UNIVERSIDADE
LUSÓFONA

CV Tool – CV Capture Data

Trabalho Final de curso

Relatório Final

Discente: António Rocha

Docente Orientador: José Brás

Trabalho Final de Curso | LIG | 27/11/22

www.ulusofona.pt

Disclaimer

Devido ao âmbito deste trabalho final de curso, em parceria com a CGI e o acordo assinado por ambas as partes relativamente á não divulgação (NDA), todos os direitos são reservados à entidade externa (CGI). É proibida a publicação e distribuição deste relatório e quaisquer informações acerca do mesmo, tendo apenas autorização para a partilha do documento em questão, com os membros do júri e responsáveis com o objetivo de avaliação do trabalho. Aplica-se o mesmo a todos os links associados a desenhos (protótipos, mockups, arquiteturas, entre outros) e repositórios de código.

Com a finalização da cadeira “TFC”, este relatório deverá ser eliminado e os acessos aos links acima descritos removidos.

Resumo

Este trabalho tem como foco a construção de uma solução end-to-end para, extrair, transformar e gerir a informação contida em curriculums vitaes (CVs), com toda a infraestrutura necessária que permita a automatização das operações do dia a dia de uma organização.

Pretende-se com o auxílio de técnicas de machine learning (ML) recuperar informação e guardar CVs, em diferentes formatos, em diferentes línguas e com outros desafios da área de recursos humanos de uma forma estruturada e otimizada.

A concretização da solução passa por implementar uma pipeline automática para ingestão de dados manualmente introduzidos pelos utilizadores, usando as melhores práticas de operações de machine learning (MLops), e disponibilizar serviços de ML. Numa primeira instância, junto com a pipeline construir um serviço de análise que extrai a informação de documentos. Numa segunda instância, construir um serviço para fazer a classificação de CVs com base em critérios específicos como experiência, competências, certificados, etc.

Este trabalho depende de outros trabalhos finais de curso em execução pelo que nesta fase é assumido que será possível integrar a solução no fim, contudo se tal não se verificar, será preciso descobrir métodos alternativos para integração.

Palavras-Chave: End-to-end, extrair, curriculums vitaes, infraestrutura, automatização, operações, organização, machine learning, recursos humanos, estruturada, otimizada, pipeline, ingestão, dados, serviços, documentos, classificação, trabalho final de curso, integração.

Abstract

This work focuses on building an end-to-end solution to extract, transform and manage the information contained in curriculums vitaes (CVs), with all the necessary infrastructure that allows the automation of day-to-day operations of an organization.

The aim, with the help of machine learning (ML) techniques, is to save CVs in a structured and optimised way that allows searching in different formats, in different languages and with other challenges specific to the human resources area.

The realization of the solution involves implementing an automatic pipeline for ingesting data manually entered by users, using the best practices of machine learning operations (MLops), and providing ML services. In a first instance, along with the pipeline build an analysis service that extracts information from documents. In a second instance, build a service to do CV classification based on specific criteria such as experience, skills, certificates, etc.

This work depends on other final course work in execution so at this stage it is assumed that it will be possible to integrate the solution in the end, however if this does not happen, it will be necessary to find alternative methods for integration.

Keywords:

End-to-end, extract, curriculums vitaes, infrastructure, automation, operations, organization, machine learning, human resources, structured, optimized, pipeline, ingestion, data, services, documents, classification, final course work, integration

Índice

Índice de Figuras

Índice de Tabelas

Índice de Anexos

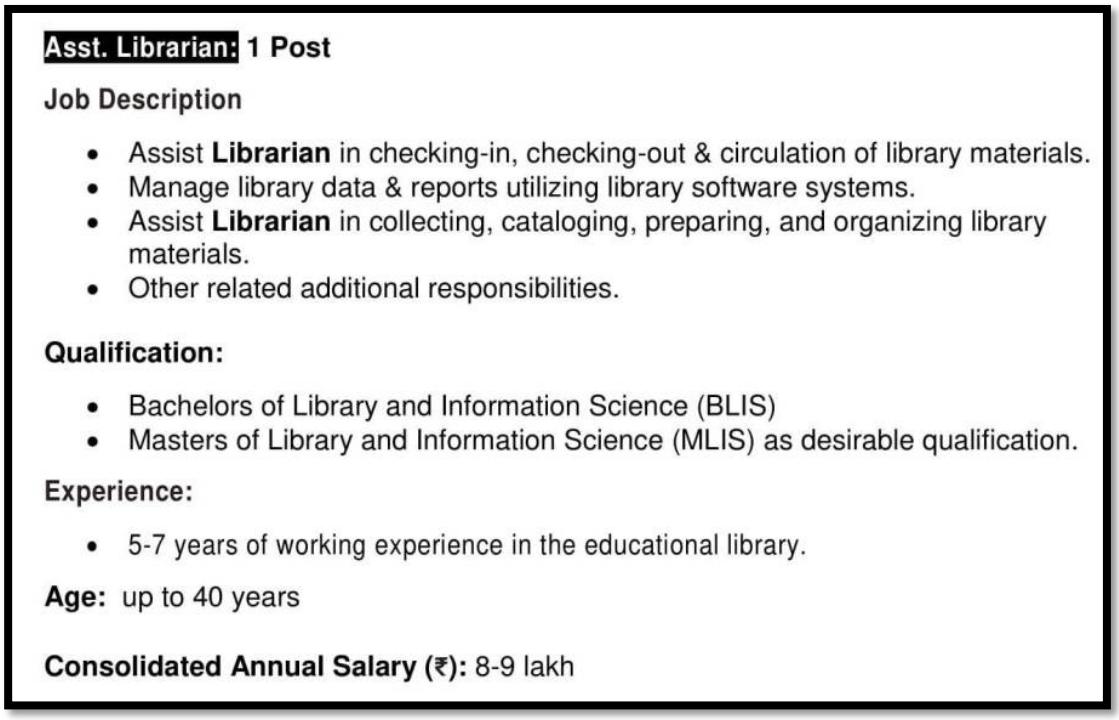
1 Identificação do Problema

Quer sejam recrutadores a procura de talento ou uma empresa a filtrar colaboradores para concorrer a concursos de novos clientes, é preciso encontrar os melhores candidatos dentro de um conjunto de currículos (CVs).

Geralmente, não é possível avaliar ao detalhe todos os CVs, e com a falta de infraestrutura para guardar e recuperar informação nova ou existente, o trabalho manual de selecionar CVs numa empresa torna-se enfadonho, ineficaz e um desperdício de tempo de recursos humanos.

Parte do processo de seleção de CVs é ler o currículo e julgar se este é ou não adequado para determinado perfil. É preciso ler e analisar a informação a partir de um currículo, e compará-lo com uma descrição alvo para fazer match. Esta é uma tarefa que a maioria dos seres humanos consegue fazer. Contudo, difícil de executar por uma máquina.

Muitos sistemas informáticos tradicionais falham nesta gestão de diferentes currículos. No entanto, com o recente progresso em machine learning (ML) e técnicas de processamento de linguagem natural (NLP) com visão computacional muitas tarefas no processo de seleção de CVs são agora possíveis de ser executadas.



Asst. Librarian: 1 Post

Job Description

- Assist **Librarian** in checking-in, checking-out & circulation of library materials.
- Manage library data & reports utilizing library software systems.
- Assist **Librarian** in collecting, cataloging, preparing, and organizing library materials.
- Other related additional responsibilities.

Qualification:

- Bachelors of Library and Information Science (BLIS)
- Masters of Library and Information Science (MLIS) as desirable qualification.

Experience:

- 5-7 years of working experience in the educational library.

Age: up to 40 years

Consolidated Annual Salary (₹): 8-9 lakh

Figura 1 - Exemplo de uma descrição de trabalho

Figura 2 - Exemplo de currículo profissional

Neste trabalho, apresentamos um serviço de análise de CVs desenvolvido com o objetivo de extrair texto de CVs armazenados como imagens.

O Serviço (API) apresenta resultados que estão de acordo com os objectivos previstos na proposta inicial. Cumprem o objetivo principal de converter eficientemente CVs baseados em imagens em texto legível, permitindo um processamento simplificado da informação

dos CVs. Ao atingir este objetivo, o serviço facilita com êxito o processo manual de introdução de dados e reduz significativamente o tempo e o esforço necessários durante a fase de seleção dos candidatos.

Embora o serviço de processamento de CVs sirva como prova de conceito, foram encontradas algumas limitações durante o seu desenvolvimento e implementação. Um desafio significativo está relacionado com a integração do texto analisado com as descrições de funções para correspondência automática (matching). O plano inicial envolvia também a criação de uma aplicação para comparar as descrições de funções com os perfis de CV analisados, o que facilitaria o matching candidato-função. No entanto, esta tarefa revelou-se impossível.

O principal obstáculo resultou da indisponibilidade de uma base de dados de descrições de funções. Apesar dos esforços para recolher uma gama diversificada de funções de vários sectores, a obtenção de um conjunto de dados substancial com descrições detalhadas de cada função e o candidato mais adequado revelou-se inviável. Consequentemente, a falta de tais dados impediu a criação de um sistema eficaz de matching entre os dados de texto extraídos e potenciais empregos.

2 Viabilidade e Pertinência

Em parceria com a CGI, foi proposto à Universidade Lusófona de Humanidades e Tecnologias (ULHT), com a orientação do professor José Brás, uma solução para gerir toda a informação proveniente de CVs. A solução integrada teria funcionalidades de registo e pesquisa de CVs de forma estruturada e otimizada.

O advento de ML permitiu às empresas automatizar muitas tarefas manuais. No entanto, a maioria das soluções end-to-end nunca chega à produção, permanecendo no domínio académico ou em soluções caseiras com pouca reprodução. As ferramentas state of the art (SOTA) na extração de informação em currículos e automatização de processos relacionados são privatizadas. Como tal, embora ML tenha feio grandes progressos neste domínio, soluções completas estão em falta no mercado.

A solução proposta por este trabalho poderá servir como uma base acessível para futuros desenvolvimentos em tarefas relacionadas.

Na prática, a falta de processos para a gestão de dados para ciência de dados é um aspeto critico que também condiciona as equipas de desenvolvimento. Sem uma metodologia e infraestrutura necessária para experimentação, não é possível para um cientista de dados ou outras equipas trabalharem os dados. Especialmente em múltiplos problemas de ciência de dados como lidar com texto em múltiplas línguas, conversão de imagem para texto e lidar com diferentes formatos. Para agilizar este processo torna-se imperativo adotar boas práticas de engenharia de dados, design, implementação e manutenção em produção de forma contínua, fiável e eficiente (MLops).

Temas como a privacidade de dados sensíveis, grandes quantidades de CVs ou dados extraídos impossíveis de serem processados por software tradicional (big data), resultados enviesados com discriminação entre características dos dados (bias e fairness) são alguns dos temas relevantes para lidar e implementar na solução ao usar boas práticas de engenharia de dados.

Temas como o automatismo de ler e analisar CVs, ferramentas acessíveis open source para a tarefa e padrões de desenho para o lançamento em produção de ML são alguns dos temas relevantes para lidar e implementar na solução ao usar boas práticas de MLops.

2.1 Análise comparativa e viabilidade do projeto

Durante o desenvolvimento do projeto, foram feitas algumas modificações na proposta inicial do projeto. Esta análise comparativa tem como objetivo demonstrar que essas alterações não comprometeram a viabilidade e relevância do Trabalho Final de Curso (TFC).

Na proposta inicial, o foco principal era a conceção de um serviço de análise de CVs de imagens para texto. E à medida que o desenvolvimento avançou, tornou-se evidente que a oferta de maior flexibilidade para acomodar vários formatos de imagem e layouts melhoraria significativamente a usabilidade do serviço. Consequentemente, refinámos o

âmbito do projeto para assegurar a compatibilidade com uma gama mais vasta de estilos de CVs e idiomas. Este refinamento do âmbito permitiu que o serviço de análise de CVs abrangesse um público mais vasto, tornando-o mais relevante em cenários reais.

O plano original também envolvia a utilização do texto extraído dos CVs para construir uma aplicação para comparar descrições de funções com perfis de CVs para fazer matching automático de funções. No entanto, devido à indisponibilidade de um conjunto de dados abrangente de descrições de funções e candidatos adequados, este aspeto deparou-se com limitações. Para resolver este desafio, adaptámos o foco do projeto para se concentrar na funcionalidade central de análise de CVs, garantindo que o objetivo principal do TFC permanecesse intacto. Embora a componente de correspondência entre empregos não pudesse ser totalmente realizada, esta mudança não prejudicou a viabilidade global do serviço de análise de CVs. Pelo contrário, permitiu-nos criar uma ferramenta de análise de CVs robusta e fiável, lançando as bases para futuras melhorias, no caso, matching de funções a CVs.

Em conclusão, a análise comparativa revela que as alterações efectuadas durante o desenvolvimento da API do serviço de análise de CVs não comprometeram a sua viabilidade e relevância como Trabalho Final de Curso. O serviço de análise de CVs continua a ser uma ferramenta útil para converter eficientemente CVs baseados em imagens em texto, demonstrando assim o sucesso do projeto em abordar desafios do reais e fornecendo uma solução eficaz para as necessidades de análise de CVs.

3 Benchmarking

3.1 Estado da arte

Normalmente as empresas possuem ou recebem muitos currículos de candidatos para preencher posições. A análise destes documentos é realizada manualmente pelos departamentos de recursos humanos e não existe uma estrutura para a armazenar e classificar a informação dos candidatos. Não é possível de uma forma sistemática garantir o acesso a todas as pessoas na organização, nem é possível aprender a analisar currículos e descrições de vagas correspondentes para tomar decisões de negócio mais informadas.

Por exemplo, os gestores que não conseguem aceder a informação dos currículos e acabam por ter de pedir o preenchimento de CVs descaracterizados com perfis a fim de se enviar para propostas de concursos.

Existe, contudo, algumas tecnologias e técnicas que colmatam alguns dos problemas encontrados nas empresas.

3.2 Soluções existentes.

De momento já existem no mercado algumas soluções que facilitam a criação da solução proposta por este TFC, pelo que nesta fase é relevante listar as alternativas para a nossa solução e identificar qual o melhor posicionamento.

Algumas das alternativas são:

- Serviços pagos.
- Serviços gratuitos.

3.3 Serviços Pagos.

Existem inúmeras empresas especializadas existentes no mercado que permitem através do uso de APIs fazer gestão de CVs. Muitas delas construídas em cima de serviços baseados na tecnologia de reconhecimento de caracteres (OCR) ou heurísticas.

Por exemplo, algumas empresas oferecem soluções de gestão de CVs integradas com *Microsoft Azure Cognitive Services*, que inclui serviços de OCR eficientes para extrair informações de currículos.

Além disso, é possível utilizar serviços como o *Google Cloud Vision OCR*, que oferece recursos para processar e entender o conteúdo de currículos digitalizados.

Nestes sites é possível fazer quase todas as componentes de ML que este trabalho se propõe a fazer com recursos a APIs.

3.4 Serviços gratuitos.

Não existe nenhum serviço gratuito que faça a ingestão, analise e/ou gestão de currículos. Existem apenas é alguns serviços separados para fazer tradução de texto.

3.5 Posicionamento

Ao analisar as alternativas observamos que as soluções existentes para implementar certas componentes são pagas. Contudo, não resolvem o problema de criar uma pipeline automática, nem criar uma fonte de conhecimento. Existe valor em construir internamente a solução proposta neste tfc e melhorar a medida que novos desenvolvimentos na área de ciência de dados vão surgindo.

3.6 Análise de benchmarking

Nesta secção, fazemos uma análise de benchmarking da API do serviço de análise de CVs, comparando as suas funcionalidades. Adicionalmente, avaliamos o alinhamento da solução com as estimativas iniciais feitas.

No processo de benchmarking, analisamos diversas funcionalidades oferecidas pela API em comparação com concorrentes identificados. A nossa solução provou ser flexível, acomodando vários formatos, idiomas e layouts de CVs. Embora nem todos os concorrentes ofereçam o mesmo nível de flexibilidade e integração, o serviço de análise de CVs surgiu como uma solução completa, satisfazendo os diversos casos de uso pensados para os utilizadores.

Ao longo do TFC, as nossas estimativas iniciais incluíam a integração da tecnologia OCR e a garantia de flexibilidade no tratamento de vários formatos de CV. Os resultados finais foram semelhantes ao esperado dado o contexto atual. Além disso, a decisão de adaptar o foco do projeto para dar prioridade à funcionalidade central de análise de CV não comprometeu a sua viabilidade ou relevância. Pelo contrário, permitiu-nos fornecer uma solução robusta e prática.

À medida que continuarmos a aperfeiçoar e a melhorar a solução, prevemos o seu crescimento contínuo e o seu impacto positivo no domínio de processamento e análise de CVs.

4 Engenharia

4.1 Enquadramento

A convite da CGI, foi efetuada uma análise ao processo de captura de dados em currículos, referente ao âmbito do trabalho final de curso CV Capture Data, da qual foram identificados um conjunto de insights relevantes para a criação de uma solução para os desafios encontrados na gestão de currículos.

Como consequência deste trabalho, destacou-se as principais preocupações da CGI, que resultaram nos seguintes desafios:

Aumentar eficiência no processamento dos currículos – Os currículos que chegam à CGI não são automaticamente processados. Muitas vezes não estão disponíveis para todas as áreas de que deles necessitam.

Melhoria da experiência de utilização e aumento na facilidade de pesquisa – Os currículos que chegam a CGI são processadas internamente de forma a dar a resposta a casos de uso específico. Enquanto é possível consultar toda a informação observando manualmente todos os currículos, todos as pesquisas mais complexas efetuados e respetivas respostas são "limitadas" e pouco ágeis na recuperação de informação relevante.

Potenciar Iniciativas de licitação – A potencial utilização de uma solução em concursos para ganhar contratos poderia oferecer uma forma mais eficiente e precisa de rastreio de currículos. Num cenário de licitação competitiva, esta solução poderia ser apresentada como uma mais-valia para um processo de seleção mais rápido e mais preciso, resultando num conjunto de candidatos mais qualificados.

Capacidade linguística – Além disso, a capacidade de uma solução para lidar com currículos multilinguais pode ser interessante. À medida que mais e mais empresas expandem as suas operações a nível global, a capacidade de trabalhar eficazmente currículos em várias línguas torna-se cada vez mais importante.

Em resposta a estes desafios identificadas, este trabalho apresenta uma proposta de colaboração com a CGI, de onde se destaca a seguinte solução:

Soluções para os desafios identificados pela CGI:

1. **Solução inteligente de análise de resumos (Componentes)**

   I. Modelo de Object Detection em imagem para fazer o processamento de resumos.

   Esta componente utiliza algoritmos, desenvolvidos em Python, de deteção de objectos pré-treinados para identificar e localizar objectos específicos ou regiões de interesse nas imagens, bounding boxes em torno do texto. Ao empregar técnicas de visão computacional no serviço disponibilizado, o modelo pode extrair eficazmente informações relevantes de imagens, tais como documentos, facturas ou formulários com inteligência (ex.: atenção ao layout em CVs).

   II. Modelo de Optical Caracter Recognition (OCR) para capturar o texto em imagens

   Esta componente funciona como um processo intermédio interno ao serviço disponibilizado (API), uma maneira eficiente, para converter documentos ou imagens digitalizados que contenham texto em formatos legíveis por máquina (objetos, strings, etc...). Ao reconhecer e extrair com precisão o texto das imagens, a solução permite a captura eficiente de dados e minimiza a necessidade de introdução manual de dados, reduzindo os erros e aumentando a produtividade.

   III. Modelo de Named Entity Recognition e técnicas tradicionais de processamento de texto para extrair informação relevante.

   Este componente combina modelos de Reconhecimento de Entidades Nomeadas (NER) com técnicas tradicionais de processamento de texto para extrair informações cruciais do texto capturado. O NER identifica e classifica entidades como informação privada, organizações, localizações, datas e muito mais. O modelo NER garante que as informações relevantes são extraídas com maior precisão, permitindo uma compreensão mais profunda dos dados e suportando as várias aplicações que dos dados extraídos dependem (ex.: extrair CVs para uma base de dados).

   Vantagens:

- Processamento automático ou para auxiliar colaboradores a acelerar os processos de extração de informação.
- Estruturação dos dados para facilitar casos de uso com base nos dados extraídos.
- Capacidade de trabalhar dados multilinguais desde que seja assegurado que os modelos de OCR e NER consigam lidar com texto multilingual.

---

2. Solução inteligente para a licitação

I. Modelo de matching entre candidatos e concursos

Esta componente utiliza um modelo de matching que automatiza o processo de identificar os candidatos mais adequados para concursos específicos. O modelo dada uma descrição de função em texto é utilizado técnicas de acálise de dados e machine learning para comparar os requisitos dos concursos com a base de conhecimento criada pelos dados extraídos da solução inteligente de análise de CVs.

Vantagens:

- Redução de tempo gasto a recolher e filtrar por currículos de todos os colaboradores

Em suma, estas soluções visam ajudar a CGI a ultrapassar as dificuldades sentidas atualmente, de forma a aumentar a eficácia do processamento de currículos, a reduzir o tempo de consulta, a análise dos currículos ás questões formuladas pelos gestores e outros stakeholders de negócio, a reduzir o esforço direcionado à licitação de concursos e a dar mais autonomia a casos de usos “data driven” com informações contidas nos currículos.

É importante realçar que são estes modelos de machine learning que farão parte dos entregáveis, idealmente implementados em python num ou mais serviços (APIs).

4.2 Objetivo

Face as necessidades identificadas pela CGI e respetiva priorização de requisitos, a solução proposta pressupõe a sua implementação em fases, sendo direcionada para o uso interno da CGI em conjunto com o trabalho final de curso CVTool para tratamento de currículos. Ficando para já, fora de âmbito a relação e comunicação entre utilizadores e esta aplicação.

Esta solução apesar de inicialmente estar direcionada para o trabalho final de curso CVTool, visa no futuro poder servir outros canais e/ou aplicações que dela necessitem.

Considerando, portanto, que se trata de uma solução de complementaridade ao trabalho final de curso CVTool da CGI, nesta fase, pretende-se com este trabalho disponibilizar componentes inteligentes que permitem resolver os desafios associados ao processamento de currículos pré-existentes fornecidos por utilizadores genéricos, apresentando como principais objetivos a atingir, os seguintes:

1. Disponibilização de uma ou mais APIs que permitam a sua invocação independentemente do canal que as utilize.

2. Permitir a através de mecanismos de monitorização de modelos de "ML", disponibilizar KPIs de performance dos modelos de ML.

3. Possibilitar através de um processo de Gestão Técnica da solução, aceder aos vários módulos inteligentes, de forma a ensinar por meio de um retreino todos os pedidos que ainda não são corretamente identificados pelas componentes da solução.

A solução proposta neste trabalho/fase irá contemplar apenas os objetivos apresentados anteriormente, sendo que a identificação de objetivos adicionais devem ser discutidos com a orientação deste trabalho e implementados em fases futuras.

4.3 Contexto (Use Cases)

Nesta secção encontra-se descrito o contexto da solução, na medida em que os diagramas de contexto descrevem os sistemas com os quais as componentes inteligentes descritas no trabalho "CV Capture Data" irão comunicar. Descreve também, de forma sucinta, as responsabilidades que cabem a cada interveniente.

Diagramas de Contexto

O diagrama seguinte representa o contexto em que se insere a solução "CV Capture Data".

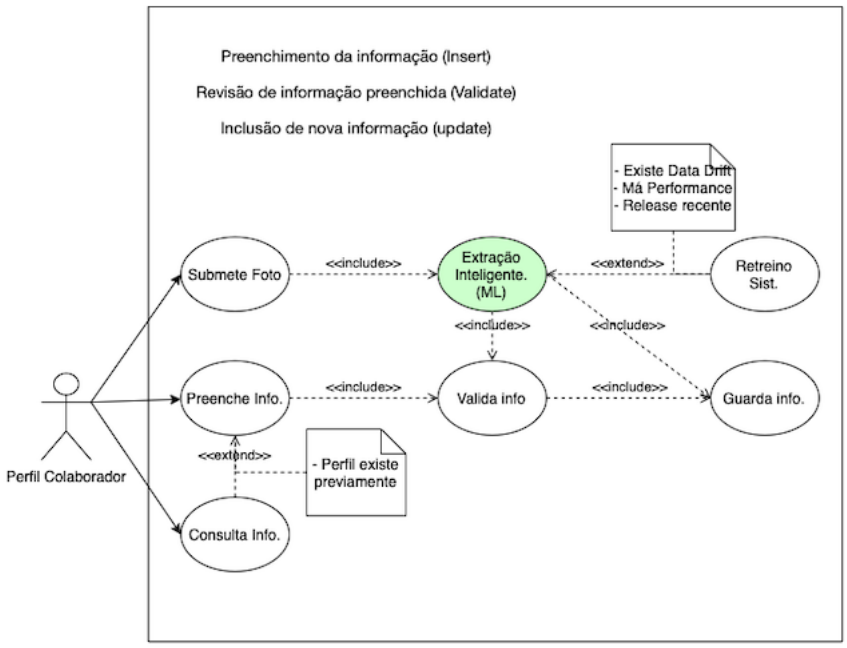


Figura 3 - Caso de Uso do Processamento de Imagem

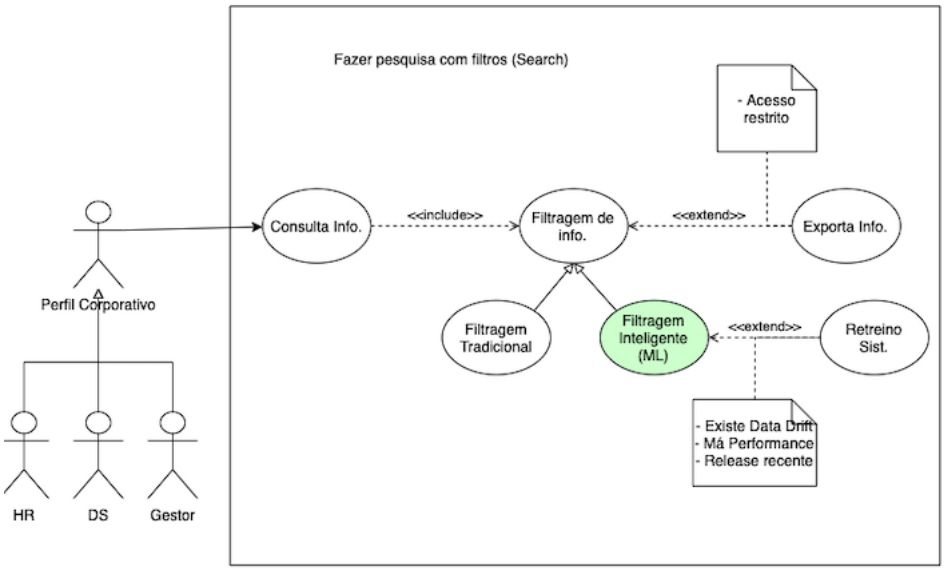


Figura 4 - Caso de Uso de Pesquisa Customizada

A solução “CV Capture Data” irá integrar com as seguintes componentes/plataformas:

CV Tool – A aplicação que deverá a partir de um ecrã̃ de designado para um pedido de processamento, invocar a componente “Inteligente” para processamento de currículos, como meio de auxilio a recolha de dados , de forma a visualizar as sugestões de dados disponibilizadas por estes modelos de machine learning.

É importante referir que no presente ano letivo existem dois projetos relacionados com a CGI: um trabalho final de curso voltado ao desenvolvimento de um serviço alimentado por modelos de Machine Learning, e outro TFC focado no desenvolvimento de um back-end e front-end.

O primeiro projeto (**CV Tool**), consiste no desenvolvimento de um back-end e front-end que funcionarão em conjunto como uma aplicação completa. O back-end será responsável por processar solicitações provenientes do front-end, gerenciar a base de dados e estabelecer a comunicação com o modelo de Machine Learning desenvolvido no segundo projeto (**CV Tool Capture Data**). O front-end oferecerá uma interface gráfica amigável aos utilizadores, permitindo a interação com o sistema. Através do front-end, os utilizadores poderão enviar currículos para processamento, visualizar resultados e gerenciar dados relevantes.

O segundo projeto consiste no desenvolvimento de modelos de Machine Learning, que serão integrados no back-end do primeiro projeto. O objetivo principal é processar imagens de currículos enviadas pelo front-end, extrair informações relevantes independentemente do layout ou idioma do currículo, e fornecer os dados estruturados para serem guardados ou apresentados. O modelo de Machine Learning será treinado utilizando técnicas de Processamento de Linguagem Natural (PLN) e Reconhecimento Ótico de Caracteres (OCR) para realizar a extração de informações dos currículos.

Ambos os projetos são interdependentes e complementares. O back-end e front-end desenvolvidos no primeiro projeto fornecem a interface do usuário para a aplicação e são essenciais para a interação dos utilizadores com o sistema. Por sua vez, o modelo de Machine Learning desenvolvido no segundo projeto é uma parte crítica para o sucesso da solução como um todo, pois é responsável por processar as imagens de currículos e extrair as informações relevantes, tornando possível a apresentação dos dados estruturados no front-end.

A integração bem-sucedida desses dois projetos resultará em uma aplicação completa e funcional para o processamento inteligente de currículos. Os utilizadores poderão enviar seus currículos através do front-end, os dados serão processados pelo modelo de Machine Learning no back-end, e, posteriormente, os resultados serão utilizados de forma organizada e compreensível. Essa colaboração entre os TFCs demonstra uma aplicação prática e eficiente do uso de Machine Learning para melhorar o tratamento de currículos em um cenário real.

4.4 Requisitos

Nesta secção encontram-se, portanto, descritos os principais requisitos identificados no âmbito do trabalho final de curso CV Capture Data.

Por convenção, a referência ao requisito será feita através do nome que identifica se se trata de um requisito funcional (RF) ou um requisito não-funcional (NF), e terminando com um número sequencial do requisito.

Por exemplo:

RF_01 – corresponde ao 1o Requisito de Negócio identificado.

NF_01 – corresponde ao 1o Requisito de Negócio Não-Funcional identificado.

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF_01] ou [NF_01] e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

Requisitos Funcionais – descrevem explicitamente as funcionalidades e serviços que a solução deverá apresentar. Documenta por exemplo: como a nova solução deve comportar-se em determinadas situações, e o que o sistema deve ou não fazer.

Requisitos Não-Funcionais – especifica todos os requisitos relacionados ao uso da nova aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e das tecnologias envolvidas (requisitos de hardware e software). Estes requisitos dizem respeito a como as novas funcionalidades descritas neste documento deverão ser entregues ao Cliente.

Nesta secção encontram-se, portanto, descritos os principais requisitos identificados no âmbito do trabalho final de curso CV Capture Data.

Tabela 1 - Requisitos Funcionais

ID	Designação	Observações	M/D (*)
RF_01	A aplicação deve ser capaz de captar texto relevante a partir de uma imagem de um currículo.	Para ajudar a processar currículos pré-existentes de utilizadores.	D
	A aplicação deve utilizar um modelo para efetuar a deteção de objetos na	Para ajudar a processar vários tipos de formatos de currículo, não só Templates.	D

RF_02	imagem, a fim de identificar as regiões de interesse.		
RF_03	A aplicação deve usar reconhecimento óptico de caracteres (OCR) para converter o texto dentro das regiões de interesse em texto legível para uma máquina.	Para facilitar o processamento de texto nos currículos em diferentes línguas.	D
RF_04	A aplicação deve utilizar um reconhecimento de entidades (NER) personalizado para identificar e extrair entidades específicas, tais como o nome do candidato, educação, experiência profissional, competências, etc.	Para facilitar a extração de informações de currículos em constante evolução (skills, educação, etc) e em diferentes línguas.	D
RF_05	A aplicação deve exportar as informações extraídas através de um endpoint (pedido REST)	Para ajudar na integração com outras aplicações.	M
RF_06	Num cenário de difícil acesso a dados, um mecanismo de etiquetar e coletar dados provenientes do input de utilizadores ao longo do tempo.	Por validar com orientação do trabalho.	D

Legenda (*):

M – Mandatário , requisito considerado indispensável para o funcionamento da solução.

D – Desejável, requisito que visa a facilidade de utilização da solução não sendo obrigatório para aceitação da solução.

Tabela 2 - Requisitos Não Funcionais

ID	Designação	Observações	M/D (*)
NF_01	A aplicação deve aderir às leis e regulamentos relevantes de proteção de dados.		M

NF_02	A aplicação deve ser capaz de processar currículos e ser capaz de extrair informações relevantes de currículos escritos em diferentes línguas.		D
NF_03	A aplicação deve ser capaz de extrair com precisão e eficiência a informação relevante. O sistema deve ser capaz de lidar com diferentes formatos e layouts de currículos.	Para facilitar o processamento de texto nos currículos em diferentes línguas.	D
NF_04	A aplicação deve ser concebida para proporcionar alta disponibilidade e tempo de resposta rápido.	Para facilitar a extração de informações de currículos em constante evolução (skills, educação, etc) e em diferentes línguas.	D
NF_05	A aplicação deve ser compatível com uma gama de plataformas de hardware e software, e deve ser concebida para funcionar com outros sistemas e aplicações.	Para ajudar na integração com outras aplicações.	M

Legenda (*):

M – Mandatário , requisito considerado indispensável para o funcionamento da solução.

D – Desejável, requisito que visa a facilidade de utilização da solução não sendo obrigatório para aceitação da solução.

4.5 Requisitos atendidos e justificativas

F_01 - A aplicação deve ser capaz de captar texto relevante a partir de uma imagem de um currículo. O requisito foi parcialmente cumprido, uma vez que não se chegou a obter dados reais em quantidades razoáveis para se atribuir o requisito como cumprido.

RF_02 - A aplicação deve utilizar um modelo para efetuar a deteção de objetos na imagem, a fim de identificar as regiões de interesse. O requisito foi cumprido.

RF_03 - A aplicação deve usar reconhecimento óptico de caracteres (OCR) para converter o texto dentro das regiões de interesse em texto legível para uma máquina. O requisito foi cumprido.

RF_04 - A aplicação deve utilizar um reconhecimento de entidades (NER) personalizado para identificar e extrair entidades específicas, tais como o nome do candidato, educação, experiência profissional, competências, etc. O requisito foi parcialmente cumprido, contudo este requisito tal como o F_01 não se chegou a obter dados reais em quantidades razoáveis para se atribuir o requisito como cumprido.

RF_05 - A aplicação deve exportar as informações extraídas através de um endpoint (pedido REST). O requisito foi cumprido.

NF_02 - A aplicação deve ser capaz de processar currículos e ser capaz de extrair informações relevantes de currículos escritos em diferentes línguas. O requisito foi cumprido.

NF_03 - A aplicação deve ser capaz de extrair com precisão e eficiência a informação relevante. O sistema deve ser capaz de lidar com diferentes formatos e layouts de currículos. O requisito foi cumprido.

NF_04 - A aplicação deve ser concebida para proporcionar alta disponibilidade e tempo de resposta rápido. O requisito foi parcialmente cumprido, pois a aplicação não foi concebida para proporcionar inferências de alta disponibilidade uma vez que para tal efeito seria necessário incorrer em custos financeiros desnecessários no ambito deste TFC.

NF_05 - A aplicação deve ser compatível com uma gama de plataformas de hardware e software, e deve ser concebida para funcionar com outros sistemas e aplicações. O requisito foi parcialmente cumprido, mas a aplicação ainda precisa de ajustes para melhorar a interoperabilidade.

4.6 Requisitos não atendidos e justificativas

Dois requisitos do projeto não foram completamente atendidos devido a circunstâncias específicas que impossibilitaram sua implementação no momento atual. São eles:

O requisito RF_06 envolve a criação de um mecanismo para etiquetar e coletar dados provenientes do input dos utilizadores ao longo do tempo. Infelizmente, este requisito não foi cumprido na fase atual do projeto devido a dificuldades de acesso a dados relevantes e incerteza em determinar a localização final da aplicação. A ausência de uma definição clara quanto à hospedagem da aplicação e à coleta de dados dificulta a implementação deste mecanismo de etiquetagem e coleta. Embora seja desejável ter esse mecanismo para aprimorar o projeto, sua implementação dependerá da validação e colaboração com terceiros.

O requisito NF_01 é mandatório e estabelece que a aplicação deve estar em conformidade com todas as leis e regulamentos relevantes de proteção de dados. No entanto, não foi possível atender a este requisito neste momento, pois atualmente não existem dados relevantes sendo manipulados pela aplicação. Portanto, não houve necessidade imediata de preocupação com

questões de privacidade e proteção de dados. É importante ressaltar que, embora esse requisito não tenha sido cumprido no momento atual, a aplicação ainda deverá ser desenvolvida posteriormente com base nos princípios de privacidade e segurança de dados para acomodar futura coleta e manipulação de dados em conformidade com as leis vigentes.

Para garantir a conformidade com as leis de proteção de dados e a implementação do mecanismo de etiquetagem e coleta de dados, será necessário estabelecer planos claros de validação e colaboração com todas as partes envolvidas. À medida que o projeto avança, esses requisitos devem ser revisados e atendidos adequadamente para assegurar a segurança, confiabilidade e conformidade do projeto com as normas regulatórias aplicáveis.

4.7 Pressupostos, Dependências e Restrições

Por convenção, a referência aos pressupostos será feita através do identificador (PR), terminando com um número sequencial do pressuposto.

Por exemplo:

PR_01 – corresponde ao 1º Pressuposto identificado no âmbito deste TFC.

Os pressupostos devem ser identificados com um identificador único. A numeração inicia com o identificador [PR_01] e prossegue sendo incrementada à medida que forem surgindo novos pressupostos.

Nesta secção encontram-se identificados os pressupostos que foram assumidos para a apresentação da nova solução.

Tabela 3 - Pressupostos

ID	Requisito(s) Associado(s)	Tema	Descrição
PR_01	RF_01	Dados – CGI	Assume-se como pressuposto que a solução terá acesso ao máximo de currículos disponíveis de colaboradores da CGI e informação sobre concursos para desenvolvimento da solução.
PR_02	RF_01	Representatividade	Assume-se como pressuposto que nesta fase do projeto que os dados fornecidos são representativos de amostras reais.
PR_03	RF_01, RF_02, RF_03, RF_04	Probabilidade de Acerto dos modelos de ML	Assume-se ainda neste contexto que alguns modelos apenas poderão ser treinados, quando existir a presença de um volume significativo de dados. A probabilidade de acerto destes

			modelos, deverá ser avaliada ao longo do projeto, sendo que, depreende-se que a mesma no início seja baixa, mas que com o decorrer do projeto, e o devido treino, a curva de aprendizagem tenda a crescer e a apresentar melhores taxas de acerto.
PR_04	RF_02, RF_04	Aprendizagem dos modelos de ML	Assume-se como pressuposto que os algoritmos dos modelos de Machine Learning não irão aprender automaticamente com os pedidos efetuados pelos utilizadores da solução. Tais pedidos serão registados na solução e arquivados para futuro retreino dos modelos e para a respetiva monitorização da eficiência dos mesmos.
PR_05	RF_01, RF_06	Disponibilização de serviços de armazenamento.	Assume-se como pressuposto, que a nova solução terá acesso a um serviço de armazenamento, para cada pedido criado via CV Tool, de forma a garantir o processamento de informação.
PR_06	RF_05	Invocação das componentes	Assume-se como pressuposto que a interface gráfica junto com a informação essencial para disponibilização da solução, será invocada a partir do ecrã via CV Tool.

4.8 Funcionalidades

Por convenção, a referência às funcionalidades descritas neste documento, será feita através do nome que as identifica (FN), seguido de um número sequencial da funcionalidade.

Por exemplo:

FN_01 – corresponde à 1ª Funcionalidade identificada no âmbito do trabalho CV Capture Data

As funcionalidades devem ser identificadas com um identificador único. A numeração inicia com o identificador [FN_01] e prossegue sendo incrementado à medida que forem surgindo novas funcionalidades.

FN_01 – Extração inteligente de informação

Esta funcionalidade permite a extração de informação relevante de um currículo fornecido, quando não existe informação prévia guardada.

Esta funcionalidade só será ativada quando: O currículo fornecido é uma imagem.

Ao executar esta funcionalidade no ecrã designado para o efeito, o utilizador é convidado a dar feedback sobre o resultado. A funcionalidade fica, entretanto, inativa.

A imagem é “Processada” e é retornado ao utilizador a informação relevante.

A partir desta funcionalidade é recolhida informação de negócio para retreinar aceitar a extração de informação (FN_06).

Informação recolhida (Modelos de dados relevantes)

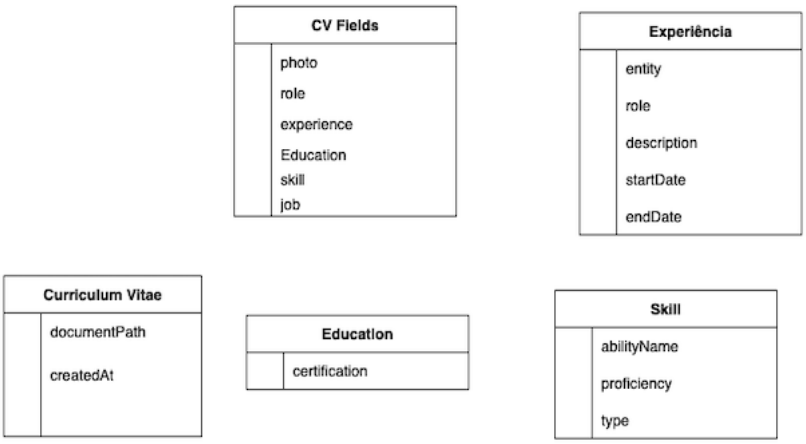


Figura 5 - Modelos de dados

Atores:

Todos os perfis.

Periodicidade de execução:

Real-Time.

Precedência de Execução:

Nenhuma.

Diagrama de Atividade:

O diagrama seguinte descreve o fluxo de um processamento em real-time de um curriculo.

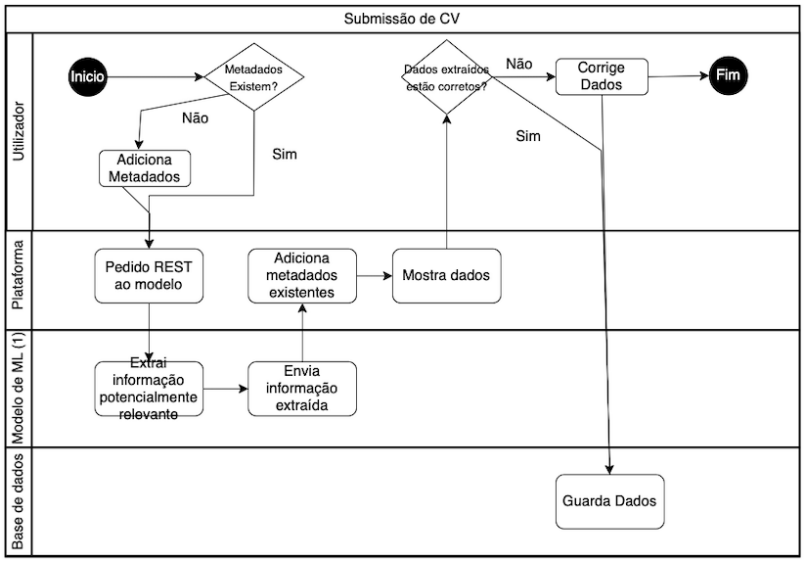


Figura 6 – Diagrama de atividades do processamento de imagem

FN_02 – Filtragem inteligente de informação

Esta funcionalidade permite a consulta das informações que se encontram guardadas na base do conhecimento que é o conhecimento extraído dos currículos.

Ao consultar a informação de um curriculo, não é possível editar a informação.

Atores:

Perfil corporativo.

Periodicidade de Execução:

Ocasionalmente (perfil restrito).

Precedência de Execução:

A executar após "**FN_01** – **Extração inteligente de informação**", descrito no presente documento ou por inserção manual de informação no sistema.

Diagrama de Atividade:

O diagrama seguinte descreve o fluxo de uma pesquisa ("search") de informações em curriculos.

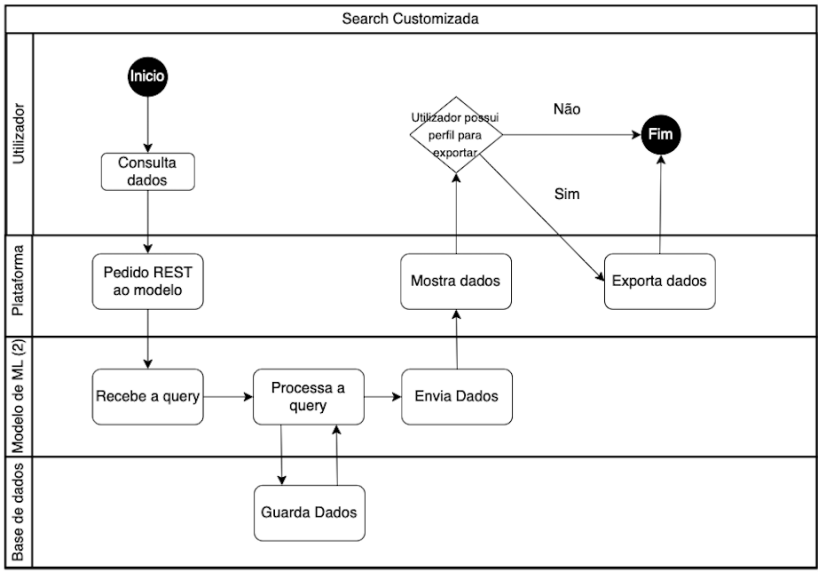


Figura 7 - Diagrama de atividades de pesquisa customizada

4.9 Mapas Aplicacionais

Visto que a CV Capture Data servirá para complementar o trabalho CV Tool, não existe ecrãs a desenvolver. Exceto os necessários para demonstrar o uso da endpoint.

4.10 Mock ups, story boards, etc…

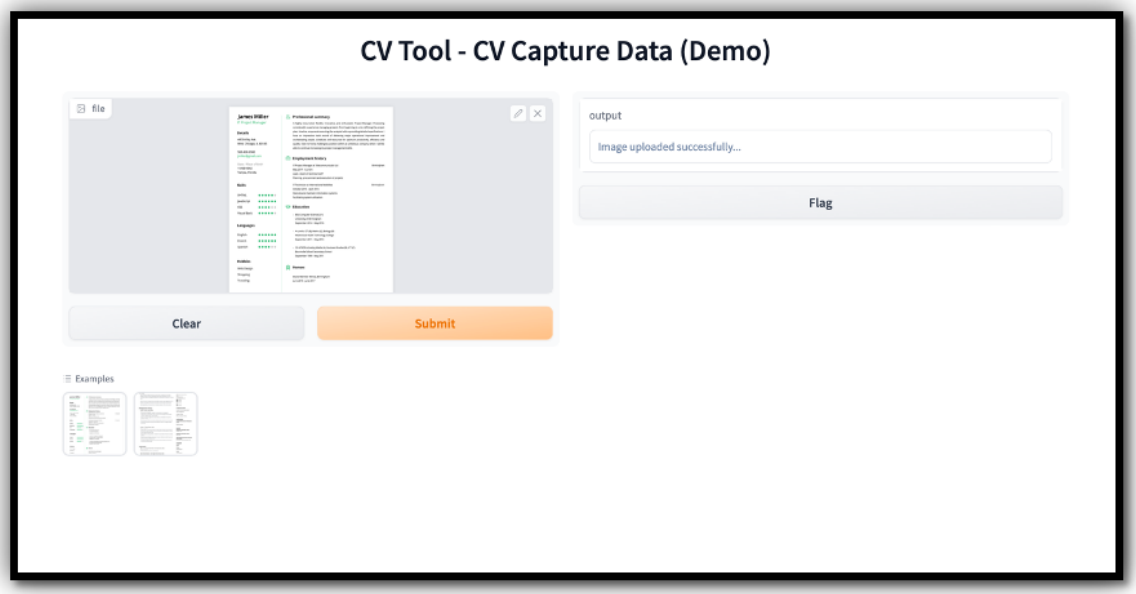


Figura 8 - Ecrã do demonstrador

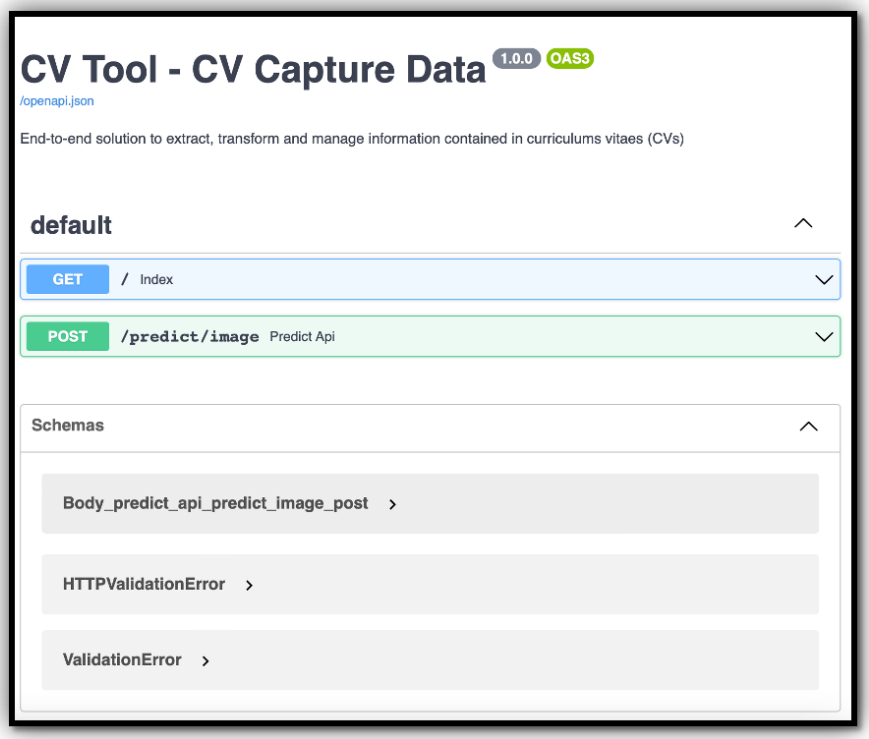


Figura 9 - Swagger API

5 Solução Proposta

5.1 Introdução

Para resolver o problema identificado anteriormente a solução proposta neste trabalho é a criação de uma pipeline para capturar dados automaticamente e com componentes de ML para fazer a gestão de currículos.

5.2 Arquitetura

Em paralelo à construção e análise do dataset final, para entregar software de qualidade e integrar rapidamente a solução será criada uma arquitetura de micro serviços para fazer a ingestão dos dados. A figura 4 representa um draft da arquitetura da pipeline visualmente

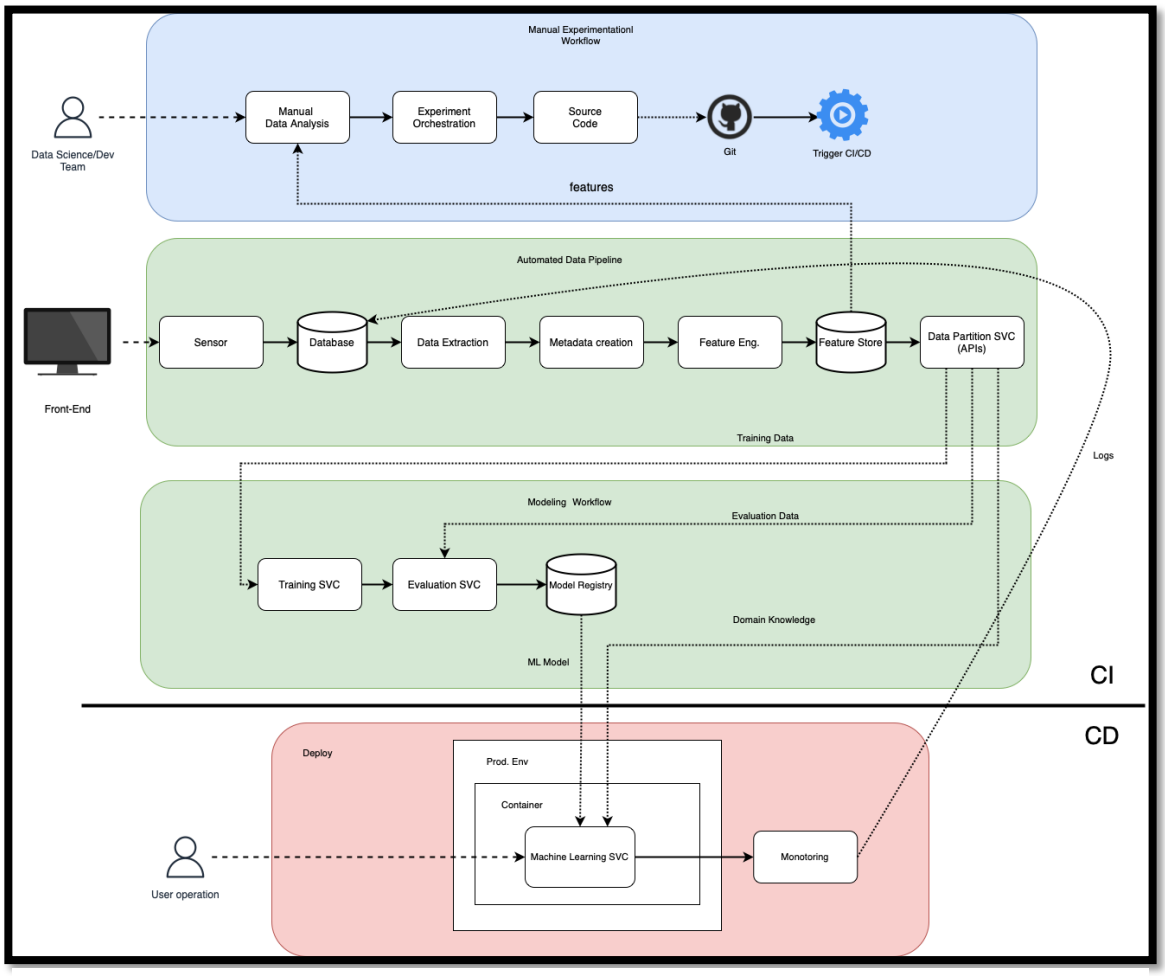


Figura 10 - Arquitetura da Solução

(ilustrada na figura a verde). No final o serviço de ánalise de CVs propriamente dito está apresentado como um serviço, ou seja, uma API. (ilustrado na figura a vermelho)

5.3 Tecnologias e Ferramentas Utilizadas

Para o desenvolvimento da solução será usado a linguagem de programação **Python**, pois é uma linguagem intuitiva e bastante usada na criação de modelos de ML.

Também serão usadas outras tecnologias, algumas apresentadas em baixo:

Pycharm Professional

Um IDE para desenvolvimento em python com suporte para testes automáticos e a licença é fornecida a alunos da lusófona.

Outras soluções open source recomendadas – **Visual Studio Code**

Label Studio

Um servidor com um UI e Templates para etiquetar dados a partir de inputs e exportar.

PostgreSQL

Uma base de dados relacional robusta e com características flexíveis para programar extensões, caso necessário.

Outras soluções recomendadas – **MySQL**

Feast

Ferramenta para construir feature stores, open source, mais conhecida e com mais extensões para machine learning. Semelhante a uma base de dados para guardar dados transformados para otimizar performance e facilitar experiências.

Outras soluções recomendadas – **Tecton**

Apache Airflow

Ferramenta padrão para orquestrar e automatizar workflows de pipelines.

Outras soluções recomendadas – **Luigi**

Pandas e/ou Numpy

Ferramentas padrão para análise e manipulação de dados.

Dependendo do problema não funciona com big data.

Outras soluções (Big Data) recomendadas – **Spark**

Jupyter/Google Colab Notebooks

Ferramentas para desenvolvimento de código com funcionalidades de visualização e partilha de código práticas.

Outras soluções recomendadas – **Python Scripts**

Pytorch

Biblioteca de python para deep learning que oferece mais controlo sobre o código.

Outras soluções recomendadas – **Tensorflow, Scikit-learn, etc...**

MLflow

Ferramenta para múltiplas funcionalidades da solução, nomeadamente para fazer controlo de experiências e registar modelos criados.

Outras soluções recomendadas – **Serviços Cloud Comuns**

FastAPI

Ferramenta de criação de APIs com funcionalidades extra para aumentar a performance, entre outras.

Outras soluções recomendadas – **Flask**

Docker & Kubernetes

Ferramentas de containarização padrão com boa integração com outras ferramentas.

Outras soluções recomendadas – **Serviços Cloud**

Jenkins

Ferramenta open source padrão para automatização de desenvolvimento de código.

Outras soluções recomendadas – **Github Actions**

Git & GitHub

Ferramenta open source padrão para versionamento de código.

Outras ferramentas

Dependendo do desenvolvimento, existe variedade de ferramentas que poderão ser testadas.

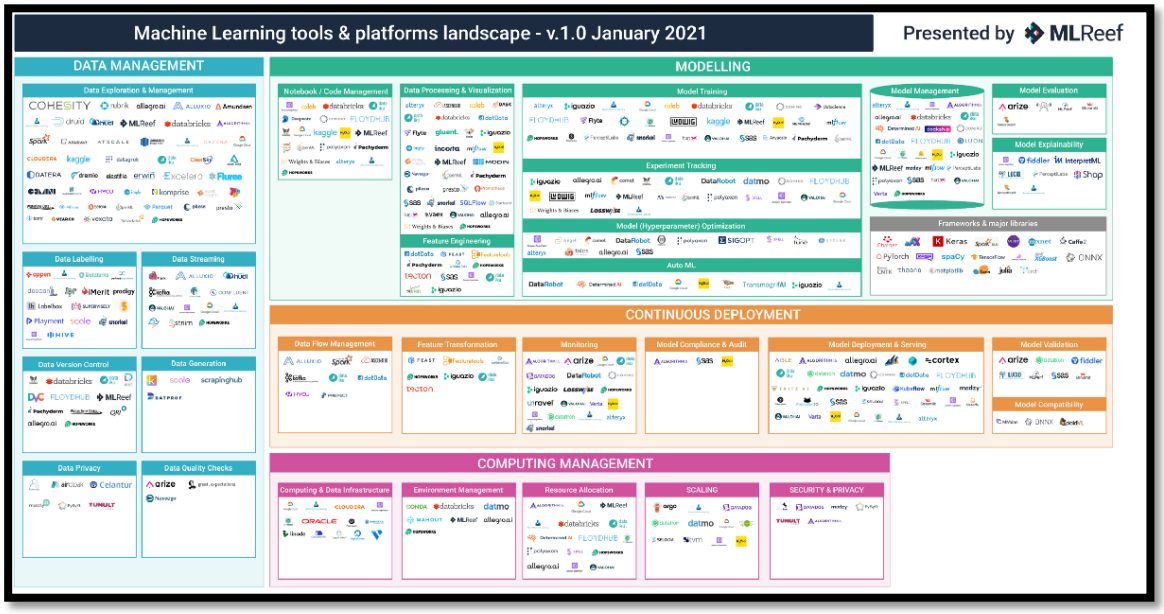


Figura 11 - Overview de algumas ferramentas para Machine Learning

5.4 Implementação

Devido a dependências com outros trabalhos finais de curso e entre componentes é necessário dividir o trabalho em etapas.

5.4.1 Criação do dataset

Descrito o processo de seleção de CVs na formulação do problema identifica-se a necessidade de obter currículos (CVs) representativos dos vários tipos de profissionais diferentes e respetivas descrições de perfis/trabalhos que se enquadram. No caso de condições específicas como templates, formatos, localização geográfica, língua do documento entre outras características também é necessário ter amostras representativas.

Este trabalho, tem como 1.ª dependência direta os dados e a sua qualidade, pelo que se os dados não existem devem ser criados, transformados ou etiquetados manualmente por especialistas de negócio numa primeira instância, ou ao longo do decorrer do projeto.

A falta de dados impossibilita experiências de possíveis soluções e a avaliação do trabalho.

O conjunto de dados final (dataset) de momento não se encontra disponível pelo que não é possível elaborar mais sobre a sua distribuição e características.

Uma vez obtido o dataset também será possível aferir sobre o tipo de aprendizagem a usar, supervised ou unsupervised learning.

Nota.: O presente trabalho está a ser realizado sobre um NDA (Non-Disclosure Agreement) pelo que poderá não ser possível elaborar ao detalhe sobre todas as informações e dados.

5.4.2 Componente de análise de currículos

Currículos não possuem um formato fixo. São documentos semiestruturados a gosto pessoal. Os mais comuns sendo em formato PDF e imagens. Após a ingestão de dados por parte do front-end (CV Tool) será necessário extrair informação dos currículos para texto e imagens independentemente do formato em que se encontrem.

Tal tarefa de extração pode ser realizada através de técnicas de visão computacional e NLP. Por exemplo, uma abordagem tradicional seria criar um modelo de deteção de imagem para reconhecer caracteres em imagens (OCR) antes de guardar a informação numa base de dados.

Existem algumas opções privadas e começam a surgir recentemente opções open source que apresentam bons resultados.

Pretende-se com este trabalho fazer uma análise mais profunda sobre a investigação realizada ao tema.

Esta componente depende da iniciação da construção da pipeline e criação de alguns dados.

5.4.3 Componente de busca em currículos

Uma vez guardado texto e imagens (dependente de componentes anteriores), numa base de dados de forma organizada e otimizada, podemos efetuar pesquisas sobre a informação ou efetuar outras operações sobre os dados.

Tal tarefa de pesquisa pode ser realizada através de técnicas de NLP. Por exemplo, uma abordagem tradicional seria criar um modelo com base em heurísticas para fazer pesquisas de currículos com base em requisitos de trabalhos.

Existem outras opções e pretende-se com este trabalho fazer uma análise mais profunda sobre a investigação realizada ao tema.

Esta componente depende da pipeline estar completa e funcional.

5.4.4 Componente de tradução

Pretende-se em paralelo com as outras componentes neste trabalho fazer uma análise mais profunda sobre a investigação realizada ao tema de tradução de texto.

5.5 Abrangência

O trabalho insere-se perfeitamente no âmbito do TFC, pois para a sua concretização espera-se construir uma solução funcional, robusta e fiável que acrescente valor para o mercado. Serão aplicadas habilidades relacionadas com o negócio escolhido, como funciona o trabalho dos recursos humanos, envolvendo todo o processo de criação e seleção de currículos profissionais, e competências informáticas para todo o processo de arquitetura, experimentação de algoritmos com ML e implementação de uma aplicação que permita a junção das funcionalidades anteriormente mencionadas.

6 Plano de Testes e Validação

Os testes são código que escrevemos, concebidos para falhar inteligivelmente quando o nosso código de desenvolvimento tem erros. Estes testes podem ajudar a apanhar alguns bugs antes de serem fundidos com a aplicação. Neste âmbito, está planeado a execução de dos seguintes métodos para testar e validar a aplicação.

- Usar ferramentas de teste, como o Pytest para fazer testes unitários e outros.
- Serão criados “doctests”, para verificar o código.
- Serão criados notebooks para auxiliar na criação da aplicação.
- Quando se começar a adicionar diferentes tipos de testes e a base de testes crescer, poderão ser necessárias ferramentas de cobertura para registar o que é verificado ou "coberto" por testes.
- Serão usadas ferramentas de Linting para uniformizar e estandardizar o código e promover boas práticas.
- Tudo será o máximo automatizado utilizando scripts, ferramentas de orquestração, CI/CD, entre outros.
- Serão criados testes personalizados e ferramentas para testar modelos de ML dependendo das componentes (Testes de fumo, expectations, entre outros...).

7 Métodos e Planeamento

Para pesquisar mais sobre as tecnologias utilizadas será feito recurso de conteúdo online e livros, bem como qualquer outro material relevante encontrado.

7.1 Calendário

Para a calendarização do projeto, foi feito um esquema gantt do cronograma com respetivos entregáveis:

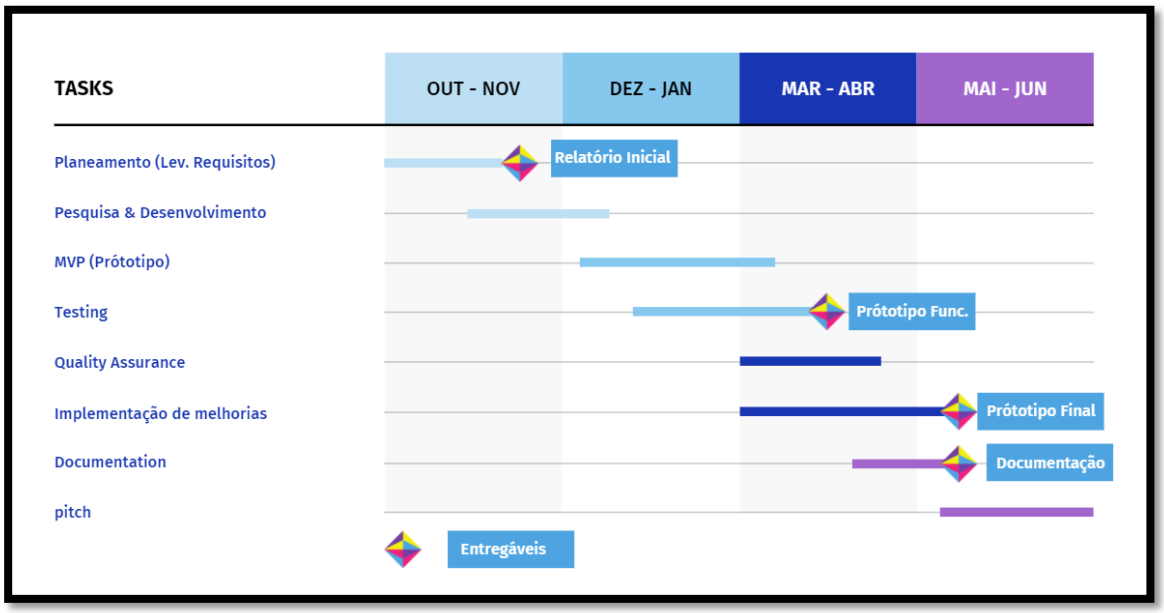


Figura 12 - Calendarização do projeto, Gantt Chart

7.2 Outros métodos

Para auxiliar na gestão do projeto, serão elaborados documentos com os procedimentos realizados (SOP – Standard Operation Procedures) em Markdown, bem como outra documentação relevante de novos desenvolvimentos.

7.3 Tarefas de Desenvolvimento (ML-Powered Lifecycle)

A construção de produtos alimentados por ML requer um processo fundamentalmente diferente em muitos aspetos do que o desenvolvimento de modelos ML numa fase de investigação ou em ambiente académico.

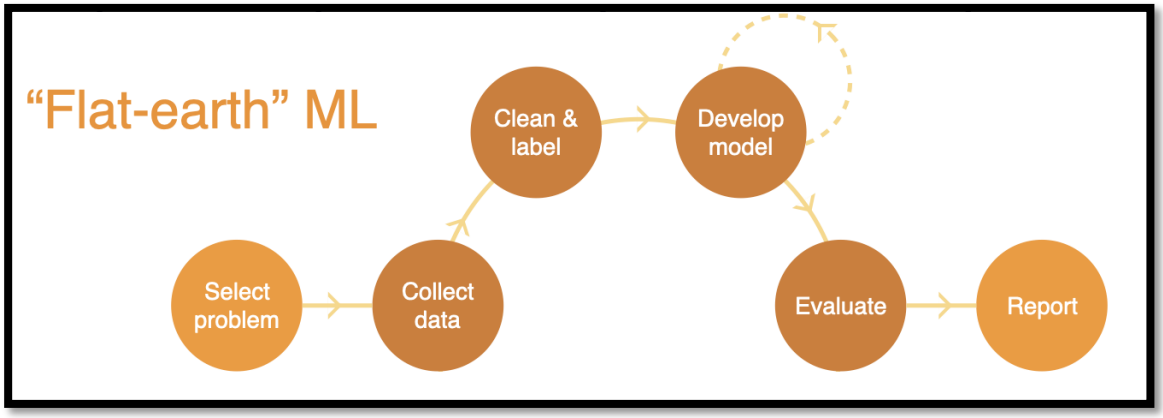


Figura 13 - Método tradicional de desenvolver ML

Tradicionalmente, constrói-se modelos de modo "flat" selecionando um problema, recolhe-se dados, limpa-se e etiqueta-se os dados, itera-se na fase de desenvolvimento de modelos até se ter um modelo que funcione bem no conjunto de dados recolhidos, avalia-se esse modelo, e apresenta-se os resultados no final.

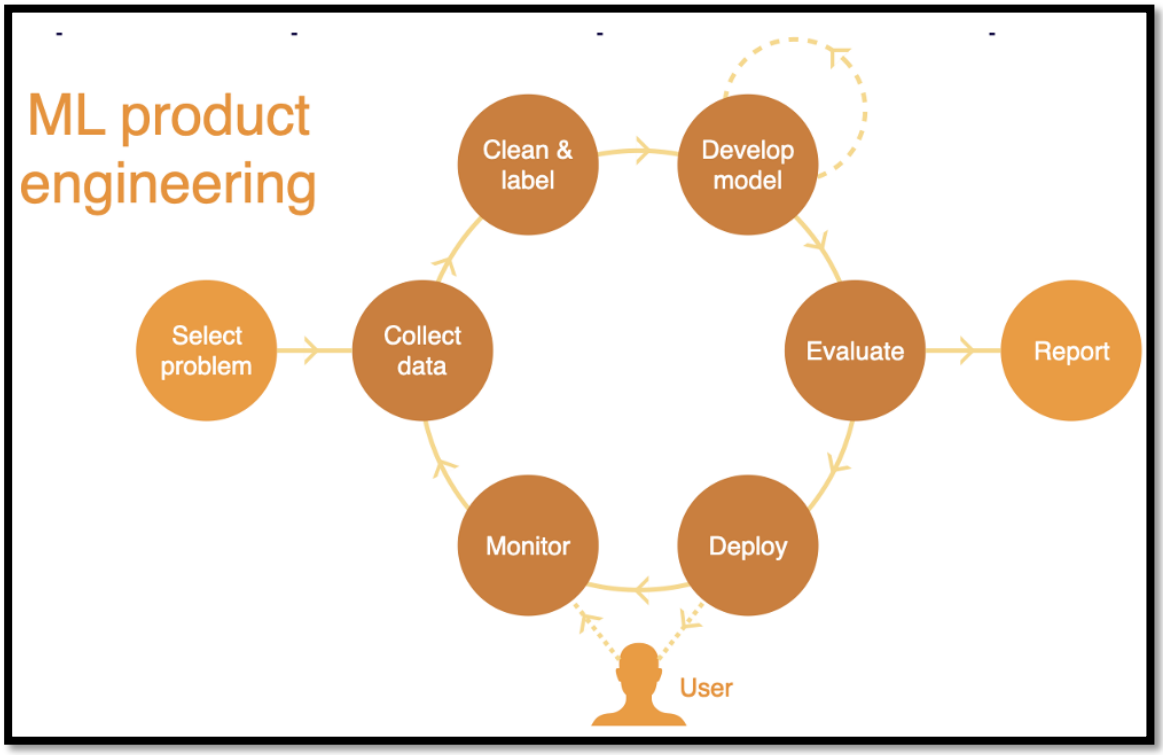


Figura 14 - Método "aceitável" de Desenvolver ML para produção

Mas os produtos aceitáveis pelos stakeholders de negócio, alimentados por ML, requerem um loop exterior onde, depois de se colocar o modelo em produção, se mede o desempenho do modelo quando interage com utilizadores reais. Depois, utilizam-se dados do mundo real para melhorar o modelo, criando um loop (Data Flywheel) que permite uma melhoria contínua.

Nesse sentido para cada componente alimentada por ML do projeto é necessário replicar as seguintes tarefas:

	TASK	STATUS	DUE DATE	ASSIGNED TO	HOURS BUDGETED	ACTUAL HOURS
Definição do problema	Sessões Stakeholders	Done	21/jul	Francisco	--	3
	Verificar viabilidade da solução	Done	21/jul	Francisco	--	1
	Workflow de utilizadores finais	Done	21/jul	Francisco	--	1
	Desenhar reward function	On Hold	21/jul	Francisco	--	0
	Desenhar critérios de sucesso	Done	21/jul	Francisco	--	1
Coleta de dados	Planear coleta de dados	Done	21/jul	Francisco	--	24
	Validar assunções	On Hold	21/jul	Francisco	--	24
	Obter dados	On Hold	21/jul	Francisco	--	24
	Etiquetar	On Hold	21/jul	Francisco	--	160
	Identificar problemas de recolha de dados	Done	21/jul	Francisco	--	2
Análise Exploratória de Dados	Inspeção de dados	On Hold	21/jul	Francisco	--	1
	Validar estrutura	On Hold	21/jul	Francisco	--	1
	Criação de dataset de aprovação (Golden Truth)	On Hold	21/jul	Francisco	--	0
	Estudo descritivo do dataset	On Hold	21/jul	Francisco	--	0
	Documentação EDA	In Progress	21/jul	Francisco	--	0
Processamento	Automatizar ETL	In Progress	21/jul	Francisco	--	0
	Preparar inputs para experiências	Done	21/jul	Francisco	--	2
	Documentar processamento	In Progress	21/jul	Francisco	--	0
Modelação	Train-Test-Split	In Progress	21/jul	Francisco	--	0
	Implementação Baseline	Done	21/jul	Francisco	--	60
	Implementação SOTA	Done	21/jul	Francisco	--	8
	Imprelementação 3rd-Party	Done	21/jul	Francisco	--	8
	Tunning de Hyperparametros	Not Started	21/jul	Francisco	--	0
	Documentar abordagens	In Progress	21/jul	Francisco	--	0
Avaliação	Comparação de performance	Not Started	21/jul	Francisco	--	0
	Análise de erro	Not Started	21/jul	Francisco	--	0
	Elaboração de relatórios de Desempenho	Not Started	21/jul	Francisco	--	0
Deployment	Identificar requisitos de UX/UI	On Hold	21/jul	Francisco	--	0
	Escolher arquitetura	Done	21/jul	Francisco	--	4
	Construir API	Done	21/jul	Francisco	--	1
	Resolver dependências	Done	21/jul	Francisco	--	1
	Otimizar performance	Not Started	21/jul	Francisco	--	0
	Criar mecanismos de feedback	On Hold	21/jul	Francisco	--	0
	Fazer deploy	Done	21/jul	Francisco	--	2
	Criar documentação	In Progress	21/jul	Francisco	--	0
Testes (em paralelo)	Logging	Not Started	21/jul	Francisco	--	0
	Estratégia de curação de novos dados	In Progress	21/jul	Francisco	--	0
	Trigger de retreino	Not Started	21/jul	Francisco	--	0
	Testes unitários	Not Started	21/jul	Francisco	--	0
	Testes de Fumo	Not Started	21/jul	Francisco	--	0
	ML test score	Not Started	21/jul	Francisco	--	0
	Linting	Not Started	21/jul	Francisco	--	0
	Comentar código	In Progress	21/jul	Francisco	--	0
	Optimização numérica	In Progress	21/jul	Francisco	--	0
	Profiling	Not Started	21/jul	Francisco	--	0
	Version control	Done	21/jul	Francisco	--	16

Figura 15 - Estimativa de Tarefas

8 Resultados

Os resultados obtidos durante o desenvolvimento da aplicação foram considerados bons, levando em conta o contexto de recursos limitados disponíveis para o projeto. Devido à ausência de dados disponíveis e a existência de diversas dependências externas ainda não resolvidas, as capacidades da aplicação foram devidamente demonstradas, mas a qualidade final do produto foi afetada. Apesar disso, os resultados alcançados até o momento servem como prova de conceito promissora, destacando o potencial da aplicação em processar currículos em formato de imagem para texto, bem como extrair informações relevantes, independentemente do idioma utilizado.

Embora os resultados obtidos sejam encorajadores, fica claro que mais trabalho e aprimoramentos são necessários para transformar a aplicação em uma solução aceitável para o uso em produção. O desafio dos diferentes layouts de currículos e idiomas pode ser superado com a implementação de técnicas de processamento de imagens e modelos de Machine Learning mais robustos, que permitirão uma melhor generalização e adaptação a cenários variados.

É importante ressaltar que o problema de fazer o matching entre a função do candidato e as informações extraídas dos currículos, infelizmente, não pode ser completamente ultrapassado sem acesso a uma base de dados relevante. Dados de referência são essenciais para realizar comparações precisas e, sem eles, a aplicação pode não atingir o nível de eficiência necessário para essa tarefa específica.

As metodologias desenvolvidas no âmbito deste projeto têm potencial para serem aplicadas em vários outros problemas relacionados a documentos, ampliando o escopo de utilização da tecnologia desenvolvida.

Outro aspecto significativo dos resultados é a possibilidade de integração da aplicação desenvolvida em diversas outras aplicações por meio de sua API. Esse recurso permite que outras soluções aproveitem as funcionalidades da aplicação para processar currículos e extrair informações valiosas, tornando-se uma ferramenta versátil e de grande utilidade para uma variedade de cenários e setores.

Em suma, os resultados alcançados representam um passo inicial sólido e promissor para a aplicação, evidenciando suas capacidades e potencialidades. Com a análise detalhada das imagens em anexo, é possível compreender melhor o desempenho e as limitações atuais da aplicação.

Bibliografia

[1] Noah gift e Alfredo Deza, *Practical MLOps: Operationalizing Machine Learning Models*, 1ª Edição, O'Reilly Media, Inc, 2021.

[2] Cathy Chen, Niall Richard Murphy, Kranti Parisa, D. Sculley e Todd Underwood, *Reliable Machine Learning*, 1ª Edição, O'Reilly Media, Inc, 2022.

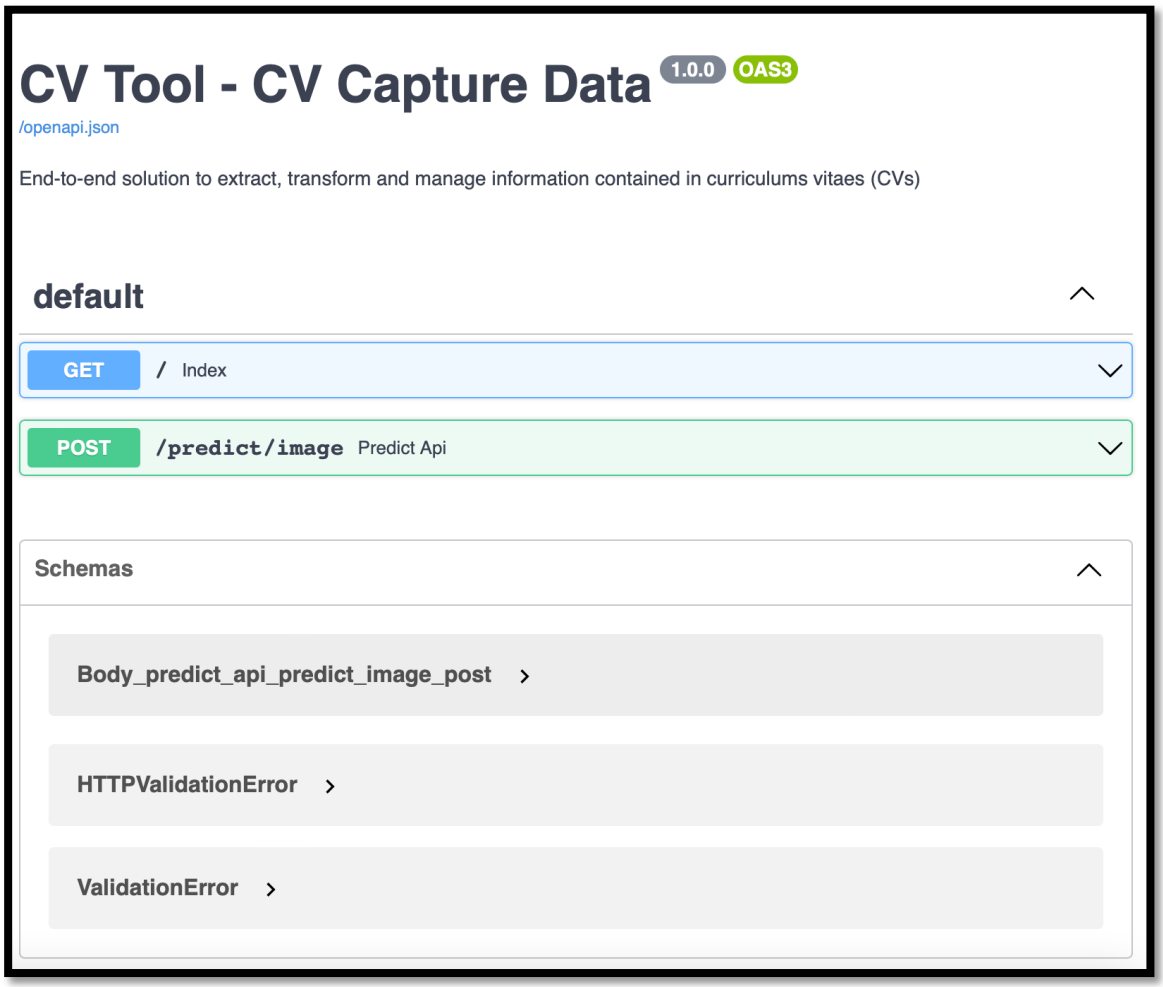
[3] Vedant Bhatia, Prateek Rawat, Ajit Kumar e Rajiv Ratn Shah, "End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT", acedido em nov 2022. Disponível em: https://doi.org/10.48550/arXiv.2109.06501

[4] Dor Lavi, Volodymyr Medentsiy and David Graus, "conSultantBERT: Fine-tuned Siamese Sentence-BERT for Matching Jobs and Job Seekers", acedido em nov 2022. Disponível em: https://doi.org/10.48550/arXiv.1910.03089

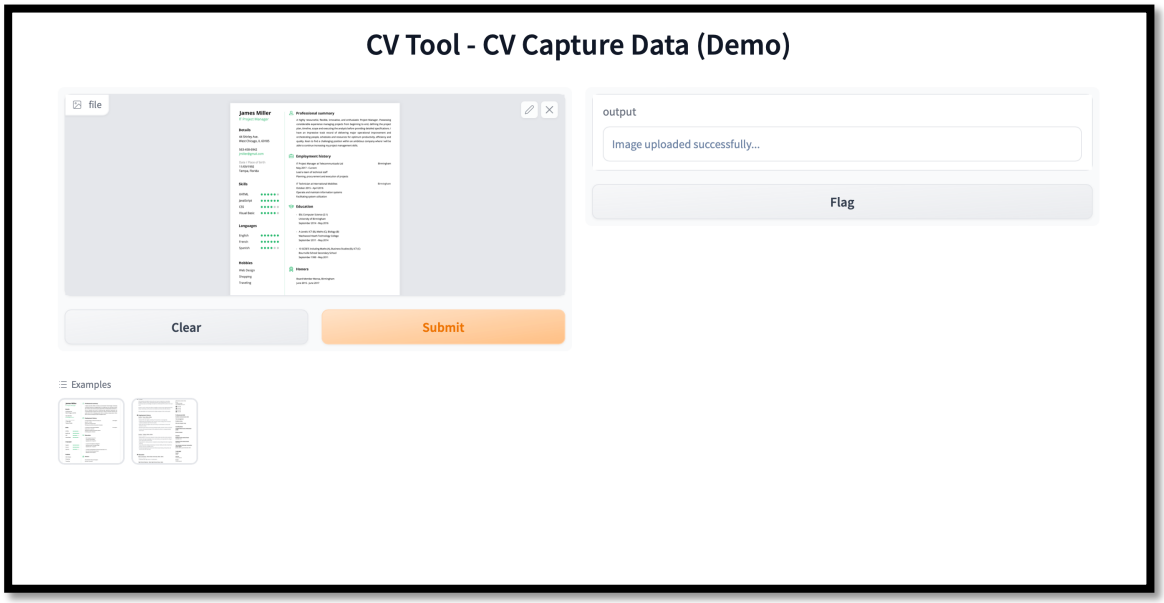
Anexos

Mock-Ups (Ilustrativos da solução)

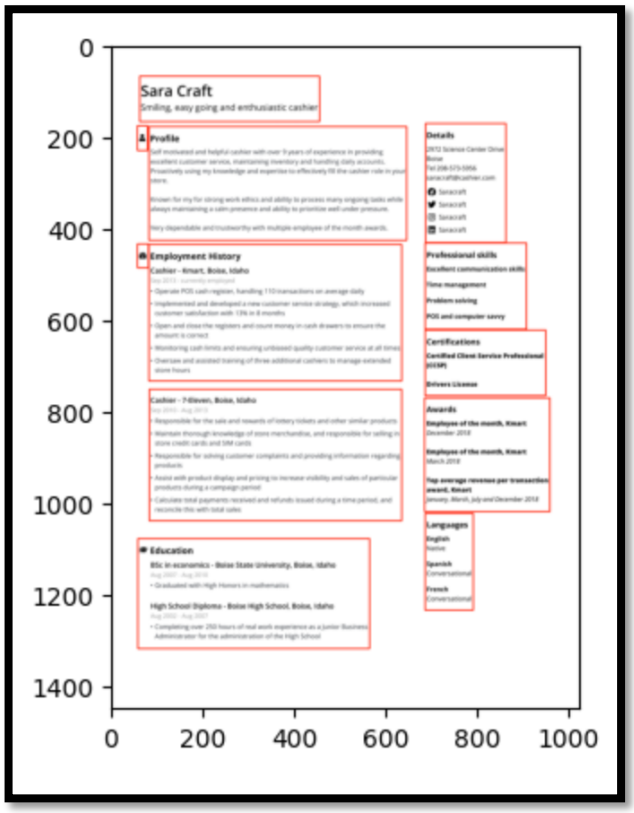
Anexo 1 - Swagger REST API (Documentação)



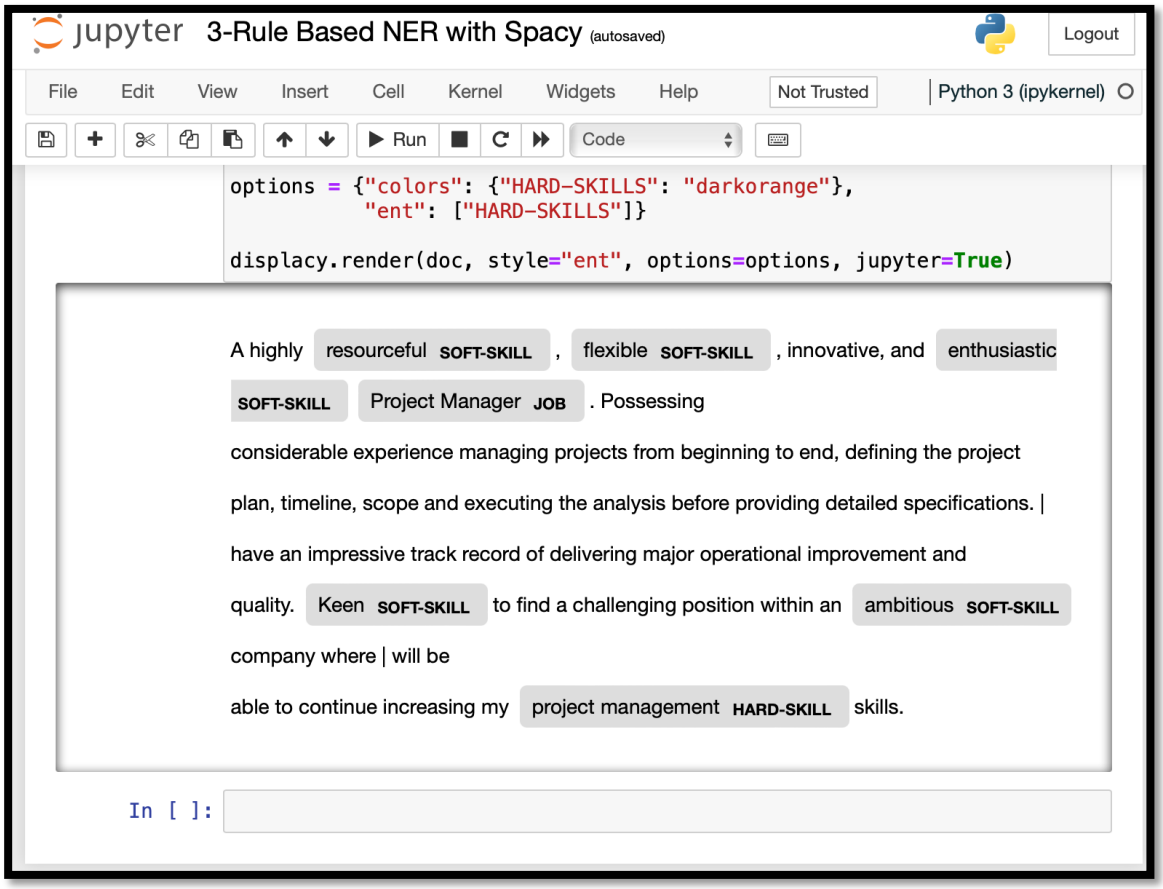
Anexo 2 - UI (Demo & Monitoring Tool)



Anexo 3 - Experiências sobre processamento de imagem e criação do Dataset



Anexo 4 - Experiência sobre extração de informação



Anexo 5 - Experiências Matching Candidato a Trabalho

	Match Percentage	Prediction
0	54.93	51.307096
1	40.86	34.897741
2	52.46	43.702701
3	37.42	41.078190
4	42.91	37.530975
5	50.82	49.236809
6	38.83	29.751871
7	46.52	38.297409
8	45.18	48.629953
9	56.70	54.928706
10	48.62	45.887242
11	13.21	34.981368
12	50.26	51.090152
13	47.51	27.700738
14	11.14	44.772802
15	50.86	39.047315
16	39.95	45.355459
17	35.77	22.783531
18	50.59	48.437389
19	4.81	25.667125
20	14.14	40.568645
21	44.96	38.111433
22	54.39	45.534947
23	46.38	38.702260
24	43.17	47.779990
25	46.27	49.771004
26	37.80	30.929884

Desafios Encontrados

Até ao momento não foi possível obter dados reais. Foram encontrados dados online para experiências o que funcionou para alguns casos de uso da aplicação e não funcionou para outros.

Glossário

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso