



Mercados de Talento Utilizando Inteligência Artificial

DANIEL FILIPE NUNES OLIVEIRA

outubro de 2022



Mercados de Talento Utilizando Inteligência Artificial

Daniel Filipe Nunes Oliveira

Aluno nº 1171139

**Dissertação para obtenção do Grau de
Mestre em Engenharia de Inteligência Artificial**

Orientador: Dr. Isabel Cecília Correia da Silva Praça Gomes Pereira, Professora Coordenadora no Instituto Superior de Engenharia do Porto;

Apoio de Supervisão: Eng. Hugo Pereira, Desenvolvedor de Software na KonkConsulting

Júri:

Presidente

Doutor António Constantino Lopes Martins, Professor Adjunto do Instituto Superior de Engenharia do Instituto Politécnico do Porto

Vogais:

Doutor José Manuel Ferreira Machado, Professor Associado com Agregação da Universidade do Minho (arguente)

Doutora Isabel Cecília Correia da Silva Praça Gomes Pereira, Professora Coordenadora do Instituto Superior de Engenharia do Instituto Politécnico do Porto (orientadora)

Porto, outubro de 2022

Dedicatória

Dois anos que passaram a voar desde a última vez que escrevi uma dedicatória. Dois anos que me marcaram muito e que apenas consigo recordar-me deles com um sorriso na boca. Dois anos desde que conheci uma rapariga/mulher que me mudou e orgulho-me dela imensamente, inspiras-me, orgulhas-me, e a cada dia que passa mais te amo, quero conhecer o Mundo contigo, rir, brincar, jogar e sobretudo ser feliz. Seja isso a viajar todos os fins de semana ou a comer um hambúrguer no nosso cantinho, seja como for, sou feliz contigo! À minha mãe e padraсто, obrigado por tudo, esta vitória, que espero que seja mais uma, também é vossa, nunca o conseguiria sem os vossos valores de resiliência, persistência e ambição. À minha avó, a minha confidente que tem todos os fins de semana, mesmo com a vida complicada e entraves de saúde que enfrenta, o leite-creme e assado à minha espera, amo-te para sempre. Aos meus “irmãos” José, Diogo e Inês que continuam a ser um motivo de orgulho enorme e que me ensinaram que família não é só a de sangue. À minha avó Rosa, não és avó de sangue, mas és de coração e amo-te ao mesmo nível como se fosses de sangue! À minha madrinha, padrinho, prima, primos, tios que adoro incondicionalmente. Aos meus colegas da licenciatura que ainda todos os dias falamos. Aos meus amigos de 4 patas, Ellie e Benji.

Dedico a todos esta pinga de suor, que apenas é possível devido a todas as que derramaram por mim. Mas dedico, especialmente, a ti avô. Dois anos que partiste. Sabes que não acredito no além, mas sei que enquanto a tua memória e os teus ensinamentos durarem, és eterno. “Venceremos, venceremos...”

A todos, adoro-vos!

Resumo

Os recursos humanos são um dos maiores ativos de qualquer empresa visto que estes providenciam a possibilidade para realizar produtos ou serviços. A revolução tecnológica, a pandemia do COVID-19 e a competitividade do mercado laboral contribuem para um clima de incerteza e permanente renovação de staff dentro das empresas. Isto significa permanências curtas dos funcionários, mas, mais importante, leva a que os repositórios de competências de uma empresa possam ficar, por vezes, empobrecidos e, deste modo, pode pôr em causa a execução dos produtos e serviços pelos quais uma empresa é reconhecida. Recentemente, têm surgido plataformas online com o objetivo de atrair, designar funções, treinar, mas sobretudo reter os talentos, tudo isto só é alcançado revendo e melhorando permanentemente as competências de cada recurso. Estas plataformas utilizam quase sempre mecanismos de inteligência artificial.

Neste trabalho, apresenta-se uma revisão literária das técnicas de inteligência artificial que podem estar presentes nestas plataformas. Esta revisão literária apresenta 4 questões de investigação que, juntas, respondem a uma questão de investigação mais ampla: “Como implementar uma destas plataformas, sendo inovador?”. A primeira questão é relativa aos processos automatizados de leitura e extração de informação de currículos. A segunda questão é relativa à inferência de competências, através de outras competências ou informações previamente extraídas de currículos, e é neste ponto que se pretende inovar quando em comparação com as soluções existentes. A terceira questão é relativa à existência de sistemas multiagente que associam recursos a tarefas de uma forma otimizada. Por último, a quarta questão de investigação é relativa à aplicabilidade de algoritmos genéticos também na associação de funcionários a tarefas numa empresa.

Os resultados da revisão literária mostram que as 4 questões foram respondidas com sucesso. A revisão literária seguiu a metodologia PRISMA, tendo sido realizada a pesquisa em 2 fontes distintas. Foram selecionados 44 artigos, de entre os 27361 encontrados, que foram filtrados através de controlo de qualidade ou leituras rápidas de títulos e resumos.

Por último, tendo em consideração a revisão literária realizada, foi implementado e testado um sistema que se assemelha a um mercado de talentos, com inferência de competências. O emparelhamento entre recursos e tarefas foi feito através de um sistema multiagente e de um algoritmo genético. Foi, ainda, realizada uma análise de usabilidade.

Desta forma, podemos concluir que o resultado final foi atingido, pois quer a revisão literária quer o sistema implementado foram atividades realizadas com sucesso.

Palavras-chave: Recursos Humanos, Algoritmo Genético, Sistemas Multiagente; Aprendizagem Profunda; Processamento de Língua Natural

Abstract

Human resources are one of the biggest assets in companies since they possess the skills and expertise to deliver products and services. The COVID-19 pandemic and the technological revolution both increased employee turnover to a level where companies can hardly keep up with the pace, leading to worst talents management. Recently, online platforms, known as talent markets, have become more and more popular and they have the main objective to attract, designate tasks, train and, above all, retain existing employees. Most of these online platforms use artificial intelligence.

This work presents a systematic review in artificial intelligence techniques that allow automatization of the processes of talent management. Four research questions were elaborated that, altogether, answer a broader research question which is: "How to implement an innovative talent market?". The first question is relative to the automatized processes of information extraction out of resumes. The second question is related to the skill inference process, it is here that innovation is expected when comparing with existing solutions. In the third and fourth research questions, literature over multiagent systems and genetic algorithms dedicated to the optimization of task execution is provided. In the review, 44 papers were selected out of an initial set of 27361.

In addition to the systematic review, a system is also proposed that resembles already existing solutions. Innovation is achieved by exploring skill inference, in addition to using already existing techniques in the area of information recognition. A multiagent system and genetic algorithms were also developed for an optimized task and employee pairing. This document also presents various tests to the system including a usability analysis.

All in all, the outcome was rewarding, the systematic literature review was a success and so was the proposed solution.

Keywords: Human Resources; Genetic Algorithm; Multiagent Systems; Deep Learning; Natural Language Processing

Agradecimentos

Em primeiro lugar, gostaria de agradecer à minha orientadora, a professora Isabel Praça, e ao meu supervisor na KonkConsulting, Hugo Pereira, pela disponibilidade e acompanhamento que me ofereceram durante a realização desta tese de mestrado. Os meus agradecimentos estendem-se, naturalmente, à minha empresa, a KonkConsulting, por toda a flexibilidade oferecida não só durante a realização da tese de mestrado, mas também pela mesma, oferecida ao longo dos semestres com unidades curriculares.

Quero agradecer, ainda, a todos os professores do Departamento de Engenharia Informática e, em particular, a todo o corpo espetacular de docentes pelo qual o Mestrado em Engenharia de Inteligência Artificial é composto. Vocês motivam, esforçam-se e fazem-nos querer saber mais. Tudo isto nota-se e é um orgulho ter ingressado neste mestrado!

A todos, por tudo, muito obrigado!

Índice

1. Introdução	1
1.1 Enquadramento da Dissertação	1
1.2 Contextualização do Problema	1
1.3 Contributos e Objetivos do Projeto	3
1.4 Planeamento do Trabalho	4
1.5 Estrutura da Dissertação	4
2. Revisão Sistemática	7
2.1 Metodologia de Pesquisa	7
2.1.1 Questões de Investigação	7
2.1.2 Fontes de pesquisa	8
2.1.3 Termos de Pesquisa	8
2.1.4 Critérios de Inclusão e Exclusão	10
2.1.5 Extração de dados	10
2.2 Resultados da Pesquisa	12
2.2.1 “Quais são as principais técnicas de IA para a extração automatizada de informação de currículos”	12
2.2.2 “Quais são as principais técnicas de IA utilizadas para inferir competências de recursos?”	13
2.2.3 “Existe algum SMA usado para emparelhar empregados com empregos/tarefas?”	14
2.2.4 “Existem algoritmos genéticos utilizados para emparelhar empregados com empregos/tarefas?”	15
2.3 Discussão dos Resultados	16
2.3.1 “Quais são as principais técnicas de IA utilizadas para a extração automatizada de informação de currículos?”	16
2.3.2 “Quais são as principais técnicas de IA para inferir competências de recursos?”	18
2.3.3 “Existe algum MAS usado para emparelhar candidatos com empregos/tarefas?”	19
2.3.4 “Existem algoritmos genéticos utilizados para emparelhar empregados com empregos/tarefas?”	21
3. Análise e Desenho da Solução Proposta	25
3.1 Tecnologias Utilizadas	25
3.2 Arquitetura do Sistema	26
3.3 Domínio Proposto	27
3.4 Casos de Uso do Sistema	29
3.5 Arquitetura da BD	30
3.6 Datasets de treino	32
3.6.1. Informações Gerais do Dataset da Extração de Informação de Currículos	32
3.6.2. Informações Gerais do Dataset dedicado à inferência de competências	32
3.7 Questões de Segurança - Como ultrapassar?	33

3.8 Modelos Utilizados para a extração e inferência de competências	34
3.8.1 Extração de Informações de um currículo	35
3.8.2 Inferência de competências	35
3.9 Arquitetura dos sistemas multiagente	35
3.9.1 Sistema Multiagente responsável pela integração com SAP	36
3.9.2 Sistema Multiagente responsável pelo processo de alocação de funcionários e recomendação de candidatos.....	36
3.10 Algoritmo Genético responsável pelo emparelhamento entre empregados e tarefas	38
4. Implementação.....	39
4.1 Pré-Processamento dos Datasets	39
4.1.1. Pré-Processamento do Dataset da Extração de Informação de Currículos	39
4.1.2. Pré-Processamento do Dataset da Inferência de Competências	40
4.2 Extração automatizada de informação de currículos	41
4.3 Inferência de competências	45
4.4 Sistemas Multiagente desenvolvido	46
4.4.1 Sistema Multiagente dedicado ao escalonamento do talento interno numa empresa	47
4.4.2 Sistema Multiagente dedicado à integração com SAP	51
4.4.3. Integração com Spring	52
4.5 Algoritmo Genético para emparelhamento de empregados e tarefas.....	53
4.6 Bases de Dados	55
4.6.1 Scripts Flyway e integração com Spring	55
4.6.2 Mapeamento das Entidades de Domínio com as Tabelas de Bases de Dados	56
4.7 Backend	58
4.8 Frontend	59
4.8.1. Registo de Candidato	59
4.8.2. Gestão de Empresas do Recrutador.....	60
4.8.3. Integração dos dados com o sistema SAP	62
4.8.4. Correr o sistema multiagente de alocações automáticas.....	62
4.9 Módulo Python de Inteligência Artificial.....	63
5. Testes.....	65
5.1 Testes Backend.....	65
5.2 Testes Modelos de IA	65
5.3 Análise de usabilidade	66
6. Conclusão	71
6.1 Conclusão	71
6.2 Objetivos concretizados	72
6.3 Limitações e Trabalho Futuro	73
6.4 Apreciação Final	74

7. Referências	75
Anexo A - Questionários de Usabilidade	81
Anexo B - Guia de Sessão para Análise de Usabilidade	84

Lista de Figuras

Figura 1 - Diagrama PRISMA	11
Figura 2 – Distribuição de Artigos para a Primeira Questão de Investigação	12
Figura 3 – Distribuição de Artigos para a Segunda Questão de Investigação	13
Figura 4 - Distribuição de artigos da Terceira Questão de Investigação.....	15
Figura 5 - Distribuição dos artigos da Quarta Questão de Investigação	16
Figura 6 - Função de <i>fit</i> do algoritmo genético implementado no contexto de emergências hospitalares.....	22
Figura 7 – Comparação entre performance do algoritmo genético e soluções usando programação linear	24
Figura 8 - Vista da arquitetura do sistema	26
Figura 9 - Domínio da Solução	28
Figura 10 - Diagrama de Casos de Uso	29
Figura 11 – Modelo Relacional da Solução	31
Figura 12 – Informações gerais do <i>dataset</i> dos Currículos.....	32
Figura 13 - Informações gerais do <i>dataset</i> de linguagens	33
Figura 14 - Diagrama de sequência da extração de competências usando o módulo de IA	34
Figura 15 - Figura dedicada à arquitetura do SMA desenvolvido para a integração com SAP ..	36
Figura 16 - Arquitetura do SMA responsável pelas alocações e recomendações de funcionários a tarefas	37
Figura 17 - Exemplo de currículo no <i>dataset</i> JSON	39
Figura 18 - Entradas do <i>dataset</i> que relaciona países com línguas faladas.....	40
Figura 19 - Arquitetura da rede neuronal NER para a extração de informação de currículos ..	42
Figura 20 - Sumário do modelo desenvolvido	43
Figura 21 - Classificações obtidas do modelo com word2vec.....	43
Figura 22 – Resultados da rede neuronal sem <i>tuning</i> de parâmetros.....	44
Figura 23 – Confirmação das competências extraídas por técnicas de inteligência artificial....	44
Figura 24 - Tempos de Treino entre <i>Random Forests</i> e KNN.....	46
Figura 25 - Notificação de geração de projeto.....	47
Figura 26 - Página de configurações de uma empresa e do <i>AllocationsAgent</i>	48
Figura 27 - Notificação da alocação de funcionários num projeto.	49
Figura 28 - Diagrama de Sequência do processo de alocações	50
Figura 29 - Processo de integração com sistema SAP.....	51
Figura 30 – Notificação de integração com sistema SAP	52
Figura 31 - Injeção de Dependências usando Spring	52
Figura 32 - Exemplo de repositório a ser passado como argumento para um agente JADE.....	52
Figura 33 - Geração de projeto e guardar notificação na respectiva tabela dentro de um agente JADE.....	53
Figura 34 - Definição de Gene usando a biblioteca Jenetics.....	53
Figura 35 - Definição de Cromossoma, usando a biblioteca Jenetics	53
Figura 36 - Função de otimização de cromossomas	54

Figura 37– Implementação do algoritmo genético.....	54
Figura 38 - Notificação de alocação do algoritmo genético.....	55
Figura 39 – Script Flyway da criação de tabela User.....	56
Figura 40 - Detalhes de comunicação para os scripts Flyway serem inseridos na respectiva base de dados PostgreSQL	56
Figura 41 - Anotações em classe de domínio.....	57
Figura 42 - Repositórios de Empresas no módulo JAVA	57
Figura 43 - Exemplo de funcionalidade acrescentada a interface de repositório	57
Figura 44 - Detalhes de conexão ao módulo de bases de dados através do módulo JAVA.....	58
Figura 45 - Padrão <i>Factory</i> para a criação de agentes	58
Figura 46 - Página de Registro de Candidato	59
Figura 47 - Página de confirmação das competências previstas pelos modelos de inteligência artificial	60
Figura 48 - Página de criação de empresa	61
Figura 49 – Página de conexões detalhe SAP.....	61
Figura 50 - Página de notificações de projeto.....	62
Figura 51 - Integração realizada com sucesso com sistema SAP	62
Figura 52- Output de “localhost:5000”	63
Figura 53 – Resposta JSON das competências reconhecidas a um funcionário usando os modelos preditivos de extração de informação e inferência de competências.....	63
Figura 54 - Exemplo de Coleção Postman	65
Figura 55 - Teste em Postman para testar o tipo de pedido	66
Figura 56 - Resultados dos testes das coleções Postman	66

Lista de Tabelas

Tabela 1 - Fases da dissertação	4
Tabela 2 - Tabela das Questões de Investigação	7
Tabela 3 - Fontes	8
Tabela 4 - Termos de Pesquisa da Primeira Questão de Investigação	8
Tabela 5 - Termos de Pesquisa da Segunda Questão de Investigação	9
Tabela 6 – Termos de Pesquisa da Terceira Questão de Investigação	9
Tabela 7 - Termos de Pesquisa da Quarta Questão de Investigação	9
Tabela 8 - Critérios de Inclusão	10
Tabela 9 - Critérios de Exclusão	10
Tabela 10 – Melhorias dos resultados do algoritmo genético com e sem o fornecimento de populações iniciais boas.....	23
Tabela 11 – Especificações do Computador Usado Para Treinar os Modelos Preditivos	35
Tabela 12 - Resultados modelos de inferência de capacidades.....	45
Tabela 13 – Perguntas da Análise de Usabilidade	67
Tabela 14 - Respostas para perguntas da <i>System Usability Scale</i>	68
Tabela 15 - Respostas possíveis ao questionário	68
Tabela 16 – Valores obtidos do questionário	69
Tabela 17 - Notação utilizada para descrever estado das tarefas da dissertação	72
Tabela 18 - Estado da dissertação - tarefas	72

Acrónimos e Símbolos

Lista de Acrónimos

CNN	<i>Convolutional Neural Network</i>
IA	Inteligência Artificial
DL	<i>Deep Learning</i>
Bi-LSTM	<i>Bidirectional Long-Short Term Memory</i>
SVM	<i>Support Vector Machine</i>
SMA	Sistema Multiagente
RH	Recursos Humanos
NLP	<i>Natural Language Processing</i>
NER	<i>Named Entity Recognition</i>
SPA	<i>Single Page Application</i>
BD	Base de Dados
AG	Algoritmo Genético
ERP	<i>Enterprise Resource Planning</i>

1.Introdução

1.1 Enquadramento da Dissertação

A dissertação presente neste documento foi desenvolvida no âmbito das unidades curriculares de TPROJIA (Trabalho Preparatório de Project/Dissertação/Estágio), IAAPLIA (Inteligência Artificial Aplicada) e PROJIA (Projecto/Dissertação/Estágio), pertencentes ao Mestrado em Engenharia da Inteligência Artificial (MEIA) do Instituto Superior de Engenharia do Porto (ISEP).

A dissertação pretende abordar a gestão de competências e recursos humanos dentro de empresas, recorrendo, para isso, a mecanismos de inteligência artificial (IA) de forma a automatizar estes processos considerados tediosos e de difícil manutenção.

Para além disso, aliados a estes novos conceitos e tecnologias, foram utilizadas as boas práticas de engenharia de software para que, no fim, fosse obtida uma solução que almejasse robustez, manutenibilidade, testabilidade e extensibilidade.

1.2 Contextualização do Problema

O departamento de RH (recursos humanos) é um dos com mais importância e responsabilidades pois é responsável por imensas tarefas de gestão, como, por exemplo, relações entre os colaboradores, pagamentos, gestão de talentos, *onboarding* de novos recursos, entre outros [1]. Uma das tarefas que este departamento possui é a gestão de talentos e competências dentro de uma empresa. Esta envolve várias subáreas que têm como principal objetivo atrair, seleccionar, treinar e, sobretudo, reter funcionários existentes [2].

De acordo com [3], o tempo médio de permanência de um funcionário numa dada empresa é de 4.1 anos, dados do Departamento de Estatística Laboral dos Estados Unidos Da América. Isto significa que as empresas estão permanentemente a mudar os seus colaboradores, algo que é custoso e do desagrado para ambas as partes. Dados de [4] revelam que a não permanência dos funcionários, nos Estados Unidos, forçam um custo de 630 mil milhões de dólares anuais (dados de 2021). A não permanência de um único funcionário no seu posto representa um custo estimado para empresa de um terço do salário anual desse mesmo trabalhador. Deste custo, a maior parte do dinheiro corresponde à produtividade perdida do funcionário, embora uma quantia significativa seja gasta na procura, contratação e treino do substituto. Assim, podemos perceber que se uma empresa está constantemente a trocar de recursos, então as competências dos funcionários também estão sempre a ser trocadas e algumas precisam de ser readquiridas caso perdidas. Este problema merece a atenção total das empresas, assim como o seu investimento, pois traduz-se diretamente no seu desempenho.

Para além disso, a revolução tecnológica causada pelos avanços da computação requer que os funcionários estejam num constante processo de treino para aprender novas e variadas competências a um ritmo estonteante e desgastante [5]. Ainda, ameaças externas como a pandemia do COVID-19 agravaram a volatilidade sentida atualmente nos mercados internacionais [5]. De acordo com [6], um cenário recreado revela que, até 2030, cerca de 400 milhões a 800 milhões de pessoas perderão o seu emprego devido à automação. No entanto, apesar disto, o estudo também revela que 375 milhões de novos postos serão criados que vão exigir diferentes tipos de competências.

Num mercado tão volátil e feroz como o vivido atualmente, a maneira mais simples e eficaz de evitar custos desnecessários com a contratação de trabalhadores é reter os já existentes, ter em atenção as competências internas numa empresa e ter ainda perceção das competências necessárias para a execução dos serviços e produtos que uma empresa publicita (os *know-how*) [7] [8]. Esta abordagem baseada em competências é tão tentadora para as empresas que, no futuro, a base de unidade do trabalho já não será a função ou posto de trabalho, mas sim as competências necessárias para executar tarefas. Um foco baseado em competências permite às empresas serem muito mais objetivas de forma a assegurarem a execução dos seus serviços. Em contrapartida, a abordagem clássica da relação trabalhador-empregado é vista como obsoleta, pois limita o funcionário ao posto que está a executar não aproveitando a totalidade das suas competências e, ainda, vai ao encontro do instinto humano de experimentar novas tarefas que poderiam ser, potencialmente, vantajosas para uma empresa [9].

Recentemente, de modo a responder a esta nova abordagem no mercado na área dos RH, tem surgido um novo tipo de ferramenta. Este tipo de ferramenta é designado de “mercados de talento”, e são maioritariamente plataformas online onde recrutadores podem apresentar oportunidades de trabalho, trabalhos temporários, tarefas de mentoria ou qualquer outra iniciativa [9]. Estes mercados (dependendo das funcionalidades oferecidas) podem, ainda, recomendar mudanças nos grupos de trabalho internos, treinos de funcionários para assumirem funções se tiverem o perfil necessário e, ainda, aquisições de candidatos específicos, tudo de acordo com as necessidades da empresa no momento ou num futuro próximo. Podemos dizer que estes tipos de plataforma atraem ou “descobrem talento que está escondido nas organizações” [10].

Os mercados de talento estão rapidamente a evoluir e a serem adotados por várias empresas famosas. Por exemplo, a NASA [11], recentemente, desenvolveu o seu próprio mercado de talento com o objetivo de listar oportunidades internas de trabalho. Para além disso, o mercado permitia a seleção de potenciais candidatos e reorganização/transferência de elementos de grupos de trabalho de forma a estimular a aprendizagem durante o percurso profissional. Outros exemplos de empresas que seguiram esta abordagem podem ser a Google, Nestlé, American Express, entre outras [12].

Recentemente, algumas empresas, tais como a Gloat [13], TalentSoft [14] e a 365Talents [15] focam-se em fornecer serviços de mercados de talento através de plataformas online tendo, a última, já realizado soluções mais internas com bastante sucesso para outras empresas como a Allianz.

A IA (inteligência artificial), a área das ciências de computadores e da engenharia dedicada a simular inteligência humana em máquinas [16], é uma área que tem estado em crescimento nas últimas décadas [17]. É já uma área de relevância na segurança, saúde, manutenção de inventários, assistentes virtuais [18], RH [19], entre muitos outros. Esta área está também presente nos ditos mercados de talento.

Os mercados de talento, como o da 365Talent, argumentam utilizar técnicas de NLP (*Natural Language Processing*), em particular uma chamada “*Job-Oriented Assymetrical Pairing System*” que pode ser vista como uma técnica semelhante a *word2vec*, no entanto, mais orientada à área dos RH e setores de recrutamento [20]. Foi concluído que era uma abordagem mais eficaz e eficiente do que a tradicional *word2vec*.

Em suma, os departamentos de RH enfrentam, hoje em dia, um desafio em preparar as suas empresas para o futuro. A revolução tecnológica e a pandemia do COVID-19 catalisaram este processo em que os funcionários têm que aprender novas competências em ritmos estonteantes. As empresas, devido a isto, passaram a encarar o funcionário como um repositório/detentor de competências que podem ajudar a empresa a alcançar os seus objetivos em vez de vê-los como meras pessoas designadas a postos de trabalho específicos.

A revisão literária efetuada nesta dissertação tem como objetivo estudar técnicas de IA que possam e são utilizados atualmente em mercado de talentos. No fim, um sistema semelhante e inovador é proposto tendo por base a revisão literária realizada.

1.3 Contributos e Objetivos do Projeto

O objetivo do projeto realizado no contexto desta dissertação é desenvolver uma solução que recorra a mecanismos de IA que possam automatizar a gestão e manutenção dos repositórios das competências que as empresas possuem. Todos os anos, milhões de euros são perdidos neste processo suscetível ao erro humano sendo que um sistema automatizado pode prevenir estas perdas. Um dos objetivos principais deste estudo é que para além de técnicas de NER (*Named Entity Recognition*) já comuns, seja realizada uma inferência de competências através de informações fornecidas. Depois disso, os sistemas multiagentes e algoritmos genéticos de alocações alocam os funcionários com as funções que potencialmente desempenham melhor de uma forma otimizada.

O projeto desta dissertação não é suposto ser um produto finalizado, mas antes uma prova de conceito que possa, caso necessário, fazer parte do portfólio de soluções da empresa KonkConsulting. Devido a isto, a pesquisa foi feita mais aprofundadamente sobre os processos mais inovadores como, por exemplo, os mecanismos dedicados à inferência de competências.

1.4 Planeamento do Trabalho

A presente dissertação teve início em novembro de 2021 e terminou em outubro de 2022. A seguinte tabela explora a duração temporal das tarefas realizadas. A comunicação entre o aluno e a orientadora foi feita via comunicação síncrona e assíncrona tendo sido utilizado maioritariamente o Microsoft Teams e sendo, ocasionalmente, usado o meio presencial no Instituto Superior de Engenharia do Porto. A comunicação entre aluno e supervisor foi feita de forma síncrona e assíncrona usando, também, o Microsoft Teams. A duração das tarefas está presente na Tabela 1.

Tabela 1 - Fases da dissertação

Fase da dissertação	Duração
Escrita da Dissertação	Novembro de 2021 – Outubro de 2022
Definição das Metodologias de Pesquisa	Novembro de 2021 – Dezembro de 2021
Revisão Literária	Novembro de 2021 – Janeiro de 2022
Implementação da Solução	Dezembro de 2021 - Julho de 2022
Testabilidade da solução	Julho de 2022 – Agosto de 2022

1.5 Estrutura da Dissertação

Estruturalmente, a presente dissertação encontra-se dividida em 6 capítulos:

- Introdução;
- Revisão Literária
- Análise e Desenho da Solução Proposta;
- Implementação;
- Testes;
- Conclusão;

Neste primeiro capítulo, o objetivo é apresentar e contextualizar o problema que motivou a realização desta dissertação, os objetivos e contributos que a solução proposta pretende resolver e ainda apresentar o plano de trabalho da sua realização.

No segundo capítulo, encontra-se a revisão literária. Numa primeira subsecção, encontram-se as metodologias por detrás do processo da escolha de artigos que incorporam a revisão literária. Posteriormente, nas duas subsecções seguintes, os artigos escolhidos são enumerados e descritos.

No terceiro capítulo, aborda-se o planeamento e design arquitetural da solução tendo por base a revisão literária do capítulo anterior. Aqui encontram-se, de uma forma detalhada, o domínio da solução implementada, uma justificação sobre as tecnologias utilizadas, uma visão sobre a estrutura de base de dados (BD) implementada, um planeamento de arquitetura dos modelos de IA implementados, descrição e pré-processamento dos *datasets* escolhidos e ainda o plano do fluxo sequencial dos algoritmos dedicados ao emparelhamento de recursos a tarefas.

No quarto capítulo, os detalhes da implementação da solução são explorados. Nesta secção encontram-se a explicação dos ecrãs desenvolvidos, integrações entre os módulos do sistema, ou alguns exemplos práticos de como a solução segue as boas práticas da engenharia de software. Para além disso, os resultados dos modelos preditivos são apresentados assim como detalhes mais técnicos das suas implementações. A integração e implementação do sistema multiagente e algoritmo genético são também detalhados.

O quinto capítulo é dedicado aos testes realizados no sistema. Neste capítulo, encontra-se, ainda, uma análise de usabilidade realizada às interfaces gráficas do sistema.

Por último, o sexto capítulo é dedicado à conclusão. Nesta fase, são enumerados os objetivos atingidos, é descrito trabalho futuro que possa ser realizado e é dada, ainda, uma apreciação pessoal.

2. Revisão Sistemática

O sucesso recente da IA em diversos campos leva a crer no potencial desta para resolver qualquer problema. Uma revisão literária permite a um investigador extrair conhecimento ao analisar documentos, isto após uma triagem cuidada e metódica para que sejam selecionados apenas os documentos relevantes ao estudo em causa. Este conhecimento permite ao investigador identificar soluções para questões levantadas, mas permite, de uma forma igualmente importante, identificar limitações associadas às soluções encontradas [21]. A revisão literária presente neste documento foi realizada seguindo a metodologia “*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*” (PRISMA) e começa por apresentar as questões de investigação levantadas. De seguida, são dadas a conhecer as fontes, assim como os termos de pesquisa utilizados. Por último, os critérios de inclusão e exclusão relativos ao controlo de qualidade obrigatório em todos os documentos são apresentados, incluindo um diagrama PRISMA que resume todo o processo descrito.

2.1 Metodologia de Pesquisa

Nesta secção, é descrita a metodologia na qual se realizou a revisão literária. Neste ponto, são descritas as questões de investigação identificadas, os termos de pesquisa definidos, os critérios de inclusão e exclusão que os artigos devem obedecer e ainda as fontes onde a pesquisa foi realizada. No fim, um pequeno resumo do processo é descrito.

2.1.1 Questões de Investigação

A principal questão de investigação identificada durante esta revisão literária foi “Como utilizar mecanismos de IA para extrair e inferir competências de candidatos e associá-los com funções disponíveis numa empresa?”. Esta questão, devido à sua complexidade, foi dividida em 4 subquestões de investigação, cada uma abordando um subdomínio relevante no problema global. Estes domínios são a extração de informação de currículos, a inferência de competências, o emparelhamento de empregados com tarefas utilizando SMA e ainda o mesmo emparelhamento de empregados com tarefas utilizando algoritmos genéticos.

As 4 questões identificadas para cada um dos domínios estão presentes na Tabela 2.

Tabela 2 - Tabela das Questões de Investigação

ID da questão de investigação	Questão de Investigação	Domínio da questão de investigação
RQ1	“Quais são as principais técnicas de IA para a extração automatizada de informação de currículo?”	Extração de informação de currículos
RQ2	“Quais são as principais técnicas de IA para inferir competências de recursos?”	Inferência de competências

RQ3	“Existe algum SMA usado para emparelhar empregados com empregos/tarefas?”	Emparelhamento de empregados com tarefas/projetos
RQ4	“Existem algoritmos genéticos usados para emparelhar empregados com empregos/tarefas?”	Emparelhamento de empregados com tarefas/projeto

2.1.2 Fontes de pesquisa

De forma a responder às quatro perguntas de investigação foram realizadas pesquisas nas fontes presentes na Tabela 3.

Tabela 3 - Fontes

Fonte	URL da fonte
B-On	https://www.b-on.pt/
IEEE-Xplore	https://ieeexplore.ieee.org/

2.1.3 Termos de Pesquisa

Com o intuito de obter artigos relevantes na resposta às questões identificadas na Tabela 2, os seguintes termos de pesquisa foram identificados e aplicados sobre as fontes presentes na Tabela 3.

Visto que os domínios das questões de investigação identificadas são bastante diferentes entre si, os termos de pesquisa utilizados para cada questão são também eles diferentes. Nas Tabela 4, Tabela 5, Tabela 6 e Tabela 7, estão identificados os termos de pesquisa para as questões de investigação RQ1, RQ2, RQ3 e RQ4, respetivamente. Os termos de pesquisa para cada pergunta estão em inglês para a obtenção do maior número de resultados possíveis e estão separados entre si por cláusulas “AND” ou “OR” para garantir os resultados mais relevantes possíveis.

Tabela 4 - Termos de Pesquisa da Primeira Questão de Investigação

Escopo da Pergunta	Termo de Pesquisa
Tópico Principal da Pergunta de Investigação	“Resume” AND
Tarefa de Processamento Textual	(“Automated” OR “Feature Extraction” OR “Information Extraction” OR “parsing”) AND
Recrutamento	(“Recruitment” OR “Recruiter” OR “Job” OR “Candidate”) AND
Técnicas de Inteligência Artificial	(“Machine Learning” OR “Artificial Intelligence” OR “Deep Learning” OR “Natural Language Processing” OR “Neural Networks”)

Tabela 5 - Termos de Pesquisa da Segunda Questão de Investigação

Escopo da Pergunta	Termo de Pesquisa
Tópico Principal da Pergunta de Investigação	"Skill Inference" AND
Tópicos do tema	("Descriptive Text" OR "Adjacent Skills" OR "Skills") AND
Técnicas de Inteligência Artificial	("Neural Networks" OR "Natural Language Processing").

Tabela 6 – Termos de Pesquisa da Terceira Questão de Investigação

Escopo da Pergunta	Termo de Pesquisa
Tópico Principal da Pergunta de Investigação	"Job Matching" AND
Tópicos do tema	("Skills" OR "Job" OR "Recruitment" OR "Candidate") AND
Técnicas de Inteligência Artificial	("Multiagent System")

Tabela 7 - Termos de Pesquisa da Quarta Questão de Investigação

Escopo da Pergunta	Termo de Pesquisa
Tópico Principal da Pergunta de Investigação	"Job Matching" AND
Tópicos do tema	("Skills" OR "Job" OR "Recruitment" OR "Candidate") AND
Técnicas de Inteligência Artificial	("Genetic Algorithm")

Conforme previamente mencionado, as quatro perguntas de investigação combinadas pretendem responder a uma pergunta de investigação mais ampla, que é "Como utilizar mecanismos de IA para extrair e inferir competências de candidatos e associá-los com funções disponíveis numa empresa?". Deste modo, os conjuntos dos termos de pesquisa de cada questão, associados entre si através da condição lógica "OR", representam os termos de pesquisa que, na realidade, pretendem responder à questão global, identificada no início desta secção.

("Skills Inference" AND ("Descriptive Text" OR "Adjacent Skills" OR "Skills") AND ("Neural Networks" OR "Natural Language Processing"))

OR

("Resume" AND ("Automated" OR "Feature Extraction" OR "Information Extraction" OR "parsing") AND ("Recruitment" OR "Recruiter" OR "Job" OR "Candidate") AND ("Machine

Learning” OR “Artificial Intelligence” OR “Deep Learning” OR “Natural Language Processing” OR “Neural Networks”))
OR
 (“Job Matching” AND (“Skills” OR “Job” OR “Recruitment” OR “Candidate”) AND (“Multiagent System”))
OR
 (“Job Matching” AND (“Skills” OR “Job” OR “Recruitment” OR “Candidate”) AND (“Genetic Algorithm”))

2.1.4 Critérios de Inclusão e Exclusão

Para um artigo fazer parte da revisão literária, foi realizado um controlo de qualidade, representado sob a forma de critérios de inclusão e exclusão. Cada artigo tem que obedecer aos critérios de inclusão e desobedecer aos critérios de exclusão, de uma forma obrigatória. Os critérios de inclusão e exclusão estão presentes nas tabelas Tabela 8 e Tabela 9, respetivamente.

Tabela 8 - Critérios de Inclusão

ID do critério de Inclusão	Critério de Inclusão
IC1	“Artigo é revisto pelos pares”
IC2	“Artigo deve abordar pelo menos uma questão de investigação”
IC3	“Artigo deve utilizar algum tipo de técnica de inteligência artificial”
IC4	“Artigo deve apresentar algum tipo de resultado (matriz de confusão, ROC)”

Tabela 9 - Critérios de Exclusão

ID do critério de Exclusão	Critério de Exclusão
EC1	“Artigo deve ser escrito em Inglês”
EC2	“Artigo tem mais de 10 anos”
EC3	“Artigo é duplicado”

2.1.5 Extração de dados

Nesta secção, uma breve descrição da metodologia de extração de artigos é fornecida.

O processo começou pela identificação de questões de investigação pertinentes. Foram identificadas as questões já mencionadas, presentes na secção 2.1.1. Tendo em

consideração as mesmas, foram definidos, para cada uma, termos de pesquisa. Os termos de pesquisa (presentes na secção 2.3) foram utilizados em fontes (presentes na secção 2.2) de forma a extrair artigos com respostas às questões de investigação. As fontes identificadas permitem filtros avançados, em particular, introdução dos termos de pesquisa com operadores lógicos “AND” e “OR” de forma a garantir resultados que se aproximem ao investigado.

Este processo aconteceu para cada questão de investigação e encontra-se representado visualmente no seguinte diagrama PRISMA da Figura 1. De uma forma combinada, para as quatro questões, foram identificados 27361 artigos. Após a aplicação dos critérios de inclusão e exclusão, provenientes do controlo de qualidade, de leituras rápidas de títulos e/ou resumos, foram seleccionados um total de 44 artigos do conjunto inicial de 27361. A metodologia de pesquisa está documentada no diagrama PRISMA final, apresentado, na Figura 1.

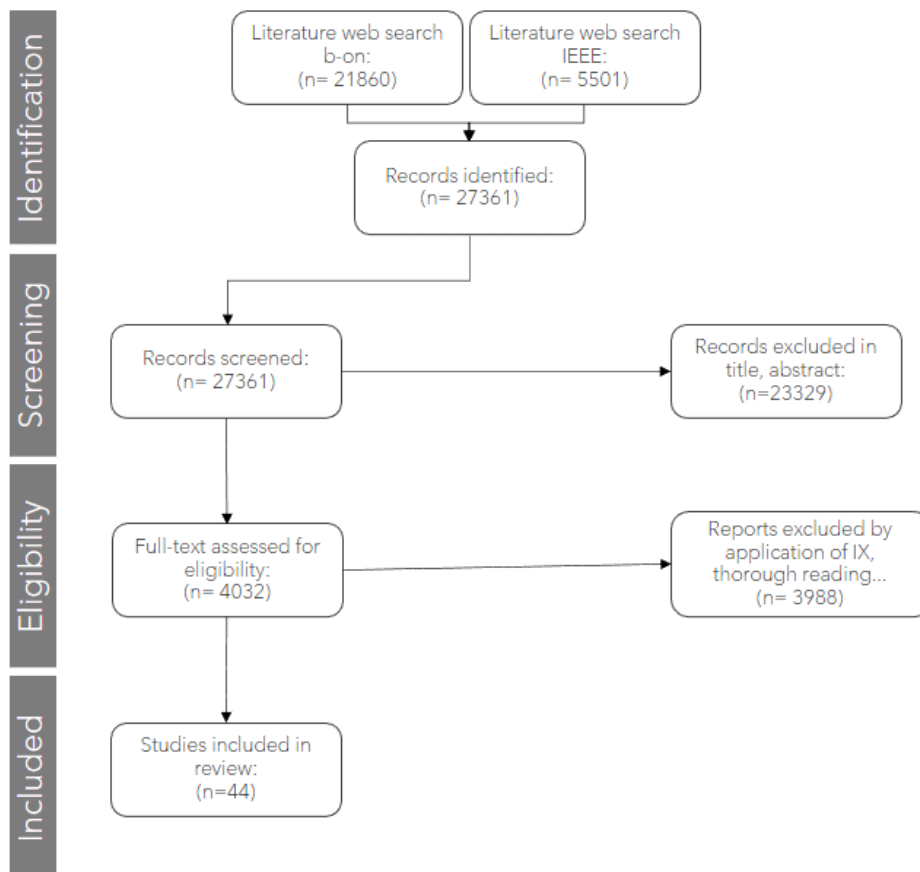


Figura 1 - Diagrama PRISMA

2.2 Resultados da Pesquisa

Nesta secção, é feita uma enumeração e breve descrição dos 44 artigos extraídos. Esta secção está organizada em quatro subsecções, cada uma dedicada a uma questão de investigação.

2.2.1 “Quais são as principais técnicas de IA para a extração automatizada de informação de currículos”

Para a primeira questão, foram encontrados cerca de 12000 artigos em que, após leitura de títulos e resumos e aplicação dos critérios de controlo de qualidade, foram reduzidos a 12 artigos escolhidos. Esta foi a pergunta de investigação que mais gerou resultados.

Vários tipos de técnicas de IA são usados para a extração de informações de currículos, o que sugere que este é já um problema amplamente explorado. A distribuição destas técnicas está presente na Figura 2.

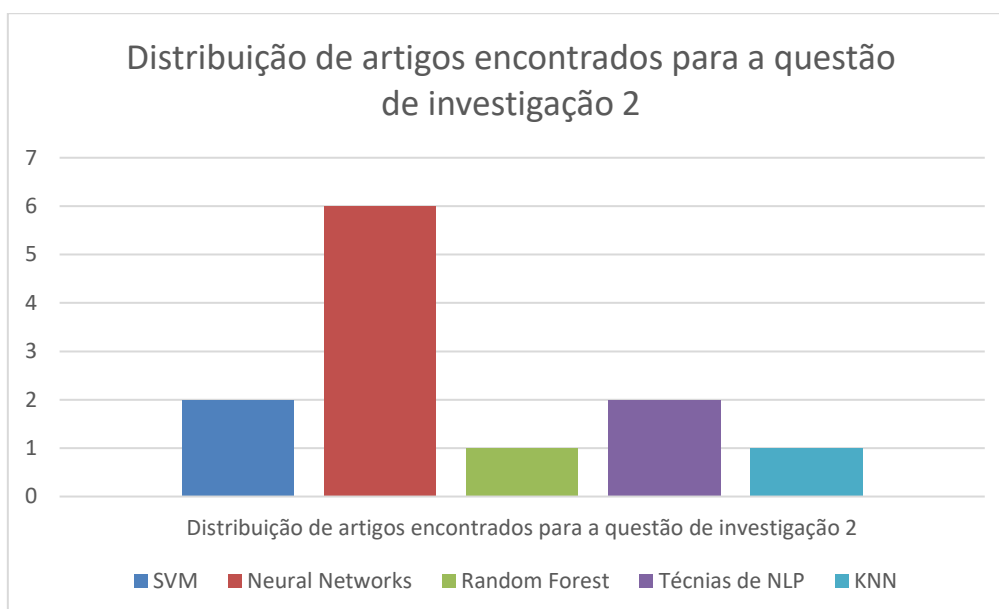


Figura 2 – Distribuição de Artigos para a Primeira Questão de Investigação

Em [22] [23] *Support Vector Machines* (SVM) são utilizados para o propósito de recomendação de currículos, atingindo resultados satisfatórios tendo em conta os baixos recursos computacionais utilizados.

A técnica de IA mais utilizada, redes neuronais (maioritariamente com arquiteturas de aprendizagem profunda), são utilizadas nos artigos [24] [25] [26] [27] [28] [29].

Random Forest ou *K-Nearest Neighbor* (KNN) são também utilizados, ainda que em contextos ligeiramente distintos. Estes são encontrados em [30] e [31].

Outros artigos, como os [32] e [33], utilizam várias técnicas exploratórias de *Natural Language Processing* (NLP) e conseguem atingir o objetivo com resultados satisfatórios.

As técnicas com melhores resultados foram, conforme expectável, as redes neuronais de aprendizagem profunda.

Todas as técnicas mencionadas acima enquadram-se em técnicas do tipo *Named Entity Recognition* (NER).

2.2.2 “Quais são as principais técnicas de IA utilizadas para inferir competências de recursos?”

Em primeiro lugar, poucos artigos que descrevem técnicas de inferências de competências foram encontrados, pelo que foi alargado o escopo para que mais artigos fossem extraídos, ainda que menos relevantes e ligeiramente fora do contexto da inferência de competências. Estas dificuldades são a prova que esta é uma questão pouco explorada e é por aqui, se possível, que podemos inovar, ainda que a falta de conhecimento existente seja uma dificuldade para planear uma solução.

Após a revisão literária, um total de 15 artigos foram seleccionados. Destes trabalhos, alguns eram focados na inferência de competências, enquanto os outros tinham em conta as competências que o recurso poderia potencialmente vir a aprender.

Várias técnicas de IA foram utilizadas nestes contextos, sendo que alguns dos artigos abordavam mais do que uma técnica. A distribuição encontra-se na Figura 3.

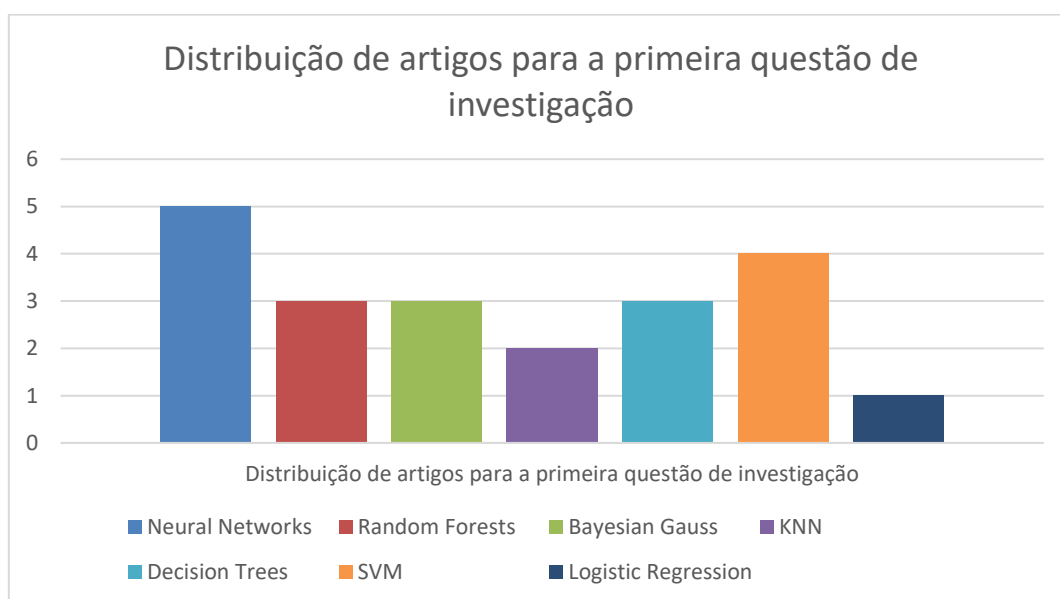


Figura 3 – Distribuição de Artigos para a Segunda Questão de Investigação

Os artigos [34] [35] e [36] apresentam *Random Forests* como sendo uma técnica viável para o efeito.

A técnica mais utilizada, as redes neuronais, estão presentes nos artigos [37] [38] [39] e [40].

KNN e *Logistic Regression* estão presentes nos artigos [41] e [42], respetivamente.

Os *Support Vector Machine* (SVM) estão presentes em vários artigos, como [43] [44] [45] e [46]. Particular destaque para o artigo [45] que utiliza e explora este tipo de técnicas para a gestão de competências dentro dos recursos das forças aéreas americanas.

Também para esta questão de investigação, as redes neuronais foram a técnica que apresentou melhores resultados nos problemas que se propunha resolver, embora a utilização de um modelo variasse consoante a natureza do problema que pretendia responder.

2.2.3 “Existe algum SMA usado para emparelhar empregados com empregos/tarefas?”

Sim, a revisão literária foi capaz de encontrar sistemas multiagentes (SMA) que emparelham recursos a tarefas, sendo que cada artigo faz este emparelhamento, tendo em conta diferentes critérios. No total, 11 artigos foram encontrados para responder à questão de investigação.

Em [47], é proposto um SMA num contexto de e-recrutamento. Este SMA recomenda aos candidatos trabalhos existentes numa dada empresa, enquanto, para os recrutadores, recomenda candidatos que poderiam realizar a função de acordo com a informação de competências fornecida.

Em [48], o autor do artigo acredita que o coletivo de inteligência de um grupo é sempre superior ao do individual, mas acredita que a performance do coletivo numa determinada tarefa é dependente da soma individual da performance que cada individuo num grupo tem na sua tarefa designada. Como tal, o autor implementou um SMA para testar a afirmação. O SMA tinha em conta a associação de recursos com tarefas de uma forma otimizada.

Em [49], é proposto um SMA para a seleção e recrutamento de arquitetos de software.

Em [50], é apresentado um SMA para o recrutamento de trabalhadores baseado em competências, este implementando um sistema bidirecional de feedback com o intuito de auto melhorias e de ajustamento de pesos para resultados futuros mais favoráveis.

Em [51] [52] [53] [54] são propostas várias arquiteturas de SMA, com o intuito de escolher o candidato mais apto para a realização de um dado trabalho ou de uma dada tarefa, tendo em atenção as competências técnicas e psicológicas que este possui.

Em [55], é proposto um SMA baseado na negociação contratual entre um candidato e uma empresa. Este artigo foi escolhido, pois utiliza as competências do candidato e outras informações, como a experiência de mercado de trabalho, para otimizar a elaboração e negociação de contratos.

Após uma apreciação da primeira fase do trabalho realizado com a orientadora, ficou acordado que SMA em contexto de desporto robótico, embora não diretamente relacionados com a questão de investigação, são muito aproximados à situação em estudo de alocação de recursos a tarefas/funções. Na robótica, robots com diferentes

características desempenham funções diferentes, tal e qual como os candidatos que apenas são escolhidos se possuírem as competências para desempenhar a função que lhes é designada. Neste contexto, foram adicionados os seguintes 2 artigos [56] [57].

A distribuição de artigos encontra-se na Figura 4.

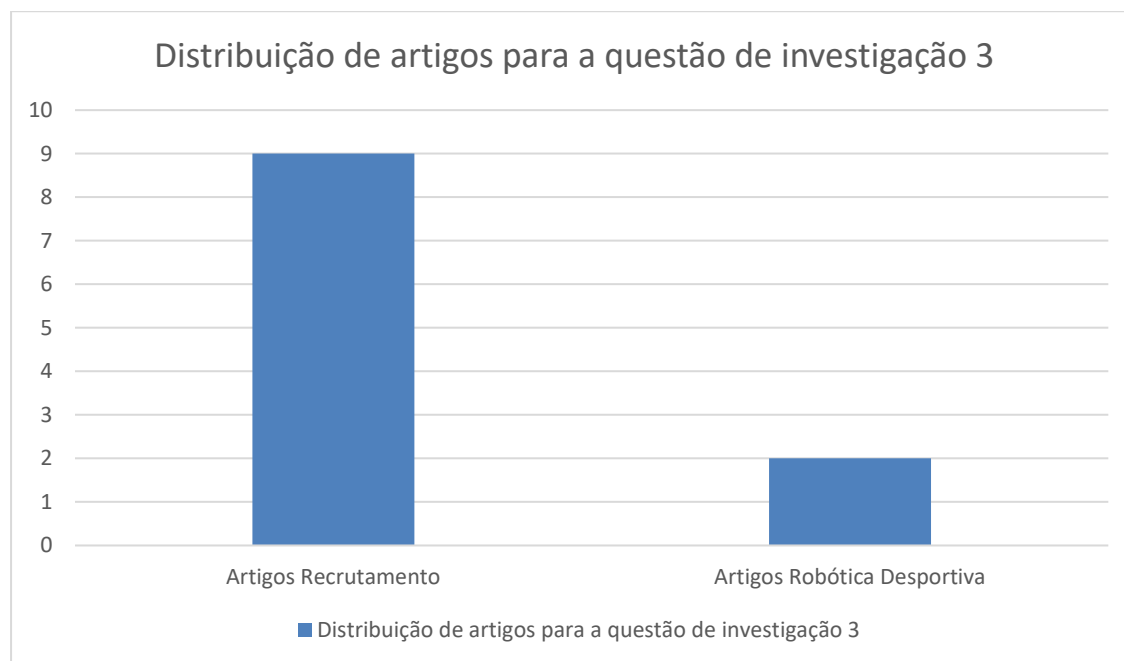


Figura 4 - Distribuição de artigos da Terceira Questão de Investigação

2.2.4 “Existem algoritmos genéticos utilizados para emparelhar empregados com empregos/tarefas?”

A revisão literária provou, com sucesso, que existem algoritmos genéticos (AG) utilizados com o intuito de emparelhamento de recursos com tarefas de forma otimizada. Ao todo, foram escolhidos 6 artigos para responder à questão de investigação elaborada.

No artigo [58], é apresentado um AG, introduzido com sucesso num contexto de otimização de recursos humanos nas emergências de um hospital, de acordo com as especialidades que cada recurso sabia realizar.

Em [59], é usado um AG que tem em conta o número de funcionários, atividades e preços para a otimização de recursos humanos num armazém de modo a responder às necessidades avultadas de uma era digital na realização de encomendas.

De seguida, em [60], foi proposto um AG que tem como principal objetivo diminuir os custos operacionais e, conseqüentemente, aumentar os lucros através de uma alocação eficaz e eficiente de recursos humanos.

No artigo em [61], é apresentado um AG para a criação de grupos de estudantes, sendo que estes necessitam de competências específicas para a resolução de uma tarefa. O algoritmo

tem como objetivo criar os grupos mais homogêneos possíveis, de forma a existir equilíbrio entre estes.

Em [62], foi implementado um AG num contexto de funcionários que trabalham em horário parcial. Esta implementação tem em conta os horários reduzidos e flexíveis dos funcionários para além das competências de cada um.

Por último, em [63], é sugerido um AG para o contexto da alocação eficiente de recursos, considerando as restrições ao nível das competências dos funcionários. Esta implementação tem ainda em atenção o erro humano na execução das competências dos funcionários.

A distribuição dos artigos encontra-se na Figura 5.

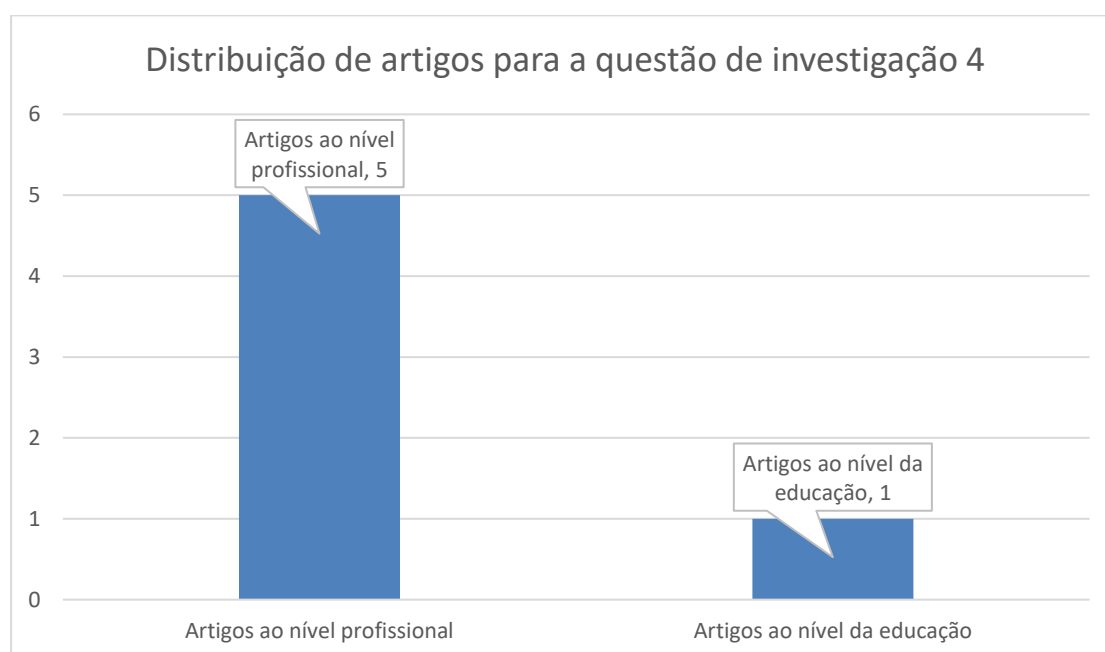


Figura 5 - Distribuição dos artigos da Quarta Questão de Investigação

2.3 Discussão dos Resultados

Nesta secção, é realizada uma análise mais profunda de cada artigo encontrado e enumerado na secção anterior. Esta secção é também composta por 4 subsecções, uma para cada questão de investigação.

2.3.1 “Quais são as principais técnicas de IA utilizadas para a extração automatizada de informação de currículos?”

A primeira questão de investigação, como previamente mencionado, é orientada em entender quais são as técnicas de IA mais utilizadas na extração de informação de currículos de candidatos. Este é um problema clássico de NER e, por causa disso, foi encontrada uma

grande quantidade artigos com vários modelos distintos. Apesar da larga quantidade de artigos, foram selecionados apenas os modelos com melhores métricas de performance, ou modelos que atinjam resultados satisfatórios consumindo baixos recursos computacionais.

Em [22], é explorada a extração de *soft* e *hard skills* de currículos. O autor do sistema utiliza, em primeiro lugar, técnicas de NLP como *N-Grams* para capturar o contexto das competências nas descrições dos trabalhos. Os autores utilizaram dois modelos distintos. O sistema de classificação com melhor performance foi, primeiramente, incorporações BERT com *tags* POS e DEP dados a um classificador de regressão logística. Um outro sistema utiliza as mesmas incorporações BERT dando-as a um classificador SVM. O primeiro modelo atingiu um F1-Score de 0.83 e o segundo alcançou um F1-Score de 0.91.

No artigo em [23], foi desenvolvido um protótipo para inferir os benefícios da automação da análise de currículos utilizando técnicas de IA, em particular, de aprendizagem automática. Foram testados vários algoritmos e modelos, no entanto, os autores concluíram que os algoritmos com melhor performance foram os SVM com uma acurácia de 0.79. Antes de chegar aos classificadores, os currículos passaram por passos de pré-processamento. Alguns passos incluíram a remoção de *tokens* especiais, a remoção palavras com apenas uma letra, técnicas de *stemming* e *lemmatization* e TF-IDF.

Em [24], é apresentada a primeira solução que utiliza DL. A ideia do projeto é recomendar currículos não tendo em conta a especificação do posto que o candidato vai ocupar, mas a preferência de pesquisa do recrutador. A solução utiliza redes neuronais do tipo CNN. Apesar de não atingir métricas de classificação interessantes, a solução proposta é interessante devido à sua abordagem distinta.

Em [25], é apresentada uma segunda solução utilizando aprendizagem profunda cujos resultados indicam que esta é uma opção viável. Neste artigo, enquadram-se as competências extraídas de um currículo com o posto de trabalho que os candidatos se estão a candidatar.

Em [26], é proposto um sistema de IA para a extração de informação de currículos onde o contexto das palavras é tido em conta através de CRF. O modelo utilizado para a previsão do tipo da palavra foram redes neuronais do tipo CNN e os resultados demonstraram que a utilização de CRF resulta numa melhoria ao nível das métricas de f1-score quando comparados com a utilização do mesmo modelo sem ter em conta o contexto.

Num outro artigo, [31], é apresentada uma outra abordagem para a questão de investigação. Esta abordagem baseia-se na leitura de informação de um currículo, sendo que o sistema, automaticamente, sugere currículos semelhantes ao que o recrutador está a ler no momento. Para tal, compara o currículo que o recrutador está a ler e vê a semelhança de todos os outros currículos existentes usando para isso um algoritmo de clusters, o KNN sugerindo os currículos mais próximos através da distância euclidiana. O autor acredita que se um recrutador de RH escolheu especificamente um currículo é porque algo lhe chamou a atenção de um modo geral e que o candidato ideal é, se não o que escolhido, parecido.

Em [27], é proposta uma abordagem com DL usando redes neurais do tipo BiLSTM-CNN. A diferença nesta proposta é que após a leitura dos currículos e classificações, um algoritmo adicional de distância euclidianas corre em que compara as competências de um candidato com as competências necessárias da função a que o candidato está a concorrer. O modelo para a extração de informação de currículos atingiu métricas de performance de 0.89 para *f1-score*, 0.84 para o *recall*, 0.94 para a precisão e 0.88 para a acurácia.

Em [28], é apresentado um outro artigo que aborda o problema desta questão de investigação e concluí que as redes neurais BiLSTM-CNN são as que atingem melhores métricas de F1-Score.

Em [32] e [33] é alcançada a mesma conclusão embora para experimentações apenas ao nível dos ramos da engenharia.

Por último, em [29], é sugerida uma abordagem de aprendizagem semi-supervisionada para a organização de currículos em secções distintas.

Tendo em conta a revisão literária realizada para esta questão de investigação, concluiu-se que as redes neurais de arquiteturas de aprendizagem profunda, nomeadamente com uma arquitetura BiLSTM, atingem os melhores resultados e, por conseguinte, são o tipo de modelo escolhido para a implementação da prova de conceito.

2.3.2 “Quais são as principais técnicas de IA para inferir competências de recursos?”

Esta questão de investigação tem como objetivo prever competências de funcionários que não estão explicitamente nos currículos. Um bom exemplo do objetivo desta funcionalidade é a inferência das linguagens que um candidato saberá falar por base nas localizações que já viveu ou trabalhou. Foram encontrados 15 artigos para responder a esta questão de investigação embora, conforme já mencionado anteriormente, tenha sido alargado o escopo da pesquisa devido à falta de resultados. A falta de resultados prova que esta é uma área ainda pouco explorada o que dificultou a extração de conhecimento.

A maior parte das vezes, as qualidades mais difíceis de entender via currículos/textos estruturados são as psicológicas que cada vez mais são importantes em empresas. Em [37], é sugerido um modelo de aprendizagem profunda, baseado em CNN-LSTM, para revolucionar a predição de competências psicológicas. Este é baseado em textos de redes sociais como o Twitter ou LinkedIn. O modelo proposto alcançou um F1-Score de 80 por cento.

No artigo em [43], é descrito um sistema para extrair características de candidatos usando as APIs do LinkedIn. O sistema é dedicado a prever competências psicológicas usando, para isso, SVM e árvores de regressão.

Em [44], é proposto um sistema que não é diretamente ligado com o contexto de aquisição de talentos, mas antes ligado à criação de um *roadmap* de treinos de um recurso já inserido dentro de uma empresa para que esta seja capaz de responder a atividades futuras. O sistema utiliza similaridades de cosseno entre as competências que o recurso possui e aquelas previstas para o desempenho das atividades futuras.

Em [45], é descrito um artigo sobre possíveis melhorias na força aérea americana. Foram utilizadas, técnicas de NLP, em particular, similaridades de cosseno entre os currículos de recursos humanos e as competências necessária para a boa execução das tarefas da força aérea americana. Importa, ainda, salientar que este tipo de técnicas e existência de sistemas de gestão de competências acontece, também, em outras empresas de renome como a NASA ou a Nestlé conforme mencionado na contextualização do problema na secção 1 deste documento.

Em [39], é proposto uma rede neuronal não só no contexto de recrutamento de staff mas também na sua manutenção e desenvolvimento. A rede neuronal, do tipo CNN-LSTM, atingiu uma acurácia de 95.313 provando, deste modo, a exequibilidade destes modelos para o contexto.

O artigo em [41], demonstra um sistema interno desenvolvido para a empresa IBM. A aplicação utiliza técnicas de IA para analisar *posts* em blogues online, redes sociais e plataformas internas da empresa de forma a subentender competências ganhas durante o percurso profissional dos funcionários na empresa. O modelo aqui utilizado foi a regressão logística e se este realizasse uma previsão, o modelo atingia uma acurácia de 80% enquanto se permitissem três tentativas para o modelo prever, uma acurácia de 96% era atingida.

Por último, em [40], é apresentado um sistema que utiliza uma rede neuronal do tipo CNN para prever as competências ganhas por um candidato no percurso profissional numa empresa de modo a automatizar a gestão interna de talentos. Neste sistema, competências de alto-nível (competências genéricas como “programação”) foram decompostas em competências de baixo nível (como “JAVA”, por exemplo”). O modelo atingiu uma acurácia de 97%.

Para a segunda questão de investigação, concluiu-se que o melhor modelo dependia do tipo de informação a ser prevista. A inferência de informações, de acordo com os resultados da revisão literária, é algo ainda pouco explorado na comunidade científica pelo que não existe uma abordagem genérica e as que existem são focadas ao domínio onde se inserem. Devido a isto, decidiu-se testar vários modelos durante a implementação da prova de conceito.

2.3.3 “Existe algum MAS usado para emparelhar candidatos com empregos/tarefas?”

A terceira questão de investigação, como previamente mencionado, é focada em encontrar exemplos de SMA utilizados para emparelhar candidatos com oportunidades de trabalho. A ideia é encontrar candidatos apropriados para trabalhos de acordo com as suas competências, educação, experiência, entre outros...

Em [47], foi desenhado um SMA com o objetivo de melhorar os processos de e-recrutamento. Este sistema contempla cinco tipos diferentes de agentes que são os *InterfaceAgents*, responsáveis pela comunicação entre funções/postos distintos, *SearcherAgents* e *RecoverAgents* que são responsáveis pelos pedidos à base de dados para trazer informações sobre os candidatos e funções. Os *OntologyAgent* que são responsáveis

por mapear pedidos em ontologias conhecidas e os *RecommenderAgents* que são o coro do sistema. Estes 2 últimos, para a recomendação de candidatos, utilizam regras através de uma interface SPARQL que tem em consideração vários fatores, sendo estes se o candidato tem qualificações necessárias para desempenhar uma função/posto, se espera o salário médio para a função designada, entre outros. Se o candidato corresponder a todos os indicadores esperados, ele/ela será o candidato ideal para desempenhar a função e será recomendado.

Em [48], foi desenvolvido um SMA, embora num contexto ligeiramente diferente do apresentado. Um SMA cooperativo foi implementado com a tarefa simples de demonstrar que a performance de um grupo é igual ao desempenho de cada elemento do grupo nas suas designadas, pelo que, se os elementos tiverem designados às tarefas mais capazes de resolverem melhor será o desempenho do grupo coletivamente.

Em [50], é proposto um SMA para recomendação de trabalhos e candidatos que foi implementado com uma ferramenta de feedback mútuo tanto para recrutadores como candidatos. O sistema utiliza *UserAgents* que representam diretamente os recrutadores e os candidatos. Cada *UserAgent* recebe mensagens de feedback de outros *UserAgents*. De acordo com as mensagens de feedback, cada *UserAgent* decide a popularidade do outro agente que está a fazer a comunicação com eles e o melhor emparelhamento é entre os agentes com melhor reciprocidade de popularidade. Os agentes com melhor popularidade entre si, representam o recrutador e candidato que melhor comunicaram e que têm os valores mais alinhados o que representa um bom candidato para o desempenhar da função proposta.

Em [51], está presente uma revisão literária em SMA para a seleção de candidatos para a realização de trabalhos. Foram encontrados, nesta revisão, 88 artigos todos eles apresentando soluções para o efeito. Num dos estudos, foi desenvolvido um sistema multiagente com dois tipos de agente que representam os candidatos e os recrutadores. O agente representador dos candidatos fornece os seus atributos técnicos e psicológicos ao agente representador do recruta que escolhe o candidato que mais se adapta às suas necessidades.

No artigo em [52], é apresentado um SMA que foi desenvolvido no contexto de formação de equipas equilibradas.

Em [53], é realizada uma outra revisão literária. Foram encontradas várias implementações provando que as implementações destes sistemas podem ser amplamente diferentes entre si pois têm em conta requisitos também diferentes.

Em [55], é apresentado um SMA utilizado nas negociações contratuais entre candidatos e recrutadores. Estes tipos de SMA podem ser complementados com SMA previamente apresentados de forma a automatizar a oferta de postos de trabalhos a candidatos oferecendo diferentes benefícios contratuais para candidatos diferentes, ou seja, candidatos com mais anos de experiência e mais competências poderiam ter melhores contratos pois desempenhariam o posto de uma forma melhor, porém, candidatos com menos experiência teriam oportunidades ainda que isso fosse refletido nas cláusulas contratuais.

Como previamente mencionado, após uma revisão do trabalho efetuado ficou acordado que o contexto da robótica desportiva era um contexto que não diferia muito do pretendido pelo que dois artigos foram adicionados numa fase posterior.

Em [56], o autor do artigo é um participante de um torneio de futebol 2D online que incentiva a aprendizagem de agentes por reforço. O participante utiliza um MAS onde triangulações são calculadas e os agentes emparelhados tomam as posições em campo de acordo com a sua distância a uma posição e as dos parceiros de equipa. Desta forma, um emparelhamento de posições é alcançado semelhante ao do contexto dos trabalhos e dos candidatos a trabalhos.

Em [57], é apresentado um artigo em tudo semelhante ao anterior, no entanto, este considerando um ambiente tridimensional. Uma vez mais as posições dos agentes são calculadas de acordo com a sua posição em campo e dos parceiros de equipa.

2.3.4 “Existem algoritmos genéticos utilizados para emparelhar empregados com empregos/tarefas?”

A quarta e última questão de investigação é direcionada a encontrar algoritmos genéticos utilizado no contexto de otimização de recursos humanos a realizar tarefas. Conforme mencionado na secção anterior, foram encontrados e escolhidos 6 artigos para responder à questão de investigação.

Em [58], é proposto um algoritmo genético que é utilizado para otimizar os recursos humanos existentes num contexto de emergências hospitalares. A performance, eficácia e eficiência dos recursos neste contexto assumem uma importância de destaque pois o sucesso ou insucesso da execução destas tarefas traduzem-se, frequentemente, na salvação ou perda de vidas humanas. Deste modo, de forma a otimizar as métricas de performance, foi implementado e aplicado um algoritmo genético no hospital de Tous, no Teerão (Irão). O algoritmo implementado tem em conta várias restrições como as especialidades dos recursos existentes, as máquinas disponíveis, o tempo médio que cada paciente gasta por ala nas urgências médicas e ainda a duração dos turnos dos recursos. Conforme visualizável na Figura 6, concluiu-se que após 500 iterações o algoritmo atingiu uma solução ótima e que prometia melhorar em 36% quando comparado com os turnos normais antes da aplicação do algoritmo. Isto traduz-se numa maior eficiência monetária, mas também, numa maior eficácia na missão de salvar vidas humanas. Entre algumas das críticas realizadas pelos autores dos estudos destaca-se a inacessibilidade aos dados dos pacientes devido à lei da proteção de dados. Acrescentar estas informações podia, potencialmente, melhorar ainda mais os resultados obtidos.

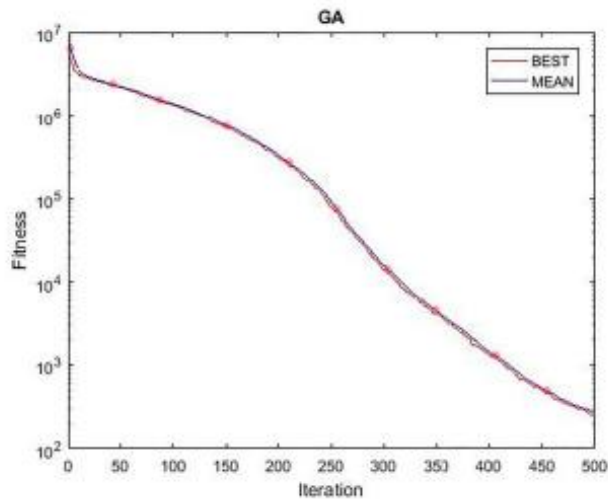


Figura 6 - Função de *fit* do algoritmo genético implementado no contexto de emergências hospitalares

Em [59], está presente um algoritmo genético que foi implementado de modo a melhorar a eficiência e produtividade de recursos humanos de modo que estejam mais preparados para as necessidades e desafios que a era digital e do marketing impõe, em particular em contextos de e-commerce em economias de larga escala como a da China. O grande objetivo passa por reduzir os custos e aumentar a margem de custos nos processos industriais de uma empresa. Para isso a correta alocação dos funcionários, as máquinas existentes num serviço e ainda os materiais necessários para a elaboração do produto são algumas das restrições que o algoritmo genético tem que ter em conta para uma correta resolução de um problema clássico que se chama de RIP (*Resource Investment Problem*). Concluiu-se que o algoritmo genético implementado se apresenta como sendo uma abordagem válida apresentando bons resultados.

Em [60], é proposto um algoritmo genético para a otimização da execução de tarefas dos trabalhadores. O objetivo é diminuir o custo da execução de tarefas de forma a aumentar o lucro proveniente da execução das mesmas. Nesta situação, o algoritmo foi aplicado num contexto do ramo da engenharia e a otimização ao nível dos trabalhadores em áreas de trabalho físicas onde as principais restrições foram as distâncias entre as áreas de trabalho (de modo a cobrir especializações dos engenheiros interdependentes), a quantidade de trabalho a ser executado, o tamanho das equipas, o balanceamento de trabalho entre as equipas formadas e ainda o número de trabalhos a realizar. A principal conclusão a retirar deste artigo e o motivo pela qual se destaca tem a ver com a importância do fornecimento de boas populações de forma a melhorar a eficiência e eficácia do algoritmo genético. Neste estudo, geraram-se aquilo que seriam considerados boas populações iniciais através de uma rede neuronal de aprendizagem profunda que foram de seguida fornecidas ao algoritmo genético de forma a gerar sugestões de organização de áreas de trabalho. A conclusão que se retirou é que o fornecimento de populações iniciais através de uma rede neuronal de aprendizagem profunda conseguiu melhorar os resultados do algoritmo genético quando comparado ao mesmo sem as boas populações iniciais geradas. Isto é visível na Tabela 10.

Embora a diferença não seja muito grande, a verdade é que as pequenas otimizações resultaram num feedback positivo por parte dos trabalhadores.

Tabela 10 – Melhorias dos resultados do algoritmo genético com e sem o fornecimento de populações iniciais boas.

TYPE	ML Improvement	Previous Run Rating
Overall FITNESS	13 %	14 %
COVERAGE	0 %	0 %
TRAVEL	1 %	0 %
UTILIZATION	0 %	0 %
BALANCE	1 %	0 %
TEAM BALANCE	38 %	38 %

Em [61], foi proposto um algoritmo genético de forma a otimizar os grupos de trabalho de estudantes de ciências dos computadores. O problema aqui em causa é em tudo semelhante ao do ambiente laboral inclusive as restrições de competências que aqui os estudantes dentro de um grupo têm que saber de modo a executar as tarefas do projeto académico. O principal destaque neste algoritmo tem a ver com um modelo de penalizações de modo a ter em conta não só grupos homogêneos de estudantes, mas também a preferência destes sobre o que trabalhar.

Em [62], é introduzido um algoritmo genético que pretende realizar otimização de tarefas dentro de um ambiente laboral sendo que o maior destaque neste artigo tem a ver com o facto que os empregados são do tipo *part-time*. A otimização é feita ao nível dos turnos de *part-time* dos empregados o que, desde logo, apresenta um conjunto de restrições entre as quais se destacam o conjunto de competências de cada empregado que tem que estar sempre assegurado de modo a garantir a execução das tarefas e, por conseguinte, dos serviços, e ainda, os turnos flexíveis de cada empregado. Esta implementação obteve resultados significativos e satisfatórios sendo que os testes foram feitos em várias lojas num contexto real. Neste artigo, é realizado ainda, um estudo que compara a abordagem do algoritmo genético com a da programação linear sendo que os resultados se encontram na Figura 7, onde a linha verde representa a programação linear (LP), a linha azul o algoritmo genético (AG) e a linha vermelha é o *Lower-Bound*, uma solução que apresenta, apesar de tudo, problemas de memória volátil.

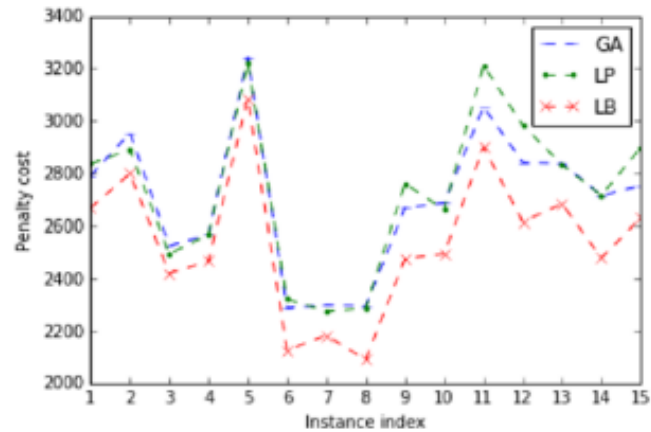


Figura 7 – Comparação entre performance do algoritmo genético e soluções usando programação linear

Deste modo, para além dos resultados mais satisfatórios concluiu-se que o algoritmo genético representa também uma melhoria ao nível da performance.

No último artigo, em [63], é apresentado um algoritmo genético com o intuito de otimizar os recursos humanos na execução de tarefas. A grande diferença desta abordagem quando comparada com as demais encontradas prende-se no facto de o algoritmo de minimização tem em conta as incertezas de horários de acordo com acontecimentos históricos (sair mais cedo do posto, entrar mais tarde, acidentes de trabalho). Esta abordagem demonstrou melhorar a confiabilidade nos resultados do algoritmo.

Concluiu-se, deste modo, que existem, de facto, algoritmos genéticos utilizados para o contexto da otimização de recursos humanos na execução de tarefas num ambiente de trabalho. Como nas secções anteriores, existem diferentes implementações de algoritmos genéticos pois os requisitos onde estes são implementados são também diferentes. De qualquer das maneiras, a revisão literária demonstra que esta é uma abordagem válida para a resolução do problema.

3. Análise e Desenho da Solução Proposta

Nesta secção, é apresentado o desenho e planeamento da solução desenvolvida. Estas decisões tiveram em conta o conhecimento extraído da revisão literária presente na secção anterior.

3.1 Tecnologias Utilizadas

A maior parte das tecnologias utilizadas no sistema foram escolhidas considerando a documentação existente, o nível de conforto do autor na utilização das mesmas e restrições necessárias para a integração com sistemas de gestão de empresas, como o sistema de gestão empresarial (ERP) da SAP.

O sistema é composto por 4 módulos distintos.

O primeiro módulo é referente ao *frontend* do sistema, ou seja, a sua interface gráfica. Este módulo foi desenvolvido utilizando Angular. Utilizou-se ainda uma *framework web* relativa ao desenvolvimento de interfaces, NGX Fundamentals. A escolha da primeira deve-se ao nível de conforto na utilização da ferramenta. Relativamente ao NGX Fundamentals, esta *framework* respeita as diretrizes obrigatórias das interfaces Fiori estabelecidas pela SAP para as suas extensões. Sobre este módulo foi realizada uma análise de usabilidade.

O segundo módulo, onde está alocada a maioria da lógica de negócio do sistema, foi desenvolvido em JAVA, sendo utilizada a *framework* Spring. Uma vez mais, as tecnologias foram escolhidas de acordo com o nível de conforto do desenvolvedor, sendo que ainda influenciou esta decisão a existência de bibliotecas como JADE e Jenetics, que foram importados como dependências de forma a implementar os SMA e AG. A nível de testes foi utilizado o Mockito e o Junit. A escolha da utilização do JADE e Jenetics está explicada mais à frente nas respetivas subsecções.

A bases de dados (BD) foi implementada utilizando o sistema de gestão PostgreSQL, devido à sua forte documentação. A injeção de dados no arranque do sistema é feita utilizando a ferramenta de migrações Flyway. Este é o terceiro módulo.

Por último, o quarto módulo é dedicado ao desenvolvimento dos modelos extratores de informações de currículos e inferência de competências. Este módulo foi desenvolvido utilizando a linguagem Python e diversas bibliotecas. A escolha de um módulo separado para o desenvolvimento, em vez de ser embutido no segundo módulo, está relacionado com a forte documentação existente na linguagem Python, mais popularmente utilizada para o desenvolvimento de sistemas de IA.

Foi utilizado, ainda, Docker de modo a realizar a abstração do sistema operativo e virtualização.

3.2 Arquitetura do Sistema

Conforme mencionado na secção anterior, a solução proposta nesta fase é composta por 4 módulos distintos. Um módulo é dedicado às interfaces gráficas do sistema, outro ao domínio e lógica de negócio do sistema, outro armazena os dados do sistema e o último é dedicado aos modelos de IA.

A arquitetura de alto nível do sistema encontra-se representado visualmente na Figura 8.

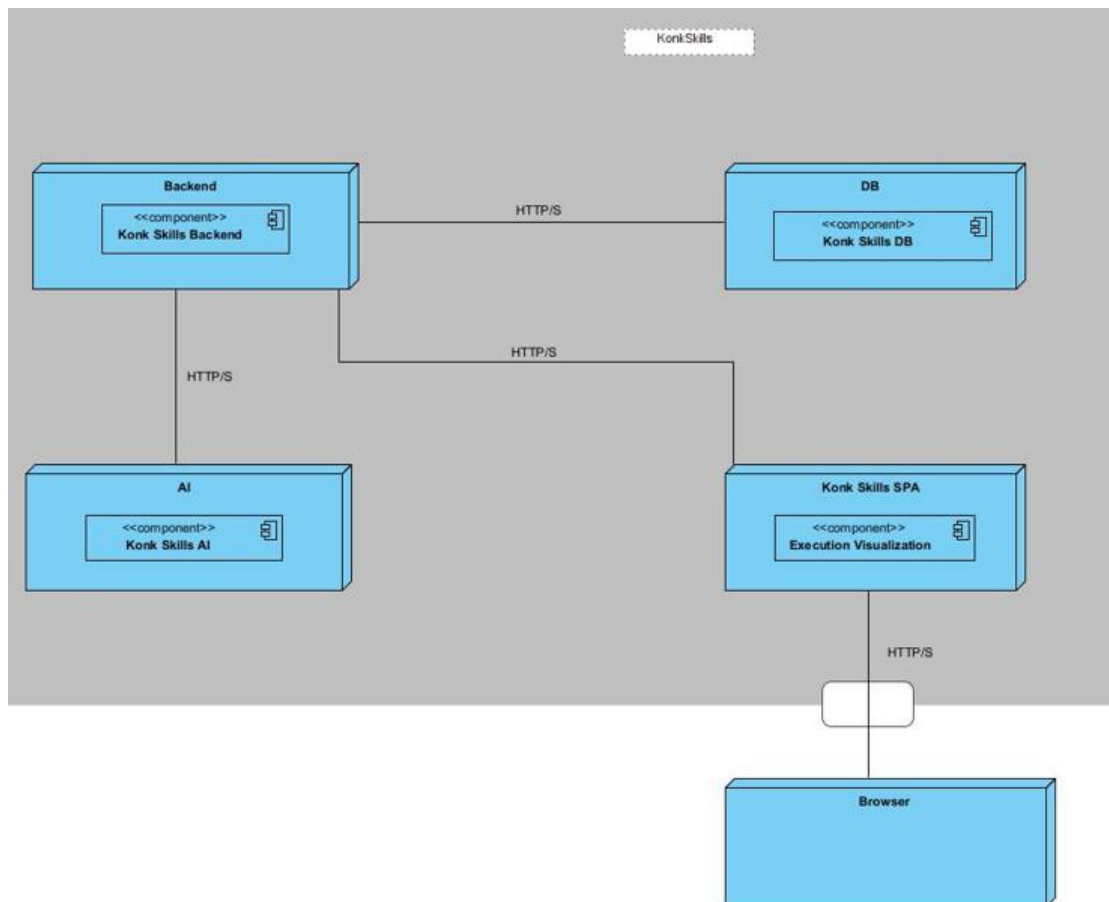


Figura 8 - Vista da arquitetura do sistema

3.3 Domínio Proposto

O domínio da solução implementada prevê, em primeiro lugar, utilizadores de três tipos distintos, sendo estes o Candidato, o Recrutador e o Empregado.

O utilizador Recrutador tem a possibilidade de criar *Companies* e cada *Company* pode ter um ou vários *Projects* que correspondem aos projetos dentro de cada empresa. O *Project* tem uma lista com 0 ou vários elementos do tipo *Skill*. Este elemento do tipo *Skill* corresponde a uma competência pelo que, a lista destes objetos num *Project*, corresponde às competências necessárias que o empregado tem de ser capaz de executar (pelo menos tem que saber executar parte delas) para poder ter rendimento e ser útil nesse mesmo projeto. O utilizador Recrutador pode ser visto como um administrador de empresas.

Por outro lado, o utilizador do tipo *Candidate* também contem uma lista do tipo *Skill*, porém, nesta situação, a lista refere-se às competências que o utilizador Candidato sabe realizar. Assim, deste modo, temos a possibilidade de comparar as competências que um candidato possui com as competências que um projeto necessita e recomendar boas associações. As *Skills* dos candidatos são retiradas e inferidas pelos modelos de inteligência artificial através dos currículos submetidos no sistema.

O utilizador do tipo *Employee* é um utilizador do tipo candidato, no entanto, já associado e contratado por uma empresa. Este tipo de utilizador existe para que os sistemas multiagente consigam realizar emparelhamentos apenas com os recursos que uma empresa realmente possui.

Os sistemas multiagente ao realizar o processo de alocações, recomendações e integrações geram objetos do tipo *Notification*. Estes objetos contêm um único atributo textual, descritivo da operação que realizou.

Saliente-se, ainda, que os objetos *Company* têm uma relação de um para um com o objeto do tipo *CompanyDetails*. Este objeto contém informações adicionais sobre a conexão com o sistema SAP da empresa. Estas informações são utilizadas, posteriormente, pelo SMA responsável para efeitos de integração.

Para além disso, o objeto *CompanyDetails* possui atributos do tipo *boolean* sobre variáveis configuráveis que os algoritmos de emparelhamento devem ter em conta ou não.

O domínio da solução encontra-se na Figura 9.

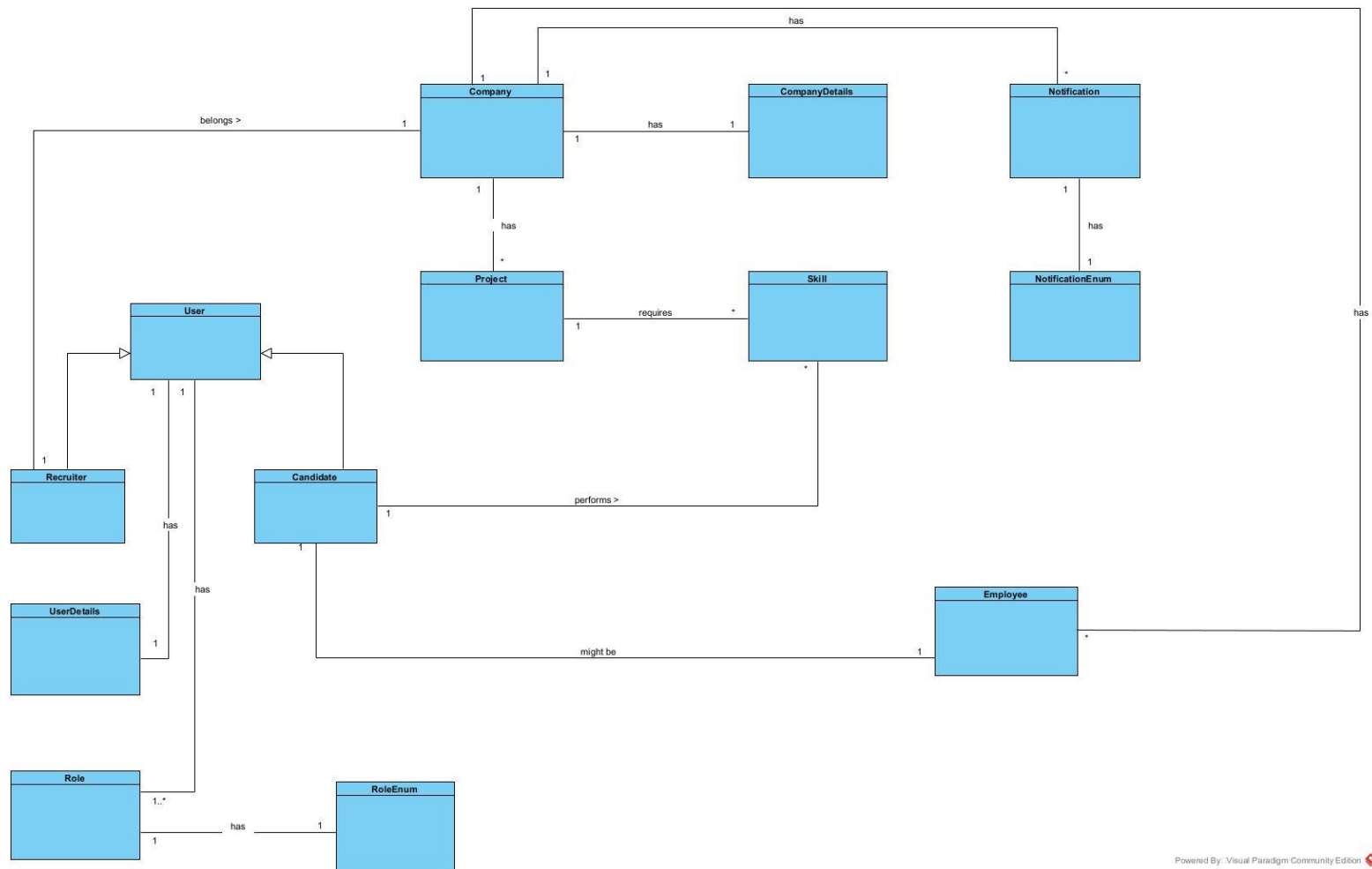


Figura 9 - Domínio da Solução

3.4 Casos de Uso do Sistema

Conforme previamente mencionado, existem três tipos de atores neste sistema: o Candidato, o Recrutador e o Empregado, sendo que todos eles executam funcionalidades distintas no sistema.

O utilizador com mais funcionalidades no sistema é o Recrutador. O Recrutador tem a capacidade de realizar todas as operações de gestão de empresas, todas as operações de gestão de projetos dentro de uma empresa e ainda a capacidade de associar e desassociar empregados a projetos. Para além disto, o utilizador Recrutador pode acionar ambos os sistemas multiagente responsáveis pelas integrações em SAP e pelas alocações automáticas de empregados a projetos/funções. Pode ainda realizar esta última ação de forma manual.

O Candidato apenas tem uma funcionalidade distinta no sistema: a capacidade de visualizar as suas informações com que se registou. O utilizador Empregado, de uma forma semelhante, pode visualizar as suas informações de registo e ainda, adicionalmente, as da empresa que representa.

Existem, naturalmente, funcionalidades comuns aos atores Recrutadores e Candidatos que é a funcionalidade de iniciar sessão no sistema e ainda a de registo. As funcionalidades todas estão presentes na Figura 10.

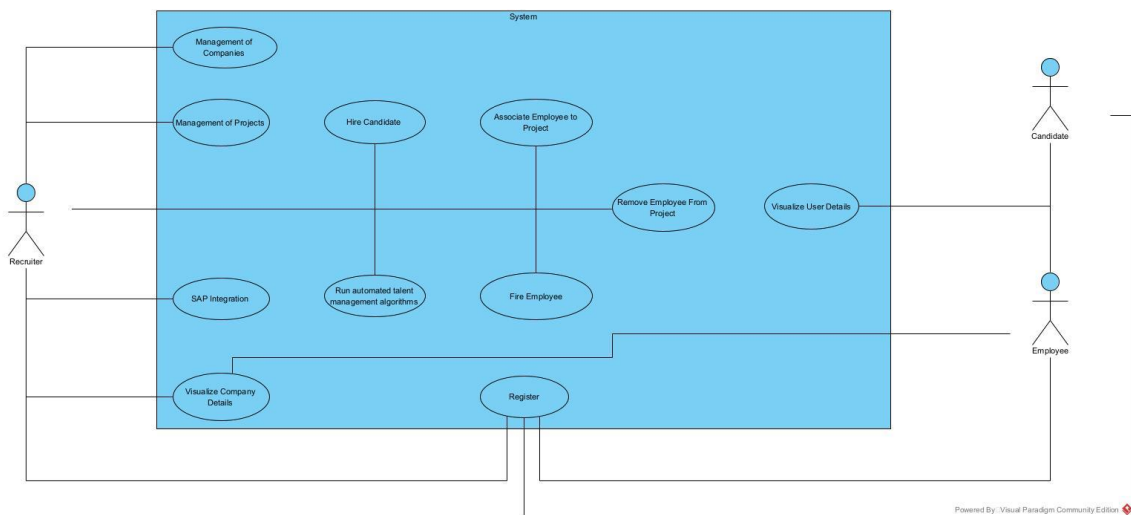


Figura 10 - Diagrama de Casos de Uso

3.5 Arquitetura da BD

A arquitetura de BD reflete o domínio e o sistema já descritos. A base de dados foi, conforme previamente mencionado, desenvolvida utilizando PostgreSQL, um sistema de gestão de base de dados relacionais sendo que dados pré-configurados foram automaticamente injetados durante o arranque da aplicação utilizando scripts Flyway, uma ferramenta *open-source* de migração de dados.

Relativamente ao modelo relacional, o projeto é composto por 21 tabelas relacionais.

De uma forma geral, o projeto contém as tabelas *User*, *Company*, *Skill*, *Project*, *Candidate*, *Recruiter*, *Employee*, *Role* e *Notification*. Estas tabelas contêm campos que descrevem cada um dos objetos que representam. As restantes tabelas que compõe o projeto representam, no fundo, as relações de associação entre as tabelas anteriormente mencionadas. Todas as tabelas de associação têm no seu nome a composição das entidades que estão a associar. A título de exemplo, a tabela do modelo relacional *ProjectSkill* relaciona os projetos com as competências que este necessita.

O modelo relacional está visualmente presente na Figura 11.

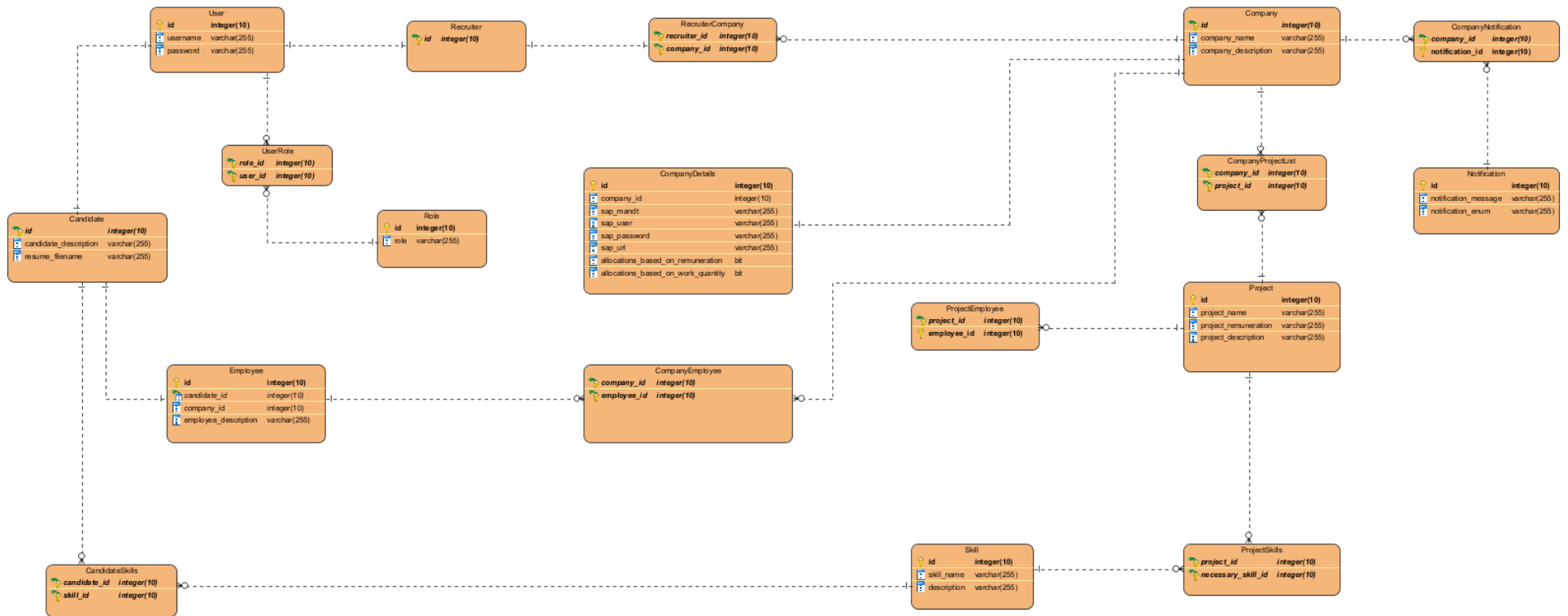


Figura 11 – Modelo Relacional da Solução

3.6 Datasets de treino

Nesta subsecção são explorados os datasets utilizados para o treino dos modelos.

3.6.1. Informações Gerais do Dataset da Extração de Informação de Currículos

Para a extração das informações dos currículos dos funcionários, foi utilizado um *dataset* com currículos de engenheiros de software indianos, portanto, direcionado à área das tecnologias da informação.

Para o desenvolvimento deste modelo, foi complexo encontrar um dataset já com as palavras essenciais pré-classificadas, devido aos problemas de RGPD que a partilha de currículos acarreta. Foi usado um *dataset* encontrado na plataforma Kaggle, ainda que este tivesse poucas entradas e específicas ao domínio das tecnologias da informação. Este *dataset* já contém uma pré-análise que identifica as competências em currículos, e ainda outras informações como local de estudos, localizações geográficas e grau de educação.

O *dataset* está disponível no Kaggle (<https://www.kaggle.com/datasets/dataturks/resume-entities-for-ner>) e contém cerca de 8000 downloads e 71000 visualizações. A grande quantidade de downloads deve-se ao facto de este tipo de dados ser relativamente raro, devido às restrições impostas pelo RGPD. Para além disso, o *dataset* contém alguns *notebooks* que providenciam ajuda importante à implementação dos modelos preditivos de NER.

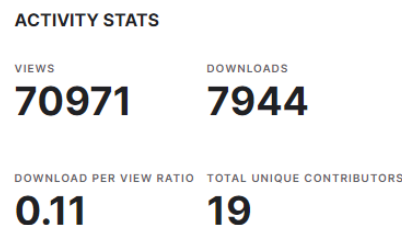


Figura 12 – Informações gerais do *dataset* dos Currículos.

3.6.2. Informações Gerais do Dataset dedicado à inferência de competências

Relativamente à inferência de competências, foi utilizado um *dataset* que relaciona os países com as línguas que são faladas nesses países. O *dataset* é público e encontra-se no Kaggle (<https://www.kaggle.com/datasets/zinovadr/language-list-by-country-and-place>). A baixa quantidade de downloads pode estar relacionada com o facto de ser um tipo de *dataset* relativamente comum.

Activity Overview

ACTIVITY STATS

VIEWS

904

DOWNLOADS

50

DOWNLOAD PER VIEW RATIO TOTAL UNIQUE CONTRIBUTORS

0.06

2

Figura 13 - Informações gerais do *dataset* de linguagens

3.7 Questões de Segurança – Como ultrapassar?

A concretização deste sistema envolve a recolha de dados pessoais, sensíveis de transmissão, pelo que teve que ser tido cuidado com as mais recentes leis da proteção de dados.

De acordo com [64], existem várias formas de contornar este problema devido à finalidade do sistema. Na realidade, de acordo com a mesma referência, desde que adquirido o consentimento do utilizador para a utilização da aplicação não seria necessário fazer nada pois a execução das funcionalidades pressupõe os dados dos quais o titular faz parte.

De qualquer das maneiras, de forma a ganhar a confiança do titular dos dados e para o sistema funcionar da forma mais transparente possível, o sistema pede a confirmação dos dados identificados e inferidos dos currículos.

Nenhum utilizador candidato pode utilizar e beneficiar das funcionalidades do sistema se não consentir e confirmar com os dados identificados pelos algoritmos implementados. Futuramente, fora do contexto de prova de conceito, será possível um candidato pedir confirmação dos dados identificados a um administrador de sistema.

Estas restrições devem-se ao facto que as leis da União Europeia não permitem que algoritmos automatizados decidam automaticamente sem validação humana assuntos sensíveis como, por exemplo, o recrutamento. Como forma de ultrapassar este contratempo, o sistema funcionará como um sistema de apoio à decisão, necessitando da confirmação de um utilizador humano para qualquer decisão automatizada com os modelos preditivos.

3.8 Modelos Utilizados para a extração e inferência de competências

A solução proposta propõe-se a extrair informações de currículos e a realizar inferência de competências sendo que, para isso, foram desenvolvidos dois modelos distintos. O primeiro modelo está relacionado com o problema clássico de *Named Entity Recognition* (NER) para a identificação de informações de candidatos através da análise de textos descritivos ou de currículos. O segundo modelo está, por sua vez, relacionado com a inferência de competências através da análise da informação identificada anteriormente durante a extração de informação dos currículos. Desta forma, a ordem lógica do sistema, conforme presente na Figura 14 é que primeiro ocorra o pedido para extrair as informações dos currículos, e, apenas depois, tendo por base as informações identificadas, é que conseguimos prever competências adicionais. Sendo este projeto uma prova de conceito, a inferência só foi feita para as sugestões de línguas que um candidato sabe falar, com base nas localizações geográficas identificadas no currículo, após o processo de *Named Entity Recognition*.

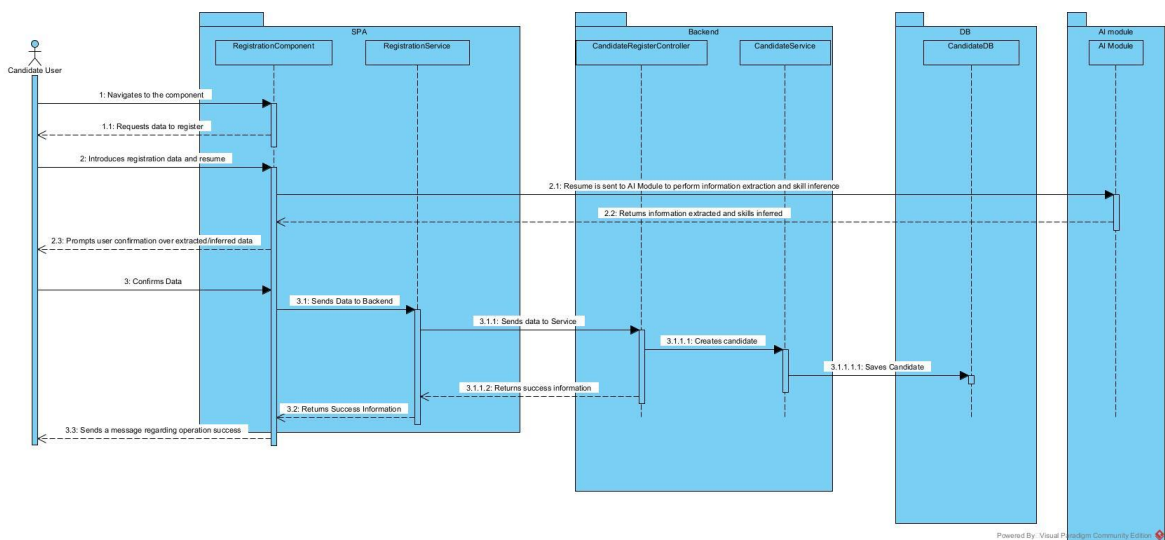


Figura 14 - Diagrama de sequência da extração de competências usando o módulo de IA

Arquiteturalmente, planeou-se usar dois modelos distintos. Um intimamente relacionado com a extração de informações de currículos e o outro com a inferência de competências.

Planeou-se ainda, utilizar o Tensorflow de forma a treinar o modelo utilizando o GPU do computador local. As especificações de *hardware* de onde os modelos foram treinados estão presentes na Tabela 11.

Tabela 11 – Especificações do Computador Usado Para Treinar os Modelos Preditivos

Componente de Computador	Especificação
CPU	AMD Ryzen 7 5800X 8-Core Processor (16 CPUs), 3.8Ghz
GPU	NVIDIA RTX 3060TI, 8GB GDDR6 LHR
RAM	16GB (8*2) DDR4-3200Mhz

3.8.1 Extração de Informações de um currículo

Para a extração de informações e competências de currículos, de acordo com a revisão literária, decidiu-se que seria desenvolvida uma rede neuronal de aprendizagem profunda, em particular, uma Bi-LSTM com *Conditional Random Fields* para a simulação da percepção do contexto. Para esta rede neuronal seria fornecido um *dataset* com palavras pré-classificadas e esta rede teria a responsabilidade de as extrair se as reconhecesse em qualquer currículo.

3.8.2 Inferência de competências

Para a inferência de competências, tendo em conta, uma vez mais, a revisão literária e ainda que as competências inferidas seriam apenas com base nas línguas que o utilizador fala, identificadas pelas localizações onde já esteve, reconhecidas no âmbito do processo anterior, decidiu-se que um modelo mais simples de aprendizagem automática seria suficiente. Tendo em conta que este é um problema simples no qual uma abordagem que não recorra a algoritmos de IA seria possível, é expectável que vários modelos obtivessem bons resultados. Deste modo, planeou-se que vários modelos de aprendizagem automática seriam testados.

3.9 Arquitetura dos sistemas multiagente

Na fase de desenho, foram definidos dois tipos de sistemas multiagente: um direcionado ao emparelhamento e recomendação de empregados a projetos/tarefas, e o outro vocacionado à integração dos dados de uma empresa com o sistema SAP associado. A possibilidade de integração foi adicionada pois a aplicação foi desenhada como sendo um *plugin* ao sistema de gestão empresarial, em particular, ao módulo de talentos do sistema SAP ERP.

O desenvolvimento dos sistemas multiagente é possível utilizando várias *framework* distintas, cada uma com as suas vantagens e desvantagens. A título de exemplo, a *framework* SPADE, é responsável pela construção de sistemas multiagente na linguagem Python. No entanto, a escolha final não incidiu sobre esta pois, aproveitou-se o facto da infraestrutura do módulo da lógica de negócio estar em JAVA, e, existirem *frameworks* de renome para a implementação de sistemas multiagente nesta. Neste âmbito, existem várias *frameworks* em JAVA como o JADEX, o JASON ou o JADE. Optou-se por escolher JADE, pois as duas primeiras *frameworks* são direcionadas a sistemas multiagente de arquiteturas BDIs ao contrário de JADE. Face à natureza do domínio do problema em questão, optou-se por utilizar JADE que oferece

mecanismos de comunicação de *Directory Facilitator* e ainda respeita os protocolos de comunicação FIPA, que são mais valias para a extensibilidade e suporte que se espera que a solução final tenha.

3.9.1 Sistema Multiagente responsável pela integração com SAP

No que diz respeito ao desenho arquitetural do sistema multiagente responsável pela integração dos dados das empresas com os sistemas SAP foram planeados 2 agentes.

Um dos agentes, o *SystemAgent* (comum a ambos os sistemas multiagentes) é considerado o sistema coordenador deste processo. Ou seja, planeou-se um sistema centralizado devido ao facto de apenas conter um único coordenador. Este agente recebe todos os dados de uma empresa, e os detalhes de conexão para se ligar ao sistema SAP. Para além disso, este agente implementa esta funcionalidade através de um *Behaviour*, que é executado uma única vez, de acordo com o desejo do utilizador (*OneShotBehaviour*)

O segundo agente, o agente integrador (*IntegrationWithSAPAgent*), é acionado por via da comunicação do *SystemAgent*. Este recebe as informações do *SystemAgent* previamente, e é o responsável pela integração e comunicação com o sistema SAP.

Em baixo, na Figura 15, está presente a arquitetura do sistema multiagente acima descrita.

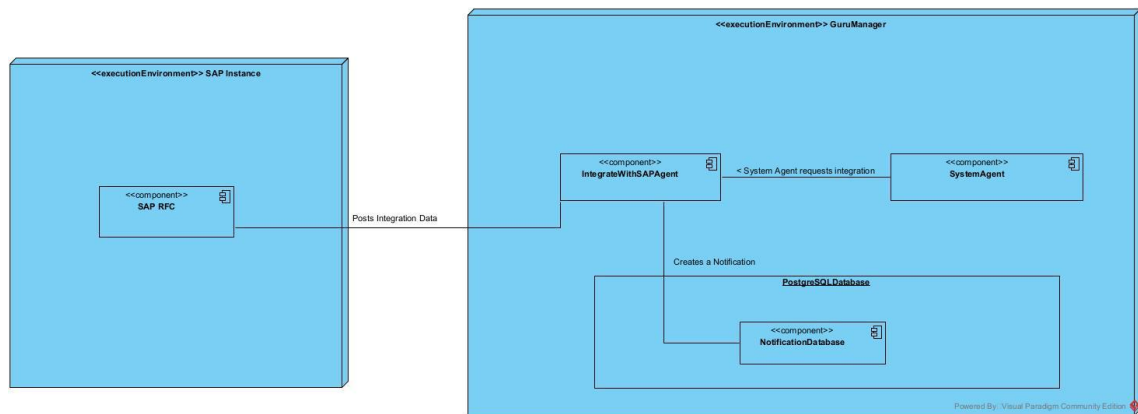


Figura 15 - Figura dedicada à arquitetura do SMA desenvolvido para a integração com SAP

3.9.2 Sistema Multiagente responsável pelo processo de alocação de funcionários e recomendação de candidatos

Relativamente ao sistema multiagente responsável pelo processamento automatizado de alocações e recomendações de funcionários a projetos, este conta com 4 tipos de agentes distintos.

O *SystemAgent* que, uma vez mais, funciona como coordenador único do processo. Tendo em conta isto, podemos inferir que este subsistema é também um sistema centralizado que utiliza a partilha de resultados para almejar uma solução para o problema global. Este agente, para esta funcionalidade, contém, também, um *behaviour*, que executa quando solicitado.

O *EmployeeAgent* representa um funcionário. Estes agentes fornecem os dados de um funcionário de uma empresa quando solicitado. Os dados fornecidos são utilizados, posteriormente, pelo *AllocationsAgent*, o agente que contém os algoritmos de escalonamento sendo estes seleccionados pelo utilizador do sistema.

Existe, ainda, uma gente que é o *GenerationDataAgent*, que tem apenas o propósito de gerar dados fictícios para uma empresa. Este agente não existiria numa solução final sendo que o seu propósito é preencher as bases de dados de forma que os algoritmos de alocações sejam devidamente testados.

Em baixo, na Figura 16 está presente um esquema onde é possível visualizar a arquitetura planeada.

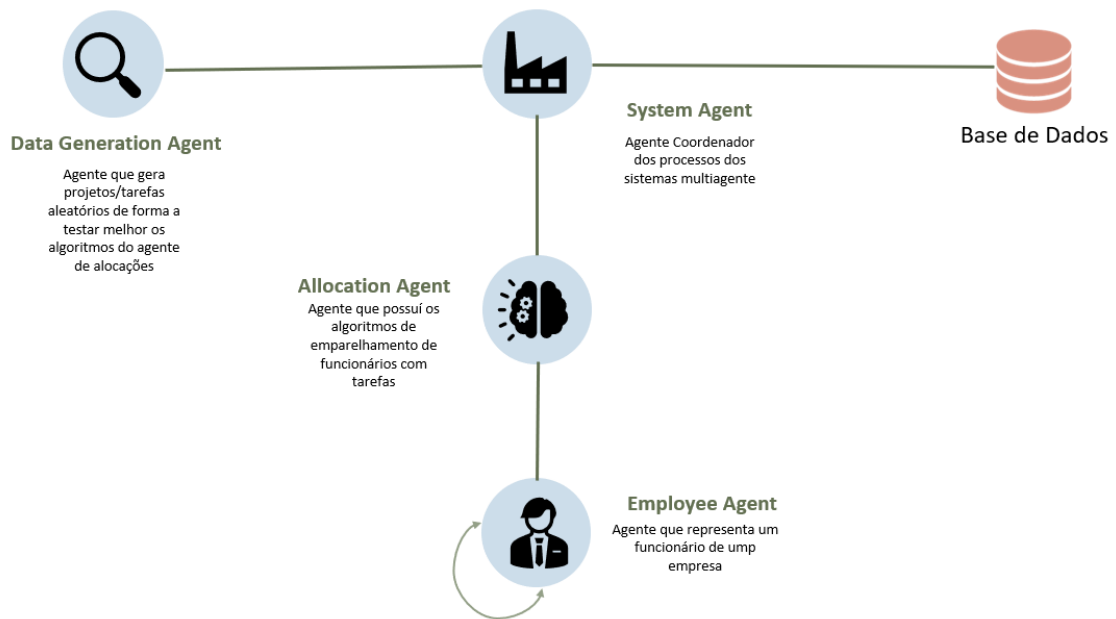


Figura 16 - Arquitetura do SMA responsável pelas alocações e recomendações de funcionários a tarefas

3.10 Algoritmo Genético responsável pelo emparelhamento entre empregados e tarefas

Conforme provado pela literatura realizadas nas secções 3 e 4, os algoritmos genéticos são aplicáveis de forma a resolver o problema de emparelhamento otimizado entre funcionários e tarefas. Por conseguinte, o utilizador pode, para além dos algoritmos lineares incluídos nos SMA, utilizar um algoritmo genético implementado, também, no *AllocationsAgent*. A implementação deste algoritmo está mais detalhadamente explicada na secção de implementação.

Para a implementação deste algoritmo decidiu-se utilizar a biblioteca Jenetics. Esta biblioteca é *open-source* e representa uma das mais bem documentadas e utilizadas para o uso de algoritmos genéticos em Java. Tendo em conta que este algoritmo está inserido nos SMA era imperativo que a implementação deste algoritmo fosse também neste módulo. De qualquer das maneiras, existem várias bibliotecas para este efeito, inclusive da Apache. A forte documentação e facilidade de customização foram os fatores decisivos para a decisão da utilização do Jenetics.

4. Implementação

O objetivo desta secção é explicar e descrever características técnicas implementadas na solução. Descrevem-se, ainda, algumas das interfaces presentes na aplicação de forma a explicar a sua usabilidade e necessidade.

4.1 Pré-Processamento dos Datasets

Nesta secção os passos de pré-processamento dos dados assim como a descrição das suas *features* são fornecidos. Esta secção está dividida em duas, cada uma dedicada ao respetivo *dataset*.

4.1.1. Pré-Processamento do Dataset da Extração de Informação de Currículos

O *dataset* obtido do Kaggle antes do pré-processamento encontra-se em formato JSON. Este é caracterizado por ter dois atributos: o primeiro atributo contém todo o currículo de um engenheiro de software em formato textual, o segundo atributo é um mapa, onde algumas palavras do currículo são associadas a uma categoria, de forma, a sinalizar informação relevante tais como competências, locais de estudo, grau de educação, entre outros...

```
"content" :
string "Abhishek Jha Application Development Associate - Accenture Bengaluru, Karnataka - Email me on Indeed: indeed.com/r/Abhishek-
Jha/18e7a8cb732bc43a • To work for an organization which provides me the opportunity to improve my skills and knowledge for my
individual and company's growth in best possible ways. Willing to relocate to: Bangalore, Karnataka WORK EXPERIENCE Application
Development Associate Accenture - November 2017 to Present Role: Currently working on Chat-bot. Developing Backend Oracle PeopleSoft
Queries for the Bot which will be triggered based on given input. Also, Training the bot for different possible utterances (Both
positive and negative), which will be given as input by the user. EDUCATION B.E in Information science and engineering B.v.b college
of engineering and technology - Hubli, Karnataka August 2013 to June 2017 12th in Mathematics Woodbine modern school April 2011 to
March 2013 10th Kendriya Vidyalaya April 2001 to March 2011 SKILLS C (Less than 1 year), Database (Less than 1 year), Database
Management (Less than 1 year), Database Management System (Less than 1 year), Java (Less than 1 year) ADDITIONAL INFORMATION Technical
Skills https://www.indeed.com/r/Abhishek-Jha/18e7a8cb732bc43a?isid=rex-download&ikw=download-top&co-IN • Programming language: C, C++,
Java • Oracle PeopleSoft • Internet Of Things • Machine Learning • Database Management System • Computer Networks • Operating System
worked on: Linux, Windows, Mac Non - Technical Skills • Honest and Hard-Working • Tolerant and Flexible to Different Situations •
Polite and Calm • Team-Player"
"annotation" : [ 14 items
  0 : { 2 items
    "label" : [ 1 item
      0 : string "Skills"
    ]
    "points" : [ 1 item
      0 : { 3 items
        "start" : int 1295
        "end" : int 1621
        "text" :
string " • Programming language: C, C++, Java • Oracle PeopleSoft • Internet Of Things • Machine Learning •
Database Management System • Computer Networks • Operating System worked on: Linux, Windows, Mac Non - Technical
Skills • Honest and Hard-Working • Tolerant and Flexible to Different Situations • Polite and Calm • Team-Player"
      }
    ]
  }
  1 : { ... } 2 items
```

Figura 17 - Exemplo de currículo no *dataset* JSON

De forma a facilitar todo o processo, realizou-se, em primeiro lugar, um pré-processamento do *dataset* onde se extraíram as palavras que foram classificadas e a sua *tag* do *dataset* JSON. Para cada palavra e *tag* foi-lhe atribuído um número, pois os modelos de aprendizagem profunda precisam de uma representação numérica do texto que estão a treinar. Para além disto aplicou-se a técnica *word2vec*, que representa palavras através de vetores e através destes consegue detetar associações de palavras, sinónimos e analogias caso haja semelhanças vectoriais. A transformação de variáveis categóricas nominais em representações numéricas, foi também realizada.

Importa, ainda, salientar que na fase de pré-processamento foram retiradas as *stopwords*. Estes tipos de palavras, tipicamente, resumem-se a determinantes, conectores e palavras que por si só não fornecem informação. Ao removê-las, os modelos de extração de informação focam-se apenas nas palavras que realmente fornecem informação, para além que o seu treino é mais eficaz.

4.1.2. Pré-Processamento do Dataset da Inferência de Competências

O *dataset* extraído para a inferência das competências é, em primeiro lugar, composto por duas *features*. Uma das *features* é relativo a um país, e a outra à língua falada nesse mesmo país. Deste modo, podemos afirmar que ambas as *features* são do tipo categórico nominal. Na Figura 18, temos 4 entradas que estão presentes no *dataset*.

149	Country	Language
150	Poland	Polish
151	Portugal	Portuguese
152	Qatar	Arabic, English
153	Romania	Romanian, Hungarian, German

Figura 18 - Entradas do *dataset* que relaciona países com línguas faladas

As variáveis, devido ao facto de serem do tipo categórico nominal, exigiram um pré-processamento, nomeadamente uma representação numérica. Excluiu-se a utilização de técnicas como o *LabelEncoding* devido ao facto de estas criarem correlações e associações que os modelos muitas vezes interpretam de uma forma errada. Desta forma, a escolha incidu sobre o *OneHotEncoding* ou um *BinaryEncoding*. Decidiu-se utilizar a segunda opção, pois a existência de cerca de 80 línguas no *dataset* implicaria problemas de *performance* ao nível da memória. Foi utilizado o *BinaryEncoding* da biblioteca *sklearn*, que, passa as variáveis categóricas nominais para um tipo categórico numérico e depois transforma a base decimal numa base binária o que permite representar mais categorias num menor número de colunas. Desta forma, é possível ser mais eficiente ao nível da memória utilizada.

É expectável que, os dados presentes na coluna dos países no *dataset* coincidam com os dados dos currículos que os modelos da extração de informação vão classificar como sendo da entidade "*Location*". De forma a garantir que os modelos de aprendizagem automática

entendem estes países, que são dados nominais, foi necessário proceder à sua alteração para dados de tipo numérico. Para isso, representaram-se estes textos através de uma matriz numérica através da biblioteca *sklearn*, nomeadamente, da classe *CountVectorizer*.

Um outro aspeto importante deste *dataset*, prende-se com a fraca quantidade de entradas e com o facto de vários países falarem a mesma língua, o que pode confundir modelos de aprendizagem automática e criar tendências. De forma a ultrapassar isto, duplicaram-se as entradas das classes minoritárias, isto é, das línguas menos faladas. Isto foi alcançado através da biblioteca *imblearn*, em específico, da sua classe *RandomOverSampler*.

Após estas etapas de pré-processamento, os dados estavam preparados para serem fornecidos ao modelo preditivo de aprendizagem automática para que este pudesse, dentro do contexto do sistema, realizar as previsões esperadas.

4.2 Extração automatizada de informação de currículos

Tendo em conta os resultados da revisão literária realizada no segundo capítulo, foi desenvolvida uma rede neuronal BiLSTM. Estes tipos de técnicas pertencem às técnicas de reconhecimento de entidades denominadas de NER (*Named Entity Recognition*), e reconhecem em grandes quantidades de texto entidades para a qual o modelo preditivo foi treinado.

Este tipo de técnicas é, tipicamente, do tipo supervisionado. O *dataset* de currículos extraído do Kaggle continha cerca de 200 currículos de, na sua maioria, engenheiros de software indianos, que já estavam pré-analisados e com entidades relevantes já identificadas, tais como: locais de trabalho, experiência, competências, entre outros... Desta forma, ao ser uma aprendizagem do tipo supervisionada, o modelo preditivo escolhido treina sobre esses dados e realiza previsões.

Um detalhe de implementação, foi que para o treino dos dados foi utilizado a técnica *word2vec*, que permite fazer com que palavras e *tokens* reconhecidos em textos de grandes dimensões, possam ser representados através de vetores, para que seja possível estabelecer correlações e associações de forma automática.

A arquitetura da rede neuronal desenvolvida assumiu o aspeto presente na Figura 19.

A primeira camada da rede neuronal, a *InputLayer*, tem como principal função receber as palavras de cada frase de *input*, sendo que cada frase tem um máximo de 22 palavras. A escolha deste valor deve-se ao tamanho médio das frases dos currículos todos. Este valor foi extraído da análise exploratória dos dados.

Na camada seguinte, foram aplicados a técnica *word2vec* que são um tipo de *Embeddings* que têm como principal função perceber e capturar palavras com relações e associações entre si, de forma que, palavras semelhantes em frases diferentes, mas com um contexto também semelhante (no fundo, sinónimos) sejam, também, capturados.

De seguida, é aplicada a camada Bi-LSTM que, tem como função também perceber o contexto e palavras envolventes de forma que a rede neuronal possua uma simulação de memória.

Por último, o *output* da terceira camada é transmitido para a quarta camada (*TimeDistributed*) aplicando uma função de ativação *softmax*.

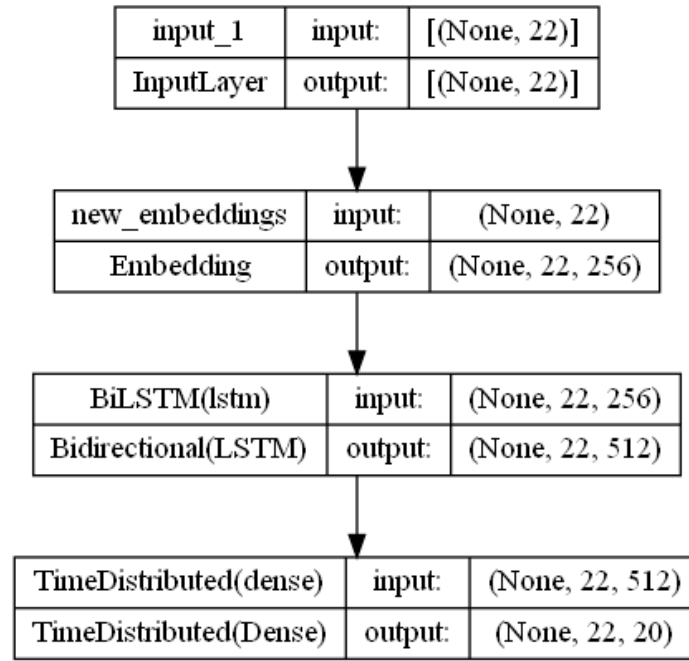


Figura 19 - Arquitetura da rede neuronal NER para a extração de informação de currículos

O modelo foi treinado para 200 *epochs*, com um split dos dados de 70% dos dados para treino, 15% para testes e 15% para validação. Devido à pequena quantidade de dados, pois o uso de currículos para estas situações põe em causa algumas das leis da proteção de dados, o modelo ficou *overfitted* reconhecendo demasiado bem as situações para as quais treinou e tendo alguma dificuldade para situações novas. De qualquer das maneiras, em alguns dos reconhecimentos, como universidades e estudos obtidos, o modelo consegue, com alguma facilidade, reconhecer casos diferentes dos quais treinou. De seguida, na Figura 20, está presente um sumário do modelo desenvolvido, figura complementar à Figura 19.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 22)]	0
new_embeddings (Embedding)	(None, 22, 256)	2524672
BiLSTM (Bidirectional)	(None, 22, 512)	1050624
TimeDistributed (TimeDistributed)	(None, 22, 20)	10260

=====
Total params: 3,585,556
Trainable params: 3,585,556
Non-trainable params: 0

Figura 20 - Sumário do modelo desenvolvido

Na Figura 21, estão presentes as métricas de performance obtidas para o modelo implementado por categoria.

	precision	recall	f1-score	support
B-College	0.93	0.86	0.89	29
B-Company	0.89	0.80	0.84	61
B-Degree	0.61	0.52	0.56	21
B-Designation	0.80	0.84	0.82	38
B-Email	0.78	1.00	0.88	14
B-GradYear	0.64	0.35	0.45	26
B-Location	0.00	0.00	0.00	2
B-Name	0.94	0.70	0.80	23
B-Skills	0.72	0.59	0.65	69
B-YearsExper	0.60	0.60	0.60	5
I-College	0.89	0.88	0.88	81
I-Company	0.69	0.75	0.72	36
I-Degree	0.74	0.69	0.71	58
I-Designation	0.79	0.89	0.84	61
I-Name	0.94	0.71	0.81	24
I-Skills	0.84	0.68	0.75	340
I-YearsExper	1.00	1.00	1.00	8
0	0.99	1.00	0.99	34766
accuracy			0.99	35662
macro avg	0.77	0.71	0.73	35662
weighted avg	0.99	0.99	0.99	35662

Figura 21 - Classificações obtidas do modelo com word2vec

As métricas de performance presentes na Figura 21 comprovam, através das média com pesos, que o modelo está *overfitted* para a situação para o qual foi treinado.

Importa, no entanto, salientar que este resultado apenas foi alcançado após o *tuning* de parâmetros das redes neurais, nomeadamente, o número de *units* e *dropout* da camada LSTM. As classificações obtidas pela rede neuronal sem o *tuning*, está presente na Figura 22.

	precision	recall	f1-score	support
B-College	0.00	0.00	0.00	151
B-Company	0.00	0.00	0.00	371
B-Degree	0.00	0.00	0.00	141
B-Designation	0.00	0.00	0.00	271
B-Email	0.08	0.06	0.07	137
B-GradYear	0.00	0.00	0.00	131
B-Location	0.00	0.00	0.00	52
B-Name	0.14	0.01	0.01	159
B-Skills	0.00	0.00	0.00	530
B-YearsExper	0.00	0.00	0.00	24
I-College	0.00	0.00	0.00	374
I-Company	0.00	0.00	0.00	198
I-Degree	0.00	0.00	0.00	337
I-Designation	0.00	0.00	0.00	450
I-Email	0.00	0.00	0.00	3
I-Location	0.00	0.00	0.00	2
I-Name	0.50	0.04	0.08	167
I-Skills	0.59	0.05	0.08	2695
I-YearsExper	0.00	0.00	0.00	20
0	0.98	1.00	0.99	250461
accuracy			0.98	256674
macro avg	0.11	0.06	0.06	256674
weighted avg	0.96	0.98	0.97	256674

Figura 22 – Resultados da rede neuronal sem *tuning* de parâmetros

O *tuning* dos parâmetros das redes neurais foi alcançado através da técnica de *Grid Search*. Esta é uma técnica exaustiva onde várias instâncias do mesmo modelo são testadas, cada um com configurações de parâmetros distintas. Seleciona-se o modelo que obteve melhores resultados nos testes.

O *endpoint* da previsão de competências foi disponibilizado para o módulo JAVA consumir através de uma API em Flask que se encontra no url “localhost:5000/skills/resume”. A API, por sua vez, retorna para o módulo do backend JAVA um mapa com as competências que reconheceu agrupados por tipo. O módulo JAVA, devido a efeitos legais obrigatórios, requer a confirmação do utilizador relativamente às suas competências previstas. Este ecrã de confirmação está presente na Figura 23.

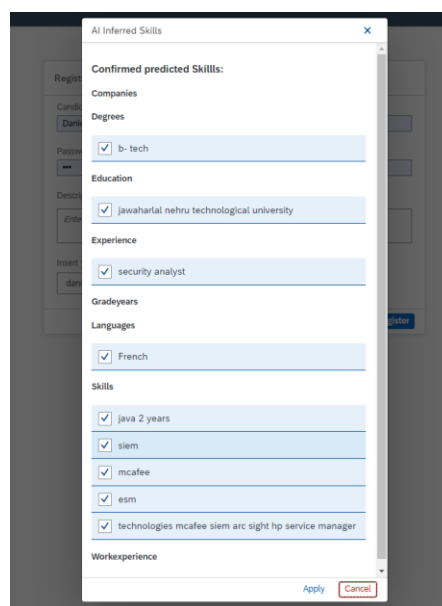


Figura 23 – Confirmação das competências extraídas por técnicas de inteligência artificial

4.3 Inferência de competências

A inferência de competências dos funcionários almeja alcançar a capacidade de reconhecer e extrair competências, que não estão explícitas textualmente após uma primeira leitura de um currículo. A revisão literária realizada no segundo capítulo comprovou que os melhores modelos para esta situação dependem muito do tipo de informação que se está a tentar inferir.

No âmbito do projeto desenvolvido, a inferência de competências limitou-se a inferir a capacidade de um candidato falar uma determinada, com base em países onde este já tenha trabalhado ou vivido. A inferência de competências é uma componente do sistema que pode facilmente ser estendida no futuro, para inferir outro tipo de competências. No entanto, atendendo à natureza de prova de conceito do projeto desenvolvido, limitou-se, nesta fase, esta funcionalidade apenas a esta situação.

De forma a atingir o proposto, implementaram-se modelos preditivos que inferem estas competências imediatamente após a extração automatizada da informação de currículos, aliás, estes dependem diretamente dos resultados provenientes da extração de informação, pois preveem línguas faladas com base em alguns dos *tokens*, classificados como sendo “*Location*”, durante o primeiro processo da extração de informação.

Tendo em conta a pequena quantidade de dados envolvida no processo que relaciona as localizações com as línguas faladas, os modelos obtiveram resultados extremamente satisfatórios. Diz-se que este é um problema de classificação pois o objetivo é, com base no país identificado, extrair a língua que é falada neste. Foram testados vários modelos de aprendizagem automática.

Em baixo, na Tabela 12, encontram-se os resultados obtidos por cada um dos modelos preditivos testados.

Tabela 12 - Resultados modelos de inferência de capacidades

	Precision	Recall	Acurácia	F1-Score
Random Forest	1	1	1	1
KNN	1	1	1	1
XGBoost	1	1	1	1
Naive-Bayes	0.99	0.98	0.99	0.99
Logistic Regression	0.99	0.98	0.99	0.98

Os resultados de todos os modelos foram extremamente satisfatórios tendo em conta que é um problema de classificação simples. De qualquer forma, com base nos resultados presentes na Tabela 12, a solução final utiliza *Random Forests*. Esta escolha deve-se, para além dos resultados obtidos, que à partida excluíram os modelos *Naive-Bayes* e *Logistic Regression*, aos recursos computacionais que cada um dos restantes modelos utiliza e aos seus tempos de treino. Devido à questão dos recursos computacionais excluiu-se a utilização do XGBoost visto que, embora tenha atingido resultados amplamente satisfatórios, a verdade é que utiliza mais recursos que os restantes modelos que também atingiram os mesmos resultados. Relativamente à escolha entre KNN e *Random Forests* esta foi devido, exclusivamente, aos tempos de treino. Nesta situação, a *Random Forests* obteve tempos menores de treino o que fez com que fosse o modelo escolhido para a solução final.

```
Training time Random Forest: 9.435506820678711s
Training time KNN: 10.1428611537591553s
```

Figura 24 - Tempos de Treino entre *Random Forests* e KNN

Para além das piores métricas de *performance*, os modelos de *Naive-Bayes* e *Logistic Regression* foram descartados visto necessitarem da utilização de uma representação numérica proveniente de *LabelEncoding*. O *LabelEncoding* embora seja uma representação válida para a transformação de variáveis categóricas nominais em numéricas, a verdade é que podem criar relações entre as variáveis que não são reais, e que os modelos podem interpretar erroneamente.

Nesta situação, não se realizou a otimização dos hiperparâmetros visto que os resultados obtidos com as definições por defeito dos algoritmos apresentaram, desde logo, resultados bastante satisfatórios.

4.4 Sistemas Multiagente desenvolvido

Esta secção é dedicada aos detalhes de implementação dos SMA desenvolvidos. Este capítulo conta com 3 subsecções sendo que duas delas são dedicadas aos dois subsistemas desenvolvidos, que são distintos nos seus objetivos, e um capítulo é dedicado à integração destes com a *framework* Spring.

De forma a favorecer a extensibilidade e manutenibilidade dos agentes, o processo de criação destes está envolvido num padrão de *design factory*. Desta forma, é possível criar agentes com um esforço mínimo, e, caso seja necessário criar mais tipos de agentes ou retificar erros no processo de criação de agentes existentes de forma simples, genérica e apenas num único local.

A comunicação entre agentes existentes em *runtime* é garantida com os serviços de páginas amarelas do JADE. Estes serviços fazem com que a comunicação entre os agentes existentes seja feita de uma forma escalável e facilmente mantida.

4.4.1 Sistema Multiagente dedicado ao escalonamento do talento interno numa empresa

O subsistema principal da aplicação desenvolvida é dedicado à distribuição dos recursos humanos dentro de uma empresa pelos projetos ou tarefas existentes. Este subsistema tem em consideração restrições existentes como as competências do recurso e tarefas, o valor monetário do projeto e ainda a alocação excessiva de funcionários a tarefas. Uma das grandes vantagens da utilização dos sistemas multiagente neste processo é que são facilmente extensíveis e escaláveis adaptando-se para o problema em questão. Usando SMA, com pouco esforço rapidamente se escala o problema para que inclua mais variáveis restritivas.

Para a concretização do problema proposto foram desenvolvidos 5 tipos de agentes.

- **SystemAgent:** O agent *SystemAgent* deve ser visto como o principal coordenador do processo. O processo de alocação de recursos é coordenado por um *behaviour* designado de “*GenerateJobPairingBehaviour*”. Este *behaviour* é do tipo *OneShotBehavior*, ou seja, corre uma única vez após a interação com um botão na interface gráfica por parte de um utilizador afeto à administração.
- **GenerationDataAgent:** Este agente tem a responsabilidade de realizar a geração de dados utilizados nos algoritmos de emparelhamento. O agente gera projetos com uma remuneração e competências aleatórias. Este agente é do tipo reativo pois espera obrigatoriamente pelo estímulo do agente comunicativo *SystemAgent* para a realização das suas funções.

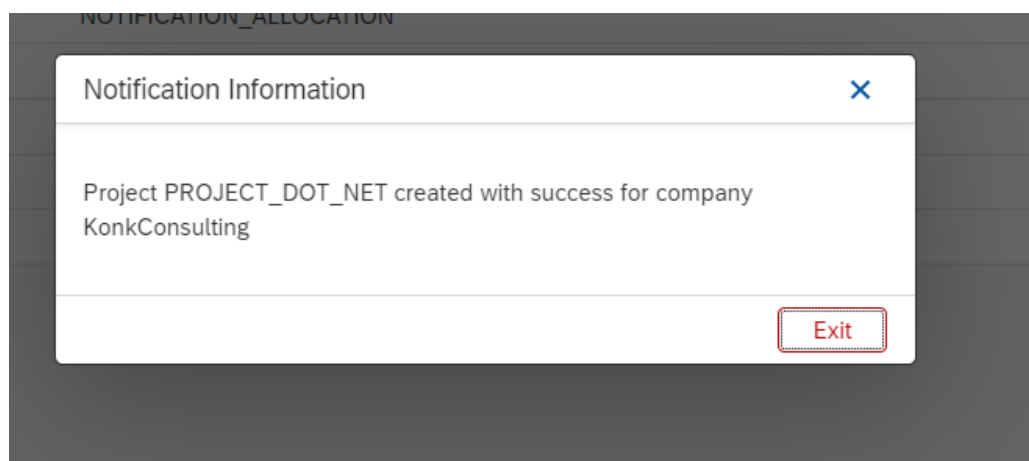


Figura 25 - Notificação de geração de projeto

- **EmployeeAgent:** O agente *EmployeeAgent* é o agente responsável por representar um recurso dentro de uma empresa. A sua única função é fornecer os seus dados

conforme o estímulo do agente comunicativo *SystemAgent*, logo, é também um agente reativo. O *EmployeeAgent* não interage com a base de dados e possui os dados de um funcionário pois estas informações foram-lhe transmitidas por argumento.

- ***AllocationsAgent***: O *AllocationsAgent* é o agente responsável por realizar o escalonamento dos recursos existentes pelas tarefas que estão no ativo numa dada empresa. Este agente contém o algoritmo de escalonamento que é configurável pelos utilizadores para ter em conta, na execução do seu algoritmo, as restrições que o utilizador deseja ter em conta.

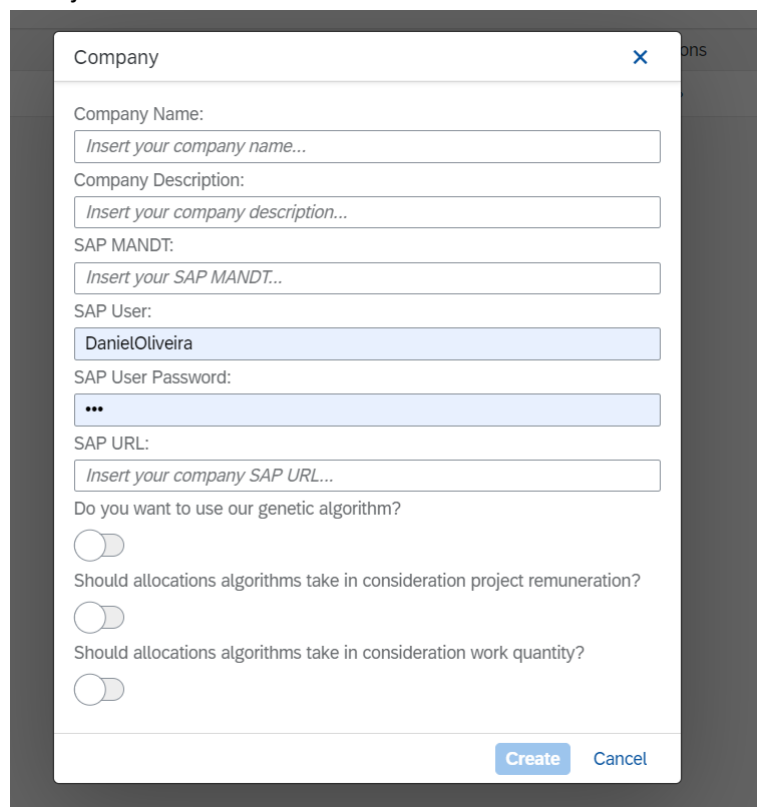


Figura 26 - Página de configurações de uma empresa e do *AllocationsAgent*

É um agente reativo pois realiza as suas funções consoante um estímulo comunicativo do *SystemAgent*. Também gera notificações sempre que realiza uma alocação num projeto.

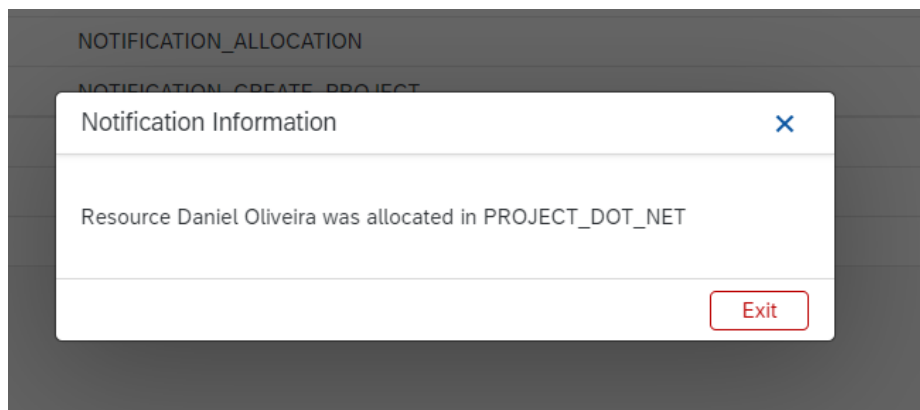


Figura 27 Notificação da alocação de funcionários num projeto.

De seguida, na Figura 28 está presente um diagrama de sequência que ilustra o processo descrito de uma forma sequencial.

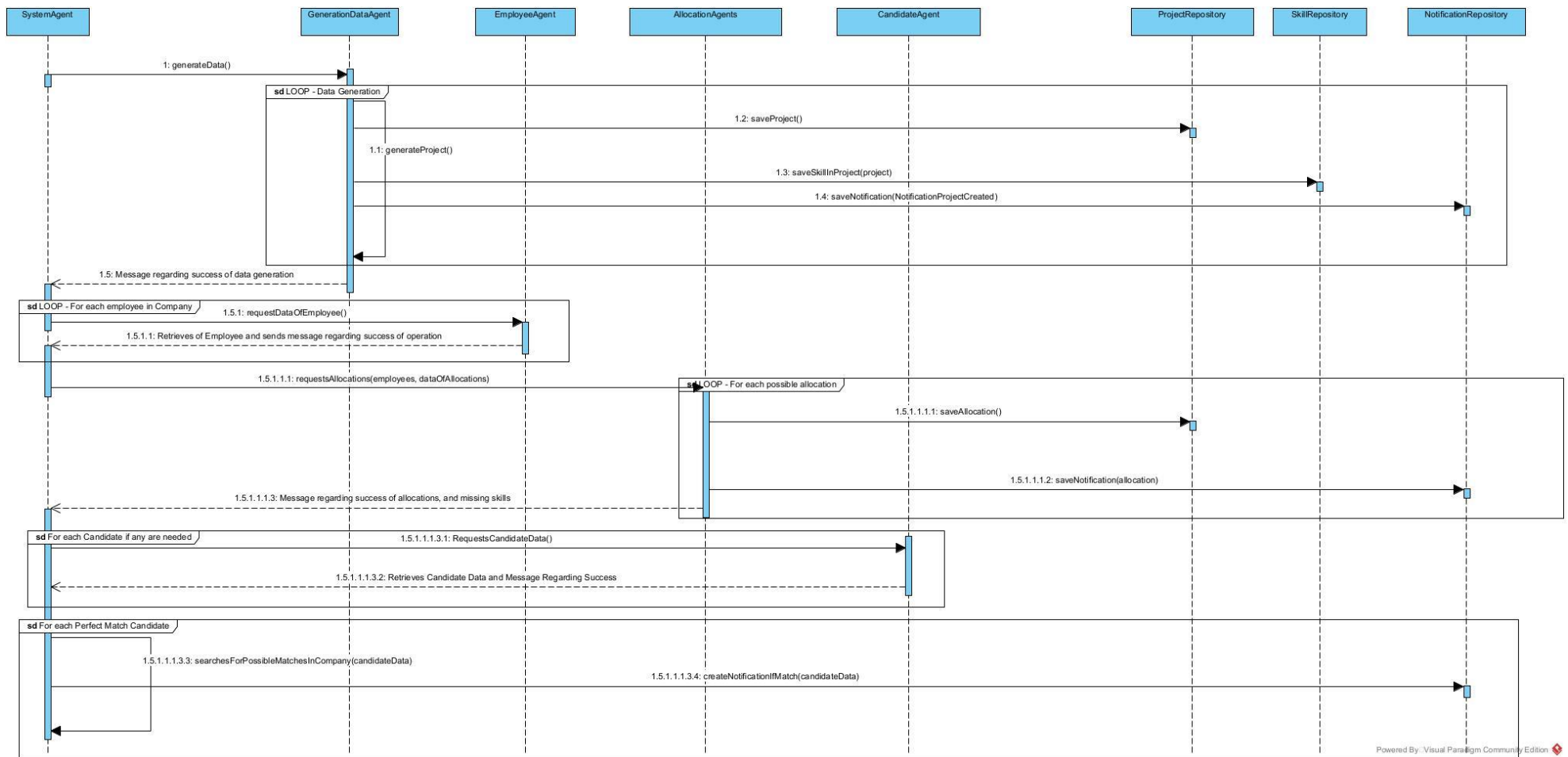


Figura 28 - Diagrama de Sequência do processo de alocações

Este subsistema multiagente contém hierarquias bem definidas sendo que o *SystemAgent* assume um papel de coordenador, deste modo, podemos afirmar que este sistema é do tipo centralizado. Os agentes partilham informações entre si para resolver o problema global das alocações logo é um sistema que realiza coordenação com partilha de resultados.

4.4.2 Sistema Multiagente dedicado à integração com SAP

Relativamente ao sistema multiagente dedicado à integração com o sistema de gestão empresarial SAP, os seguintes agentes estão ligados ao processo:

- **SystemAgent:** O processo de integração com SAP é coordenado por um *behaviour* designado por “*IntegrateWithSAPBehaviour*” que se encontra no *SystemAgent*. Este *behaviour* é do tipo *OneShotBehaviour*, ou seja, corre após solicitado por um utilizador, embora, num contexto de produção este *behaviour* provavelmente fosse do tipo *TicketBehaviour*.
- **IntegrationWithSAPAgent:** Este agente é o responsável por chamar o serviço que vai realizar o pedido para o sistema SAP e efetivamente concretizar a integração. Este agente é do tipo reativo pois espera pelo estímulo comunicativo do *SystemAgent* para realizar a sua função que é enviar os dados do sistema para o sistema empresarial SAP.

De seguida, na Figura 29 está presente um diagrama de sequência que ilustra o processo em cima descrito.

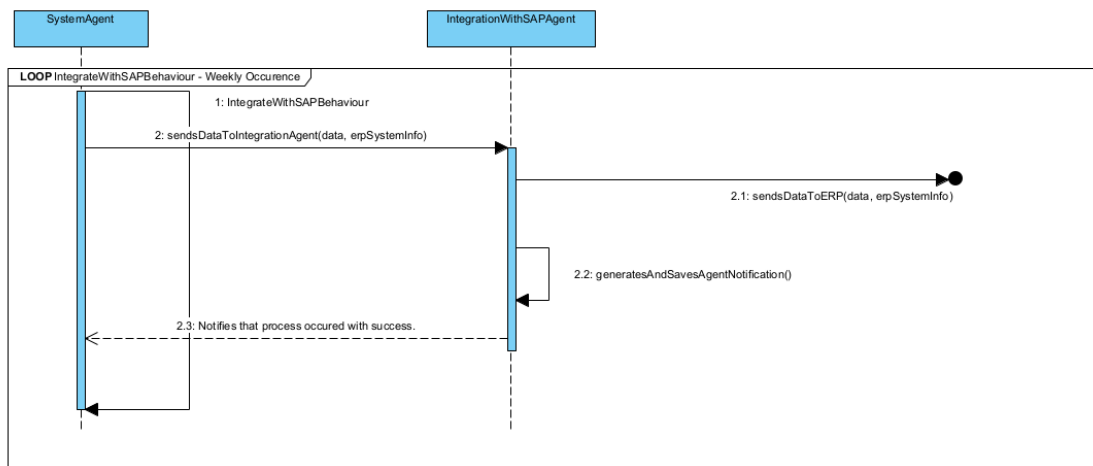


Figura 29 - Processo de integração com sistema SAP

Este subsistema multiagente contém hierarquias bem definidas entre os dois agentes sendo que o *SystemAgent* claramente assume o papel de coordenador único, ou seja, este sistema multiagente é do tipo centralizado.

As interações entre os agentes são do tipo coordenação, sendo que existe partilha de informação para a resolução do problema global. Desta forma, podemos dizer também que este subsistema coordena por comunicação realizando a partilha de resultados.

No fim, é gerada uma notificação e inserida na tabela de notificações de forma a registar que o processo ocorreu com sucesso.

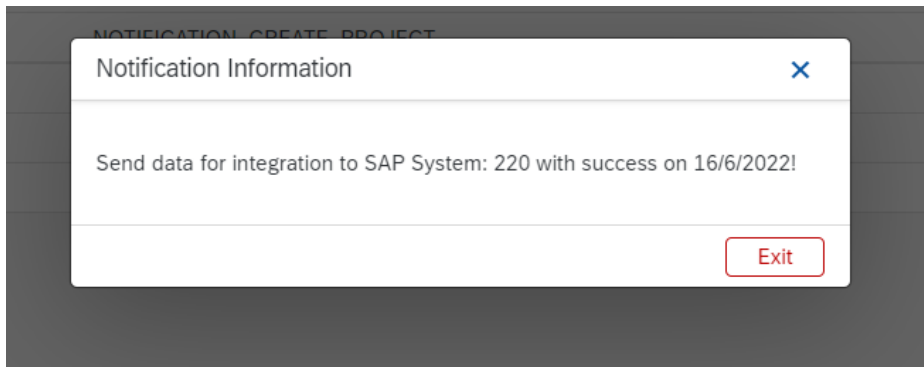


Figura 30 – Notificação de integração com sistema SAP

4.4.3. Integração com Spring

Um dos grandes desafios na implementação do SMA foi a sua integração com a *framework* Spring.

Tendo em conta que os serviços e repositórios, que o Spring marca através de anotações, são instanciados ao arranque da aplicação, e tendo em conta que os agentes do JADE são apenas instanciados em *runtime*, é impossível de injetar por dependências os serviços e repositórios nos agentes JADE. No entanto, é possível dos agentes criados em *runtime* acederem aos serviços e repositórios do Spring. Os agentes JADE, quando estão a ser instanciados, permitem passar como argumentos objetos, incluindo implementações de serviços e repositórios. Desta forma, é possível serializar e desserializar estas classes nos agentes JADE.

```
@Autowired
private NotificationRepository notificationRepository;
```

Figura 31 - Injeção de Dependências usando Spring

```
AgentsActionsFactory agentsActionsFactory = new AgentsActionsFactory();
AgentsActionsInterface agentsActionsSystemInterface = agentsActionsFactory.getAgent(AgentTypeEnum.AGENT_SYSTEM);
Object[] argsSystem = {company, notificationRepository};
agentsActionsSystemInterface.createAgent(mc, nameAgent: "systemAgent", argsSystem);
```

Figura 32 - Exemplo de repositório a ser passado como argumento para um agente JADE

Com estas classes internamente nos agentes JADE, é possível interagir com repositórios e serviços Spring.

```
private void createNotificationForProjectCreate(Project project){
    String message = "Project "+project.getProjectName()+" created with success for company "+ project.getCompany().getCompanyName();
    Notification notification = new Notification(message, NotificationEnum.NOTIFICATION_CREATE_PROJECT, project.getCompany());
    notificationRepository.save(notification);
}
```

Figura 33 - Geração de projeto e guardar notificação na respectiva tabela dentro de um agente JADE

4.5 Algoritmo Genético para emparelhamento de empregados e tarefas

Para além das opções de emparelhamento implementadas através de programação linear nos SMA existe ainda uma implementação através de um algoritmo genético (também nos SMA), que o utilizador pode decidir utilizar por configuração. Esta opção encontra-se implementada no *AllocationAgent*, visto tratar-se de mais uma opção com a função de otimização de recursos humanos na execução de tarefas.

A biblioteca Jenetics, utilizada nesta prova de conceito, permite uma alta customização do algoritmo genético desde que os dados do domínio sejam bem-adaptados aos requisitos dos métodos da biblioteca. A adaptação do domínio ao algoritmo genético começou pela definição clara entre o que seria uma população, um cromossoma e um gene. O gene, a unidade mais pequena nestes algoritmos corresponde a um objeto *Pair* que associa um recurso a um projeto dentro de uma empresa. Um cromossoma, que contém uma lista de genes, corresponde, deste modo, ao conjunto de associações de recursos e projetos. Por último, a população é o conjunto de todas as listas de possíveis alocações. A definição de gene e cromossoma na implementação da solução estão presentes nas Figura 34 e Figura 35.

```
public class AllocationGene implements Gene<Pair<Project, Employee>, AllocationGene> {
```

Figura 34 - Definição de Gene usando a biblioteca Jenetics

```
public class AllocationChromosome implements Chromosome<AllocationGene> {
```

Figura 35 - Definição de Cromossoma, usando a biblioteca Jenetics

O algoritmo genético consegue gerar melhores populações através da implementação de uma função de avaliação, adaptada para o domínio da questão. De uma forma geral, esta função de avaliação tenta sugerir que os emparelhamentos de recursos a projetos, onde estes não possuem competências corretas para participar, é algo a evitar assim como a alocação excessiva. Em contrapartida, é benéfico que um recurso seja útil para um dado projeto e isto acontece se este recurso souber executar competências que o projeto necessite. Um dos

pontos de trabalho futuro é a inclusão de mais restrições como, a título de exemplo, as relações entre os colaboradores de um mesmo projeto. A função de avaliação encontra-se na Figura 36.

```
private static double eval(Genotype<AllocationGene> gt) {
    double result = 0.0;
    for(AllocationGene allocationGene: gt.chromosome().stream().toList()){
        Pair<Project, Employee> allocationGenerated = allocationGene.allele();
        Project project = allocationGenerated.getKey();
        Employee employee = allocationGenerated.getValue();
        if(employee != null) {
            List<Skill> skillList = employee.getCandidate().getListSkills();
            List<Skill> skillsProject = project.getNecessarySkills();
            boolean isAnyAllocation = false;
            for (Skill skill : skillList) {
                if (skillsProject.contains(skill)) {
                    isAnyAllocation = true;
                    result = result + 100;
                }
            }
            if (!isAnyAllocation) {
                result = result - 20.0;
            }
        } else {
            result = result - 5.0;
        }
    }
    return result;
}
```

Figura 36 - Função de otimização de cromossomas

Importa, ainda salientar, que o algoritmo genético implementado possui uma probabilidade de mutação de 11.5%. Num contexto produtivo, é possível que esta probabilidade, por via de configuração direta, baixasse para valores que tipicamente rondam os 5%. Para evitar que o algoritmo genético consumisse mais recursos computacionais limitou-se que este gerasse um limite de 1000 populações. Em contextos reais onde uma empresa tem centenas de projetos e milhares de funcionários é possível que este valor limite não seja suficiente.

```
Engine<AllocationGene, Double> engine = Engine
    .builder(AlgorithmAllocationGeneticImpl::eval, genoType)
    .alterers(new Mutator<>(probability: 0.115)).build();
```

Figura 37 – Implementação do algoritmo genético

Os algoritmos genéticos, tipicamente, costumam fornecer melhores soluções o quão melhores forem as populações iniciais fornecidas. No algoritmo implementado, foram dadas

populações com emparelhamentos aleatórios. Uma forma de melhorar os resultados do algoritmo genético é fornecer populações já adequadas através de heurísticas ou modelos preditivos. Uma sugestão que pode ser adotada caso a prova de conceito se torne num produto.

Os resultados extraídos do algoritmo genético são, também eles, registados através de notificações geradas para visualização do Recrutador que interage com o sistema. Um exemplo destas notificações está presente na Figura 38.

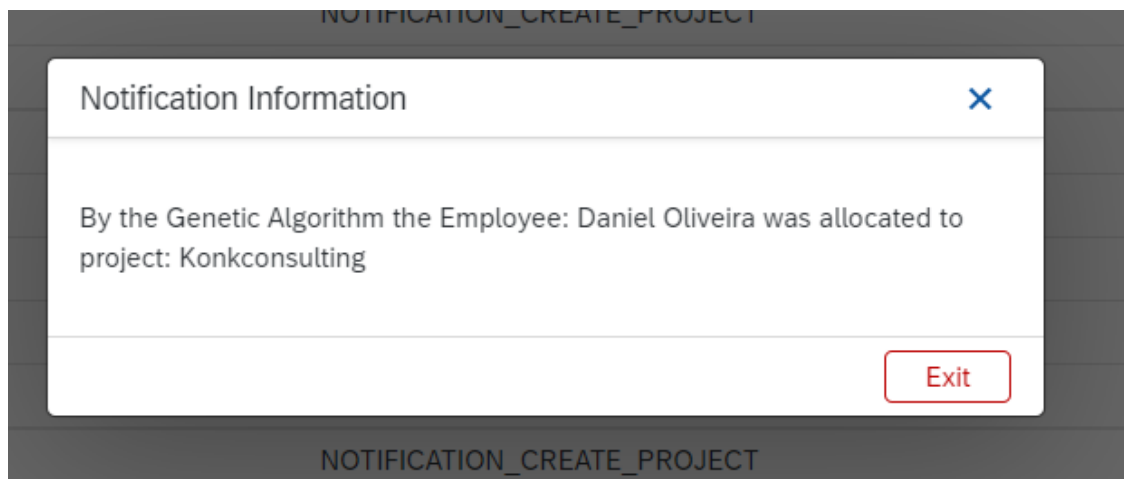


Figura 38 - Notificação de alocação do algoritmo genético

4.6 Bases de Dados

Conforme mencionado previamente, as bases de dados foram desenvolvidas em PostgreSQL e o seu módulo está virtualizado e integrado com Docker.

4.6.1 Scripts Flyway e integração com Spring

De forma a proceder à criação estrutural das tabelas de base de dados e fazer com que estas corram sempre que a aplicação é inicializada, procedeu-se à criação de scripts de Flyway. O Flyway é uma ferramenta de migração que permite fazer alterações ao *schema* de bases de dados alterando assim a sua estrutura (através da criação de tabelas ou *constraints*) e ainda adicionar dados a tabelas criadas.

Na aplicação desenvolvida, os scripts de Flyway são sobretudo utilizados para a construção de tabelas, inserção de dados e para a construção de *constraints* que o domínio da aplicação exige. Todos estes scripts estão presentes no seguinte diretório “resources/migration/”. Todos os scripts têm uma nomenclatura específica exigida pela Flyway, visto correrem de uma forma sequencial onde a ordem é relevante. Na aplicação desenvolvida, os scripts dedicados à criação estrutural das tabelas têm a seguinte nomenclatura “V01.00.00_X__table_Y”, onde X é a ordem pela qual a tabela tem que ser criada, e Y é o nome da tabela em si. Por outro lado,

os scripts dedicados à criação de *constraints* e inserção de dados têm um nome semelhante com uma única troca onde, em vez de, no fim, se chamarem “*table*” chamam-se “*constraints*” e “*data*” respetivamente.

```
CREATE TABLE "postgres"."user_person"(  
  "id" BIGSERIAL UNIQUE NOT NULL,  
  "username" CHARACTER VARYING (1000),  
  "password" CHARACTER VARYING (1000)
```

Figura 39 – Script Flyway da criação de tabela User

Os scripts de Flyway, de forma a comunicarem com a base de dados correta necessitam de ter os detalhes de conexão explicitamente fornecidos. Tal é alcançado através do ficheiro de propriedades da aplicação criado com a aplicação Spring. Na Figura 40, estão presentes os detalhes de comunicação (com exceção da palavra-passe) fornecidos ao Flyway para proceder à comunicação com a base de dados correta.

```
flyway.url=jdbc:postgresql://localhost:5432/postgres  
spring.flyway.schemas=postgres  
flyway.user=postgres
```

Figura 40 - Detalhes de comunicação para os scripts Flyway serem inseridos na respetiva base de dados PostgreSQL

4.6.2 Mapeamento das Entidades de Domínio com as Tabelas de Bases de Dados

O mapeamento e interação entre as classes de domínio e as tabelas de dados, é alcançado através de ORM (*Object-Relational Mapping*) uma técnica que permite a interação com bases de dados relacionais através do paradigma orientado a objetos. A *framework* Spring permite este paradigma através de anotações aplicadas diretamente sobre as classes de domínio e sobre os seus atributos.

```

@Entity
@Table(name = "skill")
public class Skill {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String skillName;

    private String description;

    public Skill(){
        // ORM
    }

    public Skill(String skillName, String description) {
        this.skillName = skillName;
        this.description = description;
    }

    public Long getId() { return this.id; }

    public void setId(Long id) { this.id = id; }

    public String getSkillName() { return skillName; }

    public void setSkillName(String skillName) { this.skillName = skillName; }

    public String getDescription() { return description; }

    public void setDescription(String description) { this.description = description; }
}

```

Figura 41 - Anotações em classe de domínio

As operações realizadas com a base de dados são alcançadas através de interfaces de repositórios, que estendem classes do tipo *CrudRepository* (fornecidas pelo Spring) que são genéricas e permitem efetuar operações básicas no módulo de bases de dados.

```

import com.example.mainapi.project.domain.company.Company;
import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

@Repository
public interface CompanyRepository extends CrudRepository<Company, Long> {
}

```

Figura 42 - Repositórios de Empresas no módulo JAVA

Não obstante as funcionalidades oferecidas, é ainda possível acrescentar interações específicas nestas classes, no entanto, requer a inserção de código SQL que é um dos objetivos que as técnicas de ORM propõem-se a mitigar.

```

@Query(value = "SELECT cd From CompanySAPDetails cd WHERE cd.company.id = :companyID")
public Optional<CompanySAPDetails> findById(
    @Param("companyID") Long companyID
);

```

Figura 43 - Exemplo de funcionalidade acrescentada a interface de repositório

Para conectar o módulo de JAVA ao módulo de bases de dados foi necessário adicionar os detalhes de conexão das bases de dados no ficheiro de propriedades da aplicação.

```
spring.datasource.url=jdbc:postgresql://localhost:5432/postgres
spring.datasource.username=postgres
```

Figura 44 - Detalhes de conexão ao módulo de bases de dados através do módulo JAVA

4.7 Backend

Conforme previamente mencionado, o módulo do *Backend*, foi escrito na linguagem JAVA com auxílio da *framework* Spring, a mais documentada para o desenvolvimento de aplicações Web.

Para além de padrões de desenvolvimento de aplicações que almejam a boa e fácil manutenção das aplicações foram também seguidos padrões de arquitetura de design do código que almejam alcançar os mesmos objetivos. Nesta vertente, destaque para o processo de criação de agentes através de um padrão *factory*. Desta forma, a criação, remoção ou alteração de agentes é feita com o mínimo esforço possível de implementação.

```
/**
 * Agents Actions Factory to turn.
 */
public class AgentsActionsFactory {
    /**
     * Factory Method
     * @param agentTypeEnum
     * @return
     */
    public AgentsActionsInterface getAgent(AgentTypeEnum agentTypeEnum) {
        if(agentTypeEnum.equals(AgentTypeEnum.AGENT_SYSTEM)){
            return new AgentsActionsSystemInterfaceImpl();
        } else if(agentTypeEnum.equals(AgentTypeEnum.AGENT_CANDIDATE)){
            return new AgentsActionsCandidateInterfaceImpl();
        } else if(agentTypeEnum.equals(AgentTypeEnum.AGENT_EMPLOYEE)){
            return new AgentsActionsEmployeeInterfaceImpl();
        } else if(agentTypeEnum.equals(AgentTypeEnum.AGENT_GENERATION_DATA)){
            return new AgentsActionsGenerationDataImpl();
        } else if(agentTypeEnum.equals(AgentTypeEnum.AGENT_ALLOCATIONS)){
            return new AgentActionsAllocationsImpl();
        } else if(agentTypeEnum.equals(AgentTypeEnum.AGENT_INTEGRATION)){
            return new AgentActionsIntegrationImpl();
        }
        return null;
    }
}
```

Figura 45 - Padrão *Factory* para a criação de agentes

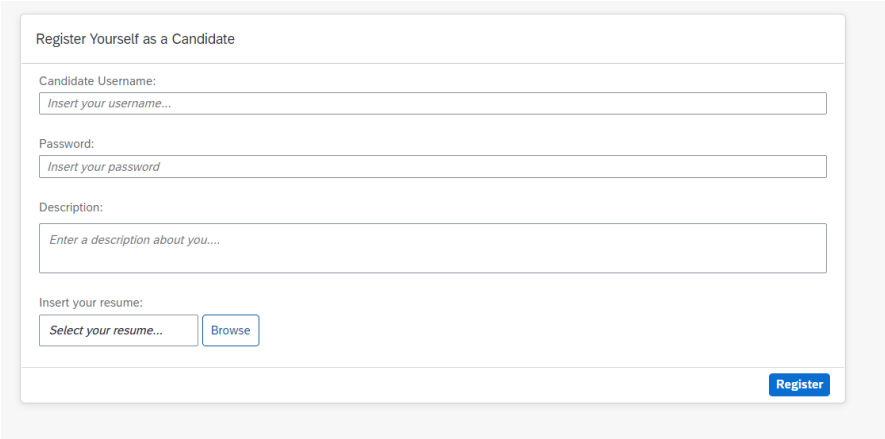
4.8 Frontend

O *frontend* da aplicação, conforme mencionado nas secções anteriores, foi desenvolvido utilizando Angular e a *framework* NGX Fundamentals, visto serem as tecnologias melhor dominadas e que respeitam, também, as diretrizes visuais para aplicações relacionadas com a SAP. O desenvolvimento deste módulo tem o propósito de servir como suporte visual ao projeto, pelo que esta é também a razão da inexistência de testes.

De seguida, estão presentes algumas das páginas desenvolvidas, em particular, as que estão associadas a modelos de IA.

4.8.1. Registo de Candidato

Se o utilizador desejar registar-se como candidato é-lhe apresentada a página da Figura 46.



The image shows a web form titled "Register Yourself as a Candidate". It contains the following elements:

- Candidate Username:** A text input field with the placeholder text "Insert your username...".
- Password:** A text input field with the placeholder text "Insert your password".
- Description:** A larger text area with the placeholder text "Enter a description about you...".
- Insert your resume:** A section containing a file selection button labeled "Select your resume..." and a "Browse" button.
- Register:** A blue button located at the bottom right of the form.

Figura 46 - Página de Registo de Candidato

A página é em tudo semelhante à de registo do Recrutador, sendo que a maior diferença é a existência de mais campos de informação, nomeadamente, a possibilidade de inserção do currículo. Todos os campos são obrigatórios com exceção do campo de descrição.

Após a inserção de todos os dados necessários e do currículo, é apresentada uma página ao utilizador com as competências que os modelos preditivos conseguiram extrair das informações submetidas do utilizador.

AI Inferred Skills

Confirmed predicted Skills:

Degrees

- b- tech

Education

- jawaharlal nehru technological university

Experience

- security analyst

Languages

- French

Skills

- java 2 years
- siem
- mcafee
- esm
- technologies mcafee siem arc sight hp service manager

Apply Cancel

Figura 47- Página de confirmação das competências previstas pelos modelos de inteligência artificial

O utilizador, para finalizar o registo, tem que confirmar as competências previstas pelos modelos. Após a finalização do registo o utilizador retorna à página de login e aparece uma mensagem de sucesso no fundo do ecrã.

4.8.2. Gestão de Empresas do Recrutador

O Recrutador tem a responsabilidade de gerir as empresas que representa no sistema. Isto significa que, é da responsabilidade deste ator, manter atualizado as configurações de conexão ao respetivo sistema SAP. Para além disso, é ainda da responsabilidade deste ator escolher os algoritmos responsáveis pelas associações entre recursos/tarefas e, ainda, a quantidade de variáveis que estes têm em conta. Estas informações são definidas na página da criação da empresa, presente na Figura 48.

Company [X]

Company Name:

Company Description:

SAP MANDT:

SAP User:

SAP User Password:

SAP URL:

Do you want to use our genetic algorithm?

Should allocations algorithms take in consideration project remuneration?

Should allocations algorithms take in consideration work quantity?

[Create](#) [Cancel](#)

Figura 48- Página de criação de empresa

Estes detalhes introduzidos podem, a qualquer momento, ser visualizados e para isso o recrutador necessita de navegar até à página da empresa, presente na Figura 49. Este pode ainda editar os dados introduzidos clicando no botão “Edit” presente na mesma figura.

SAP GuruManager

Company KonkConsulting [Edit](#) [Run Talent Management](#) [Run SAP Integration](#)

Company Information Existing employees Notifications Projects Company Details

Company SAP Connection Details

SAP MANDT
https://sap.rfc:220//

SAP User
GURU_USER

SAP URL
https://sap.rfc:220//

Work Allocations Based On Remuneration

Work Allocations Based On Work Quantity

[Edit Details](#)

Figura 49 – Página de conexões detalhe SAP

4.8.3. Integração dos dados com o sistema SAP

Para realizar uma integração com SAP o utilizador tem que utilizar a página da empresa que pretende integrar. Nesta página, no canto superior direito, existe um botão com uma descrição “Run SAP Integration”. O clique neste botão inicia automaticamente o sistema multiagente que faz uma única integração com SAP de acordo com os dados presentes na empresa. Após o clique, uma mensagem no fundo da página dispara indicando o sucesso da inicialização da operação.

Após a finalização da operação, é guardada uma notificação no projeto relativamente a uma integração bem ou mal sucedida. Esta está presente na página de notificações do projeto, presente na Figura 50.

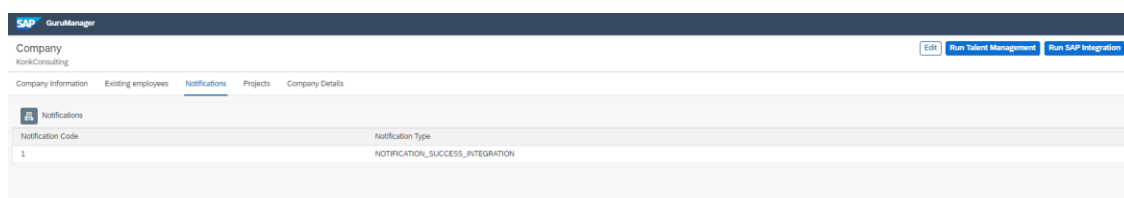


Figura 50 - Página de notificações de projeto

O utilizador pode ainda obter mais informações sobre a operação ao clicar na entrada da notificação pretendida. Esta ação abre uma página com mais informações relativas à integração, presente na Figura 51.

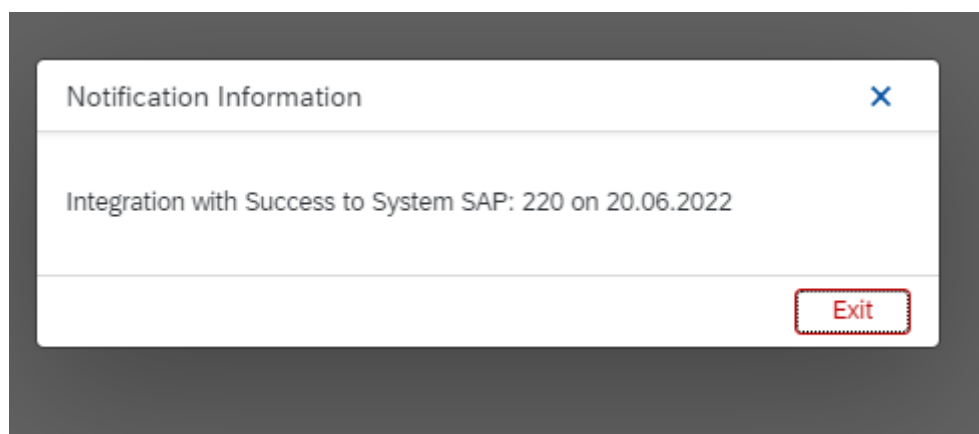


Figura 51 - Integração realizada com sucesso com sistema SAP

4.8.4. Correr o sistema multiagente de alocações automáticas

De forma a correr o sistema multiagente dedicado a alocações automáticas o recrutador tem que aceder à página de projeto e clicar no botão “Run Talent Management”. O pressionar deste botão despoleta o sistema multiagente dedicado ao processo da gestão dos talentos da empresa. Este processo gera, ao longo do tempo, notificações para a tabela de notificações. É, portanto, um processo semelhante ao do processo de integração de dados com o sistema SAP.

4.9 Módulo Python de Inteligência Artificial

As competências preditivas dos modelos e algoritmos de inteligência artificial são expostos para o módulo de *backend* JAVA através de APIs desenvolvidas em Flask, uma *framework* minimalista dedicada ao desenvolvimento de APIs dentro da linguagem Python.

No total foram desenvolvidas duas APIs dentro deste módulo. Uma delas, a que está exposta através de um método GET no URL em “localhost:5000”, tem apenas o propósito de auxiliar os desenvolvedores para perceber o estado operacional do módulo. Se o módulo estiver a correr, uma mensagem é dada como output relativamente ao estado do módulo, no caso do módulo estar desligado, é enviada uma mensagem de erro.

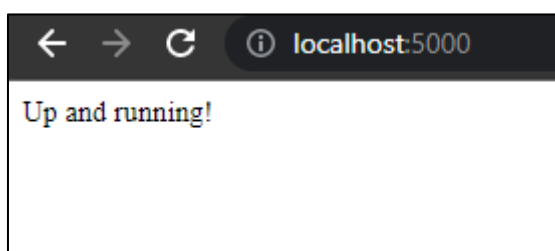


Figura 52- Output de “localhost:5000”

A segunda API, está presente em “localhost:5000/skills/resume” e é através dela que o processo de reconhecimento e inferência de competências dos candidatos após a submissão de um currículo ocorre. O retorno final é enviado para o módulo JAVA sobre a forma de mapa em JSON com as competências extraídas organizadas pelo seu tipo.

```
"Companies": [],  
"Degrees": [  
  "b- tech"  
],  
"Education": [  
  "jawaharlal nehru technological university"  
],  
"Experience": [  
  "security analyst"  
],  
"Gradeyears": [],  
"Languages": [  
  "French"  
],  
"Skills": [  

```

Figura 53 – Resposta JSON das competências reconhecidas a um funcionário usando os modelos preditivos de extração de informação e inferência de competências

5. Testes

Nesta secção são descritos os resultados dos testes para cada um dos módulos do sistema implementados e ainda uma análise de usabilidade às interfaces gráficas desenvolvidas.

5.1 Testes Backend

Os testes unitários implementados relativamente a este módulo foram feitos recorrendo a 2 ferramentas. A primeira, o Junit, é uma ferramenta *open-source* que é dedicada à criação de testes automatizados para a linguagem JAVA, a sua escolha deveu-se à popularidade da ferramenta. A segunda, o Mockito, é uma ferramenta, também *open-source*, que permite simular chamadas a base de dados e simular ainda o retorno provenientes dessas chamadas, deste modo, o teste é direcionado à lógica implementada e não às chamadas às bases de dados que são testadas, por sua vez, através de testes de integração.

No módulo desenvolvido dedicado à lógica de negócio apenas foram testados os packages relativamente aos controladores e aos serviços implementados. Todas as outras classes desenvolvidas não contêm lógica de negócio, mas sim configurações pelo que foram considerados desnecessários testes unitários às mesmas.

5.2 Testes Modelos de IA

Os testes realizados aos modelos desenvolvidos de IA foram implementados através da elaboração de testes de integração. Os testes de integração foram realizados através do Postman, uma plataforma de auxílio ao desenvolvimento e/ou testes de APIs. Esta ferramenta permite a elaboração de coleções que são, na realidade, um conjunto sequencial e lógico de pedidos a APIs

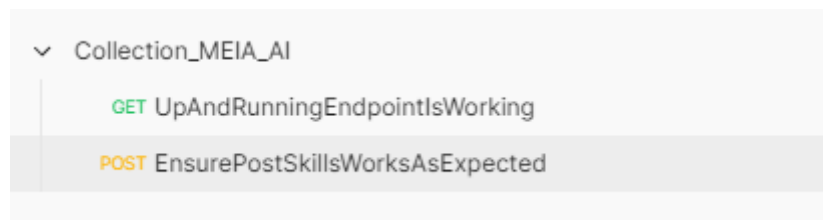


Figura 54 - Exemplo de Coleção Postman

Estas coleções, para além de permitirem testar empiricamente em tempo real os *endpoints* desenvolvidos permitem, ainda, incorporar código Javascript com o objetivo de testar os *endpoints* no que diz respeito ao seu conteúdo e sucesso (entre outros).

```
pm.test("Status code is 200", function () {  
    pm.response.to.have.status(200);  
});
```

Figura 55 - Teste em Postman para testar o tipo de pedido

Na solução, foi implementada uma coleção Postman que testa os dois *endpoints* da aplicação. Um *endpoint* indica se o módulo corre com sucesso e é testado no que diz respeito ao sucesso da resposta e o segundo *endpoint* é testado no que diz respeito ao sucesso da resposta, mas também ao conteúdo (este é o *endpoint* da extração de informação de currículos e inferência de competências). No total foram realizados 10 testes de integração.

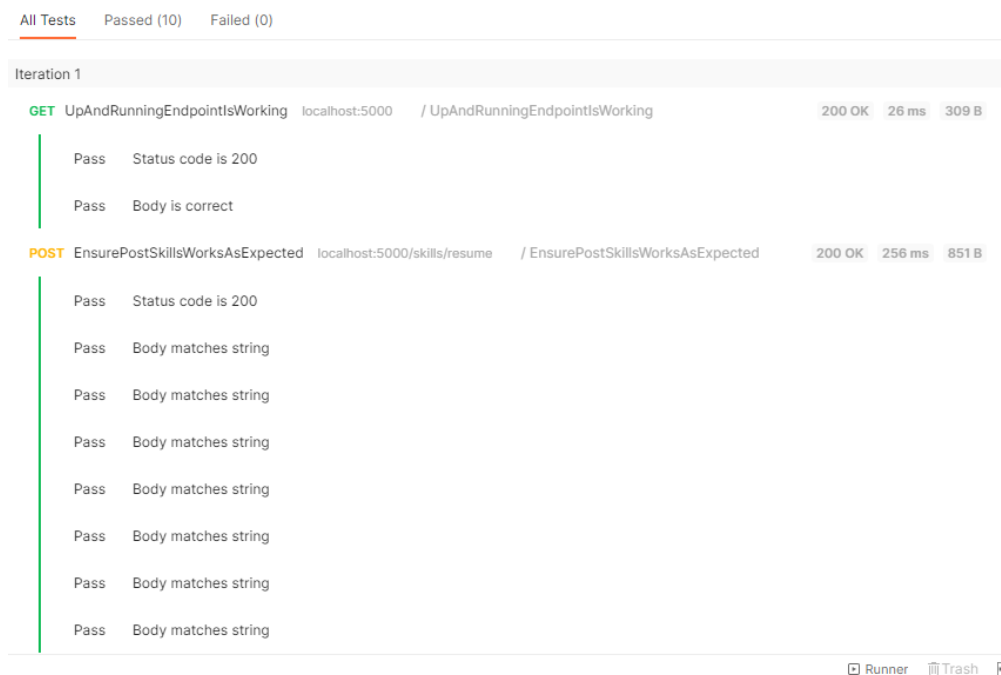


Figura 56 - Resultados dos testes das coleções Postman

5.3 Análise de usabilidade

Nesta secção está descrita a realização de uma análise de usabilidade ao sistema desenvolvido. Esta análise de usabilidade foi conduzida pelo implementador e autor do sistema.

Muitas vezes os sistemas de software pretendem responder a problemas de domínios bastante complexos, no entanto, a usabilidade destes sistemas é, por vezes, descurada, o que pode fazer o sistema fracassar e não ser aceite.

Uma das formas de minimizar estes problemas é através da realização de uma análise de usabilidade. A análise de usabilidade pretende, portanto, testar a qualidade da comunicação entre o sistema e o utilizador, ou seja, pretende inferir a facilidade de interação.

As análises de usabilidade, tipicamente, podem pertencer a dois tipos distintos: tipo formativo, quando o sistema testado se encontra numa fase de protótipo e em desenvolvimento; tipo sumativo, quando o sistema já se encontra na sua fase final de implementação e já é, praticamente, a versão definitiva de lançamento. No presente caso, a análise de usabilidade foi realizada durante o desenvolvimento do sistema pelo que se diz do tipo formativa.

As análises de usabilidade são realizadas durante uma sessão com o inquirido. O inquirido interage com o sistema através da especificação de passos pré-definidos. O realizador do inquérito (nesta situação é o seu desenvolvedor) está presente de forma a prestar o auxílio e para realizar a medição de métricas importantes tais como o tempo de execução dos passos.

No final da sessão, o inquirido responde a um questionário que contém 11 perguntas. As 10 primeiras perguntas seguem um modelo da escala de Likert e contém 5 possíveis respostas que variam entre o “Discordo Plenamente” e o “Concordo Plenamente” com opção de resposta neutra. O conjunto das 10 primeiras perguntas é tipicamente chamado de SUS (*System Usability Scale*). O SUS promove o cálculo de uma pontuação final que pode ser vista como uma classificação geral do sistema relativamente à sua usabilidade. O problema deste tipo de escalas é que, por ter um intervalo de valores de 0 a 100 a sua interpretação por vezes é subjetiva pelo que tipicamente, e acontece na presente análise de usabilidade, os inquiridores acrescentam ao SUS uma pergunta de carácter geral ao sistema para inferir um resultado mais fiável geral. Este tipo de abordagens é válido visto que, conforme vários estudos comprovam, as perguntas de carácter geral têm uma correlação alta com o resultado do SUS.

As perguntas que constaram da análise de usabilidade estão presentes na Tabela 13.

Tabela 13 – Perguntas da Análise de Usabilidade

Chave	Pergunta
P1	Acho que gostaria de frequentar este website frequentemente
P2	Acho este website desnecessariamente complexo
P3	Acho este website fácil de utilizar
P4	Acho que necessitaria de apoio de um profissional para utilizar corretamente este website
P5	Acho que as funcionalidades deste website estavam bem integradas e facilmente acessíveis
P6	Acho que existia demasiada inconsistência no website
P7	Acho que a maioria das pessoas aprenderia a utilizar este website rapidamente

P8	Acho este website muito confuso de se usar
P9	Senti-me confiante a utilizar este website
P10	Precisei de aprender muita coisa antes de utilizar este website
P11	De modo geral, como classificaria a facilidade de interação com este website

As respostas possíveis para as 10 primeiras perguntas estão visíveis na Tabela 14:

Tabela 14 - Respostas para perguntas da *System Usability Scale*

Chave	Resposta
R1	Discordo Plenamente
R2	Discordo
R3	Neutro
R4	Concordo
R5	Concordo Plenamente

As respostas possíveis para a pergunta de carácter geral, encontra-se na Tabela 15:

Tabela 15 - Respostas possíveis ao questionário

Chave	Resposta
R1	Pior imaginável
R2	Péssima
R3	Pobre
R4	Mais ou Menos
R5	Boa
R6	Excelente
R7	Melhor imaginável

As respostas obtidas, talvez devido à homogeneidade do tipo de pessoas inquiridas, foram bastante similares e, embora não excelentes, foram amplamente satisfatórias. A Figura 56 suporta, de uma forma visual, esta conclusão.

Pergunta 11

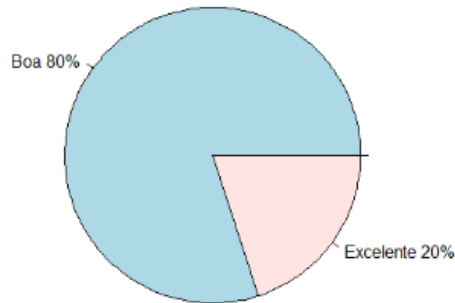


Figura 56 - Respostas Pergunta 11

O SUS, conforme dito anteriormente, pressupõe o cálculo de um valor final que corresponde à satisfação geral da utilização do sistema e relaciona-se com a pergunta com a chave P11.

O cálculo do SUS é dado através da seguinte fórmula:

- Para cada questão em que o índice da pergunta é ímpar, subtrai-se um valor da pontuação.
- Para cada questão em que o índice da pergunta é par, subtrair a pontuação a 5.
- Somar tudo e multiplicar o resultado por 2.5

Os valores obtidos abaixo, representado na seguinte tabela através das medianas das respostas, indicam que, o SUS obtido foi satisfatório o que condiz com a pergunta P11 realizada e ainda com o gráfico da Figura 56.

Tabela 16 – Valores obtidos do questionário

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	SUS TOTAL
4	2	4	2	4	2	4	2	4	2	70

6. Conclusão

O capítulo da conclusão está dividido em 3 subsecções distintas. A primeira subsecção é relativa ao progresso do desenvolvimento dos objetivos propostos, aqui foi elaborada uma tabela para sumarizar os objetivos que foram concluídos, os que ainda estão por concluir ou a serem concluídos. A segunda subsecção é dedicada a enumerar as limitações da solução proposta, ou ainda a enumerar possível trabalho futuro. Por último, na terceira secção é dada uma apreciação pessoal que finaliza a secção das conclusões.

6.1 Conclusão

Recentemente, o mercado da empregabilidade está a evoluir, catalisado por eventos como a revolução tecnológica e a pandemia do COVID-19. Esta evolução e instabilidade levam a que as empresas visualizem de forma diferente os seus recursos humanos. As empresas não estão interessadas na tradicional relação empregador-empregado, mas em ter ao seu dispor as competências que possam satisfazer as necessidades produtivas de uma empresa. Esta abordagem levou à invenção e implementação de plataformas online apelidadas de “mercados de talento” que se comprometem à manutenção das competências dentro de empresas através da aquisição de novos recursos, mas, sobretudo, através da retenção dos recursos já existentes.

A revisão literária, de um conjunto inicial de 27361 artigos excluiu, numa primeira fase, 23329 através de leituras rápidas do título ou resumos. Dos restantes 4032, 3988 não respeitaram os critérios de controlo qualidade, quer seja os critérios de inclusão ou exclusão. No total, foram escolhidos e analisados 44 artigos no contexto de 4 questões de investigação.

Para a primeira questão de investigação “Quais são as principais técnicas de inteligência artificial utilizadas para a extração automatizada de informação de currículos” foram encontrados vários artigos, tendo sido selecionados 12. Foram apresentadas técnicas de variados tipos, no entanto, destaque, para as redes neuronais, de arquiteturas de aprendizagem profunda, que atingiram os melhores resultados.

A segunda questão de investigação “Quais são as principais técnicas de inteligência artificial utilizadas para a inferência de competências”, foi respondida tendo em consideração 15 artigos. Dos artigos selecionados foi possível entender que várias técnicas de inteligência artificial são usadas para o efeito, no entanto, destacam-se as redes neuronais que apresentam os melhores resultados e apareceram em 5 dos 15 artigos escolhidos. Importa, no entanto, salientar que, nesta situação, o melhor modelo depende do tipo de informação que se tenta inferir.

No contexto da terceira questão de investigação “Existe algum sistema multiagente usado para emparelhar empregados com empregos/tarefas?” foram selecionados 11 artigos. Numa

fase inicial foram selecionados 9 artigos, que comprovam a viabilidade de utilizar sistemas multiagente para o efeito. Numa fase posterior, após sessões de feedback com a orientadora, foram adicionados 2 artigos, com foco no desporto robótico, já que se assemelham ao problema de emparelhamento levantado.

Por último, na quarta questão de investigação “Existe algum algoritmo genético usado para emparelhar empregados com empregos/tarefas” foram selecionados 6 artigos. Estes mostraram que esta é uma abordagem válida.

Todos os artigos encontrados nesta revisão literária foram importantes, para a elaboração de uma análise e design com o intuito do desenvolvimento de uma prova de conceito que se assemelha a um mercado de talento.

A fase da implementação teve uma duração de 4 meses e, apesar de várias limitações, chegou-se a uma solução que cumpre os objetivos propostos e está pronta para fazer parte do portefólio de soluções da empresa KonkConsulting. As conclusões satisfatórias alcançadas são corroboradas pelos resultados dos testes realizados e pela satisfação obtida na análise de usabilidade.

6.2 Objetivos concretizados

Os estados da concretização dos objetivos propostos para esta dissertação encontram-se na Tabela 18 , utilizando a notação presente na Tabela 17.

Tabela 17 - Notação utilizada para descrever estado das tarefas da dissertação










Terminado	
A terminar	
Por desenvolver	

Tabela 18 - Estado da dissertação - tarefas

Contextualização do Problema	
Revisão Literária	

Desenho da Solução	
Implementação da Solução	
Testes para a solução	
Análise de usabilidade da solução	
Conclusão da escrita do relatório	

Através dos estados dos objetivos propostos, podemos concluir que a este nível, a dissertação foi um sucesso.

6.3 Limitações e Trabalho Futuro

Embora se considere que todos os objetivos da presente dissertação foram atingidos, a verdade é que poderiam ter sido alcançados melhores resultados dadas melhores condições logísticas e de hardware.

Em primeiro lugar, as métricas de performance da rede neuronal poderiam ser melhor se se usasse um melhor GPU. Para além disso, apesar de se considerar que os resultados foram excelentes, dado o tempo disponibilizado para a implementação, na realidade, a solução proposta está distante dos resultados e confiabilidade de soluções já existentes no mercado como, por exemplo, o Spacy. Isto é uma crítica e uma limitação do sistema.

Para além disso, no que diz respeito ao sistema multiagente, embora a solução desenvolvida já tenha em consideração vários cenários, e apesar de já ser possível tornar o algoritmo de escalonamento mais ou menos complexo, consoante as opções que o utilizador escolha, podiam ter sido tidas em conta mais variáveis, tais como: relações entre colaboradores, preferências de horário, entre outros... Embora estas variáveis não tenham sido incluídas nesta prova de conceito, no futuro, face à natureza arquitetural dos sistemas multiagente, que privilegiam a extensibilidade, é possível com poucos recursos e esforço acrescentar mais variáveis e tornar a solução ainda mais flexível e fiável.

Outro dos pontos a melhorar no futuro está relacionado com a inferência de competências dos colaboradores. Tendo em conta que o projeto desenvolvido é entendido como sendo uma prova de conceito, apenas foi implementada a inferência das línguas que um recurso fala com base nas localizações identificadas. Ora, é possível inferir ainda outras competências como,

por exemplo, a experiência de colaboradores dependendo se trabalhou num departamento durante um largo período temporal. No fundo, a crítica é que poderiam ter sido inferidas mais competências, havendo mais tempo para a implementação do projeto.

Ainda na secção do trabalho futuro, a autenticação e autorização da aplicação foram implementadas com recurso ao Spring Security. Os algoritmos utilizados na encriptação, foram de baixa complexidade o que põe em causa a segurança da aplicação. Caso esta prova de conceito seja transformada num produto final teriam de ser repensadas as questões relativas à segurança.

Por último, os testes cobrem, neste momento, todas as funcionalidades principais do sistema e isto já se traduz numa cobertura interessante, no entanto, esta poderia ter valores ainda melhores caso o período para implementação fosse ainda mais longo.

6.4 Apreciação Final

O desenvolvimento desta dissertação de mestrado foi, pessoalmente, um momento bastante importante para mim. A resolução individual do problema que me foi apresentado, obrigou-me a ganhar autonomia e a trabalhar em algumas componentes de inteligência artificial que até então não tivera a oportunidade de trabalhar na prática. Para além disso, permitiu-me adquirir a capacidade de desenvolver um sistema de raiz, direccionado a uma problemática que me foi apresentada e que, sem dúvida, é de importância tendo em conta o contexto económico e laboral mundial dos dias de hoje. É com enorme satisfação que me revejo no trabalho que apresento e estou expectante para perceber de que forma é que este projeto vai evoluir.

Por último, gostaria de salientar toda a estrutura do Departamento de Engenharia Informática e, em particular, os intervenientes do Mestrado de Engenharia da Inteligência Artificial. Foram 2 anos de muito trabalho e esforço, porém todas as qualidades intelectuais, humanísticas e emocionais oferecidas por parte dos docentes tornaram este percurso muito positivo e, sem dúvida, memorável.

7. Referências

- [1] V. Team, "Valamis.com," Valamis, 21 4 2021. [Online]. Available: <https://www.valamis.com/hub/talent-management>. [Acedido em 16 12 2021].
- [2] G. Team, "Gartner Glossary," Gartner, [Online]. Available: <https://www.gartner.com/en/human-resources/glossary/talent-management>. [Acedido em 16 12 2021].
- [3] U.S Department of Labor, "EMPLOYEE TENURE IN 2020," 22 9 2020. [Online]. Available: <https://www.bls.gov/news.release/pdf/tenure.pdf>. [Acedido em 16 12 2021].
- [4] L. Andre, "Finances Online," 2021. [Online]. Available: <https://financesonline.com/employee-turnover-statistics/>. [Acedido em 16 12 2021].
- [5] H. Silva e H. Shalovinsky, "Skills - The new currency," IBM, 25 05 2021. [Online]. Available: <https://www.ibm.com/blogs/think/be-en/2021/05/25/skills-the-new-currency/>. [Acedido em 17 12 2021].
- [6] J. Manyika, S. Lund, M. Chui, J. Bughin, J. Woetzel, P. Batra, R. Ko e S. Sanghvi, "Jobs lost, jobs gained: What the future of work will mean for jobs, skills, and wages," McKinsey Global Institute, 28 11 2017. [Online]. Available: <https://www.mckinsey.com/featured-insights/future-of-work/jobs-lost-jobs-gained-what-the-future-of-work-will-mean-for-jobs-skills-and-wages>. [Acedido em 17 12 2021].
- [7] D. Blake, "Why skills are the currency of the future of work," NBusiness, 19 4 2019. [Online]. Available: <https://www.thenationalnews.com/business/why-skills-are-the-currency-of-the-future-of-work-1.849310>. [Acedido em 18 12 2021].
- [8] K. Bravery e L. Lyons, "SKILLS IS THE NEW CURRENCY FOR WORKFORCE TRANSFORMATION," The HR Director, 12 4 2021. [Online]. Available: <https://www.thehrdirector.com/business-news/skills/leading-companies-use-skills-as-the-new-currency-for-workforce-transformation/>. [Acedido em 18 12 2021].
- [9] A. Etziom, "The Talent Marketplace, Explained," Gloat, 6 12 2021. [Online]. Available: <https://gloat.com/blog/the-talent-marketplace-explained/>. [Acedido em 18 12 2021].
- [10] J. Bersin, "Talent Marketplace Platforms Explode Into View," Josh Bersin, 3 7 2020. [Online]. Available: <https://joshbersin.com/2020/07/talent-marketplace-platforms-explode-into-view/>. [Acedido em 18 12 2021].

- [11] N. Skytland, "A Marketplace for Talent," NASA, 17 10 2019. [Online]. Available: <https://blogs.nasa.gov/futureofwork/2019/10/17/a-marketplace-for-talent/>. [Acedido em 18 12 2021].
- [12] T. Horda, "Talent marketplace: Embracing agile talent mobility," HR Forecast, 7 4 2021. [Online]. Available: <https://hrforecast.com/talent-marketplace-embracing-agile-talent-mobility/>. [Acedido em 16 12 2021].
- [13] G. Team, "Gloat," Gloat, [Online]. Available: <https://gloat.com/>. [Acedido em 16 12 2021].
- [14] T. Team, "TalentSoft," TalentSoft, [Online]. Available: <https://www.talentsoft.com/>. [Acedido em 16 12 2021].
- [15] 3. Team, "356Talents," 356Talents, [Online]. Available: <https://365talents.com/en/>. [Acedido em 18 12 2021].
- [16] J. Mccarthy, "WHAT IS ARTIFICIAL INTELLIGENCE?," California, 2007.
- [17] T. N. Y. Team, "Sunday Reading: The Rise of Artificial Intelligence," New Yorker. [Online].
- [18] R. Brown, "Where is Artificial Intelligence Used Today?," 4 12 2019. [Online]. Available: <https://becominghuman.ai/where-is-artificial-intelligence-used-today-3fd076d15b68>. [Acedido em 16 12 2021].
- [19] B. Dutta, "What is the Role of AI in Human Resource Management?," 16 10 2021. [Online]. Available: <https://www.analyticssteps.com/blogs/what-role-ai-human-resource-management>. [Acedido em 18 12 2021].
- [20] T. Bernard, T. Moreau, C. Viricel, P. Mougel, C. Gravier e F. Laforest, "Learning Joint Job Embeddings using a Job-Oriented Asymmetrical Pairing System," em *above, jobs are evolving quickly, new jobs including new descrip*24th European Conference on Artificial Intelligence - ECAI 2020, Santiago de Compostela, 2020.
- [21] R. Piper, "How to write a systematic literature review: a guide for medical studentes," Edinburgh, 2013.
- [22] I. Wings, R. Nanda e K. Adebayo, "A Context-Aware Approach for Extracting Hard and Soft Skills," em *10th International Young Scientists Conference on Computational Science*, 2021.
- [23] P. Roy, S. Chowdhary e R. Bhatia, "A Machine Learning approach for automation of Resume Recommendation system," em *International Conference on Computational Intelligence and Data Science (ICCIDS 2019)*, 2020.

- [24] H. Ma, Y. Xu, W. Ma, Z. Lin e K. Jiang, "A Muti-Field Feature Interaction Convolutional Neural Network for Resume Recommendation," em *2020 International Symposium on Autonomous Systems (ISAS) 978-0-7381-1262-6/20/\$31.00 ©2020 IEEE 186*, 2020.
- [25] Y. Deng, H. Lei, X. Li e Y. Lin, "An Improved Deep Neural Network Model for Job Matching," em *2018 International Conference on Artificial Intelligence and Big Data*, 2018.
- [26] A. S. C e C. Raseek, "Combination of Neural Networks and Conditional Random Fields for Efficient Resume Parsing," em *2018 International CET Conference on Control, Communication, and Computing (IC4) | July 05 – 07, 2018 | Trivandrum 978-1-5386-4966-4/18/\$31.00 ©2018 IEEE 388*, 2018.
- [27] N. Tulumen, G. Akgun, A. Nohutcu, G. Genç e S. Genç, "Hybrid Job and Resume Matcher," em *2021 6th International Conference on Computer Science and Engineering*, 2021.
- [28] Y. Sun, F. Zhuang, H. Zhu, Q. Zhang, Q. He e H. Xiong, "Market-oriented job skill valuation with cooperative composition neural network," 2021.
- [29] B. Gaur, G. Saluja, H. Sivakumar e S. Singh, "Semi-supervised deep learning based named entity recognition model to parse education section of resumes".
- [30] G. B, S. Mandal e R. V, "Automatic Extraction of Segments from Resumes using Machine Learning," em *2020 IEEE 17th India Council International Conference*, 2020.
- [31] T. K, U. V, S. Kadiwal e S. Revanna, "Design and Development of Machine Learning based Resume Ranking System," *Global Transitions Proceedings*, 2021.
- [32] R. Valdez-Almada, O. Rodriguez-Elias, C. Rose-Gómez e M. Velázquez-Mendoza, "Natural Language Processing and Text Mining to Identify Knowledge Profiles for Software Engineering Positions," em *2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT)*, 2017.
- [33] M. Tikhonova e A. Gavrishchuk, "NLP methods for automatic candidate's CV segmentation," em *2019 International Conference on Engineering and Telecommunication (EnT)*, 2019.
- [34] G. Marvin, M. Jackson e G. Alam, "A Machine Learning Approach for Employee Retention Prediction," 2021.
- [35] J. Reddy, S. Regella e S. Seelam, "Recruitment Prediction using Machine Learning," em *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, 2020.

- [36] M. Abouof, S. Singh, H. Otrok, R. Mizouni e E. Damiani, *Machine Learning in Mobile Crowd Sourcing: A Behavior-based Recruitment Model*, 2021.
- [37] H. Ahmad, M. Asghar, M. Zubair, A. Khan e A. Mosavi, "A Hybrid Deep Learning Technique for Personality Trait Classification From Text," *IEEE Access*, 2021.
- [38] T. Huynh, K. Nguyen, N. Nguyen e A. Nguyen, "Job Prediction: From Deep Neural Network Models to Applications," em *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, Vietnam, 2020.
- [39] P. Mwaro, K. Ogada e W. Cheruwyot, "Neural Network Model for Talent Recruitment and Management for Employee Development and Retention," em *2021 IEEE AFRICON*, 2021.
- [40] K. Jiechieu e N. Tsopze, "Skills prediction based on multi-label resume classification using CNN with model predictions explanation," em *Neural Computing and Applications*, 2021, pp. 5069-5087.
- [41] K. Varshney, V. Chenthamarakshan, S. Fancher, J. Wang, D. Fang e A. Mojsilovic', "Predicting Employee Expertise for Talent Management in the Enterprise," em *KDD '14: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014.
- [42] A. Apatean, E. Szakacs e M. Tilca, "MACHINE-LEARNING BASED APPLICATION FOR STAFF RECRUITING," 2017.
- [43] E. Faliagka, K. Ramantas, A. Tsakalidis e G. Tzimas, "Application of Machine Learning Algorithms to an online Recruitment System," em *ICIW 2012 : The Seventh International Conference on Internet and Web Applications and Services*, 2012.
- [44] S. Sarangarajan, K. Chitra e S. S, "Automation of Competency & Training Management using Machine Learning Models," India, 2021.
- [45] D. Shulcker, N. Lim , L. Matthews, G. Grimm, A. Lawrence e P. Firoz, "Can Artificial Intelligence Improve Air Force Talent Management," em *Can Artificial Intelligence Improve Air Force Talent Management*, RAND, 2021.
- [46] G. Velakanti e A. Mathur, "Machine learning approach to find the abilities in a candidate for steady employment in engineering field: A literature survey," *Elsevier*, 2020.
- [47] O. Salazar, J. Jaramillo, D. Ovalle e J. Guzmán, "A case-based multi-agent and recommendation," Medellín, 2015.
- [48] P. Krafft, "A Simple Computational Theory of General Collective," em *Topics in Cognitive Science*, Massachusetts, 2017, pp. 374-392.

- [49] H. Li, L.-Y. Hao, X. Ge, J. Gao e S. Guo, An agent-based approach for crowdsourcing software design, Dalian, 2016.
- [50] T. Mine, T. Kakuta e A. Ono, "Reciprocal Recommendation for Job Matching with Bidirectional Feedback," em *2013 Second IIAI International Conference on Advanced Applied Informatics*, Fukuoka, 2013.
- [51] R. Shang, Y. Ma, F. Ali, C. Hu, S. Nazir, H. Wei e A. Khan, "Selection of Crowd in Crowdsourcing for Smart Intelligent Applications: A Systematic Mapping Study," *Hindawi Scientifical Programming*, 2021.
- [52] W. Wang, J. Jiang, B. An, Y. Jiang e B. Chen, "Toward Efficient Team Formation for Crowdsourcing in Noncooperative Social Networks," em *TRANSACTIONS ON CYBERNETICS*, 2017.
- [53] J. Jiang, A. Bo, Y. Jiang, D. Lin, Z. Bu, J. Cao e Z. Hao, "Understanding Crowdsourcing Systems from a Multiagent Perspective and Approach," 2018.
- [54] E. Azofeifa e G. Novikova, "Verification of a labor market domain using an academic crowdsourcing system," *Computer Science and Computer Engineering*, 2020.
- [55] T. Szapiro e P. Szufel, "Simulated Negotiation Outcomes Through Recommendation Crowding," 2013.
- [56] H. Akiyama, "Acquiring Multiagent Cooperative Behavior in the RoboCup Soccer Simulation," em *2019 IEEE 11th International Workshop on Computational Intelligence and Applications*, Hiroshima, 2019.
- [57] D. Iarosh, G. Reneva, A. Kornilova e P. Konovalov, "Multiagent System of Mobile Robots for Robotic Football," em *2019 26th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, Saint Petersburg, 2019.
- [58] A. Apornak, A. Keramati, S. Raissi e K.-D. Kaveh, "Human resources optimization in hospital emergency using the genetic algorithm approach," *International Journal of Healthcare Management*, May 2020.
- [59] Y. Li e L. Wang, "A Genetic Algorithm Model for Human Resource Management Optimization in the Internet Marketing Era," *Mathematical Problems in Engineering*, 2022.
- [60] A. Starkey, G. Owusu, H. Hagrass e R. Chimatapu, "Stacked Auto Encoder Based Hybrid Genetic Algorithm for Workforce Optimization," United Kingdom, 2018.
- [61] R. Nand, A. Sharma e K. Reddy, "Skill-Based Group Allocation of Students for ProjectBased Learning Courses Using Genetic Algorithm: Weighted Penalty Model," em *2018 IEEE*

International Conference on Teaching, Assessment, and Learning for Engineering (TALE), Australia, Wollongong, 2018.

- [62] N. Xue, D. Landa-Silva e I. Triguero, "A Genetic Algorithm With Composite Chromosome for Shift Assignment of Part-time Employees," 2018.
- [63] J. Cai, Z. Peng, S. Ding e J. Sun, "A Robust Genetic Algorithm to Solve MulSkill Resource Constrained Project Scheduling Problem with Transfer Time and Uncertainty Skills," em *2020 IEEE 16th International Conference on Control & Automation (ICCA)*, Japão, Hokkaido, 2020.
- [64] U. European, "A proteção de dados ao abrigo do RGPD".
- [65] Z. Cong, X. Zhang, H. Wang e H. Xu, "Human resource recommendation algorithm based on ensemble learning and Spark," em *P Conf. Series: Journal of Physics: Conf. Series 887*, 2017.
- [66] T. N. Y. Team, "Sunday Reading: The Rise of Artificial Intelligence," *New Yorker*, 13 10 2019. [Online]. Available: <https://www.newyorker.com/books/double-take/sunday-reading-the-rise-of-artificial-intelligence>. [Acedido em 16 12 2021].
- [67] P. Souza, "Absolute answerability in the Era of Artificial Intelligence and Machine Learning: A talent management perspective," em *2019 International Conference on Digitization (ICD)*, 2019.
- [68] R. Piper, "How to write a systematic literature review: a guide for medical students," University of Edinburgh , Edinburgh, 2013.

Anexo A – Questionários de Usabilidade

6/19/22, 9:26 PM

Questionário de satisfação da aplicação web desenvolvida da gestão de talentos - GuruManager

Questionário de satisfação da aplicação web desenvolvida da gestão de talentos - GuruManager

Foi convidado a participar num estudo sobre a análise de usabilidade sobre uma aplicação web relativa á gestão de talentos e onboardings numa empresa. O estudo está a ser realizado por Daniel Oliveira no âmbito da unidade curricular de Projeto/Dissertação/Estágio (PROJIA) do Mestrado de Engenharia de Inteligência Artificial do Instituto Superior de Engenharia do Porto em parceria com a KonkConsulting, uma consultoria informática dedicada ao desenvolvimento de software ligado á gestão de recursos humanos.

*Obrigatório

1. Ao clicar "Concordo" está (1) a aceitar participar no estudo, onde lhe vai ser pedido que complete um questionário seguindo os passos do guia de sessão fornecido e que (2) autoriza a utilização dos dados preenchidos para o âmbito descrito acima. *

Marcar tudo o que for aplicável.

Concordo com os termos de utilização.

Questionário sobre a aplicação desenvolvida

2. Para cada linha, indique o seu grau de concordância. Realize primeiro os passos presentes no guia de sessão disponibilizado *

Marcar apenas uma oval por linha.

	Discordo Plenamente	Discordo	Neutro	Concordo	Concordo Plenamente
Acho que gostaria de utilizar este website frequentemente.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acho este website desnecessariamente complexo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acho este website fácil de usar.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acho que necessitaria de apoio de um técnico para utilizar este website.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acho que as várias funcionalidades deste website estavam bem integradas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acho que existia demasiada inconsistência neste website.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acho que a maioria das pessoas aprenderia a utilizar este website rapidamente.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Acho este website muito confuso de se usar.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Senti-me muito	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

confiante ao utilizar
este website.

Precisei de aprender
muita coisa antes de
começar a utilizar
este website.

3. De modo geral, como classificaria a facilidade de utilização da aplicação desenvolvida? *

Marcar apenas uma oval.

- Pior imaginável
 Péssima
 Pobre
 Mais ou menos
 Boa
 Excelente
 Melhor Imaginável
-

Este conteúdo não foi criado nem aprovado pela Google.

Google Formulários

Anexo B – Guia de Sessão para Análise de Usabilidade

Guia da Sessão

Sessão para a avaliação da usabilidade da aplicação desenvolvida para a gestão de talentos numa empresa

Obrigado por participar. O objetivo desta sessão é interagir com um sistema de gestão de talentos de uma empresa que foi desenvolvido durante a unidade curricular de Projeto/Estágio/Dissertação do Mestrado em Engenharia da Inteligência Artificial no Instituto Superior de Engenharia do Porto em parceria com a Konkconsulting, uma consultoria informática situada no Porto. Para a realização desta análise de usabilidade será alvo de um teste de eficiência que consiste na execução e cronometração de passos pré-definidos que se encontram na página seguinte.

Após a execução destes passos, será alvo de um questionário baseado na interação com o sistema.

De uma forma resumida, a análise de usabilidade pode ser resumida nos seguintes passos:

- Introdução e descrição da sessão
- Teste de eficiência seguindo passos pré-definidos
- Questionário relativo à interação com o sistema

Não existem respostas certas ou erradas. O alvo da avaliação é o website desenvolvido e não o utilizador em si pelo que deve tomar o tempo que desejar para realizar os passos necessários e responder de forma honesta ao questionário. Qualquer comentário adicional é também bem-vindo.

A sessão será conduzida por Daniel Oliveira, que auxiliará para responder a qualquer dúvida.

Hiperligação para o questionário:

<https://docs.google.com/forms/d/1j5Bg9ksp9-WcCExTSlwBig31poboFjpM5TyE1gTvkiw/edit>

Passos a seguir

Para começar a análise de usabilidade, abra o hyperlink fornecido na página anterior e a página da aplicação e realize os seguintes passos por ordem:

1. Acesse ao Menu de Login.
2. Acesse ao Menu de Registro
3. Registre-se como um Recrutador
4. Registre um outro utilizador como um candidato.
5. Acesse ao sistema como um recrutador.
6. Acesse ao Menu de gestão de empresas
7. Crie uma empresa
8. Acesse à empresa para visualizar os seus detalhes.
9. Edite as informações de uma empresa.
10. Acesse ao menu de uma empresa para visualizar os detalhes alterados.
11. Crie um projeto.
12. Acesse à página de projeto para adicionar uma capacidade necessária para a execução de um projeto.
13. Acesse ao menu de candidatos disponíveis para contratar.
14. Visualize uns detalhes de um candidato e contrate-o para a sua empresa.
15. Corra a integração com SAP para uma empresa.
16. Corra os algoritmos de alocação automáticas de funcionários a uma empresa.
17. Visualize se o candidato que contratou foi alocado dentro do menu de detalhes de um projeto.
18. Aloque o candidato manualmente a um projeto