



Natural Language Querying

TIAGO GABRIEL DA SILVA

novembro de 2019



Natural Language Querying

Tiago Gabriel da Silva

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

**Orientador: Prof. Doutor Paulo Gandra de Sousa
Coorientador: Eng.º Ricardo Magalhães**

Júri:

Presidente:

Prof.^a Doutora Isabel Azevedo, DEI/ISEP

Vogais:

Prof. Doutor Nuno Escudeiro, DEI/ISEP

Para a Patrícia,
Para os meus Pais,
Para os meus Amigos

Resumo

O paradigma de interação entre Homem e Máquina tem vindo a mudar nos últimos anos. Se ao longo das últimas décadas, o ser humano tem vindo a interagir com o computador através da escrita (linha de comandos) ou das interfaces gráficas, mais recentemente, surge a interação por linguagem natural. Como potenciar a comunicação entre o Homem e os sistemas usados diariamente, usando linguagem natural? Recorrendo ao Processamento de Linguagem Natural, um campo de estudo ligado à Inteligência Artificial, que pode envolver técnicas de *Machine Learning* ou *Deep Learning*, torna-se possível transformar a linguagem do ser humano numa representação adaptada aos sistemas computacionais.

A presente tese debruça-se na conceção de um módulo de linguagem natural para interface com o produto Critical Manufacturing MES, que permita a consulta de informação relativa aos estados de um processo de fabrico. Neste âmbito, foi desenvolvido um protótipo que consiste num *chatbot* que assenta sobre o modelo concetualizado neste trabalho e que visa a compreensão da linguagem natural, assim como a extração de informação de contexto fabril.

Palavras-chave: Indústria 4.0, Inteligência Artificial, Interfaces de Linguagem Natural, *Machine Learning*, *Manufacturing Execution System*

Abstract

The interaction paradigm between man and machine has been changing in the last years. Over the last decades, humans have been interacting with the computer through writing (command line) or graphical interfaces. Recently, emerges the interaction through natural language. How to enhance the communication between man and the system used on daily basis, by using natural language? The usage of Natural Language Processing, a field of study of Artificial Intelligence, which may involve Machine Learning or Deep Learning techniques, allows the transformation of human language into a representation adapted to computation systems.

This thesis focuses on the design of a natural language module to integrate into the product named Critical Manufacturing MES, which enables consulting information regarding a manufacturing process. As result, a prototype has been developed, which consists of chatbot that is based on the conceptualized model in this work and that aims the understanding of natural language, as well as the extraction of information about the production state.

Agradecimentos

O trabalho desempenhado ao longo de quase um ano não seria possível sem a presença de várias pessoas, as quais marcam a minha vida todos os dias, e das quais tenho apoio incondicional.

Quero agradecer aos meus pais, pela educação, pelo apoio, pelo incentivo e principalmente, pelos valores que me foram inculcados, que fizeram de mim a pessoa que sou hoje. Alguns anos conturbados passaram e hoje tudo faz sentido – *I'm in hell without you, cannot cope without you two, shocked at the world that I see*. Obrigado.

Agradeço aos meus amigos, por aturarem as minhas longas “palestras”, por me ouvirem, na alegria e na tristeza, e por incentivarem o meu sucesso (por ordem alfabética): Diogo, Egidio, Francisco, Joel, Sérgio, Wilson. Grande abraço.

Para o Diogo, obrigado por seres o *brother from another mother*. Ligeiramente picuinhas, eu sei, mas não precisamos de palavras.

Deixo o meu agradecimento ao Instituto Superior de Engenharia do Porto, em particular aos professores que acompanharam o meu percurso e que foram uma inspiração.

Agradeço à Critical Manufacturing, em especial ao Engenheiro Ricardo Magalhães, pela incentivo dado neste projeto, pelas ideias e pelos desafios que se demonstraram difíceis, mas que com diálogo foram possíveis alcançar.

Ainda no contexto Critical Manufacturing, quero também deixar o meu agradecimento a todas as pessoas com quem tive o prazer de trabalhar. Quero deixar um agradecimento especial ao Rui Santos, por ter sido o meu mentor na jornada CMF e por me ter “dado asas”. Obrigado pelas longas conversas musicais, técnicas e dos momentos de riso. Com certeza que nos vamos cruzar, especialmente em muitos concertos! *Rock On!*

Ao meu orientador, Doutor Paulo Gandra de Sousa, que confiou em mim e no desempenho deste trabalho, que se demonstrou preocupado quando não dava notícias (peço desculpa, professor) e me ajudou a traçar o caminho seguido neste trabalho.

E, como sou uma pessoa que deixa o melhor para o fim, agradeço à Patrícia, minha amiga, confidente, apoiante número um, psicóloga e ouvinte nas horas vagas. Sempre acreditaste em mim, e continuas a fazê-lo, e não há palavras suficientes para descrever a minha gratidão – *Love of my Life*.

A todos, o meu mais profundo obrigado!

Conteúdo

Lista de Figuras	xiii
Lista de Tabelas	xv
Lista de Código	xvii
Lista de Acrônimos	xix
1 Introdução	1
1.1 Problema	2
1.2 Objetivos	2
1.3 Âmbito e Pressupostos	3
1.4 Critérios de Sucesso	3
1.5 Contribuições	4
1.6 Plano e Método de Trabalho	5
2 Contexto	7
2.1 A Empresa	7
2.2 O Produto	8
2.2.1 <i>Manufacturing Execution Systems</i>	8
2.2.2 Critical Manufacturing MES	10
2.3 Relevância do Problema	12
2.3.1 Na Indústria 4.0	12
2.3.2 No Critical Manufacturing MES	13
2.4 Síntese	14
3 Estado da Arte	15
3.1 Processamento de Linguagem Natural	15
3.1.1 História	15
3.1.2 Fundamentos	16
3.2 Interfaces de Linguagem Natural	16
3.2.1 Sistemas <i>Pattern Matching</i>	18
3.2.2 Sistemas Baseados em Sintaxe	18
3.2.3 Sistemas de Gramática Semântica	18
3.2.4 Sistemas de Representação Intermediária de Linguagem	19
3.3 Trabalhos de Referência	19
3.3.1 Aplicados no Panorama Fabril	19
3.3.2 Outras Aplicações	21
3.3.3 Sinopse	25
3.4 Ferramentas	25
3.4.1 NLTK	25
3.4.2 Stanford CoreNLP	26
3.4.3 spaCy	26
3.4.4 TensorFlow	26
3.4.5 Rasa	26
3.4.6 Amazon Lex	27

3.4.7	IBM Watson Assistant	27
3.4.8	Microsoft LUIS	27
3.4.9	Sinopse	27
3.5	Síntese	28
4	Conceção	29
4.1	Apreciação Prática	29
4.1.1	Gramática Baseada em Semântica	29
4.1.2	Pesquisa Semântica	30
4.1.3	Reconhecimento de Intenções e Entidades	31
4.1.4	Sinopse	33
4.2	Modelo Proposto	33
4.2.1	Visão Geral	33
4.2.2	Casos de Uso	34
4.2.3	Arquitetura	36
4.3	Síntese	37
5	Desenvolvimento	39
5.1	Especificação	39
5.1.1	Visão Geral	39
5.1.2	Arquitetura	40
5.2	Configuração	41
5.2.1	Respostas Estáticas	41
5.2.2	Intenções e Entidades	41
5.3	Implementação	42
5.4	Síntese	43
6	Validação	45
6.1	Compreensão da Linguagem Natural	45
6.2	Resposta às Questões-Chave	45
6.3	Síntese	46
7	Conclusões	49
7.1	Avaliação de Objetivos	49
7.2	Resposta ao Problema	49
7.3	Limitações e Trabalho Futuro	50
	Bibliografia	53
A	Plano de Trabalho Previsto	57
B	Análise de Valor	59
B.1	O Processo de Inovação	59
B.2	O <i>Fuzzy Front End</i> de Inovação	60
B.2.1	Identificação da Oportunidade	61
B.2.2	Análise da Oportunidade	61
B.2.3	Geração, Enriquecimento e Seleção de Ideias	62
B.2.4	Definição do Conceito	64
C	Protótipo	65
C.1	Configuração	65
C.2	Validação	68

Lista de Figuras

2.1	Ambiente <i>Manufacturing Execution System</i> (MES) e as suas camadas, baseado em Ugarte, Artiba e Pellerin (2009, p. 526)	9
2.2	Funções do <i>Manufacturing Execution System</i> (MES) e o seu enquadramento, extraído de Ugarte, Artiba e Pellerin (2009)	10
2.3	Arquitetura do Critical Manufacturing MES e tecnologias usadas, retirado de Critical Manufacturing (2019c)	11
2.4	Interface gráfica do Critical Manufacturing MES para acesso a informação do processo de fabrico	14
3.1	<i>Single Layer Perceptron</i> (SLP), extraído de Beysolow (2018)	16
3.2	Sistema não-contextual, extraído de A. Akula, Sangal e Mamidi (2013)	17
3.3	Sistema contextual, extraído de A. Akula, Sangal e Mamidi (2013)	18
3.4	Arquitetura do sistema Surukam-NLI, extraído de Ramesh, Jain, Sundaresan et al. (2016)	20
3.5	Arquitetura do sistema PRECISE, extraído de Popescu, Etzioni e Kautz (2003)	22
3.6	Arquitetura do sistema NaLIX, extraído de Yunyao Li, Yang e Jagadish (2005)	23
3.7	Arquitetura do sistema GINLIDB, extraído de El-Mouadib et al. (2009)	24
4.1	Amostra de teste com <i>chatbot</i> desenvolvido usando Microsoft LUIS	32
4.2	Visão da solução desejada para o produto	33
4.3	Conceitos associados ao reconhecimento de intenções e entidades e os respetivos relacionamentos	34
4.4	Áreas funcionais de sistema	35
4.5	Casos de uso identificados para a área funcional QA	35
4.6	Arquitetura do protótipo, apresentando um vista genérica e uma mais específica do componente <i>NL Engine</i>	36
4.7	Comunicação entre os componentes do módulo de linguagem natural	38
5.1	Vista geral da interação entre o <i>chatbot</i> e os serviços disponibilizados	40
5.2	Arquitetura do protótipo, seguindo o modelo proposto	40
5.3	Esquema referente à comunicação do <i>chatbot</i> com o Microsoft LUIS	43
5.4	Diagramas referentes à estrutura do protótipo na extração, transformação e carregamento da informação de domínio	44
6.1	Imagens referentes à avaliação de intenções e entidades do protótipo	46
6.2	Imagens referentes aos resultados esperados pelo protótipo, dadas as diferentes perguntas	47
6.3	Imagens referentes às respostas dadas pelo protótipo, dadas diferentes perguntas	48
A.1	Diagrama de <i>Gantt</i> referente ao planeamento do projeto	57
B.1	O processo de inovação, extraído de P. A. Koen et al. (2002)	59
B.2	A representação do modelo <i>New Concept Development</i> (NCD), extraído de P. A. Koen et al. (2002)	60
B.3	Número de artigos de Processamento de Linguagem Natural (PLN) pesquisados por ano, extraído de Liu, Yan Li e Thomas (2017)	61

B.4	<i>Mindmap</i> das ideias e conceitos gerados	63
C.1	Definição das entidades esperadas	65
C.2	Intenções definidas, contendo as expressões e respectivas entidades	66
C.3	Continuação das intenções definidas, contendo as expressões e respectivas entidades	67
C.4	Outras imagens relativas à avaliação de intenções e entidades do protótipo	68

Lista de Tabelas

3.1	Sumário dos trabalhos estudados	25
5.1	Intenções e entidades dada a expressão de exemplo	41

Lista de Código

4.1	Excerto de uma gramática, extraído de Bird, Klein e Loper (2009)	29
4.2	Excerto da definição semântica da frase que lida com listagem de produtos	30
5.1	Excerto do JSON devolvido pelo Microsoft LUIS	42

Lista de Acrônimos

API	<i>Application Programming Interface.</i>
ATN	<i>Augmented Transition Network.</i>
CPS	<i>Cyber-Physical System.</i>
DL	<i>Deep Learning.</i>
ERP	<i>Enterprise Resource Planning.</i>
FFE	<i>Fuzzy Front End.</i>
HTTP	<i>Hypertext Transfer Protocol.</i>
IA	Inteligência Artificial.
IHC	Interação Humano-Computador.
ILN	Interface de Linguagem Natural.
ILNBD	Interface de Linguagem Natural para Bases de Dados.
IoS	<i>Internet of Services.</i>
IoT	<i>Internet of Things.</i>
JSON	<i>JavaScript Object Notation.</i>
MES	<i>Manufacturing Execution System.</i>
ML	<i>Machine Learning.</i>
NCD	<i>New Concept Development.</i>
NPD	<i>New Product Development.</i>
OWL	<i>Web Ontology Language.</i>
PLN	Processamento de Linguagem Natural.
RDF	<i>Resource Description Framework.</i>
SLP	<i>Single Layer Perceptron.</i>
SQL	<i>Structured Query Language.</i>
UML	<i>Unified Modeling Language.</i>
XML	<i>Extensible Markup Language.</i>

Capítulo 1

Introdução

Num mercado crescentemente competitivo e exigente, a necessidade de inovar, de obter vantagem competitiva e simultaneamente, tornar os processos industriais simples e altamente eficazes, recorrendo às tecnologias mais atuais, abrem caminho a uma nova mudança. O fenómeno da Indústria 4.0 surge como a nova (quarta) revolução industrial, baseando-se nas mais recentes tecnologias, que incluem os sistemas ciber-físicos, a *Internet of Things* (IoT) e a *Internet of Services* (IoS), as quais se baseiam na comunicação através da Internet, permitindo uma interação contínua e partilha de informação entre humanos, entre máquinas e entre o ser humano e máquina (Roblek, Meško e Krapež 2016).

A Indústria 4.0 assenta numa variedade de conceitos fundamentais, de diferentes áreas de conhecimento, nomeadamente a noção de *Smart Factory*¹, a capacidade de auto-organização através da descentralização dos sistemas produtivos, e a interação entre o mundo físico e o digital (Lasi et al. 2014, Fundamental Concepts, p.240). No entanto, é a capacidade de adaptação à necessidade humana, principalmente a Interação Humano-Computador (IHC), que se explora no presente trabalho.

Segundo Green, Heer e Manning (2015, p.1) “as áreas de Inteligência Artificial (IA) e Interação Humano-Computador (IHC) estão, cada vez mais, a influenciar-se mutuamente. Alguns sistemas amplamente usados como o Google Translate, Facebook Graph Search e RelatelQ, escondem a complexidade de sistemas de larga escala de IA através de interfaces intuitivas.”². Apesar de terem propósitos diferentes, ambas as áreas se complementam, na medida em que se focam na relação entre ser humano e máquina. Se a IA tem como objetivo emular o intelecto humano, já a IHC foca-se em abordagens empíricas de usabilidade e fatores humanos, que influenciam a forma como os utilizadores interagem com o computador (Green, Heer e Manning 2015).

A capacidade dum sistema interpretar a linguagem dos seres humanos e apresentar a informação de uma forma adequada, principalmente no contexto da Indústria 4.0, destaca-se como um fator impulsionador da adaptabilidade do mundo digital à necessidade humana. Nesse sentido, a área de Processamento de Linguagem Natural (PLN), a qual se debruça na capacidade dos computadores “entenderem” a linguagem humana (Beysolow 2018, p.1), permite construir ferramentas capazes de definir ações, extrair conhecimento dum sistema e apresentá-lo num formato adequado, a partir de conteúdo textual especificado pelo utilizador, de acordo com a sua própria linguagem. Portanto, este trabalho foca-se no estudo do PLN aplicado às Interfaces de Linguagem Natural (ILN), e na definição de uma abordagem e desenvolvimento dum protótipo que permita interação com um sistema, através de linguagem natural, possibilitando consulta de informação acerca de estados de processos de fabrico.

O capítulo atual faz um pequeno enquadramento do trabalho, apresenta o problema, os objetivos e âmbito da tese, os critérios de sucesso definidos, as contribuições do trabalho para a área e o plano de trabalho e respetivo método a ser seguido.

¹Fábrica Inteligente, equipada com sensores, atuadores e sistemas autónomos, permitindo assim um controlo autónomo de processo.

²Tradução livre do autor. No original “The fields of artificial intelligence (AI) and human-computer interaction (HCI) are influencing each other like never before. Widely used systems such as Google Translate, Facebook Graph Search, and RelatelQ hide the complexity of large-scale AI systems behind intuitive interfaces.”.

Posteriormente, no Capítulo 2, contextualiza-se o problema, fazendo uma descrição da empresa e negócio, e aborda-se o propósito e relevância da resolução do mesmo.

O Capítulo 3 descreve o estado da arte, introduzindo os conceitos importantes de explorar, os trabalhos de referência e outros trabalhos relevantes na área, bem com ferramentas tipicamente usadas no enquadramento do PLN, perspetivando estratégias e possíveis abordagens.

No Capítulo 4 expõe-se o processo de experimentação prática e o modelo proposto como solução, revelando a visão, os requisitos identificados e arquitetura prevista.

Seguidamente, os Capítulos 5 e 6 abordam, respetivamente, o processo de desenvolvimento do protótipo com base no modelo proposto e a validação deste, tendo em conta os critérios de sucesso estipulados.

Por fim, o Capítulo 7 anuncia as conclusões do trabalho realizado, inferindo sobre os objetivos atingidos e a resposta ao problema, evidenciando-se limitações da solução e problemas por explorar.

1.1 Problema

O conceito de *Manufacturing Execution System* (MES), um sistema que, além de gerir as operações dum determinado processo fabril, mantém dados relativos às diversas etapas inerentes ao processo em questão, está intrinsecamente relacionado com a Indústria 4.0, uma iniciativa que se destina a criar fábricas inteligentes, usando tecnologias como os *Cyber-Physical Systems* (CPS), a IoT e *Cloud Computing* (Zhong et al. 2017). O Critical Manufacturing MES é um destes sistemas. Contudo, a capacidade de adaptação às características dos utilizador é um requisito complexo, que nem sempre é passível de ser cumprido. Isso pode dificultar o utilização do produto, numa perspetiva de acesso a informação relevante para o processo e de apoio à decisão (Zhong et al. 2017). Por outras palavras, se o utilizador pretende efetuar uma determinada pesquisa, necessita de conhecer os detalhes da ferramenta a usar, ao invés de simplesmente “pedir” ao sistema que lhe devolva o resultado. Posto isto, levanta-se a seguinte questão: *Será que o uso de linguagem natural na interação com o produto pode melhorar a sua usabilidade e simplificar processo de apoio à decisão?*. Partindo desta questão, procura-se um módulo de linguagem natural para interface com o Critical Manufacturing MES.

1.2 Objetivos

Com este trabalho pretende-se a conceção de um módulo de linguagem natural. Para tal, também se objetiva o desenvolvimento de um protótipo, baseado no modelo concebido, capaz de compreender três níveis de linguagem natural (aplicada ao contexto do produto): adimensional (sem entidades de domínio associadas), unidimensional (lida com uma entidade de domínio) e multidimensional (lida com múltiplas entidades de domínio).

Para os objetivos mencionados esperam-se as seguintes tarefas:

1. *Contextualizar o problema numa perspetiva de negócio* – análise detalhada do problema, as implicações que tem para negócio, para o produto MES e para os seus utilizadores, descrevendo a relevância do problema e valor intrínseco à sua resolução (Capítulo 1 e 2);
2. *Estudar soluções disponíveis no mercado e/ou ferramentas de processamento de linguagem natural* – obtenção de informação da área de conhecimento envolvida, de soluções semelhantes, trabalhos de referência e de ferramentas tipicamente usadas na implementação de soluções análogos (Capítulo 3);
3. *Definir abordagem considerando as diversas opções apresentadas* – comparação e avaliação das diversas opções identificadas, concluindo acerca do caminho a seguir (Capítulo 3 e 4);

4. *Desenvolvimento de prova de conceito* – implementação do protótipo de acordo com o modelo conceitualizado (Capítulo 4 e 5);
5. *Validação da prova de conceito* – concluir acerca do protótipo desenvolvido, face aos critérios de sucesso estipulados (Capítulo 6).

1.3 Âmbito e Pressupostos

Embora os objetivos estejam definidos, surge a necessidade de explicitar sucintamente o âmbito deste trabalho, bem como os pressupostos a ter em consideração. Por conseguinte, os seguintes assuntos serão abordados:

- A contextualização do problema da empresa com a prova de conceito a ser desenvolvida, o seu enquadramento com a Indústria 4.0 e utilidade para o cliente final;
- Os conceitos teóricos e adversidades inerentes ao problema, ainda que explorados de uma forma genérica, evitando abordar pormenores ou especificidades do tema;
- A apresentação e explicação dos exemplos de resolução de problemas semelhantes por parte de terceiros – trabalhos de referência –, fazendo um levantamento das características relevantes para este projeto;
- As ferramentas disponíveis e relevantes para este contexto, passíveis de ser aplicadas na solução final;
- O método científico e processo de engenharia adotado na busca duma abordagem para resolução do problema em questão.

Por outro lado, alguns tópicos são demasiado amplos para serem explorados, ou simplesmente não se enquadram nos objetivos desta tese, pelo que não serão abordados:

- O enquadramento do problema com outros MES. Apenas é contemplada a realidade do problema no contexto do Critical Manufacturing MES;
- As ferramentas para linguagem natural que não mostrem evidências de relevância para o problema;
- A inclusão de diferentes domínios no protótipo desenvolvido.

Quanto aos pressupostos deste trabalhos, enunciam-se a seguir.

- O termo “domínio” é empregue ao longo do texto para denotar um conjunto de características que descrevem uma família de conceitos comuns a um determinado processo. Por exemplo, duas empresas que produzem equipamentos médicos, apesar de poderem ter processos de fabrico diferentes, abordam o mesmo domínio;
- Assume-se o inglês como a língua a aplicar no protótipo, pois é a língua principal do Critical Manufacturing MES. Nesse sentido, o inglês abrange todo o tipo de interação com o protótipo, incluindo as questões a serem avaliadas;
- Também se assume o inglês para os artefactos produzidos, pelo motivo evidenciado no ponto anterior e pelo facto de ser a linguagem mais usada pela comunidade de desenvolvimento.

1.4 Critérios de Sucesso

A avaliação do resultado final é imprescindível para a concluir acerca do sucesso do trabalho, permitindo perceber se a conjectura fundada a respeito da prova de conceito é aceite. Desse modo, apresentam-se os critérios de sucesso a serem considerados:

1. *O protótipo desenvolvido é capaz de compreender a linguagem natural* – tem a capacidade de identificar os metadados associados à pergunta colocada, ou seja, de identificar linguagem natural adimensional, unidimensional e multidimensional;
2. *O protótipo dá resposta às questões que lhe são colocadas* – que implica responder às questões-chave, esperando que resposta fornecida seja semelhante à resposta que seria esperada.

Relativamente às questões-chave abordadas no ponto 2, prevêem-se que estejam relacionadas com o domínio do produto. Alguns exemplos são:

- *What's the total of dispatch operations per week during October?* – *i.e.*, Quantas operações *dispatch* foram executadas por semana, durante o mês de outubro?;
- *How many trackout operations were executed by product, on September, per shift?* – *i.e.*, Quantas operações *track-out* foram executadas, por produto e turno, durante o mês de setembro?;
- *What's the average primary quantity of trackin operations on burnin step, on september?* – *i.e.*, Qual a média de quantidade primária, no passo *burnin* do processo, por turno, no mês de Setembro?;
- *How many materials, by operation and on wire inspection step, have primary quantity inferior to 1000?* – *i.e.*, Quantos materiais, por operação e no passo *wire inspection*, têm a quantidade primária inferior a 1000?.

As questões apresentadas são exemplos fornecidos pelo supervisor do projeto e correspondem aos tipos de pesquisa mais relevantes no Critical Manufacturing MES.

Relativamente à metodologia de validação dos critérios os critérios mencionados, optam-se por testes manuais de interação com a plataforma subjacente e também com o protótipo desenvolvido.

1.5 Contribuições

O trabalho desenvolvido pretende providenciar uma abordagem de resolução do problema apresentado. Não se aspira fornecer uma solução definitiva, espera-se sim, contribuir com conhecimento de carácter teórico e prático (*i.e.*, um protótipo), que possibilite a integração futura de uma nova funcionalidade num produto já existente. No decorrer deste trabalho foram abordados temas relativos a MES e a IA, especificamente PLN e ILN. Resumidamente, as contribuições esperadas são a seguir mencionadas:

- Estado da arte no domínio de Processamento de Linguagem Natural e sistemas análogos à solução a desenvolver;
- Definição de um modelo que possibilita a “conversão” da linguagem natural em informação pertinente para o utilizador do sistema;
- Especificação e desenvolvimento do protótipo, considerando uma possível integração com o sistema Critical Manufacturing MES.
- Documentação dos requisitos de sistema, incluindo análise e desenho, constando os respetivos artefactos de *Unified Modeling Language* (UML);

De uma forma geral, é realçada a contribuição para o avanço do conhecimento no domínio da IA, mais especificamente na área das ILN, aplicada ao contexto dos sistemas MES.

1.6 Plano e Método de Trabalho

Um plano de trabalho é uma ferramenta importante na gestão de qualquer projeto, na medida em que descreve as fases que o compõem e as tarefas inerentes. Ele está sujeito a alterações ao longo do tempo de vida do projeto, pelo que o plano definido inicialmente pode ser encontrado no Apêndice A. Relativamente às fases do trabalho, a informação sucinta de cada uma é aqui explanada:

- *Conceção* – engloba as tarefas que relacionadas com o problema, o seu enquadramento, o estudo do valor e estado da arte. Ou seja, a visão do projeto;
- *Análise* – nesta fase, faz-se um estudo exploratório e de carácter empírico de forma a experimentar a aplicação de diferentes abordagens. Definem-se os casos de uso e arquitetura considerada para o modelo. A presente fase consiste no estudo e preparação para a aplicação do modelo no contexto prático;
- *Desenvolvimento* – desenvolve-se o protótipo com base no modelo conjecturado na fase anterior, envolvendo um período de experimentação;
- *Validação* – avalia-se a solução com base nos critérios de sucesso definidos e consequentemente, podem-se registar melhorias e retirar as devidas conclusões;
- *Documentação* – engloba a escrita da tese como veículo de transmissão de conhecimento obtido.

Quanto ao método de trabalho a seguir na realização deste trabalho, são considerados os seguintes passos:

1. *Revisão de literatura disponível sobre o contexto do problema* – com o objetivo de perceber o estado atual do Critical Manufacturing MES e as implicações que o módulo pode trazer, assim como concluir acerca da relevância do problema e do valor da solução para o produto;
2. *Revisão de literatura existente acerca de PLN e ILN, paradigmas arquiteturais relacionados e trabalhos de referência* – adquirir conhecimentos sobre o estado do PLN e das ILN, quais os trabalhos de referência na área, outros também relevantes, técnicas e ferramentas usadas, identificando aspetos relevantes para o trabalho;
3. *Idealização do modelo* – depois da análise dos conhecimentos adquiridos com as revisões realizadas nos passos descritos anteriormente, parte-se para a experimentação de diversas abordagens, definição dos casos de uso e arquitetura prevista;
4. *Implementação do protótipo e validação* – o foco é pôr em prática a solução concetualizada nas fases anteriores. Após a implementação, valida-se a mesma, de acordo com os critérios de sucesso definidos;
5. *Elaboração da documentação* – por fim, passa-se à escrita do presente documento e de documentos de suporte, baseando-se nas observações, nas experiências e conclusões obtidas ao longo do projeto.

Capítulo 2

Contexto

A contextualização do problema é necessária na medida em que contribui para a sua compreensão e resposta. Neste capítulo apresenta-se a empresa que visa ter uma solução para o problema exposto, fazendo uma breve descrição do seu negócio e do seu produto. De seguida, explicita-se a sua relevância para Indústria 4.0 e para o Critical Manufacturing MES, procurando perceber as motivações que levam à necessidade de resolução, estabelecendo-se o caminho para o estudo de problemas da mesma natureza.

2.1 A Empresa

A Critical Manufacturing é uma empresa fundada em 2009, com sede e centro de engenharia na Maia (Porto, Portugal), subsidiárias em Dresden (Alemanha), Suzhou (China), Austin (Estados Unidos da América) e um escritório comercial em Taiwan. O seu objetivo é proporcionar à indústria uma solução de gestão e controlo de produção, procurando reduzir os custos, flexibilizar o processo para satisfazer a procura e capacitar a organização de uma maior agilidade, visibilidade e fiabilidade (Critical Manufacturing 2019f). O compromisso da empresa foca-se no desenvolvimento de “soluções de vanguarda, indo de encontro aos desafios mais importantes da indústria e disponibilizar à lista crescente de clientes satisfeitos, soluções de elevado valor acrescentado, no prazo e orçamento requerido”¹ (Critical Manufacturing 2019f).

A estratégia da empresa está sintetizada na sua missão, visão e valores. Se a missão descreve a razão da empresa existir, ou seja, o seu propósito, já a visão retrata o que se aspira alcançar (Cady et al. 2011, pp. 65-66). Isto posto, a missão e visão são divulgados a seguir (Critical Manufacturing 2019h):

- *Missão* – “Trazer valor através da convergência de inteligência, operações e tecnologias de automação para a Indústria 4.0.”².
- *Visão* – “Tornar a Indústria 4.0 uma realidade para todos fabricantes.”³.

Relativamente aos valores, são estes que suportam a visão, moldam a cultura empresarial e são a essência da sua identidade. Como tal, de seguida apresentam-se os valores da Critical Manufacturing (Critical Manufacturing 2019h):

- *Inovação* – “Exceder as expectativas dos clientes através das soluções mais eficientes e de mais alto valor para indústria.”⁴.

¹Tradução livre do autor. No original “[...] solutions that address the most urgent industry challenges and provide our growing list of satisfied customers with the highest value solution, on-time and on-budget.”.

²Tradução livre do autor. No original “We drive business value through the convergence of intelligence, operations, and automation technologies for Industry 4.0.”.

³Tradução livre do autor. No original “We will make Industry 4.0 a reality for all manufacturers.”.

⁴Tradução livre do autor. No original “We constantly exceed our customers' expectations through the most efficient and high value-added manufacturing solutions.”.

- *Agilidade* – “Adaptar as pessoas, processos e soluções de forma a responder à evolução do mundo da manufatura de alta tecnologia.”⁵.
- *Compromisso* – “Defender o sucesso contínuo dos clientes e da empresa.”⁶.

Resumidamente, a estratégia da empresa foca a melhoria contínua dos processos de fabrico dos clientes, o que possibilita melhor qualidade para o produto final, procurando estar na vanguarda da tecnologia e inovação.

2.2 O Produto

Nos últimos anos, o mercado dos sistemas de informação empresariais tem vindo a crescer, sobretudo pela necessidade das empresas aumentarem a sua produtividade e conseqüentemente, melhorarem a sua competitividade (Ugarte, Artiba e Pellerin 2009). Ugarte, Artiba e Pellerin (2009) afirma que, embora sistemas *Enterprise Resource Planning* (ERP) sejam cada vez mais usuais nas empresas, no sentido de gerir as suas operações, estes falham quando aplicados num contexto fabril, ou seja, no “chão de fábrica”. Os departamentos produtivos beneficiam de *software* personalizado, que responda às necessidades específicas do foro produtivo/industrial (Ugarte, Artiba e Pellerin 2009).

Nestas circunstâncias surge o conceito de MES, fruto da necessidade das empresas de manufatura progredirem no mercado, num ponto de vista de melhoria da reatividade, da qualidade, dos custos de produção e dos prazos de entrega (MESA 1997). Desse modo, as funções de um MES estão sobretudo ligadas a atividades de manufatura, que representa uma parte substancial do valor acrescentado em empresas deste setor (Ugarte, Artiba e Pellerin 2009).

Com o objetivo de apresentar o produto, nesta secção faz-se um enquadramento genérico do conceito MES e posteriormente, foca-se o caso específico do Critical Manufacturing MES.

2.2.1 Manufacturing Execution Systems

A organização MESA⁷, uma comunidade mundial, sem fins lucrativos, que junta empresas de manufatura, de prestação de serviços, analistas, académicos e estudantes, com o propósito de melhorar os resultados do negócio e as operações de produção, através da implementação e implantação de tecnologias de informação e das melhores práticas de gestão, deu o primeiro passo na definição formal de MES (MESA 1997):

“Os Manufacturing Execution Systems (MES) fornecem informações que possibilitam a otimização de atividades de produção, desde o lançamento do pedido até aos produtos acabados. Usando dados atualizados e precisos, o MES orienta, inicia, responde e relata as atividades da fábrica à medida que elas ocorrem. A resposta rápida, resultante das mudanças nas condições, associada ao foco na redução de atividades sem valor acrescentado, impulsiona a eficácia das operações e processos fabris. O MES melhora o retorno dos ativos operacionais, bem como o prazo de entrega, gestão de stock, margem bruta e desempenho do fluxo de caixa. O MES fornece informações críticas acerca das

⁵Tradução livre do autor. No original “We continuously adapt our people, processes and solutions to respond to the evolving world of high-tech manufacturing.”.

⁶Tradução livre do autor. No original “We champion the continued success of our customers and our company.”.

⁷*Manufacturing Enterprise Solutions Association*. Disponível em <http://www.mesa.org>

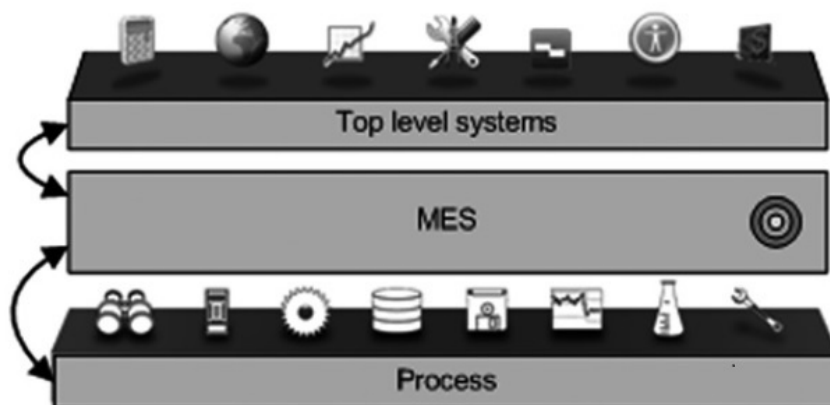


Figura 2.1: Ambiente *Manufacturing Execution System* (MES) e as suas camadas, baseado em Ugarte, Artiba e Pellerin (2009, p. 526)

atividades de produção em toda a empresa e cadeia logística através de comunicações bidirecionais."⁸.

Portanto, o MES age como um intermediário entre os diversos processos existentes no “chão de fábrica” e os sistemas de “alto nível”, existindo comunicação bidirecional entre as camadas, como se demonstra na Figura 2.1. O MES tanto pode fornecer informação acerca dos custos de produção, de indicadores de *performance*, do estado das ordens de fabrico ou rendimento produtivo, como pode também obter dados sobre o planeamento das atividades fabris, parâmetros operacionais, receitas ou instruções de fabrico, por forma a inferir de forma inteligente sobre a fábrica e os seus processos (MESA 1997). Esta bidirecionalidade na comunicação e abrangência no processo produtivo faz com que o MES tenha um papel crucial na Indústria 4.0, já que pode acomodar a integração, descentralização e novas tecnologias, ainda que nem todos os sistemas deste tipo tenham sido desenhados dessa forma (Critical Manufacturing 2019i).

Com o intuito de dar resposta às necessidades de diversos ambientes produtivos, as funções apresentadas na Figura 2.2 e discriminadas a seguir são essenciais para um MES, nomeadamente no suporte, no controlo e na rastreabilidade de cada atividade produtiva (McClellan 2001; MESA 1997; Ugarte, Artiba e Pellerin 2009):

1. *Operações/Agendamento de detalhes* – sequenciamento e distribuição temporal das atividades fabris, por forma a otimizar a *performance*, com base nos recursos disponíveis;
2. *Gestão do processo* – controlo do fluxo de trabalho, baseado nas atividades produtivas reais e planeadas;
3. *Controlo documental* – gestão e distribuição de informação relativa a produtos, processos, ordens de fabrico, assim como recolher os certificados e condições de trabalho;
4. *Aquisição de dados* – monitorização, recolha e tratamento de dados sobre os processos, os materiais e operações, por pessoas, máquinas ou controlos;
5. *Gestão laboral* – supervisão no uso de pessoal de operações num determinado turno, com base nas qualificações, padrões de trabalho e nas necessidades de negócio;

⁸Tradução livre do autor. No original “Manufacturing Execution Systems (MES) deliver information that enables the optimization of production activities from order launch to finished goods. Using current and accurate data, MES guides, initiates, responds to, and reports on plant activities as they occur. The resulting rapid response to changing conditions, coupled with a focus on reducing non value-added activities, drives effective plant operations and processes. MES improves the return on operational assets as well as on-time delivery, inventory turns, gross margin, and cash flow performance. MES provides mission-critical information about production activities across the enterprise and supply chain via bi-directional communications.”.

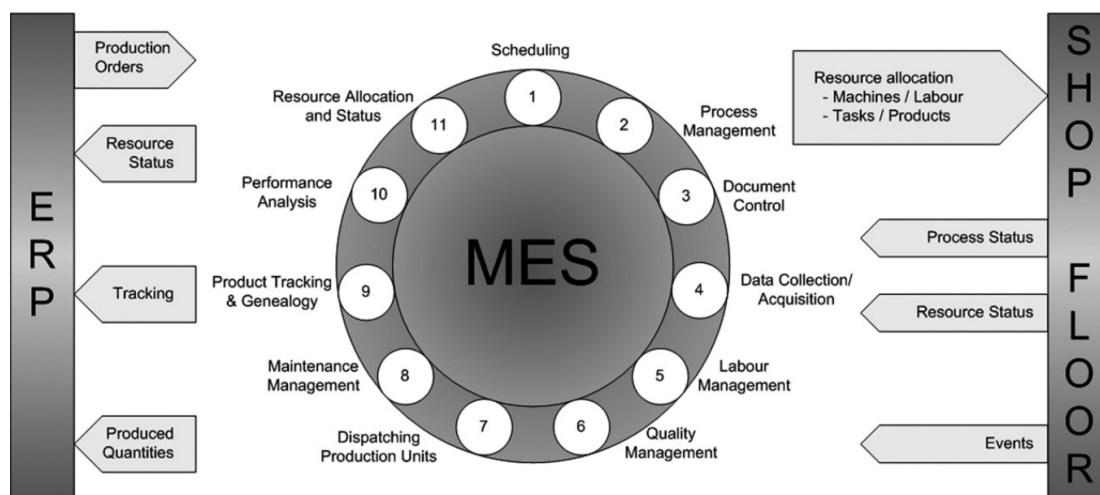


Figura 2.2: Funções do *Manufacturing Execution System* (MES) e o seu enquadramento, extraído de Ugarte, Artiba e Pellerin (2009)

6. *Gestão da qualidade* – registo e análise das características do produto e do processo face aos requisitos ideais;
7. *Expedição de unidades de produção* – dar a ordem para envio de materiais ou ordens para certos setores da fábrica, com o intuito de iniciar um processo ou sub-processo;
8. *Gestão de manutenção* – planeamento e execução de tarefas que visam manter o equipamento e outros ativos capazes de executar a sua tarefa, de forma eficaz;
9. *Genealogia e rastreabilidade do produto* – monitorização do progresso das unidades, amostras ou lotes de saída, para a criação de histórico completo do produto;
10. *Análise de desempenho* – comparação dos resultados medidos com os objetivos e métricas definidas pela corporação, pelos clientes ou órgãos reguladores;
11. *Estado e alocação de recursos* – orientação sobre o que as pessoas, máquinas ou ferramentas devem fazer, acompanhando que já fizeram e o que estão a fazer no momento.

As funções do MES enunciadas servem como base para praticamente qualquer fábrica, fornecendo ferramentas a gestores de fábrica, departamentos de qualidade e manutenção, estando interrelacionadas. Por isso, tratam-se de funções críticas para a maioria dos fabricantes, na medida em que a exigência de processos novos e mais rigorosos no negócio é viável, possibilitando o sucesso no mercado (MESA 1997). Logo, torna-se evidente que o MES traz benefícios para as corporações, alguns alcançáveis num período curto de tempo – aumento de eficiência e redução de custos; redução no tempo de execução de ordens de fabrico; redução dos custos associados ao trabalho; diminuição ou eliminação de papelada; redução da quantidade de material em processamento; utilização de máquinas mais eficaz –, enquanto que outros, possíveis a longo prazo – melhoria geral dos processos; maior satisfação do cliente; melhoria na conformidade regulamentar; maior agilidade; melhoria nos prazos de entrega; maior visibilidade da cadeia logística (Critical Manufacturing 2019i).

2.2.2 Critical Manufacturing MES

O Critical Manufacturing MES afirma-se como o futuro do MES. Trata-se de uma plataforma de *software* com um vasto conjunto modular de aplicações e ferramentas, que dotam os utilizadores de indústrias complexas de agilidade, visibilidade e fiabilidade. O produto adapta-se a diversos processos fabris e às suas operações, sendo fácil a sua implantação, independentemente da infraestrutura

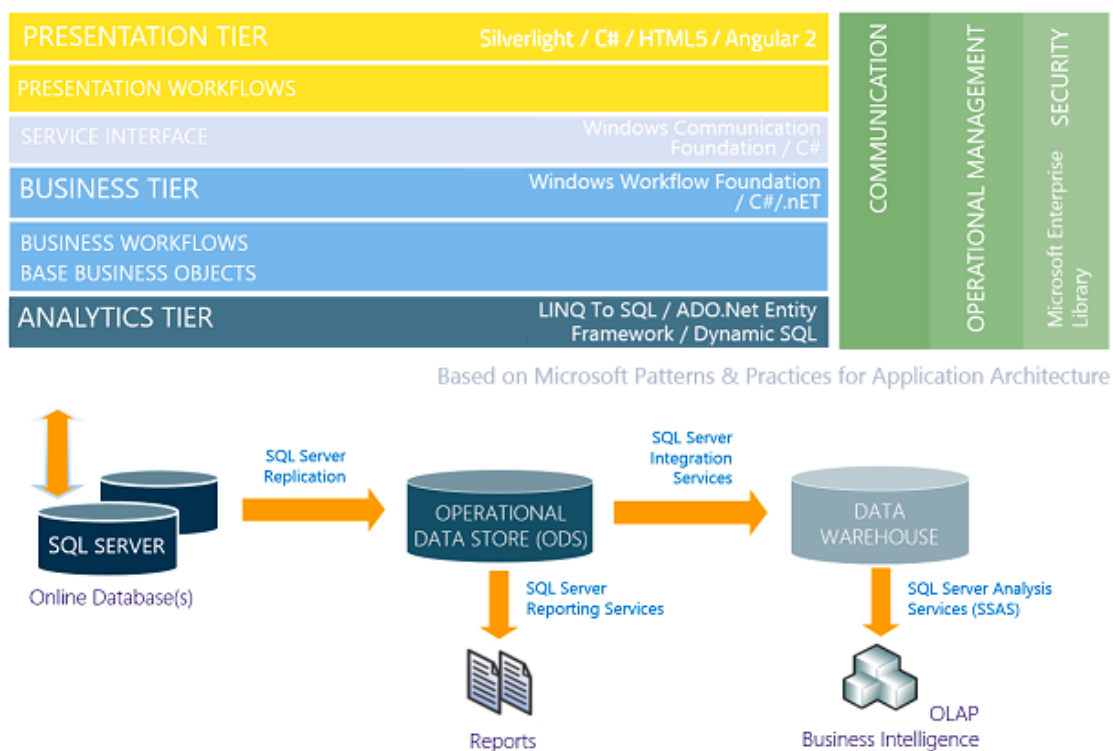


Figura 2.3: Arquitetura do Critical Manufacturing MES e tecnologias usadas, retirado de Critical Manufacturing (2019c)

existente, permitindo o controlo de produção e de custos na empresa e cadeia logística, resultando nas seguintes vantagens (Critical Manufacturing 2019d):

1. Apresenta um **baixo custo total de posse**⁹, visto que a empresa reduz as despesas associadas à implantação, operação e manutenção do sistema;
2. Fornece um largo conjunto de capacidades que dão resposta aos mais variados requisitos, demonstrando a sua **cobertura funcional**;
3. **Capacita o utilizador na sua função**, sendo este capaz de desenhar e colocar em produtivo o plano da fábrica, rastreando os materiais e os detalhes do processo;
4. É sistema modular, o que o torna **extensível, flexível e escalável**, dando aos seus utilizadores acesso a inteligência operacional, de forma fácil e rápida;
5. A arquitetura logicamente descentralizada, ligada à conectividade a diferentes protocolos para equipamentos e dispositivos e ao suporte de produtos preparados para IoT e CPS, tornam o produto **preparado para a Indústria 4.0**.

Relativamente à arquitetura do produto (ver Figura 2.3), a Critical Manufacturing baseou-se nas tecnologias mais recentes para dotar a sua plataforma da capacidade de adaptação aos diversos ambientes produtivos. Posto isto, a infraestrutura consiste em três camadas, que além de fornecerem particionamento, modularidade e escalabilidade das aplicações, foram projetadas para funcionar em conjunto. Além disso, esta foi desenhada de forma a ser customizável e extensível, dado que cada cliente pode ter os seus próprios requisitos (Critical Manufacturing 2019c). Quanto às especificidades de cada camada, são descritas de seguida, numa perspetiva de entender a responsabilidade de cada uma delas e o seu contributo para a plataforma (Critical Manufacturing 2019c).

⁹ *Total Cost of Ownership (TCO)*. É uma estimativa financeira usada para avaliar os custos diretos e indiretos associados a uma compra.

- *Camada de Apresentação (Presentation Tier)* – projetada para trazer ao utilizador uma experiência rica e interativa. Dispõe de várias capacidades (e.g., permitir aos utilizadores criar a sua própria interface gráfica ou desenvolver ecrãs para um propósito em particular, numa fábrica ou setor) e é desenvolvida com suporte multi-plataforma, executando em qualquer sistema operativo *desktop* ou móvel;
- *Camada de Negócio (Business Tier)* – implementa e expõe todas as funcionalidades como serviços, estando disponíveis vários protocolos de comunicação. Contém uma sub-camada de orquestração usada para definir os diversos fluxos de negócio, fornecendo a capacidade de coordenação usando os objetos de negócio. Por fim, a sub-camada de objetos de negócio segue um modelo hierárquico, o qual facilita o desenvolvimento de entidades com um comportamento comum;
- *Camada de Dados (Data Tier)* – desenhado para suportar as capacidades de armazenamento de dados, possibilitando a integração com fontes de dados externas, geração e modificação de relatórios e mineração de dados.

O Critical Manufacturing MES é usado em diversas indústrias, particularmente a indústria de semicondutores (Critical Manufacturing 2019g), de equipamentos médicos (Critical Manufacturing 2019e), de montagem eletrónica (Critical Manufacturing 2019b), procurando dar resposta aos desafios inerentes a cada uma delas.

2.3 Relevância do Problema

Quando um problema surge, é preciso perceber a razão que o torna efetivamente um problema. Ou seja, é necessário explicar a importância de encontrar uma resposta. Nesta secção aborda-se esse assunto, na perspetiva da Indústria 4.0 e do produto Critical Manufacturing MES.

2.3.1 Na Indústria 4.0

A Indústria 4.0, ou quarta revolução industrial, introduz uma necessidade de vanguarda da tecnologia, abrindo caminho para a mecanização, automatização e digitalização dos processos fabris (Lasi et al. 2014). Do vários conceitos identificados como pilares da revolução, tais como a descentralização ou a auto-organização, destaca-se a capacidade de adaptação ao ser humano (Lasi et al. 2014; Zhong et al. 2017). Assim, cimenta-se a visão de fábricas inteligentes, em que os operadores (operadores de linha, engenheiros e gestores) interagem com um ambiente complexo, rico em dados, intuitivamente recolhendo a informação, fazendo com que a IHC seja uma necessidade (Gorecky et al. 2014; Kassner et al. 2017).

Conforme descrito em Gorecky et al. (2014), as interfaces gráficas representam o elemento intermediário entre o utilizador e o sistema subjacente, havendo a necessidade de serem móveis, sensíveis ao contexto em que se enquadram, e que permitam a filtragem da informação, providenciando apenas as possibilidades relevantes para o problema. Por isso, a integração de reconhecimento de voz e texto (Inteligência Artificial) e reconhecimento gestual (Realidade Aumentada) nas interfaces industriais constitui uma estratégia para potenciar a simbiose entre humano e sistema. A visão passa por, não só aproveitar as capacidades das máquinas, mas também por habilitar os trabalhadores de novas capacidades e engenhos, que lhes possibilite tomar melhores decisões, de forma mais rápida (Romero et al. 2016). O trabalho Romero et al. (2016) refere ainda a existência de diferentes interações do operador com partes do sistema, que seriam desejáveis, das quais se destacam as que dizem respeito à interação cognitiva:

- *Operador e Assistente Inteligente Pessoal* – um assistente inteligente é um agente de *software*, dotado de IA, que foi desenvolvido com o objetivo de ajudar o operador a comunicar com máquinas, computadores, bases de dados e outros de sistemas de informação, assim como,

assistir nas tarefas de gestão de tempo e na execução de tarefas, através duma interação humanizada. Alguns cenários de uso onde esta interação constitui uma enorme vantagem são na procura e aquisição de informação armazenada, na leitura de instruções de uso de um determinado recurso, na gestão de inventário e no planeamento de atividades;

- *Operador e Análise de Big Data* – a análise de *Big Data* corresponde ao processo de coletar, organizar e examinar grandes conjuntos de dados, com o objetivo de obter informação relevante e prever eventos cruciais. Ajuda os operadores (e.g., engenheiros de produção) a atingir melhores previsões, fornecer visibilidade sobre os indicadores de desempenho do processo, diminuindo o tempo de reação aquando a ocorrência de problemas, levando a decisões mais rápidas e incisivas. Está conectado com outras partes do sistema, como o assistente virtual mencionado no ponto anterior.

Portanto, a junção destes dois mundos é desejável, pelas vantagens que traz para o operador na sua relação com o sistema, que inclui o seu empoderamento, a melhoria da capacidade de decisão e no desempenho do processo, o que potencia o valor do produto. Este trabalho, cujo problema representa uma ínfima parte duma problemática maior, equaciona a coligação entre ser humano e máquina, possibilitando o avanço da tecnologia usada na Indústria 4.0.

2.3.2 No Critical Manufacturing MES

O Critical Manufacturing MES, como o próprio nome indica, é um MES, que tal como foi abordado anteriormente (ver Secção 2.2), tem um papel crucial na Indústria 4.0. Por essa razão, é esperado que também herde as problemáticas levantadas no contexto desta revolução industrial. De acordo com Almada-Lobo (2016), no futuro, os MES devem estar preparados para lidar com aspetos fundamentais da Indústria 4.0, como a descentralização, a integração vertical de sistemas, a conectividade, mobilidade, usabilidade, capacidade analítica e integração na *cloud*.

Este produto é usado em ambiente produtivo, por operadores de linha e engenheiros/gestores de produção, e fornece uma vista holística sobre a fábrica, permitindo desempenhar diversas funções, tais como a gestão de tarefas/instruções de fabrico, gestão do estado das ordens de fabrico, visualização e geração de relatórios dos custos de produção e *performance* do processo. Para esta última, o Critical Manufacturing MES oferece acesso à informação através da sua interface gráfica (ver Figura 2.4), possibilitando a exportação dos dados para ferramentas como o *Excel* ou *Power BI* para uma análise mais detalhada (Critical Manufacturing 2019a). Contudo, este tipo de abordagem é inadequada para alguns casos, dado a dinâmica do processo de fabrico, a qual obriga a pesquisa de informação rápida, para a toma de decisões expedita. Por outro lado, a solução atual exige o conhecimento da forma como dados se encontram armazenados, ainda que não detalhadamente. Por exemplo, perguntas como “Qual o índice de desempenho, por máquina, do setor *S*?” ou “Quantas peças *P* foram produzidas, durante o mês de Janeiro, na máquina *M1*?” podem constituir um problema de aquisição de informação. Por um lado, o responsável pela aquisição desse conteúdo vai necessitar de conhecer minimamente o modelo de dados do sistema para que consiga extrair esse conhecimento. Por outro, a dificuldade de adquirir essa informação rapidamente, que além de abrandar o processo de apoio à decisão, não promove a interação natural entre homem e produto.

A introdução de IA sobre a forma de linguagem natural visa melhorar a usabilidade, aproximando o operador do sistema subjacente, facilitando o acesso à informação relevante e melhorando o processo de apoio à decisão. Para além disso, a inclusão deste tipo de tecnologia apresenta uma vantagem competitiva para o produto, que permite fortalecer a sua notoriedade, potenciar o seu volume de vendas e consolidar-se como parte do topo dos MES que rumam à quarta revolução industrial.

NAME	MODIFIEDBY	MODIFIEDON	PARENTMATERIALNAME	PRIMARYQUANTITY	PRIMARYUNITS	PRODUCTNAME	STEPIAME
Cookie02	CMFcmfsu	10/09/2019 06:35 PM		20	Kg	Belgas	Mixing
Cookie02.1	CMFcmfsu	10/07/2019 11:50 PM	Cookie02	5	Kg	Belgas	Mixing
Cookie02.2	CMFcmfsu	10/07/2019 11:50 PM	Cookie02	5	Kg	Belgas	Mixing
Cookie02.3	CMFg-araujo	10/10/2019 06:30 PM	Cookie02	5	Kg	Belgas	Mixing
Milka-01	CMFcmfsu	10/07/2019 11:50 PM		10	Kg	Milka	Chocolate Preparation
Milka-02	CMFcmfsu	10/07/2019 11:50 PM		20	Kg	Milka	Chocolate Preparation
Milka-03	CMFcmfsu	10/07/2019 11:50 PM		30	Kg	Milka	Chocolate Preparation
Milka-04	CMFRINMAGALHAES	10/11/2019 10:06 AM		32	Kg	Milka	Chocolate Mixing
Milka-05	CMFRINMAGALHAES	10/10/2019 02:28 PM		20	Kg	Milka	Chocolate Mixing
Clone 1	CMFcmfsu	10/09/2019 03:14 PM		0	Kg	Belgas	Moulding
Clone 2	CMFcmfsu	10/09/2019 02:20 PM		0	Kg	Belgas	Mixing
Clone 3	CMFcmfsu	10/09/2019 02:16 PM		20	Kg	Belgas	Mixing
Clone 4	CMFcmfsu	10/09/2019 02:16 PM		20	Kg	Belgas	Mixing
Clone 5	CMFcmfsu	10/09/2019 02:16 PM		20	Kg	Belgas	Mixing
Clone 6	CMFcmfsu	10/09/2019 02:16 PM		20	Kg	Belgas	Mixing
Clone 7	CMFcmfsu	10/09/2019 02:16 PM		20	Kg	Belgas	Mixing
Clone 8	CMFcmfsu	10/09/2019 03:44 PM		20	Kg	Belgas	Mixing
Material20191000001	CMFcmfsu	10/09/2019 02:20 PM	Clone 2	10	Kg	Belgas	Mixing
Material20191000002	CMFcmfsu	10/09/2019 02:20 PM	Clone 2	5	Kg	Belgas	Mixing
Material20191000003	CMFcmfsu	10/09/2019 02:20 PM	Clone 2	5	Kg	Belgas	Mixing
Material20191000004	CMFcmfsu	10/09/2019 03:01 PM	Clone 1	10	Kg	Belgas	Moulding

Figura 2.4: Interface gráfica do Critical Manufacturing MES para acesso a informação do processo de fabrico

2.4 Síntese

Neste capítulo tentou-se enquadrar o problema na realidade em que se insere. Para tal, apresentou-se a Critical Manufacturing, a sua visão, missão e valores, os quais estão intrinsecamente ligados ao conceito da Indústria 4.0.

Seguiu-se a introdução ao produto da Critical Manufacturing, descrevendo o conceito de MES, um sistema intermediário entre os processos existentes em contexto fabril, as funções que este alberga e quais os seus benefícios. Depois, focou-se no caso específico do produto, procurando expor as suas características e arquitetura.

Finalmente, fundamentou-se a importância do problema, quer para a Indústria 4.0, quer para o Critical Manufacturing MES, numa perspetiva de mostrar as razões que levam à necessidade de uma solução, que incluem a melhoria de usabilidade e do processo de apoio à decisão.

Capítulo 3

Estado da Arte

O presente capítulo tem como objetivo a exploração da área de estudo associada ao problema. Começa-se por introduzir o conceito de PLN, dando um contexto histórico e explorando os fundamentos desta temática. Posteriormente, introduz-se a temática mais específica do problema, descrevendo as suas características, nomeadamente os tipos de abordagem adotados e arquiteturas de sistema típicos. Segue-se a apresentação dos trabalhos relevantes para o campo de estudo, quer na realidade do problema, quer noutros cenários. Por último, exploram-se as ferramentas tipicamente usadas na resolução de problemas associados ao PLN e que são passíveis de ser usadas no desenvolvimento de ILN.

3.1 Processamento de Linguagem Natural

O PLN é um campo da Ciência da Computação, Inteligência Artificial (IA) e Linguística, que explora a forma como os computadores podem ser usados na compreensão, manipulação e geração automática da linguagem natural, em forma de texto ou voz (Beysolow 2018; Chowdhury 2003; Pinto 2015). Nos últimos anos, a área tem-se tornado bastante popular com o acesso fácil a informação através da Internet, estando presente em implementações de *chatbots*, verificadores ortográficos em telemóveis e assistentes de IA nos *smartphones*, tais como a Cortana¹ ou Siri² (Beysolow 2018; Pinto 2015).

3.1.1 História

O início do PLN remonta aos anos 40, com o desenvolvimento da Ciência da Computação, aliada aos avanços na Linguística, que levou ao aparecimento da teoria da linguagens formais. Muito sucintamente, esta teoria consiste na modelação de estruturas complexas e respetivas regras, ou seja, permitem especificar e reconhecer linguagens a partir de modelos matemáticos (e.g., um alfabeto é uma estrutura simples, a qual é constituída por letras que podem formar palavras, em diferentes idiomas). Por outro lado, os avanços na IA, particularmente com o modelo *Single Layer Perceptron* (SLP), apresentado na Figura 3.1, também contribuíram para este campo (Beysolow 2018). O SLP é a base dos modelos neuronais usados nos dias de hoje. Warren McCulloch e Walter Pitts propuseram este modelo, baseado na analogia entre células nervosas (neurónios) e os processos computacionais, que permite computar a soma dos pesos (w_{ni}) associados a cada entrada (x_n), dando uma resposta binária consoante o valor da soma (\sum) varia de acordo com um determinado valor limite, decidindo se uma determinada ação será executada (Hertz 2018).

Ao longo dos anos, várias técnicas foram surgindo na tentativa de resolver problemas associados à compreensão da linguagem natural. Mas, nos últimos 20 anos, notou-se o aumento de interesse no PLN, juntamente com *Machine Learning* (ML), sobretudo devido ao aumento do poder computacional e à facilidade de acesso a dados etiquetados através da Internet (Beysolow 2018).

¹Disponível em <https://www.microsoft.com/en-us/cortana>.

²Disponível em <https://www.apple.com/siri/>.

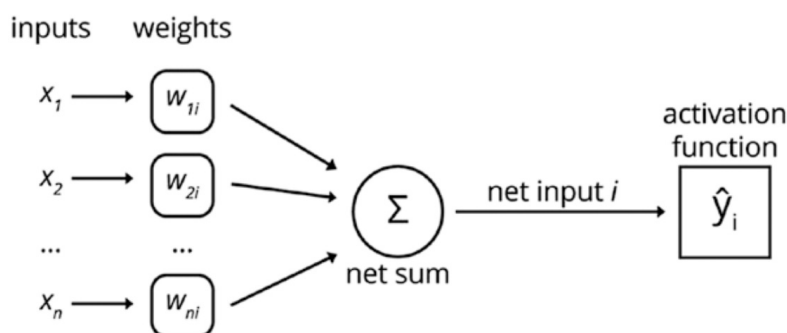


Figura 3.1: *Single Layer Perceptron (SLP)*, extraído de Beysolow (2018)

3.1.2 Fundamentos

O cerne de qualquer tarefa de PLN está relacionado com a compreensão da própria linguagem (Chowdhury 2003). O desenvolvimento deste tipo de aplicações incorre em alguns problemas tais como a processo de pensamento, a representação e significado linguístico e/ou conhecimento do domínio, que estão associados à ambiguidade da linguagem natural (Chowdhury 2003; Pinto 2015). Por outras palavras, a ambiguidade surge quando não é possível atribuir um significado único a uma dada expressão. Se uma pessoa é capaz de o fazer, baseada na sua experiência, capacidade de interpretação de contexto ou na sua cultura, já um computador não tem essa mesma capacidade (Pinto 2015). Por isso, Chowdhury (2003) enuncia que, para ser capaz de compreender a linguagem natural, é importante considerar os vários níveis de conhecimento interdependentes que o ser humano usa na extração de significado:

- *Nível fonético ou fonológico* – encarrega-se a pronúncia;
- *Nível morfológico* – lida com os *tokens*, ou seja, partes nucleares de um frase (e.g., palavras, sufixos, prefixos, sinais de pontuação, dígitos, entre outros);
- *Nível léxico* – trata do significado léxico dos símbolos e análise de partes do discurso;
- *Nível sintático* – lida com a gramática e a estrutura frásica;
- *Nível semântico* – encarrega-se de clarificar o significado da frase ou das palavras;
- *Nível pragmático* – ocupa-se da relação entre a linguagem e o contexto, ou seja, contempla a relação com o mundo exterior;
- *Nível de discurso* – suporta o agrupamento de diferentes frases, identificando a relação entre elas, de forma a compreender o contexto.

Um sistema PLN pode envolver todos ou alguns destes níveis, cujas atividades permitem solucionar pequenas partes de um problema mais complexo. Algumas destas tarefas ou ferramentas incluem técnicas de segmentação de palavras e construção frásica, *parsing* sintático e estatístico, métodos de desenho de modelos de conhecimento estruturados, redes neuronais e modelos de linguagem neuronal (Chowdhury 2003; Jurafsky e Martin 2014).

3.2 Interfaces de Linguagem Natural

Uma ILN é um componente que aceita expressões de consulta ou comandos em linguagem natural e providencia as respostas apropriadas, ou seja, esta deve ser capaz de traduzir as frases nas respetivas ações para o sistema (Chowdhury 2003). Em geral, estas são usadas para consulta de informação, tipicamente mantida em bases de dados (relacionais), pelo que são frequentemente designadas na

literatura por Interfaces de Linguagem Natural para Bases de Dados (ILNBD) (A. R. Akula 2015; Mony, Rao e Potey 2014).

As ILNBD apresentam um problema clássico na área de PLN e constitui um campo de estudo em desenvolvimento. Genericamente, a solução inerente a este problema pode ser dividida em duas fases: processamento linguístico, em que a frase de pesquisa é mapeada e traduzida para a *query* de *Structured Query Language* (SQL) correspondente, usando funções de mapeamento adequadas; processamento na base de dados, na qual é executado a gestão de acesso ao sistema e execução da respetiva consulta (Mony, Rao e Potey 2014). Este tipo de sistemas é capaz de responder a uma grande variedade de *queries* de linguagem natural mas são pouco usados comercialmente, principalmente pela pouca robustez nas capacidades de processamento de contexto (A. Akula, Sangal e Mamidi 2013), pela falta de cobertura linguística ou pelo facto de utilizador poder assumir inteligência por parte do sistema (Mony, Rao e Potey 2014; Patel e Dave 2015). Ainda assim, existem várias vantagens que contribuem para o desenvolvimento deste tipo de aplicações, nomeadamente a facilidade e simplicidade de utilização, o facto de ser mais adequado para questões que envolvem negação ou quantificação ou a sua tolerância a erros gramaticais (Mony, Rao e Potey 2014; Nihalani, Silakari e Motwani 2011; Patel e Dave 2015).

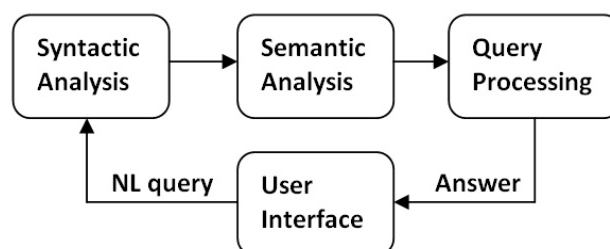


Figura 3.2: Sistema não-contextual, extraído de A. Akula, Sangal e Mamidi (2013)

Num ponto de vista processual, A. Akula, Sangal e Mamidi (2013) menciona que os sistemas de ILNBD podem ser divididos em dois tipos: não-contextuais e contextuais. Num sistema não-contextual (ver Figura 3.2) existe a necessidade de modelo semântico de base, descrevendo regras de domínio. Na fase de análise sintática (*Syntactic Analysis*) é extraída a informação linguística da *query* de linguagem natural. Por sua vez, na análise semântica (*Semantic Analysis*) identificam-se as entidades, atributos a partir da resposta da fase anterior e dos modelos semânticos. Finalmente, na fase de processamento de *queries* (*Query Processing*), as entidades identificadas são mapeadas num grafo, computando-se o caminho mais curto. Dessa forma, é gerada a *query* SQL e executada para obter os resultados (A. Akula, Sangal e Mamidi 2013). Já uma ILNBD contextual (ver Figura 3.3) recolhe informação acerca do contexto numa “conversa” com o utilizador. Neste caso, as capacidades de processamento devem ser contidas na arquitetura através da inserção de uma nova etapa (*Context Processing*), mantendo intactas as responsabilidades de cada fase (A. Akula, Sangal e Mamidi 2013). Relativamente às abordagens ou estratégias de desenvolvimento deste tipo de *software*, são várias e cada uma delas apresentam particularidades que podem influenciar a forma como o sistema é desenhado (Nihalani, Silakari e Motwani 2011; Patel e Dave 2015):

- *Abordagem simbólica (baseada em regras)* – a linguagem é analisada e é aplicada lógica baseada em regras, de forma a capturar o significado da linguagem. O conhecimento encontra-se mapeado em regras ou noutras formas de representação;
- *Abordagem empírica (baseada em experiências)* – aplica análise estatística ou outro tipo de análises orientadas aos dados. Maior parte dos métodos de PLN aplicam técnicas estatísticas com modelos *n-gram*, *Hidden Markov* ou gramáticas de contexto livre;
- *Abordagem de conexão (baseado em redes neuronais)* – baseada em representações distribuídas que correspondem a regularidades estatísticas na linguagem. Uma vez que as capacidades

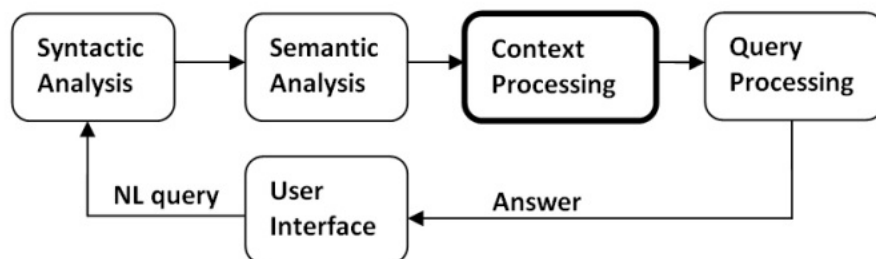


Figura 3.3: Sistema contextual, extraído de A. Akula, Sangal e Mamidi (2013)

da linguagem humana assentam na rede neuronal no cérebro, as redes neuronais artificiais apresentam um ponto fulcral na modelação do processamento de linguagem.

Quanto à arquitetura das ILNBD, além de serem variadas, possuem também diferentes interpretações ao problema. Cada uma apresenta vantagens e desvantagens, pelo que é necessário explorar, de forma resumida, as características de cada.

3.2.1 Sistemas Pattern Matching

Os primeiros esforços no desenvolvimento de sistemas deste género começaram em meados do século XX. O conceito de *Pattern Matching* permite mapear diretamente o *input* do utilizador para a obter o resultado desejado. A implementação destes sistemas implica que os detalhes da base de dados estejam presentes no código, ou seja, torna a solução limitada a um contexto específico e ao número e complexidade de padrões existentes (Nihalani, Silakari e Motwani 2011). A principal vantagem desta abordagem prende-se à simplicidade de implementação, pelo que não há necessidade em conceber módulos de interpretação ou *parsing* da linguagem (Nihalani, Silakari e Motwani 2011; Patel e Dave 2015).

3.2.2 Sistemas Baseados em Sintaxe

Os sistemas baseados em sintaxe possibilitam que a questão do utilizador seja analisada sintaticamente, dando origem a uma árvore que é diretamente mapeada para uma expressão SQL. Para isso, estes sistemas usam uma gramática que descreve as estruturas sintáticas das perguntas dos utilizadores (Nihalani, Silakari e Motwani 2011). Geralmente, é difícil mapear todas as regras que constituem a gramática e o processo de escolha de quais as regras devem ser representadas é complexo. Outro problema é o facto de uma frase poder ter múltiplas corretas árvores de análise sintática, que quando traduzidas, podem levar a diferentes resultados. Também a dificuldade de transformar a árvore de análise sintática diretamente numa linguagem genérica de base de dados é um problema complexo de resolver (Patel e Dave 2015). A principal vantagem desta abordagem é o facto de fornecer informação acerca da estrutura frásica, possibilitando o mapeamento da semântica em regras produtivas (nós da árvore de análise sintática) (Nihalani, Silakari e Motwani 2011).

3.2.3 Sistemas de Gramática Semântica

Apesar da sua semelhança com os sistemas baseados em sintaxe, a ideia inerente a um sistema deste tipo é a simplificar a árvore de análise sintática, através da combinação de alguns nós ou remoção dos mesmos. Posto isto, um sistema de gramática semântica é capaz de refletir melhor a representação semântica da frase, sem as estruturas complexas na árvore, com a possibilidade de designar nomes para os nós, reduzindo a ambiguidade. As principais desvantagens desta abordagem prendem-se com a necessidade de conhecimento prévio do domínio, tornando-se difícil a transposição

para um outro e a estrutura específica das árvores de análise sintática não poderia ser usado noutra aplicação (Nihalani, Silakari e Motwani 2011; Patel e Dave 2015).

3.2.4 Sistemas de Representação Intermediária de Linguagem

Atualmente, as ILN transformam a linguagem natural numa representação intermediária, definida internamente. Assim, a *query* lógica representada na linguagem intermédia expressa o significado da questão colocada pelo utilizador em termos dos conceitos do domínio, os quais são independentes da estrutura da base de dados. Posteriormente, a *query* lógica é traduzida na linguagem genérica de base de dados e avaliada. Esta arquitetura surgiu da dificuldade de traduzir diretamente a linguagem natural para a SQL, ou outra semelhante. O processo de transformação da *query* lógica para a linguagem de base de dados pode conter várias fases, dependendo da necessidade do sistema (Nihalani, Silakari e Motwani 2011).

3.3 Trabalhos de Referência

A investigação e revisão dos trabalhos considerados mais representativos e relacionados com esta tese é fulcral, na perspetiva de perceber as inovações que cada um trouxe para o campo de estudo das ILN e em que medida se enquadram com o problema em resolução.

3.3.1 Aplicados no Panorama Fabril

A pesquisa na literatura existente revela a existência de poucos casos de especificação e/ou implementação de ILN no contexto industrial, e nenhum dos casos se refere à aplicabilidade nos MES. Como referido inicialmente (ver Secção 3.2), esta matéria é relativamente recente, cujo estudo está em desenvolvimento. Ainda assim, são expostos os trabalhos considerados de relevo neste âmbito.

GSPAS

Denominado de *Global Study Process Allocations System*, abreviado em GSPAS, é um sistema de IA desenvolvido pela Ford e aplicado no planeamento de processo, análise dos processos de montagem e tradução de instruções para construção dos veículos (N. Rychtycky 2007). A ideia inerente ao GSPAS é o uso de uma linguagem interna, denominada de *Standard Language*, baseada em linguagem natural, mas obedecendo a um conjunto de regras padronizadas internamente, e que permite a execução de ações no processo. As especificações para a montagem de um determinado veículo são escritas em “folhas de processo”³, pelos engenheiros de produção, e são posteriormente entregues ao sistema, que lê as instruções usando técnicas de PLN, nomeadamente recorrendo a *Augmented Transition Network* (ATN) para o *parsing* sintático, converte-as num conjunto de operações a realizar e estima o tempo que cada operador necessita para cumprir as tarefas de montagem (Plesco e N. Rychtycky 2012; N. Rychtycky 2007).

Mediante o referido em N. Rychtycky (2007), a base de conhecimento do sistema é totalmente configurável, devido à natureza dinâmica do processo de fabrico automóvel, havendo a necessidade de atualização frequente do domínio. Também é mencionado que, só nos primeiros seis meses de funcionamento, este sistema foi capaz de identificar com sucesso 1100 “folhas de processo” com problemas associados ao fator humano, prevenindo que fossem colocadas erradamente em produção, evitando mais de 17 milhões de dólares de prejuízo. O plano para futuro é integrar tecnologias *Resource Description Framework* (RDF) e *Web Ontology Language* (OWL), *standards* da Ontologia

³*Manufacturing Process Sheets*. Documento que lista as operações a executar para obter o produto final.

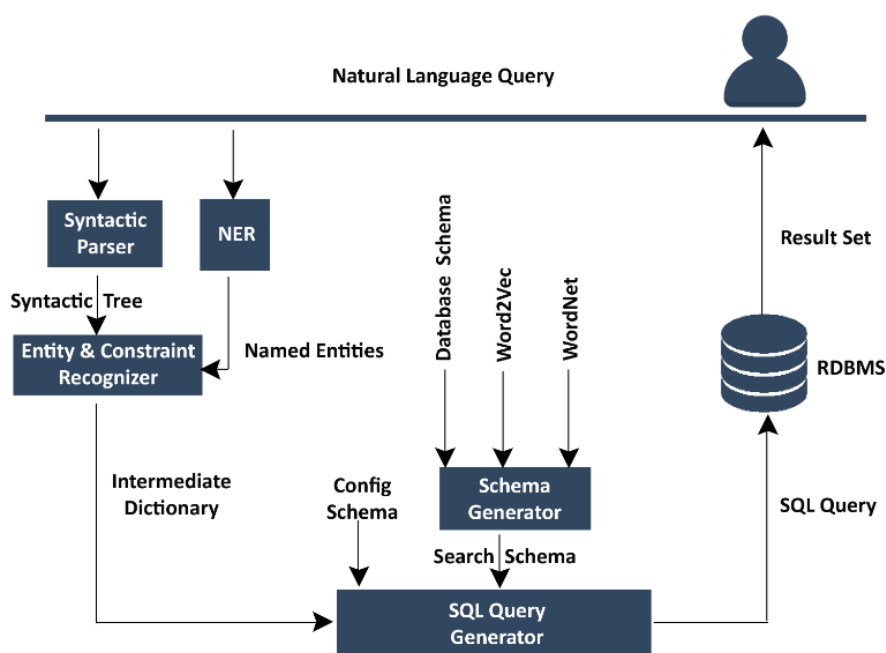


Figura 3.4: Arquitetura do sistema Surukam-NLI, extraído de Ramesh, Jain, Sundaresan et al. (2016)

e Web Semântica, com perspectiva de permitir aos engenheiros de produção consultar, em tempo apropriado, informação acerca do processo fabril (N. Rychtyckyj 2007).

Surukam-NLI

Este sistema consiste numa ILNBD que aplica técnicas de *parsing* sintático e ML para mapeamento semântico da linguagem natural, procurando extrair entidades e regras, que constituem a linguagem intermédia, e seguidamente usar esses elementos na conversão em SQL (Ramesh, Jain, Sundaresan et al. 2016). Ele foi testado com base de dados *AdventureWorks*⁴ da Microsoft, a qual contém cenários de cariz empresarial, designadamente contendo informação de recursos humanos (e.g., salário de operadores, data de entrada), vendas e *marketing* (e.g., dados de clientes e vendas) e informação de produto (e.g., tipos de produção, custo de produção) (Ramesh, Jain, Sundaresan et al. 2016).

Na Figura 3.4 consta a arquitetura do Surukam-NLI, que é aqui explicada, a alto nível: dada uma *query* de linguagem natural, esta é transformada numa representação intermédia (*Intermediate Dictionary*) que contém as entidades e regras identificadas pelo *Entity & Constraint Recognizer*. De seguida, o *SQL Query Generator*, baseado na esquema de base de dados previamente configurado, converte o *input* recebido na *query* SQL e executa-a contra a respetiva base de dados (Ramesh, Jain, Sundaresan et al. 2016).

As métricas retiradas evidenciam que este sistema possui uma precisão de cerca de 80%, necessitando de mais pesquisa e estudo para abordar *queries* complexas, que envolvem, por exemplo, operações complementares à conversão para SQL ou ambiguidade na linguagem natural (Ramesh, Jain, Sundaresan et al. 2016).

⁴Disponível em <https://docs.microsoft.com/en-us/sql/samples/>.

3.3.2 Outras Aplicações

Para além dos trabalhos alcançados em contexto semelhante a esta tese, são apresentados outros igualmente relevantes na temática das ILN, embora não aplicados na área industrial.

LUNAR

O LUNAR é um sistema que dá resposta ao domínio de amostras de rochas trazidas da lua e foi a primeira ILNBD (Nihalani, Silakari e Motwani 2011; Patel e Dave 2015). O desenvolvimento deste sistema surgiu da necessidade de possibilitar aos cientistas envolvidos no estudo das rochas lunares poderem obter informação para formular e testar as suas hipóteses, de uma forma simples e intuitiva. O LUNAR permitia ao cientista executar diversas ações como fazer questões, computar médias e taxas, criar listas baseadas em critérios de seleção ou comparar medidas de diferentes investigadores, usando informação de duas bases de dados, uma contendo dados de análises químicas e a outra com dados de referências bibliográficas. Apesar de ter sido desenvolvido como protótipo, este sistema apresentou um desempenho satisfatório, sendo que cerca de 78% dos pedidos foram respondidos com sucesso (A. Woods, Kaplan e Webber 1972).

LADDER

O LADDER é um sistema desenhado para consultar informação sobre navios da Marinha Americana, por forma a auxiliar os gestores da Marinha no processo de tomada de decisão (Hendrix et al. 1978; Nihalani, Silakari e Motwani 2011). O sistema, que usa gramática semântica para tratar *queries* a uma base de dados distribuída, apresenta uma arquitetura de três camadas, cada uma correspondente a um componente do sistema: o INLAND – *Informal Natural Language Access to Navy Data* –, é responsável por aceitar a *query* de linguagem natural, produzir a respetiva *query* de base de dados a partir da decomposição da mesma em fragmentos, sendo posteriormente combinados para unidades sintáticas a alto nível, para que sejam reconhecidas, dando origem a um comando enviado para o próximo componente; o IDA – *Intelligent Data Access* –, compõe uma resposta com base no comando recebido e organiza a sequência correta de *queries* a realizar; o FAM – *File Access Manager* –, o último componente, tem a responsabilidade de gerir o acesso à base de dados distribuída (Hendrix et al. 1978).

Chat-80

Segundo Nihalani, Silakari e Motwani (2011), o Chat-80 é um dos sistemas PLN mais referenciados nos anos 80. O Chat-80 foi desenvolvido pensando na adaptabilidade a diversos domínios, de forma fácil e eficiente. Foi implementado em *Prolog* e incluía uma base de conhecimento com factos geográficos de mais de 150 países (domínio de geografia mundial) e vocabulário inglês suficiente para interação com uma base de dados, que neste caso específico seria implementada totalmente em *Prolog*. Os autores concordaram que a aplicação devia lidar com um conjunto restrito de linguagem natural relevante para o domínio, uma vez que dessa forma se torna uma linguagem de *query* formal mas acessível para o utilizador (Warren e Pereira 1982).

Janus

O Janus é uma aplicação PLN com a capacidade de “comunicar” com múltiplos sistemas, tais como bases de dados, sistemas periciais, dispositivos gráficos, sendo capaz de avaliar a *query* de linguagem natural e inferir acerca de quais os recursos a utilizar, sem que o utilizador se apercebesse da complexidade do sistema (Nihalani, Silakari e Motwani 2011; Resnik 1989). O fluxo do Janus consistia em extrair as expressões da *query* de linguagem natural, usando uma linguagem desenvolvida para o

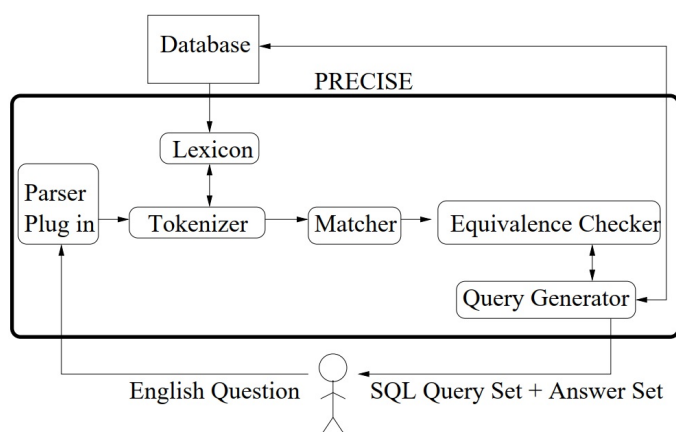


Figura 3.5: Arquitetura do sistema PRECISE, extraído de Popescu, Etzioni e Kautz (2003)

feito, denominada *World Model Language*; traduzir essas expressões para uma representação simplificada e normalizada; aplicar o algoritmo desenvolvido para encontrar a combinação adequada de serviços a disponibilizar, de modo a satisfazer o pedido do utilizador; por fim, a criação e execução de um plano para extração da informação (Resnik 1989).

PRECISE

O PRECISE é um sistema desenvolvido na Universidade de Washington, cuja base de dados alvo é relacional, usando SQL, e que introduz o conceito de frases semanticamente tratáveis, ou seja, *queries* que podem ser traduzidas para uma representação semântica única (Mony, Rao e Potey 2014; Nihalani, Silakari e Motwani 2011). Popescu, Armanasu et al. (2004) menciona que a distinção entre questões semanticamente tratáveis e as complexas resulta num processo de tratamento da linguagem natural mais simples e pode ser usado para compensar erros de *parsing* sintáticos.

A Figura 3.5 apresenta a arquitetura deste sistema, no qual se destaca o *Parser Plugin*, um componente que permite ao Precise adaptar-se aos avanços na tecnologia de *parsing*, sem que haja a necessidade de adaptar todo o sistema (Popescu, Armanasu et al. 2004). Quanto aos restantes componentes: o *Lexicon* extrai os *tokens* de uma dada frase e encontra sinónimos dessas expressões; o *Tokenizer* verifica se, para cada potencial *token*, outras palavras estão também presentes na questão, e associa-lhes um determinado tipo de elemento de base de dados (e.g., valor, atributo, relação); o *Matcher* procede à correspondência entre os *tokens* e os respetivos elementos da base de dados; o *Query Generator*, como o próprio nome indica, é responsável por gerar a *query* de SQL; o *Equivalence Checker* testa se existem soluções distintas e, em caso de as encontrar, o sistema questiona o utilizador acerca da interpretação semântica da questão (Popescu, Etzioni e Kautz 2003).

Este sistema foi avaliado em dois domínios: o primeiro, referente a viagens aéreas e o segundo, associado à geografia dos Estados Unidos da América. De acordo com Nihalani, Silakari e Motwani (2011), no primeiro caso, 95.8% das questões são semanticamente tratáveis, pelo que a precisão do sistema atinge os 94% e, no segundo caso, 77.5% das questões são tratáveis em termos de semântica, obtendo 100% de precisão, destacando-se assim o seu desempenho.

NaLIX

O NaLIX – *Natural Language Interface for an XML Database* – é uma ILNBD desenvolvida na Universidade de Michigan, com o intuito de obter informação genérica a partir de uma base de dados em *Extensible Markup Language* (XML) (Yunyao Li, Yang e Jagadish 2005). De acordo com Yunyao

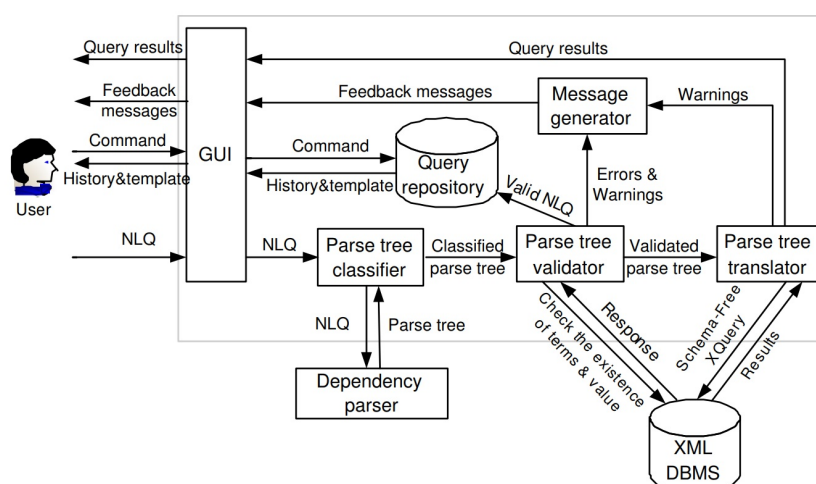


Figura 3.6: Arquitetura do sistema NaLIX, extraído de Yunyao Li, Yang e Jagadish (2005)

Li, Yang e Jagadish (2005), o desafio consiste em traduzir uma *query* de linguagem natural para uma *query* corretamente estruturada para uso numa base de dados, permitindo assim ao utilizador usar operações complexas (e.g., agregação, combinação, junção, entre outras).

Relativamente à arquitetura do NaLIX (ver Figura 3.6), o sistema consiste em duas partes: a primeira é responsável pela tradução da *query* de linguagem natural para XQuery⁵, envolvendo os componentes *Parse Tree Classifier*, *Parse Tree Validator* e *Parse Tree Translator*; a segunda suporta a formulação da *query* de base de dados correspondente, usando os componentes *Query Repository* e *Message Generator* (Yunyao Li, Yang e Jagadish 2005).

De salientar que a linguagem de *query* usada pelo NaLIX (*Schema Free XQuery*) não necessita que seja explicitado qual o *schema* a ser usado, sendo que é capaz de encontrar automaticamente, para uma dada coleção de expressões/palavras-chave, todas as relações existentes entre estes elementos. Assim, é possível abstrair o sistema do domínio existente (Yunyao Li, Yang e Jagadish 2005; Patel e Dave 2015).

GINLIDB

Este sistema – *Generic Interactive Natural Language Interface to Databases* – foi desenvolvido em 2009 com o propósito de ser genérico o suficiente para se adaptar a bases de dados diferentes, dada a base de conhecimento apropriada (El-Mouadib et al. 2009). A arquitetura do GINLIDB, apresentada na Figura 3.7, consiste em dois principais componentes: *Linguistic Handling Component*, o qual gere a exatidão da *query* de linguagem natural, nomeadamente a estrutura gramatical e possibilidade de ser corretamente convertida para SQL; *SQL Construting Component*, responsável por construir a *query* de SQL apropriada e gerir a ligação à base de dados (El-Mouadib et al. 2009).

O GINLIDB destaca-se pelo seu processo de análise sintática, no qual usa ATN, que verifica se a estrutura dos *tokens* é permitida na estrutura gramatical. Este processo é suportado pelo *parser* desenvolvido para o sistema usando uma gramática de contexto livre. Outro destaque deste sistema diz respeito à base de conhecimento usada, que é extensível pelo utilizador, permitindo adicionar novas palavras no dicionário, associar-lhes os respetivos sinónimos e definir o *schema* da base de dados em uso, mapeando assim o domínio. Em termos de resultados, o sistema mostrou-se capaz de responder às questões mais comuns, embora não tenha sido testado em diferentes domínios (El-Mouadib et al. 2009).

⁵Disponível em https://www.w3schools.com/xml/xquery_intro.asp.

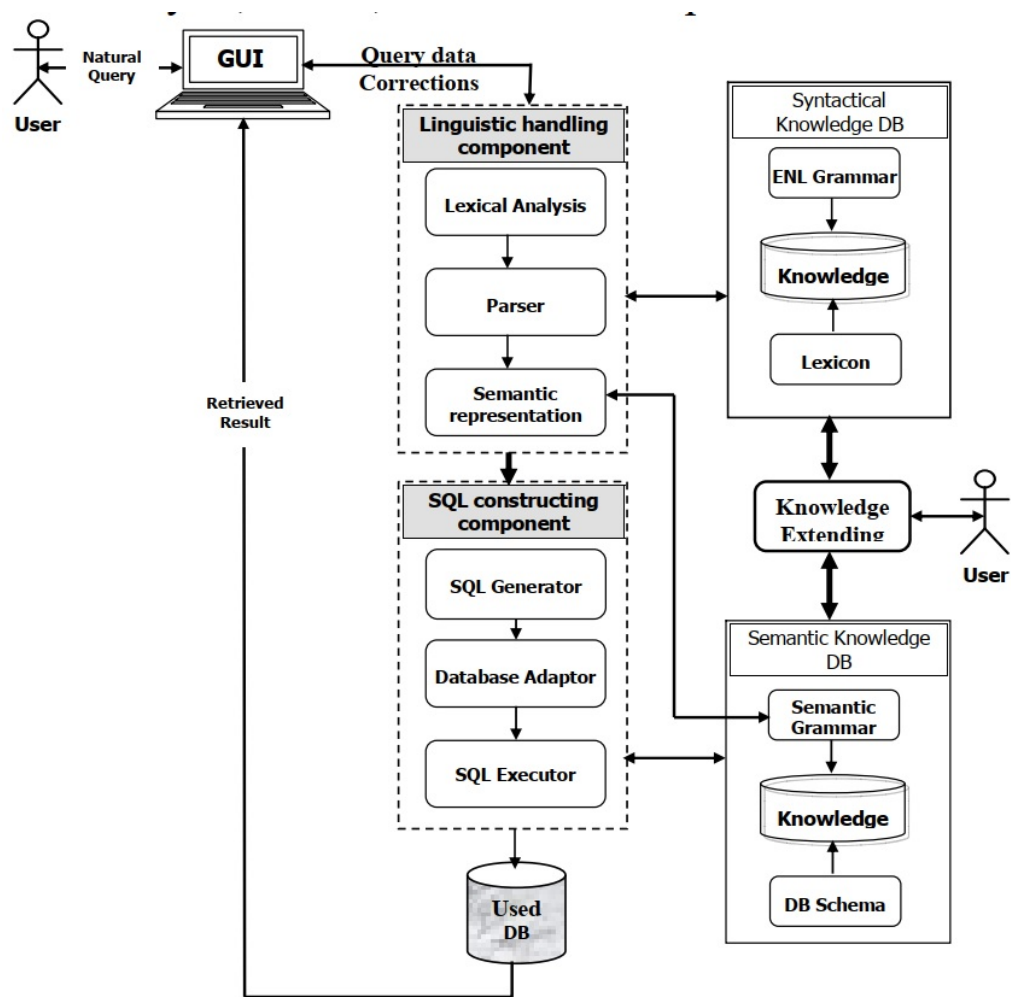


Figura 3.7: Arquitetura do sistema GINLIDB, extraído de El-Mouadib et al. (2009)

Tabela 3.1: Sumário dos trabalhos estudados

Nome	Domínio	Abordagem	Técnica
GSPAS	Montagem de veículos automóveis	Análise léxica e sintática	ATN e Representação Intermediária de Linguagem
Surukam-NLI	Gestão empresarial de contexto produtivo	Análise sintática e abordagem empírica	Representação Intermediária de Linguagem
LUNAR	Amostras de rochas da Lua	Conexão (redes neuronais)	Baseada em sintaxe
LADDER	Navios da Marinha Americana	Abordagem empírica	Gramática semântica
Chat-80	Genérico	Baseada em diálogo	Gramática semântica
Janus	Genérico	Baseada em menus	Representação Intermediária de Linguagem
PRECISE	Viagens aéreas e geografia	Análise léxica e restrições semânticas	<i>Match</i> de palavras-chave e <i>queries</i> tratáveis
NaLIX	Genérico	Busca de palavras-chave	Engenharia reversa de sistema baseado em sintaxe
GINLIDB	Genérico	Análise léxica e sintática	ATN e gramática livre de contexto

3.3.3 Sinopse

Na Tabela 3.1 discriminam-se os trabalhos abordados anteriormente, focando especialmente a abordagem utilizada e a respetiva técnica. Uma análise cuidada sobre os trabalhos demonstra que não existe uma abordagem recorrente. Existe sim, uma combinação adequada à realidade do problema a resolver, que não deve ser generalizada. Também é possível constatar que a representação intermediária de linguagem é a técnica mais frequente, pelo que parece ser adequado a sua aplicação.

Para a solução a conceber, qualquer das combinações pode ser válida, pelo que a integração com técnicas mais recentes de ML ou *Deep Learning* (DL), aliados à representação intermediária de linguagem, podem constituir um novo passo na implementação da ILN. Nesse sentido, o uso de ferramentas PLN já existentes ou o desenvolvimento de raiz de uma para esse efeito são alternativas a serem consideradas.

3.4 Ferramentas

Atualmente, estão disponíveis várias ferramentas para lidar com o PLN, sendo que se torna necessário destacar as mais relevantes para aplicação numa ILN e perceber qual pode ser a mais adequada na resolução do problema, dadas as particularidades de cada uma. Posteriormente, faz-se o comparativo das ferramentas, pressupondo as que parecem ser mais apropriadas para a solução.

3.4.1 NLTK

O NLTK, *Natural Language Toolkit*, foi criado em 2001 na Universidade da Pennsylvania e, desde então, tem-se expandido graças à comunidade *open source* que contribui para o projecto (Beysolow 2018). Esta biblioteca, desenvolvida em *Python* para processamento de linguagem natural e análise textual, caracteriza-se pela sua simplicidade, consistência, extensibilidade e modularidade, apresentando um conjunto de funções otimizadas para suportar estas tarefas listadas (Beysolow 2018; Perkins 2010). De acordo com Lobur, Romanyuk e Romanyshyn (2011), a combinação de *Python* e NLTK dá a capacidade a qualquer programador de resolver facilmente tarefas de PLN, evitando demasiado tempo a estudar os conceitos inerentes. Neste contexto, a biblioteca está integrada com *WordNet*⁶, uma base de dados de relações semânticas entre nomes, verbos, adjetivos e advérbios da Língua Inglesa (Lobur, Romanyuk e Romanyshyn 2011).

⁶Disponível em <https://wordnet.princeton.edu/>

O NLTK providencia poderosas ferramentas estatísticas, está preparado para trabalhar com grandes *datasets*, criar modelos linguísticos robustos, possibilitando a sua extensão para componentes que possam ser usados em sistemas produtivos (Beysolow 2018; Lobur, Romanyuk e Romanyshyn 2011).

3.4.2 Stanford CoreNLP

O Stanford CoreNLP é uma *framework* que providencia um conjunto de ferramentas para analisar discurso, reconhecer entidades, normalizar datas, identificar a estrutura frásica e dependência sintática dos termos, entre outras (Stanford OpenNLP 2019). Ele possui uma *Application Programming Interface* (API) rica, sendo assim acessível em múltiplas linguagens de programação e é a biblioteca mais usada em projetos de pesquisa cuja temática é o PLN (Al Omran e Treude 2017; Stanford OpenNLP 2019).

3.4.3 spaCy

O spaCy é uma biblioteca para métodos avançados de processamento de linguagem natural, desenvolvida com *Python* e *CPython*, e cujo objetivo é suportar a conceção de aplicações de foro comercial (Al Omran e Treude 2017). Esta biblioteca suporta as funcionalidades de PLN a partir de modelos estatísticos pré-treinados especificamente para o spaCy, tornando-o rápido e preciso (spaCy 2019).

3.4.4 TensorFlow

O projeto Google Brain⁷ começou, em 2011, com o objetivo de explorar redes neuronais de larga escala, quer para pesquisa, quer para uso nos produtos da Google (Abadi, Agarwal et al. 2016). O TensorFlow é um sistema de segunda geração, sucessor do DistBelief⁸, usado para a implementação e implantação de modelos de ML de elevada escala (Abadi, Agarwal et al. 2016). Este usa grafos *dataflow*, um modelo de grafo que expressa as possibilidades de execução concorrente de partes de um programa, para representar computacionalmente o estado partilhado e as operações responsáveis pela mutação desse estado, o que inclui operações matemáticas individuais, os respetivos parâmetros, as suas regras de atualização e o pré-processamento dos dados de entrada (Abadi, Barham et al. 2016; Dennis 2011).

Num contexto de PLN, usando TensorFlow, pode aplicar-se redes neuronais convolucionais para tarefas de classificação, como análise de sentimento, deteção de *spam* ou categorização de tópicos (Britz 2019). Embora as redes neuronais convolucionais sejam tipicamente usadas na identificação de imagens, podem também ser usadas em tarefas PLN, usando as palavras como entrada, ao invés dos pixels, sendo a sua computação rápida e eficiente (Britz 2019).

3.4.5 Rasa

O Rasa é uma *framework open source* que permite o desenvolvimento de *chatbots*, possibilitando, entre muitas outras capacidades, a compreensão de intenções e entidades envolvidas na conversação (Rasa 2019). Esta *framework* baseia algumas das suas funcionalidades em ferramentas PLN já existentes, dando-lhe flexibilidade nas tarefas inerentes à compreensão da linguagem natural (Bocklisch et al. 2017). Também importante referir que o Rasa é modular, ou seja, é constituído por dois componentes: o Rasa NLU, responsável pela compreensão da linguagem, classificando intenções e extraindo entidades; o Rasa Core, o motor de construção de diálogos; sendo possível usar cada um

⁷Disponível em <https://ai.google/research/teams/brain/>

⁸Disponível em <https://ai.google/research/pubs/pub40565>

deles em conjunto ou isoladamente, facilitando também a integração com outros sistemas (Bocklisch et al. 2017; Rasa 2019). O domínio consumido pelo Rasa é configurável, através de ficheiros *JavaScript Object Notation* (JSON) ou *Markdown*, e os seus componentes disponibilizam uma API apoiada em *Hypertext Transfer Protocol* (HTTP), o que contribui para a escalabilidade e facilidade de integração do sistema, respetivamente (Bocklisch et al. 2017).

3.4.6 Amazon Lex

O Lex é um produto do *Amazon Web Services* (mais conhecida como AWS), uma plataforma de serviços da Amazon na *cloud*, para construir interfaces de conversação integradas com qualquer aplicação, usando voz ou texto, fornecendo funcionalidades de reconhecimento automático de discurso, respetiva conversão para texto e compreensão de linguagem natural, o que possibilita a identificação das intenções manifestadas pelo utilizador, numa forma simples de usar, implantar e escalar (Amazon 2019). Como demonstrado em AWS Machine Learning Blog (2019), o Amazon Lex garante a sua customização por forma a desenvolver um *chatbot* capaz de interagir com o utilizador, convertendo os seus pedidos de linguagem natural para SQL, obtendo os dados de uma base de dados relacional, e por fim, apresentá-los.

3.4.7 IBM Watson Assistant

O Watson Assistant é um produto do IBM Watson, uma plataforma de serviços na *cloud*, orientados à inteligência artificial e disponibilizado pela IBM (IBM 2019a). Este produto oferece uma interface de conversação que pode ser integrada em qualquer aplicação ou dispositivo, e tal como os seus concorrentes, destaca-se por usar técnicas avançadas de compreensão de linguagem natural (suportando intenções e entidades), por ser simples de implantar e fácil de usar (IBM 2019b). O IBM Watson Assistant também permite a conceção de *chatbots* capazes de interpretar o pedido do utilizador e extrair o conteúdo desejado de uma base de dados (IBM Blog 2019).

3.4.8 Microsoft LUIS

O LUIS, acrónimo para *Language Understanding Intelligent Service*, é um serviço da Microsoft, baseado na *cloud* e parte do produto *Cognitive Services*, e que por sua vez integra na plataforma Azure, que aplica métodos de ML à linguagem natural, permitindo obter informação relevante acerca da mesma (Microsoft 2019c). O intuito do LUIS é tornar possível a criação de soluções que apliquem modelos específicos de ML para a compreensão de linguagem específica de um determinado domínio, sem qualquer tipo de perícia nesta área (Williams et al. 2015). Assim, o LUIS permite a construção do próprio modelo, através da identificação de entidades, de intenções e das frases que lhes são associadas, oferecendo diversas ferramentas ao desenvolvedor para que este processo seja mais simples (Microsoft 2019c). Relativamente à sua aplicabilidade ao problema, a combinação do LUIS com o QnA Maker (outro serviço integrante do *Cognitive Services*) resulta numa solução capaz de identificar as intenções e entidades, entregando respostas pré-fabricadas ou informação de uma determinada fonte (Microsoft 2019b).

3.4.9 Sinopse

As ferramentas apresentadas previamente podem ser divididas por tipo; as cinco primeiras referem-se a bibliotecas ou *frameworks* – NLTK, Stanford CoreNLP, spaCy, TensorFlow e Rasa –, as últimas três dizem respeito a serviços disponibilizados na *cloud* – Amazon Lex, IBM Watson Assistant e Microsoft LUIS. De notar que a natureza de cada uma é diferente, o que influencia o comparativo entre

elas. Portanto, essa comparação não deve ser feita pelas funcionalidades que cada uma oferece, mas sim pelo o valor que poderá trazer à solução.

Uma análise com base na bibliografia estudada parece destacar o spaCy e o Rasa como as ferramentas mais adequadas para uma solução interna, ou seja, sem acesso à Internet, pois parecem estar preparadas para a implementação de soluções comerciais e não se tratam de ferramentas na *cloud*. No caso do NLTK, Stanford CoreNLP ou o TensorFlow, parecem requerer maior estudo e tempo de adaptação.

Por outro lado, as ferramentas *cloud* parecem providenciar todo o tipo de funcionalidades necessárias para a compreensão de linguagem natural e extração de informação. Aliás, todas elas apresentam estratégias para o desenvolvimento de *chatbots* com a capacidade descrita anteriormente. Teoricamente, a abordagem seguida por este tipo de ferramentas parece constituir uma forma robusta de responder ao problema, pelo que deve ser explorada num ponto de vista experimental – usando o Microsoft LUIS, por exemplo.

3.5 Síntese

Este capítulo introduziu a área do PLN, fazendo referência à sua história e aos fundamentos intrínsecos à compreensão da linguagem humana, sobretudo os níveis de conhecimento usados na extração de significado.

De seguida, focaram-se as ILN, sendo o campo de maior relevo para esta tese. Explanaram-se os sistemas contextuais e não-contextuais, as abordagens e arquiteturas aplicadas no desenvolvimento deste tipo aplicacional, explicando-se de uma forma global os conceitos associados.

Mediante a exposição dos trabalhos relevantes, quer no panorama fabril, associado à natureza do problema em mãos, quer noutros, foi dado a conhecer as abordagens seguidas por cada um deles, procurando constatar uma tendência de aplicação de técnica e abordagem, que não foi encontrada, levando assim a concluir que o conjunto de ML ou DL e representação intermediária de linguagem pode contribuir para a solução.

Por fim, fez-se o estudo das ferramentas para PLN, entre elas o NLTK, o TensorFlow e o Amazon Lex, fazendo-se um resumo das vantagens e desvantagens de cada, tendo em conta a bibliografia estudada. Destacam-se o spaCy, o Rasa e as ferramentas *cloud* como bons candidatos para desenvolvimento do protótipo.

Capítulo 4

Conceção

Neste capítulo apresenta-se o processo de idealização do módulo de linguagem natural. Inicialmente, explica-se o processo de experimentação de algumas ferramentas ou abordagens analisadas no capítulo anterior, procurando perceber se a sua aplicação prática é plausível. Depois, mostra-se o modelo concetualizado, concretamente a visão geral, em que se faz uma descrição do modelo na sua globalidade, os casos de uso identificados e por fim, a arquitetura definida.

4.1 Apreciação Prática

Com o propósito de colocar em prática o estudo das ILN, decidiu-se a realização de experiências envolvendo algumas abordagens e ferramentas previamente estudadas (ver Capítulo 3). Posto isto, optou-se por desenvolver pequenas provas de conceito, de forma a fazer uma rápida avaliação das abordagens, técnicas e ferramentas usadas. Para cada uma, é abordado em que consiste, algumas observações pertinentes e referência dos pontos favoráveis e desfavoráveis.

4.1.1 Gramática Baseada em Semântica

Como proposto em Bird, Klein e Loper (2009, p. 361-403), o uso do NLTK permite o desenvolvimento de gramáticas que possibilitam a análise de uma frase pela sua semântica, tal como o exemplo demonstrado a seguir.

```
S[SEM=(?np + WHERE + ?vp)] -> NP[SEM=?np] VP[SEM=?vp]
VP[SEM=(?v + ?pp)] -> IV[SEM=?v] PP[SEM=?pp]
VP[SEM=(?v + ?ap)] -> IV[SEM=?v] AP[SEM=?ap]
NP[SEM=(?det + ?n)] -> Det[SEM=?det] N[SEM=?n]
PP[SEM=(?p + ?np)] -> P[SEM=?p] NP[SEM=?np]
AP[SEM=?pp] -> A[SEM=?a] PP[SEM=?pp]
NP[SEM='Country="greece"'] -> 'Greece'
NP[SEM='Country="china"'] -> 'China'
Det[SEM='SELECT'] -> 'Which' | 'What'
N[SEM='City FROM city_table'] -> 'cities'
IV[SEM=''] -> 'are'
A[SEM=''] -> 'located'
P[SEM=''] -> 'in'
```

Código 4.1: Excerto de uma gramática, extraído de Bird, Klein e Loper (2009)

O código apresentado em 4.1, usando o NLTK, para a pergunta “What cities are located in China?” permite gerar a seguinte *query* de SQL: *SELECT City FROM city_table WHERE Country="china"*. Com esta metodologia é possível mapear diretamente linguagem natural em SQL, sendo que a gramática funciona como a base de conhecimento. Por conseguinte, torna-se viável a codificação de um *parser* que usa as gramáticas definidas, verificando alguma é capaz de dar resposta à *query* de linguagem natural.

Pontos Favoráveis

- Viabilidade de desenvolvimento de uma solução customizada;
- Utilização de ferramenta *open source*;
- Tradução imediata de linguagem natural para SQL.

Pontos Desfavoráveis

- Dificuldade em criar ou estender gramáticas;
- Complexidade em adaptar para múltiplas línguas ou domínios;
- Necessidade de conhecimento linguístico, por forma a adaptar a gramática às regras de uma determinada língua;
- Ambiguidade da semântica. Diferentes expressões podem ou devem gerar o mesmo SQL;
- Manutenção difícil.

4.1.2 Pesquisa Semântica

Nesta abordagem usou-se uma *framework* que não consta nas ferramentas estudadas (ver Secção 3.4), principalmente porque o seu desenvolvimento encontra-se estagnado desde 2013. Ainda assim, contemplou-se o seu uso para testar esta abordagem, no ponto de vista prático. O Quepy foi desenvolvido em *Python* e tem como objetivo a transformação de linguagem natural em *SPARQL Protocol and RDF Query Language (SPARQL)*¹ e *Metaweb Query Language (MQL)*², linguagens de *query* usadas com tecnologia RDF³, “um modelo padrão para permuta de dados na Web”⁴, intrinsecamente ligado ao campo da Web Semântica (W3C 2019).

Neste caso, o uso do Quepy implicou a definição da linguagem específica de domínio através de código *Python*, respeitando as restrições colocadas na documentação da *framework*, de maneira a que esta fosse capaz de reconhecer o padrão introduzido.

```
from quepy.dsl import FixedType, FixedRelation
from quepy.parsing import Lemma, QuestionTemplate

class NameOf(FixedRelation):
    "Name of the entity's property"
    relation = "product_name"
    reverse = True

class IsProduct(FixedType):
    "Defines the entity's type"
    fixedtype = "product"

class ListProducts(QuestionTemplate):
    "Questions such as 'List products' or 'Listing products'"
    regex = Lemma("list") + Lemma("product")

    def interpret(self, match):
        product = IsProduct()
        name = NameOf(product)
        return name, "enum"
```

Código 4.2: Excerto da definição semântica da frase que lida com listagem de produtos

¹Disponível em <https://www.w3.org/TR/rdf-sparql-query/>.

²Disponível em <https://github.com/nchah/freebase-mql>.

³Disponível em <https://www.w3.org/RDF/>.

⁴Tradução livre do autor. No original “is a standard model for data interchange on the Web.”.

No excerto demonstrado em 4.2, define-se a expressão regular que caracteriza a questão esperada, para além de detalhes relacionados com a linguagem específica de domínio. A *query* obtida pode ser posteriormente usada numa base de dados RDF, como o objetivo de recolher os dados. De notar que, neste caso, sendo o propósito a consulta de uma base de dados relacional, seria necessária uma forma de a transpor ou replicar numa camada compatível com tecnologia RDF. Para isso, existem ferramentas capazes de executar esse mapeamento, como por exemplo o D2RQ⁵.

Pontos Favoráveis

- Viabilidade de desenvolvimento de uma solução customizada;
- Tecnologia *open source*;
- Linguagem específica de domínio permite abstração aos detalhes de implementação associados ao reconhecimento de linguagem natural;
- Reconhecimento da linguagem natural é feita através da análise de expressões regulares, o que torna fácil compreender a estrutura frásica;
- Transformação da linguagem natural numa representação que obedece à especificação da tecnologia RDF, definida pela W3C.

Pontos Desfavoráveis

- As regras de domínio e o modelo de dados devem ser definidos diretamente no código, não possibilitando a extensibilidade por configuração;
- Rigidez da linguagem usada, já que o uso de expressões regulares implica uma estrutura minimamente estática para que o reconhecimento seja bem sucedido;
- O uso de tecnologias RDF levam a uma nova camada aplicacional, que envolve o uso de ferramentas não relacionadas com o PLN e que leva a mais esforço na manutenção do sistema;
- Necessidade de estudo aprofundado ou conhecimento pericial na área de Web Semântica.

4.1.3 Reconhecimento de Intenções e Entidades

Neste contexto, utilizou-se o Microsoft LUIS, uma das ferramentas previamente estudadas (ver Secção 3.4). A razão para esta escolha deve-se principalmente a dois fatores: o facto de ser uma plataforma bastante documentada e pela Microsoft disponibilizar (ainda que de forma limitada) este tipo de ferramentas a quem possua subscrição de estudante.

Para esta prova de conceito criou-se um *chatbot* capaz de responder a algumas perguntas colocadas pelo utilizador, identificando qual a sua intenção e quais as entidades envolvidas, como se apresenta a seguir:

- *Good Morning* – corresponde à intenção de *Greeting* (saudação), não existindo entidades associadas;
- *What's the weather like in Porto?* – intenção de *CheckWeather* (verificar condições meteorológicas), cuja entidade dinâmica é o *Place* (lugar) indicado pelo utilizador.
- *Book me a flight to Lisbon next week* – intenção de *BookFlight* (agendar voo), que possui duas entidades dinâmicas: o lugar de destino (*Place*) e o momento da viagem (*Datetime*).

⁵Disponível em <http://d2rq.org/>.

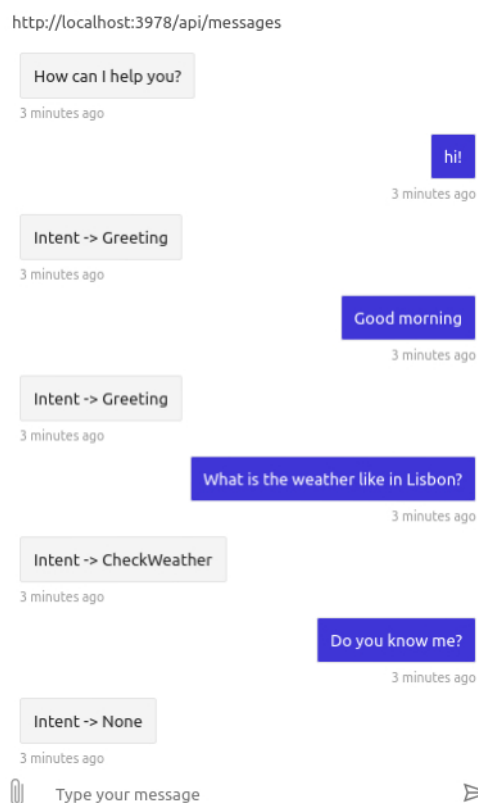


Figura 4.1: Amostra de teste com *chatbot* desenvolvido usando Microsoft LUIS

O *chatbot* recorreu ao Microsoft LUIS para a identificar a intenção do utilizador, e com base nisso, respondia com a intenção e entidades que foi capaz de identificar (ver Figura 4.1). Notou-se ser possível dotar o *chatbot* de comportamento para manipular os metadados obtidos, e assim, obter dados de uma base de dados relacional ou obter dados estáticos dum serviço de respostas pré-fabricadas, tal como o QnA Maker⁶.

Pontos Favoráveis

- As plataformas disponíveis na *cloud* constituem um ponto de partida para uma solução personalizada;
- A solução é fácil de desenvolver, estender e integrar;
- A base de conhecimento pode ser totalmente configurável;
- Robustez na identificação de intenções e entidades, através do uso de modelos de ML;

Pontos Desfavoráveis

- A aplicação desta abordagem numa biblioteca customizada implica o estudo teórico de ML aplicado ao PLN e conseqüentemente, maior esforço de desenvolvimento;
- A adição de novas intenções à base de conhecimento leva também à adição de comportamento para lidar com os mesmas.

⁶Disponível em <https://www.qnamaker.ai/>

4.1.4 Sinopse

As abordagens descritas anteriormente baseiam-se nas observações e experiências realizadas, no ponto de vista prático. Portanto, a conclusão aqui exposta leva em consideração os pontos favoráveis e desfavoráveis de cada uma.

A abordagem que parece a mais adequada é a que diz respeito ao reconhecimento de intenções e entidades, principalmente pela facilidade de compreensão e aplicação do conceito. Além do mais, os pontos desfavoráveis mencionados são de índole operacional, pelo que não parecem comprometer a concepção de um modelo e desenvolvimento do respetivo protótipo.

Relativamente às abordagens descartadas, aponta-se que a primeira – gramática baseada em semântica – se usada no contexto comercial, seria difícil manter o seu desenvolvimento e capacidade de cobrir uma gama aceitável de questões válidas. Já a segunda – pesquisa semântica –, ainda que assente sobre uma tecnologia especificada e aprovada pelo W3C, necessitaria de uma camada adicional (RDF), contribuindo assim para o aumento do esforço em configuração e manutenção do sistema. Por isso, a terceira abordagem revela-se a mais adequada e será contemplada no protótipo a desenvolver.

4.2 Modelo Proposto

O estudo das ILN, associado à aplicação prática de algumas ferramentas e abordagens estudadas, permitiram retirar conclusões relevantes de considerar na concepção do módulo. Nesta secção apresenta-se o modelo idealizado segundo uma abordagem de reconhecimento de intenções e entidades.

4.2.1 Visão Geral

A visão incide na concepção um módulo de linguagem natural para interface com o Critical Manufacturing MES, que permita aos operadores de linha, engenheiros de produção e gestores, consultar informação acerca do processo fabril. Como se apresenta na Figura 4.2, o sistema MES disponibiliza uma interface para que o utilizador interaja com ele, através de texto, esperando obter informação relevante do processo, no formato que melhor se enquadrar.

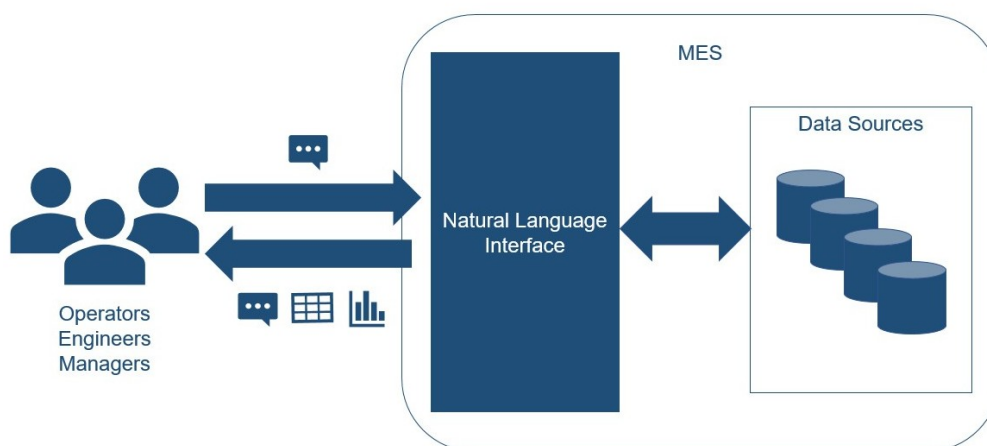


Figura 4.2: Visão da solução desejada para o produto

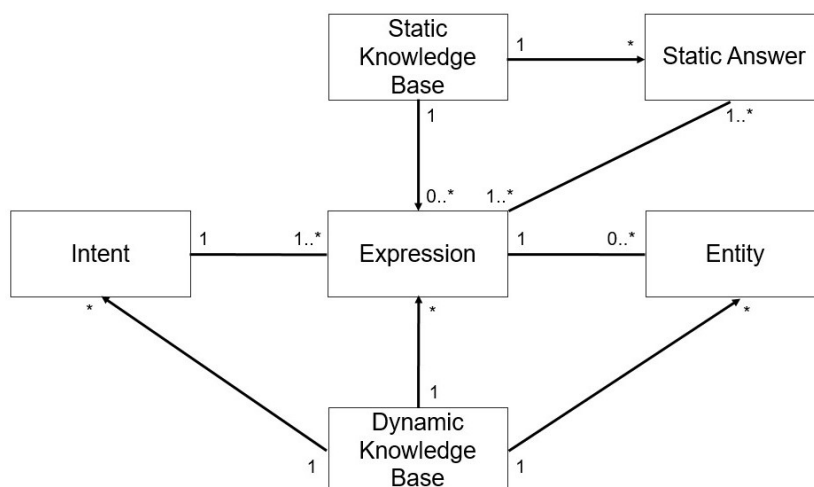


Figura 4.3: Conceitos associados ao reconhecimento de intenções e entidades e os respectivos relacionamentos

O modelo tem em conta três fatores: a experiência de utilizador, a extensibilidade das bases de conhecimento e integração com o Critical Manufacturing MES. Relativamente à estratégia de reconhecimento de intenções e entidades a ser aplicada no modelo, definem-se os conceitos que lhe estão inerentes, apresentados na Figura 4.3:

- *Dynamic Knowledge Base* – Base de conhecimento dinâmica. Fonte de dados que liga as expressões às intenções e entidades correspondentes e é usado no treino dos modelos ML para previsão;
- *Static Knowledge Base* – Base de conhecimento estática. Dicionário onde consta uma determinada expressão e respetiva resposta. Pode ser usada para implementar conversação casual ou lidar com expressões que não constam na base de dados dinâmica;
- *Intent* – Intenção. Mapeia ação que o utilizador deseja executar. Normalmente estão-lhe associadas diversas expressões;
- *Expression* – Expressão. Corresponde ao exemplo de estrutura de uma pergunta que o utilizador pode fazer;
- *Entity* – Entidade. Tipicamente refere-se a um conceito de domínio ou do mundo real;
- *Static Answer* – Resposta Estática. Definida na base de conhecimento estática, associada a diferentes expressões. Por exemplo, expressões como “Olá” ou “Viva” podem ter associadas respostas como “Oi” e “Olá”.

Portanto, uma intenção é um conceito composto, base do modelo, que se apresenta como uma indireção face à questão colocada pelo utilizador, ou seja, a questão é analisada pela ação que lhe está associada e não propriamente pelo seu significado, *e.g.*, um pedido para obter os materiais mais fabricados num determinado setor constitui uma intenção única, apesar de apresentar entidades diferentes, consoante o domínio.

4.2.2 Casos de Uso

De um ponto de vista de utilização, o modelo aqui proposto deve saber lidar com *Perguntas*, *Respostas* e *Bases de Conhecimento*, que se podem considerar as principais áreas funcionais de sistema. Todas têm implicações no comportamento do módulo. Em relação às *Perguntas* e *Respostas*, uma vez que estão profundamente ligadas, apresentando funcionalidades comuns, são então enquadradas numa

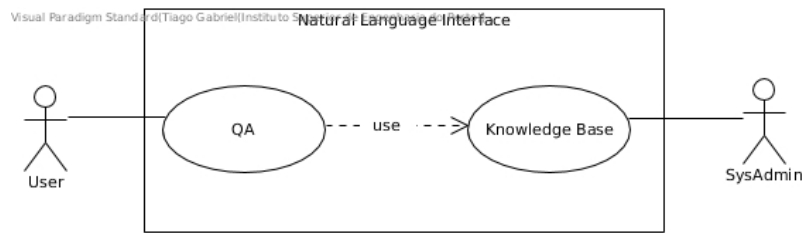
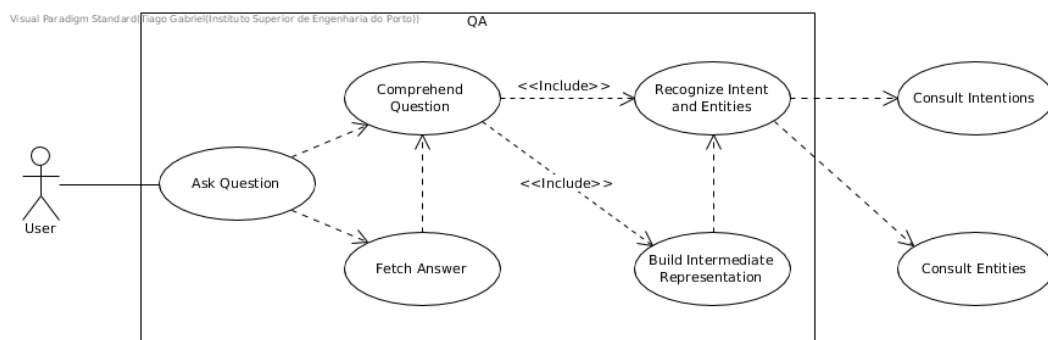


Figura 4.4: Áreas funcionais de sistema

área funcional única, denominada *QA*. A Figura 4.4 dá uma visão geral das áreas funcionais do sistema, que são descritas em seguida:

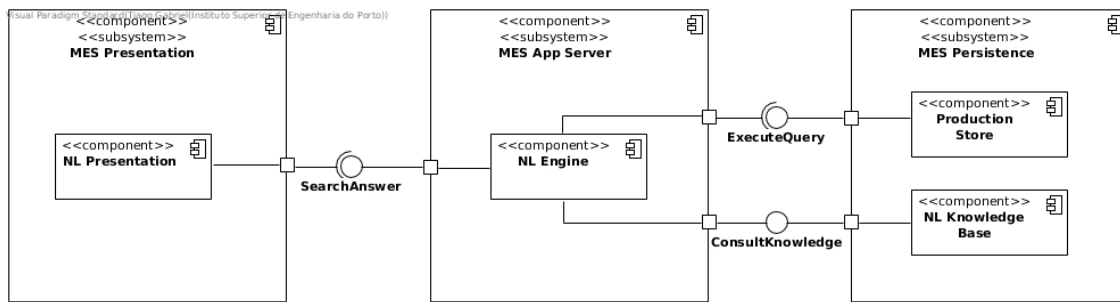
- *QA* – corresponde ao conjunto de funcionalidades relacionadas com as perguntas colocadas pelos utilizadores e respetiva procura de respostas;
- *Knowledge Base* – área funcional *Base de Conhecimento*. Apresenta funcionalidades para a gestão de intenções, expressões, entidades, ou seja, tudo o que as restantes áreas funcionais necessitam para o seu funcionamento.

A área funcional *QA* é essencial no contexto deste trabalho, já que se trata da base de interação com o utilizador. Por isso, a Figura 4.5, apresentada a seguir, detalha algo mais essa área funcional.

Figura 4.5: Casos de uso identificados para a área funcional *QA*

Os casos de uso apresentados no diagrama são descritos a seguir:

- *Ask Question* – Fazer Pergunta. É a funcionalidade principal. O utilizador faz a pergunta com o objetivo de obter a resposta que procura. Depende da compreensão da pergunta e da obtenção de resposta;
- *Comprehend Question* – Compreender a Pergunta. O sistema procura compreender a pergunta feita e traduz para uma representação passível de ser usada na fase de procura da resposta nas fontes de dados disponíveis;
- *Recognize Intent and Entities* – Reconhecer a Intenção e as Entidades. O sistema faz o reconhecimento da intenção e das entidades da pergunta feita com base no modelo de ML treinado com os dados que constam na base de conhecimento;
- *Build Intermediate Representation* – Construir Representação Intermédia. O sistema constrói uma representação intermediária que contém os metadados da pergunta colocada – intenção, entidades e outros dados relevantes;
- *Consult Intentions* – Consultar as Intenções. Permite a consulta de intenções mantidas na base de conhecimento;



(a) Arquitetura genérica do protótipo

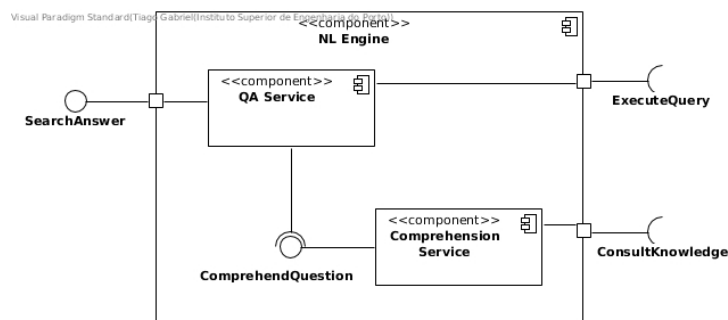
(b) Arquitetura detalhada do *NL Engine*

Figura 4.6: Arquitetura do protótipo, apresentando um vista genérica e uma mais específica do componente *NL Engine*

- *Consult Entities* – Consultar as Entidades. Permite a consulta as entidades mantidas na base de conhecimento;
- *Fetch Answer* – Obter a Resposta. O sistema usa o resultado (representação intermediária) da fase de compreensão da pergunta para converter essa representação numa compatível com a fonte de dados a interagir, obtendo assim a resposta.

Os casos de uso apresentados são a base das funcionalidades do módulo, e por isso, serão explorados em contexto prático no capítulo seguinte.

4.2.3 Arquitetura

Dada a visão geral do modelo e dos seus casos de uso, pretende-se expor a estrutura da solução numa perspetiva lógica, enquadrando-a no contexto Critical Manufacturing MES. Assim, a Figura 4.6 demonstra a arquitetura do modelo, primeiro num vista a alto nível e depois mais focado no elemento principal, explicando a responsabilidade inerente a cada componente.

- *NL Presentation* – responsável pela interação com o utilizador. Integra a camada de apresentação do Critical Manufacturing MES e, como tal, é desenvolvido de acordo com as especificidades do subsistema em que se insere;
- *NL Engine* – o módulo de linguagem natural, ou seja, o “motor” que permite a tradução de linguagem natural em intenções e respetivas entidades. Integra a camada aplicacional de serviços do produto;
- *QA Service* – trata-se de um subcomponente o *NL Engine*, que conhece as partes envolvidas no processo de aquisição de resposta, sendo responsável por orquestrar esse processo. Trabalha em conjunto com o *Comprehension Service* com o objetivo de identificar e mapear a intenção

e entidades de uma dada *query* de linguagem natural para obter a resposta da fonte de dados produtivos. Numa analogia à anatomia humana, pode ser considerado o “cérebro” do processo;

- *Comprehension Service* – outro subcomponente do *NL Engine*, trabalha com a base de conhecimento definida (*NL Knowledge Base*) para executar a tarefa de compreender a pergunta colocada. É neste componente que se insere a ferramenta escolhida para processamento de linguagem natural;
- *NL Knowledge Base* – base de conhecimento de domínio, incluída na camada de persistência do Critical Manufacturing MES. É configurada pela equipa de desenvolvimento e deve mapear as entidades de domínio e as intenções em que estão envolvidas;
- *Production Store* – armazém de dados de negócio. Contém a informação que o utilizador deseja obter através de linguagem natural.

Apesar da elucidação acerca da responsabilidade de cada componente no sistema, é importante detalhar a forma como estes interagem entre si, para atingir o objetivo. A Figura 4.7 mostra como se desenrola a comunicação entre os diversos componentes, que se passa a descrever: o utilizador (*User*) questiona o sistema através da interface gráfica (*NL Presentation*). A questão é encaminhada para o *QA Service* que se encarrega de “pedir” ao *Comprehension Service* para que lhe forneça a compreensão sob a forma de representação intermediária. Posto isto, o *Comprehension Service* consulta a base de conhecimento (*NL Knowledge Base*) para obter o conteúdo existente e, aplicando modelos de ML, faz o reconhecimento das intenções e entidades presentes na questão. O *QA Service* trata de converter a representação numa linguagem compatível com o *Production Store* e executa a *query* gerada. Aquando a aquisição dos dados, o *QA Service* trata de “documentá-los”, ou seja, colocar metadados que sejam importantes para processamento posterior. Por fim, a *NL Presentation* processa o resultado para que este seja apresentado num formato adequado ao utilizador.

4.3 Síntese

Neste capítulo descreveu-se o processo que levou à especificação do módulo, dando ênfase aos detalhes da sua arquitetura.

Começou-se por descrever a apreciação feita, em contexto prático, a algumas das ferramentas e abordagens anteriormente estudadas, fazendo referência aos seus pontos favoráveis e desfavoráveis. A conclusão retirada foi a de que a abordagem de reconhecimento de intenções e entidades seria a escolha adequada para o modelo a conceber.

Posteriormente, apresentou-se o modelo proposto, descrevendo a visão contemplada, as áreas funcionais – *QA* e *Knowledge Base* – e os casos de uso identificados, detalhando os mais importantes. Finalmente, a arquitetura num ponto de vista lógico, explicando os componentes e o fluxo de trabalho entre eles, no qual se apresenta um cenário genérico do funcionamento do processo de tradução e resposta.

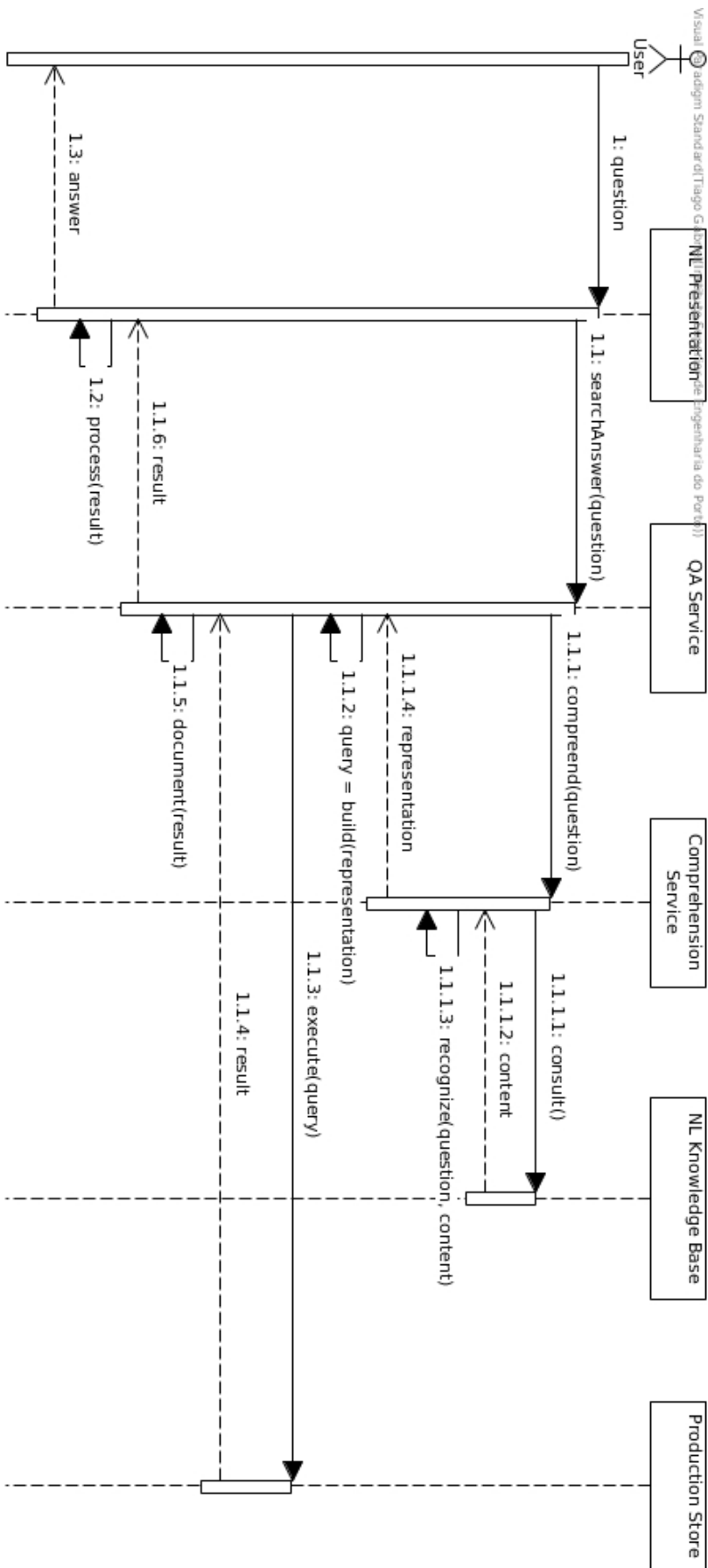


Figura 4.7: Comunicação entre os componentes do módulo de linguagem natural

Capítulo 5

Desenvolvimento

Após a idealização do modelo é conveniente a sua aplicação prática. Nesse sentido, este capítulo aborda o processo de desenvolvimento do protótipo, um *chatbot* que recebeu o nome *Proto*, construído com base no modelo proposto, usando os serviços cognitivos da Microsoft disponíveis na *cloud*. No decorrer do capítulo enuncia-se o processo de implementação do protótipo, fazendo referência à transposição do modelo para a prática, da configuração necessária, nomeadamente das bases de conhecimento, e detalhes relevantes da desenvolvimento desta prova de conceito.

5.1 Especificação

Dado o modelo proposto (ver Capítulo 4), é crucial a sua transposição para a realidade dum *chatbot*. Como referido anteriormente, este foi desenvolvido usando os serviços cognitivos da Microsoft, sendo então pertinente a introduzir alguns conceitos gerais e a arquitetura que foi especificada.

5.1.1 Visão Geral

Um *chatbot* é um tipo de ILN que providencia uma experiência rica ao utilizador, dando-lhe a sensação que está a comunicar com outro ser humano (Microsoft 2019a, Concepts). Conforme descrito em Microsoft (2019a, Principles of bot design), um *chatbot*, tal como qualquer outra interface, para além de garantir uma ótima experiência de utilização, deve também seguir quatro orientações:

1. Resolução do problema do utilizador com o mínimo de etapas possível;
2. Apresentação de uma solução de forma mais fácil e rápida do que as suas alternativas;
3. Disponibilidade em diferentes dispositivos e plataformas;
4. Fácil acessibilidade, garantindo que o utilizador sabe exatamente o que fazer.

Neste quadro, o protótipo deve ter a capacidade de conversação aliada à responsabilidade de transmitir a informação pretendida, quando requisitada pelo utilizador. Para este efeito, os serviços cognitivos da Microsoft (Azure Cognitive Services) disponibiliza dois serviços (Microsoft 2019b): o Microsoft LUIS e QnA Maker (ver Figura 5.1).

O Microsoft LUIS mantém a base de conhecimento referente às intenções e entidades, tendo a capacidade de compreensão de linguagem natural. Por outro lado, o QnA Maker age como um dicionário, no qual contém respostas estáticas face a perguntas pré-definidas que não tenham uma intenção específica. Ao *chatbot* cabe a responsabilidade de orquestrar o processo de conversação, ficando encarregue de executar as ações vinculadas à intenção identificada pelo Microsoft LUIS ou “conversar” com o utilizador, quando nenhuma intenção é identificada, obtendo respostas-padrão contidas QnA Maker.

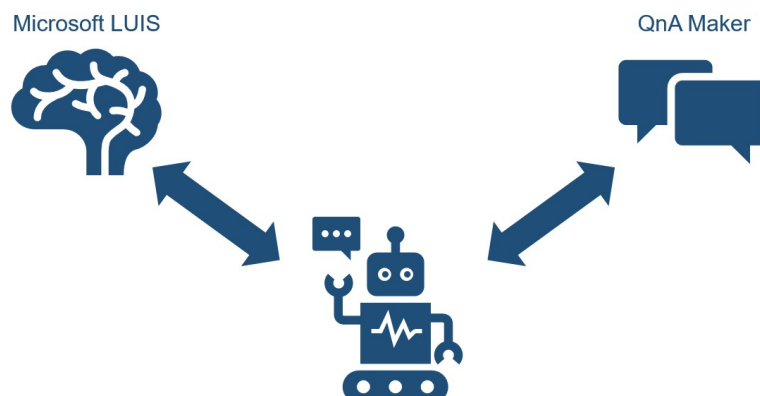


Figura 5.1: Vista geral da interação entre o *chatbot* e os serviços disponibilizados

Para trabalhar com estes serviços, a Microsoft disponibiliza um conjunto de ferramentas *open source*, Bot Framework Tools¹, que ajudam no desenvolvimento e suporte de soluções. No caso do Microsoft LUIS, é também facultada uma API na *Web*.

5.1.2 Arquitetura

Tendo em conta a visão geral apresentada e o modelo proposto, a Figura 5.2 mostra a arquitetura definida para o protótipo, ao qual se deu o nome de *Proto*. A seguir, faz-se uma descrição de cada componente e de que forma se enquadra na arquitetura do modelo previamente especificado.

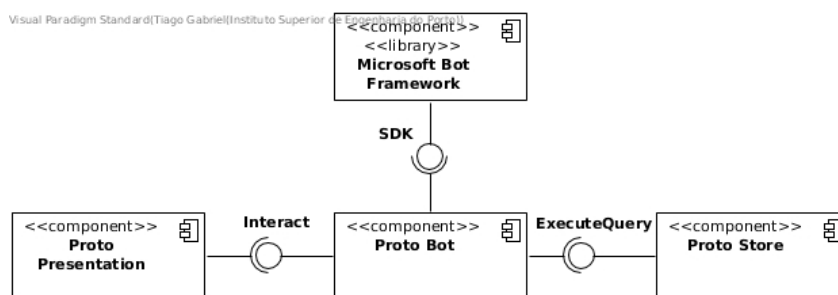


Figura 5.2: Arquitetura do protótipo, seguindo o modelo proposto

- *Proto Presentation* – responsável pela interação com o utilizador. No modelo corresponde ao componente *NL Presentation*;
- *Proto Bot* – o *chatbot* propriamente dito. Tal como o seu análogo no modelo, o *QA Service*, conhece as partes envolvidas no processo, portanto orquestra-o;
- *Microsoft Bot Framework* – a biblioteca que dispõe o acesso às bases de conhecimento, possibilitando o reconhecimento das entidades e a aquisição de respostas estáticas. Por outras palavras, permite o acesso ao Microsoft LUIS e ao QnA Maker. Equivale ao *Comprehension Service* presente no modelo proposto;
- *Proto Store* – simula o armazém de dados de negócio. Análogo do componente *Production Store*.

¹Disponível em <https://github.com/microsoft/botframework#bot-framework-tools>.

Tabela 5.1: Intenções e entidades dada a expressão de exemplo

Expressão de Exemplo	Intenção	Entidades
What's the total of operations per week?	SumOperationsByWeek	-
What's the total of operations per week on <u>January</u> ?	SumOperationsByWeek	DatetimeV2
How many <u>trackout</u> operations were done by product?	CountOperationsByProduct	DimensionValue
How many <u>trackout</u> operations were executed by product on <u>February</u> ?	CountOperationsByProduct	DimensionValue, DatetimeV2
How many <u>trackout</u> operations were executed by product on <u>February</u> , per shift?	CountOperationsByProductPerShift	DimensionValue, DatetimeV2
What's the average <u>primary quantity</u> of <u>trackin</u> operations on <u>burnin</u> step?	AverageOperationsOnStep	Dimension, DimensionValue
What's the average <u>primary quantity</u> of <u>trackin</u> operations on <u>burnin</u> step on <u>January</u> ?	AverageOperationsOnStep	Dimension, DimensionValue, DatetimeV2
How many materials are with <u>primary quantity less than 1000</u> ?	CountMaterialsOnCondition	Dimension, Operator, Number
How many materials, on <u>wire inspection</u> step have a <u>primary quantity inferior to 1000</u> ?	CountMaterialsOnConditionOnStep	Dimension, DimensionValue, Operator, Number
How many materials by <u>operation</u> on <u>wire inspection</u> step have <u>primary quantity over 1000</u> ?	CountMaterialsOnConditionOnStepGrouped	Dimension, DimensionValue, Operator, Number

5.2 Configuração

Antes de partir para a implementação do *Proto*, primeiro é mandatório a configuração das suas bases de conhecimento. No panorama dos serviços cognitivos da Microsoft, para este tipo de configuração, podem ser usados ficheiros JSON, *Markdown* ou o portal disponibilizado para cada uma das ferramentas. No desenvolvimento deste protótipo optou-se por usar o portal, pois não se pretende dar suporte à aplicação, mas sim aplicar o modelo especificado.

5.2.1 Respostas Estáticas

As respostas estáticas dizem respeito à capacidade de conversação do *chatbot*, ou seja, questões que tenham respostas constantes, *i.e.*, uma pergunta referente à idade do *Proto* deve ter uma resposta pré-definida. O QnA Maker é responsável por manter esta base de conhecimento.

O processo de criação duma base de conhecimento similar está descrita detalhadamente na documentação² disponibilizada pela Microsoft. O *Proto* não contém respostas estáticas para além das que dizem respeito à sua personalidade, e cujo conjunto de dados, que descrevem os cenários de conversação, é disponibilizado por predefinição no portal.

5.2.2 Intenções e Entidades

Relativamente a esta base de conhecimento, mantida no Microsoft LUIS, consideram-se as perguntas definidas inicialmente, no âmbito dos critérios de sucesso para o trabalho (ver Secção 1.4). A análise das perguntas leva a variações, sendo apresentadas na Tabela 5.1, juntamente com as intenções e entidades que lhes corresponde. Uma observação cuidada mostra que, referente à mesma intenção, podem existir diferentes entidades. Em geral, isto deve-se ao facto de algumas delas, como a *DatetimeV2* e *Value*, serem entidades pré-fabricadas, ou seja, encontram-se na plataforma Microsoft LUIS por omissão, sendo inferidas automaticamente. Portanto, deixa de ser necessário definir uma intenção específica para superar este tipo de casos. Para todos os efeitos, a documentação da Microsoft detalha este tipo de entidades. Nos restantes casos, as entidades são definidas pelo desenvolvedor, podendo ter funções (*Roles*) consoante a sua posição na frase, detalhe que também se pode encontrar a documentação oficial. Com base nestas funções, torna-se possível a existência de múltiplos conceitos referentes à mesma entidade, para a mesma frase.

²Disponível em <https://docs.microsoft.com/en-us/azure/cognitive-services/QnAMaker/Quickstarts/create-publish-knowledge-base>.

Terminada a configuração, a base de conhecimento é publicada no Microsoft Azure, no serviço aplicacional criado para o efeito. Quaisquer alterações ou melhorias da base de conhecimento existente levam a um novo treino do modelo e consequentemente, a uma nova publicação.

5.3 Implementação

Após terminada a fase de configuração, segue-se a implementação do protótipo. Nesta fase não foi considerado o acesso a uma fonte de dados relacional, pelo facto de não ser relevante na validação do modelo especificado. Ao invés, usa-se o conjunto de dados, em formato textual, fornecido pelo supervisor deste trabalho, para simulação da fonte de dados de domínio. Portanto, a solução opta por extrair a informação necessária dos metadados gerados pelo Microsoft LUIS, e com base na intenção cria a *query* adequada para obter o conteúdo relevante, e finalmente devolve a resposta ao utilizador (ver Código 5.1 e Figura 5.3). Para o caso da geração de SQL, a estratégia seria muito semelhante. Por exemplo, para a questão *How many trackout operations were executed by product on February?*, um excerto da resposta devolvida pelo Microsoft LUIS é apresentada em seguida:

```
{
  "query": "how many trackout operations were executed by product on february?",
  "prediction": {
    ...
    "topIntent": "CountOperationsByProduct",
    ...
  },
  "entities": {
    ...
    "$instance": {
      "Operation": [
        {
          "role": "Operation",
          "type": "DimensionValue",
          "text": "trackout",
          "startIndex": 9,
          "length": 8,
          "score": 0.9338435,
          "modelType": 1,
          "modelType": "Entity Extractor",
          "recognitionSources": [
            "model"
          ]
        }
      ]
    },
    "datetimeV2": [
      {
        "type": "builtin.datetimeV2.daterange",
        "text": "february",
        "startIndex": 57,
        "length": 8,
        "modelType": 2,
        "modelType": "Prebuilt Entity Extractor",
        "recognitionSources": [
          "model"
        ]
      }
    ]
  }
}
}
```

Código 5.1: Excerto do JSON devolvido pelo Microsoft LUIS

Os diagramas apresentados na Figura 5.4 procuram detalhar a forma como foi desenhado o processo de extração de conhecimento. Ao receber uma mensagem, o *Probot* recolhe o resultado devolvido pelo Microsoft LUIS e procura criar um novo *Intent* de domínio aplicacional. Desta forma, usa-o para obter a resposta que deve apresentar. Por sua vez, o *Intent* extrai a informação necessária do *LuisResult*, gera a *query* para comunicação com a fonte de dados (*ProtoStore*) e finalmente, procura obter a informação necessária, gerando uma resposta adequada para o utilizador.

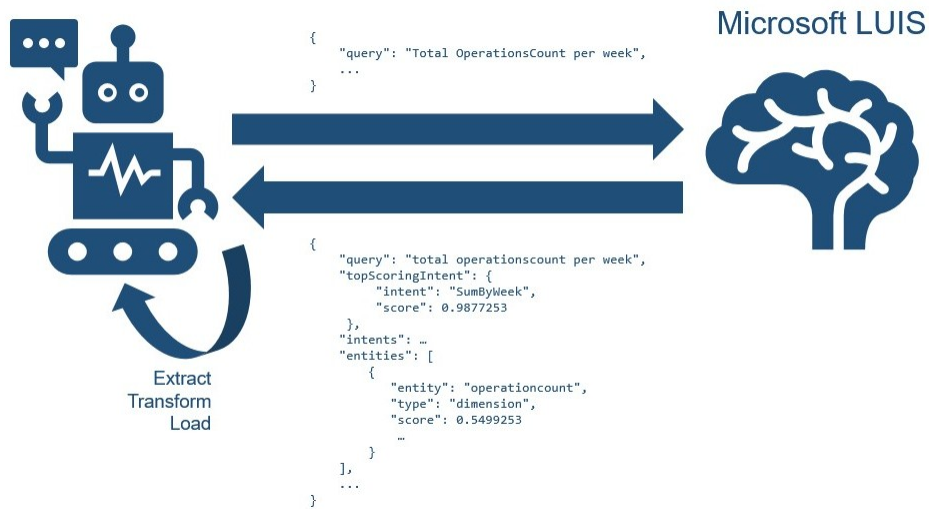


Figura 5.3: Esquema referente à comunicação do *chatbot* com o Microsoft LUIS

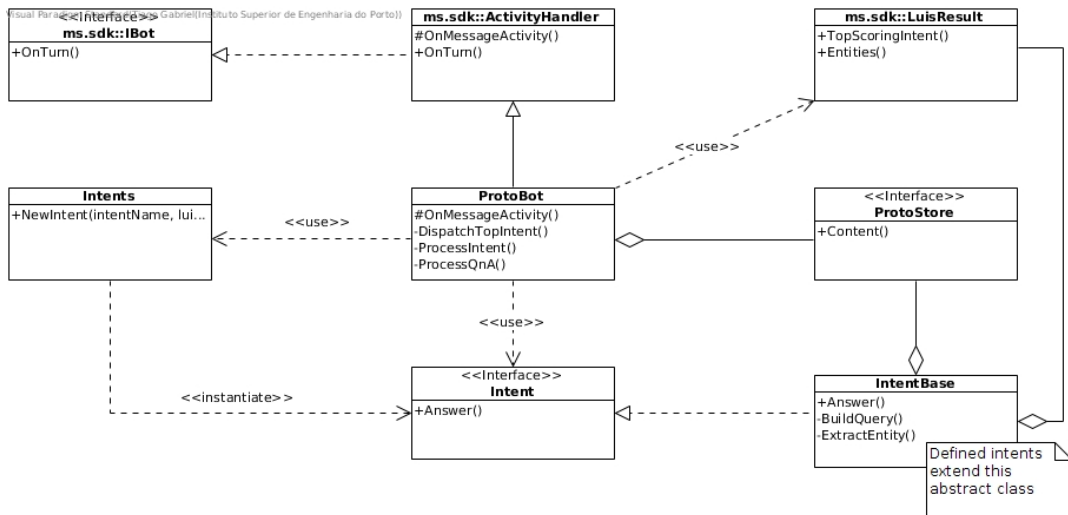
5.4 Síntese

Neste capítulo abordaram-se os principais detalhes do protótipo *Proto*, que foi desenvolvido de acordo com o modelo proposto.

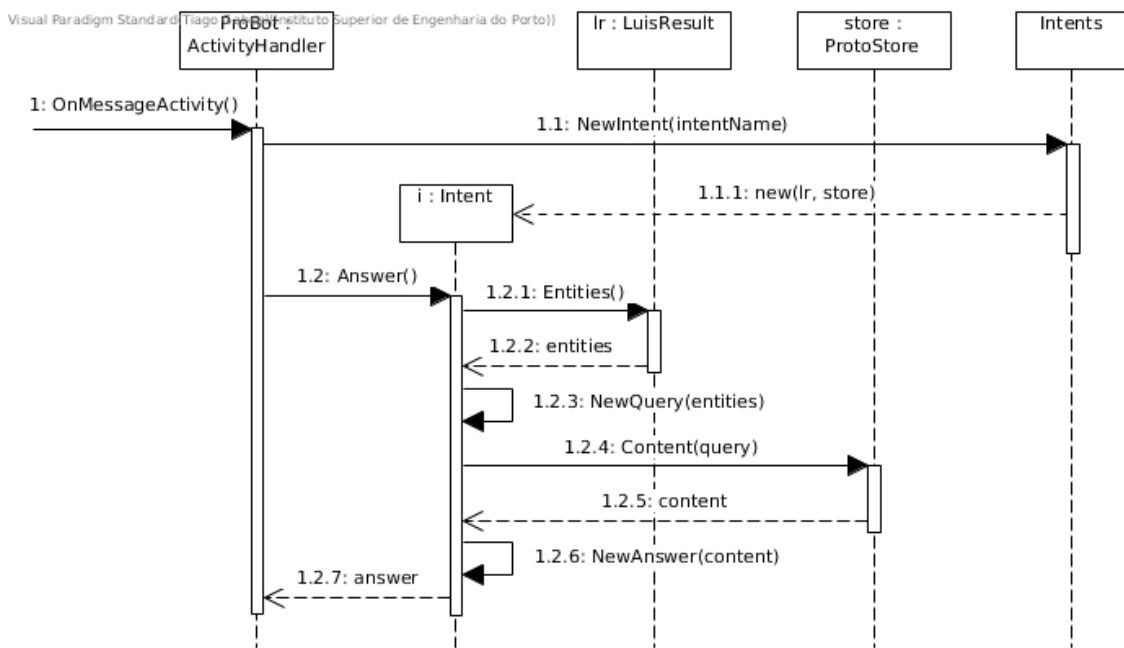
Inicialmente, expôs-se a sua especificação, tentando-se explicar a forma como o Microsoft LUIS e o QnA Maker seriam usados no contexto deste desenvolvimento e enquadrando a arquitetura do *Proto* com a do modelo previamente idealizado.

Seguiu-se a interpelação ao processo de configuração das bases de conhecimento usando os serviços Microsoft, quer para respostas estáticas, quer para intenções e entidades. Relativamente a esta última, demonstraram-se aquelas que seriam esperadas para o protótipo, levando em consideração as perguntas definidas no âmbito dos critérios de sucesso (ver Secção 1.4).

No final, explicaram-se os conceitos particulares da implementação do *Proto*, designadamente a forma como os metadados obtidos do Microsoft LUIS são extraídos, transformados e usados para o carregamento de informação da respetiva fonte de dados.



(a) Vista genérica de classes usadas no protótipo



(b) Vista genérica do fluxo das classes usadas no protótipo

Figura 5.4: Diagramas referentes à estrutura do protótipo na extração, transformação e carregamento da informação de domínio

Capítulo 6

Validação

O atual capítulo apresenta o processo de validação do modelo proposto, levando em consideração o protótipo desenvolvido. O objetivo é explicar em que se debruçou este processo, demonstrando os resultados obtidos. Primeiramente, foca-se na capacidade do protótipo compreender a linguagem natural, a nível adimensional (sem entidades), unidimensional (com uma entidade) e multidimensional (com várias entidades), ou seja, tenta-se averiguar a avaliação correta das intenções e entidades das questões. De seguida, avalia-se a resposta às questões-chave, procurando-se evidências na competência da extração da informação que se pretende pesquisar com a linguagem natural.

6.1 Compreensão da Linguagem Natural

O correto funcionamento do *Proto* depende, antes de tudo, da compreensão da linguagem natural e conseqüentemente, da extração dos conceitos ligados à pergunta colocada. Por isso, usaram-se as perguntas definidas no âmbito dos critérios de sucesso, e apresentadas na Tabela 5.1, no processo de validação. Na Figura 6.1 mostram-se algumas evidências da análise de perguntas e respetiva extração de intenção e entidades.

O processo de validação consistiu na submissão das perguntas no portal do Microsoft LUIS, o qual possui uma funcionalidade que permite testar o modelo ML configurado. A submissão de uma pergunta resulta numa “inspeção” completa, discriminando a intenção, classificação resultante, e quais as entidades encontradas na expressão.

Efetivamente, os testes realizados demonstram que protótipo foi capaz de avaliar as perguntas colocadas face à intenção e entidades esperadas visto que, das dez questões avaliadas, ele conseguiu classificá-las de acordo com os critérios. Por outras palavras, valida-se o que protótipo identifica a linguagem natural adimensional, unidimensional e multidimensional.

6.2 Resposta às Questões-Chave

Nesta fase da validação, o intuito é avaliar a transformação da representação intermediária e conseqüente extração da informação. Para esta validação, considerou-se parte do conjunto de dados fornecido pelo supervisor do projeto, que é usado pelo protótipo como fonte de dados. Computaram-se os resultados esperados manualmente, usando uma folha de cálculo, de forma a poder compará-los com a resposta obtida (ver Figura 6.2). Posto isto, os cenários considerados para validação são apresentados em seguida:

1. Nível de linguagem adimensional – usa-se a questão *What's the total of operations per week?*;
2. Nível de linguagem unidimensional – a questão considerada é *How many trackout operations were done by product?*;

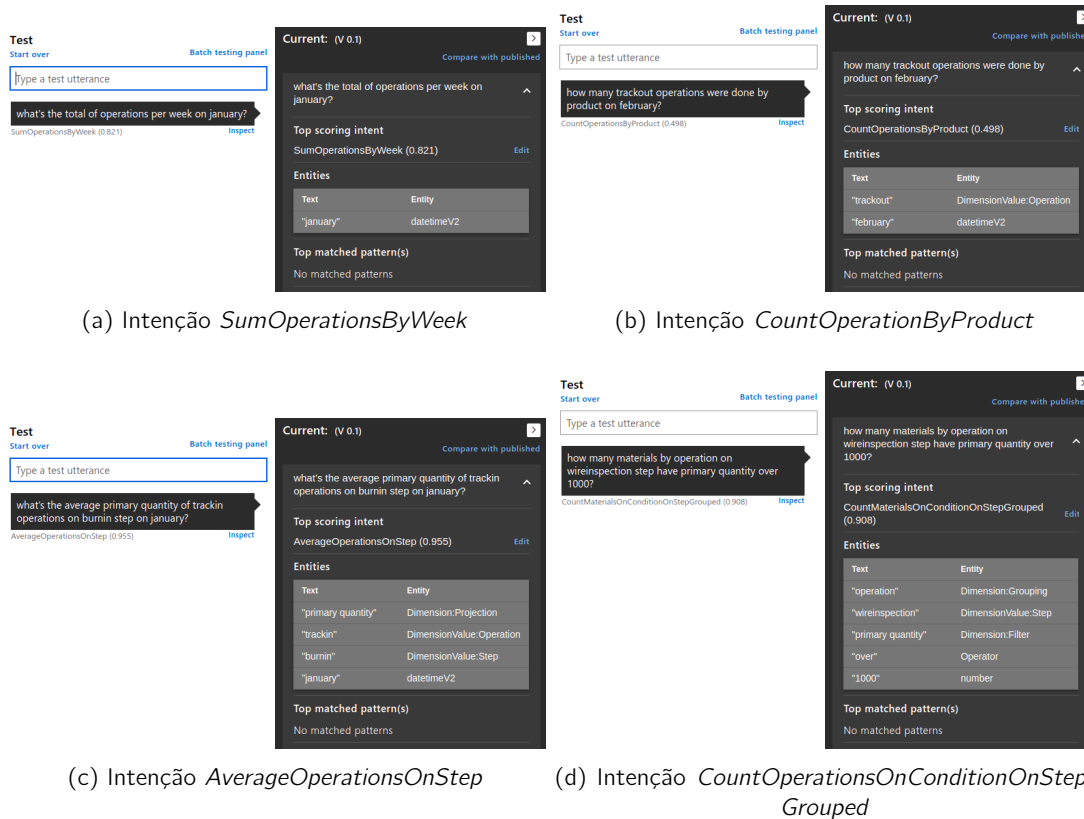


Figura 6.1: Imagens referentes à avaliação de intenções e entidades do protótipo

3. Nível de linguagem multidimensional – corresponde à questão *How many trackout operations were executed by product on February, per shift?*

Na Figura 6.3 apresentam-se as respostas apresentadas pelo protótipo, no contexto dos critérios apresentados. Adicionalmente, demonstra-se também a forma como o protótipo “reage” quando não possui a resposta que o utilizador espera (ver Figura 6.3d). Como se pode observar, a solução consegue dar resposta às perguntas colocadas, estando de acordo com as respostas esperadas.

6.3 Síntese

Este capítulo abordou o processo de validação do protótipo desenvolvido, tendo em conta dois requisitos: a compreensão da linguagem natural nos vários níveis considerados e a resposta às questões-chave apresentadas no capítulo anterior.

Para a primeira verificou-se que a solução é capaz de lidar com a linguagem natural, extraindo as intenções e entidades que são esperadas.

No caso da segunda, demonstrou-se que o protótipo responde a questões-chave, e também que é capaz de responder adequadamente em caso de não conter a resposta procurada pelo utilizador.

	A	B
1	FullDate	(All)
2		
3	Row Labels	Sum of OperationCount
4	1	1316
5	2	32391
6	3	5364
7	4	20143
8	5	29760
9	6	12454
10	Grand Total	101428
11		
12	Sum Operations By Week	
13		

(a) Primeiro critério

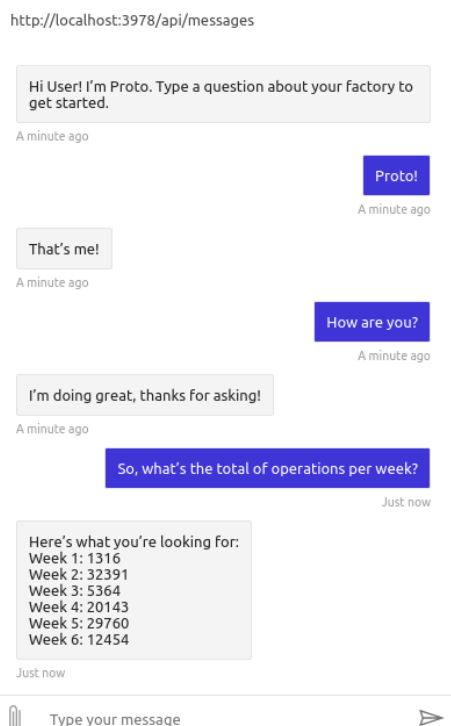
	A	B	C
1	OperationName	TrackOut	
2			
3	Row Labels	Sum of OperationCount	
4	SL040885AQ-HK0G	13230	
5	WAF04G08SS0DDR3Q-X	448	
6	Grand Total	13678	
7			
8	Count Operations By Product		
9			
10			

(b) Segundo critério

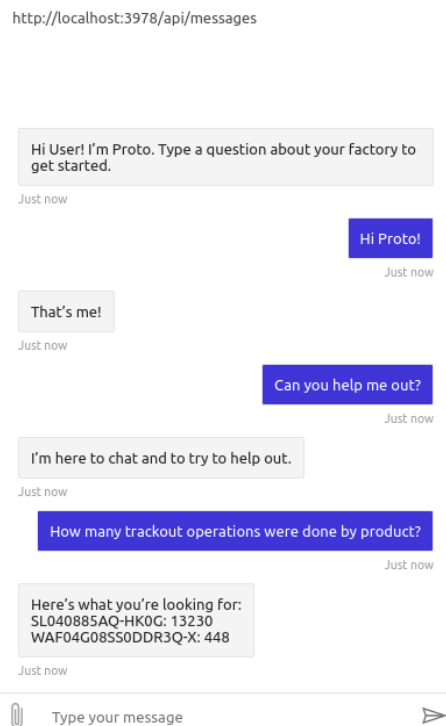
	A	B	C
1	OperationName	TrackOut	
2	FullDate	(Multiple Items)	
3			
4	Row Labels	Sum of OperationCount	
5	☐ ShiftA	558	
6	SL040885AQ-HK0G	529	
7	WAF04G08SS0DDR3Q-X	29	
8	☐ ShiftB	565	
9	SL040885AQ-HK0G	546	
10	WAF04G08SS0DDR3Q-X	19	
11	☐ ShiftC	394	
12	SL040885AQ-HK0G	373	
13	WAF04G08SS0DDR3Q-X	21	
14	Grand Total	1517	
15			
16	Count Operations By Product Per Shift		
17			

(c) Terceiro critério

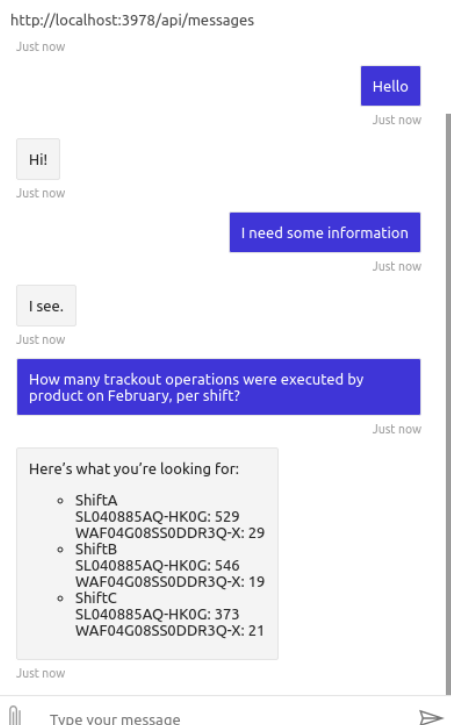
Figura 6.2: Imagens referentes aos resultados esperados pelo protótipo, dadas as diferentes perguntas



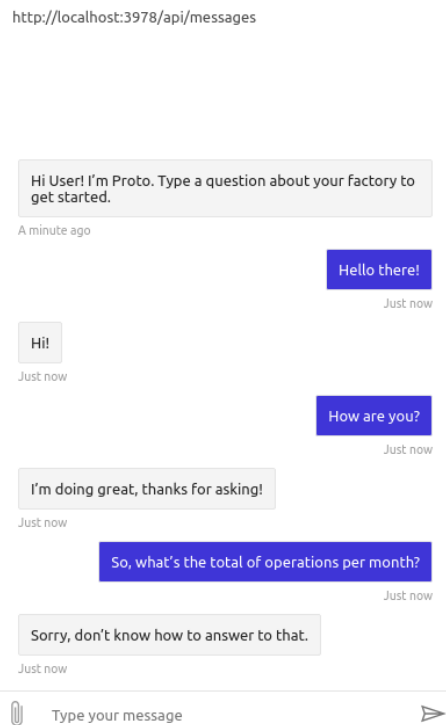
(a) Primeiro critério



(b) Segundo critério



(c) Terceiro critério



(d) Comportamento do protótipo em caso de não conter a intenção esperada

Figura 6.3: Imagens referentes às respostas dadas pelo protótipo, dadas diferentes perguntas

Capítulo 7

Conclusões

Com o presente trabalho tentou-se contribuir para o conhecimento na área das ILN, através da concepção de um módulo de linguagem natural. Assim, é conveniente salientar as conclusões alcançadas, relacioná-las com os objetivos, criticar as suas limitações e referir as perspectivas futuras. Por isso, inicia-se por avaliar em que medida os objetivos foram cumpridos. Depois, especificam-se as contribuições e de que forma se apresentam como respostas para o problema. Finalmente, são constatadas as limitações do trabalho desenvolvido e perspectivas de trabalho futuro, apontando problemas em aberto ou as alternativas a contemplar.

7.1 Avaliação de Objetivos

O trabalho descrito nesta tese consistiu na concepção de um módulo de linguagem natural, levando ao desenvolvimento de um protótipo que se demonstra capaz de compreender perguntas em formato textual, associadas ao domínio da manufatura, possibilitando a extração de informação das fontes de dados produtivos.

Na base da concepção do módulo e desenvolvimento do protótipo esteve um estudo teórico focado no PLN, mais especificamente das ILN, por forma conhecer o estado da arte desta área, nomeadamente para a compreensão de abordagens e seleção de ferramentas pertinentes para o trabalho. A abordagem adotada é fruto deste estudo, que não define nenhuma como sendo a ideal, mas sim aquela que melhor se enquadra na realidade do problema, oferecendo melhores contrapartidas. Assim, o modelo idealizado assenta sobre uma abordagem designada no âmbito desta tese por reconhecimento de intenções e entidades.

Para comprovar a validade do módulo concebido, realizaram-se testes focados na capacidade do protótipo compreender a linguagem natural de diferentes níveis e de dar resposta às questões colocadas, que visam a extração de conhecimento acerca de processos de fabrico. O protótipo respondeu adequadamente aos critérios de sucesso fixados, pelo que é legítimo concluir que os objetivos propostos foram atingidos.

7.2 Resposta ao Problema

O trabalho concretizado advém do pressuposto que o uso de linguagem natural no contexto do Critical Manufacturing MES pode melhorar a usabilidade do sistema e conseqüentemente, simplificar o processo de apoio à decisão dos seus clientes. Nesse sentido, tiveram-se como objetivos a concepção de um módulo de linguagem natural e desenvolvimento de um protótipo, aplicando o modelo idealizado.

Ainda que se demonstre que o módulo concetualizado permite a compreensão da linguagem natural e a resposta a questões-chave no processo de fabrico, nada se pode concluir acerca do seu impacto na usabilidade ou no processo de decisão, dado à natureza da solução desenvolvida, sendo ela um protótipo.

O trabalho atende à resolução do problema através da idealização de um módulo de linguagem natural, cuja aplicação prática se prova eficaz, ainda que de forma limitada. A especificação deste módulo, bem como o desenvolvimento do protótipo servem de base para a integração de ILN no contexto do Critical Manufacturing MES.

7.3 Limitações e Trabalho Futuro

Difícilmente se pode considerar um trabalho neste contexto, ou em muitos outros, como acabado. Este trabalho tem algumas limitações, que ao serem identificadas, possibilitam a introdução de novas ideias e desafios. Por conseguinte, apresentam-se algumas das limitações encontradas, sugerindo formas das ultrapassar, sempre que possível, e também algumas indicações de trabalho futuro:

- Uma das grandes limitações do protótipo é não fazer uso da capacidade de interação com o utilizador para obter *feedback* sobre as respostas dadas, usando-o para melhorar a previsão das intenções. Nesse sentido, a inclusão de um novo componente na arquitetura proposta, para a gestão do *feedback*, poderia contribuir para a robustez da fase de compreensão de linguagem natural;
- Ainda que o protótipo esteja preparado para a integração de novas fonte de dados, o processo de transformação da representação intermediária para a linguagem específica dessa fonte pode ser uma tarefa complexa e morosa, *e.g.*, a conversão de representação intermediária para SQL. Posto isto, o constituição de uma camada de abstração (“fachada”) para o acesso a essas fontes de dados, pelo uso de ferramentas como o GraphQL ou no contexto da Web Semântica, pode contribuir para o acesso mais simples a novas fontes de dados;
- Apesar do protótipo ser capaz de tratar entidades temporais, apenas o faz para os meses, ou seja, exclui datas e horas específicas ou outras variações. O Microsoft LUIS produz os metadados necessários para a transformação, pelo que é adequada a implementação de estratégias para a normalização das datas, com intuito de serem usadas no carregamento de dados;
- Para usar o protótipo é necessário o acesso à Internet, ou seja, não há mecanismo que possibilite o seu uso sem acesso à rede. Nesse sentido, uma das sugestões é a adaptação do modelo idealizado para o desenvolvimento de uma solução interna, usando as ferramentas estudadas. Assim, há a necessidade de aprofundar o conhecimento delas e em que medida podem ser aplicadas com o modelo concebido;
- A solução desenvolvida não é genérica o suficiente para se adaptar a diferentes domínios, uma vez que parte do modelo de dados deve constar na base de conhecimento. Para isto, requer-se um estudo mais aprofundado da área de IA aplicada à linguagem natural ou às ILN;
- Para dar resposta a perguntas multidimensionais, o protótipo classifica as entidades da frase com base na sua posição na frase, sendo então necessário que o modelo ML esteja treinado para esse efeito. Mas, pressupõe-se que a introdução de um “fluxo de conversação” possa simplificar este processo, na medida em que se introduz o fator “contexto”, quebrando o nível multidimensional em vários níveis unidimensionais, permitindo ao utilizador indicar o que pretende, passo a passo;
- O uso de outras línguas para além do inglês, no contexto deste protótipo, obrigariam a redefinir uma nova base de conhecimento para cada língua nova a adicionar. No entanto, e caso seja um requisito relevante, pode-se realizar um estudo que leve a concluir acerca das estratégias existentes e qual apresenta mais e melhores benefícios para uma solução final, adaptando o modelo para tal;
- O protótipo não pode ser transposto diretamente para uma solução interna, devido à sua natureza. Por isso, de acordo com a ferramenta que seja escolhida para o efeito, propõe-se que

esse processo de escolha leve também em consideração o esforço necessário para a transição, pelo que, das ferramentas estudadas, supõe-se que Rasa apresenta melhores contrapartidas.

Como se pode constatar, há ainda imenso trabalho a fazer, englobando desenvolvimento e mais pesquisa. No entanto, isso significa novos desafios e caminhos a percorrer, permitindo que este trabalho evolua, originando uma (primeira) ILN no contexto MES, o que significa mais um passo para a Indústria 4.0.

Bibliografia

- A. Woods, W, Ronald Kaplan e Bonnie Webber (1972). «The Lunar Sciences Natural Language Information System». Em:
- Abadi, Martín, Ashish Agarwal et al. (2016). «Tensorflow: Large-scale machine learning on heterogeneous distributed systems». Em: *arXiv preprint arXiv:1603.04467*.
- Abadi, Martín, Paul Barham et al. (2016). «Tensorflow: A system for large-scale machine learning». Em: *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283.
- Akula, Arjun R (2015). «A Novel Approach Towards Building a Generic, Portable and Contextual NLIDB System». Em: *International Institute of Information Technology Hyderabad*.
- Akula, Arjun, Rajeev Sangal e Radhika Mamidi (2013). «A novel approach towards incorporating context processing capabilities in nlib system». Em: *Proceedings of the sixth international joint conference on natural language processing*, pp. 1216–1222.
- Al Omran, Fouad Nasser A e Christoph Treude (2017). «Choosing an NLP library for analyzing software documentation: a systematic literature review and a series of experiments». Em: *Proceedings of the 14th International Conference on Mining Software Repositories*. IEEE Press, pp. 187–197.
- Almada-Lobo, F (2016). «The Industry 4.0 revolution and the future of manufacturing execution systems (MES)». Em: *Journal of Innovation Management* 3, pp. 16–21.
- Amazon (2019). *Amazon Lex*. <https://aws.amazon.com/lex/>. (Acedido em 11/09/2019).
- AWS Machine Learning Blog (2019). *Building a conversational business intelligence bot with Amazon Lex*. <https://aws.amazon.com/blogs/machine-learning/building-a-conversational-business-intelligence-bot-with-amazon-lex/>. (Acedido em 11/09/2019).
- Beysolow, T (2018). *Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing*. Apress. isbn: 9781484237328. url: <https://books.google.pt/books?id=pU2atgEACAAJ>.
- Bird, S, E Klein e E Loper (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media. isbn: 9780596555719. url: <https://books.google.pt/books?id=KGIbfiiP1i4C>.
- Bocklisch, Tom et al. (2017). «Rasa: Open source language understanding and dialogue management». Em: *arXiv preprint arXiv:1712.05181*.
- Britz, Denny (2019). *Understanding convolutional neural networks for NLP*. <http://www.wildml.com/2015/11/understanding-convolutional-neuralnetworks-for-nlp>. (Acedido em 10/09/2019).
- Cady, Steven et al. (2011). «Mission, vision, and values: What do they say?» Em: *Organization Development Journal* 29, pp. 63–78.
- Chowdhury, Gobinda G (2003). «Natural language processing». Em: *Annual Review of Information Science and Technology* 37.1, pp. 51–89. doi: 10.1002/aris.1440370103. url: <https://onlinelibrary.wiley.com/doi/abs/10.1002/aris.1440370103>.
- Critical Manufacturing (2019a). *Business Intelligence*. <https://www.criticalmanufacturing.com/en/services/business-intelligence>. (Acedido em 30/09/2019).
- (2019b). *Electronics Industry*. <https://www.criticalmanufacturing.com/en/industries/electronics>. (Acedido em 05/02/2019).
- (2019c). *Infrastructure Framework for Manufacturing Equipment Integration, Data Analysis and Business Intelligence*. <https://www.criticalmanufacturing.com/en/critical-manufacturing-mes/architecture>. (Acedido em 05/02/2019).

- Critical Manufacturing (2019d). *Integrated Manufacturing Execution System*.
<https://www.criticalmanufacturing.com/en/critical-manufacturing-mes/overview>.
(Acedido em 05/02/2019).
- (2019e). *Medical Device Industry*. <https://www.criticalmanufacturing.com/en/industries/medical-device>. (Acedido em 05/02/2019).
- (2019f). *Overview*. <https://www.criticalmanufacturing.com/en/company/overview>. (Acedido em 01/02/2019).
- (2019g). *Semiconductor Industry*. <https://www.criticalmanufacturing.com/en/industries/semiconductor>. (Acedido em 05/02/2019).
- (2019h). *Strategy*. <https://www.criticalmanufacturing.com/en/company/strategy>. (Acedido em 01/02/2019).
- (2019i). *What is MES?* <https://www.criticalmanufacturing.com/en/critical-manufacturing-mes/what-is-manufacturing-execution-system>. (Acedido em 03/02/2019).
- Dennis, JackB (2011). «Data Flow Graphs». Em: *Encyclopedia of Parallel Computing*. Ed. por David Padua. Boston, MA: Springer US, pp. 512–518. isbn: 978-0-387-09766-4. doi: 10.1007/978-0-387-09766-4_294. url: https://doi.org/10.1007/978-0-387-09766-4_294.
- Gorecky, Dominic et al. (2014). «Human-machine-interaction in the industry 4.0 era». Em: *2014 12th IEEE international conference on industrial informatics (INDIN)*. Ieee, pp. 289–294.
- Green, Spence, Jeffrey Heer e Christopher D. Manning (2015). «Natural Language Translation at the Intersection of AI and HCI». Em: *Commun. ACM* 58.9, pp. 46–53. issn: 0001-0782. doi: 10.1145/2767151. url: <http://doi.acm.org/10.1145/2767151>.
- Hendrix, Gary G et al. (1978). «Developing a natural language interface to complex data». Em: *ACM Transactions on Database Systems (TODS)* 3.2, pp. 105–147.
- Hertz, John A (2018). *Introduction to the theory of neural computation*. CRC Press.
- IBM (2019a). *About Watson*. <https://www.ibm.com/watson/about>. (Acedido em 11/09/2019).
- (2019b). *IBM Watson Assistant Overview*. <https://www.ibm.com/cloud/watson-assistant/>. (Acedido em 11/09/2019).
- IBM Blog (2019). *Database-driven chatbot tutorial adapted to latest IBM Watson Assistant features*. <https://www.ibm.com/cloud/blog/updated-tutorial-database-driven-chatbot>. (Acedido em 11/09/2019).
- Jurafsky, Dan e James H Martin (2014). *Speech and language processing*. Vol. 3. Pearson London.
- Kassner, Laura et al. (2017). «The social factory: connecting people, machines and data in manufacturing for context-aware exception escalation». Em: *Proceedings of the 50th Hawaii International Conference on System Sciences*.
- Koen, Peter A (2004). «The fuzzy front end for incremental, platform and breakthrough products and services». Em: *PDMA Handbook*, pp. 81–91.
- Koen, Peter A et al. (2002). «Fuzzy front end: effective methods, tools, and techniques». Em: *The PDMA toolbook 1 for new product development*.
- Koen, Peter et al. (2001). «Providing clarity and a common language to the “fuzzy front end”». Em: *Research-Technology Management* 44.2, pp. 46–55.
- Lasi, Heiner et al. (2014). «Industrie 4.0». Em: *WIRTSCHAFTSINFORMATIK* 56.4, pp. 261–264. issn: 1861-8936. doi: 10.1007/s11576-014-0424-4. url: <https://doi.org/10.1007/s11576-014-0424-4>.
- Li, Yunyao, Huahai Yang e HV Jagadish (2005). «NaLIX: an interactive natural language interface for querying XML». Em: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, pp. 900–902.
- Liu, Dapeng, Yan Li e Manoj A Thomas (2017). «A roadmap for natural language processing research in information systems». Em: *Proceedings of the 50th Hawaii International Conference on System Sciences*.
- Lobur, Mykhailo, AnDriy Romanyuk e Mariana Romanyshyn (2011). «Using NLTK for educational and scientific purposes». Em: *2011 11th international conference the experience of designing and application of CAD systems in microelectronics (CADSM)*. IEEE, pp. 426–428.
- McClellan, Michael (2001). «Introduction to manufacturing execution systems». Em: *MES Conference & Exposition, Baltimore, Maryland*, pp. 1–7.

- MESA, International (1997). «Mes Explained: A High Level VSION». Em: *MESA International White Paper* 1, p. 25.
- Microsoft (2019a). *Azure Bot Service Documentation*. <https://docs.microsoft.com/en-us/azure/bot-service/>. (Acedido em 03/10/2019).
- (2019b). *Use Cognitive Services with natural language processing (NLP) to enrich bot conversations*. <https://docs.microsoft.com/en-us/azure/cognitive-services/luis/choose-natural-language-processing-service>. (Acedido em 11/09/2019).
- (2019c). *What is Language Understanding (LUIS)?* <https://docs.microsoft.com/en-us/azure/cognitive-services/luis/what-is-luis>. (Acedido em 11/09/2019).
- Mony, Manju, Jyothi M Rao e Manish M Potey (2014). «An Overview of NLIDB Approaches and Implementation for Airline Reservation System». Em: *International Journal of Computer Applications* 107.5.
- El-Mouadib, Faraj A et al. (2009). «Generic interactive natural language interface to databases (ginlibb)». Em: *International journal of computers* 3.3, pp. 301–310.
- Nihalani, Neelu, Sanjay Silakari e Mahesh Motwani (2011). «Natural language interface for database: a brief review». Em: *International Journal of Computer Science Issues (IJCSI)* 8.2, p. 600.
- Patel, Jaina e Jay Dave (2015). «A Survey: Natural Language Interface to Databases». Em: *International Journal of Advance Engineering and Research Development (IJAERD)*.
- Perkins, Jacob (2010). *Python text processing with NLTK 2.0 cookbook*. Packt Publishing Ltd.
- Pinto, Sara Catarina Silva (2015). «Processamento de linguagem natural e extração de conhecimento». Tese de mestrado.
- Plesco, Craig e Nestor Rychtyckyj (2012). «Machine Translation as a Global Enterprise Service at Ford». Em: *AMTA 2012, Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas*.
- Popescu, Ana-Maria, Alex Armanasu et al. (2004). «Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability». Em: *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, p. 141.
- Popescu, Ana-Maria, Oren Etzioni e Henry Kautz (2003). «Towards a Theory of Natural Language Interfaces to Databases». Em: *Proceedings of the 8th International Conference on Intelligent User Interfaces*. IUI '03. Miami, Florida, USA: ACM, pp. 149–157. isbn: 1-58113-586-6. doi: 10.1145/604045.604070. url: <http://doi.acm.org/10.1145/604045.604070>.
- Ramesh, Sree Harsha, Jayant Jain, Krishna R Sundaresan et al. (2016). «Towards Building A Domain Agnostic Natural Language Interface to Real-World Relational Databases». Em: *Proceedings of the 13th International Conference on Natural Language Processing*, pp. 305–314.
- Rasa (2019). *Rasa NLU Documentation*. <https://rasa.com/docs/rasa/>. (Acedido em 11/09/2019).
- Reid, Susan E e Ulrike De Brentani (2004). «The fuzzy front end of new product development for discontinuous innovations: A theoretical model». Em: *Journal of product innovation management* 21.3, pp. 170–184.
- Resnik, Philip (1989). *Access to multiple underlying systems in JANUS*. Rel. téc. BBN SYSTEMS e TECHNOLOGIES CORP CAMBRIDGE MA.
- Roblek, Vasja, Maja Meško e Alojz Krapež (2016). «A Complex View of Industry 4.0». Em: *SAGE Open* 6.2. doi: 10.1177/2158244016653987. url: <https://doi.org/10.1177/2158244016653987>.
- Romero, David et al. (2016). «Towards an operator 4.0 typology: a human-centric perspective on the fourth industrial revolution technologies». Em: *International conference on computers and industrial engineering (CIE46) proceedings*.
- Rychtyckyj, N. (2007). «Intelligent Systems for Manufacturing at Ford Motor Company». Em: *IEEE Intelligent Systems* 22.1, pp. 16–19. doi: 10.1109/MIS.2007.13.
- spaCy (2019). *spaCy - Usage*. <https://spacy.io/usage>. (Acedido em 10/09/2019).
- Stanford OpenNLP (2019). *Stanford OpenNLP*. <https://stanfordnlp.github.io/CoreNLP/index.html>. (Acedido em 10/09/2019).
- Ugarte, B Saenz de, A Artiba e R Pellerin (2009). «Manufacturing execution system – a literature review». Em: *Production Planning & Control* 20.6, pp. 525–539. doi: 10.1080/09537280902938613. url: <https://doi.org/10.1080/09537280902938613>.

- Utama, Prasetya et al. (2018). «An End-to-end Neural Natural Language Interface for Databases». Em: *arXiv preprint arXiv:1804.00401*.
- W3C (2019). *Resource Description Framework (RDF)*. <https://www.w3.org/RDF/>. (Acedido em 20/09/2019).
- Warren, David HD e Fernando CN Pereira (1982). «An efficient easily adaptable system for interpreting natural language queries». Em: *Computational Linguistics* 8.3-4, pp. 110–122.
- Williams, Jason D et al. (2015). «Fast and easy language understanding for dialog systems with Microsoft Language Understanding Intelligent Service (LUIS)». Em: *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 159–161.
- Zhong, Ray Y et al. (2017). «Intelligent Manufacturing in the Context of Industry 4.0: A Review». Em: *Engineering* 3.5, pp. 616–630. issn: 2095-8099. doi: <https://doi.org/10.1016/J.ENG.2017.05.015>. url: <http://www.sciencedirect.com/science/article/pii/S2095809917307130>.

Apêndice A

Plano de Trabalho Previsto

Neste apêndice apresenta-se o plano de trabalho definido inicialmente para o projeto.

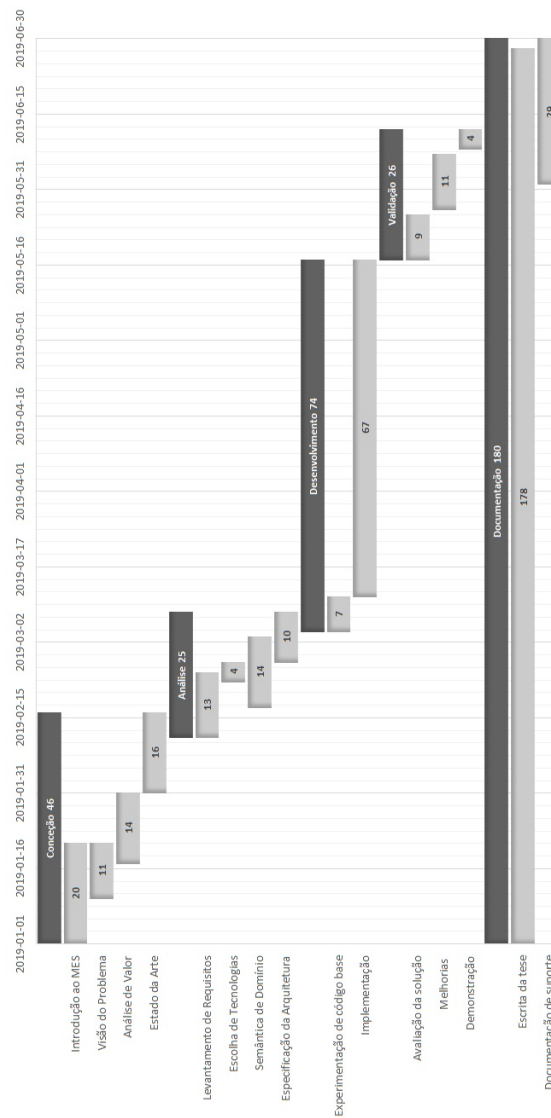


Figura A.1: Diagrama de *Gantt* referente ao planeamento do projeto

Apêndice B

Análise de Valor

Esta capítulo visa a análise da oportunidade de negócio que surge com a nova funcionalidade, cuja abordagem é explorada neste trabalho.

B.1 O Processo de Inovação

De acordo com P. A. Koen et al. (2002), o processo de inovação, representado na Figura B.1, está dividido em três áreas – o *Fuzzy Front End* (FFE), *New Product Development* (NPD) e a comercialização – que correspondem às fases inerentes ao *New Concept Development* (NCD), um modelo desenvolvido por um conjunto de empresas, com o objetivo de “[...] fornecer uma linguagem e compreensão comum para as atividades *front end*”¹ (P. Koen et al. 2001).

O FFE representa uma oportunidade para melhoria de todo o processo de inovação, focando todas as atividades que antecedem o desenvolvimento do produto, com o propósito de potenciar o valor, a importância e a probabilidade de sucesso das fases que se seguem. Ou seja, consiste no investimento do tempo em atividades de discussão da ideia, por forma a identificar e estruturar o problema ou oportunidade (P. A. Koen et al. 2002; Reid e De Brentani 2004). Porém, as atividades inerentes ao FFE são fundamentalmente diferentes da fase NPD, pelo que se torna necessária a definição de vocabulário específico, permitindo a geração de conhecimento e clara distinção entre as diferentes fases do processo (P. A. Koen et al. 2002).

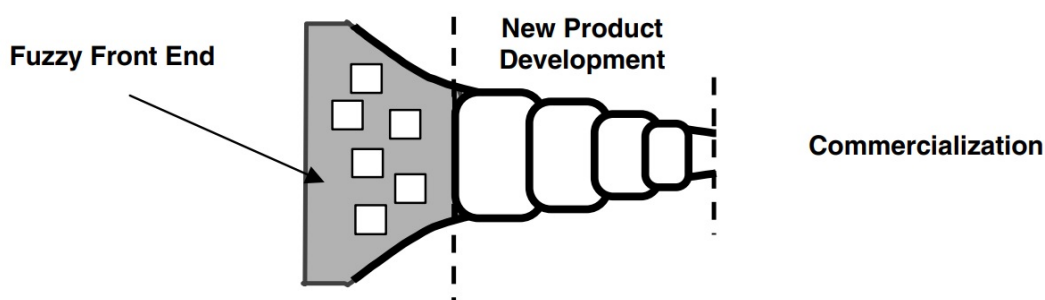


Figura B.1: O processo de inovação, extraído de P. A. Koen et al. (2002)

O modelo NCD, demonstrado na Figura B.2, baseado num modelo relacional ao invés de um processo linear, visa providenciar uma terminologia para o FFE (P. A. Koen et al. 2002). A área interna define os cinco elementos chave do *Front End of Innovation*: a identificação de oportunidade (*Opportunity Identification*), a análise de oportunidade (*Opportunity Analysis*), a geração e enriquecimento de ideias (*Idea Generation and Enrichment*), a seleção de ideias (*Idea Selection*) e a definição do conceito (*Concept Definition*). O motor central (*Engine*) corresponde à liderança, cultura e estratégia organizacional, que suporta os elementos que compõem o FFE, são controláveis pela organização e

¹Tradução livre de autor. No original “[...] to provide a common language and insights on the front end activities.”.

possibilita a interação entre eles. Já na periferia, encontram-se os fatores de influência (*Influencing Factors*), geralmente incontrolláveis pela organização, consistem nas capacidades organizacionais, na estratégia de negócio, no mundo exterior, nomeadamente os canais de distribuição, clientes, fornecedores, concorrentes, política governamental ou legislação, ou quaisquer fatores que possam influenciar todo o processo de inovação (P. A. Koen et al. 2002; P. Koen et al. 2001).

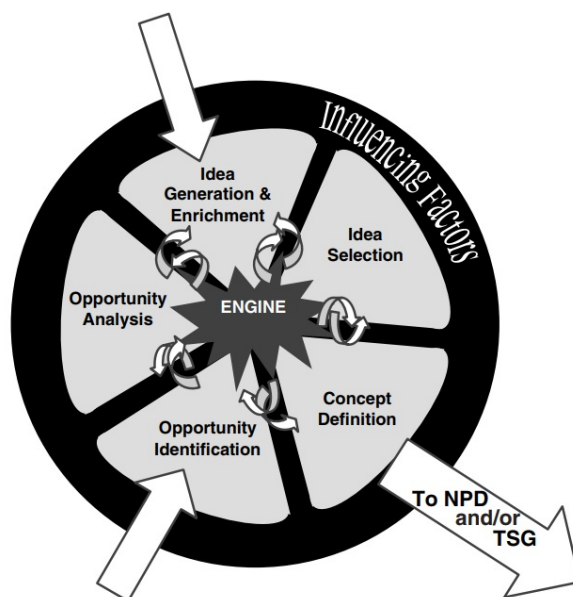


Figura B.2: A representação do modelo *New Concept Development* (NCD), extraído de P. A. Koen et al. (2002)

Quanto à representação do modelo, as partes internas são designadas de elementos por oposição a processos, pois estes implicam estrutura, que pode não ser possível ser aplicada. O formato circular indica que é esperado que as ideias fluam, circulem e itelem ao longo dos elementos, por qualquer ordem ou combinação, permitindo o uso dos elementos, repetidamente. Este comportamento é intrínseco às atividades do FFE, permitindo uma definição clara do mercado, dos requisitos, dos riscos associados e do plano de negócio, tornando mais eficazes as fases de desenvolvimento e comercialização, devido à redução do tempo total de projeto, fruto da diminuição da repetição de algumas atividades (P. A. Koen et al. 2002).

B.2 O Fuzzy Front End de Inovação

Como mencionado anteriormente, o FFE corresponde a um conjunto de atividades geralmente caóticas, imprevisíveis e não estruturadas que antecedem o desenvolvimento de um produto (P. A. Koen 2004). Todavia, é preciso perceber a natureza do produto a desenvolver, de forma a melhor enquadrar o processo de inovação.

Segundo P. A. Koen (2004), pode-se caracterizar os produtos de acordo com a extensão da mudança ou do processo: incremental, requer pouca mudança a nível do produto ou do processo, uma vez que geralmente consiste na redução de custos, melhoria, extensão ou reposicionamento no mercado de produtos já existentes; plataforma, estabelecem uma arquitetura básica para uma nova geração de produtos ou processos; pioneiro, envolve uma mudança significativa no processo ou produto.

O presente trabalho visa o desenvolvimento dum módulo de linguagem natural para o Critical Manufacturing MES, uma plataforma já estabelecida, ou seja, trata-se de uma extensão ao produto já existente, enquadrando-se no tipo incremental. A ideia surge do processo de planeamento estratégico da empresa com a finalidade de trazer novas funcionalidades aos seus clientes, melhorando

a qualidade do produto. Portanto, nas secções seguintes, aplica-se a metodologia explicitada, no sentido de enriquecer a proposta de projeto apresentada pela Critical Manufacturing.

B.2.1 Identificação da Oportunidade

O PLN é uma área de investigação que explora a forma como os computadores podem manipular a linguagem natural (texto ou voz) para executar determinadas tarefas. Aplica-se em diversos campos de estudo: tradução, processamento de texto, interfaces com o utilizador, reconhecimento de voz, sistemas periciais (Chowdhury 2003).

Utama et al. (2018) menciona que, apesar da expressividade da SQL, os utilizadores necessitam de algum conhecimento técnico para perceber como extrair informação de um sistema, o que conduziu à investigação para o desenvolvimento de interfaces alternativas que permitam aos utilizadores, sem conhecimento técnico, explorar e interagir com os dados, de forma conveniente. Também Popescu, Etzioni e Kautz (2003) menciona que a necessidade de interfaces de linguagem natural se torna mais evidente, devido ao número de pessoas sem conhecimentos técnicos que acedem a informação através de *browsers* ou telemóveis, tornando paradigmas como o reconhecimento de voz mais atrativos.

Nesse sentido, a Critical Manufacturing tenciona o desenvolvimento do módulo de linguagem natural para que os utilizadores do produto, sem conhecimento orientado às tecnologias de informação, possam fácil, rápida e intuitivamente consultar o sistema. Desta forma, a funcionalidade destaca o produto pelo uso de novas tecnologias, facilita-se a interação com o sistema, reduzindo-se o tempo de formação técnica associado ao mesmo.

B.2.2 Análise da Oportunidade

A pesquisa realizada por Liu, Yan Li e Thomas (2017), apresentada na Figura B.3, cuja metodologia consistiu na pesquisa de termos como “Natural Language Processing” e “NLP” em bases de dados académicas, determina que há uma tendência crescente de interesse por esta área. Nos últimos anos, a quantidade de dados textuais disponíveis nas redes sociais ou em sistemas de comunicação, juntamente com a necessidade de acesso a informação, contribuíram para o avanço e adoção comercial do PLN (Liu, Yan Li e Thomas 2017).

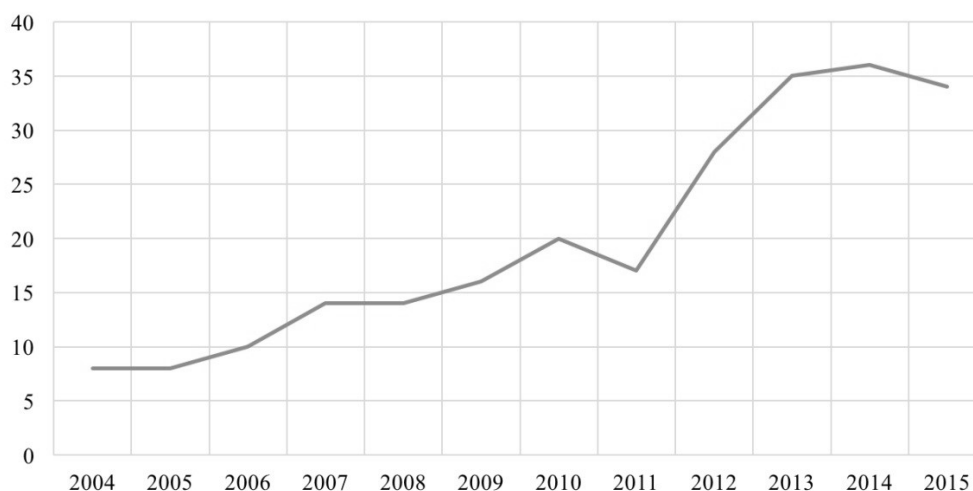


Figura B.3: Número de artigos de Processamento de Linguagem Natural (PLN) pesquisados por ano, extraído de Liu, Yan Li e Thomas (2017)

Quanto ao segmento de mercado no qual se integra, cresce a visão de fábricas inteligentes, associadas à quarta revolução industrial, prezando a integração do operador humano num ambiente complexo e rico em dados (Kassner et al. 2017). Almada-Lobo (2016) afirma que a revolução supracitada é já conhecida pelas empresas, o que lhe permite tomar ações no sentido de definir o seu modelo de fabrico e o seu plano de transformação, particularmente na adaptação do MES de forma a manter o desempenho, qualidade e agilidade nas desafios espoletados pelas empresas de manufatura. Portanto, a interação entre o ser humano e o sistema pode melhorar o processo de fabrico e potenciar o negócio, na medida em que o operador, em vez do trabalho manual repetitivo que pode facilmente ser automatizado, passa a tomar decisões no processo para resolução de problemas, as quais requerem acesso à informação correta e de forma atempada (Kassner et al. 2017). É nesse sentido que o Critical Manufacturing MES ganha vantagem com o desenvolvimento desta nova funcionalidade.

B.2.3 Geração, Enriquecimento e Seleção de Ideias

No seguimento deste assunto, foram realizadas duas reuniões com o supervisor do projeto na Critical Manufacturing, em que foram discutidos alguns requisitos operacionais e de usabilidade, restrições ao desenvolvimento da solução, como a preferência por uso de ferramentas de PLN que possam ser mantidas internamente e a sua facilidade de utilização, e ideias para futuras implementações, as quais podem ter um impacto na especificação arquitetural do protótipo.

Em relação às ideias e conceitos contempladas no *mindmap* da Figura B.4, o presente projeto pretende dar resposta a praticamente todos, tendo em consideração que, numa fase inicial, o cumprimento de todos é praticamente inatingível. A descrição de cada conceito é feito de seguida:

- *Tecnologia* – a ideia inerente ao trabalho assenta sobre as temáticas de PLN, especificamente Interfaces de Linguagem Natural, e *Data Warehouses*. Esta consiste no estudo aprofundado deste tipo de interfaces orientado à consulta em armazéns de dados e disseminação do conhecimento internamente, para que no futuro, o projeto possa ter continuidade;
- *Estado da Arte* – abordagem de ferramentas e soluções análogas, com o objetivo de especificar uma arquitetura para o sistema. Este processo dá origem aos documentos de especificação que devem ser usados para consulta por parte dos desenvolvedores, quer numa perspetiva de conhecimento arquitetural, quer das ferramentas que são usadas;
- *Clientes* – uma vez que a Critical Manufacturing possui clientes com diferentes realidades, a ideia é que o módulo final esteja preparado para elaborar consultas em qualquer domínio, de forma configurada ou cerne da solução. Contudo, como já abordado anteriormente, no contexto deste trabalho, apenas um domínio será considerado;
- *Utilizadores* – a solução deverá responder às necessidades de qualquer utilizador, desde os mais técnicos (Engenheiros de Produção) aos menos técnicos (Operadores). Porém, o protótipo terá em consideração os utilizadores mais comuns do Critical Manufacturing MES;
- *Restrições* – nesta temática, foram discutidas alternativas como a avaliação do custo de uma ferramenta proprietária, uso de ferramentas *open source* ou o desenvolvimento interno da própria biblioteca de PLN, de modo a garantir que não existem dependências externas à plataforma. Também foram discutidas a eficiência da solução em contexto produtivo, a facilidade de aprendizagem e usabilidade da mesma. Assim, a usabilidade do módulo de linguagem natural será estudada a partir do mecanismo de *feedback* provido na solução e através de inquéritos aos utilizadores, o que também se aplica para a aprendizagem da ferramenta. Relativamente aos restantes tópicos, não houve conclusão acerca das ideias a serem selecionadas;
- *Requisitos* – pressupõe-se o uso de auto-aprendizagem para adaptação automática do módulo ao *feedback* do utilizador, ainda que para o protótipo, a resposta a uma simples pergunta como “A resposta obtida foi-lhe útil?” é suficiente. Também a integração com o produto, quer a nível aplicacional, quer a nível de processo deve ser considerada, o que resultará na organização

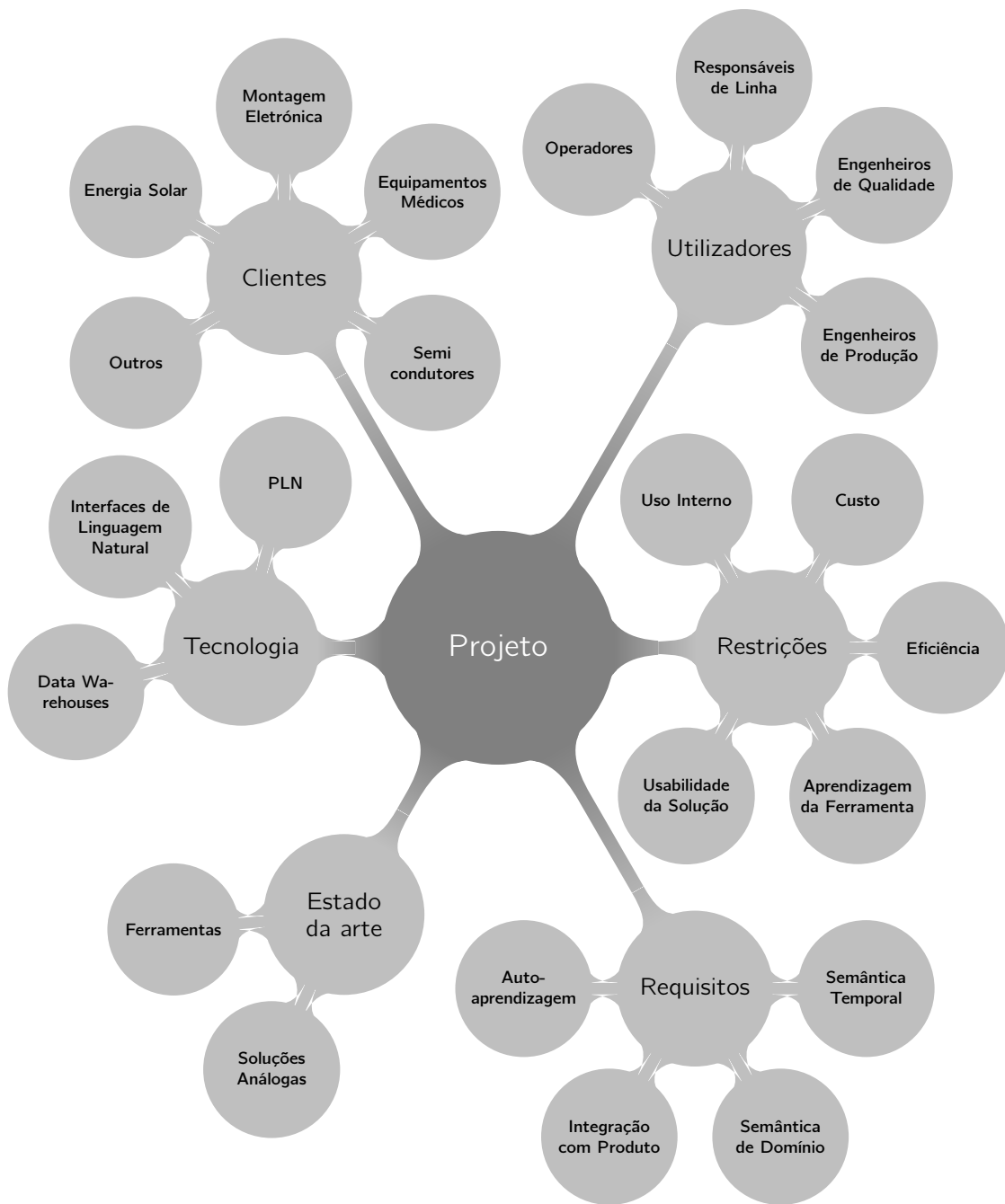


Figura B.4: Mindmap das ideias e conceitos gerados

de *meetups* com as equipas responsáveis pelo processo de manutenção da plataforma. Quanto aos restantes conceitos, não houve conclusão acerca das ideias selecionadas.

B.2.4 Definição do Conceito

Todo o processo presente no modelo NCD de FFE culmina com a definição do conceito, a fase que encaminha o projeto para a implementação (P. A. Koen et al. 2002).

O presente trabalho, denominado de “Natural Language Querying” consiste na conceção de um módulo de linguagem natural para interface com o Critical Manufacturing MES e desenvolvimento de um protótipo que servirá de base para o módulo final para interface com o Critical Manufacturing MES. Esse módulo final permitirá a consulta e pesquisa de estados do processo de fabrico por utilizadores com pouco ou nenhum conhecimento associado a tecnologias de informação, garantindo a interação com o sistema de uma forma simples, fácil e intuitiva, melhorando o processo numa perspetiva de apoio à decisão (ver Secção 1.1). Os objetivos deste projeto estão descritos na Secção 1.2, a metodologia e critérios de sucesso na Secção 1.4, e o respetivo plano de trabalho na Secção 1.6.

O projeto traz benefícios para a empresa e o seu produto, pelo estudo e adoção de tecnologia de PLN num contexto industrial, pela possível melhoria de usabilidade do sistema e pela evolução no processo de apoio à decisão dos seus clientes. Uma vez que o Critical Manufacturing MES é um produto bem posicionado no mercado, não se esperam riscos a nível comercial. Contudo, a uso de tecnologia recente, cujos conceitos não estão totalmente estudados e cujos trabalhos de investigação são limitados, pode provocar atrasos no desenvolvimento do projeto ou incumprimento do orçamento definido. Não obstante, o projeto avança com o desenvolvimento de um protótipo, fase que decidirá a inclusão do módulo na plataforma da Critical Manufacturing.

Apêndice C

Protótipo

Neste apêndice são mostrados alguns artefactos recolhidos ao longo das fases de conceção, desenvolvimento e validação do protótipo.

C.1 Configuração

Nesta secção apresentam-se algumas imagens referentes ao processo de configuração levado no protótipo.

Entities



+ Create new entity + Add prebuilt entity + Add prebuilt domain entity			Search entities ... 
<input type="checkbox"/> Name ^	Type	Labeled Utterances	
datetimeV2	Prebuilt	N/A	
Dimension	List	14	
DimensionValue	Simple	18	
number	Prebuilt	N/A	
Operator	List	N/A	

Figura C.1: Definição das entidades esperadas

AverageOperationsOnStep 

Labelled entities: DimensionValue, Dimension

<input type="checkbox"/> Example utterance	Score ?
Enter an example of what a user might say and hit Enter.	
average Dimension:Projection of DimensionValue:Operation operations on DimensionValue:Step step on datetimeV2	0.95
what 's the average Dimension:Projection of DimensionValue:Operation operations on DimensionValue:Step step on datetimeV2	0.97
avg Dimension:Projection of DimensionValue:Operation operations on DimensionValue:Step step on datetimeV2	0.95

(a) Intenção *AverageOperationOnStep*CountMaterialsOnCondition 

Labelled entities: Dimension

<input type="checkbox"/> Example utterance	Score ?
Enter an example of what a user might say and hit Enter.	
number of materials with Dimension:Filter Operator number	0.92
how many materials with Dimension:Filter Operator number	0.93
materials with Dimension:Filter Operator number	0.94

(b) Intenção *CountMaterialsOnCondition*CountMaterialsOnConditionOnStep 

Labelled entities: DimensionValue, Dimension

<input type="checkbox"/> Example utterance	Score ?
Enter an example of what a user might say and hit Enter.	
how many materials on DimensionValue:Step step have Dimension:Filter Operator number	0.84
materials on DimensionValue:Step step have Dimension:Filter Operator number	0.77
number of materials on DimensionValue:Step step have Dimension:Filter Operator number	0.86


(c) Intenção *CountMaterialsOnConditionOnStep*

Figura C.2: Intenções definidas, contendo as expressões e respectivas entidades

CountMaterialsOnConditionOnStepGrouped 

Labelled entities: Dimension, DimensionValue

<input type="checkbox"/> Example utterance	Score ?
Enter an example of what a user might say and hit Enter.	
number of materials by Dimension:Grouping on DimensionValue:Step step have Dimension:Filter Operator number	0.91
how many materials by Dimension:Grouping on DimensionValue:Step step have Dimension:Filter Operator number	0.81
materials by Dimension:Grouping on DimensionValue:Step step have Dimension:Filter Operator number	0.86

(a) Intenção *CountMaterialsOnConditionOnStepGrouped*CountOperationsByProduct 

Labelled entities: DimensionValue

<input type="checkbox"/> Example utterance	Score ?
Enter an example of what a user might say and hit Enter.	
how many DimensionValue:Operation operations by product on datetimeV2	0.88
number of DimensionValue:Operation operations by product on datetimeV2	0.88
DimensionValue:Operation operations by product on datetimeV2	0.91

(b) Intenção *CountOperationsByProduct*SumOperationsByWeek 

Labelled entities: None

<input type="checkbox"/> Example utterance	Score ?
Enter an example of what a user might say and hit Enter.	
total of operations per week on datetimeV2	0.94
total operations per week on datetimeV2	0.94
what ' s the total operations per week on datetimeV2	0.93

(c) Intenção *SumOperationsByWeek*

Figura C.3: Continuação das intenções definidas, contendo as expressões e respectivas entidades

C.2 Validação

Nesta secção apresentam-se algumas imagens referentes ao processo de validação do protótipo.

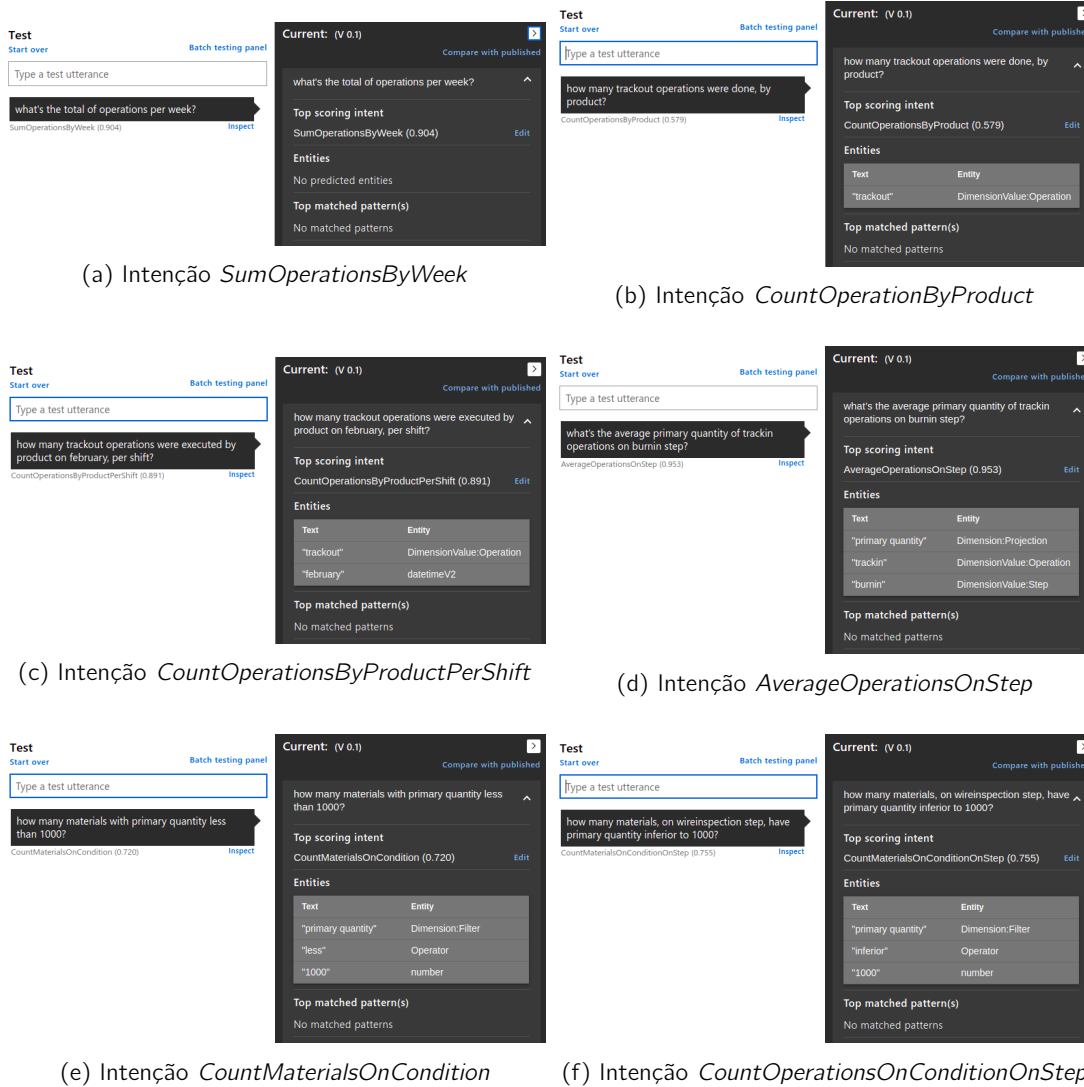


Figura C.4: Outras imagens relativas à avaliação de intenções e entidades do protótipo