



Profile Matcher

JOSÉ NUNO FREITAS CARDOSO

Outubro de 2020





Profile Matcher

José Nuno Freitas Cardoso

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

Orientador ISEP: Nuno Filipe Fonseca Vasconcelos

Escudeiro

Profile Matcher

Agradecimentos

Quero agradecer a todos que tornaram possível a realização desta tese.

Agradeço ao ISEP, pelo conhecimento e boas práticas adquiridas ao longo da licenciatura e do mestrado.

Agradeço também ao docente Nuno Escudeiro pela orientação e pelos conselhos dados, que certamente contribuíram positivamente para o desenvolvimento desta tese.

Por fim, agradeço à minha família e amigos, pelo apoio dado ao longo do meu percurso académico.

Resumo

A seleção de candidatos a um emprego é na sua essência um processo de ajustamento de um currículo a um dado perfil descrito pela oferta de emprego. É um processo demoroso que frequentemente exige a análise e avaliação individual e manual de centenas de currículos. Este mesmo cenário, de ajustamento de competências a um dado perfil profissional, verifica-se também, embora em circunstâncias substancialmente distintas, quando um estudante pretende definir o seu percurso académico direcionando-o para uma determinada profissão.

A utilização de ferramentas automáticas pode trazer benefícios, como maior celeridade e eficiência, a estes processos. Este artigo retrata o trabalho desenvolvido na implementação da aplicação Profile Matcher. Uma aplicação Web que permite criar *templates* de emprego ou ocupação profissional e compará-los automaticamente com um conjunto de currículos. Tem como fim identificar os melhores candidatos para uma oferta de emprego ou as melhores ofertas de emprego para um dado currículo fornecendo indicações adicionais sobre as competências que um dado estudante deve desenvolver para um melhor posicionamento face a uma dada profissão. Espera-se que esta aplicação permita guiar o percurso académico de um estudante assim como diminuir o tempo dos processos de recrutamento.

Palavras-chave: Ferramentas automáticas, Seleção de candidatos, Estudantes, Aprendizagem automática, Rede neuronal, Doc2Vec

Abstract

The selection of candidates for a job is essentially a process of adjusting a resume to a given profile described by the job offer. It is a lengthy process that often requires the analysis and evaluation of hundreds of resumes individually and manually. This same scenario, of skills adjustment to a given professional profile, also occurs, although in substantially different circumstances, when a student wants to define his or her academic path by directing him or her to a given profession.

The use of automatic tools can bring benefits, such as greater speed and efficiency, to these processes. This article portrays the work done in developing the Profile Matcher application. A web application that allows you to create job or occupation templates and automatically compare them with a set of resumes. Its goal is to identify the best candidates for a job offer or the best job offers for a given resume by providing additional indications on the skills that a given student should develop for a better position in a given profession. This application is expected to guide a student's academic path as well as decrease the time of recruitment processes.

Keywords: Automatic tools, Candidate selection, Students, Machine learning, Neural Network, Doc2Vec

Índice

1	<i>Introdução</i>	xvii
1.1	Contexto	xvii
1.2	Problema	xvii
1.3	Objetivo do projeto	xix
1.3.1	Recrutadores	xix
1.3.2	Candidatos a emprego	xx
1.4	Resultados esperados	xx
1.5	Organização do relatório	xx
2	<i>Estado de arte e Análise de valor</i>	23
2.1	Estado de arte	23
2.1.1	Soluções existentes	23
2.1.2	Desenvolvimento de ferramentas automáticas	25
2.1.3	Bibliotecas de Aprendizagem automática	27
2.2	Análise de valor	28
2.2.1	New Concept Development Model	28
2.2.2	Valor	30
2.2.3	Proposta de valor	31
2.2.4	Modelo Canvas	32
2.2.5	Modelos de análise de valor	33
3	<i>Avaliar soluções existentes</i>	37
3.1	Análise das ferramentas	37
3.1.1	Desempenho	37
3.1.2	Grafos	38
3.1.3	Facilidade de aprendizagem	38
3.1.4	Visualização gráfica	38
3.1.5	Debugging	38
3.1.6	Serialização/Desserialização	39
3.2	Decisão	39

4	Análise e Desenho	45
4.1	Casos de uso	46
4.1.1	Requisitos funcionais	46
4.1.2	Interação entre utilizador e sistema	48
4.2	Modelo de domínio	55
4.3	Modelo de dados	58
4.4	Arquitetura de software	58
5	Implementação	61
5.1	Aplicação para Matching	61
5.1.1	Definição de modelo	61
5.1.2	Doc2Vec	62
5.1.3	Extração de informação	65
5.2	Web API	65
5.2.1	Autenticação e segurança	66
5.2.2	Preenchimento automático de perfis	67
5.2.3	Endpoints	69
5.3	Aplicação Web – Profile Matcher	71
5.3.1	Interface	72
5.3.2	Implementação	80
6	Avaliação	83
6.1	Amostra	83
6.2	Abordagem	84
6.3	Resultados	84
7	Conclusões	87
7.1	Objetivos realizados	87
7.2	Limitações e trabalho futuro	88

Índice de Figuras

Table of Figures

Figure 1: <i>Rede neuronal com camadas ocultas</i> [14].....	26
Figure 2: Fuzzy Front End[19].....	28
Figure 3: New Concept Development[20].....	29
Figure 4: Diagrama de Verna Allee[25].....	34
Figure 5: Cadeia de valores de Michael Porter[27].....	35
Figure 6: Utilizador não registado.....	46
Figure 7: <i>Estudante/Candidato</i> à procura de emprego.....	46
Figure 8: Recrutador.....	47
Figure 9: Especialista em Recrutamento.....	48
Figure 10: UC1 - Registrar.....	48
Figure 11: UC2 - Visualização de características necessárias para ofertas de emprego.....	49
Figure 12: UC3 - Visualização de recomendações de ofertas de emprego.....	49
Figure 13: UC4 - Consulta de lista de modelos existentes.....	50
Figure 14: UC5 – Criação de ofertas de emprego com base num modelo.....	51
Figure 15: UC6 – Consulta de lista das suas ofertas de emprego.....	52
Figure 16: UC7 – Consulta de lista de candidatos adequados a uma oferta de emprego.....	52
Figure 17: UC8 – Adição de modelos.....	53
Figure 18: UC9 – Edição de modelos.....	54
Figure 19: UC10 – Consulta de modelos.....	54

Figure 20: UC11 - Remoção de modelos.....	55
Figure 21: Modelo de Domínio.....	56
Figure 22: Modelo de dados.....	58
Figure 23: Diagrama de componentes.....	58
Figure 24: Exemplo gráfico de Word2Vec[32].....	63
Figure 25: JWT Token.....	67
Figure 26: Toggle Admin - fluxo de sequência.....	71
Figure 27: Login/Registo.....	72
Figure 28: Menu.....	72
Figure 29: Menu (como administrador).....	73
Figure 30: Perfil de utilizador.....	74
Figure 31: Ofertas de emprego.....	75
Figure 32: Criação de oferta de emprego.....	76
Figure 33: Gestão de modelos de emprego.....	77
Figure 34: Gestão de permissões.....	78
Figure 35: Ofertas recomendadas.....	78
Figure 36: Competências recomendadas.....	79
Figure 37: Preferências.....	79
Figure 38: Login - Diagrama de Sequência.....	80
Figure 39: ConnectionController - Post.....	81

Índice de Tabelas

Index of Tables

Table 1: Análise de <i>desempenho</i> *	37
Table 2: Matriz de comparações (Matriz A)	40
Table 3: Matriz de comparações normalizada	40
Table 4: Análise de ferramentas de <i>aprendizagem automática</i>	42
Table 5: Matriz de comparação - Desempenho	42
Table 6: Matriz de comparação - Grafos	42
Table 7: Matriz de comparação - Facilidade de aprendizagem	43
Table 8: Matriz de comparação - Visualização gráfica	43
Table 9: Matriz de comparação - Debugging	43
Table 10: Matriz de comparação - Serialização/Desserialização	43
Table 11: Modelo de emprego	62
Table 12: Excerto de características (communication skills)	64
Table 13: Endpoints da Web API	69
Table 14: Resultados da Avaliação (Resultados arredondados a duas casas decimais)	85
Table 15: Objetivos e grau de realização	88

Glossário

AHP	Método de análise hierárquica
API	Application Programming Interface
C++	Linguagem de programação
CUDA	Compute Unified Device Architecture – uma API desenvolvida pela NVIDIA, que permite a utilização do GPU para uso arbitrário.
CRUD	Create Read Update Delete
CV	Curriculum Vitae
Google Summer of Code	Concurso internacional anual da Google para estudantes universitários
Endpoint	Pontos de acesso a recursos de uma Web API. Tipicamente acedidos através de um endereço Web.
FFE	Fuzzy Front End
MVC	Padrão Model View Controller
MVVM	Padrão Model View ViewModel
NCD	New Concept Development
NPD	New Product Development
Python	Linguagem de programação
TSG	Technology State Gate
XPath	Linguagem para selecionar nós de um documento XML

1 Introdução

1.1 Contexto

O curriculum Vitae (CV) é provavelmente a ferramenta mais utilizada por recém-licenciados e jovens para expressar as suas competências, quando procuram emprego ou estágios. Um CV com as competências necessárias para uma oferta de emprego é um fator crucial para entrar no mercado laboral.

Atualmente, as empresas cada vez mais necessitam de contratar pessoas, o mais rápida e eficazmente possível. Por isso, o tempo e a eficiência no processo de seleção são fatores bastante importantes.

Não é viável que um currículo cumpra os requisitos para todas as oportunidades de emprego. Idealmente, um CV deve ser adaptado de forma a servir especificamente para cada oferta de emprego que o estudante se candidate.

Por outro lado, os recrutadores investem tempo e recursos valiosos no processo de recrutamento. A análise dos currículos dos candidatos, de modo a perceber o quão apropriados são para uma determinada oferta de emprego é uma tarefa crucial, mas trabalhosos e que consomem tempo.

De modo a satisfazer as necessidades de ambas as partes — dos recrutadores e de quem procura emprego, seria interessante a utilização de ferramentas automáticas. Seria assim possível identificar quais as competências mais relevantes para um emprego, para alguém que procure emprego. Os recrutadores poderiam também comparar as competências de candidatos mais rapidamente.

Hoje em dia, a utilização de ferramentas automáticas já tem sido mais frequente no mercado de trabalho, com o fim de reduzir os recursos que seriam necessários num processo de seleção manual.

1.2 Problema

As empresas na Europa demoram, em média, 29 dias a recrutarem um candidato, desde o primeiro contacto, até à formalização da oferta. Em determinadas áreas como em engenharia, este valor sobe até aos 33 dias, em média. O custo médio neste processo é de \$4,129[1]. Estes valores devem-se

essencialmente porque a maioria das empresas revê resumos de candidatos manualmente, conforme descrito mais sucintamente no “Estado de arte”.

Adicionalmente, os estudantes ou candidatos à procura de emprego têm, por vezes, dificuldades em apresentar CV que demonstrem adequadamente as suas qualidades, assim como as características desejadas pela empresa que se estão a candidatar. “Cerca de metade dos CV que contêm mais que duas páginas são descartados pelos recrutadores” e “em média não duram mais que cinco a sete segundos a verem um CV”[2].

É importante saber sintetizar um CV e realçar as competências do candidato que são relevantes para a oferta de emprego em questão. Inserir informação em excesso, que não seja de interesse para o cargo que o candidato esteja a candidatar, pode diminuir a probabilidade de ser escolhido.

O CV é por isso, um fator muito importante, numa fase inicial da seleção. Faz sentido que os candidatos personalizem o seu CV de forma diferente, para oportunidades de emprego diferentes. Isto deve-se ao facto de diferentes oportunidades de emprego requererem diferentes competências. Por isso, é essencial os candidatos saberem resumir o seu CV, de modo a refletir especialmente as competências que sejam relevantes à oportunidade de emprego.

Adicionalmente, quem procura emprego, pode necessitar de se candidatar a várias ofertas de emprego, até ser selecionado. De acordo com um estudo[3], “em média, cada oferta de emprego, [em grandes empresas], atrai 250 candidatos. Entre os quais, entre quatro a seis são chamados para uma entrevista, e apenas um(a) deles é selecionado”. Este estudo foi feito apenas em grandes empresas, o que significa que o número médio de candidatos por oferta de emprego noutras empresas é muito provavelmente menor. De qualquer forma, é de notar que a percentagem de candidatos aceites é bastante baixa, o que implica que quem procura emprego, necessite normalmente, de se candidatar a várias oportunidades de emprego.

É evidente que existe uma falta de eficiência no processo de seleção, tanto no lado de quem recruta, como de quem pretende ser recrutado. O problema em questão não aborda necessariamente a competência, ou a falta dela dos candidatos, em desempenhar um emprego. Trata-se sim de uma falta de comunicação das competências entre os candidatos e os recrutadores. Uma solução que auxiliasse tanto os estudantes em comunicar as suas qualidades, como os recrutadores em avaliar corretamente essas mesmas qualidades, tornaria o processo de seleção nas empresas muito mais eficiente.

1.3 Objetivo do projeto

Este projeto tem como objetivo evitar os problemas descritos na secção anterior. O fim deste projeto é por isso, auxiliar o processo de seleção de candidatos para um emprego, assim como ajudar os candidatos a identificar habilitações que precisem de ser melhoradas afim de que estejam aptos para executar uma dada atividade profissional.

Com esta tese, sugerida pelo docente Nuno Escudeiro, pretende-se desenvolver um sistema que analise CV e propostas de emprego e indique quais os melhores emparelhamentos. Esta solução pode ser utilizada quer em processos de pré-seleção de candidatos, quer em casos em que um candidato pretenda saber quais as competências a desenvolver para que tenha sucesso numa hipotética candidatura a emprego. Para ambos os casos, são previamente definidas diversas áreas de negócio. O sistema deverá auxiliar o processo de escolha do melhor subconjunto de candidatos, recorrendo a técnicas de aprendizagem automática.

Assim sendo, este sistema vai ser utilizado tanto por recrutadores como por quem procure emprego. Existem objetivos distintos para cada um destes intervenientes. Pretende-se que os recrutadores utilizem o sistema para procurar candidatos a emprego e que os candidatos a emprego o utilizem para identificar as competências a desenvolver para um potencial emprego.

1.3.1 Recrutadores

Os recrutadores vão poder divulgar as suas ofertas de emprego a potenciais candidatos. Esta aplicação terá como objetivo auxiliar os recrutadores a filtrar candidatos com base nos seus CV ou através de outros dados fornecidos.

O facto de este processo ser automático tornará o processo de seleção mais rápido. Poderá também eliminar qualquer eventual discriminação (e.g. por idade, género, etc.).

É pretendido que a aplicação identifique o maior número de características relevantes para uma determinada oferta de emprego.

Devido a limitações tecnológicas, ou por falta de dados fornecidos pelos candidatos, não é expectável que sejam identificadas ou avaliadas todas as características relevantes para um emprego. Ou seja, o objetivo deste sistema não é o de substituir por completo a revisão manual dos recrutadores, mas sim servir como uma ferramenta complementar no processo de seleção.

1.3.2 Candidatos a emprego

Os candidatos a emprego vão poder criar um perfil de candidato. Este perfil contém as habilitações atuais e características do candidato. Sempre que possível, estes dados são extraídos automaticamente através de redes sociais (e.g. LinkedIn, Facebook), CV, entre outros. Quando não for possível obter estes dados, total ou parcialmente, os mesmos são solicitados ao candidato, através de um questionário.

Pretende-se que a aplicação indique quais são as habilitações que devem ser melhoradas ou obtidas, para os candidatos a emprego. Quanto aos recrutadores, pretende-se que a aplicação auxilie a escolher o candidato mais adequado a uma oferta de emprego.

Este projeto só terá valor, se os resultados obtidos forem fiáveis e mais próximos da realidade possível. Por isso, foi decidido que se utilizassem técnicas de aprendizagem automática e métodos estatísticos — conforme referido na secção “Hipótese”, de modo a identificar padrões dos candidatos que têm mais e menos sucesso.

1.4 Resultados esperados

Pretende-se com este projeto responder às seguintes questões:

- É possível identificar padrões de candidatos com base nos seus perfis, com técnicas de aprendizagem automática?
- É mais rápido utilizar a aplicação desenvolvida, em vez de uma análise manual, no processo de seleção de candidatos?
- É viável selecionar candidatos, com o auxílio desta aplicação?

De modo a responder a estas questões, é necessário avalia-las estatisticamente, algo que é descrito na secção “Avaliação”.

1.5 Organização do relatório

Este documento está dividido em sete capítulos.

O primeiro capítulo contextualiza o problema identificado e os objetivos desta tese, assim como hipóteses do projeto que se pretendam testar.

O segundo capítulo descreve o estado de arte do problema identificado, assim como das tecnologias utilizadas, relevantes para este projeto. Adicionalmente, descreve a análise de valor definida para este projeto.

O terceiro capítulo tem como objetivo avaliar as tecnologias mais adequadas para este projeto, com base em determinados critérios definidos.

O quarto capítulo descreve a análise e o desenho desenvolvidos para as aplicações a serem desenvolvidas.

O quinto capítulo descreve e documenta a implementação do projeto.

O sexto capítulo diz respeito à avaliação realizada, de modo a identificar se as hipóteses definidas no primeiro capítulo são válidas.

Por fim, o sétimo capítulo descreve as conclusões obtidas após o desenvolvimento deste projeto.

2 Estado de arte e Análise de valor

2.1 Estado de arte

2.1.1 Soluções existentes

Com a evolução constante da tecnologia, ferramentas como o LinkedIn, surgiram de forma a conectar candidatos com recrutadores, o que facilitou bastante tanto empregar, como arranjar emprego. No entanto, devido ao vasto número de utilizadores inscritos nestas ferramentas, tornou-se difícil conseguir filtrar de forma eficaz os mesmos, de modo a obter um candidato ideal.

A utilização de ferramentas automáticas, para a seleção de candidatos, surgiu para combater este problema, e tem vindo a ser cada vez mais utilizada pelas empresas. Estas ferramentas adquirem dados sobre os candidatos, em redes sociais como o LinkedIn ou outras fontes de informação, e preveem quais são os mais adequados, de acordo com determinados critérios pré-definidos.

A “Launchpad Recruits” é um exemplo de uma plataforma deste tipo, treinada através de aprendizagem automática. Neste caso, esta aplicação avalia os candidatos através dos seus comportamentos em entrevistas por vídeo. Ou seja, através de comportamentos como “gestos, movimentos dos olhos, expressões ou palavras escolhidas”[4] esta ferramenta prevê quais os candidatos que, à partida, sejam adequados para uma determinada tarefa.

A “Transformify” é outro exemplo. Esta plataforma facilita tanto o recrutamento, como a procura de emprego. Quem procura emprego, fornece dados como as suas competências, salários desejados e localização. O objetivo desta aplicação é de automaticamente sugerir aos recrutadores os candidatos que têm competências semelhantes às desejadas para uma determinada oferta de emprego, assim como indicar as melhores oportunidades para quem procura emprego[5].

A “Ideal” é uma ferramenta que analisa vários dados de candidatos, como currículos, e filtra os melhores candidatos. Uma característica única desta plataforma é que permite avaliar uma base de dados de candidatos já existente, de modo a analisar mais precisamente os candidatos, com o contexto da empresa. Possui também um “chatbot” capaz de comunicar com os candidatos[6].

Estas tecnologias trazem vantagens, como evitar discriminações na seleção de candidatos (e.g. raciais, de género, etc.) ou o facto de o recrutamento ser feito mais rapidamente. Algumas destas ferramentas incluem “chatbots” que são

capazes de enviar mensagens para marcar entrevistas com potenciais candidatos[7].

De acordo com um estudo feito pela "Jobvite"[8], 49% dos recrutadores estão a favor destas ferramentas, enquanto 7% estão contra. De acordo com o mesmo estudo, 13% dos recrutadores dizem que estas tecnologias já têm impacto nos seus planos de seleção, e 22% que a sua empresa está a ponderar aplicá-las. Existe, portanto, uma reação geralmente positiva, à utilização de ferramentas automáticas. No entanto, a adesão a estas ferramentas por parte das empresas é ainda relativamente baixo.

Por outro lado, existe também muito debate sobre a eficácia das ferramentas que existem hoje. Os critérios que a ferramenta automática utiliza, têm impacto diretamente na escolha dos melhores candidatos. Se uma ferramenta utilizar critérios que não sejam adequados, esta torna-se menos fiável. Outra desvantagem é a potencial exclusão de candidatos que se destaquem, positivamente, dos outros. Métodos de aprendizagem no recrutamento tendem a definir critérios para um "candidato ideal", o que significa que candidatos que tenham características únicas, possam ser excluídos na pré-seleção. Neste caso, é essencial que sejam fornecidos dados em quantidade e qualidade, de modo que estas ferramentas sejam treinadas para considerar vários "perfis de candidatos ideais".

Pode-se concluir que, apesar de existirem alguns problemas nas ferramentas automáticas de recrutamento utilizadas atualmente, elas permitem selecionar candidatos de forma muito mais rápida do que quando selecionados manualmente. Apesar de haver, geralmente, uma reação positiva à utilização destas ferramentas, não existe ainda um grande número de empresas que as utilizem. Possivelmente por ser uma área ainda muito recente, também ainda não existem muitas ferramentas automáticas fiáveis. No entanto, a tendência é que estas venham a aumentar e a melhorar com o tempo.

2.1.2 Desenvolvimento de ferramentas automáticas

Existem algumas estratégias para a implementação de ferramentas automáticas. Vão ser descritas algumas técnicas utilizadas, relevantes para o desenvolvimento do Profile Matcher.

Data mining

“*Data mining* é o processo de descobrir padrões em largos conjuntos de dados, através de métodos como aprendizagem automática, estatística e sistemas de base de dados”[9].

Atualmente, em grande parte devido à evolução da Internet, existe uma quantidade enorme de dados, acessíveis publicamente. Em junho de 2019, 4,4 mil milhões de pessoas tinham acesso à Internet[10]. Entre as quais, 2,32 mil milhões têm pelo menos uma conta criada no Facebook[11].

Data mining é por isso, cada vez mais relevante no processo de procura de informação.

Aprendizagem automática

“Aprendizagem automática é o estudo científico de algoritmos e de modelos estatísticos que sistemas computacionais usam para executar uma tarefa, sem usar instruções explicitamente”[12].

No processo de *Data mining*, muitas vezes não é viável identificar padrões estatisticamente. No caso de a quantidade de dados ser elevada e/ou haver dados que estejam dependentes de várias variáveis, torna-se inviável identificar padrões puramente por estatística.

“Algoritmos de aprendizagem automática consistem em construir um modelo matemático baseado num conjunto de dados, conhecido por *training data*”[12]. Este modelo matemático vai sendo moldado pelo algoritmo, consoante os dados que obteve do *training data*, de modo que este modelo consiga fazer previsões e/ou decisões sobre dados futuros, o mais corretamente possível.

Rede neuronal artificial

“É um sistema computacional inspirado pelas redes neuronais biológicas. Consistem em aprenderem progressivamente como fazer determinadas tarefas, considerando exemplos. Por exemplo, no reconhecimento de imagens, estes sistemas podem aprender a identificar imagens que contenham gatos ao analisar exemplos de imagens que tenham manualmente sido identificados como ‘gato’ ou ‘não gato’”[13].

Tipicamente, as redes neurais são divididas em duas fases – fase de treino e fase de teste. A primeira fase consiste em analisar inputs fornecidos e em corrigir a rede, quando o output obtido não é o esperado, e em fortalecer a rede caso contrário. Posteriormente é feita a segunda fase, onde são fornecidos inputs e é calculado o número de falsos positivos.

Camada de rede neuronal

Uma camada representa uma coleção de nós que operam em conjunto numa rede neuronal. Numa rede neuronal, existem camadas de input, de output e potencialmente camadas ocultas (*hidden layers*).

Nas camadas de input, os nós representam os dados de uma tarefa.

Nas camadas de output, os nós representam os resultados da tarefa.

As camadas ocultas são camadas intermédias que tentam identificar padrões com base em diferentes conjuntos de características. Por exemplo, no reconhecimento de imagem de uma cara, a primeira camada oculta poderia tentar identificar a forma do rosto, enquanto a camada seguinte identificaria os olhos, etc. O objetivo destas camadas é, portanto, de dividir a tarefa em sub-tarefas mais simples, e identificar padrões em cada uma das sub-tarefas[14]. É de notar que estas sub-tarefas são, por norma, definidas pela própria rede neuronal durante o treino. Ou seja, no exemplo anterior, não é definido explicitamente que uma das sub-tarefas seria identificar os olhos. A própria rede neuronal é que vai identificando e moldando essas sub-tarefas.

A imagem seguinte representa visualmente os nós e as camadas, assim como as suas relações numa rede neuronal.

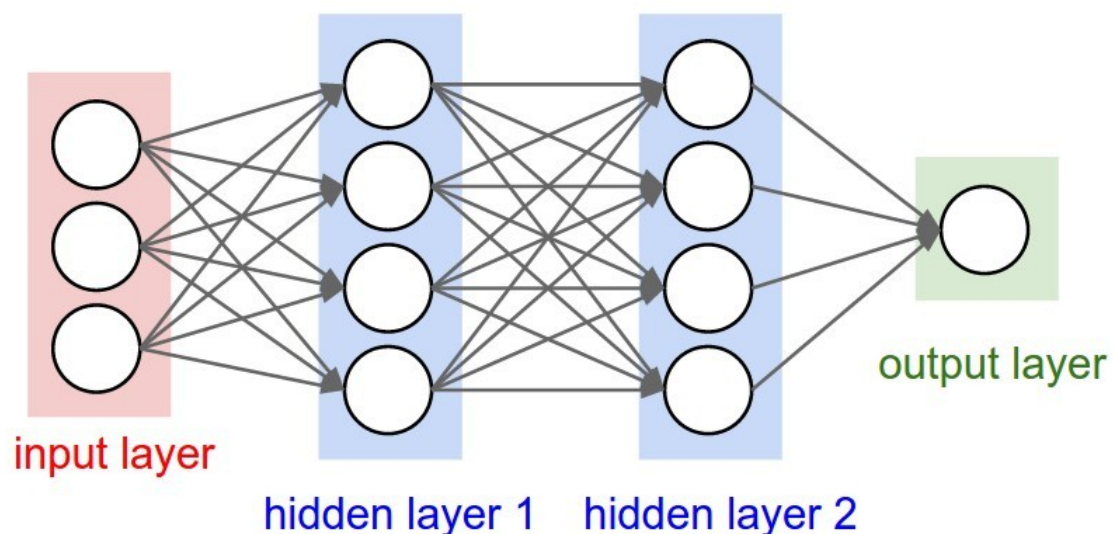


Figure 1: Rede neuronal com camadas ocultas[14]

Existem várias classes de redes, como *Convolutional Neural Network*, *Residual Neural Network*, entre outros. Cada uma destas classes tem formas diferentes de identificar padrões. Algumas destas classes podem ser mais ou menos adequadas, dependendo da tarefa em questão.

Rede neuronal profunda

“Uma Rede neuronal profunda é uma rede neuronal artificial com várias camadas entre as camadas de input e as de output. A rede tenta descobrir uma manipulação matemática correta para transformar um input num output.”[13]

2.1.3 Bibliotecas de Aprendizagem automática

Existem várias bibliotecas que facilitam o acesso a algoritmos de aprendizagem automática. Estas bibliotecas são tipicamente usadas visto, por norma, serem mais otimizados e permitem um desenvolvimento mais rápido de redes neuronais, que uma implementação feita “de raiz”, o que garante um melhor desempenho e estabilidade.

Algumas das bibliotecas mais utilizadas, com repositório público no Github, são o “Tensorflow”, “Keras” e “PyTorch”[15].

Tensorflow

O Tensorflow é um projeto open-source desenvolvido e mantido pela Google. Empresas como *airbnb*, *Coca-cola* ou *Intel* utilizam esta ferramenta[16].

Pode ser utilizado com recurso a aceleração de GPU, em processadores gráficos NVIDIA, que suportem os *drivers* da CUDA. Esta opção tem um impacto positivo na performance. Alternativamente pode também ser utilizado através do CPU.

Existem API estáveis em Python e em C++, mantidos pela Google. No entanto, existem outras API noutras linguagens, desenvolvidas pela comunidade[16].

Keras

Um engenheiro da Google, François Chollet, começou por desenvolver esta biblioteca, em 2015.

Keras é uma ferramenta de alto nível, que “pode ser usada em conjunto com o Tensorflow, Microsoft Cognitive Toolkit, R, Theano ou PlaidML. Tem como objetivo uma experimentação rápida com redes neuronais profundas e foca-se em ser de fácil utilização, modular e extensível”[17].

Segundo o autor, Chollet, esta biblioteca não deve ser usada isoladamente, mas sim em conjunto com outras ferramentas, como o Tensorflow, por exemplo[17].

A API do Keras foi desenvolvida em Python.

PyTorch

PyTorch é uma ferramenta para redes neurais, baseada no Torch, que permite a utilização de aceleração de GPU, em processadores gráficos NVIDIA, que suportem os “drivers” da CUDA[18].

Permite criar novas camadas de redes neurais se necessário, através da API do Torch.

Esta ferramenta foi desenvolvida em Python, pelo Facebook, em 2016.

Qual destas ferramentas é a mais viável de ser utilizada depende de várias variáveis. A decisão tomada e a sua justificação é descrita sucintamente no capítulo “Avaliar soluções existentes”.

2.2 Análise de valor

2.2.1 New Concept Development Model

De acordo com Peter Koen, “o processo de inovação deve ser dividido em três áreas: *Fuzzy Front End*, *New Product Development Process* e comercialização”[19]. O modelo de Peter Koen é descrito na seguinte imagem.

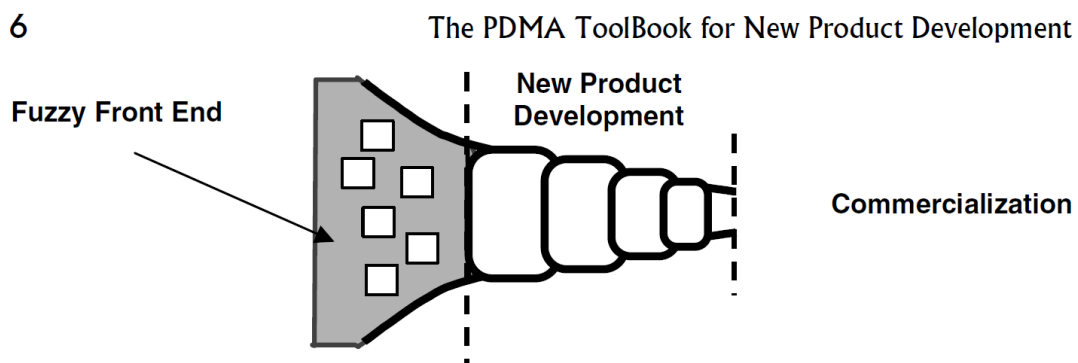


Figure 2: Fuzzy Front End[19]

“Muitas das práticas utilizadas na parte do NPD não se aplicam no FFE (...) devido à natureza do trabalho, data de comercialização, nível de financiamento, (...) entre outros, serem fundamentalmente diferentes.”[19]

Assim sendo, visto existir uma diferença considerável entre o FFE e o NPD, surgiu o New Concept Development (NCD), que é demonstrado na imagem seguinte.

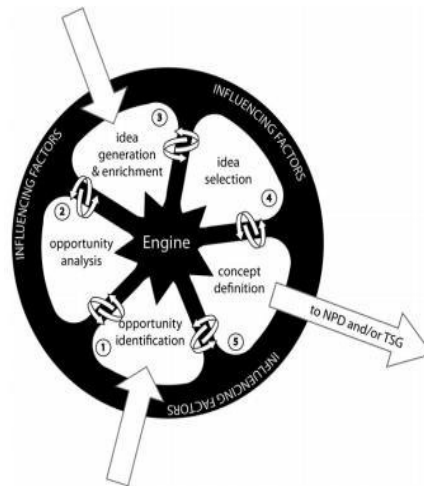


Figure 3: New Concept Development[20]

Este modelo consiste em quatro partes chave:

- O motor (*Engine*), que representa o apoio à gestão, é a base dos cinco elementos no modelo NCD.
- Os cinco elementos presentes no modelo. As ideias e conceitos devem ser iteradas por estes cinco elementos.
- Os pontos de partida no desenvolvimento de um projeto, que podem ser indicados tanto no “Idea Generation & Enrichment” como no “Opportunity Identification”. Estes pontos de partida são identificados pelas setas apontadas para o interior do modelo.
- Como os conceitos saem do modelo e entram no NPD ou no processo de TSG (Technology State Gate), representados pelas setas apontadas para o exterior do modelo[19].

De seguida, são identificados os cinco elementos do modelo, no projeto “Profile Matcher”.

- **Identificação de oportunidade**
 - A oportunidade deste projeto consiste em facilitar a seleção de candidatos, tanto no lado de quem recruta, como de quem se pretende candidatar, aplicando técnicas de aprendizagem automática.
- **Análise de oportunidade**
 - Atualmente não existem muitas soluções no mercado. A seleção de candidatos ainda é feita manualmente na maioria das organizações. Assim sendo, com o Profile Matcher espera-se melhorar a eficiência e

a rapidez no processo de seleção de candidatos nas organizações, assim como na procura de emprego nos candidatos a emprego.

- **Geração de ideias e enriquecimento**

- Nesta fase, foram consideradas algumas ideias de como obter informação necessária para aplicar técnicas de aprendizagem automática. Entre as quais, foi considerado obter essa informação através de data-mining ou através de questionários de indivíduos que já se tenham candidatado no passado. Adicionalmente, também foi debatido se este sistema seria disponibilizado aos utilizadores através de uma aplicação mobile, web ou ambos.

- **Seleção de ideias**

- Com base nas ideias ponderadas no parágrafo anterior, as que foram selecionadas são:
 - Extração de informação, principalmente através de API públicas ou de websites de procura de emprego (e.g. LinkedIn, Indeed, etc.).
 - Desenvolvimento de uma aplicação web.

- **Definição de conceito**

- O conceito deste projeto baseia-se em desenvolver uma aplicação capaz de identificar padrões entre candidatos e uma aplicação web, que faça uso dos padrões identificados, para facilitar a seleção de candidatos.

2.2.2 Valor

O valor representa os benefícios que um produto ou serviço fornece a um cliente. Tipicamente questiona-se qual é o máximo valor monetário que um cliente esteja disposto a pagar pelo produto ou serviço.

O valor pode variar consoante variáveis externas, como a alteração de preços da concorrência, ou durante uma fase onde há uma maior aderência na área de negócio do produto, ou serviço.

No caso do Profile Matcher, a concorrência pode ter um impacto no seu valor. No entanto, a aplicação deverá manter-se relevante ao longo do tempo, pois a procura de emprego e seleção de candidatos são cruciais no mercado de trabalho, em qualquer local e em qualquer altura.

Valor para o cliente

O valor para o cliente representa o valor que o cliente considera que um produto ou serviço tenha.

O valor para o cliente e o valor “real” do produto ou serviço não são necessariamente semelhantes.

O cliente, que pode ser tanto um recrutador como um estudante ou candidato a emprego, pode ver valor no Profile Matcher visto auxiliar o processo de seleção de candidatos, assim com auxilia a escolha de um novo emprego.

Valor percebido

“O valor percebido consiste na avaliação que os clientes fazem, relativamente à qualidade ou desejo de um produto, ou serviço.”[21]

O *marketing* é um grande fator para influencia o valor percebido dos clientes/potenciais clientes[21].

O valor percebido do Profile Matcher consiste na ideia de que a aplicação aumenta a eficácia e a rapidez no processo de seleção ou de procura, no mercado de trabalho.

Benefícios e sacrifícios

De acordo com os conceitos definidos nesta secção, os benefícios e os sacrifícios do Profile Matcher são descritos na tabela seguinte.

Benefícios	Sacrifícios
Facilidade de introdução de dados	Resultados não são garantidamente corretos
Facilidade de uso	Custo (no acesso premium)
Recomendações de competências a melhorar	
Maior eficiência no processo de seleção	

2.2.3 Proposta de valor

A proposta de valor é uma frase que identifique de forma clara, medível e demonstrável, os benefícios que os clientes obtêm quando compram o produto ou serviço[22].

Com isto em mente, e com base nos benefícios do Profile Matcher, a proposta de valor deste projeto é definida pela seguinte frase:

"Uma aplicação destinada a jovens à procura de emprego, que sugere competências que o candidato deva melhorar, assim como a recrutadores, que recomenda os candidatos mais adequados para uma oferta de trabalho."

2.2.4 Modelo Canvas

"O Modelo Canvas é um *template* que tem como objetivo desenvolver ou documentar um modelo de negócio existente." [23]

É uma ferramenta útil no processo de análise de valor, pois permite avaliar, de forma sintetizada, se o modelo de negócio tem as condições necessárias para ser viável. A tabela seguinte, demonstra o Modelo Canvas desenhado para este projeto.

Table 1: Modelo Canvas

Parceiros Chave • ISEP	Atividades chave • Execução de aprendizagem automática • Desenvolvimento da aplicação • Manutenção	Proposta de valor • Auxílio no processo de recrutamento • Recomendação de competências a adquirir	Relações com os clientes • Redes Sociais • E-mails	Segmentos de clientes • Recrutadores • Estudantes • Jovens à procura de emprego
	Recursos chave • Infraestrutura informática • Servidores • Licenças de Software		Canais de distribuição • Website • Redes Sociais	
Estrutura de custos • Servidores • Custos de desenvolvimento • Custos de manutenção		Fluxo de receitas • Publicidade • Acesso <i>premium</i> com funcionalidades extra		

2.2.5 Modelos de análise de valor

“O objetivo de uma rede de valor é de gerar sucesso económico (...) para os seus participantes.”[24]

Verna Allee acredita que a relação e a colaboração mútua entre membros de uma organização pode ser a base do sucesso. Considera assim, que numa rede de valor, idealmente, os participantes contribuem mutuamente, de modo a obterem sucesso próprio, assim como sucesso na rede na totalidade[24].

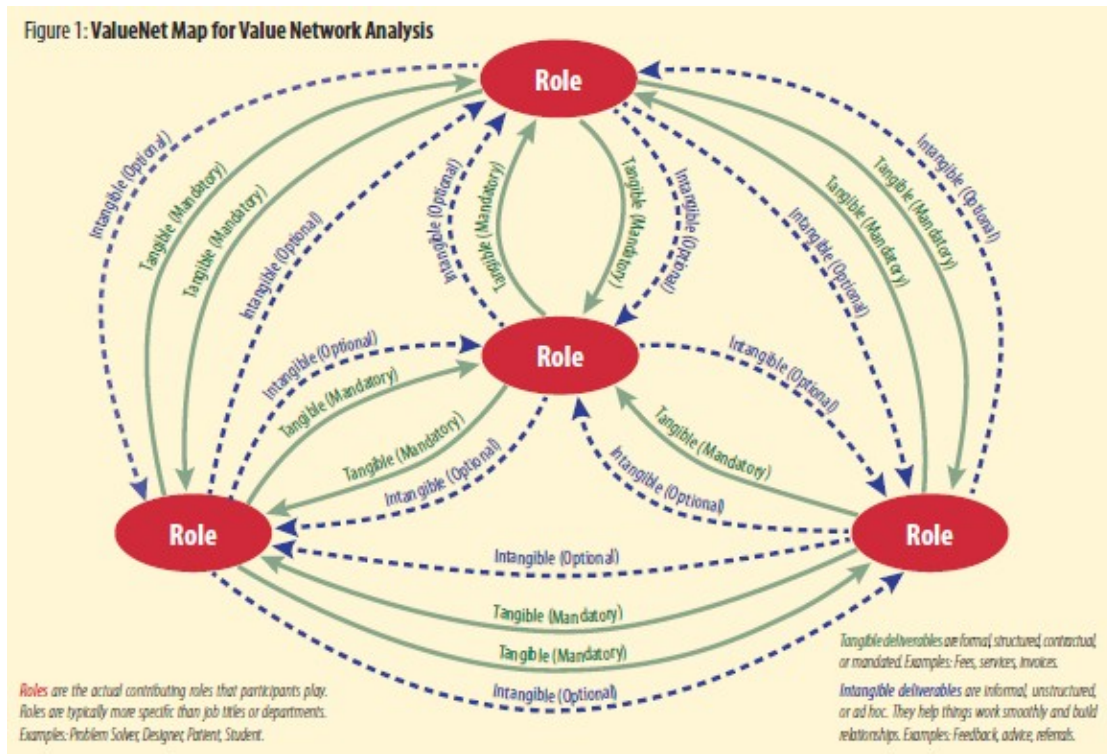


Figure 4: Diagrama de Verna Allee[25]

“A cadeia de valores de Michael Porter refere que existe uma cadeia de eventos que ocorrem numa empresa, desde a procura de matéria-prima, até à entrega de produtos.”[26]

Segundo Porter, qualquer um dos nove passos, demonstrados na imagem seguinte, têm impacto no produto final. Por isso, se uma empresa pretender adicional valor a um produto ou serviço, deve referir-se a esta cadeia de valores[26].

Figure 1: Porter's Generic Value Chain



Figure 5: Cadeia de valores de Michael Porter[27]

Com base nos dois modelos referidos, o modelo de Verna Allee parece ser o mais adequado a este projeto, visto a relação com o ISEP ou potencialmente de outras parcerias, serem essenciais para a divulgação do Profile Matcher.

3 Avaliar soluções existentes

Das ferramentas de aprendizagem automática descritas na secção anterior, pretende-se identificar aquela que seja mais adequada para este projeto. Algumas das características a ter em conta para a decisão da ferramenta a usar são as seguintes:

- Desempenho
- Facilidade de utilização e de aprendizagem
- Visualização gráfica do progresso
- Facilidade em guardar e carregar modelos treinados
- Acessibilidade em identificar problemas

Tendo em conta que a biblioteca Keras não é pretendida ser utilizada isoladamente, foi decidido analisar as ferramentas Tensorflow ou PyTorch.

3.1 Análise das ferramentas

3.1.1 Desempenho

Para avaliar o desempenho das ferramentas e, tendo em conta que ambas permitem a aceleração de GPU da NVIDIA, foram analisados testes publicados pela NVIDIA entre várias ferramentas de aprendizagem automática, incluindo as duas a serem analisadas[28].

A tabela seguinte mostra um excerto dos testes realizados, com o PyTorch e com o Tensorflow, dado um mesmo dataset, camada e versão da placa gráfica (i.e. nas mesmas condições).

*Table 1: Análise de desempenho**

Ferramenta	Taxa de transferência (imagens por segundo)/Network usado		
Tensorflow	832 (Inception V3)	1237 (ResNet-50 V1.5)	148 (SSD v1.1)
PyTorch	607 (Inception V3)	1028 (ResNet-50 V1.5)	265 (SSD v1.1)

*DataSet: ImageNet2012; Tipo de rede: Convolutional Neural Network; Versão do GPU: V100-SXM3-32GB-H

De acordo com estes testes, parece haver um ligeiro melhor desempenho com o Tensorflow (em média 739 imagens/segundo contra 633 imagens/segundo). No entanto, é de notar que o desempenho foi melhor com o PyTorch quando utilizado o SSD v1.1, o que significa que a melhor ferramenta a utilizar está também dependente dos tipos de *Network Layer* utilizados.

3.1.2 Grafos

O Tensorflow e o PyTorch têm conceitos distintos na definição de grafos. Com o Tensorflow, é necessário definir um grafo e apenas depois o modelo pode ser computado. Com o PyTorch, os grafos são definidos dinamicamente. Isto permite a modificação dos mesmos, durante a computação do modelo, visto o código ser interpretado ao invés de ser compilado[16][18].

A possibilidade de modificar grafos dinamicamente é vista como uma vantagem do PyTorch, visto ser habitual alterar frequentemente grafos, de modo a tentar obter uma solução mais otimizada.

3.1.3 Facilidade de aprendizagem

É possível utilizar ambas as ferramentas em Python. O PyTorch segue as convenções do Python e utiliza maioritariamente dependências *standard* do Python (e.g. NumPy). Apesar do Tensorflow também utilizar dependências *standard*, tem também dependências desenvolvidas pelo Tensorflow.

Isto não é necessariamente um aspeto negativo do Tensorflow, mas significa que há uma maior dificuldade em aprender os conceitos desta ferramenta.

3.1.4 Visualização gráfica

O Tensorflow tem acesso a uma biblioteca chamada Tensorboard, que permite visualizar graficamente e interpretar grafos. O PyTorch pode utilizar o TensorboardX, que é baseado no Tensorboard. São ferramentas bastante úteis, pois permitem identificar potenciais problemas, que seriam muito dificilmente identificados sem uma perspetiva visual[29].

3.1.5 Debugging

O debugger *standard* do Python é normalmente utilizado em ambas as ferramentas (assumindo que seja usado Python no Tensorflow). No entanto, o

Tensorflow inclui um debugger que pode ser utilizado em conjunto com o Tensorboard, de modo a identificar problemas visualmente, à medida que o grafo é computado[29].

Esta ferramenta ainda é uma versão *alpha* e não está totalmente funcional, mas que mesmo assim traz bastante valor.

3.1.6 Serialização/Desserialização

Serializar e desserializar grafos e/ou modelos é essencial, caso contrário as computações feitas seriam perdidas quando a aplicação terminasse. O Tensorflow permite a serialização/desserialização de grafos inteiros, incluindo parâmetros e operações utilizadas. Adicionalmente, através de outras linguagens como C++ ou Java, é possível desserializar esse mesmo grafo. Quanto ao PyTorch, apenas é possível armazenar pesos de modelos[16][18].

Apesar de o armazenamento de pesos de modelos ser suficiente nalguns casos, o Tensorflow é bastante mais vantajoso neste aspeto.

3.2 Decisão

Decidiu-se aplicar o método de análise hierárquica (AHP) para identificar qual a ferramenta que seja mais adequada para o desenvolvimento desta aplicação.

Assim sendo, definiu-se uma matriz de comparações (Matriz A), demonstrada na tabela seguinte.

Table 2: Matriz de comparações (Matriz A)

Critério	Desempenh o	Grafo s	Facilidade de aprendizage m	Visualização gráfica	Debuggi ng	Serialização
Desempenh o	1	2	3	1	1	2
Grafos	1/2	1	1	1/2	1/2	1
Facilidade de aprendizage m	1/3	1	1	1/2	1/2	1/2
Visualização gráfica	1	2	2	1	1	2
Debugging	1	2	2	1	1	2
Serialização	1/2	1	2	1/2	1/2	1

Depois de normalizada, obteve-se a seguinte matriz:

Table 3: Matriz de comparações normalizada

Critério	Desempenh o	Grafos	Facilidade de aprendizage m	Visualizaçã o gráfica	Debuggi ng	Serializaçã o	Prioridad e relativa
Desempenh o	6/26	2/9	3/11	2/9	2/9	4/17	0,2342
Grafos	3/26	1/9	1/11	1/9	1/9	2/17	0,1095
Facilidade de aprendizage m	3/39	1/9	1/11	1/9	1/9	1/17	0,0933
Visualização gráfica	6/26	2/9	2/11	2/9	2/9	4/17	0,2191
Debugging	6/26	2/9	2/11	2/9	2/9	4/17	0,2191
Serialização	3/26	1/9	2/11	1/9	1/9	2/17	0,1247

O vetor próprio pode ser definido como:

$$\begin{bmatrix} 0,23 \\ 0,11 \\ 0,09 \\ 0,22 \\ 0,22 \\ 0,12 \end{bmatrix}$$

De seguida, calculou-se a Razão de Consistência para medir o quão consistentes foram os julgamentos.

Considerando x , onde x é o vetor próprio.

$$\begin{bmatrix} 1 & 2 & 3 & 1 & 1 & 2 \\ 0,5 & 1 & 1 & 0,5 & 0,5 & 1 \\ 0,33 & 1 & 1 & 0,5 & 0,5 & 0,5 \\ 1 & 2 & 2 & 1 & 1 & 2 \\ 1 & 2 & 2 & 1 & 1 & 2 \\ 0,5 & 1 & 2 & 0,5 & 0,5 & 1 \end{bmatrix} * \begin{bmatrix} 0,23 \\ 0,11 \\ 0,09 \\ 0,22 \\ 0,22 \\ 0,12 \end{bmatrix} = \lambda_{max} \begin{bmatrix} 0,23 \\ 0,11 \\ 0,09 \\ 0,22 \\ 0,22 \\ 0,12 \end{bmatrix}$$

⇔

$$\begin{bmatrix} 1,4 \\ 0,66 \\ 0,56 \\ 1,31 \\ 1,31 \\ 0,75 \end{bmatrix} = \lambda_{max} \begin{bmatrix} 0,23 \\ 0,11 \\ 0,09 \\ 0,22 \\ 0,22 \\ 0,12 \end{bmatrix}$$

⇔

$$\lambda_{max} = \text{média}(1,4/0,23, 0,66/0,11, 0,56/0,09, 1,31/0,22, 1,31/0,22, 0,75/0,12) = 6,08$$

O valor próprio é de 6,08, logo o índice de consistência é:

$$IC = (\lambda_{max} - n) / (n - 1) = (6,08 - 6) / (6 - 1) = 0,016$$

Visto $n=6$, a razão de consistência é:

$$RC = IC / 1,24 = 0,016 / 1,24 = 0,013$$

Como $0,013 < 0,1$, podemos concluir que os valores utilizados são consistentes.

Assim sendo, podemos comparar as ferramentas analisadas. A tabela seguinte sintetiza a análise feita entre as alternativas.

Table 4: Análise de ferramentas de aprendizagem automática

Biblioteca	Tensorflow	PyTorch
Desempenho	Ligeiramente melhor	Ligeiramente pior
Grafos	Estáticos	Dinâmicos
Facilidade de aprendizagem	Integrado com bibliotecas de <i>data science</i> do Python, mas com vários conceitos específicos do Tensorflow	Maioritariamente integrado com bibliotecas de <i>data science</i> do Python
Visualização gráfica	Possível com Tensorboard	Possível com TensorboardX
Debugging	Contém um debugger próprio do Tensorflow	Debugger <i>standard</i> do Python
Serialização/ Desserialização	Permite armazenar grafos inteiros	Apenas permite armazenar pesos de modelos

Com base nesta análise, podemos construir a matriz de comparação paritária para cada critério. Essas matrizes são indicadas nas tabelas seguintes, já normalizadas.

Table 5: Matriz de comparação - Desempenho

Ferramenta	Tensorflow	PyTorch	Prioridade relativa
Tensorflow	2/3	2/3	0,67
PyTorch	1/3	1/3	0,33

Table 6: Matriz de comparação - Grafos

Ferramenta	Tensorflow	PyTorch	Prioridade relativa
Tensorflow	1/4	1/4	0,25
PyTorch	3/4	3/4	0,75

--	--	--	--

Table 7: Matriz de comparação - Facilidade de aprendizagem

Ferramenta	Tensorflow	PyTorch	Prioridade relativa
Tensorflow	2/3	1/3	0,67
PyTorch	1/3	2/3	0,33

Table 8: Matriz de comparação - Visualização gráfica

Ferramenta	Tensorflow	PyTorch	Prioridade relativa
Tensorflow	0,5	0,5	0,5
PyTorch	0,5	0,5	0,5

Table 9: Matriz de comparação - Debugging

Ferramenta	Tensorflow	PyTorch	Prioridade relativa
Tensorflow	3/4	3/4	0,75
PyTorch	1/4	1/4	0,25

Table 10: Matriz de comparação - Serialização/Desserialização

Ferramenta	Tensorflow	PyTorch	Prioridade relativa
Tensorflow	2/3	2/3	0,67
PyTorch	1/3	1/3	0,33

De seguida, multiplicamos as matrizes de comparação com os respetivos pesos de cada critério.

$$\begin{bmatrix} 0,67 & 0,25 & 0,67 & 0,5 & 0,75 & 0,67 \\ 0,33 & 0,75 & 0,33 & 0,5 & 0,25 & 0,33 \end{bmatrix} * \begin{bmatrix} 0,23 \\ 0,11 \\ 0,09 \\ 0,22 \\ 0,22 \\ 0,12 \end{bmatrix} = \begin{bmatrix} \mathbf{0,60} \\ 0,39 \end{bmatrix}$$

Depois de analisadas as duas ferramentas, obtemos um valor mais elevado para o Tensorflow (0,60 contra 0,39). Assim sendo, conclui-se que o **Tensorflow** é a ferramenta mais adequada para o desenvolvimento deste projeto, de acordo com os critérios dados.

4 Análise e Desenho

Decidiu-se aplicar técnicas de aprendizagem automática para o desenvolvimento desta solução. Assim sendo, é necessário previamente obter dados sobre indivíduos que se candidataram a empresas, como os CV que utilizaram, o tipo de emprego a que se candidataram e possivelmente informação pessoal (e.g. idade, naturalidade, etc.). Estes dados deverão ser obtidos utilizando técnicas de “data mining”, através de API públicas, redes sociais (como LinkedIn) ou outros. Posteriormente, uma parte destes dados serão treinados e o restante será testado. Este processo é descrito mais sucintamente na “Implementação”.

Depois de serem obtidos resultados da aplicação referida, será implementada uma aplicação web, que será utilizada diretamente pelos utilizadores. Os utilizadores serão estudantes/candidatos ou recrutadores. Esta aplicação tirará proveito dos resultados concluídos da solução anterior.

Os candidatos poderão fornecer o seu CV (e.g. em formato PDF). Quando possível, os dados serão extraídos automaticamente pela aplicação. O utilizador opcionalmente poderá completar ou editar alguns campos, caso não seja possível obter todos os dados de forma automática.

Os recrutadores poderão criar ofertas de emprego nesta aplicação, onde serão indicados quais são as qualificações necessárias. Posteriormente, o utilizador pode comparar diferentes candidatos que se tenham registado nesta aplicação.

A aplicação que vai ser responsável por treinar e testar os dados obtidos dos candidatos, vai ser arquitetada monoliticamente. Isto deve-se ao facto de não existir um output visual nesta aplicação. Adicionalmente, aplicações de aprendizagem automática tendem a ser bastante modulares e de não terem uma grande dimensão. Por estes motivos, as vantagens que o padrão MVC teria relativamente ao monolítico, como o baixo acoplamento entre a lógica de negócio e a representação visual, ou a sua escalabilidade, não teriam tanto proveito nesta aplicação.

Relativamente à aplicação Web e à Web API, os argumentos anteriores já não se aplicariam neste caso. Decidiu-se aplicar uma arquitetura baseada em MVC, visto fornecer um baixo acoplamento entre a lógica de negócio e a representação visual. A decisão da arquitetura nesta aplicação pode também depender das ferramentas Web utilizadas, se for utilizada alguma, visto algumas delas favorecerem a utilização de uma determinada arquitetura.

4.1 Casos de uso

4.1.1 Requisitos funcionais

Existem quatro atores na aplicação - O utilizador não registado, o estudante/candidato à procura de emprego, o recrutador e o especialista de recrutamento.

Utilizador não registado

O utilizador não registado é um utilizador sem sessão iniciada. O utilizador não registado pode fazer o registo de uma conta.

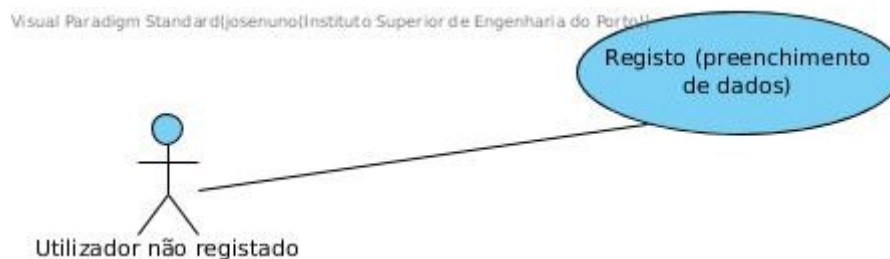


Figure 6: Utilizador não registado

Estudante/Candidato à procura de emprego

O estudante/candidato é um utilizador registado, com intenção de procurar emprego. O estudante/candidato pode visualizar quais as características que podem ser melhoradas e ver ofertas de emprego.

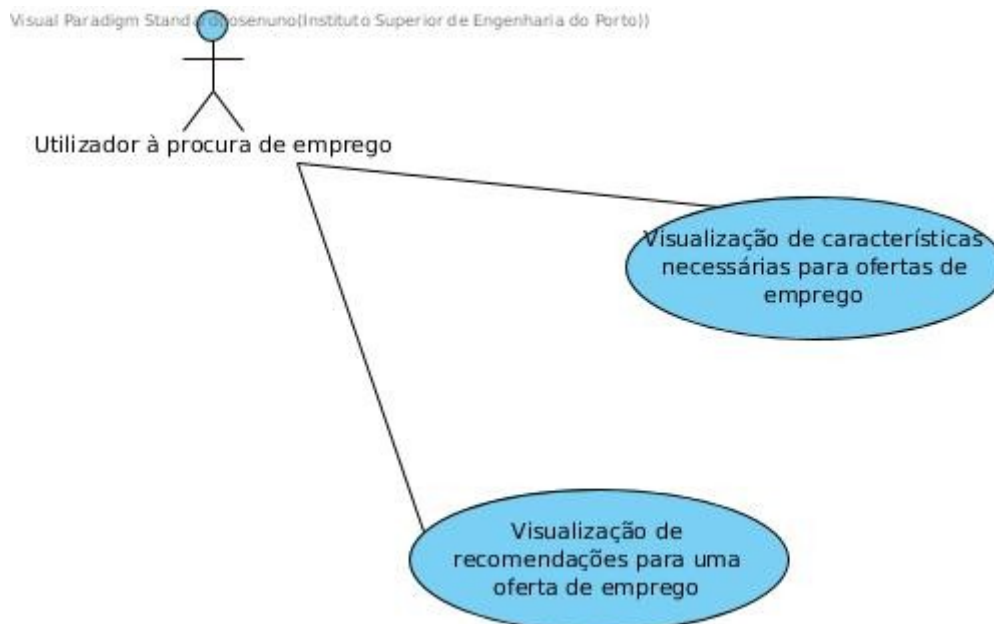


Figure 7: Estudante/Candidato à procura de emprego

Recrutador

O recrutador é um utilizador registado, com intenção de procurar candidatos às suas ofertas de emprego. O recrutador pode consultar os modelos de emprego existentes, criar ofertas de emprego, consultar as suas ofertas de emprego e consultar os candidatos que são mais adequados para uma determinada oferta de emprego.

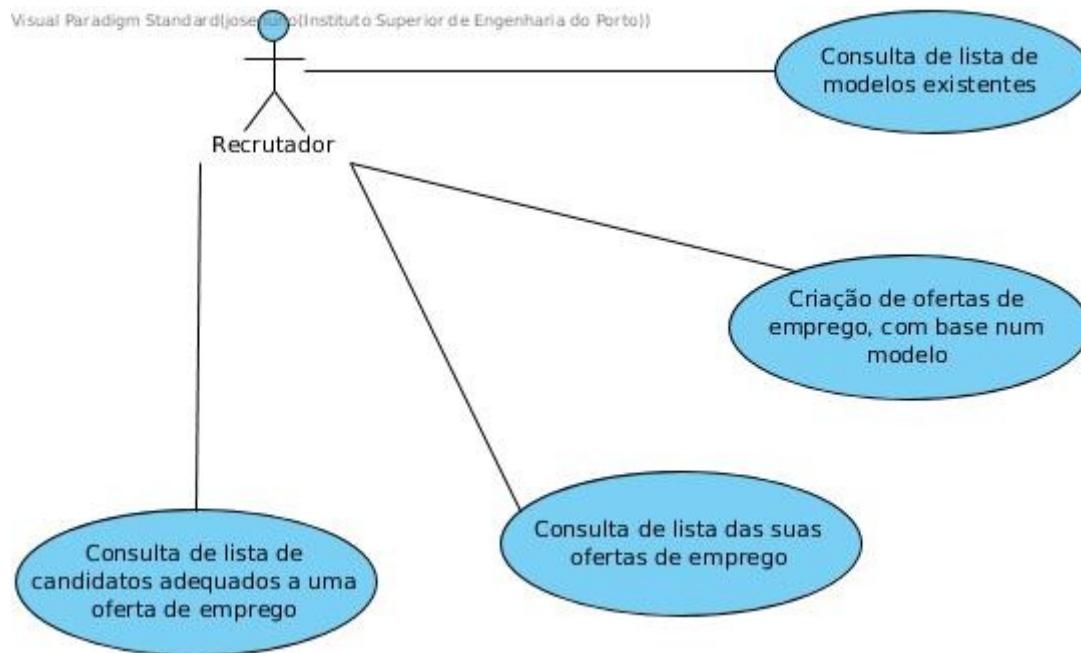


Figure 8: Recrutador

Especialista em Recrutamento

O especialista em recrutamento ou administrador é um utilizador registado, que faz a gestão dos modelos de emprego. O especialista em recrutamento pode criar, editar, remover e consultar modelos de emprego.

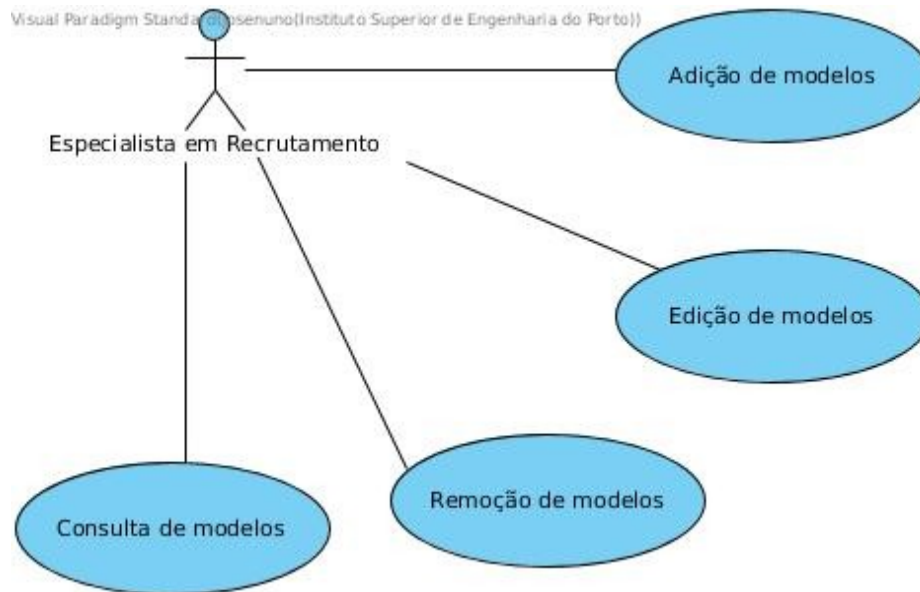


Figure 9: Especialista em Recrutamento

4.1.2 Interação entre utilizador e sistema

UC1 - Registrar Utilizador

O utilizador não registado pode registar-se, preenchendo dados como as suas credenciais. O sistema responde se o utilizador se registou com sucesso.

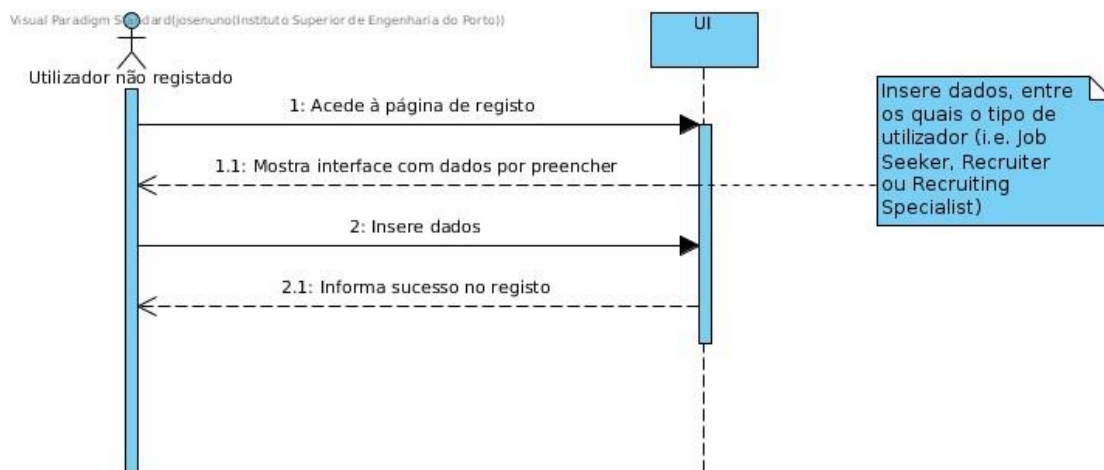


Figure 10: UC1 - Registrar

UC2 - Visualização de características necessárias para ofertas de emprego

O estudante/candidato pode visualizar as características desejadas de uma determinada oferta de emprego. É apresentada uma lista ao utilizador com as ofertas de emprego existentes. O utilizador seleciona uma e visualiza as características da oferta de emprego.

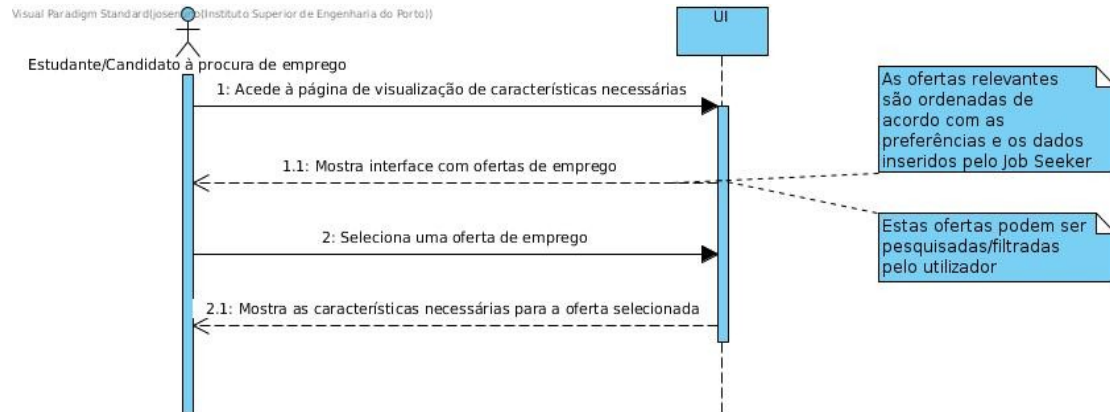


Figure 11: UC2 - Visualização de características necessárias para ofertas de emprego

UC3 - Visualização de recomendações de ofertas de emprego

O estudante/candidato pode visualizar as suas características, que possam ser melhoradas para uma determinada oferta de emprego. É apresentada uma lista ao utilizador com as ofertas de emprego existentes. O utilizador seleciona uma e visualiza as suas características que podem ser melhoradas.

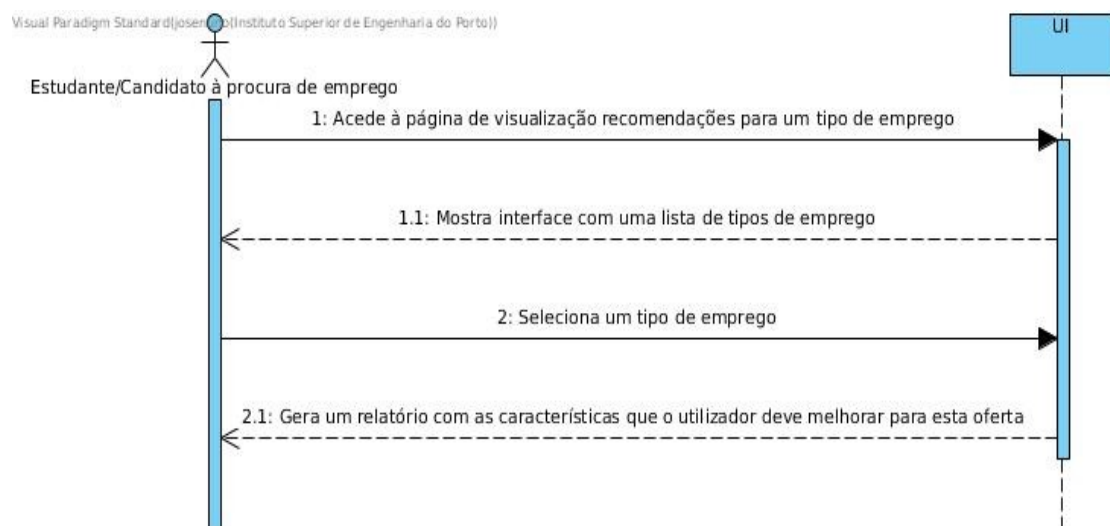


Figure 12: UC3 - Visualização de recomendações de ofertas de emprego

UC4 - Consulta de lista de modelos existentes

O recrutador pode consultar uma lista de modelos de emprego existentes. É apresentada uma lista ao utilizador com os modelos de emprego existentes. O utilizador seleciona uma e visualiza as características do modelo de emprego.

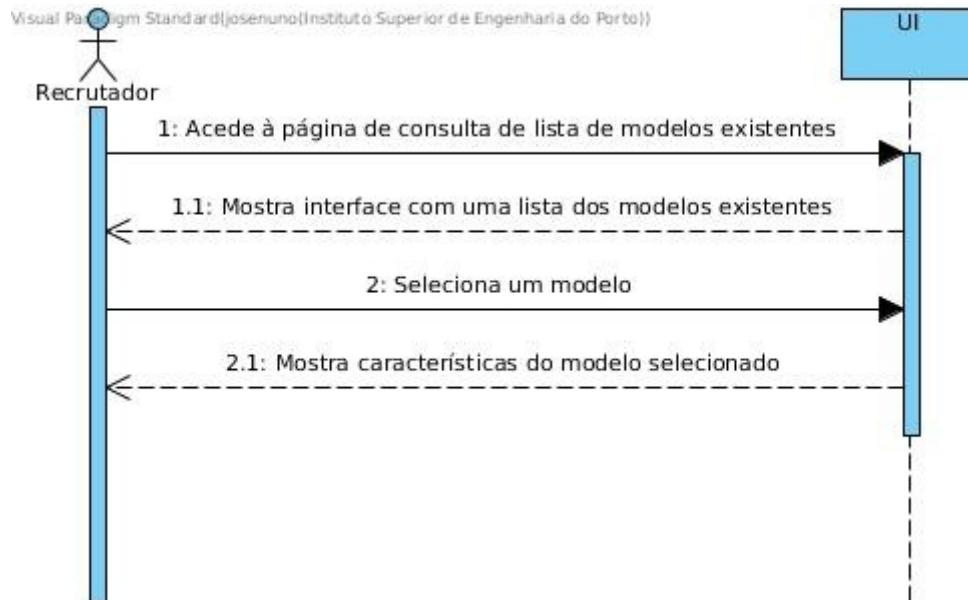


Figure 13: UC4 - Consulta de lista de modelos existentes

UC5 - Criação de ofertas de emprego com base num modelo

O recrutador pode criar ofertas de emprego, baseadas num modelo de emprego. É apresentada uma lista ao utilizador com os modelos de emprego existentes. O utilizador seleciona uma e são apresentadas as características desse modelo. O utilizador pode alterar algumas das características e/ou validar a criação da oferta de emprego.

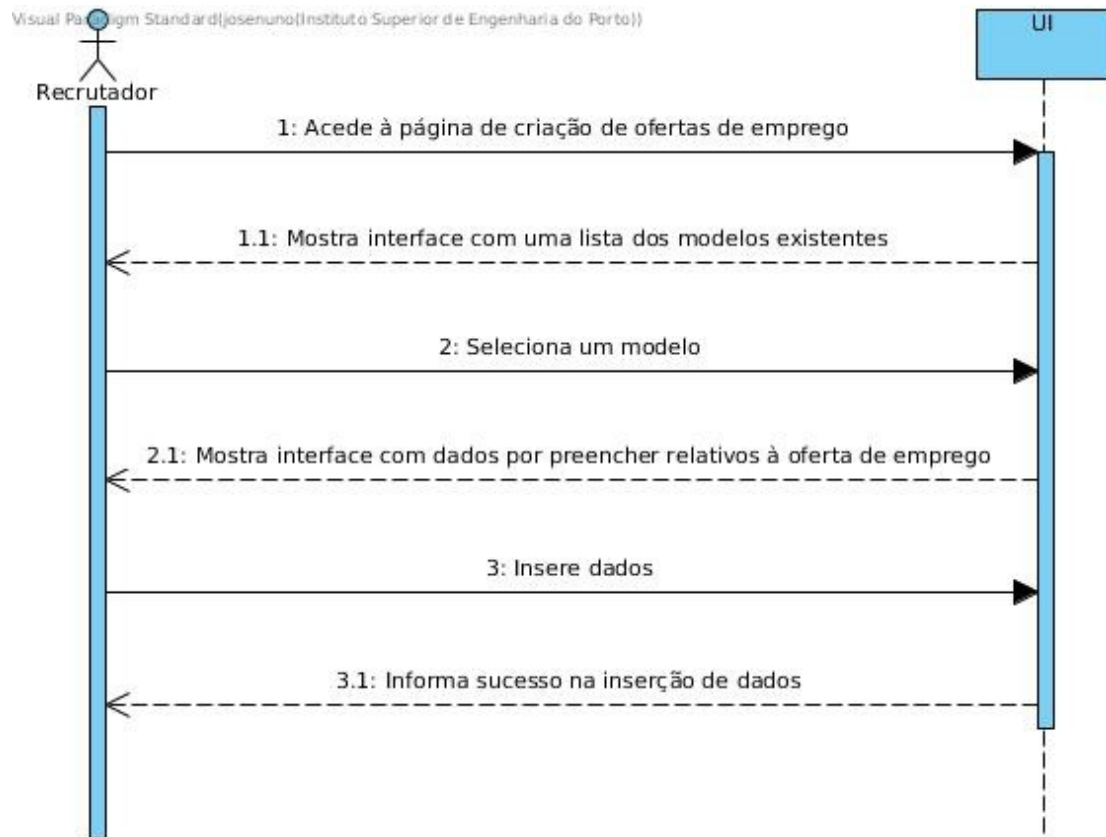


Figure 14: UC5 - Criação de ofertas de emprego com base num modelo

UC6 - Consulta de lista das suas ofertas de emprego

O recrutador pode consultar uma lista das suas ofertas de emprego. É apresentada uma lista ao utilizador com as suas ofertas de emprego, assim como as suas características.

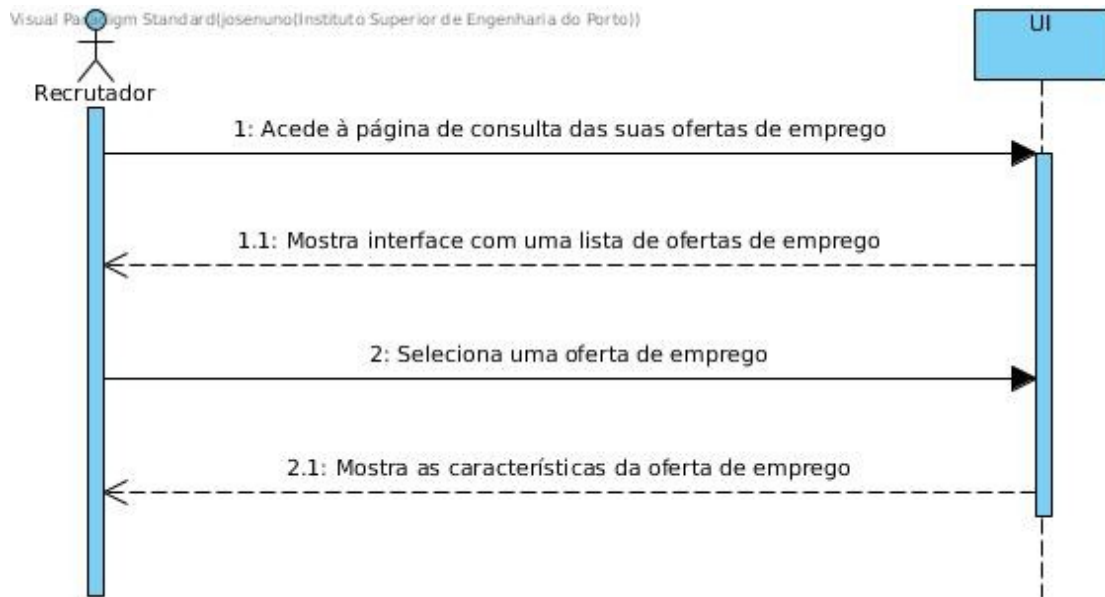


Figure 15: UC6 – Consulta de lista das suas ofertas de emprego

UC7 – Consulta de lista de candidatos adequados a uma oferta de emprego

O recrutador pode consultar uma lista dos candidatos mais adequados a uma oferta de emprego. É apresentada uma lista ao utilizador com as suas ofertas de emprego. O utilizador seleciona uma e é apresentada uma lista de candidatos ordenada de acordo com o quão adequado(a) é o(a) candidato(a).

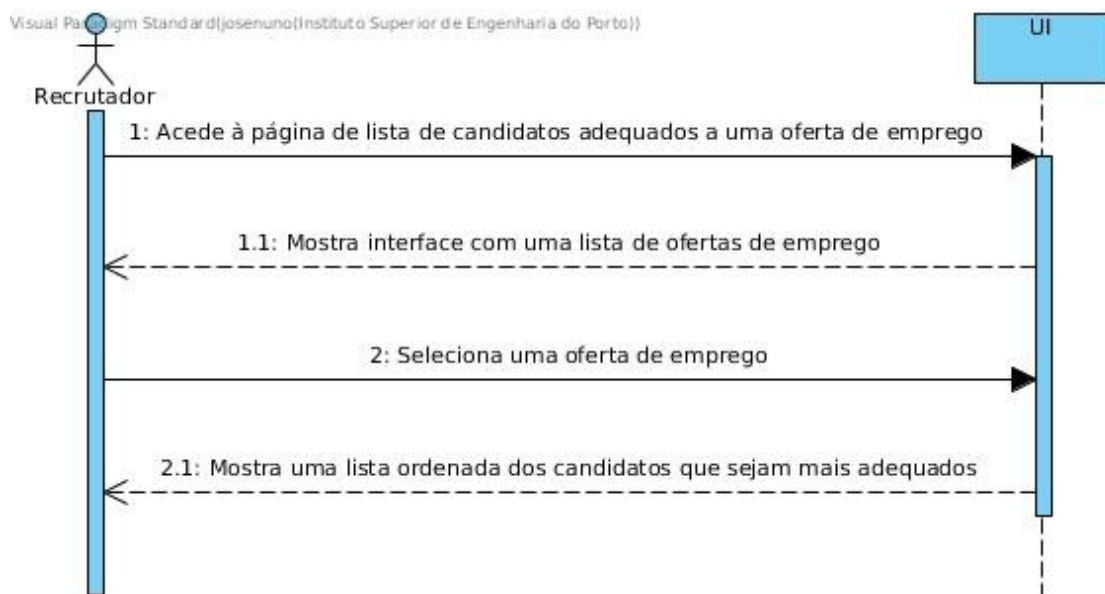


Figure 16: UC7 – Consulta de lista de candidatos adequados a uma oferta de emprego

UC8 - Adição de modelos

O especialista em recrutamento pode adicionar modelos de emprego. O utilizador insere as características do modelo de emprego. O sistema valida e adiciona o modelo de emprego.

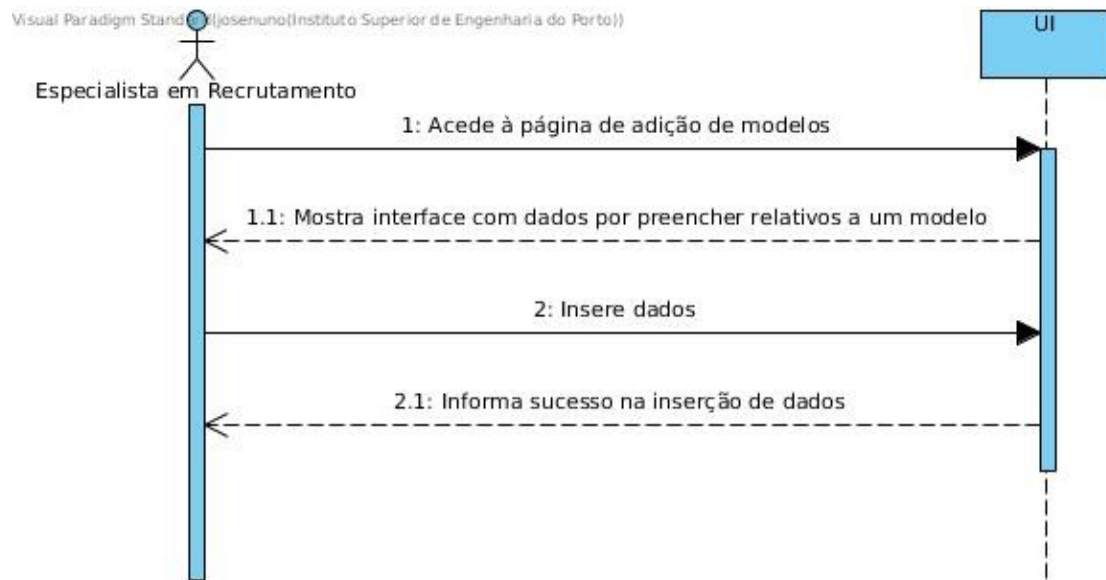


Figure 17: UC8 - Adição de modelos

UC9 - Edição de modelos

O especialista em recrutamento pode editar modelos de emprego. É apresentada uma lista ao utilizador com os modelos de emprego existentes. O utilizador seleciona um e edita as características que desejar. O sistema valida e modifica o modelo de emprego.

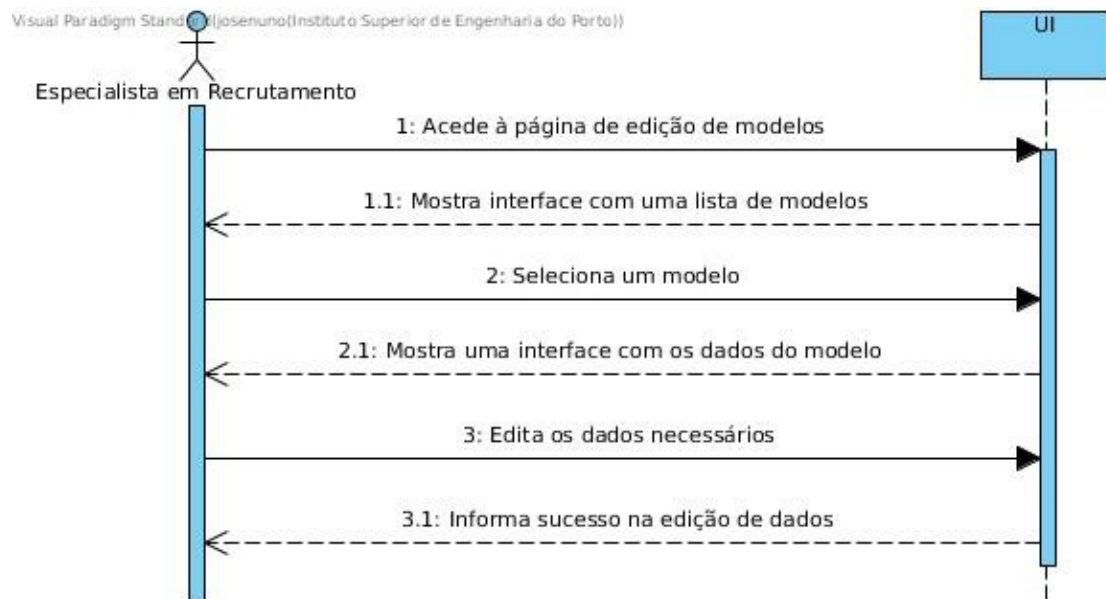


Figure 18: UC9 – Edição de modelos

UC10 – Consulta de modelos

O administrador pode consultar os modelos de emprego. É apresentada uma lista ao utilizador com os modelos de emprego existentes. O utilizador seleciona um e são apresentadas as características do modelo de emprego selecionado.

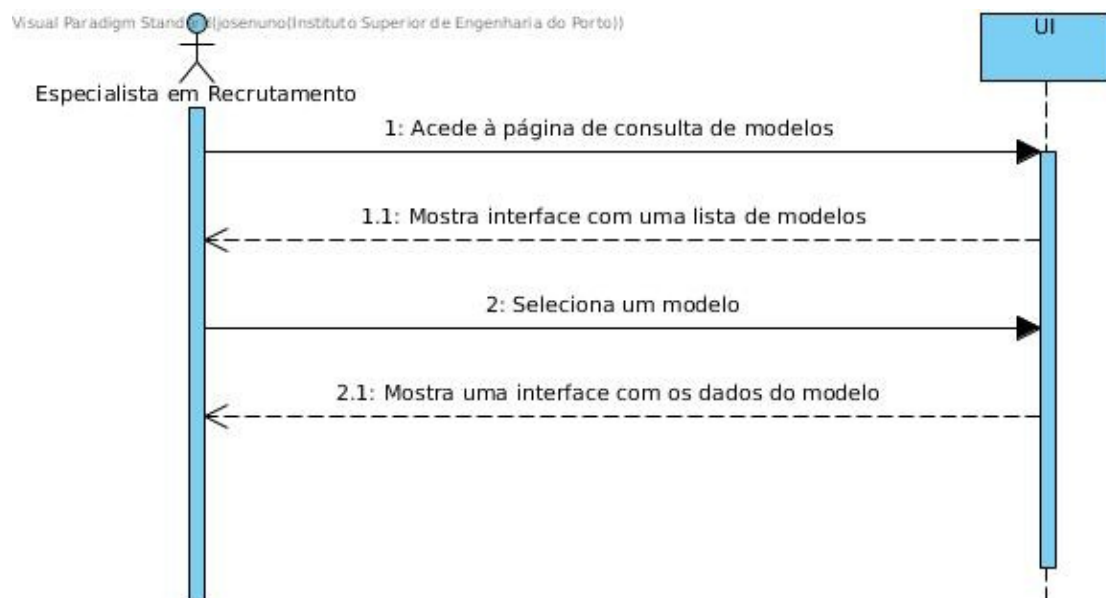


Figure 19: UC10 – Consulta de modelos

UC11 - Remoção de modelos

O especialista em recrutamento pode remover modelos de emprego. É apresentada uma lista ao utilizador com os modelos de emprego existentes. O utilizador seleciona um e o sistema remove o modelo de emprego.

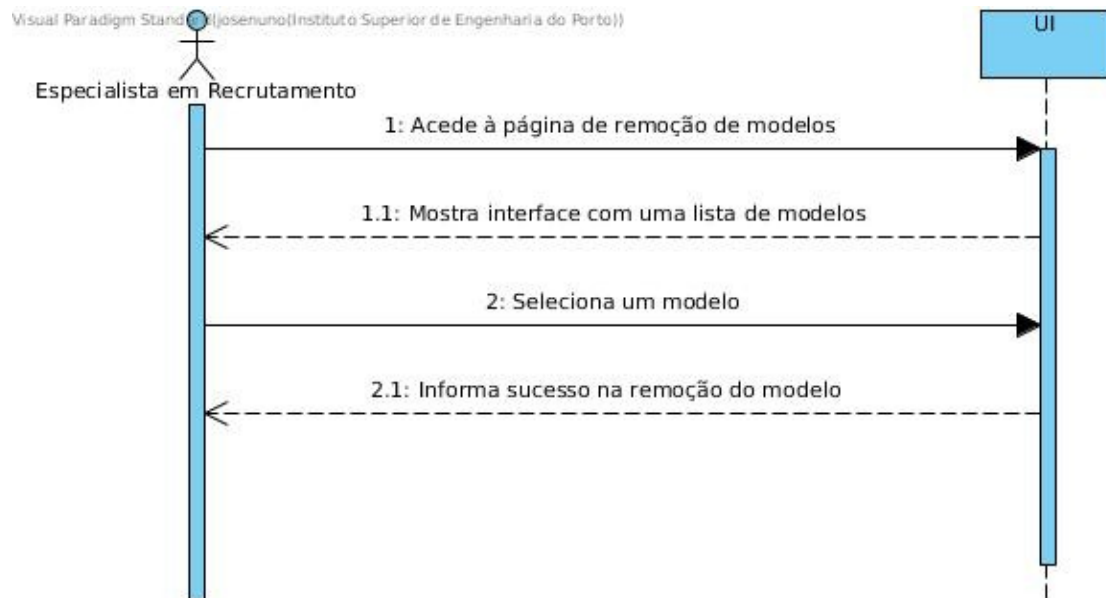


Figure 20: UC11 - Remoção de modelos

4.2 Modelo de domínio

O diagrama da figura seguinte representa o modelo de domínio definido para o Profile Matcher.

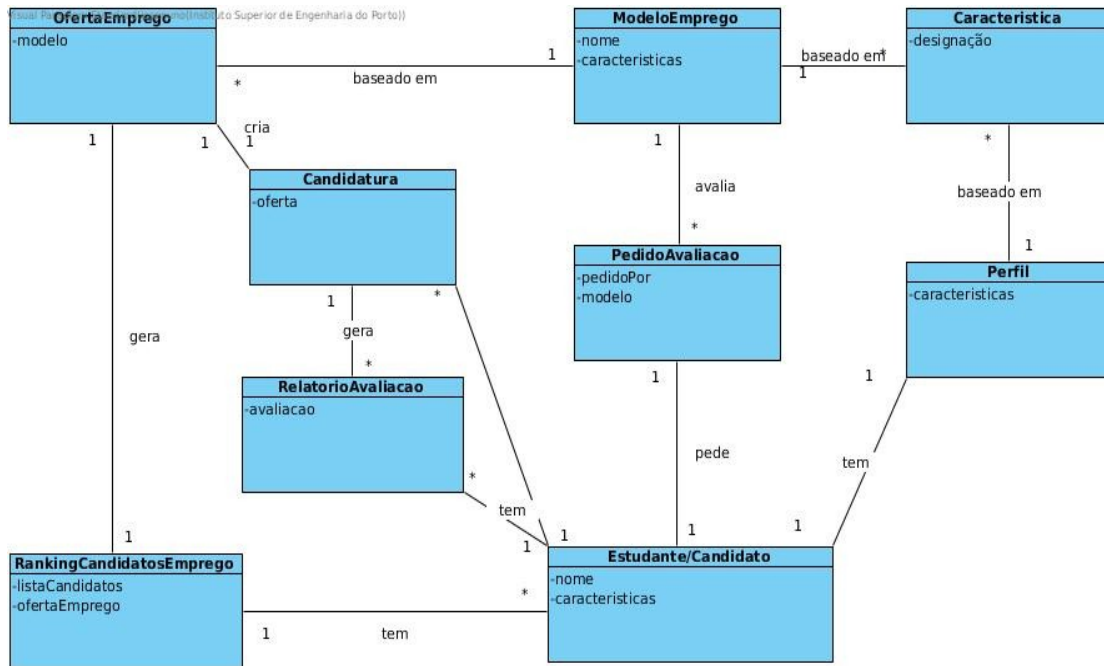


Figure 21: Modelo de Domínio

- Estudante/Candidato – Entidade que representa um estudante ou um candidato; Tem um perfil associado; Pode pedir uma avaliação do seu perfil, para um determinado modelo; Tem várias candidaturas a emprego; Pode pedir uma avaliação do seu perfil, para uma determinada candidatura.
- Perfil – Entidade que representa o perfil de um estudante ou um candidato; É baseado em várias características.
- Característica – Entidade que representa uma competência ou um outro tipo de dado de um indivíduo.
- ModeloEmprego – Entidade que representa um modelo para um emprego; É baseado em várias características de um perfil ideal para o modelo.
- OfertaEmprego – Entidade que representa uma oferta de emprego; É baseada num modelo de Emprego.
- RankingCandidatosEmprego – Entidade que representa uma lista ordenada dos candidatos mais adequados para uma oferta de emprego.
- RelatorioAvaliacao – Entidade que representa um relatório com recomendações de características a melhorar de um candidato, para uma dada candidatura.

- PedidoAvaliacao – Entidade que representa um pedido de uma avaliação das características de um candidato, para um dado modelo de emprego.
- Candidatura – Entidade que representa uma candidatura feita por um candidato a uma determinada oferta de emprego.

4.3 Modelo de dados

O diagrama da figura seguinte representa um modelo de dados, a ser utilizado para a estrutura da base de dados. O modelo espelha o modelo de domínio.

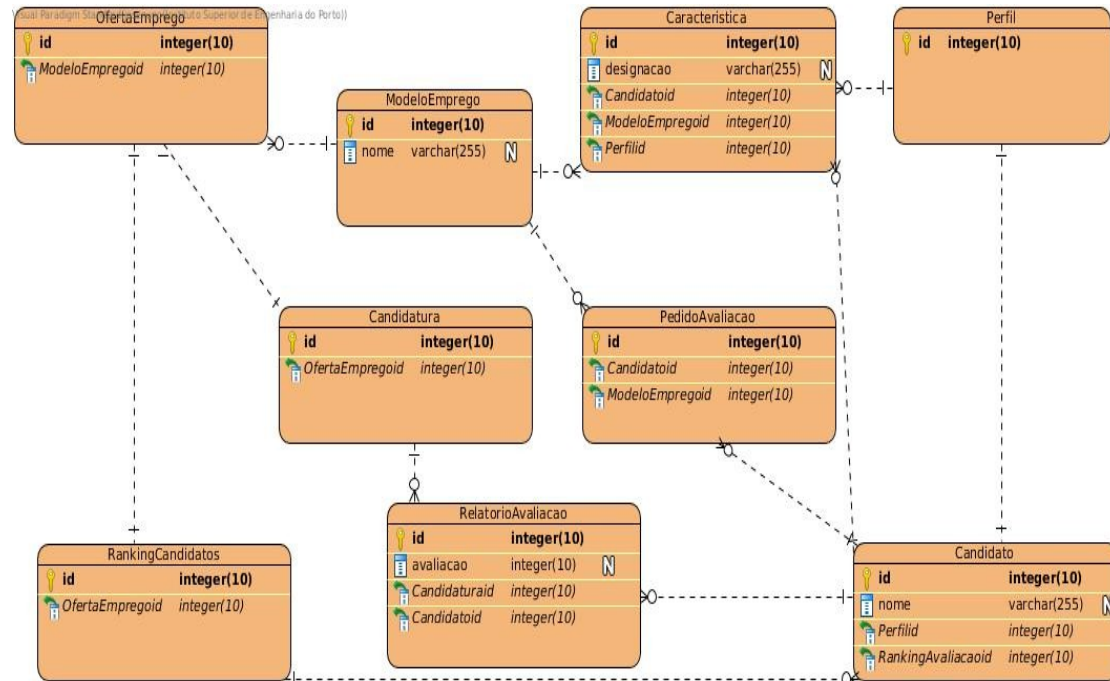


Figure 22: Modelo de dados

4.4 Arquitetura de software

O seguinte diagrama de componentes representa as integrações entre os sistemas, numa perspetiva de alto nível.

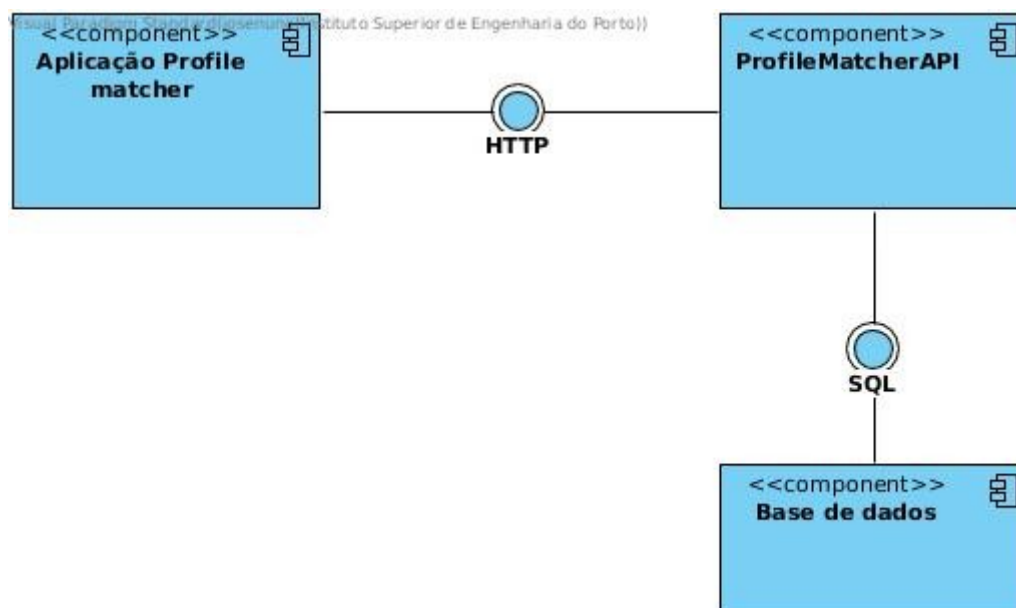


Figure 23: Diagrama de componentes

A aplicação "Profile Matcher", vai ser a aplicação que os utilizadores vão utilizar diretamente. É uma aplicação Web, e por isso deverá ser acedida através de um Web Browser.

Esta aplicação consome o "ProfileMatcherAPI", que é uma Web API. Todas as ações Create-Read-Update-Delete (CRUD) são pedidas pela aplicação Web, à Web API. Visto estes dois componentes estarem separados fisicamente, esta comunicação é feita através de HTTP.

O ProfileMatcherAPI serve como ponte entre o Profile Matcher e a Base de dados. Recebe os pedidos feitos pelos utilizadores, através do Profile Matcher, e valida os mesmos. Dependendo do tipo do pedido, e assumindo que o pedido seja válido, esta aplicação faz alterações nos registos da Base de dados.

A Base de dados, pelo menos numa fase inicial, vai ser instalada no mesmo servidor que contém o ProfileMatcherAPI. Assim sendo, esta comunicação é feita via TCP localmente.

5 Implementação

Este capítulo refere o trabalho realizado no desenvolvimento das aplicações desenhadas na secção anterior. As três secções seguintes mostram o processo da implementação de cada uma das aplicações.

5.1 Aplicação para Matching

O objetivo desta aplicação é o de identificar o quão adequado um candidato é para uma determinada oferta de emprego. Para tal, a aplicação deve obter dados relevantes do perfil do candidato, assim como da oferta de emprego em questão, de modo que seja possível aplicar um algoritmo adequado para fazer o *matching* entre ambos.

5.1.1 Definição de modelo

Existe uma grande diversidade na estrutura e na informação que é colocada nas ofertas de emprego, assim como nos currículos dos candidatos. Isto dificulta o processo de analisar a informação fornecida pelo utilizador. Com isto em conta, decidiu-se moldar um modelo genérico que abstraia essa informação. Este modelo teria várias categorias (e.g. Experiência profissional, educação, etc.) e subcategorias de características consideradas relevantes, e que habitualmente são publicadas nas ofertas de emprego, assim como nos currículos de candidatos.

Este modelo deve ser o mais genérico possível, de modo a ser adaptável para diversas áreas de negócio diferentes. Assim sendo, e após a análise de várias ofertas de emprego e de currículos, ficou definido o modelo representado na tabela seguinte.

Table 11: Modelo de emprego

Categorias	Subcategorias
Work experience	Field Duration Role Responsibilities/tasks
Education	Study field Year Degree level
Hard skills	Knowledge on Experience with Communication skills
Soft skills	Team skills Leadership skills Organizational skills Interpersonal skills
Certifications	Field

Desta forma, as características do candidato e da oferta de emprego seriam adaptadas conforme este modelo, o que facilitaria a identificação do quão adequado é um candidato a uma oferta de emprego.

5.1.2 Doc2Vec

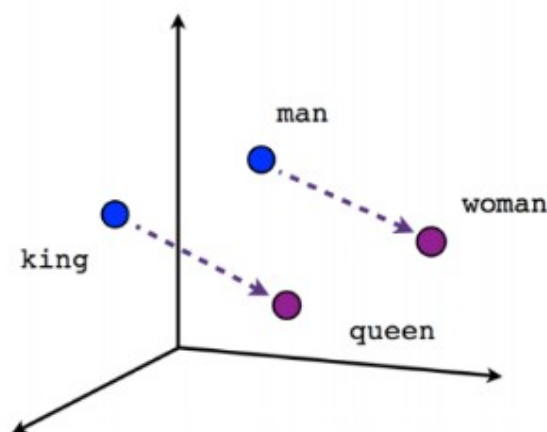
Após a definição do modelo descrito na secção anterior, pretende-se comparar duas instâncias desse modelo, uma instância para um candidato e outra para uma oferta de emprego, e concluir o quão semelhantes são.

Uma possível solução seria comparar as categorias de ambas as instâncias, frase a frase e calcular a frequência de frases ou de palavras iguais. No entanto, esta hipótese não seria viável, pois duas palavras que fossem sinónimas ou semelhantes, seriam identificadas como se fossem totalmente diferentes.

Uma solução mais adequada, seria ter em conta o quão semelhantes são as frases e assim, calcular a média do grau de semelhança de todos os pares de frases. Um valor mais elevado significaria que o candidato tinha maior probabilidade de ser adequado para uma oferta de emprego.

Para tal, decidiu-se utilizar uma técnica de processamento de linguagem natural, chamada Doc2Vec, baseada noutra técnica Word2Vec, ambos criados por equipas de investigação da Google[30].

A técnica Word2Vec consiste em gerar vetores que representem palavras. O objetivo é gerar estes vetores, de modo a que a distância entre eles, represente a semelhança das palavras[31]. Por exemplo, um vetor que represente a palavra “man” deverá ter uma distância relativamente curta a outro vetor que represente a palavra “woman”. Por outro lado, um vetor “king”, deverá estar mais distante de um vetor “woman” e ao mesmo tempo próximo de “man”. A imagem seguinte demonstra este exemplo numa representação gráfica com três dimensões.



Male-Female

Figure 24: Exemplo gráfico de Word2Vec[32]

Por norma, aplicações reais de Word2Vec são representados em dezenas ou centenas de dimensões. Neste exemplo os vetores são representados em apenas três dimensões, por ter um vocabulário de apenas quatro palavras, e por ser mais facilmente visualizado graficamente.

Não se sabe inicialmente o quão semelhantes as palavras são. É por isso necessário treinar previamente um modelo de Word2Vec através de uma rede neuronal. Pretende-se com este treino ajustar as posições dos vetores, de modo a que a distância vetorial entre eles indique a semelhança das palavras que esses mesmos vetores representam.

Este treino ao modelo de Word2Vec é feito fornecendo uma lista de frases, relacionadas com ofertas de emprego, visto neste caso em particular, fazer mais sentido descobrir relações entre palavras deste tema. A tabela seguinte contém um excerto das características recolhidas, para a subcategoria *communication skills*.

Table 12: Excerto de características (communication skills)

Answering Telephones
Business Correspondence
Calling Clients
Client Relations
Communication
Correspondence
Customer Service
Directing Clients
(...)

Foi aplicado o *Skip-Gram Model*[33]. Este algoritmo, dada uma palavra de uma frase, tenta prever as palavras vizinhas na mesma frase. Isto significa que o modelo vai ser ajustado de modo a que esta previsão seja a mais correta possível.

A técnica Doc2Vec é semelhante ao Word2Vec. Para além dos vetores que representem cada uma das palavras de uma frase, nesta técnica também é considerado mais um vetor. Este vetor representa uma identificação única do parágrafo onde se insere a frase[30].

Para o treino, foi utilizado o Tensorflow, conforme indicado na secção “Avaliar soluções existentes”. Adicionalmente foi utilizada uma biblioteca em Python, chamada “gensim”[34]. Esta biblioteca facilita a utilização de alguns algoritmos relacionados com Natural Language Processing (NLP), entre os quais o Doc2Vec. Para o efeito, e numa tentativa de melhorar a qualidade do treino, foram

descartadas as palavras que ocorram menos de 5 vezes na lista de frases anteriormente referida. Neste treino, foram feitas 20 iterações a esta lista de frases e aplicado um *learning rate* de 0.025, que vai decaindo à medida que são feitas as iterações. Isto significa que o modelo está mais propenso a ser modificado nas iterações iniciais, quando desconhece a relação entre as palavras, do que nas finais.

5.1.3 Extração de informação

A lista de frases anterior foi obtida através de ofertas de emprego e de currículos disponibilizados online. Esta informação foi obtida através de *Web scrapping* (i.e. extração de dados de websites com recurso a ferramentas automáticas [35]), Web API públicas ou manualmente.

Foi necessário ter um cuidado especial na extração de informação via *Web scrapping*. Várias páginas Web não permitem que informação (total ou parcialmente) seja extraída com ferramentas automáticas. Para além deste ponto poder estar descrito nos termos e condições da página, existe também a convenção de disponibilizar um ficheiro “robots.txt”. Este ficheiro serve precisamente para indicar o que é ou não permitido extrair automaticamente na página Web em questão. Toda a informação recolhida via *Web Scrapping* foi obtida respeitando estes pontos.

O *Web Scrapping* foi feito com recurso a duas bibliotecas de Python: *urllib*[36] e *lxml*[37]. A primeira biblioteca serviu para descarregar conteúdo de páginas, enquanto que a segunda serviu para filtrar esse conteúdo, com recurso à linguagem *XPath*. Posteriormente, foi feita uma revisão manual na informação obtida, para descartar, o melhor possível, a informação que não era relevante.

As páginas Web mais utilizadas para este processo foram o [hiring monster](#) e o [Indeed](#)

5.2 Web API

Esta aplicação tem como finalidade servir de “ponte” entre a aplicação Web – Profile Matcher, e a aplicação para *matching*. Para além desta ligação entre as duas aplicações, também será utilizada para fazer login/registo de contas, assim como a gestão do armazenamento de dados dos utilizadores.

Foi decidido usar o “Node.js”[38] para o desenvolvimento da Web API e MongoDB[39] para gestão e armazenamento de base de dados. Os motivos destas escolhas basearam-se na familiaridade pessoal nestas ferramentas e na

boa interoperabilidade entre elas, visto ambas serem programadas em Javascript.

Nesta API foram desenvolvidos *endpoints* com o objetivo de serem consumidos pela aplicação Web, descrita no capítulo seguinte. Antes de referir quais os *endpoints* desenvolvidos, é relevante demonstrar como foi implementada a autenticação nesta aplicação.

5.2.1 Autenticação e segurança

Foi definido um sistema de autenticação, de modo a poder delegar a cada utilizador que endpoints tem ou não acesso na Web API, e conseqüentemente ao que tem acesso no Profile Matcher.

Tendo em conta que os modelos de emprego devem ser geridos apenas por utilizadores especialistas na área, decidiu-se fazer a distinção entre utilizadores regulares e administradores (i.e. utilizadores especializados em áreas de negócio). Apenas os administradores teriam a possibilidade de gerir os modelos de emprego, assim como de definir quais os utilizadores que são administradores.

Concluiu-se também a necessidade da possibilidade de navegar na aplicação Web sem estar registado, para possibilitar o registo ou para iniciar sessão.

Ainda foi ponderada a possibilidade de adicionar mais um tipo de utilizador, para haver a distinção entre empregadores e candidatos a emprego. No entanto, visto haver a possibilidade de eventualmente um candidato pretender criar uma oferta de emprego, ou um empregador de ser candidato, decidiu-se não fazer esta distinção.

Concluindo, existem três papéis que um utilizador pode ter: “Administrador”, “Utilizador regular” e “Não registado”.

Um dos pontos fulcrais na segurança informática, é que a informação transmitida garanta confidencialidade, autenticidade e integridade. Com isto em mente, a comunicação entre o cliente e o servidor é feita com mensagens encriptadas, via SSL.

Adicionalmente, a autenticação dos pedidos é feita com recurso a JWT tokens. Esta norma garante transmissões seguras de informação em objetos JSON[40]. O utilizador inicialmente inicia sessão e recebe um token que deverá ser utilizado nos pedidos seguintes, que requeiram autenticação.

O token consiste em três secções: *Header*, *Payload* e *Signature*.

Extração de texto

Em primeiro lugar, o currículo fornecido pelo utilizador (em formato pdf), é lido com o auxílio da ferramenta `npm-pdfreader`[41]. Esta ferramenta lê o texto do documento, linha a linha, assim como a posição de cada palavra lida.

Divisão do texto

Depois de obtido o texto do documento, o mesmo é dividido em diversas categorias (e.g. competências, informação pessoal, etc.). Este passo foi feito com recurso a técnicas de aprendizagem automática. Adicionalmente, foi aproveitado o facto de existirem ferramentas *open-source* que resolvem parcialmente este problema.

Foram utilizadas as ferramentas `ResumeParser`[42] e `Pyrespaser`[43]. Ambas transformam o conteúdo de um currículo, num objeto JSON, dividido em várias categorias. As categorias mais relevantes dessas ferramentas para o Profile Matcher são: *gender, e-mail, work experience, skills* e *education and training*.

Algumas das categorias referidas coincidem com as do modelo descrito na Figura 62. Nestes casos, as características relativas a estas categorias podem ser atribuídas diretamente no modelo. Noutros casos, é necessário fazer algumas adaptações após esta extração, de modo a poderem ser utilizadas no modelo.

Adaptação das categorias

Resta adaptar as categorias que não são equivalentes ao modelo. Um dos casos é a categoria *skills*, que deve ser dividida nas subcategorias de *hard skills* e *soft skills* (categorias que existem no modelo).

A adaptação das categorias são feitas com base no modelo `Doc2Vec`, referido na secção “Aplicação para *matching*”.

A categoria da característica que tenha um maior grau de semelhança, é assumida ser a categoria mais adequada. Por exemplo, se uma dada

característica obtida no processo de divisão de texto tiver um maior grau de semelhança relativamente a uma característica da subcategoria *leadership skills*, então assume-se que essa característica pertença a essa subcategoria.

5.2.3 Endpoints

Os *endpoints* desenvolvidos são descritos na tabela seguinte.

Table 13: Endpoints da Web API

Endpoint	Método(s)	Descrição	Autenticação
/characteristics	GET, POST	Gestão das características do perfil de um utilizador	Regular
/models	GET, POST, PUT, DELETE	Gestão dos modelos de ofertas de emprego	Administrador
/offers	GET, POST	Gestão de ofertas de emprego	Regular
/offers/model/{model_id}	GET	Devolve as ofertas de emprego baseadas num determinado modelo de emprego	Regular
/offers/bestCandidates	GET	Devolve os melhores candidatos de uma oferta de emprego	Regular
/offers/recommendedOffers	GET	Devolve as ofertas de emprego recomendadas para um utilizador	Regular
/offers/recommendedModels	GET	Devolve os modelos de emprego recomendados para um utilizador	Regular
/profile	POST	Criação de um perfil através do envio de um CV	Regular
/recommended_skills/model	GET	Devolve as competências de um utilizador a melhorar, para um modelo de emprego	Regular

/recommended_skills/ offer	GET	Devolve as competências de um utilizador a melhorar, para uma oferta de emprego	Regular
/settings	GET, POST	Gestão das definições de utilizadores	Regular
/users	GET	Devolve todos os utilizadores registados	Administrador
/users/validate_token/ {token}	GET	Devolve se um dado JWT token, é válido	Não registado
/users/toggle_admin/ {user_id}	POST	Atribui ou remove o papel de Administrador a um utilizador	Administrador
/users/register	POST	Registo de um utilizador (será utilizador regular inicialmente)	Não registado
/users/login	POST	Início de sessão. Devolve um JWT Token.	Não registado

Os *endpoints* foram implementados tendo em conta a utilização de JWT tokens, conforme descrito na secção anterior. Os *endpoints* que não requeiram o registo do utilizador ignoram o token. Os outros validam a sua autenticação. No caso do pedido requerer que o utilizador seja administrador, tal é verificado segundo a autenticação que o utilizador tenha na base de dados. O diagrama de sequência seguinte demonstra este processo, no *endpoint* "users/toggle_admin".

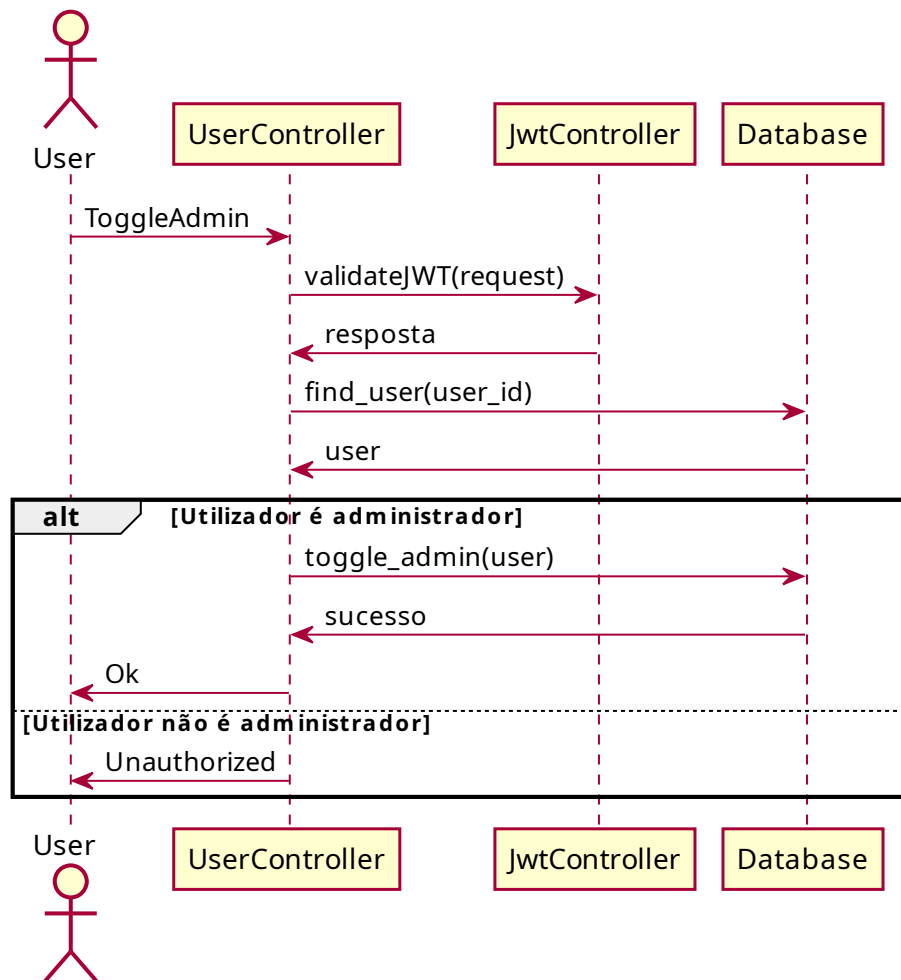


Figure 26: Toggle Admin - fluxo de sequência

5.3 Aplicação Web – Profile Matcher

Por fim, o Profile Matcher vai tirar partido da Web API desenvolvida e consequentemente da aplicação de matching. Vai ser esta a aplicação que os utilizadores vão utilizar diretamente.

Foi decidido usar o “React.js”[44] para o desenvolvimento desta aplicação. Tal como com a Web API, os motivos destas escolhas basearam-se na familiaridade pessoal nesta ferramenta e na boa interoperabilidade entre esta e o “Node.js”. Adicionalmente foi utilizada a biblioteca “reactstrap”, que permite ter componentes do “bootstrap” em React[45].

5.3.1 Interface

A primeira página é a de início de sessão e de registo, como se pode ver na imagem seguinte.

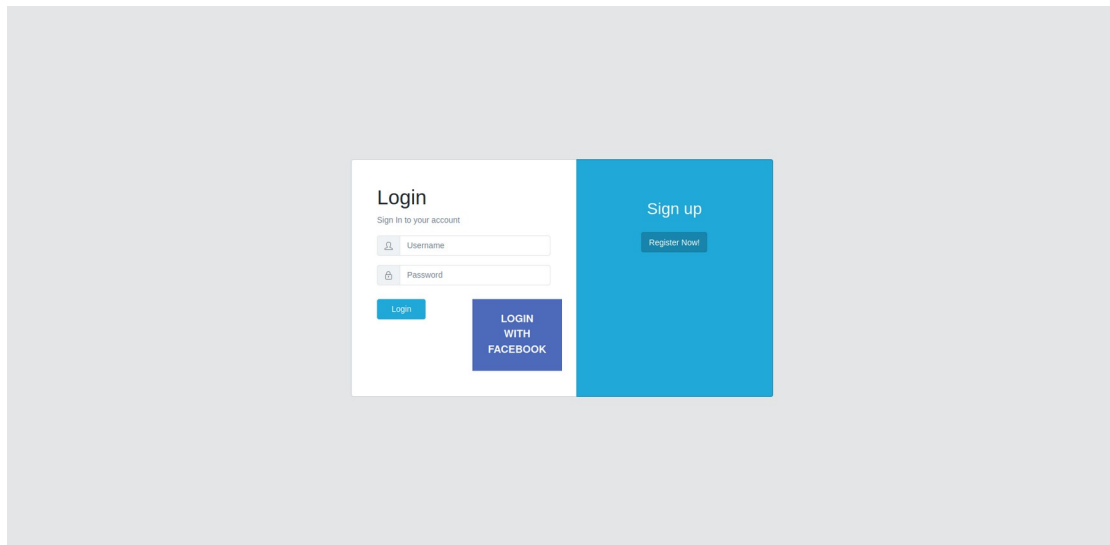


Figure 27: Login/Registo

Os pedidos feitos entre a aplicação Web e a Web API são encriptados, via SSL, com o fim de garantir a confidencialidade das credenciais do utilizador.

Após o início de sessão, é apresentada a página demonstrada na imagem seguinte.

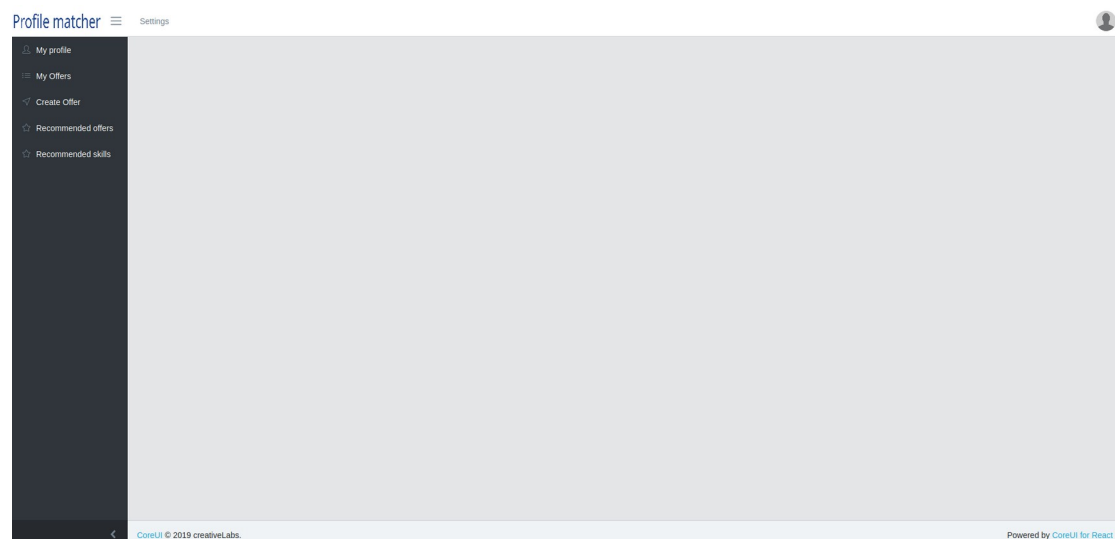


Figure 28: Menu

As funcionalidades da aplicação são acedidas na barra lateral do lado esquerdo. Algumas das funcionalidades podem estar ocultas, dependendo se o utilizador é

administrador ou não (e.g. A gestão de modelos de emprego não aparece na última imagem, visto o utilizador não ser administrador). A imagem seguinte mostra o mesmo menu, com sessão iniciada como administrador.

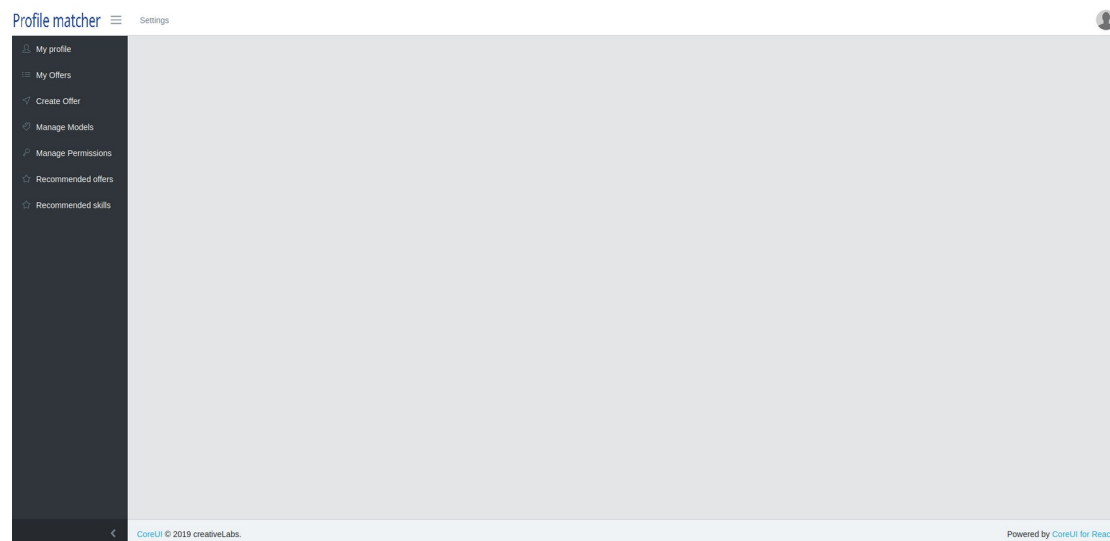


Figure 29: Menu (como administrador)

A primeira funcionalidade – “My Profile”, permite consultar e editar o perfil do utilizador. Por defeito este perfil vem vazio, pelo que deve ser preenchido manualmente e/ou automaticamente através da leitura de um CV. A imagem seguinte mostra essa página.

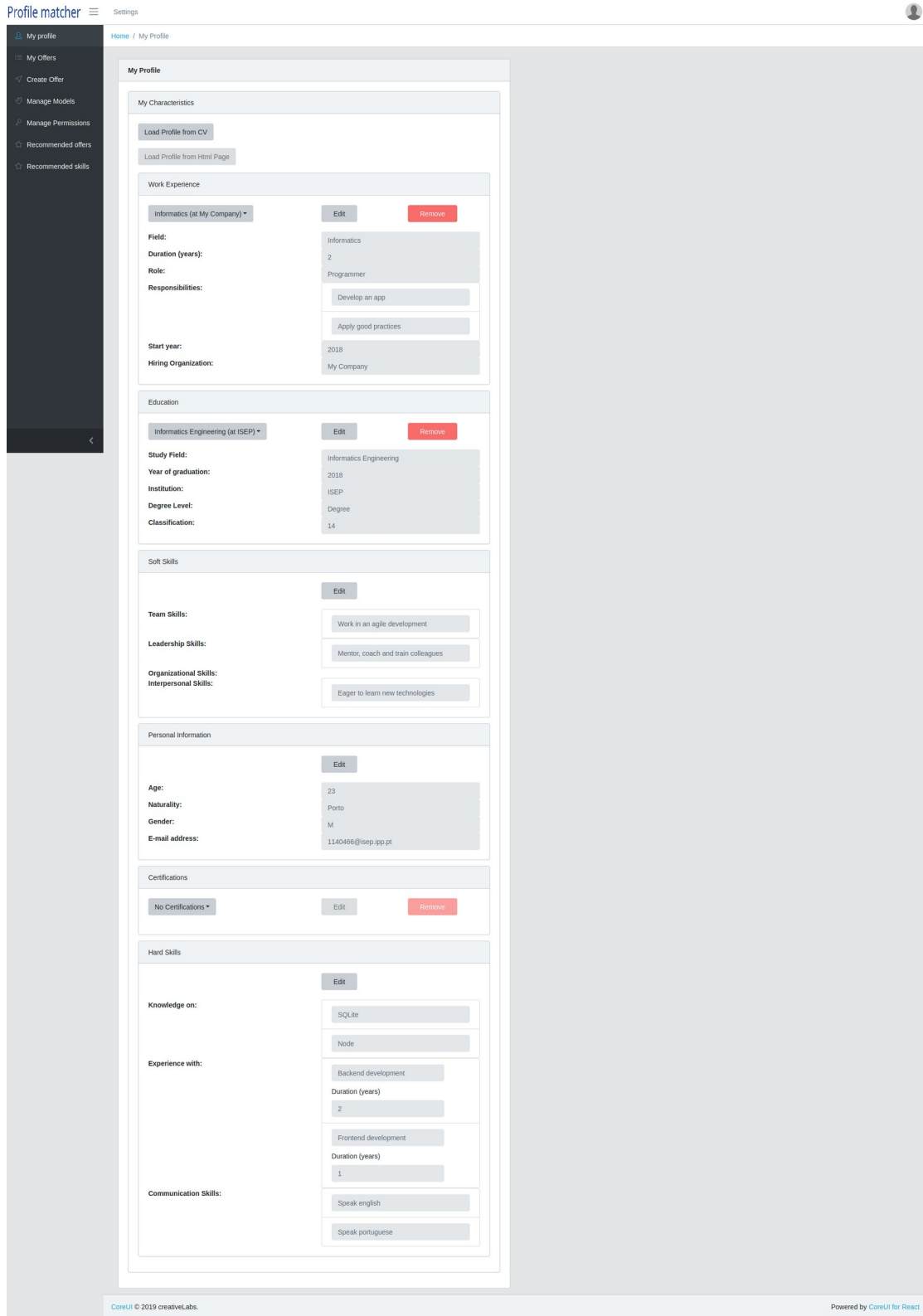


Figure 30: Perfil de utilizador

A funcionalidade seguinte – “My Offers”, permite visualizar as suas ofertas, assim como os candidatos mais adequados para cada uma das ofertas de emprego. É possível comparar graficamente dois ou mais candidatos. A imagem seguinte mostra essa funcionalidade.

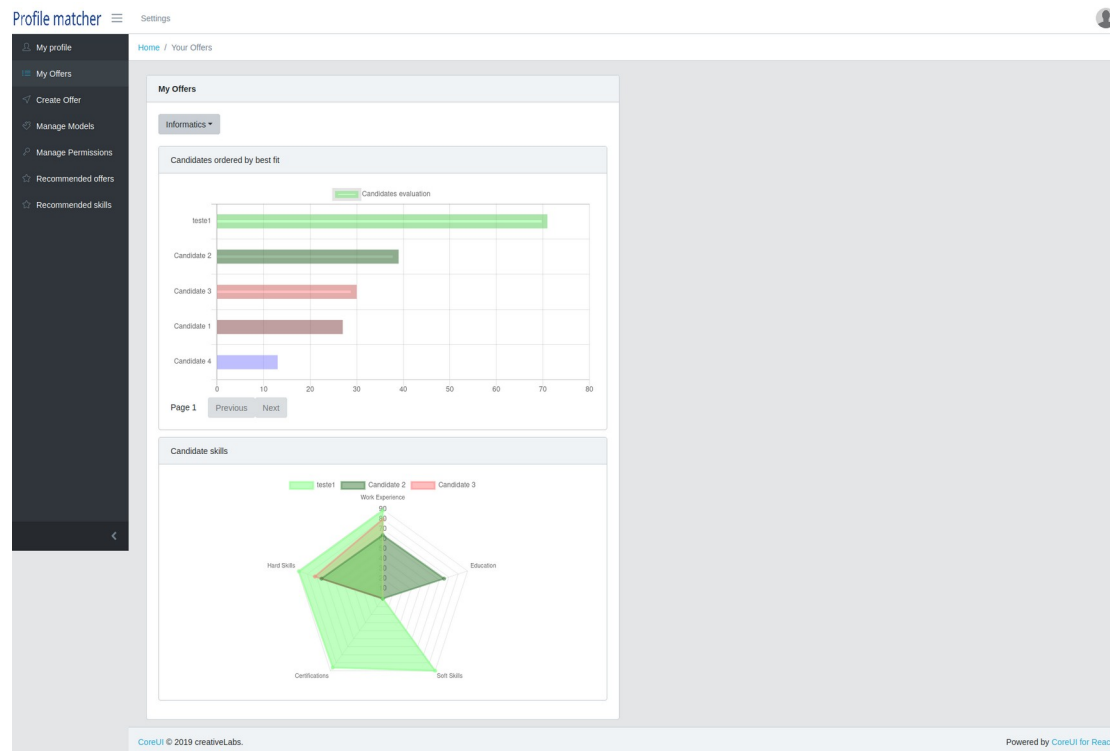


Figure 31: Ofertas de emprego

A funcionalidade seguinte é “Create Offer”. Permite criar ofertas de emprego, com base num modelo de emprego. Inicialmente o utilizador escolhe um modelo de emprego, a partir de uma lista, e de seguida preenche os dados necessários para a criação da oferta, como demonstrado na imagem seguinte.

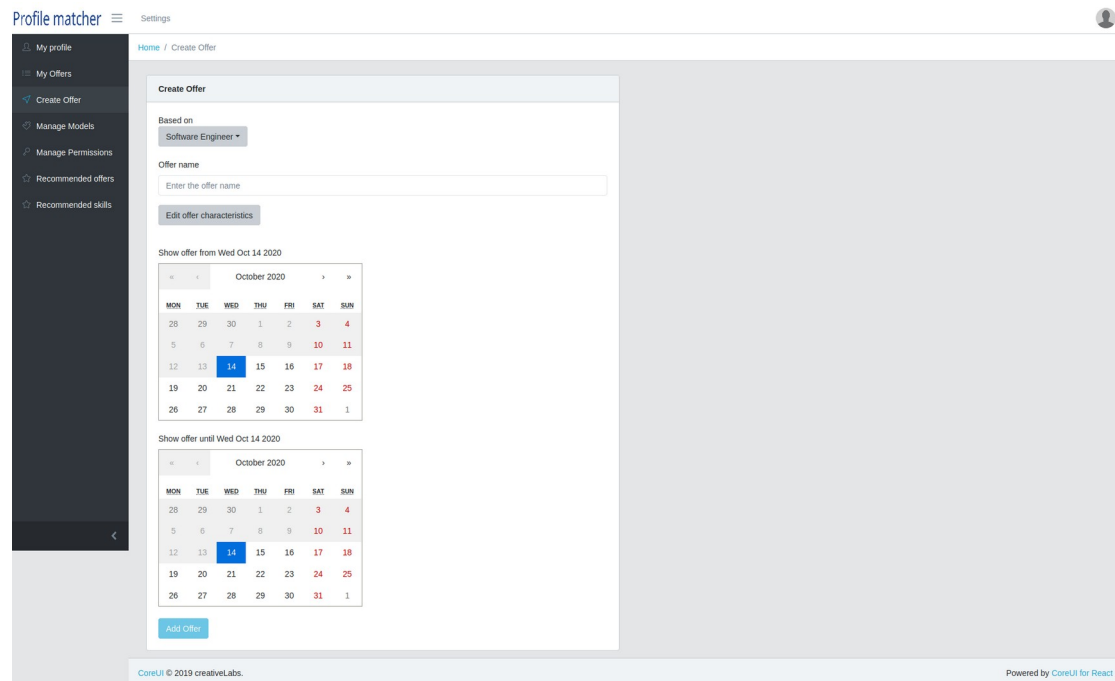


Figure 32: Criação de oferta de emprego

A funcionalidade “Manage Models” permite criar, editar, remover e consultar os modelos de emprego. A imagem seguinte demonstra a criação de um modelo.

Profile matcher Settings

Home / Manage Models

Manage Models

Create a new model

Work Experience

Law Edit Remove

Field: Law

Duration (years): 3

Duration Priority (1-10): 10

Role: Lawyer

Responsibilities:

Education

Law (at) Edit Remove

Study Field: Law

Institution:

Degree Level: Degree

Classification: 0

Classification Priority (1-10): 1

Soft Skills

Edit

Team Skills:

Leadership Skills:

Organizational Skills:

Interpersonal Skills:

Certifications

No Certifications Edit Remove

Hard Skills

Edit

Knowledge on:

Civil law

Intellectual property

Experience with:

Communication Skills:

Speak English

Cancel Save

CoreUI © 2019 creativeLabs. Powered by CoreUI for React

Figure 33: Gestão de modelos de emprego

A funcionalidade seguinte – “Manage Permissions”, permite atribuir ou remover o estatuto de administrador a utilizadores.

The screenshot shows the 'Manage user permissions' page in the Profile Matcher application. The sidebar on the left contains the following menu items: My profile, My Offers, Create Offer, Manage Models, Manage Permissions (highlighted), Recommended offers, and Recommended skills. The main content area displays a table with the following data:

Username	Role	Action
teste1	Admin	Revoke Admin
teste2	User	Give Admin
joscard	User	Give Admin
baduser	User	Give Admin
Zé Nuno	User	Give Admin

At the bottom of the page, there is a footer with the text 'CoreUI © 2019 creativeLabs.' and 'Powered by CoreUI for React'.

Figure 34: Gestão de permissões

A funcionalidade “Recommended Offers” serve para visualizar as ofertas recomendadas ao candidato/estudante. É possível comparar graficamente as diferentes ofertas de emprego, como é demonstrado na imagem seguinte.

The screenshot shows the 'Recommended Offers' page in the Profile Matcher application. The sidebar on the left contains the following menu items: My profile, My Offers, Create Offer, Manage Models, Manage Permissions, Recommended offers (highlighted), and Recommended skills. The main content area displays a horizontal bar chart titled 'Candidates ordered by best fit' and a radar chart titled 'Candidate skills'.

The horizontal bar chart shows the following data:

Offer	Score
Informatics	70
Lawyer to represent my client	45
Experienced Lawyer (Civil)	35
Manager at my shop	35

The radar chart shows the following data for the three offers:

Offer	Work Experience	Education	Soft Skills	Certifications	Hard Skills
Informatics	90	80	70	60	50
Lawyer to represent my client	80	70	60	50	40
Experienced Lawyer (Civil)	70	60	50	40	30

At the bottom of the page, there is a footer with the text 'CoreUI © 2019 creativeLabs.' and 'Powered by CoreUI for React'.

Figure 35: Ofertas recomendadas

A funcionalidade seguinte – “Recommended Skills”, permite visualizar as competências que a aplicação sugere que sejam melhoradas. É possível

visualizar competências a melhorar relativamente a modelos de emprego ou a ofertas de emprego.

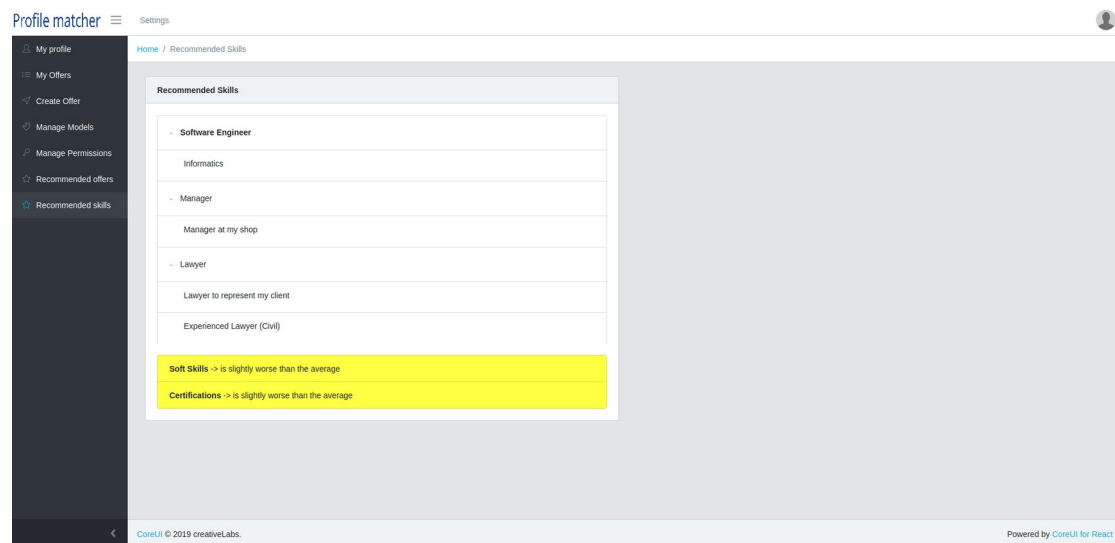


Figure 36: Competências recomendadas

Por fim, existe ainda uma página para as preferências do utilizador. Nesta página o utilizador pode indicar se pretende que o seu nome de utilizador seja visível para os recrutadores, ou se se pretende manter anónimo.

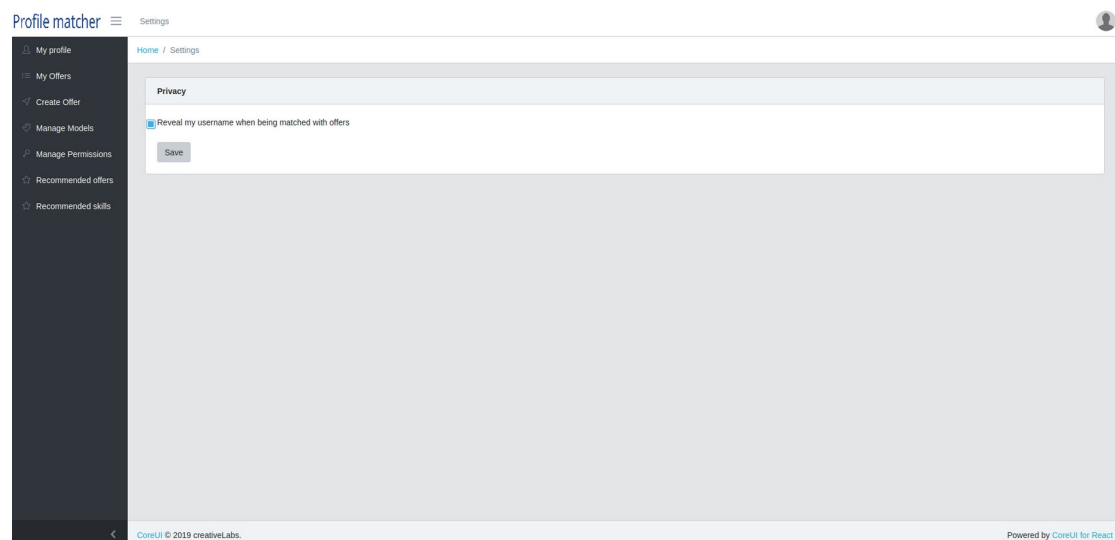


Figure 37: Preferências

5.3.2 Implementação

Esta aplicação foi desenvolvida com base no padrão MVC. Assim sendo, o código-fonte está dividido em três pastas: *Models*, *Views*, *Controllers*. A primeira representa abstrações de conceitos, como o modelo de emprego. Os *Views* contêm componentes, em formato JSX, juntamente com ficheiros CSS. Cada um destes componentes define como é desenhada uma página, ou uma parte de uma página. Por fim, os *Controllers* são responsáveis pelo comportamento destas páginas.

O formato JSX é uma extensão sintática ao Javascript, tipicamente utilizado em aplicações desenvolvidas em React.[46] Permite implementar páginas web de forma mais intuitiva, assim como de gerir eventos mais facilmente. O excerto de código seguinte é um exemplo da utilização desta extensão.

```
const element = <h1>Hello World</h1>;
```

O equivalente sem JSX, seria:

```
const element = React.createElement('h1', null, 'Hello world');
```

A implementação e a interação entre os *controllers*, modelos e componentes é semelhante nos casos de uso implementados. A figura seguinte é um diagrama de sequência que demonstra o processo no início de sessão.

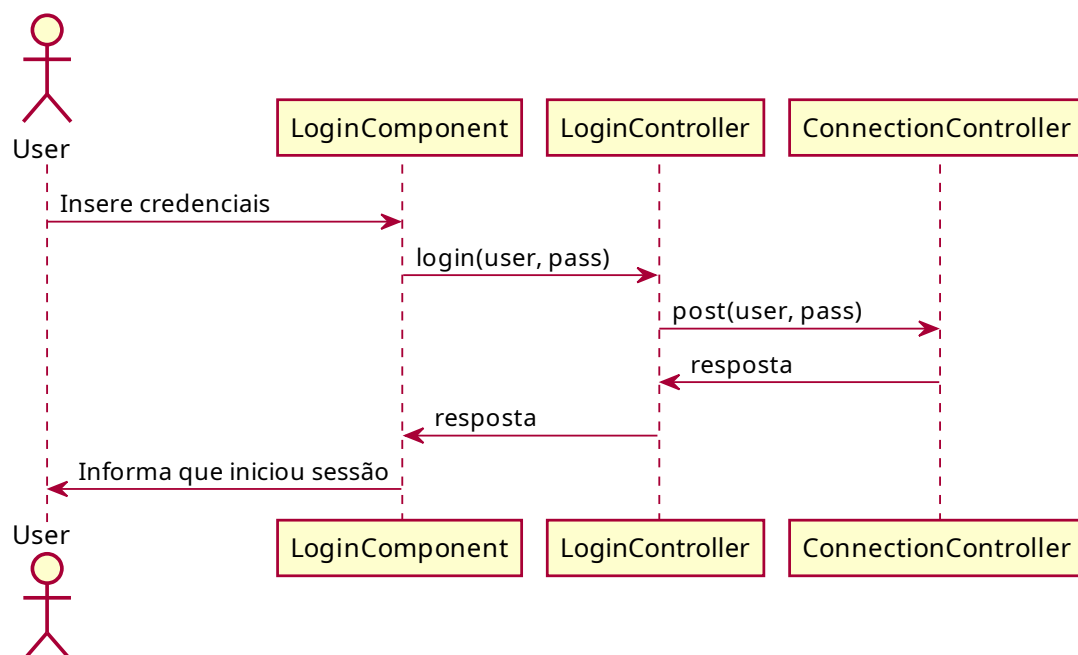


Figure 38: Login - Diagrama de Sequência

É importante referir o papel da classe *ConnectionController* que aparece neste diagrama. Tem como objetivo facilitar fazer pedidos HTTP para a Web API. A figura seguinte é um excerto de código da função *post*, referenciada no diagrama de sequência anterior.

```
static post(data, endpoint, callback, jwt = null) {
  const request = new XMLHttpRequest();
  request.onload = function() {
    callback({
      status: this.status,
      response: this.responseText
    });
  };
  request.open("POST", API_URL + endpoint);
  request.setRequestHeader('Content-type', 'application/json; charset=utf-8');
  if (jwt) {
    request.setRequestHeader('Authorization', 'Bearer ' + jwt);
  }
  request.send(JSON.stringify(data));
}
```

Figure 39: ConnectionController - Post

A função utiliza a biblioteca “XMLHttpRequest” para fazer os pedidos em HTTP. Se for necessário, permite enviar um JWT Token no cabeçalho. Esta classe é utilizada em todos os casos de uso que requeiram comunicação direta com a Web API.

6 Avaliação

Dado um conjunto de perfis de profissionais, e os resultados que o sistema produz, deve ser identificada a frequência de falsos positivos e negativos. Posteriormente, conclui-se se o sistema é ou não eficaz na escolha de profissionais. É fundamental que a quantidade e qualidade dos dados seja suficiente, para que as conclusões tiradas representem o mais próximo da realidade possível. Portanto, os dados devem ser avaliados estatisticamente, de modo a identificar com que confiança podemos admitir que são significativos.

Pretende-se testar se a aplicação tem uma taxa de sucesso de pelo menos 80%. Uma primeira hipótese ponderada, seria utilizar uma amostra de CV de candidatos, assim como o número de ofertas de emprego em que foi rejeitado e aceite, e comparar os resultados com os da aplicação. No entanto, a quantidade deste tipo de informação, disponibilizada publicamente, é escassa por questões de confidencialidade, o que torna esta solução pouco viável.

Devido a esta restrição, foi aplicada uma solução alternativa. Dada uma amostra de CV de candidatos que foram aceites num emprego, assim como o cargo desse mesmo emprego, verifica-se se a aplicação aconselha ao candidato ofertas de emprego deste mesmo cargo. Se a aplicação aconselhar ao candidato outro tipo de ofertas de emprego, assume-se que é um falso positivo. Esta solução é mais viável, pois existe uma quantidade de currículos, disponibilizadas publicamente, e uma variedade muito superior quando comparado com a primeira solução.

6.1 Amostra

Para que esta solução seja relevante estatisticamente, é essencial que a quantidade das amostras seja suficiente. Para esta avaliação foram utilizadas 12 ofertas de emprego e um total de 191 perfis de currículos de candidatos. Esta amostra foi obtida a partir de currículos e ofertas de emprego disponibilizadas online. A seleção das amostras foi aleatória. Houve uma filtragem manual no final da seleção para eliminar currículos semelhantes ou inválidos (e.g. com informação insuficiente no currículo).

6.2 Abordagem

A abordagem utilizada para a avaliação consiste em fazer o *matching* entre cada um dos candidatos, com todas as ofertas de emprego. De seguida, é atribuída uma pontuação, consoante o número de ofertas que sejam mais recomendadas, do que a oferta a que o candidato realmente se está a candidatar. Ou seja, esta pontuação será a maior possível, quando a oferta de emprego recomendada corresponde à oferta de emprego que o candidato se candidatou. Esta pontuação varia entre 0 e 1, e é proporcionalmente menor, quanto maior for o número de ofertas mais recomendadas que a oferta candidatada pelo candidato.

Por fim, calcula-se a média das pontuações, e este resultado é assumido como a taxa de sucesso do *matching* da aplicação. Pretende-se que este resultado seja de pelo menos 80%.

6.3 Resultados

Table 14: Resultados da Avaliação (Resultados arredondados a duas casas decimais)

Oferta de emprego	Pontuação média	Desvio-padrão
Business Analyst	0,54	0,25
Software Developer	0,88	0,15
Data Scientist	0,75	0,11
Designer	0,58	0,32
Assistant Manager	0,32	0,26
Executive	0,69	0,20
Office Manager	0,47	0,22
School Teacher	0,38	0,15
Civil Engineer	0,40	0,27
Financial Analyst	0,66	0,20
Attorney	0,68	0,29
Marketing	0,69	0,22
Média	0,59	0,22

O resultado final é de 59%, abaixo dos 80% desejados, o que significa que ainda há espaço para melhorar o algoritmo desenvolvido para o *matching*. É de notar que, em ofertas de emprego onde houve uma maior quantidade de currículos e ofertas utilizadas para o treino do mecanismo de aprendizagem automática, os resultados tendem a ser mais positivos. Em particular o “Software Developer”, com uma pontuação média de 88% e um desvio-padrão de apenas 15%. Isto

pode significar que houve uma fraca variedade de amostras no processo do treino, e é algo que pode ser corrigido no futuro, de modo a obter melhores resultados no geral.

Foram colocadas as seguintes questões, na “Introdução”:

- É possível identificar padrões de candidatos com base nos seus perfis, com técnicas de aprendizagem automática?
- É mais rápido utilizar a aplicação desenvolvida, em vez de uma análise manual, no processo de seleção de candidatos?
- É viável selecionar candidatos, com o auxílio desta aplicação?

Os resultados obtidos mostram algum potencial, relativamente à primeira questão. Apesar de os resultados estarem aquém do inicialmente esperado, uma maior variedade e uma maior quantidade na amostra, no processo de treino, podem fazer com que seja viável identificar padrões de candidatos com base nos seus perfis.

Como indicado na “Introdução”, em média as empresas demoram cerca de 29 dias a recrutarem um candidato[1]. Apesar desta aplicação não substituir por completo o trabalho manual no processo de seleção, pode-se afirmar que o seu uso, como ferramenta de auxílio, acelera este processo. Obviamente que para além da rapidez do processo, é essencial que o mesmo seja eficaz. Algo que neste momento ainda pode e deve ser melhorado.

Relativamente à última questão, não se pode afirmar que seja viável, neste momento, selecionar candidatos com o auxílio desta aplicação. No melhor dos casos, poderá ser viável apenas em determinadas ofertas de emprego.

7 Conclusões

O objetivo desta tese era de desenvolver uma aplicação que auxilie o processo de seleção de candidatos, que apesar de ser uma secção fulcral das empresas, ainda é feito manualmente em grande parte dos casos. Adicionalmente, que a aplicação guiasse os estudantes ou candidatos jovens no início da sua carreira profissional.

Depois de uma análise ao estado de arte, optou-se por desenvolver uma aplicação Web, que auxilie ambas as partes referidas. Concluiu-se que a implementação de técnicas de aprendizagem automática era a solução mais adequada, devido à complexidade do problema.

Foi desenvolvida uma Web API e uma aplicação responsável por fazer o *matching* entre candidatos e ofertas de emprego, juntamente com a aplicação Web. Todas estas aplicações foram desenvolvidas, e estão funcionais, assim como todos os requisitos funcionais.

Por fim, foi feita uma avaliação que analisasse o desempenho do *matching* entre candidatos e ofertas de emprego. Esta avaliação não foi tão positiva como inicialmente esperado, o que deixa espaço para melhorias no futuro.

Em suma, o resultado final é geralmente positivo, apesar dos resultados obtidos na avaliação do *matching*. De seguida, são descritos mais sucintamente os objetivos alcançados, assim como as limitações do trabalho.

7.1 Objetivos realizados

Esta secção retrata os objetivos para a aplicação mencionados anteriormente, e se foram ou não realizados com sucesso. A tabela seguinte demonstra esses objetivos e o seu grau de realização.

Table 15: Objetivos e grau de realização

Requisito funcional	Grau de realização
Registo	Realizado
Visualização de características necessárias para ofertas de emprego	Realizado
Visualização de recomendações para uma oferta de emprego	Realizado
Consulta de lista de modelos existentes	Realizado
Criação de ofertas de emprego, com base num modelo	Realizado
Consulta de lista das suas ofertas de emprego	Realizado
Consulta de lista de candidatos adequados a uma oferta de emprego	Realizado
Adição de modelos de emprego	Realizado
Edição de modelos de emprego	Realizado
Remoção de modelos de emprego	Realizado
Consulta de modelos de emprego	Realizado

Os requisitos funcionais descritos foram todos realizados com sucesso.

7.2 Limitações e trabalho futuro

Apesar de todos os requisitos funcionais descritos na secção anterior terem sido desenvolvidos com sucesso, existem algumas limitações na aplicação e trabalho a ser feito no futuro. Esta secção descreve essas limitações.

A maior limitação do projeto é o facto da aplicação para *matching*, não ser ainda viável para utilização no mercado de trabalho, como indicado na secção de

“Avaliação”. Seria necessário fazer uma pesquisa mais profunda e/ou com fornecimento de dados de currículos de empresas, de modo a obter uma amostra maior, e principalmente com maior variedade. Conforme já referido, certas ofertas de emprego, particularmente das que foram obtidas mais currículos, tendem a ter resultados melhores. Apesar dessa afirmação não ter sido comprovada estatisticamente, é um indício positivo de que o processo utilizado para a aprendizagem automática tem potencial.


Adicionalmente, a implantação do sistema foi sempre feita localmente. Para que este possa ser usado por diversos utilizadores e online, seria necessário implementar uma nova estratégia de implantação. A estratégia deveria ter em conta questões como a escalabilidade e custos associados, algo que não foi relevante até agora.

Futuramente, também seria interessante fazer uma renovação em termos de *design* à aplicação Web, para que seja mais apelativa e de maior facilidade de utilização.

Bibliografia

Bibliography

- 1: 2016 Human Capital Report - 2016-Human-Capital-Report, 2016, <https://www.shrm.org/hr-today/trends-and-forecasting/research-and-surveys/Documents/2016-Human-Capital-Report.pdf>
- 2: THE QUALITIES OF A GOOD CV AND RESUME, 2014, https://doi.org/10.1007/978-1-4939-0647-5_1
- 3: 50 HR & Recruiting Stats That Make You Think|50 HR & Recruiting Stats That Make You Think, 2020, <https://www.glassdoor.com/employers/blog/50-hr-recruiting-stats-make-think/>
- 4: Making Selection Easier with Automated Recruitment, 2019, <https://www.launchpadrecruits.com/insight-articles/automated-recruitment>
- 5: HR Software, ATS, Diversity Recruitment, 2019, <https://www.transformify.org/>
- 6: Ideal | AI For Recruiting Software | Maximize Quality of Hire, 2019, <https://ideal.com/>
- 7: How Artificial Intelligence Impacts the Candidate Experience, 2018, <https://modernhire.com/how-artificial-intelligence-impacts-the-candidate-experience/>
- 8: 2017 Automation Nation Report: Recruiter Perspectives - Jobvite, 2017, <https://www.jobvite.com/blog/jobvite-news-and-reports/2017-automation-nation-report-recruiter-perspectives-automated-future/>
- 9: Soumen Chakrabarti, Martin Ester, Usama Fayyad, Johannes Gehrke, Jiawei Han, Shinichi Morishita, Gregory Piatetsky-Shapiro, Wei Wang, Data mining Curriculum: A Proposal, 2006
- 10: World Internet Users Statistics and 2019 World Population Stats, 2019, <https://www.internetworldstats.com/stats.htm>

- 11: Facebook Reports Fourth Quarter and Full Year 2018 Results, 2018, https://s21.q4cdn.com/399680738/files/doc_financials/2018/Q4/Q4-2018-Earnings-Release.pdf
- 12: Tom Mitchell, Machine learning, 1997
- 13: Yoshua Bengio, Aaron Courville, Pascal Vincent, Representation Learning: A Review and New Perspectives, 2014
- 14: , 2020, <http://cs231n.github.io/neural-networks-1/>
- 15: machine-learning · GitHub Topics, 2020, <https://github.com/topics/machine-learning>
- 16: TensorFlow, 2020, <https://www.tensorflow.org/>
- 17: François Chollet, Xception: Deep Learning with Depthwise Separable Convolutions, 2017
- 18: pytorch/pytorch: Tensors and Dynamic neural networks in Python with strong GPU acceleration, 2020, <https://github.com/pytorch/pytorch>
- 19: P. Koen, Greg Ajamian, S. Boyce, Allen Clamen, Eden S. Fisher, S. Fountoulakis, A. Johnson, P. Puri, R. Seibert, Fuzzy Front End:Effective Methods, Tools,and Techniques, 2002
- 20:  (JPEG Image, 364 × 339 pixels), 2020, https://www.researchgate.net/profile/Igor_Casenote/publication/322031431/figure/fig5/AS:574878987362304@1514073239341/Figura-5-Funcionamento-do-modelo-New-Concept-Development-Fonte-Koen-et-al-2014.png
- 21: Carol M. Kopp, Perceived Value Definition, 2020
- 22: Barnes, C., Blake, H., & Pinder, D., Creating & Delivering Your Value Proposition: Managing Customer Experience for Profit, 2009
- 23: Barquet, Ana Paula B., et al, Business model elements for product-service system, 2011

- 24: Verna Allee, Understanding Value Networks, 2001
- 25: VNA-Map-Template-from-NPQ.png (PNG Image, 579 × 392 pixels) - Scaled (94%), 2020, <https://partneringresources.com/wp-content/uploads/VNA-Map-Template-from-NPQ.png>
- 26: Porter, M., Strategy and the Internet, 2001
- 27: Value-Chain-Analysis, 2020, <https://www.marketing91.com/wp-content/uploads/2014/05/Value-Chain-Analysis.jpg>
- 28: NVIDIA Data Center Deep Learning Product Performance | NVIDIA Developer, 2020, <https://developer.nvidia.com/deep-learning-performance-training-inference>
- 29: tensorflow/tensorboard: TensorFlow's Visualization Toolkit, 2015, <https://github.com/tensorflow/tensorboard/>
- 30: Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean, Efficient Estimation of Word Representations in Vector Space, 2013
- 31: Emeric Dynomant, Stéfan J. Darmoni, Emeline Lejeune, Gaëtan Kerdelhué, Jean-Philippe Leroy, Vincent Lequertier, Stéphane Canu and Julien Grosjean, Study of a new approach to generate related articles, 2019
- 32: 1*keqyBCQ5FL6A7DZLrXamvQ.png, 2020, https://miro.medium.com/max/327/1*keqyBCQ5FL6A7DZLrXamvQ.png
- 33: Xuedong Huang and Fileno Allea and Hsiao-wuen Hon and Mei-yuh Hwang and Ronald Rosenfeld, The SPHINX-II Speech Recognition System: An Overview, 1992
- 34: gensim · PyPI, 2020, <https://pypi.org/project/gensim/>
- 35: Daniel Glez-Peña, Anália Lourenço, Hugo López-Fernández, Miguel Reboiro-Jato, Florentino Fdez-Riverola, Web scraping technologies in an API world, 2014
- 36: urllib3 · PyPI, 2020, <https://pypi.org/project/urllib3/>

- 37: lxml · PyPI, 2020, <https://pypi.org/project/lxml/>
- 38: Node.js, 2020, <https://nodejs.org/>
- 39: The most popular database for modern apps | MongoDB, 2020, <https://www.mongodb.com/>
- 40: JSON Web Token Introduction, 2020, <https://jwt.io/introduction/>
- 41: npm-pdfreader, 2020, <https://github.com/adrienjoly/npm-pdfreader>
- 42: antonydeepak/ResumeParser, 2020, <https://github.com/antonydeepak/ResumeParser>
- 43: OmkarPathak/pyrespaser, 2020, <https://github.com/OmkarPathak/pyrespaser>
- 44: React - A JavaScript library for building user interfaces, 2020, <https://reactjs.org/>
- 45: reactstrap - React Bootstrap 4 components, 2020, <https://reactstrap.github.io/>
- 46: Introducing JSX - React, 2020, <https://reactjs.org/docs/introducing-jsx.html>