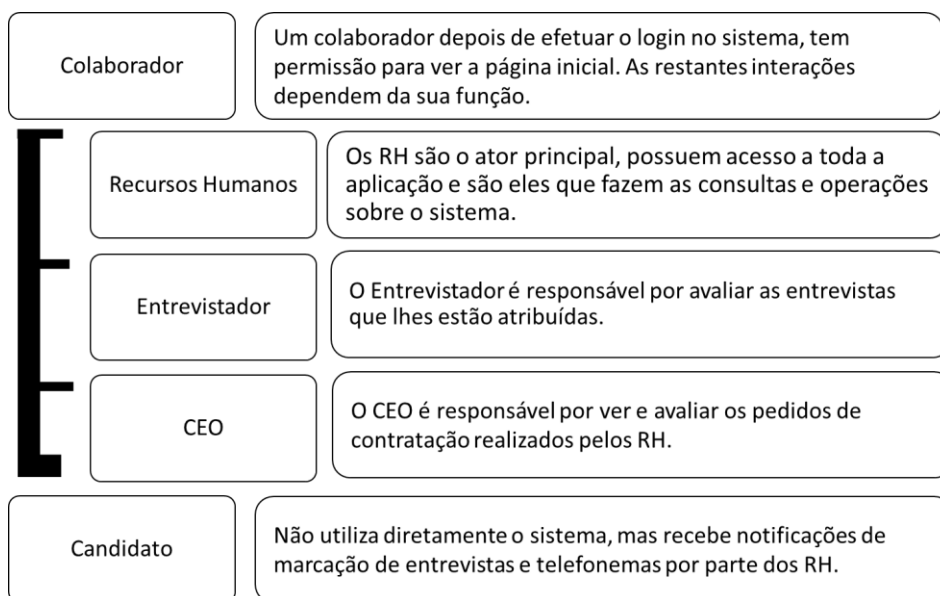


Figura 17 - Descrição de Atores no sistema

O diagrama de caso de uso representado na Figura 18 permite obter uma visão geral do sistema e descreve sumariamente as funcionalidades e atividades requeridas, tendo ligação com o diagrama de atividade geral representado na Figura 23.

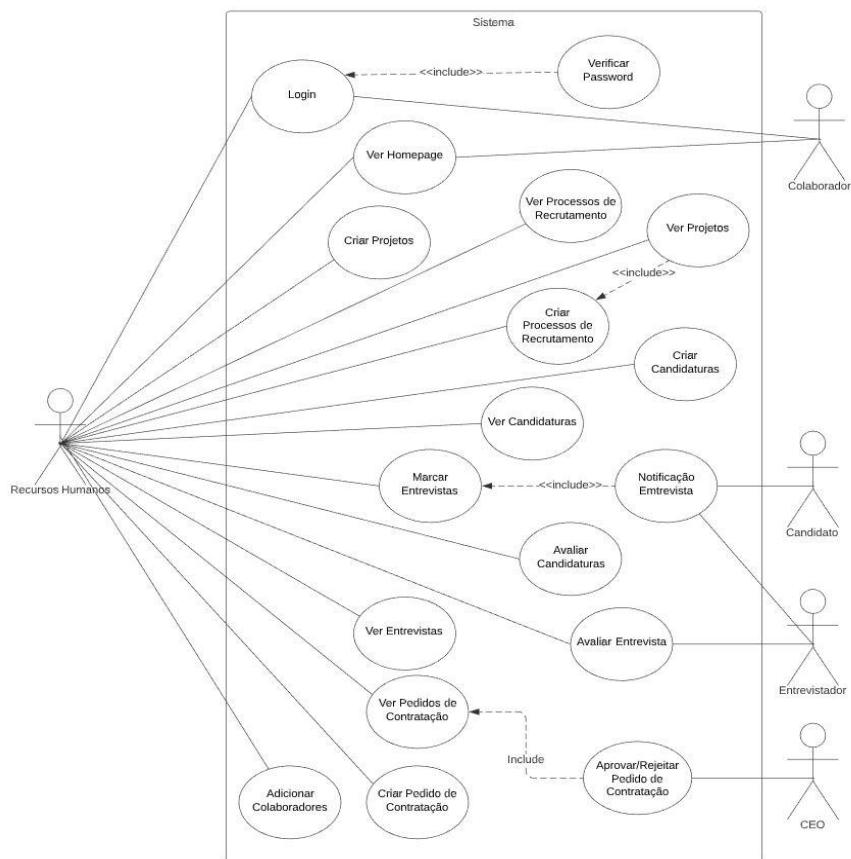


Figura 18 - Caso de Uso do SI (Fonte: Elaboração própria)

Os CDU foram agrupados em quatro principais grupos de processos de maneira a explicar detalhadamente cada processo:

- Processo de Recrutamento – Este processo inicia na procura de recursos com base na definição de um perfil de candidato para colmatar uma necessidade num determinado projeto;
- Processo de Candidaturas – Este processo passa pela recolha e registo de diversas candidaturas, cada candidatura tem informação sobre o candidato e tem um processo de recrutamento associado;
- Processo de Seleção – Este processo têm como objetivo escolher os melhores candidatos. As entrevistas e a sua avaliação permitem conhecer melhor cada um dos candidatos e o seu potencial. Os mesmos apoiam os processos de decisão e seleção do melhor candidato;
- Processo de Contratação – Este processo incide sobre a contratação de um novo recurso e a sua assinatura de contrato que conclui com a integração de um novo colaborador e na diminuição de necessidade de recursos para um determinado projeto.

Seguidamente são apresentados e detalhados os CDU do SI, permitindo assim descrever as diferentes funcionalidades requeridas no sistema.

- Processo de Recrutamento

Tabela 2 - Caso de Uso e Fluxo de Eventos relativos ao Processo de Recrutamento

Diagrama de Caso de Uso - Processo de Recrutamento		
Descrição	O utilizador pode ver, criar e terminar processos de recrutamento. Os processos de recrutamento têm sempre um projeto e competências associadas.	
Atores	Recursos Humanos	
Fluxo de Eventos		
Código	Ações do Ator	Ações do Sistema
UC-05	Criar Processos de Recrutamento e adicionar o projeto e as competências ao processo	Adiciona os registos às tabelas com a informação preenchida. Adiciona as ligações entre as diferentes tabelas.
UC-06	Ver Processos de Recrutamento	Pesquisa através de uma consulta os processos de recrutamento existentes.
UC-07	Avaliar Processos de Recrutamento	Define o registo como o utilizador pretende (por exemplo coloca como processo de recrutamento terminado).

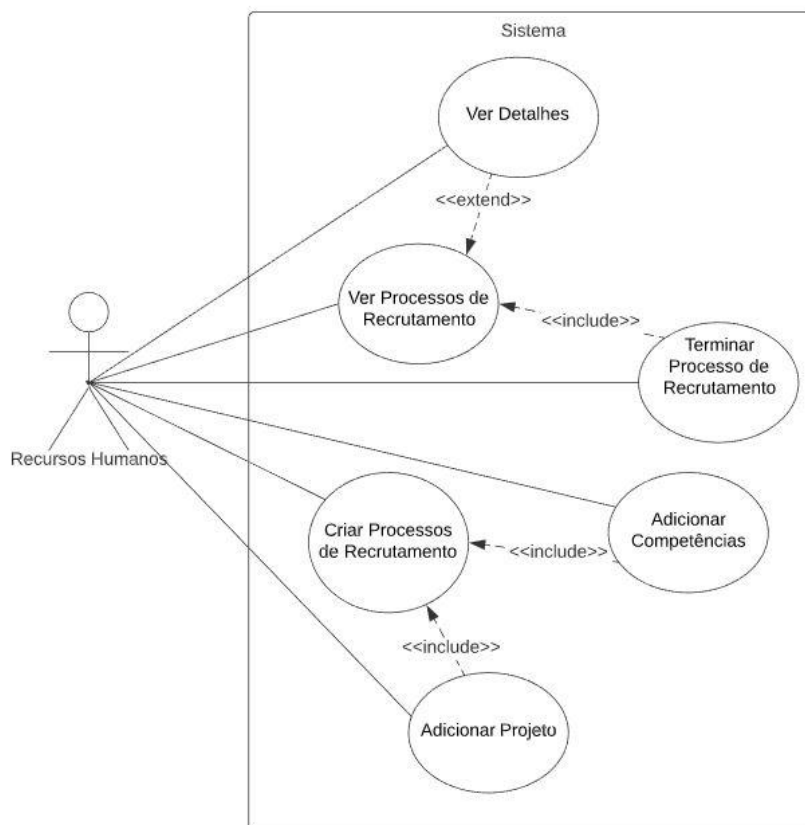


Figura 19 - Caso de Uso relativo ao Processo de Recrutamento (Fonte: Elaboração própria)

- Processo de Candidaturas

Tabela 3 - Caso de Uso e Fluxo de Eventos relativos ao Processo de Candidaturas

Diagrama de Caso de Uso - Processo de Candidaturas		
Descrição	O utilizador pode ver, criar, editar e avaliar candidaturas.	
Atores	Recursos Humanos	
Fluxo de Eventos		
Código	Ações do Ator	Ações do Sistema
UC-08	Criar Candidatura e adicionar o processo e as competências do candidato	Adiciona os registos às tabelas com a informação preenchida. Adiciona as ligações entre as diferentes tabelas.
UC-09	Ver Candidaturas	Pesquisa através de uma consulta as candidaturas existentes.
UC-10	Avaliar Candidaturas	Define o registo como o utilizador pretender (por exemplo exclui a candidatura).
UC-11	Editar Candidaturas	Guarda as alterações efetuadas na respetiva tabela

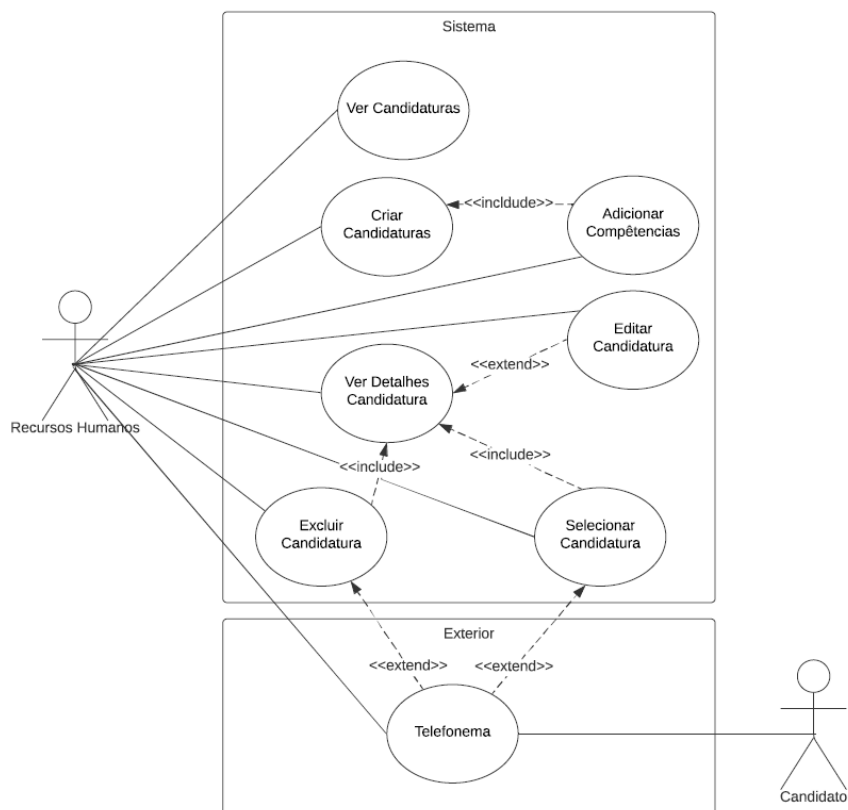


Figura 20 - Caso de Uso relativo ao Processo de Candidaturas (Fonte: Elaboração própria)

- Processo de Seleção

Tabela 4 - Caso de Uso e Fluxo de Eventos relativos ao Processo de Seleção

Diagrama de Caso de Uso - Processo de Seleção		
Descrição	O utilizador pode marcar, reagendar, ver e avaliar entrevistas.	
Atores	Recursos Humanos, Entrevistador e Candidato	
Fluxo de Eventos		
Código	Ações do Ator	Ações do Sistema
UC-12	Marcar Entrevista	Adiciona um registo à tabela com a informação preenchida. Adiciona as ligações entre as diferentes tabelas.
UC-15	Reagendar Entrevista	Guarda as alterações efetuadas na respetiva tabela.
UC-13	Ver Entrevistas	Pesquisa através de uma consulta as entrevistas existentes.
UC-14	Avaliar Entrevistas	Uma entrevista é avaliada através de um conjunto de questões onde são registadas as respostas do candidato, no final esta informação é adicionada a um registo na tabela, sendo incluída a avaliação da entrevista e a respetiva observação.

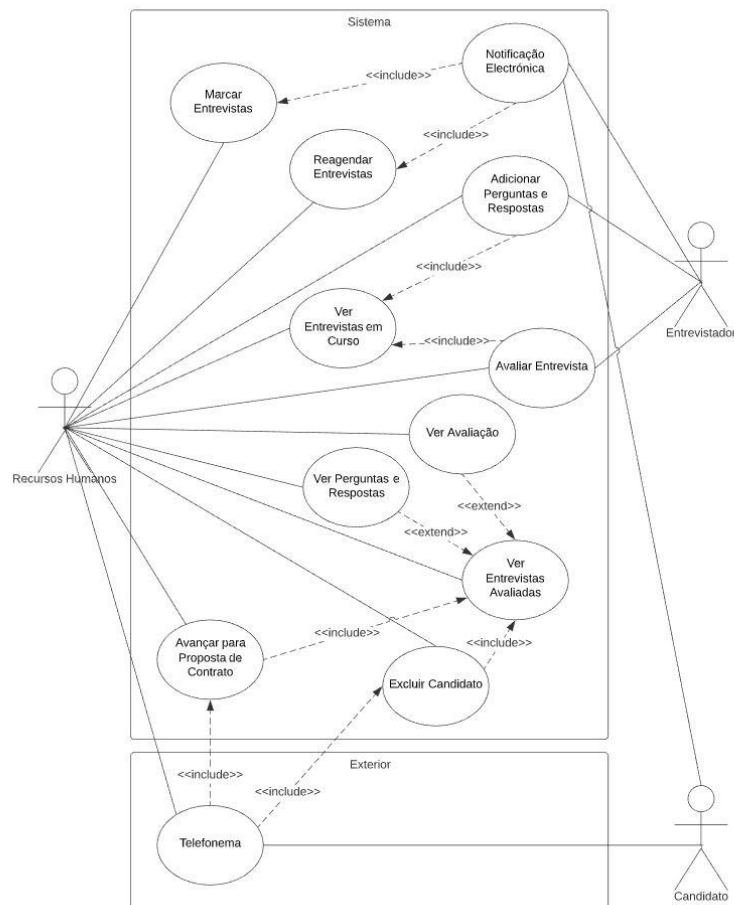


Figura 21 - Caso de Uso relativo ao Processo de Seleção (Fonte: Elaboração própria)

- Processo de Contratação

Tabela 5 - Caso de Uso e Fluxo de Eventos relativos ao Processo de Contratação

Diagrama de Caso de Uso - Processo de Contratação		
Descrição	O utilizador pode criar, ver e avaliar Pedidos de Contratação para um candidato.	
Atores	Recursos Humanos e CEO	
Fluxo de Eventos		
Código	Ações do Ator	Ações do Sistema
UC-16	Criar Pedido de Contratação	Adiciona um registo à tabela com a informação preenchida. Adiciona as ligações entre as diferentes tabelas.
UC-17	Ver Pedido de Contratação	Pesquisa através de uma consulta os Pedidos de Contratação existentes e ao seleccionar é possível ver todos os detalhes.
UC-10	Avaliar Pedido de Contratação	Define o registo como o utilizador pretender (por exemplo aprova o contrato). Permite desta forma avaliar rejeitando ou aprovando o contrato.

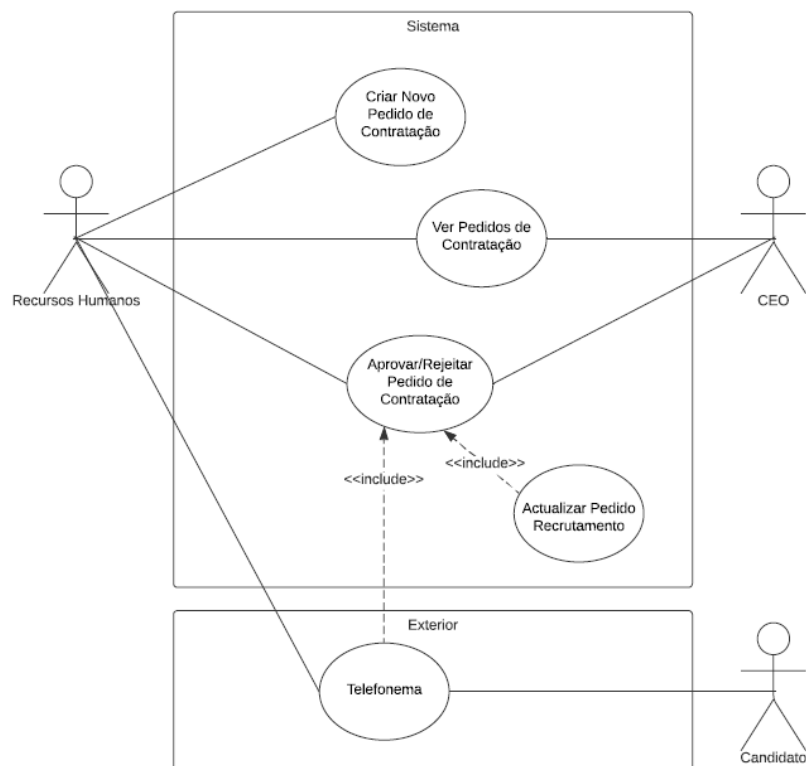


Figura 22 - Caso de Uso relativo ao Processo de Contratação (Fonte: Elaboração própria)

Diagrama de Atividade

No diagrama de atividade representado na Figura 23, é possível visualizar e compreender como é realizado todo o fluxo dentro do sistema e como os diferentes processos se interligam.

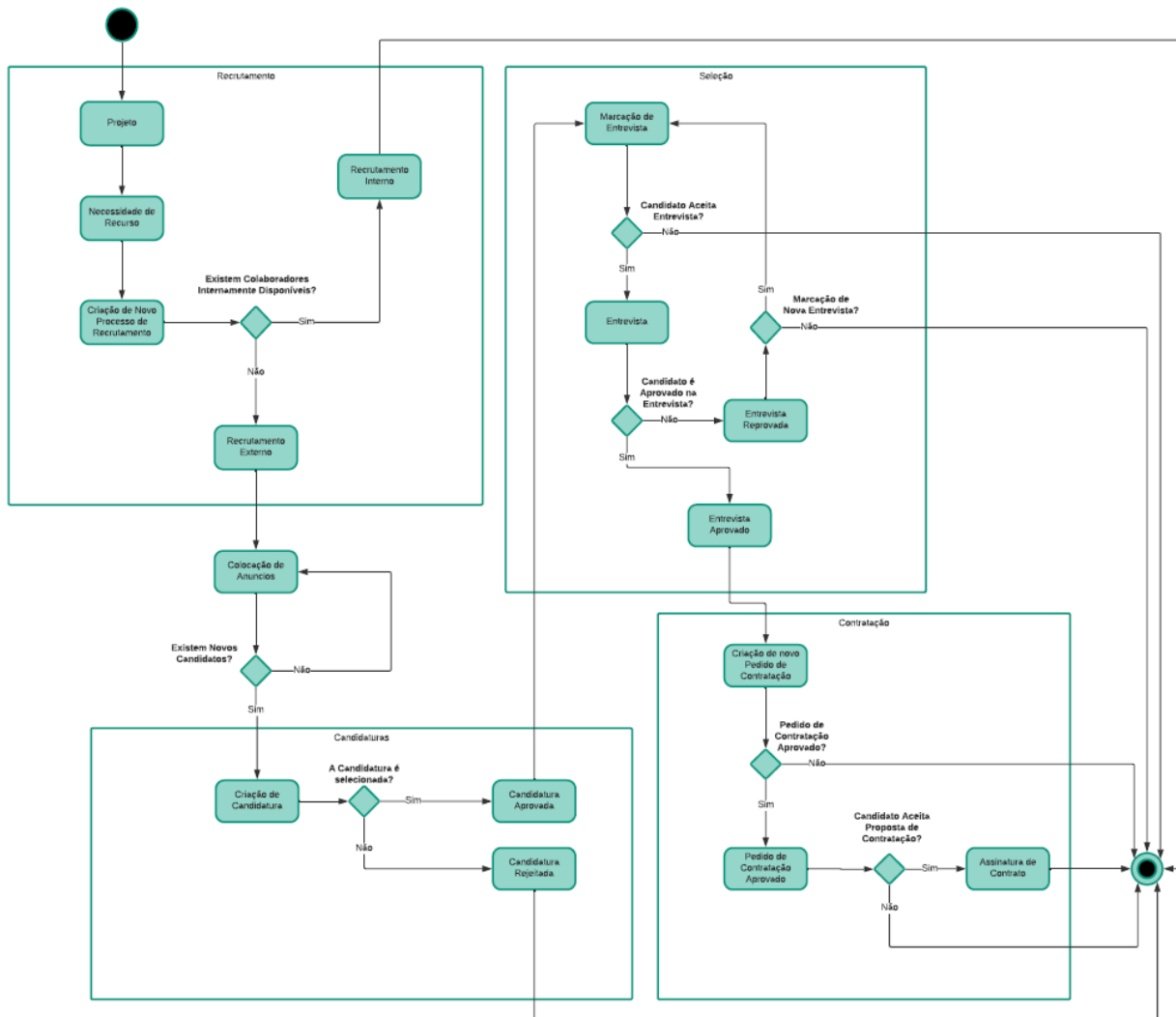


Figura 23 - Diagrama de Atividade do SI (Fonte: Elaboração própria)

Diagrama de Estado

No diagrama de estado apresentado na Figura 24, é possível compreender como é conduzida a evolução de estados dentro da aplicação e como os mesmos se interligam. Inicialmente é criado um processo de recrutamento, que origina o estado “Em Processamento”, de seguida chegam as candidaturas e surgem vários estados para os candidatos, que ao seguirem para “Entrevista Marcada” significa que uma Entrevista está como “Agendada”. A mesma evolui para “Avaliada” ou “Cancelada” e o estado do candidato pode passar para “Fase de Proposta”, neste estado surge então o pedido de contratação que é criado como “Em Análise”, de seguida após a decisão pode ser “Aprovado” ou “Rejeitado”. Caso seja “Aprovado” o candidato segue então para “Contratado” e quando o processo de recrutamento tiver o número de pessoas necessárias o mesmo passa para “Concluído”.

Esta organização de estados permite que dentro do SI, exista uma evolução do estado, estando todos os estados interligados ao estado do candidato. Isto vai permitir criar a lógica necessária para a evolução de um processo do início ao fim dentro do sistema.

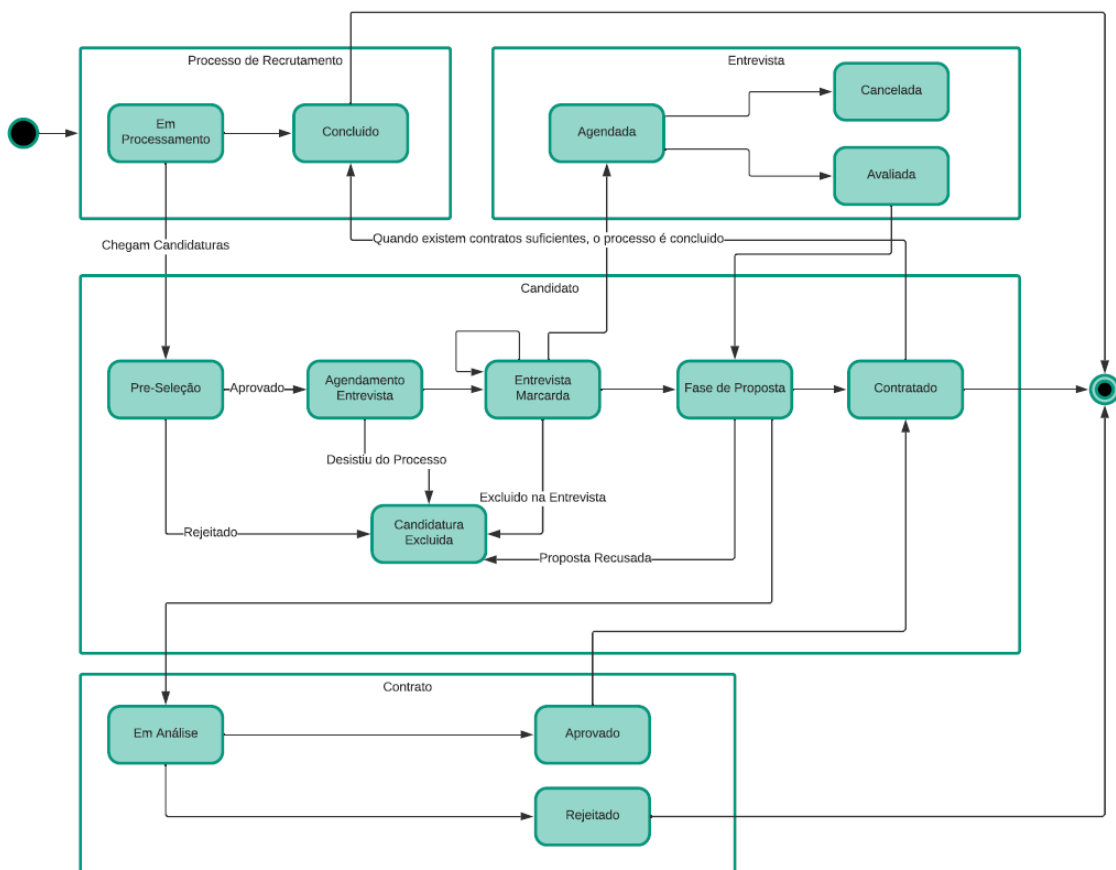


Figura 24 - Diagrama de Estado do SI (Fonte: Elaboração própria)

4.3. DESENHO E ANÁLISE DE SISTEMA

Este capítulo apresenta o trabalho realizado relativo ao desenho e a análise do SI, através dos diagramas de classes, modelo lógico de dados e o desenho de interfaces. Este desenvolvimento permitiu a melhor compreensão e serviu de base para posteriormente se desenvolver o SI.

4.3.1. Diagrama de Classe

Na Figura 25 está representado o diagrama de classes utilizado no SI. O mesmo permitiu mapear a estrutura do sistema de uma forma simples e eficaz e serviu de base para a programação do SI.

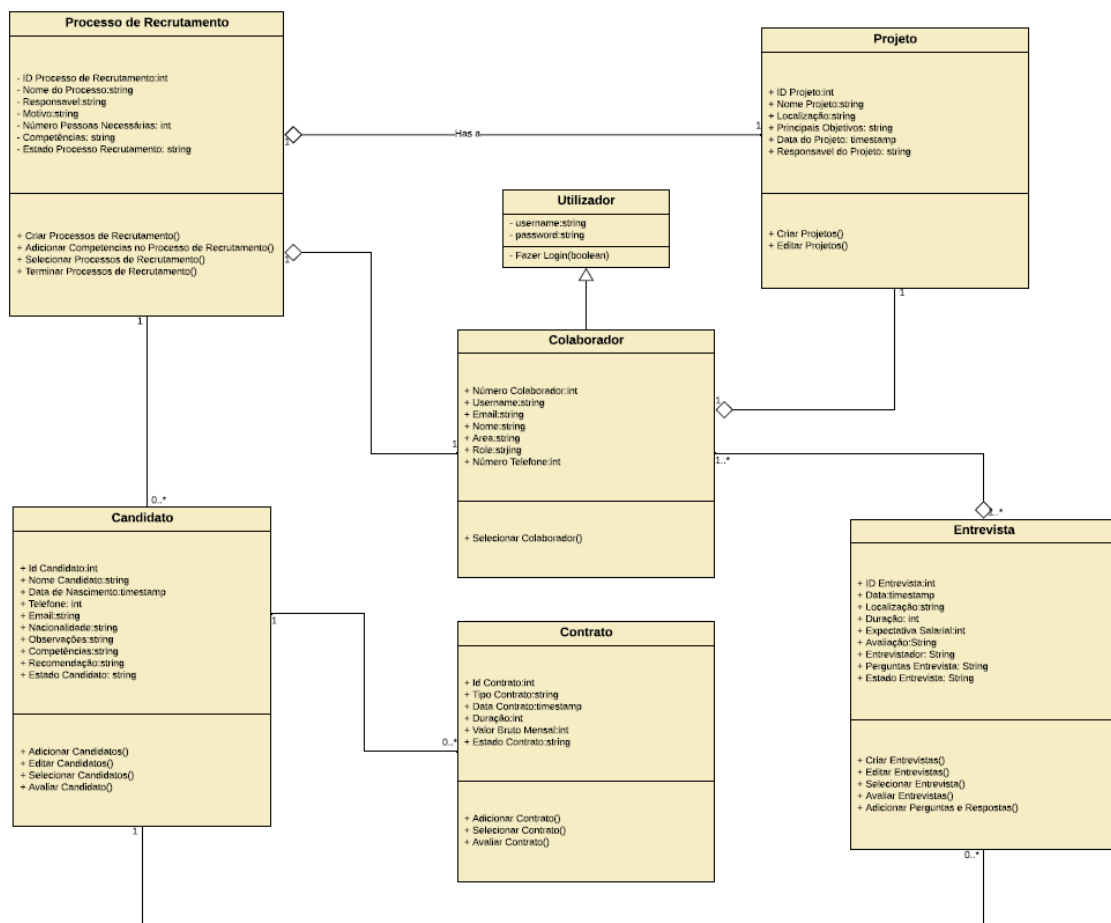


Figura 25 - Diagrama de Classes do SI (Fonte: Elaboração própria)

4.3.2. Modelo Lógico de Dados

O modelo lógico de dados permitiu mapear as diversas entidades, os seus atributos e as suas relações. Este modelo serviu de base para o desenvolvimento do modelo físico de dados e que por sua vez permitiu uma melhor compreensão para o desenvolvimento do sistema. Na Figura 26 é apresentado o modelo lógico de dados utilizado no desenvolvimento do SI.

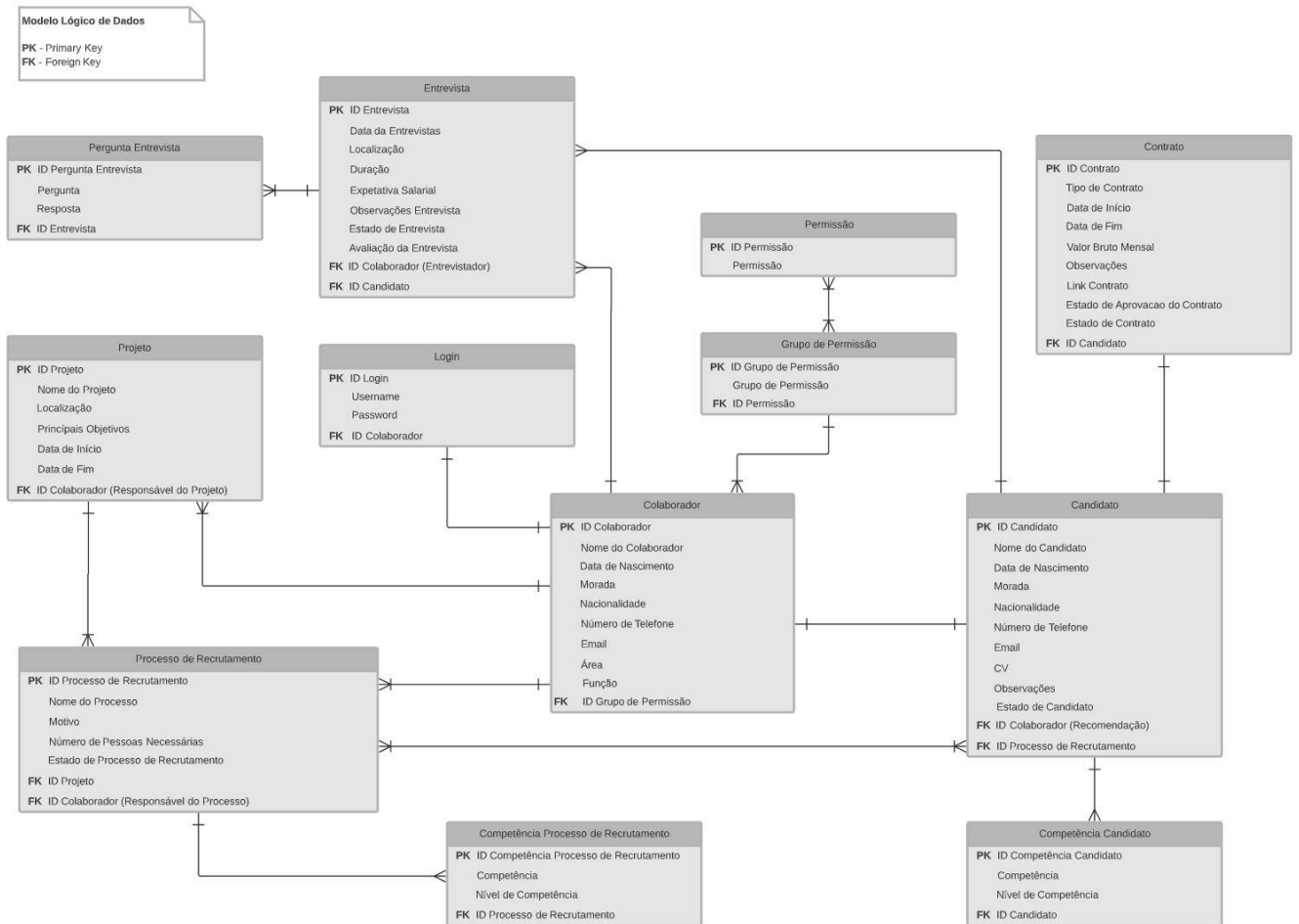


Figura 26 - Modelo Lógico de Dados do SI (Fonte: Elaboração própria).

4.3.3. Desenho de Interfaces

No desenho de interfaces foi proposto organizar e perceber como seriam desenvolvidos os ecrãs e as suas ligações.

O desenho das interfaces foi dividido em seis principais páginas para o SI (Figura 27), sendo elas: Homepage, Projetos, Processo de Recrutamento, Candidaturas, Entrevistas e Contratação. Deste modo foi decidido criar um menu lateral com as seis páginas. Nestas seis páginas principais são contidas as informações gerais, através de tabelas com a informação geral para depois se poder seleccionar e consultar.

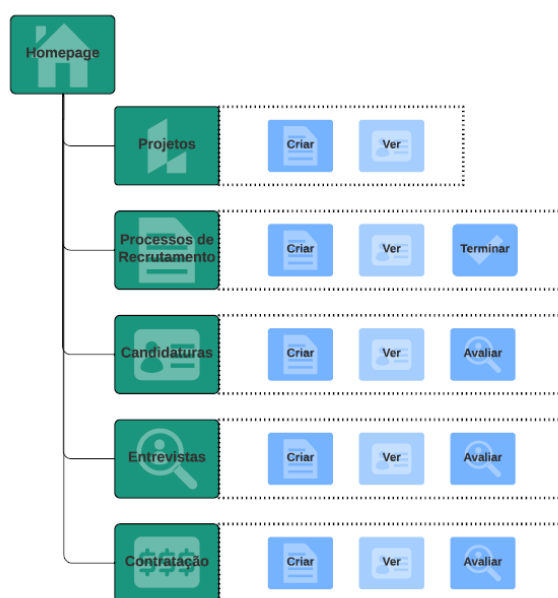


Figura 27 – Esquema com a organização das páginas e as suas funcionalidades (Fonte: Elaboração própria)

Em cada página será possível ver uma tabela com todos os registos associados, e ao clicar num desses registos irá proceder-se a visualização do registo seleccionado, podendo em alguns casos criar ações sobre os mesmos. Isto possibilita a circulação e atualização de informação entre o utilizador e o sistema.

Através do esquema presente na Figura 27 foi possível tornar a compreensão do desenho dos ecrãs da aplicação mais simples e intuitiva.

4.4. DESENVOLVIMENTO DO SISTEMA

4.4.1. Arquitetura

A arquitetura Model-View-Controller (MVC) é um tipo de arquitetura de software normalmente utilizado para implementar interfaces de utilizador, lógica de controlo e manipulação de dados. Este divide-se em três componentes: *Model*, *View* e *Controller*, estando todos interligados entre si.

- *Model*: Este tem a função de definir o comportamento ao nível do núcleo funcional (Varajão, 1998). É responsável pela lógica de negócio através da sua interação com a BD, permitindo desta forma a leitura, escrita e validação de dados.
- *View*: Esta é a camada visualizada pelo utilizador, sendo seu objetivo a demonstração dos dados ao mesmo.
- *Controller*: É responsável pela gestão de fluxo da aplicação, fazendo intermediação entre o *Model* e o *View* nas respostas aos pedidos do utilizador.

A arquitetura MVC permite a divisão entre a parte do que é a lógica de negócio e o que é a demonstração do software. É um sistema que tem como vantagem a modularidade, ou seja, a construção do software pode ser feita de forma separada, podendo ao longo da construção, partes serem reutilizadas para outras aplicações. É devido a esta capacidade de modularidade que a aplicação é dividida nas três camadas acima referidas, pois simplifica e auxilia na manutenção e reutilização das partes. A Figura 28 representa o comportamento da arquitetura MVC.

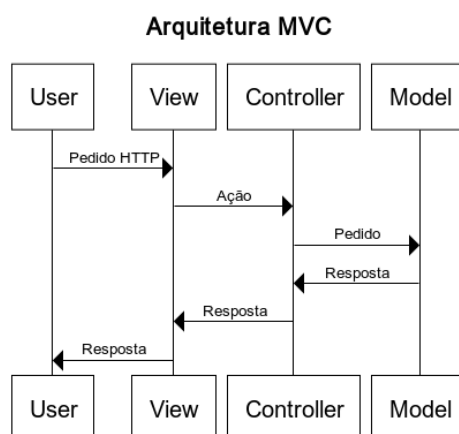


Figura 28 - Diagrama de Sequência com a Arquitetura MVC (Fonte: Elaboração própria).

O padrão MVC é utilizado em diversas *frameworks*, como é o caso daquela que será utilizada neste projeto: Oracle ADF.

O Oracle ADF é uma *framework* para o desenvolvimento de aplicações construído através da Java Enterprise (JEE) que permite, de forma simples, criar aplicações ponta a ponta, aumentando desta forma a sua produtividade. Por sua vez, JEE é uma plataforma padronizada, robusta, escalável e segura, que forma a base para muitas das aplicações que são desenvolvidas nos dias de hoje.

A arquitetura do Oracle ADF é baseada no padrão MVC estando dividida em quatro camadas: *Model*, *View*, *Controller* e *Business Service*.

Na Figura 29 pode ser observada a arquitetura Oracle ADF.

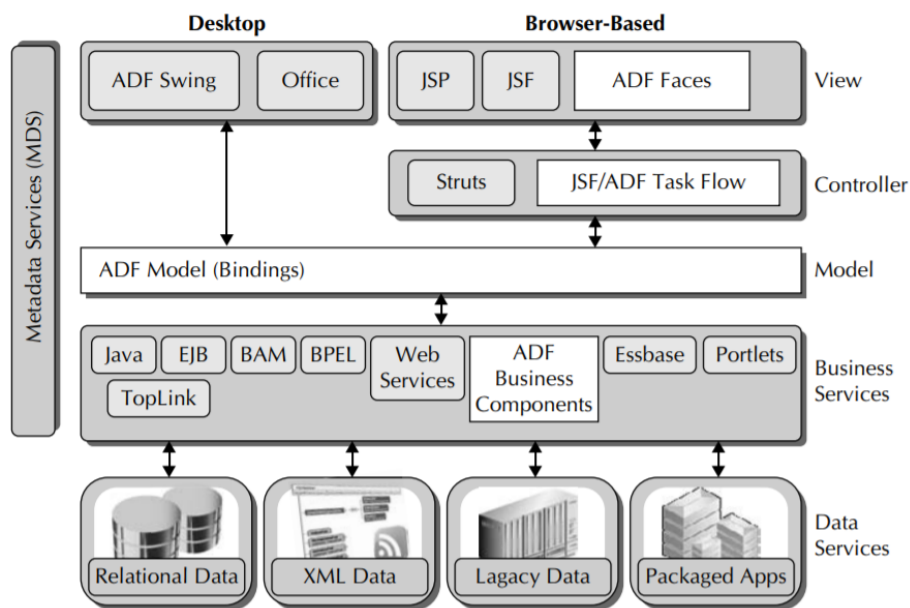


Figura 29 - Arquitetura Oracle ADF (Fonte: Ronald, 2011)

Na Figura 29 é possível verificar as quatro camadas existentes e os vários componentes associados, sendo possível observar na mesma que nesta *framework* surge uma nova camada denominada *Business Service*. Esta permite gerir o acesso a dados provenientes de diferentes fontes e age sobre a lógica de negócio, desta forma o seu desenvolvimento torna-se mais flexível e simples.

Nesta camada o serviço mais relevante para efeitos do projeto é o ADF *Business Components*, este permite fazer a ligação à BD e a interação com a mesma. Este é composto por três principais componentes: *Entity Objects*, *View Objects (VOs)* e *AppModules*.

- Os *Entity Objects* representam entidades de tabelas na BD, permitindo desta forma manipular a lógica do negócio. Com isto é possível criar regras e comportamentos na lógica de negócio de uma forma eficaz e consistente. Na Figura 30 podemos visualizar o exemplo da *Entity Object* criada para o Projeto e os seus respetivos atributos.

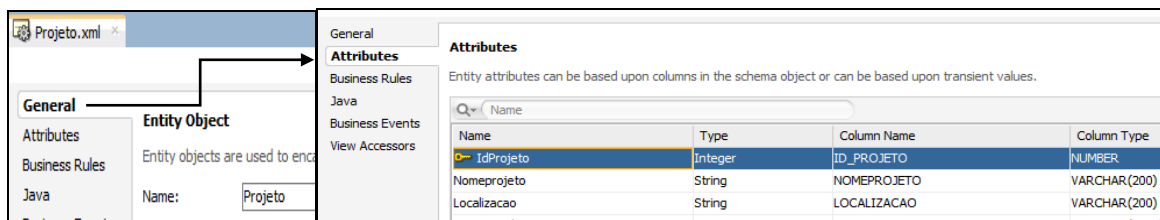


Figura 30 – Exemplo da *Entity Object* – Projetos

- Os VOs permitem consultar informação através de consultas (*queries*) de SQL e permitem que os dados possam ser modulados e alterados (inserir/editar) por um utilizador. Na Figura 31 podemos ver o exemplo da *query* utilizada no VO para permitir apresentar uma tabela com os candidatos.

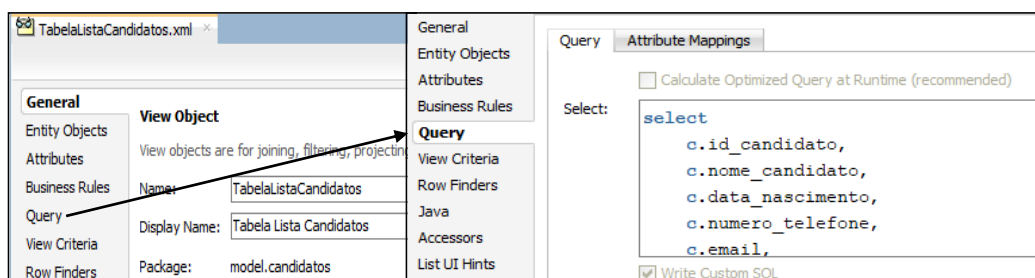


Figura 31 – Exemplo do VO - Tabela Lista de Candidatos

- O *AppModule* permite definir um modelo de dados dinâmico, podendo ser controlado sobre chamadas, através de procedimentos e funções. O mesmo permite assim testar e perceber o comportamento de alguns dados. Na Figura 32, podemos visualizar o *AppModule* dos candidatos, com os vários VOs utilizados.

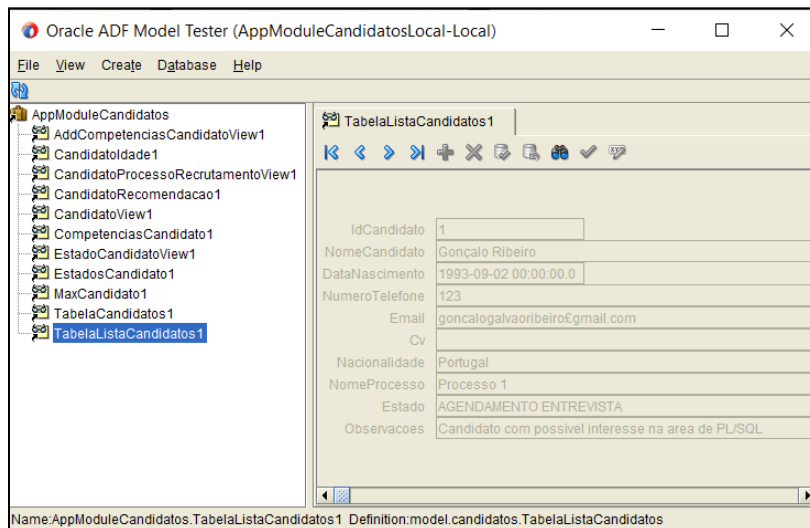


Figura 32 – Exemplo do *AppModule* relativo aos Candidatos

Nesta arquitetura a camada *Model* tem a função de criar um elo de ligação entre a interface do utilizador e os *Business Services*, permitindo que ao nível do *Controller* esteja mapeado e disponível o acesso aos dados consistentes da implementação do *Business Service*, mapeando o desenvolvimento de forma simples e consistente, sem a obrigatoriedade de escrever código.

Existem dois conceitos inerentes ao ADF *Model*, sendo eles: *Data Controls* e *Declarative Bindings*.

- Os *Data Controls* permitem expor coleções de dados e operações provenientes da lógica de negócio na qual é possível criar componentes, de forma automática, através de *drag and drop*, permitindo assim o mapeamento direto à BD, simplificando desta forma todo o processo.
- As *Declarative Bindings* permitem aceder aos dados do *Data Control*, onde é possível mapear os detalhes das operações e das coleções de dados.

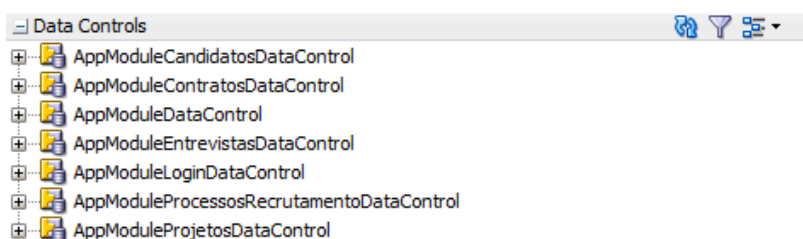


Figura 33 – Exemplo dos *Data Controls* utilizados na aplicação

Podemos concluir que o *Model* simplifica bastante o processo de desenvolvimento, pois permite uma forma de unir qualquer interface de utilizador a qualquer serviço de negócio, através de uma forma bastante simples e consistente.

Nesta arquitetura a camada *Controller* define o fluxo entre a navegação das páginas, processa as informações de entrada do utilizador e integra com a JavaServer Faces (JSF). Nesta camada estão contidos vários elementos como os *Task Flows*, *Managed Beans*, entre outros.

- Os *Task Flows* do ADF definem e fornecem uma abordagem modular para definir os vários fluxos necessários entre páginas.
- Os *Managed Beans* são classes com código Java, que permitem erguer algum tipo de lógica adicional à aplicação.

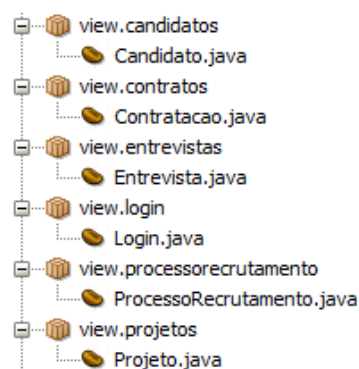


Figura 34 - Exemplo das *Managed Beans* utilizadas no desenvolvimento da aplicação

Nesta arquitetura a camada *View*, representa o *front-end* da aplicação, esta permite criar as páginas através de JSF fornecendo desta forma a interface do utilizador para a aplicação.

O JSF é uma estrutura padrão bastante utilizada para quem é responsável pelo desenvolvimento de Java, pois define uma Application Programming Interface (API) e um conjunto básico de *tags* em Extensible Markup Language (XML) para criar interfaces de usuário baseadas na *Web*. A JSF compreende um conjunto de componentes visuais, que são suportados pelo ADF Faces presentes nesta camada.

O ADF Faces fornece uma estrutura JEE que permite o desenvolvimento de aplicações baseadas em padrões da *web* de maneira visual e eficaz. O seu desenvolvimento pode ser feito através dos vários componentes *drag and drop* disponíveis, na Figura 35 podemos ver a interface dos componentes do ADF Faces.

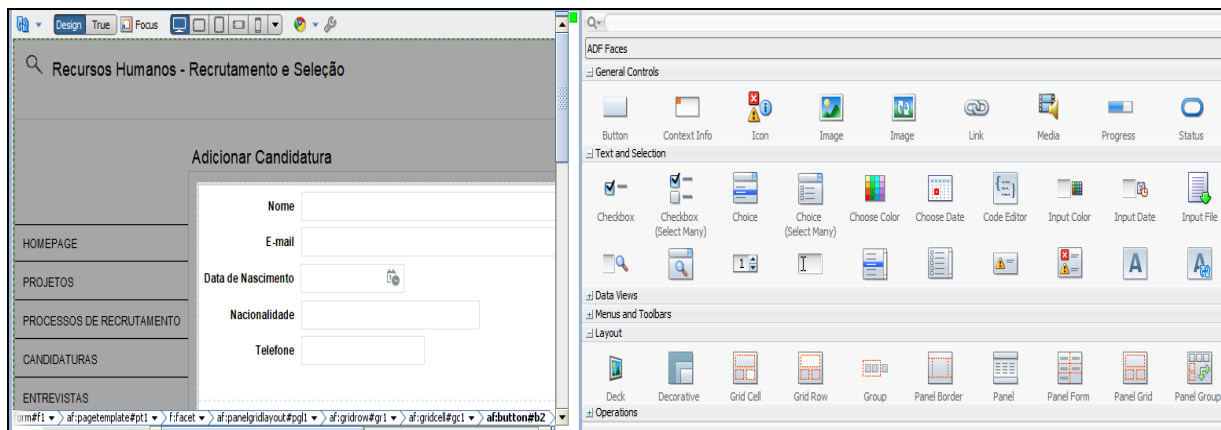


Figura 35 – Exemplo da Interface do JDeveloper - Oracle ADF com a página para adicionar uma nova candidatura

Como foi possível compreender, esta *framework* permite desenvolver aplicações de uma forma mais rápida, devido a biblioteca de componentes de software reutilizáveis e pelos recursos de suporte no design. A sua arquitetura permite separar a lógica de negócio e a navegação entre páginas e interface do utilizador, indo ao encontro da arquitetura MVC, permitindo assim obter uma forma de desenvolvimento bastante simplificada e produtiva.

4.4.2. Modelo Físico de Dados

O Modelo Físico de Dados foi desenvolvido no sistema de gestão de BD da Oracle através do Oracle *Database 18c Express Edition*. O mesmo representa o modelo utilizado de base no SI, contendo todas as tabelas e as suas relações.

Na implementação do modelo físico de dados, fazendo a sua passagem do modelo lógico para o físico, foram tomadas algumas decisões, entre elas:

- A existência de uma tabela de Códigos para inserir todo o tipo de códigos necessários dentro do SI, entre eles: Estados, Competências, Nível de Competência, Permissões, etc.
- A criação de uma tabela associativa para armazenar os estados para cada entidade representativa, permitindo assim armazenar vários estados criados para cada entidade.

Na Figura 36 é apresentado o modelo físico de dados exportado através do Oracle SQL Developer.

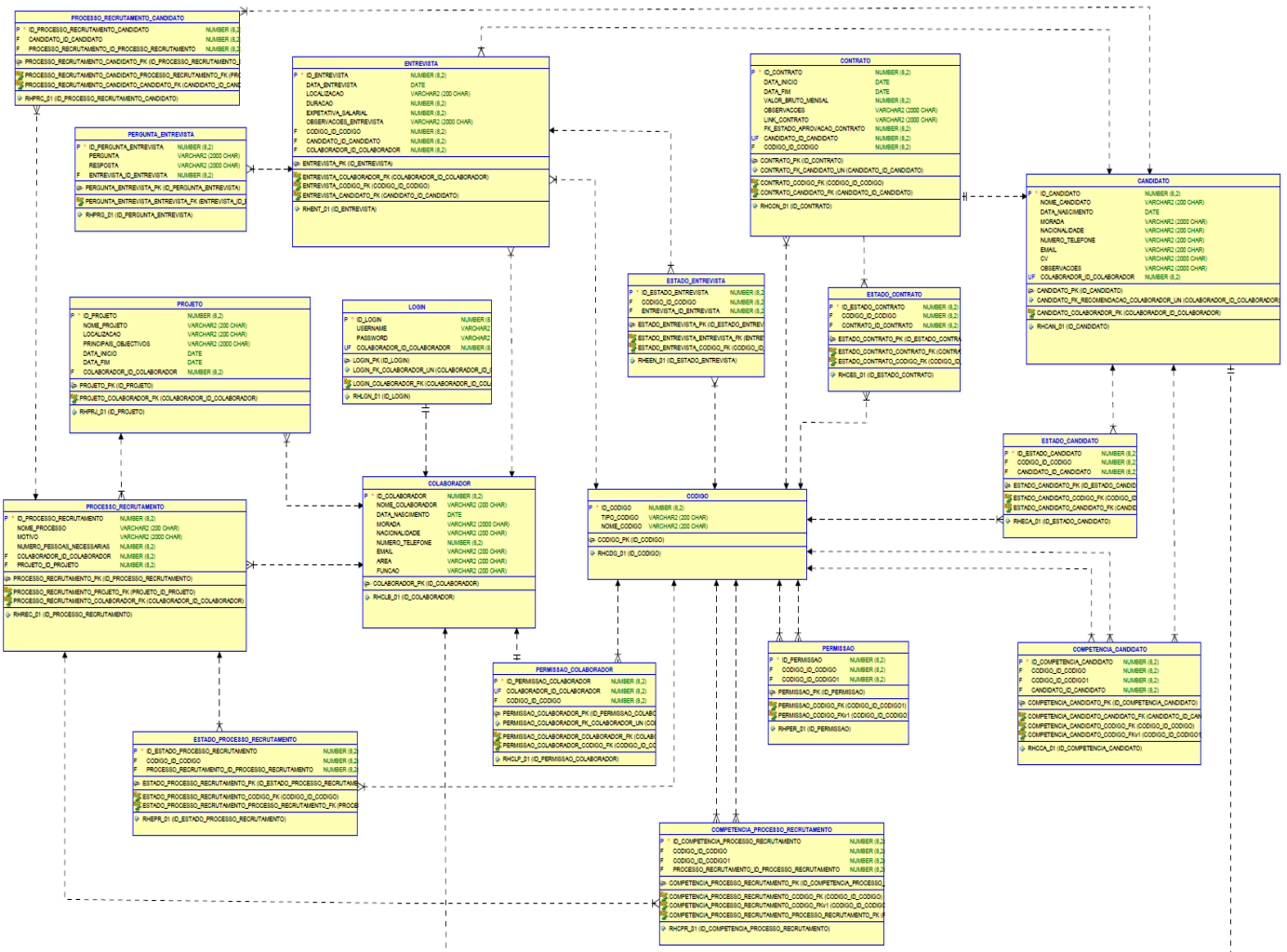


Figura 36 - Modelo Físico de Dados do SI (Fonte: Elaboração própria).

4.4.3. Programação do Sistema

O capítulo programação do sistema pretende demonstrar o desenvolvimento da aplicação. A aplicação está dividida em dezassete páginas. Cada página representa uma JSF que é composta por XML, cada JSF contém uma *Page_Definition*, na qual estão todas as *Declarite Bindings* que mapeiam ligações entre o ecrã e a BD através do *Data Control*. Os *Data Controls* por sua vez disponibilizam de uma forma simples e intuitiva toda a lógica de negócio criada ao nível do *Business Component*, tendo sido criado um conjunto de coleções de dados, entidades e operações de ligação à BD para fornecer a lógica necessária.

A *framework* utilizada permitiu criar de uma forma simples e rápida os diversos ecrãs e as diferentes lógicas. Tudo isto utilizando componentes *drag and drop*, que permitiram gerar todo o código de uma forma mais simples e rápida.

Este desenvolvimento foi dividido em três importantes fases. A primeira fase passou por desenvolver o DDL para a aplicação em Oracle SQL, permitindo ter assim disponível a BD. De seguida procedeu-se à ligação à BD desenvolvida, utilizando um componente da *framework* criando uma conexão com a BD implementada (Figura 37).

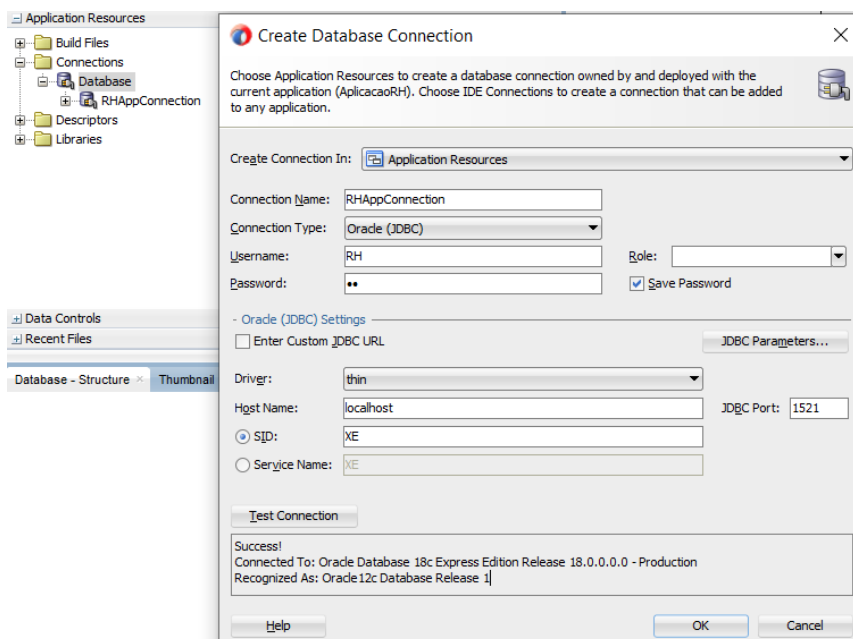


Figura 37 - Componente de conexão com a BD do sistema

Numa segunda fase foi criado todo o mapeamento das entidades da BD e iniciada a construção da lógica de negócio necessária, que consistiu no desenvolvimento de VOs, *Entity Objects* e *AppControllers*. A construção da lógica de negócio foi principalmente dividida em 6 principais grupos, a lógica por detrás dos Projetos, dos Processos de Recrutamento, dos Candidatos, das Entrevistas, da Contratação e do Login.

Este desenvolvimento permitiu não só criar, mas também testar os componentes, e reutilizá-los, se necessário, estando estes diretamente interligados à BD.

Ainda nesta fase à medida que se avançou na construção de todos estes componentes com a lógica de negócio, foram criados os vários *Data Controls* para cada parte da lógica de negócio. Cada *Data Control* representa um agrupamento lógico de dados para cada um dos seis grupos, isto simplificou e permitiu que na construção das páginas fosse possível, através de *drag and drop*, serem utilizados os objetos-filho listados sob o *Data Control* permitindo assim construir tabelas, botões e outros componentes de forma automática e direta à BD.

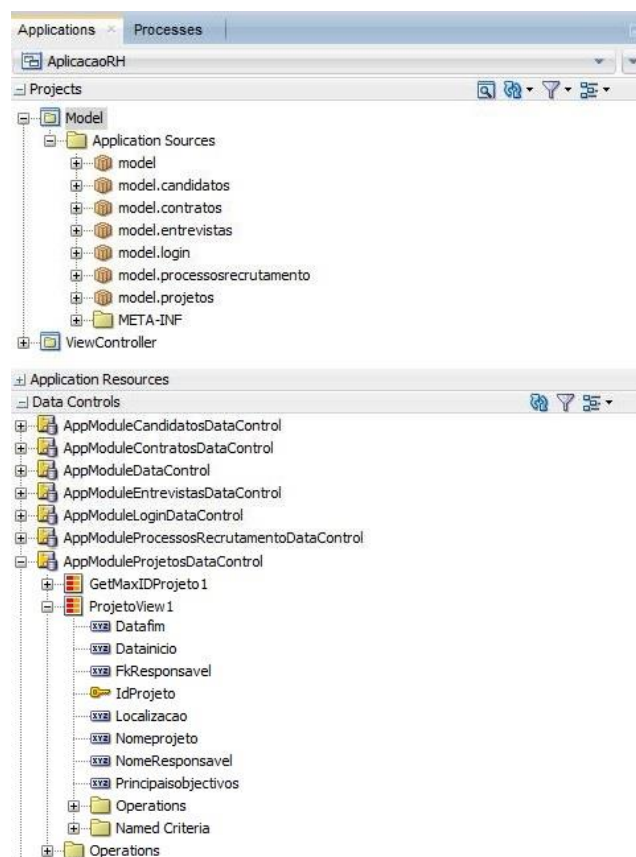


Figura 38 - *Data Controls* e objetos filhos listados utilizados na aplicação.

Na terceira fase foi construído um *template* simples e intuitivo de maneira a ser utilizado por todas as páginas à medida que foram criadas. Após criado o *template* foram criadas as diferentes páginas no formato JSF, depois para cada uma foi criado o seu conteúdo, utilizando os dados provenientes pelo *Data Control* através das *Declarative Bindings*.

Cada página JSF ao utilizar componentes provenientes do *Data Control* originou uma *Page Defenition*, onde estão contidos todas os mapeamentos existentes à lógica do negócio utilizada. No decorrer do desenvolvimento das páginas foi sempre possível regressar, se necessário, às outras fases para adicionar novas lógicas, por exemplo, como novos VOs.

Nesta fase também foram criadas as classes de java denominadas *Managed Beans*, que suportam alguma da lógica por de trás de cada ecrã.

Segue-se agora a explicitação de cada uma das 6 classes principais que constituem a aplicação:

- A classe Login (Colaborador), permite saber se um determinado utilizador do sistema está registado na BD com as credenciais utilizadas. Esta classe permitiu saber se o utilizador tem acesso a entrar no sistema ou não. E deste modo saber quem é este mesmo no sistema através da disponibilização dos seus dados. Esta classe é responsável por gerir as funcionalidades dos ecrãs de Login e Homepage.
- A classe Projeto, é responsável por gerir todas as funcionalidades no ecrã de Projetos e Novo Projeto, permitindo a criação de novos projetos no sistema e a edição de projetos existentes.
- A classe Processo de Recrutamento é responsável por gerir todas as funcionalidades nos écrans Processo de Recrutamento, Detalhes Processo Recrutamento e Novo Processo de Recrutamento. Esta classe permite a criação de novos processos, permite também adicionar competências ao processo, e contém a lógica para poder filtrar pelo registo selecionado e visualizar os seus detalhes. É através desta Classe que é possível terminar o processo de recrutamento, atualizando o seu estado, passando do estado de “em processamento” para “terminado”.
- A classe Candidatos é responsável por gerir todas as funcionalidades nos ecrãs Candidaturas, Detalhes Candidatura e Nova Candidatura. Esta classe contém a lógica para adicionar novos candidatos, adicionar competências ao candidato, editar candidato, alterar o seu estado e selecionar o mesmo e ver os seus detalhes. Todos os botões nos quais acionam estados para

um candidato, simplesmente criam um novo registo na BD com o estado de evolução do candidato, podendo este passar desde “Selecionado” a “Contratado”.

- A classe Entrevista é responsável por gerir todas as funcionalidades nos ecrãs Entrevistas e Detalhes/Avaliação Entrevista. Esta classe permite e suporta o agendamento/criação de entrevistas, a seleção da entrevista e a sua respetiva avaliação incluindo a lógica necessária para inserir perguntas e respostas.
- A classe Contratação é responsável por gerir todas as funcionalidades nos ecrãs Contratação e Detalhes/Avaliação Contratação, esta classe contém a lógica para criar pedidos de contratação, para filtrar pelo contrato selecionado e ver os seus detalhes e para aprovar ou rejeitar um pedido de contratação.

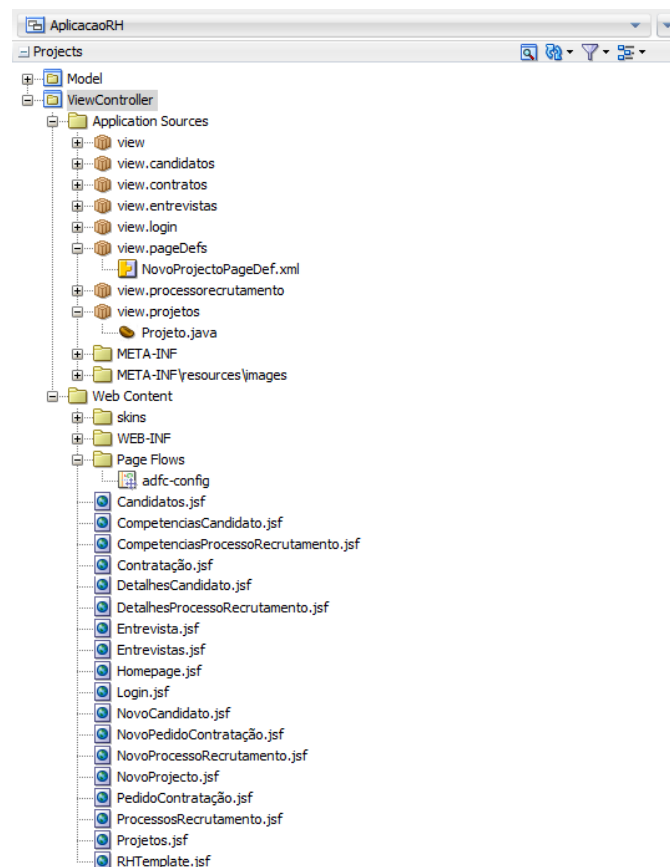


Figura 39 – Painel com JSF's, *Managed Beans* (p. ex. Projeto.java), *Page Definition* (p. ex. Novo Projeto) and *Task Flow* (adfc-config) utilizado no SI.

Ainda nesta terceira fase foi criado o *Task Flow* da aplicação, que representa os respetivos fluxos entre as páginas na aplicação, permitindo as suas ligações. Na Figura 40 está representado o *Task Flow* final da aplicação, onde se pode constatar que existem dezassete páginas, sendo que seis estão

interligadas entre si através de um ADF *wild-card*, isto vai permitir ser possível navegar entre as mesmas, representando assim as seis páginas principais na aplicação (Homepage, Projetos, Processos de Recrutamento, Candidatos, Entrevistas e Contratação), que vão corresponder ao menu de navegação.

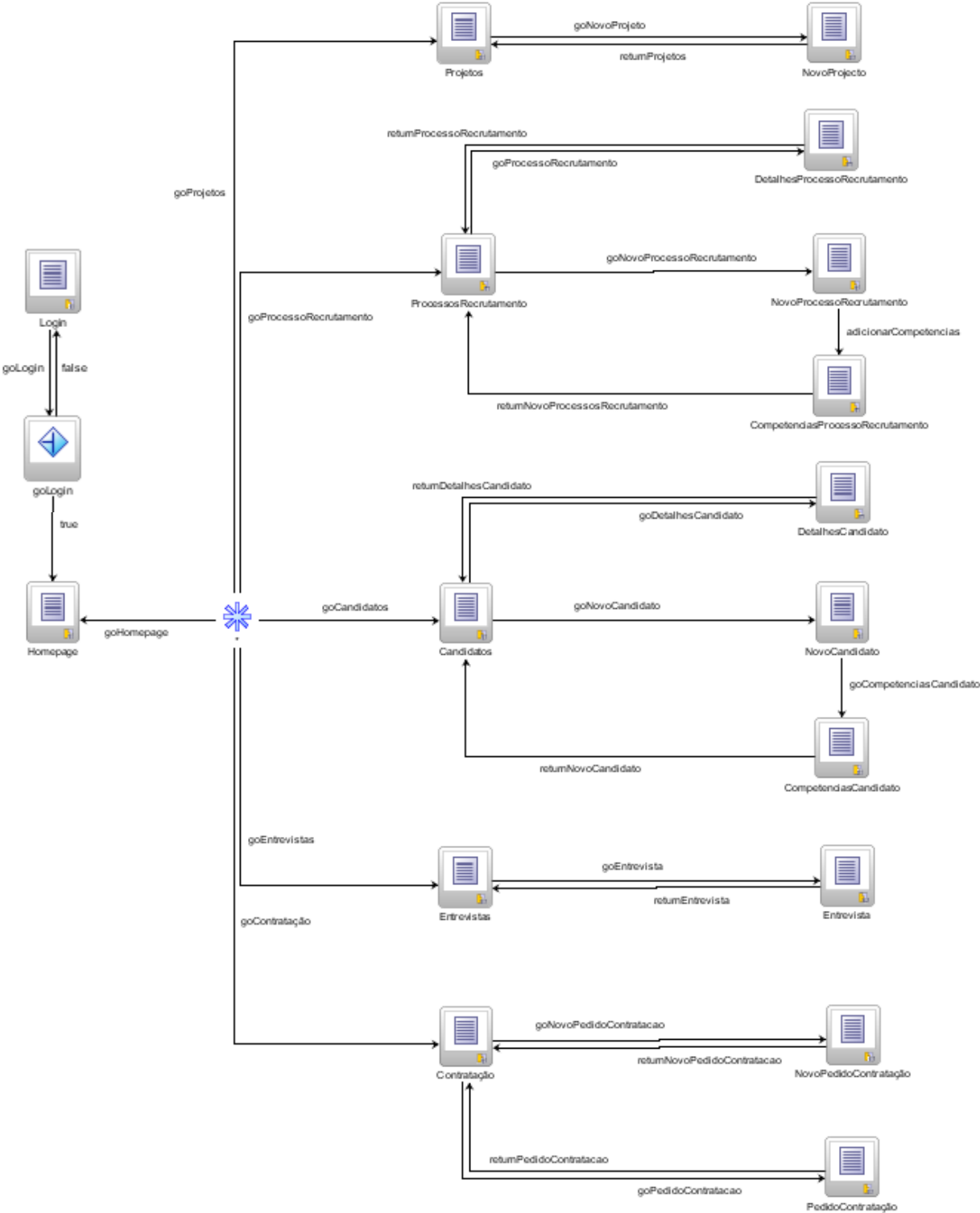


Figura 40 - Task Flow da Aplicação

Segue a apresentação de todos os ecrãs desenvolvidos e as suas funcionalidades, acompanhados de *print screens* da própria aplicação. Serão apresentados excertos de código onde é possível visualizar alguma da lógica desenvolvida, para assim ser possível a melhor compreensão da própria aplicação e do modo como funciona. Como se segue na Figura 41, quando se inicia a aplicação, o primeiro ecrã a surgir é o de Login.



Figura 41 - Ecrã de Login

No ecrã de Login representado na Figura 42 é representada a entrada no SI. O utilizador tenta entrar com as suas credenciais: *username* e *password* e depois o sistema verifica se as mesmas existem. Na Figura 42, podemos visualizar a lógica que permite verificar se existe ou não o utilizador na BD.

The screenshot shows the development environment for the application. On the left, the 'View Criteria' configuration window is open, showing the 'View Object Class' set to 'model.login.GetUsersDetailsImpl'. In the center, a diagram shows the 'tryLogin' activity with a 'Login' output variable that is set to 'false' if the activity fails and 'true' if it succeeds. On the right, the 'General' tab of the activity configuration shows the 'Activity ID' as 'tryLogin' and the 'Default Outcome' as 'false'. Below these, a table lists the 'Cases' and their 'Outcomes':

Expression *	Outcome *
#{login.isLogin() == true}	true
#{login.isLogin() == false}	false

At the bottom, a code snippet in Java implements the login logic:

```
// Username e Password recebida pelo utilizador
String username = getUsername().getValue().toString();
String password = getPassword().getValue().toString();
// Receber a informação do utilizador que se está a tentar fazer login - Password recebida pela BD
GetUsersDetailsImpl getLoginDetails = (GetUsersDetailsImpl) AppModule.findViewObject("GetUsersDetails1");
String BD_user_password = getLoginDetails.getUserDetails(username);
if (password.equals(BD_user_password))
{
    // Login OK
    setLogin(true);
}
else {
    setLogin(false);
    showMessage("Não tem permissões para aceder ao sistema - RH - Recrutamento e Selecao", "error");
}
```

Figura 42 – Lógica para verificar se o *Username* e *Password* existem na BD

Assim sendo como na Figura 42 indica, existe um VO que permite verificar se um *username* e *password* estão registados na BD, essa validação é feita através da criação de um *View Criteria* e de um *VO Class*, sendo esta uma classe de Java que está interligada ao VO e permite o seu controlo. Através desta configuração é possível filtrar pelo *username* selecionado, e saber se o mesmo existe e coincide com a *password* colocada. Deste modo a classe de C Login, irá verificar através da VO Class, e através da criação de um *router* no *task flow*, em que caso existem os valores que o utilizador inserir, o mesmo é direcionado para o ecrã da homepage e caso não exista, será enviada uma mensagem a dizer que o utilizador ou a palavra-passe estão erradas e através do router, o mesmo permanecerá na mesma página.

Após entrar na Homepage, o utilizador, com base na sua função irá aceder na totalidade ou apenas a algumas partes do todo da aplicação. Neste ecrã é possível que o utilizador, que se autenticou, veja as entrevistas que tem marcadas. O menu disponível na parte lateral esquerda, é a base de navegação entre as seis principais páginas dentro do sistema.

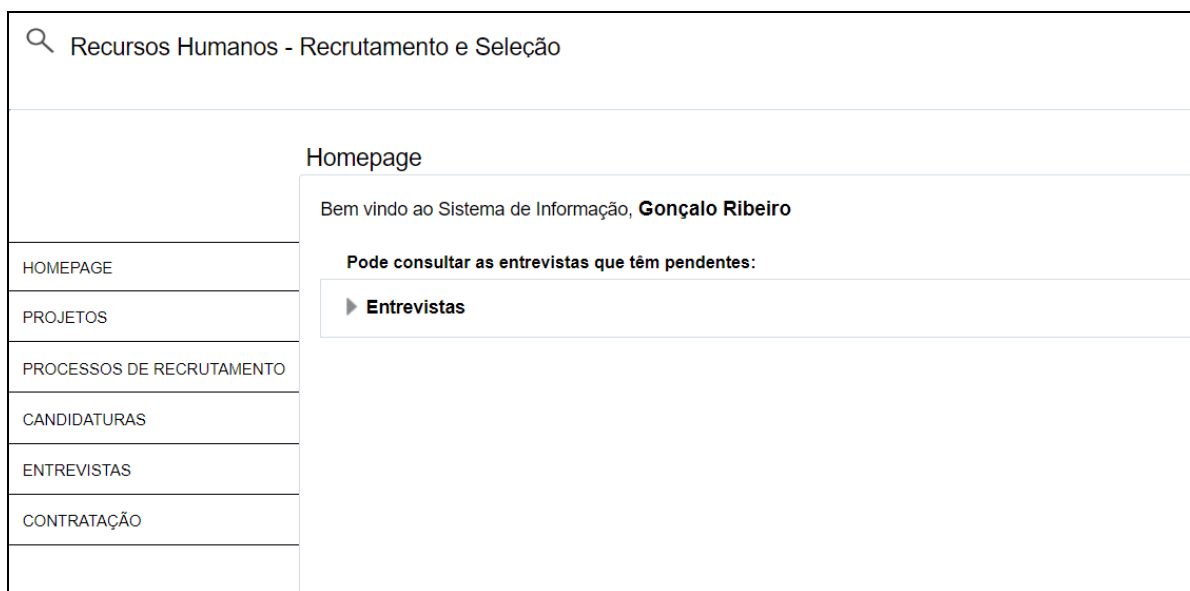


Figura 43 - Ecrã de Homepage

Ao clicar no “Menu - Projetos” seguimos para o ecrã dos projetos, onde estão disponíveis e listados todos os projetos na aplicação, sendo estes a base de qualquer processo de recrutamento dentro do SI. É sempre, inicialmente, necessário criar ou selecionar um projeto para depois se puder criar os vários processos de recrutamento. O ecrã na Figura 44, apresenta um *print screen* do ecrã onde se podem consultar todos os projetos disponíveis no SI.

Recursos Humanos - Recrutamento e Seleção

Projetos

Criar Novo Projeto

Projetos Disponíveis:

Nº Projeto	Nome	Nº Responsável	Responsável	Localização	Data Inicio	Data Fim	
6	Projeto Financeiro	1	Gonçalo Ribeiro	Lisboa	01-06-2020	30-06-2021	Editar
5	Projeto Exemplo 1	2	Utilizador 1	Lisboa	08-06-2020	01-09-2020	Editar
4	Projeto A	3	Utilizador RH	Lisboa	29-05-2020	25-06-2020	Editar
3	Projeto SI	1	Gonçalo Ribeiro	Lisboa	18-05-2020	06-05-2021	Editar
2	Projeto Gon	1	Gonçalo Ribeiro	Lisboa	14-05-2020	26-05-2021	Editar
1	Projeto ABC	1	Gonçalo Ribeiro	Lisboa	14-05-2020	14-05-2021	Editar

Figura 44 - Ecrã de Projetos

A construção da tabela apresentada na Figura 44, foi implementada através da utilização do *Data Control* relativa aos Projetos. É aqui e através de *drag and drop* que uma grande parte do código XML dentro da JSF é pré-preenchido e o *page definition* é criado e são mapeadas as *declaritive bindings*, permitindo deste modo aceder automaticamente aos detalhes dos dados. Na Figura 45 podemos visualizar este código gerado composto por várias *tags* XML relativas à tabela “projetos” e as várias *bindings* que mapeiam os mesmos na ligação a BD.

```

<af:panelGroupLayout id="pg111">
  <af:table value="#{bindings.ProjetoView1.collectionModel}" var="row"
    rows="#{bindings.ProjetoView1.rangeSize}"
    emptyText="#{bindings.ProjetoView1.viewable ? 'No data to display.' : 'Access Denied.'}"
    rowBandingInterval="0" fetchSize="#{bindings.ProjetoView1.rangeSize}" id="t5"
    selectedRowKeys="#{bindings.ProjetoView1.collectionModel.selectedRow}"
    selectionListener="#{bindings.ProjetoView1.collectionModel.makeCurrent}"
    rowSelection="single"
    filterModel="#{bindings.ProjetoView1Query.queryDescriptor}"
    filterVisible="true"
    queryListener="#{bindings.ProjetoView1Query.processQuery}" varStatus="vs"
    binding="#{ProjetosBean.tabelaProjeto}" scrollPolicy="page"
    inlineStyle="width:auto;">
    <af:column headerText="Nº Projeto" id="c9"
      sortProperty="#{bindings.ProjetoView1.hints.IdProjeto.name}"
      filterable="true" sortable="true" width="100"
      inlineStyle="font-weight: bold;">
      <af:outputText value="#{row.IdProjeto}"
        shortDesc="#{bindings.ProjetoView1.hints.IdProjeto.tooltip}"
        id="ot20"/>
    </af:column>
    <af:column headerText="Nome" id="c17"
      sortProperty="#{bindings.ProjetoView1.hints.Nomeprojeto.name}"
      filterable="true" sortable="true" width="160"
      inlineStyle="font-weight: bold;">
      <af:outputText value="#{row.Nomeprojeto}"
        shortDesc="#{bindings.ProjetoView1.hints.Nomeprojeto.tooltip}"
        id="ot14"/>
    </af:column>
  </af:table>
</af:panelGroupLayout>

```

Figura 45 - Código XML gerado com a tabela Projetos

Ao clicar no botão “Novo Projeto” somos direcionados para o ecrã de criação de um Novo Projeto, aqui é possível criar um novo projeto com todo o detalhe necessário, preenchendo o formulário apresentado na Figura 46. Ao adicionarmos este novo projeto voltamos para a página Projetos e podemos visualizar este já criado.

Figura 46 - Ecrã de Criação de Novo Projeto

Após pressionarmos no “Menu – Processos de Recrutamento” somos direcionados para a página Processos de Recrutamento (Figura 47), onde é possível visualizar todos os processos de recrutamento existentes no SI. Nesta página pode-se ser direcionado para uma página com os detalhes de um processo de recrutamento, podendo ser exibidas todas as suas características, incluindo as suas competências e outra página para a criação de um novo processo de recrutamento.

N° Processo	Processo Recrutamento	N° Responsável	Responsável Processo	Nome do Projecto	Área de Negócio	Contratados	Detalhes
7	Recrutamento Exemplo	2	Utilizador 1	Projeto 1	IT	1/5	Detalhes
6	Recrutamento OLAP	1	Gonçalo Ribeiro	Projeto ABC	IT	1/4	Detalhes
5	Recrutamento Projeto Financeiro	1	Gonçalo Ribeiro	Projeto Financeiro	IT	1/5	Detalhes
4	Recrutamento Exemplo 1	2	Utilizador 1	Projeto Exemplo 1	IT	1/4	Detalhes
3	Recrutamento RH	3	Utilizador RH	Projeto A	RH	1/2	Detalhes
2	Recrutamento ABC	1	Gonçalo Ribeiro	Projeto ABC	IT	1/10	Detalhes
1	Recrutamento Gon	1	Gonçalo Ribeiro	Projeto Gon	IT	2/5	Detalhes

Figura 47 - Ecrã Processos de Recrutamento

No sistema todas as *declaritive bindings* geradas através de *drag and drop* pelo *Data Control*, permitem que nos ecrãs seja possível visualizar a informação proveniente da BD, isto é possível devido a criação automática de um *Page Definition* que permite a ligação entre as *declaritive bindings* e o *Data Control* para cada página JSF. Na Figura 48 podemos visualizar o *Page Definition* do ecrã Processos de Recrutamento, contendo todas as *bindings* utilizadas e o seu mapeamento.

```

<bindings>
<tree IiterBinding="TabelaProcessoRecrutamentoIterator" id="TabelaProcessoRecrutamento1">
  <nodeDefinition DefName="model.processosrecrutamento.TabelaProcessoRecrutamento"
    Name="TabelaProcessoRecrutamento10">
    <AttrNames>
      <Item Value="IdProcessoRecrutamento"/>
      <Item Value="NomeProcesso"/>
      <Item Value="PkColaboradorResponsavel"/>
      <Item Value="PkProjeto"/>
      <Item Value="NomeProjeto"/>
      <Item Value="Area"/>
      <Item Value="Motivo"/>
      <Item Value="NumeroPessoasNecessarias"/>
      <Item Value="NumPessoasContratadas"/>
      <Item Value="NomeColaborador"/>
      <Item Value="NomeEstado"/>
    </AttrNames>
  </nodeDefinition>
</tree>
  
```

Figura 48 - *Page Definition* da página Processo de Recrutamento

De seguida ao clicarmos no botão “Criar Novo Processo de Recrutamento” somos direcionados para a página com o formulário de criação de um processo de recrutamento. É neste formulário que é seleccionado o projeto a ser utilizado para a criação de um processo de recrutamento. De seguida, e após o preenchimento de todos os detalhes, são adicionadas todas as competências necessárias ao processo de recrutamento (Figura 49).

Recursos Humanos - Recrutamento e Seleção

Novo Processo de Recrutamento

Nome do Processo: Recrutamento PL/SQL

Nº Colaboradores Necessários: 5

Colaborador Requerente: Utilizador 1

Nome do Projecto: Projeto 1

Motivo do Pedido: Necessidade de colaboradores com conhecimentos básicos em PL/SQL

Retroceder Avançar

Adicionar Competências ao Processo de Recrutamento

Podem adicionar várias competências:

Competência: Competencia 1

Nível: BÁSICO

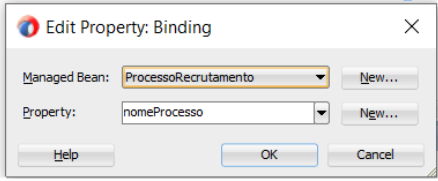
Adicionar Competências

Terminar

Figura 49 - Ecrãs Criação de Novo Processo de Recrutamento e Competências Processo de Recrutamento

A lógica utilizada na criação de novos registos foi efetuada através dos mapeamentos dos componentes com as classes de java utilizadas, através das mesmas são depois efetuadas as validações e por sua vez é feita a inserção de registos. Cada componente e a sua respetiva *binding* existente no formulário é mapeada e torna-se acessível na classe de Java. Na Figura 50 é possível observar o mapeamento do componente e *binding* “nome do processo” correspondente a uma *input-text*, onde se procede à criação de dois métodos de *Get* e *Set* para este elemento de *input* e assim é possível aceder ao valor inserido na interface. Deste modo através do método criado, ou seja, esta ligação a partir dos dois métodos *Get* e *Set*, só resta fazer as validações necessárias e com os *inputs* do utilizador criar o registo.

```
//GET USER INPUTS
String nomeProcesso = getNomeProcesso().getValue().toString();
```



```
<af:inputText label="Nome do Processo" id="it1"
binding="#{ProcessoRecrutamento.nomeProcesso}"
columns="50" maxLength="200"/>
```

```
public void setNomeProcesso(RichInputText nomeProcesso) {
    this.nomeProcesso = nomeProcesso;
}

public RichInputText getNomeProcesso() {
    return nomeProcesso;
}
```

```
Row row = createRow();
row.setAttribute("IdProcessoRecrutamento", idProcessoRecrutamento);
row.setAttribute("NomeProcesso", nomeProcesso);
row.setAttribute("FkColaboradorResponsavel", numColabRequerente);
row.setAttribute("FkProjeto", idProjeto);
row.setAttribute("Motivo", motivoPedido);
row.setAttribute("NumeroPessoasNecessarias", numColaboradoresNecessarios);
insertRow(row);
```

Figura 50 - Lógica para o mapeamento de elemento “Nome Processo” entre a JSF e a classe Java Processos de Recrutamento e excerto de código para inserir novo registo

Após a criação do Processo de Recrutamento, voltamos à página principal do Processo de Recrutamento onde é possível visualizar o processo criado. De seguida se consultarmos os detalhes de um processo de recrutamento vamos para o ecrã Detalhes Processo de Recrutamento. Neste ecrã apresentado na Figura 51 podemos visualizar todos os detalhes do processo de recrutamento inclusive o número de pessoas necessárias, as suas competências necessárias e os detalhes do projeto que está associado.

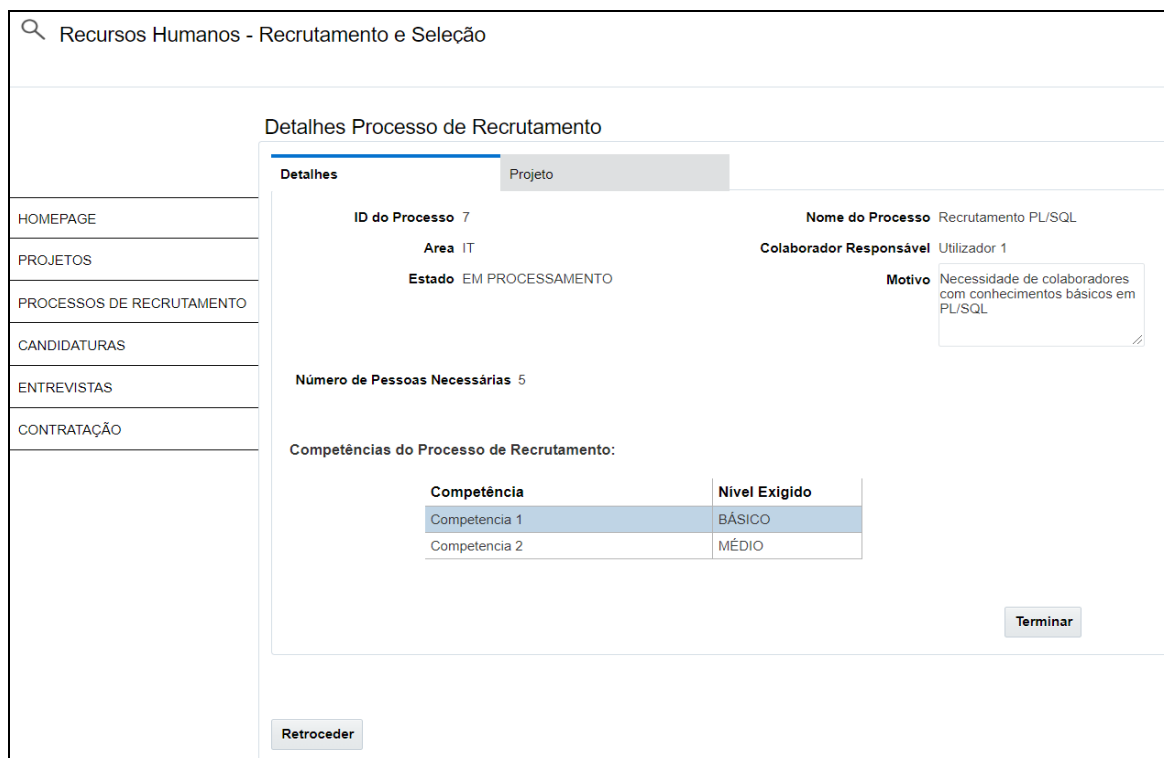


Figura 51 - Ecrã Detalhes Processo de Recrutamento

Após o processo de recrutamento concluído segue-se para o ecrã Candidaturas (Figura 52) através de um *click* no “Menu - Candidaturas”. Cada candidatura presente no SI é composta por um Candidato. Neste ecrã é possível ver todas as candidaturas e os seus estados associados e a partir da mesma é-se direcionado para a criação de uma nova candidatura e surge a possibilidade da visualização ao detalhe de cada candidatura.

Recursos Humanos - Recrutamento e Seleção

Candidaturas

Criar Nova Candidatura

Consultar Candidaturas:

N°	Nome Candidato	Data de Nascimento	Telefone	Email	Processo Recrutamento	Estado Candidato	
6	Madalena Galvão	07-06-1996	912312311	madalenaagalvan@exemplo...	Recrutamento Exemplo	AGENDAR ENTREVISTA	Detalhes
5	Ricardo Antunes	08-04-1992	939393933	ricardo.antunes@exemplo.com	Recrutamento Projeto Fina...	CANDIDATURA EXCLUIDA	Detalhes
4	Manuel Oliveira	01-03-2000	11111123	manueloliveira@exemplo.com	Recrutamento Exemplo 1	CONTRATADO	Detalhes
3	João Alexandre	20-05-1993	91123123	joaoalexandre@exemplo.com	Recrutamento RH	CONTRATADO	Detalhes
2	Maria Camilo	26-12-1994	912531231	mariacamilo@email.com	Recrutamento Gon	CONTRATADO	Detalhes
1	Gonçalo Galvão	10-05-1996	913840724	goncaloribeiro@email.com	Recrutamento Gon	CONTRATADO	Detalhes

Figura 52 - Ecrã de Candidaturas

Ao clicarmos no botão “Criar Nova Candidatura” somos direcionados para a página com o formulário de criação de uma nova candidatura (Figura 53). É neste formulário que é feito o preenchimento de todos os detalhes do candidato e é escolhido o processo de recrutamento associado. De seguida, e após o preenchimento de todos os detalhes, são adicionadas todas as competências do candidato.

Recursos Humanos - Recrutamento e Seleção

Adicionar Candidatura

Nome: Gonçalo Exemplo

E-mail: goncaloexemplocandidatura@exemplo.com

Data de Nascimento: 22-06-1995

Nacionalidade: Portuguesa

Telefone: 913840724

Recomendado Por: []

Processo Recrutamento: Recrutamento Exemplo

Observações: Candidato interessante com alguma experiência

Retroceder Avançar

Adicionar Competências ao Candidato

Pode adicionar várias competências:

Competência: Competencia 1

Nível: MÉDIO

Adicionar Competências

Terminar

Figura 53 - Ecrã Criação de Candidatura e Adiciona Competências Candidato

Após a sua criação, voltamos à página principal das candidaturas onde é possível visualizar a mesma criada. Ao consultarmos Detalhes de uma candidatura, somos direcionados para o ecrã de “Detalhes de Candidatura” onde é possível ver todos os detalhes do candidato, editar o mesmo e avaliar o seu estado (Figura 54). É possível ser selecionado o candidato para entrevista, agendar a mesma ou excluir o candidato alterando assim o seu estado.

Recursos Humanos - Recrutamento e Seleção

Detalhes da Candidatura

Detalhes

Nº Candidatura 8 **Telefone** 913840724

Nome Candidato Gonçalo Exemplo **Email** goncalotesteexemplo@exemplo.com

Data de Nascimento 22-06-1995 **Processo Recrutamento** Recrutamento Exemplo

Idade 24 **Estado do Candidato** PRE SELEÇÃO

Nacionalidade Portuguesa **Recomendado por**

Observações Candidato interessante com alguma experiência

Competências Candidato:

Competencia	Nivel
Competencia 1	MÉDIO
Competencia 2	BÁSICO

Editar Candidato **Carregar CV**

Selecionar para Entrevista **Excluir**

Retroceder

Figura 54 - Ecrã Detalhes de Candidatura

É possível em vários dos ecrãs mostrar apenas a informação de um determinado registo, utilizando uma propriedade denominada *setPropertyListener* onde é permitido selecionar o registo escolhido. No exemplo da Figura 55 o ID do candidato selecionado é guardado numa variável na classe de Java, através desta propriedade, deste modo temos acesso ao ID selecionado dentro da classe e depois toda a Informação é filtrada com base nesse ID e com base nas necessidades de negócio. Desta forma, em todos os ecrãs sempre que foi necessário filtrar informação para um registo em particular, apenas temos de fazer a filtragem nos VOs que são necessários de mostrar. Isso é efetuado através de *View Criterias* que foram criados em alguns VO de maneira a poder filtrar por uma determinada variável num determinado contexto. Na Figura 55, podemos visualizar a filtragem através do id do candidato selecionado onde este é utilizado como *View Criteria* de maneira a apresentar apenas os dados para o mesmo no VO, isto tudo acontece diretamente através da classe de Java utilizada.

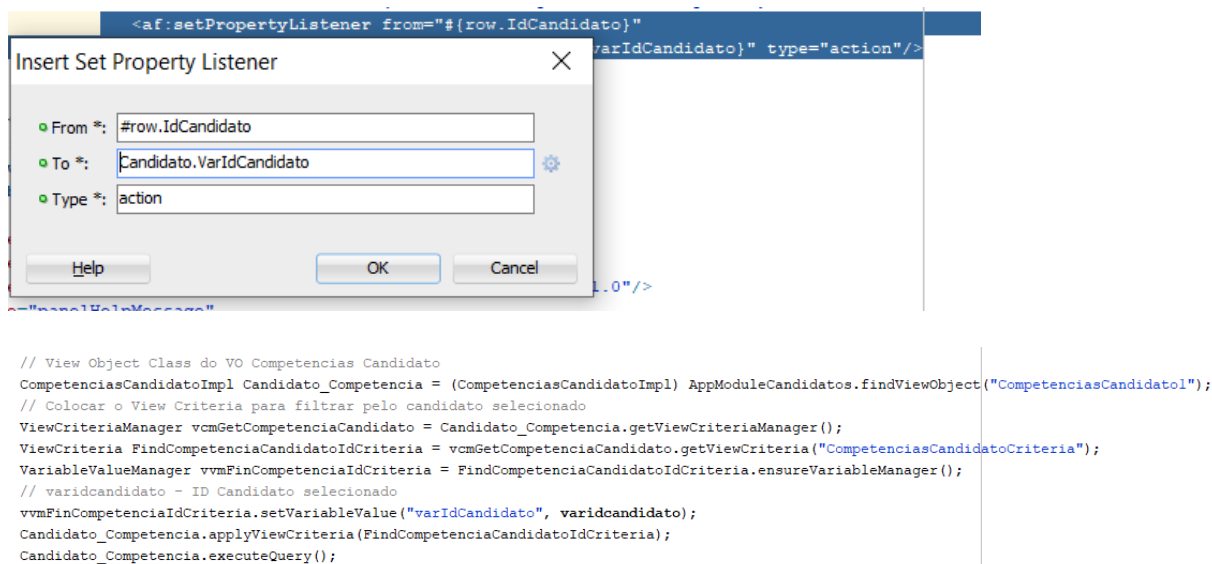


Figura 55 - Exemplo de código para filtrar as competências de um determinado candidato na página Detalhes Candidatura

Ainda no ecrã Detalhes de Candidato quando um candidato é selecionado para entrevista procede-se à marcação da mesma, para tal o utilizador apenas tem que clicar no botão “Agendar Entrevista” e surge um *pop-up* onde é necessário o preenchimento de um formulário com os seguintes campos: Entrevistador, Data, Duração e a sua Localização (Figura 56).

Figura 56 - *Popup* Agendar Entrevista

Após agendarmos uma entrevista, o candidato passa para um estado de “Entrevista Agendada” e a entrevista fica marcada para a data selecionada. Para se realizar a avaliação da entrevista, acede-se ao “Menu - Entrevistas” onde estão apresentadas todas as entrevistas (Figura 57). Neste ecrã existem duas visualizações através de dois painéis (*tabs*), uma com as entrevistas agendadas e outra com as entrevistas que já se realizaram e foram avaliadas.

Recursos Humanos - Recrutamento e Seleção

Entrevistas

Entrevistas em Curso Entrevistas Avaliadas

Candidato	Data	Horas	Localização	Entrevistador	Processo de Recrutamento		
Gonçalo Exemplo	02-09-2020	18	Lisboa	Utilizador 1	Recrutamento Exemplo	Avaliar	Reagendar
Madalena Galvão	10-09-2020	18	Lisboa	Utilizador 1	Recrutamento Exemplo	Avaliar	Reagendar

Figura 57 - Ecrã Entrevistas

Estas duas visualizações de entrevistas em curso e entrevistas avaliadas, visíveis através de uma tabela, são possíveis através da criação de VO com consultas (*queries*) personalizadas para cada um dos painéis (*tabs*), na Figura 58 podemos ver o exemplo do VO com a *query* que permite visualizar as entrevistas ativas.

The screenshot shows a 'Test Query' window with the following SQL code:

```

select distinct
  candidato.nome_candidato,
  entrevista.data_entrevista,
  to_char(entrevista.data_entrevista, 'hh24') as horas,
  entrevista.localizacao as local_entrevista,
  entrevista.expetativa_salarial,
  entrevistador.nome_colaborador as entrevistador,
  processo_recrutamento.nome_processo as processo_recrutamento,
  coalesce(avaliacao.nome_codigo, 'FOR AVALIAR') as avaliacao,
  e_entrevista.nome_codigo as estado_entrevista,
  e_entrevista.fk_estado as id_estado_entrevista,
  entrevista.id_entrevista,
  e_candidato.nome_codigo as estado_candidato,
  e_candidato.fk_estado as id_estado_candidato,
  colaborador.area
from
  entrevista
inner join
  candidato on entrevista.fk_candidato = candidato.id_candidato
left join
  (select ee.fk_estado, cdg.nome_codigo, ee.fk_entrevista from estado_entrevista ee, codigo cdg,

```

Figura 58 – VO com *query* de SQL para Entrevistas Ativas

No ecrã Entrevistas ao clicar no botão “Avaliar” prosseguimos para o ecrã Detalhes/Avaliar Entrevista (Figura 59). Este ecrã está dividido em 3 painéis (*tabs*), o primeiro painel apresenta os detalhes do candidato e os detalhes da entrevista, de maneira a facilitar o entrevistador no enquadramento dos dados sobre os mesmos. No segundo painel é possível adicionar várias perguntas e respostas a esta

entrevista e visualizar as mesmas. Ao longo da entrevista o candidato pode ir respondendo às várias questões e o entrevistador pode marcar as suas perguntas e respostas para ficarem guardadas no sistema e mais tarde serem consultadas caso seja necessário rever essa informação. Por fim no terceiro painel é possível avaliar as entrevistas através de uma *dropdown* onde o avaliador pode classificar as entrevistas conforme a seguinte escala (“Não Suficiente”, “Suficiente”, “Bom” e “Muito Bom”), é também introduzido uma expectativa salarial que o candidato pretende receber e algumas observações. Assim que esteja tudo preenchido apenas tem de submeter a avaliação e de seguida caso pretenda segue o candidato para uma fase de contratação ou pode agendar outra entrevista ou excluir simplesmente o mesmo.

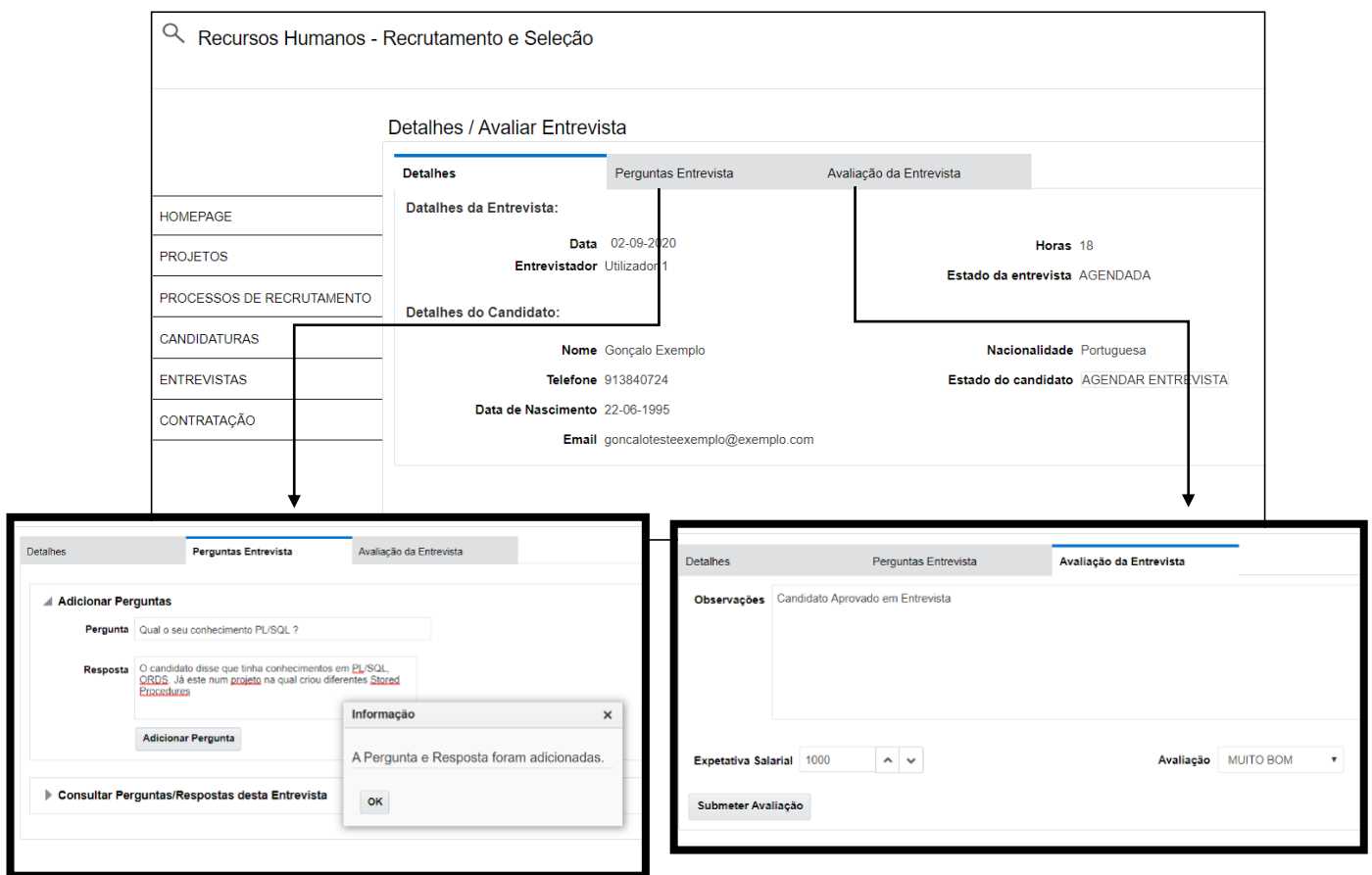


Figura 59 - Ecrã Detalhes / Avaliar Entrevista

No desenvolvimento das páginas sempre que foi necessário realizar operações para criar e actualizar dados foi necessário criar a *entity object* que representa a tabela na BD e simultaneamente é também gerado o VO que representa a mesma *entity*. Este permite ser manobrado através da criação de um VO Class com métodos em Java para inserir e actualizar.

Na Figura 60 podemos visualizar a lógica utilizada, através da criação do VO Class (PerguntasRespostasViewImpl) interligado ao Entity Object “PerguntasRespostas”, tendo sido criado um método simples para inserir novos registos e de seguida apenas foi necessário mapear os diferentes componentes utilizados na JSF com a classe de Java Entrevista, e proceder a chamada do método para inserir a as perguntas e respostas.

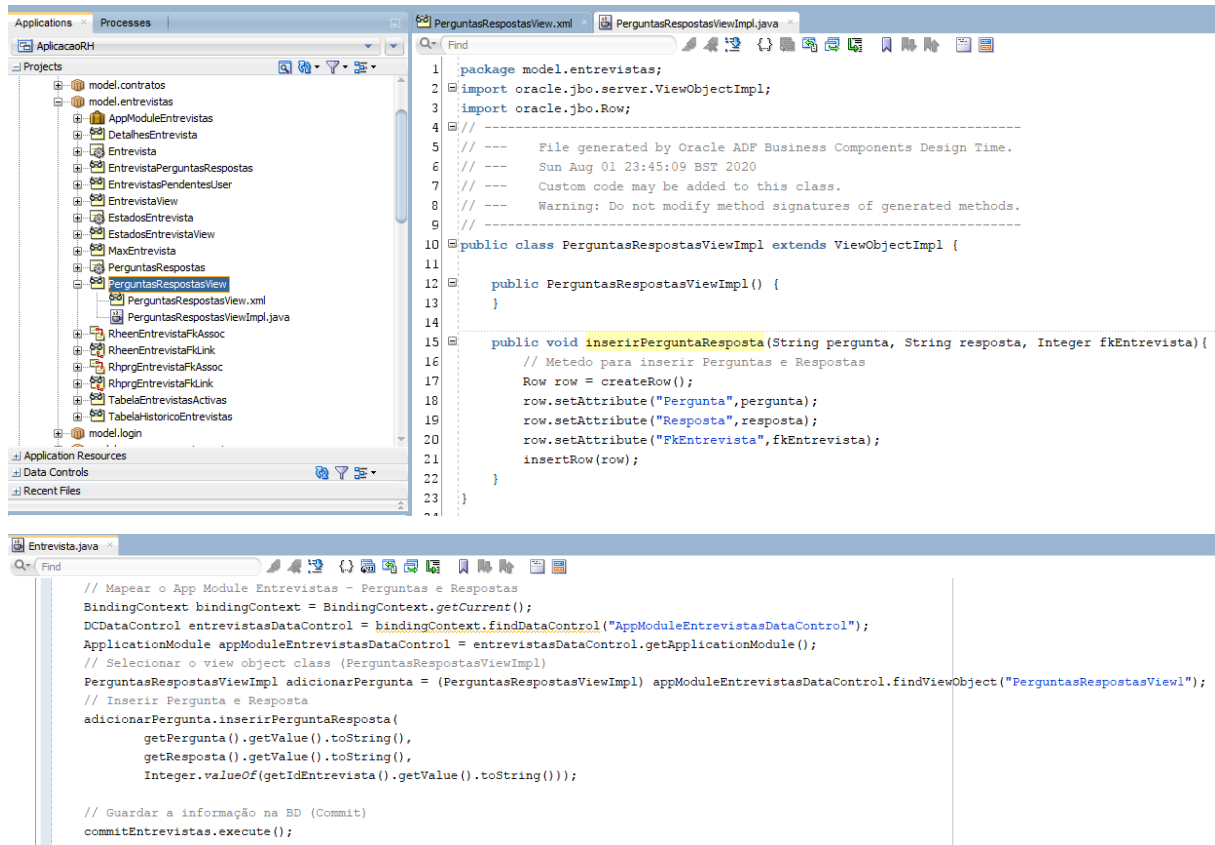


Figura 60 - Lógica para Adicionar Perguntas e Respostas na Entrevista

Por fim, após selecionarmos candidatos para contratação o utilizador apenas tem de clicar no “Menu - Contratação” navegando para a página de Contratação (Figura 61), na qual apresenta um histórico dos pedidos de contratações existentes e o seu estado. A partir deste ecrã podemos criar um novo pedido de contratação ou ver os detalhes e proceder à avaliação do pedido de contratação pretendido.

Recursos Humanos - Recrutamento e Seleção

Contratação

[Criar Novo Pedido de Contratação](#)

Consultar Contratos:

Nº Pedido	Candidato	Processo de Recrutamento	Tipo	Início	Fim	Valor Bruto Mensal	Estado Atual	
82	Gonçalo Exemplo	Recrutamento Exemplo	ESTÁGIO	30/09/2020	10/03/2021	1500	EM ANALISE	Detalhes
81	Madalena Galvão	Recrutamento Exemplo	ESTÁGIO	30/09/2020	10/02/2021	1100	APROVADO	Detalhes
61	Manuel Oliveira	Recrutamento Exemplo 1	SEM TERMO	10/06/2020		1200	APROVADO	Detalhes
41	João Alexandre	Recrutamento RH	ESTÁGIO	29/05/2020	06/05/2021	2000	REJEITADO	Detalhes
21	Maria Camilo	Recrutamento Gon	ESTÁGIO	20/05/2020	13/05/2021	1200	APROVADO	Detalhes
1	Gonçalo Galvão	Recrutamento Gon	ESTÁGIO	14/05/2020	27/05/2021	1000	APROVADO	Detalhes

Figura 61 - Ecrã Contratação

Ao clicarmos no botão “Novo Pedido de Contratação”, iremos para o ecrã Novo Pedido de Contratação (Figura 62). Este ecrã permite criar um pedido de contratação para candidatos que foram selecionados para contratação, sendo apenas necessário selecionar o candidato pretendido e preencher o resto do formulário. O tipo de contrato pode ser Estágio e Sem termo. Caso seja contrato sem termo apenas é necessário preencher o campo Data de Início e o Valor Bruto Mensal, ao clicar em Submeter, o nosso pedido passa para a página principal de contratação como um novo registo.

Recursos Humanos - Recrutamento e Seleção

Novo Pedido de Contratação

Candidato:

Tipo de Contrato: ESTÁGIO

Data Início:

Data Fim:

Valor Bruto Mensal:

[Retroceder](#) [Submeter](#)

Figura 62 - Ecrã Novo Pedido de Contratação

Na Figura 63 podemos ver os componentes utilizados dentro da JSF na construção deste ecrã. Todos os ecrãs desenvolvidos neste SI, têm como base este tipo de desenvolvimento através dos componentes do ADF Faces, facilitando assim a montagem do ecrã contendo uma biblioteca rica em vários componentes.

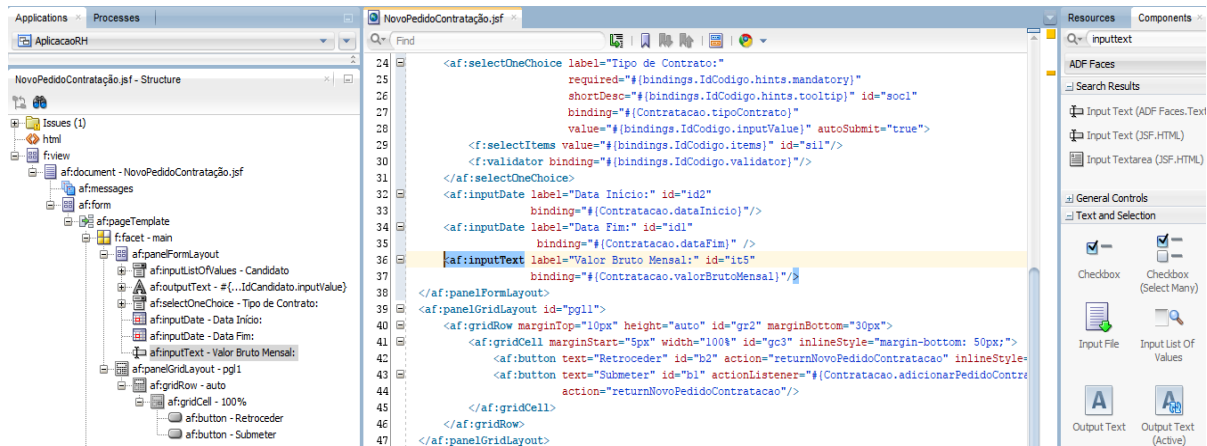


Figura 63 – Interface do JDeveloper retirada da JSF utilizada no ecrã Novo Pedido de Contratação

Por fim, de maneira a concluir o processo de contratação é selecionado o pedido de contratação pretendido, através do *click* no botão “Detalhes”, direcionando para o ecrã Detalhes / Avaliar Pedido de Contratação. Este ecrã permite ver os detalhes do pedido de contratação, o seu estado e as suas observações, se já foi aprovado ou não. O contrato só é aprovado após a decisão da empresa, tendo o SI um botão para aprovar o contrato e caso o mesmo tenha desistido do processo de contratação, ou mesmo o CEO não tenha decidido contratar, o pedido é rejeitado, através de um botão para o efeito. Esta ação vai criar um estado de aprovado, caso seja escolhido aprovar, ou rejeitado caso tenha sido escolhido rejeitar. Deste modo caso o candidato seja aprovado, este no SI passa a ser denominado como contratado através do seu estado, e deste modo o processo de recrutamento é atualizado para necessitar de menos uma pessoa para contratar. Assim que todas as pessoas do processo de recrutamento estejam concluídas o mesmo é terminado.

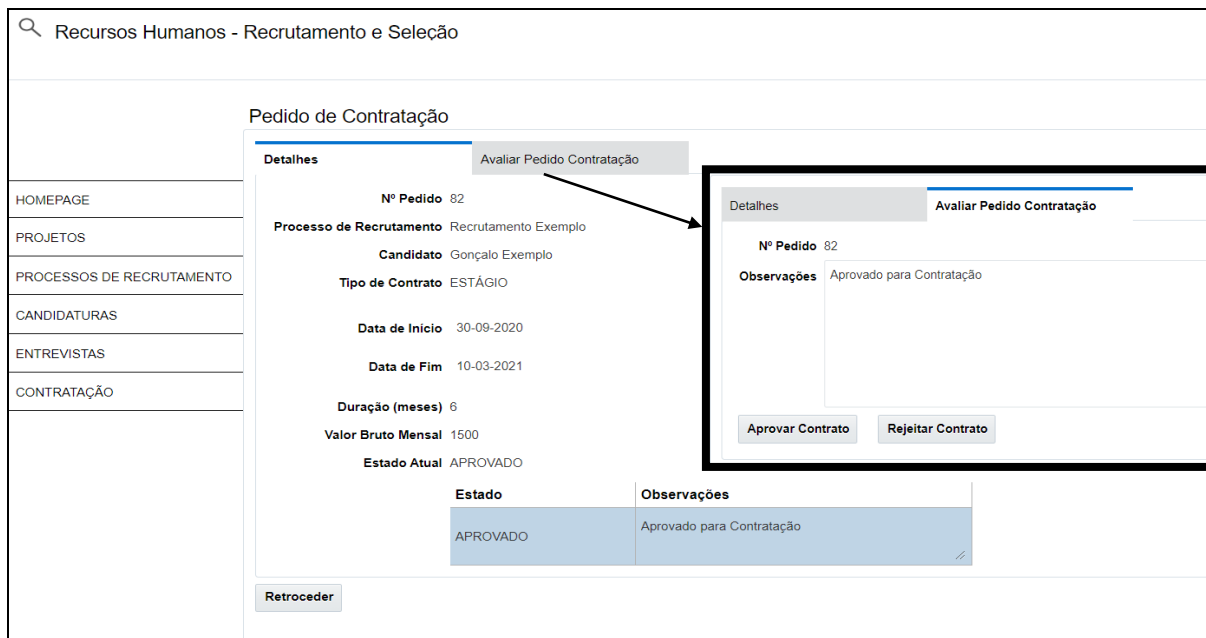


Figura 64 - Ecrã Pedido de Contratação

A lógica por detrás dos estados, como o caso do aprovado ou rejeitado, é efetuada através da criação de um método que permite adicionar um registo na BD com o estado pretendido e que em grande parte este é acionado através dos botões presentes na aplicação assim que pressionados. Na Figura 65 podemos ver como, ao clicar no botão “Aprovar Contrato”, um simples método na classe de Java Contratação é acionado, adicionando os registos necessários na BD, com o estado de aprovado.

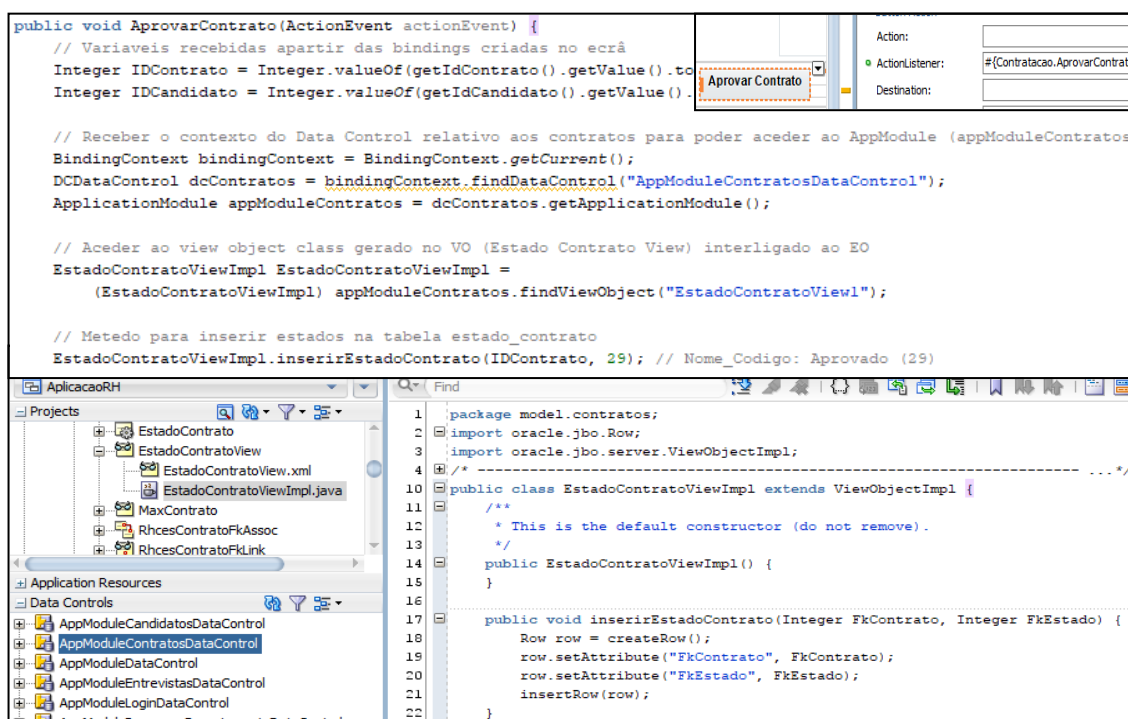


Figura 65 - Código com lógica para Aprovar um contrato na classe Contratos

4.4.4. Testes

Ao longo do projeto, e à medida que foi efetuado o desenvolvimento do sistema, foram feitos vários testes a cada um dos seus componentes, indo ao encontro de um desenvolvimento orientado a testes. Desta forma foi criado um caderno de testes de maneira a testar todas as funcionalidades implementadas, o que permitiu que à medida que fossem desenvolvidas novas funcionalidades, as mesmas fossem testadas, do início ao fim.

Na Tabela 6 podemos verificar o caderno de testes onde contém a descrição do teste, o resultado esperado e o resultado obtido. Este tipo de teste permitiu perceber se o SI estava de acordo com as necessidades e permitiu identificar algum tipo de erro ou bug.

Tabela 6 - Caderno de Testes

Componente	Descrição do Teste	Resultado Esperado	Resultado Obtido
Login e Homepage	Verificar se um utilizador consegue fazer Login e visualizar a Homepage com todos os menus disponíveis.	O utilizador ao inserir no SI o seu <i>username</i> e <i>password</i> , se o mesmo existir na BD deve poder entrar no sistema e ver todos os menus.	O utilizador ao colocar as suas credencias e ao dar sucesso, entra na homepage e consegue ver todos os menus disponíveis.
Login e Homepage	Verificar se um utilizador que não exista na BD ao tentar fazer Login, recebe mensagem com erro.	Ser devolvida mensagem de erro a dizer "Login errado, utilizador ou password errados".	É devolvido uma mensagem de erro a dizer "Login errado, utilizador ou password errados".
Projetos	Verificar se a formatação permite uma boa apresentação e o acesso a todo a página.	Permite ver toda a informação na vertical e na horizontal, permitindo o acesso a todas as opções.	É possível ver toda a informação na vertical e horizontal permitindo o acesso a todas as opções.
Projetos	Verificar se permite editar e atualizar os projetos existentes (ecrã e base de dados).	Permite editar e atualizar os projetos existentes (ecrã e base de dados).	Permite editar e atualizar os projetos existentes (ecrã e base de dados).
Projetos	Verificar se permite criar um novo projeto com sucesso (ecrã e base de dados).	Permite criar um novo projeto com sucesso (ecrã e base de dados).	Permite criar um novo projeto com sucesso (ecrã e base de dados).
Processos de Recrutamento	Verificar se a formatação permite uma boa apresentação e o acesso a todo a página.	Permite ver toda a informação na vertical e na horizontal, permitindo o acesso a todas as opções.	É possível ver toda a informação na vertical e horizontal permitindo o acesso a todas as opções.
Processos de Recrutamento	Verificar se é possível filtrar e consultar os detalhes de um Processo de Recrutamento.	Permite filtrar um Processo de Recrutamento e permite consultar os detalhes desse processo.	Permite filtrar um Processo de Recrutamento e permite consultar os detalhes desse processo.
Processos de Recrutamento	Verificar se o Processo de Recrutamento foi criado com sucesso (ecrã e base de dados).	O Processo de Recrutamento é criado com sucesso (ecrã e base de dados).	O Processo de Recrutamento é criado com sucesso (ecrã e base de dados).

Processos de Recrutamento	Verificar se o processo de recrutamento foi terminado com sucesso (ecrã e base de dados).	Ao clicar no botão "terminar", o processo de recrutamento é terminado com sucesso (ecrã e base de dados).	Ao clicar no botão "terminar", o processo de recrutamento é terminado com sucesso (ecrã e base de dados).
Processo de Recrutamento	Verificar se permite adicionar/submeter uma competência ao Processo (ecrã e base de dados).	A Competência é adicionada com sucesso ao processo (ecrã e base de dados).	A competência é adicionada com sucesso ao processo (ecrã e base de dados).
Processos de Recrutamento	Verificar se é possível visualizar os detalhes do Projeto associado ao Processo de Recrutamento.	É possível visualizar toda a informação do Projeto associado ao Processo de Recrutamento.	É possível visualizar toda a informação do Projeto associado ao Processo de Recrutamento.
Candidaturas	Verificar se a formatação permite uma boa apresentação e o acesso a toda a página.	Permite ver toda a informação na vertical e na horizontal, permitindo o acesso a todas as opções.	Permite ver toda a informação na vertical e na horizontal, permitindo o acesso a todas as opções.
Candidaturas	Verificar se é possível filtrar e consultar os detalhes de uma Candidatura.	Permite filtrar um Processo de Recrutamento e permite consultar os detalhes desse processo.	Permite filtrar um Processo de Recrutamento e permite consultar os detalhes desse processo.
Candidaturas	Verificar se a Candidatura foi criada com sucesso (ecrã e base de dados).	A Candidatura é criada com sucesso (ecrã e base de dados).	A Candidatura é criada com sucesso (ecrã e base de dados).
Candidaturas	Verificar se permite selecionar um candidato para entrevista.	Permite selecionar um candidato para entrevista.	Permite selecionar um candidato para entrevista.
Candidaturas	Verificar se permite agendar entrevista para um candidato (ecrã e base de dados).	A Entrevista é agendada com sucesso (ecrã e base de dados).	A Entrevista é agendada com sucesso (ecrã e base de dados).
Candidaturas	Verificar se permite editar detalhes do candidato.	Permite editar detalhes do candidato.	Permite editar detalhes do candidato.
Candidaturas	Verificar se permite excluir candidato.	Candidato excluído com sucesso.	Candidato excluído com sucesso.
Candidaturas	Verificar se permite adicionar uma competência a um candidato (ecrã e base de dados).	A Competência é adicionada com sucesso ao candidato (ecrã e base de dados).	A Competência é adicionada com sucesso ao candidato (ecrã e base de dados).
Entrevistas	Verificar se a formatação permite uma boa apresentação e o acesso a toda a página.	Permite ver toda a informação na vertical e na horizontal, permitindo o acesso a todas as opções.	Permite ver toda a informação na vertical e na horizontal, permitindo o acesso a todas as opções.
Entrevistas	Verificar se é possível filtrar e prosseguir para a avaliação de uma entrevista.	Permite filtrar uma entrevista e permite prosseguir para a avaliação de uma entrevista.	Permite filtrar uma entrevista por todos os campos apresentados, exceto por "Data e Hora" e permite com sucesso prosseguir para a avaliação de uma entrevista.
Entrevistas	Verificar se permite ver as entrevistas já realizadas.	Permite ver as entrevistas já realizadas.	Permite ver as entrevistas já realizadas.
Entrevistas	Verificar se permite avaliar uma entrevista.	A avaliação da entrevista é realizada com sucesso (ecrã e base de dados).	A avaliação da entrevista é realizada com sucesso (ecrã e base de dados).

Entrevistas	Verificar se permite adicionar perguntas e respostas (ecrã e base de dados).	Permite adicionar perguntas e respostas com sucesso (ecrã e base de dados).	Permite adicionar perguntas e respostas com sucesso (ecrã e base de dados).
Entrevistas	Verificar se permite selecionar candidato para fase de contratação.	O candidato é selecionado para a fase de pedido de contratação com sucesso.	O candidato é selecionado para a fase de pedido de contratação com sucesso.
Entrevistas	Verificar se permite reagendar entrevista.	A entrevista é reagendada com sucesso.	A entrevista é reagendada, mas é necessário fazer <i>refresh</i> na página Entrevistas.
Contratação	Verificar se a formatação permite uma boa apresentação e o acesso a toda a página.	Permite ver toda a informação na vertical e na horizontal, permitindo o acesso a todas as opções.	Permite ver toda a informação na vertical e na horizontal, permitindo o acesso a todas as opções.
Contratação	Verificar se permite filtrar e consultar os detalhes e avaliação de um pedido de contratação.	Permite filtrar e consultar os detalhes e avaliação de um pedido de contratação.	Permite filtrar e consultar os detalhes e avaliação de um pedido de contratação.
Contratação	Verificar se é possível criar um novo pedido de contratação (ecrã e base de dados).	O pedido de contratação é criado (ecrã e base de dados).	O pedido de contratação é criado (ecrã e base de dados).
Contratação	Verificar se permite aprovar pedido de contratação.	Pedido de contratação aprovado com sucesso.	Pedido de contratação aprovado com sucesso.
Contratação	Verificar se permite rejeitar pedido de contratação.	Pedido de contratação rejeitado com sucesso.	Pedido de contratação rejeitado com sucesso.

Com base neste caderno de testes foi assegurado o funcionamento da aplicação, indo desta forma ao encontro dos seus requisitos funcionais. Permitiu validar se a informação estava a ser persistida na BD e foram identificados e corrigidos alguns erros, assim como acrescentadas validações para corrigir possíveis erros no que toca ao formato dos dados.

Por fim foram realizados testes de usabilidade no sistema, de modo a perceber se a experiência do utilizador era bem recebida e para compreender a interação com as diversas funcionalidades em funcionamento no sistema. Os resultados dos testes foram positivos, e os utilizadores partilharam algum *feedback*, na medida em que foram identificadas possíveis novas funcionalidades a serem discutidas e implementadas no futuro.

5. CONSIDERAÇÕES FINAIS

5.1. SÍNTESE

O presente projeto surgiu através da verificação da necessidade de um SI para a gestão do recrutamento e seleção em consultoria. Deste modo foi realizado um estudo das áreas de GRH, consultoria, SI e o desenvolvimento de software particularmente o conjunto de metodologias de desenvolvimento de software existentes. Este estudo permitiu compreender melhor estas áreas, e perceber os conceitos adjacentes às mesmas.

Este estudo permitiu que na fase do desenvolvimento do projeto, fossem tomadas as melhores decisões e fosse possível alavancar todo o desenvolvimento envolvido. A parte prática do projeto iniciou-se através do levantamento e análise dos requisitos funcionais, tendo sido recolhidas todas as necessidades funcionais para a necessidade identificada. O levantamento das necessidades originou e permitiu criar a base para posteriormente se desenvolver através do desenho e da análise do sistema.

Ao longo desta fase prática foi necessário identificar as linguagens de programação a serem utilizadas no projeto, de maneira a perceber e a interpretar a melhor forma de desenvolvimento do sistema. O estudo das metodologias de desenvolvimento de software, também permitiram adotar boas práticas no seu desenvolvimento. Todas estes conhecimentos e estudos permitiram selecionar a melhor arquitetura a ter em conta e implementar toda a parte de programação do sistema. Nesta fase, foi realizado a BD relacional e foram criados os diferentes écrans e lógicas necessárias. O SI à medida que foi sendo desenvolvido foi também sendo testado e originou um caderno de testes com vários tipos de testes ao longo dos diferentes processos adjacentes permitindo assegurar o correto funcionamento da aplicação. O resultado final do desenvolvimento do sistema permite assegurar uma forma eficaz da gestão do RSC na consultora em estudo, permitindo assim colmatar a necessidade identificada.

5.2. CONTRIBUIÇÕES DO PROJETO

Este projeto contribuiu para analisar e estudar quais as necessidades existentes no ramo em questão e desta forma desenvolver algo que pudesse colmatar essa necessidade. Logo o resultado final deste projeto foi o principal contributo tanto para a empresa em questão como também para outras que possam seguir o exemplo. Todos os objetivos do projeto foram alcançados e as suas funcionalidades foram criadas de acordo com as especificações identificadas para a organização em estudo. O projeto foi bem recebido pelos utilizadores nos testes de usabilidade e vai dar mais eficiência e eficácia à organização, permitindo ter acesso a um sistema inovador.

5.3. LIMITAÇÕES E TRABALHO FUTURO

Uma limitação neste projeto foi em questões temporais, pois o limite de tempo estabelecido pode ter prejudicado a complementação mais aprofundada do desenvolvimento do projeto. É necessário ter em conta que como este é um projeto de dimensão curta, não está preparado para grandes volumes de dados, podendo vir a ter custos adicionais.

Existiram algumas limitações relacionadas com a tecnologia em uso, tendo sido identificados alguns erros e bugs que posteriormente foram corrigidos. No que toca ao produto final, o mesmo responde a todas as necessidades pré-estabelecidas, mas no futuro pode existir a necessidade de criação de novas funcionalidades de modo a complementar os processos existentes.

Outra limitação é a necessidade de manutenção ao longo do tempo. Num trabalho futuro uma das funcionalidades a ser implementada é uma página de administração para permitir que um utilizador possa configurar alguns detalhes relacionados com os processos já implementados. O sistema apresenta-se pronto para ser alterado de uma forma simples, acrescentando ou retirando partes, permitindo que a organização dê continuidade ao mesmo.

BIBLIOGRAFIA / REFERÊNCIAS BIBLIOGRAFICAS

- Agile Manifesto (2001). Princípios do Manifesto Ágil, (<https://agilemanifesto.org/iso/ptpt/principles.html>), acessado em 27 de dezembro de 2019.
- Anderson, D. (2010) *Kanban: Successful Evolutionary Change for Your Technology Business*, Blue Hole Press.
- Armstrong, M. & Taylor, S. (2014). *Armstrong's handbook of human resource management practice*. United Kingdom: Kogan-Page.
- Bedwell, W., Wildman, J., Diazgranados, D., Salazar, M., Krames, W. & Sala, E. (2012). Collaboration at work: An integrative multilevel conceptualization. *Human Resource Management Review*, 22, 128-145.
- Beck, K. & Fowler, M. (2000) *Planning Extreme Programming*, Addison Wesley, 1ªed.
- Bilhim, J. (2006) *Teoria Organizacional: Estruturas e Pessoas*. Instituto Superior de Ciências Sociais e Políticas. Universidade Técnica de Lisboa, 5ªed., Lisboa.
- Boehm, B. & Turner, R. (2003). *Balancing Agility and Discipline: A Guide for the Perplexed*. Boston: Addison-Wesley.
- Booch G., Jacobson, I. & Rumbaugh J. (1999). *Unified Modeling Language -User's Guide*, Addison-Wesley.
- Camara, P.; Guerra, P. & Rodrigues, J. (2010). *Novo humanator*. Alfragide: Publicações Dom Quixote
- Canback S. (1998), *The logic of management consulting (part one)*, *Journal of Management Consulting* 10 (2).
- Chiavenato, I. (1995). *Recursos humanos*. São Paulo: Atlas.
- Chraif, M. (2013), *Work Psychology Treaty - Practical applications in organizations and human resources*, Bucharest.
- Cockburn, A. (2004). *Crystal Clear: A Human-Powered Methodology for Small Teams*. Boston: Addison-Wesley.
- Conboy, K., (2009). Agility from first principles: reconstructing the concept of agility in information systems development. *Information Systems Research*, 20, 329–354.

- Costa, S., Neto, A. & Bazzoli, L. (2016). Executivos da Geração Y: Sucessos e Fracassos na Retenção de Trainees. *Revista gestão organizacional*, 12(1), 80-92.
- Currenti, S.; Borra, S. & Appolloni, A. (2019). *Survey of the EUROPEAN MANAGEMENT CONSULTANCY. European federation of management consultancies associations*, p1-45 FEACO (2019).
- Engholm (2010). *Engenharia de Software na Prática* - Hélio Engholm Jr.
- Fatemi, F. (2016). The True Cost Of A Bad Hire - It's More Than You Think in Forbes, (<https://www.forbes.com/sites/falonfatemi/2016/09/28/the-true-cost-of-a-bad-hire-its-more-than-you-think/#42af64c34aa4>), acessado em 18 de junho de 2019.
- Fehily, C. (2014). *SQL - Database Programming*, Questing Vole Press.
- Fowler, M. & Highsmith, J. (2001) *The agile manifesto, Software Development*.
- Greiner, L. & Metzger, R. (1983), *Consulting to Management*, Prentice-Hall: Englewood Cliffs, New Jersey.
- Highsmith, J. (2002) - *Agile Project Management*. Boston: Pearson Addison-wesley, 2002.
- Karlsson, M. (2014). *Enabling virtual communities of practice: a case-study of Swedish-Indian collaboration in IT development*.
- Kroenke, D. M. (2008) *Experiencing MIS*, Pearson Prentice Hall, Upper Saddle River, NJ.
- Laudon, K. & Laudon, J. (2012). *Management information systems: managing the difital firm*. England:Pearson.
- Larman, C. (2002). *Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design and the Unified Process*. Englewood Cliff, NJ: Prentice Hall.
- LinkedIn (2018). Tendências globais de recrutamento para 2018 in LinkedIn, (<https://business.linkedin.com/pt-br/talent-solutions/recruiting-tips/tendencias-globais-de-recrutamento-para-2018>), acessado em 18 de junho de 2019.
- Michel, M. (2007). Tipos de recrutamento e sua importância para uma gestão adequada de pessoas aplicadas a empresas. *Revista científica eletrônica de administração*, 13.
- Moreira, M. (2011). *Metodologia de Desenvolvimento de Software – RUP*. Pitágoras.

O'Brien, J.A. (1993) *Management Information Systems: A Managerial End User Perspective*, Irwin.

Oliveira, L.N.Q., Fonseca, P.S., & Bonadiman, T.K.P. (2014). Consultoria empresarial: O diferencial nas empresas modernas.

Pressman, R. (2010). *Software engineering: a practitioner's approach*. New York: McGraw-Hill companies.

Rational Unified Process Best Practices for Software Development Teams in IBM (1998), (https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf), acessado a 18 de junho de 2019.

Robertson, I.T.; Smith, M. (2001) *Personnel selection. Journal of Occupational and Organizational Psychology* 74 (4), 441-472.

Ronald, G. (2011). Quick Start Guide to Oracle Fusion Development: Oracle JDeveloper and Oracle ADF. McGraw Hill: US

Schuster, W. E. & Friedrich, M. P. A. (2019). A Importância da Consultoria Empresarial na Gestão Financeira das Micro e Pequenas Empresas. *Revista de Administração IMED*, Passo Fundo, v. 7, n. 2, p. 183-205, dez. 2017. ISSN 2237-7956.

Serrano, A.; M. C. & Guerreiro, A. (2004), *Gestão de Sistemas e Tecnologias de Informação*. FCA – Editora de Informática.

Schildt, H., (2007), *Java™: The Complete Reference, Seventh Edition*, The McGraw-Hill Companies.

Schwaber, K. and Beedle, M. (2001). *Agile Software Development with Scrum*. Englewood Cliffs, NJ: Prentice Hall.

Sommerville, I. (2011). *Software Engineering*, 9th Edition. Pearson Addison Wesley.

The Economist (2017). The world's most valuable resource is no longer oil, but data, (<https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data>) acessado em 28 de dezembro de 2019.

Varajão, J. E. Q. (1998) *A Arquitetura da Gestão de Sistemas de Informação*, FCA.

Vasco, C., Vithoft, M., Estante, C. (2011). Comparação entre Metodologias RUP e XP. Materiais de aprendizagem de Fundamentos de Engenharia de Software da Programa de Pós Graduação em Informática Aplicada (<http://www.ppgia.pucpr.br/~alcides/Teaching/mestrado/FundamentosEngenh>

[ariaSoftware/artigos/ResumosApresentacoes/RUPvsXP_draft.pdf](#)), acessido em 18 de junho de 2019.

Vieira, R. (2019). Sete tendências no futuro do recrutamento in Sapo, (<https://eco.sapo.pt/2019/05/30/sete-tendencias-no-futuro-do-recrutamento/>), acessido em 18 de junho de 2019.

Warrick, D.D. (2017). What leaders need to know about organizational culture, *Business Horizons*: 60, 395-404.

ANEXOS

A. Modelo Físico de Dados

Diagram: Modelo de Dados Físico
 Author: Gonçalo
 Created on: 2020-04-15 15:40:48 UTC

