



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO ENGENHARIA DE AUTOMAÇÃO E
SISTEMAS

Brunno Abner Machado

**Uma arquitetura de *softbot* inteligente como serviço para análise automática
de dados em gerenciamento remoto de produção**

Florianópolis

2021

Brunno Abner Machado

**Uma arquitetura de *softbot* inteligente como serviço para análise automática
de dados em gerenciamento remoto de produção**

Dissertação submetida ao Programa de Engenharia
de Automação e Sistemas da Universidade Federal
de Santa Catarina para a obtenção do título de
Mestre em Engenharia de Automação e Sistemas
Orientador: Prof. Ricardo José Rabelo, Dr.
Coorientador: Prof. Saulo Popov Zambiasi, Dr.

Florianópolis

2021

Ficha de identificação da obra elaborada pelo autor, através do Programa de
Geração Automática da Biblioteca Universitária da UFSC.

Machado, Brunno Abner

Uma arquitetura de softbot inteligente como serviço para análise automática de dados em gerenciamento remoto de produção / Brunno Abner Machado ; orientador, Ricardo José Rabelo, coorientador, Saulo Popov Zambiasi, 2021.

158 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Florianópolis, 2021.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Softbot. 3. Gerenciamento de Produção. 4. Modelos de Maturidade. 5. Agentes Inteligentes. I. Rabelo, Ricardo José. II. Zambiasi, Saulo Popov. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. IV. Título.

Brunno Abner Machado

Uma arquitetura de *softbot* inteligente como serviço para análise automática de dados em gerenciamento remoto de produção

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Jomi Fred Hübner, Dr.
Instituição PPGEAS/UFSC

Prof. João Carlos Espindola Ferreira, Dr.
Instituição POSMEC/UFSC

Prof. Jonny Carlos da Silva, Dr.
Instituição POSMEC/UFSC

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Engenharia de Automação e Sistemas.

Prof. Werner Kraus Junior, Dr.

Coordenação do Programa de Pós-Graduação em Engenharia de Automação e Sistemas

Prof. Ricardo José Rabelo, Dr.
Orientador

Florianópolis, 2021.

Dedico este trabalho ao meu pai, à minha mãe e à minha esposa.

AGRADECIMENTOS

Aos meus pais Paulo Roberto Machado e Solange Regina Machado, e a minha irmã Agnes Cristina Machado, por serem sempre incentivadores e motivadores em todas as etapas da minha vida.

Ao meu orientador Prof. Dr. Ricardo José Rabelo, pela ajuda e suporte ao longo de todo o trabalho realizado.

Ao meu Coorientador Prof. Dr. Saulo Popov Zambiasi, pela prontidão e rapidez em todas as minhas dúvidas.

Ao Sr. Paulo Narciso Filho e ao Prof. Dr. Augusto Humberto Bruciapaglia pelo incentivo e confiança na minha candidatura ao mestrado.

À empresa Harbor pela parceria e disponibilidade da realização deste trabalho.

Aos meus colegas de trabalho, que sempre foram amigos, dispostos a ajudar e prestativos em qualquer situação.

Aos professores e ao pessoal administrativo do PPGEAS, especialmente ao senhor Enio Snoeijer e ao coordenador Prof. Dr. Werner Kraus Junior pelas suas dedicações e disposições para ajudar a todos os alunos.

Por fim, agradeço à minha esposa Luciana Evangelista Rosa Moreira pelo amor, compreensão, companheirismo, apoio e paciência durante todo o mestrado e na vida.

“Os dias prósperos não vêm por acaso;
nascem de muita fadiga e
persistência.”

(Henry Ford)

RESUMO

O gerenciamento de produção envolve muitas atividades. Para atender os requisitos da indústria 4.0, muitos sistemas foram desenvolvidos para coletar informações do chão de fábrica a fim de melhorar as tomadas de decisão. Estudos empíricos mostram que isso gera uma sobrecarga de informações para serem lidas no gerenciamento de produção, causando estresse, análises incorretas e algumas vezes a decisões baseadas em palpites, principalmente em PMEs. Usando análise de negócios e modelos de maturidade, este trabalho apresenta Livia, um softbot com capacidades de conversação baseado em orientação a serviço que auxilia o chamado Gerente de produção como serviço. Disponibilizado na nuvem e trabalhando com dados do chão de fábrica através de integração com o MES, a Livia ajuda o gerente a identificar os seus problemas, calcular a maturidade da empresa baseada em um modelo de maturidade, sugerir ações corretivas, além de oferecer suporte proativamente. A sua arquitetura é modular permitindo que as técnicas de IA sejam trocadas facilmente. Este trabalho foi desenvolvido em parceria com o provedor MES. Os resultados obtidos atendem os objetivos propostos e apresentados nas conclusões.

Palavras-chave: MES, Softbot, Chatbot, Assistente Virtual, Gerenciamento de Produção, Software como serviço, Modelo de Maturidade, Análise de Negócio, Análise Prescritiva, Inteligência Artificial.

ABSTRACT

Production management involves many activities. In order to deal with Industry 4.0 requirements, many systems have developed solutions to gather real-time information from the shopfloor for more reliable decision-making. Empirical studies have been showing that this has created an overload of information to be handled by managers, causing stress, incorrect analyses and sometimes guessing-based decision-making, especially in SMEs. Using business analytics and maturity models concepts, this work shows Livia, a service-oriented-based softbot with chatting capabilities that implements what we are naming as production management-as-a-service. Deployed in a cloud and working on companies' shopfloor information got via a MES system, Livia helps managers identify their main problems, calculate companies' maturity model based on them, suggest corrective actions, besides proactively doing many supporting actions. Its architecture is modular allowing AI techniques to be easily exchanged. This work has been implemented close to a MES provider as a module of its system. Results are presented and discussed in the end.

Keywords: Production Management, Softbot, Maturity Model, Business Analytics, MES, Manufacturing Execution System, Software-as-a-Service, SaaS.

LISTA DE FIGURAS

Figura 1 - Quatro fases básicas do ciclo da Pesquisa-ação	30
Figura 2 – Tipos de Análise de Negócio	39
Figura 3 – Níveis de Maturidade CMMI	40
Figura 4 - Blocos da Arquitetura Orientada a Serviços	43
Figura 5 – Cenário atual	59
Figura 6 – Cenário pretendido	60
Figura 7 – Representação Macro	64
Figura 8 - Arquitetura Base	66
Figura 9 - Arquitetura Proposta	68
Figura 10 - Diagrama de Arquitetura	69
Figura 11 – Diagrama de Sequência Simplificado	71
Figura 12 – Diagrama de Sequência Simplificado Completo	72
Figura 13 – Exemplo de Dataframe	74
Figura 14 - Modelo caixa preta	75
Figura 15 – Níveis de Maturidade do Modelo Proposto	80
Figura 16 – Diagrama de Casos de Uso Simplificado Reativo	82
Figura 17 – Diagrama de Casos de Uso Simplificado Proativo	84
Figura 18 - <i>Dashboard</i> geral de OEE	87
Figura 19 – Níveis de maturidade do modelo Harbor	89
Figura 20 – Exemplo da interface de maturidade	90
Figura 21 – Método de busca dos contextos ARISA	94
Figura 22 – Modelagem de contextos Implementada	95
Figura 23 – Parte da árvore de contextos implementada	96
Figura 24 – Interface de criação de diálogos	98
Figura 25 – Interface de configuração de sinônimos	99
Figura 26 – Exemplo de programação na linguagem Lua	101
Figura 27 – Exemplo de codificação de comportamento em linguagem Lua	102
.....	
Figura 28 – Fluxograma Lógico Geral	105
Figura 29 – Fluxograma Lógico da Verificação de Cadastros	106
Figura 30 – Fluxograma Lógico da Verificação de Ordens	107

Figura 31 – Diagrama de Sequência Generalizado para Diagnóstico.....	110
Figura 32 – Prescrição usando sistema especialista	112
Figura 33 – Geração diária de relatório por empresa.....	112
Figura 34 – Diagrama de Implementação	115
Figura 35 – Exemplo de uso reativo.....	118
Figura 36 – Geração diária de relatório por empresa.....	120

LISTA DE QUADROS

Quadro 1 - Trabalhos retornados na SLR.....	47
Quadro 2 – Trabalhos selecionados na SLR.....	48
Quadro 3 – Requisitos Funcionais.....	61
Quadro 4 – Requisitos Não Funcionais	62
Quadro 5 – Perguntas do questionário aplicado.....	122

LISTA DE TABELAS

Tabela 1 – Tempos de Execução da Aplicação Intermediária	136
Tabela 2 – Redução semanal de carga de trabalho por cliente	142
Gráfico 1 – Utilização de chatbot como solução	124
Gráfico 2 – Diferencial em análises.....	125
Gráfico 3 – Grau de conforto com softbots	125
Gráfico 4 – Utilização como serviço	127
Gráfico 5 – Aceitação da solução.....	128
Gráfico 6 – Necessidade de conhecimento.....	128
Gráfico 7 – Aceleração da indústria 4.0	129
Gráfico 8 – Inovação	130
Gráfico 9 – Substituição de trabalhadores	131
Gráfico 10 – Capacidade prescritiva	132
Gráfico 11 – Necessidade de proatividade	133
Gráfico 12 – Atendimento aos requisitos	134

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
AV	Assistente Virtual
BP	Business Process
ERP	Enterprise Resource Planning
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IMAP	Internet Message Access Protocol
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
MES	Manufacturing Execution System
NLP	Natural Language Processing
OASIS	Organization for the Advancement of Structured Information Standards
OAuth	Open Authorization
PGEAS	Pós-Graduação em Engenharia de Automação e Sistemas
PLN	Processamento de Linguagem Natural
PMaaS	Production Manager as a Service
PME	Pequena e Média Empresa
POC	Proof Of Concept
QA	Quality Assurance
REST	Representational State Transfer
SaaS	Software as a Service
SLR	Systematic Literature Review
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture
SQL	Structured Query Language
UML	Unified Modeling Language
URI	Uniform Resource Identifier

URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WCM	World Class Manufacturing
WSDL	Web Service Description Language
XML	eXtensible Markup Language
XSD	XML Schema Definition

SUMÁRIO

1	INTRODUÇÃO	19
1.1	PROBLEMA GERAL E ESPECÍFICO	21
1.2	JUSTIFICATIVA	23
1.3	PERGUNTA DE PESQUISA	24
1.4	OBJETIVOS	24
1.4.1	Objetivo Geral	24
1.4.2	Objetivos Específicos	25
1.5	CONTRIBUIÇÃO CIENTÍFICA	25
1.6	PROPOSIÇÃO DE VALOR	25
1.7	ADEQUAÇÃO ÀS LINHAS DE PESQUISA DO PROGRAMA	26
1.8	ORGANIZAÇÃO DO DOCUMENTO	26
2	ASPECTOS METODOLÓGICOS	28
2.1	ENQUADRAMENTO METODOLÓGICO	28
2.2	A PESQUISA-AÇÃO	30
2.3	ROTEIRO BÁSICO DE PESQUISA	31
2.4	DELIMITAÇÃO DA PESQUISA	32
3	FUNDAMENTAÇÃO TEÓRICA	34
3.1	MES	34
3.2	ASSISTENTES VIRTUAIS	36
3.3	ANÁLISE DE NEGÓCIO	38
3.4	MODELOS DE MATURIDADE	40
3.5	SERVIÇOS WEB	41
3.5.1	Serviços <i>RESTful</i>	41
3.6	ARQUITETURA ORIENTADA A SERVIÇOS	42
3.7	SOFTWARE COMO SERVIÇO	43
3.8	GERENTE DE PRODUÇÃO COMO SERVIÇO	44
3.9	INTELIGÊNCIA ARTIFICIAL	45
3.9.1	Sistemas Especialistas	46
3.10	REVISÃO SISTEMÁTICA DA LITERATURA	46
4	ARQUITETURA PROPOSTA	58

4.1	CENÁRIO PRETENDIDO	58
4.2	REQUISITOS.....	61
4.2.1	Requisitos Funcionais.....	61
4.2.2	Requisitos Não Funcionais	62
4.3	GERENTE DE PRODUÇÃO COMO SERVIÇO (PMAAS).....	63
4.4	ARQUITETURA BASE.....	65
4.5	ARQUITETURA PROPOSTA	67
4.6	MÓDULO DE NLP	70
4.7	MÓDULOS DE INTERFACES	73
4.8	MÓDULO DE INTELIGÊNCIA	75
4.8.1	Pré-processamento	76
4.9	INTEGRIDADE DE DADOS.....	76
4.10	ANÁLISE DE DADOS	77
4.11	MÓDULO DE UTILIDADES	78
4.12	MODELO DE MATURIDADE.....	78
4.13	CASOS DE USO.....	81
4.13.1	Casos de uso do comportamento reativo.....	81
4.13.2	Casos de uso do comportamento proativo	84
5	PROTÓTIPO IMPLEMENTADO.....	86
5.1	ELEMENTOS DE IMPLEMENTAÇÃO.....	86
5.2	LIVEMES	87
5.3	MATURIDADE HARBOR.....	88
5.4	ESCOLHA DAS TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO	91
5.5	LIVIA: INSTÂNCIA DA ARISA NEST	92
5.5.1	Arisa nest.....	92
5.5.2	Integração com o Telegram	92
5.5.3	Conversação	93
5.5.3.1	<i>Contextos.....</i>	93
5.5.3.2	<i>Diálogos.....</i>	97
5.5.3.3	<i>Sinônimos</i>	98

5.5.4	Crenças	99
5.5.5	Interface com serviços externos	100
5.5.6	Comportamento proativo agendado	101
5.6	IMPLEMENTAÇÃO DA APLICAÇÃO INTERMEDIÁRIA	103
5.6.1	Análise de maturidade	104
5.6.1.1	<i>Verifica Cadastros</i>	106
5.6.1.2	<i>Verifica Ordens</i>	108
5.6.2	Sequência de comunicação	108
5.6.3	Implementações da inteligência	111
5.6.3.1	<i>Técnicas de IA</i>	113
5.6.3.2	<i>Envio de e-mail</i>	114
5.7	DEPLOY	114
6	ASPECTOS DE VALIDAÇÃO E TESTES	117
6.1	VALIDAÇÃO	117
6.1.1	Utilização em casos selecionados	117
6.1.1.1	<i>Validação do comportamento reativo</i>	117
6.1.1.2	<i>Validação do comportamento proativo</i>	120
6.1.1.3	<i>Validação da mudança de contexto dinâmica</i>	121
6.1.2	Questionário	121
6.1.2.1	<i>Perguntas do questionário</i>	122
6.1.2.2	<i>Respostas dos entrevistados</i>	123
6.2	TESTES	135
6.2.1	Teste unitário	135
6.2.2	Teste de portabilidade	135
6.2.3	Teste de desempenho	136
6.2.4	Teste de escalabilidade	137
6.2.5	Testes dos algoritmos de IA	137

7	ASPECTOS DE AVALIAÇÃO E ANÁLISE DE RESULTADOS	139
7.1	AVALIAÇÃO DA ARQUITETURA.....	139
7.2	AVALIAÇÃO DO USO	141
7.2.1	Comparação temporal entre Livia e especialista	141
7.2.2	Comparação de qualidade da análise	142
7.2.2.1	<i>Profundidade da análise</i>	142
7.2.2.2	<i>Integridade da análise.....</i>	143
7.3	AVALIAÇÃO DA IA	143
8	CONSIDERAÇÕES FINAIS.....	145
8.1	DIFICULDADES ENCONTRADAS	145
8.2	RESPOSTA À PERGUNTA DE PESQUISA.....	146
8.3	CONCLUSÕES.....	147
8.4	TRABALHOS FUTUROS	149
	REFERÊNCIAS	151

1 INTRODUÇÃO

O gerenciamento de produção ou *Production Management* (PM) compreende o tipo de atividades relacionadas ao planejamento, coordenação, supervisão, controle e tomada de decisão sobre recursos e resultados dos processos de negócios (KAUSHIK, 2014). Na Indústria 4.0, o PM também deve considerar o aumento da digitalização e interconexão de produtos inteligentes, serviços, sistemas de manufatura, cadeias de valor e modelos de negócios na Internet das Coisas, Serviços e Pessoas (CAMARINHA-MATOS, FORNASIERO e AFSARMANESH, 2017).

Isso traz novos requisitos técnicos gerais para informações corporativas e sistemas e tecnologias operacionais, como (LEE, BAGHERI e KAO, 2014) (LAMNABHI-LAGARRIGUE, ANNASWAMY e ENGELL, 2017): controle distribuído e descentralizado; autonomia de decisão; colaboração de sistemas ciberfísicos (CPS - Cyber-Physical Systems); virtualização; recursos de adaptação e *plug-and-play*; comportamento emergente e auto-organização; supervisão e resiliência; controle e otimização orientados a dados e em tempo real; e interação simbiótica de CPSs e humanos.

Neste contexto, um dos pilares da Indústria 4.0 tem a ver com uma gestão totalmente baseada em dados (*data-driven*), permanentemente coletados da empresa como um todo, para mais assertivas e adequadas tomadas de decisão (KUSIAK, 2018). Esta coleta de dados não é simples. Além de aspectos ligados a infraestruturas computacionais de suporte, há que se lidar com inúmeros problemas de integração e interoperabilidade de sistemas, além de posteriores sínteses e organização de dados para ajudar na gestão das empresas, seja atuando diante de problemas, seja antevendo problemas (ROMERO e VERNADAT, 2016).

Para lidar com isso, alguns trabalhos e soluções no mercado têm buscado o desenvolvimento de ambientes nos quais os gerentes podem ter acesso mais fácil, rápido, sistemático e preciso às informações relacionadas ao chão de fábrica das empresas, para apoiar decisões mais ágeis e confiantes com maior simbiose homem-máquina (por exemplo, (ROMERO e VERNADAT, 2016) (RABELO, ROMERO e ZAMBIASI, 2018)).

A implementação de tais ambientes é, no entanto, complexa. Em relação aos princípios básicos - a coleta em tempo real de informações do chão de fábrica, alguns trabalhos têm proposto abordagens baseadas em *Business Intelligence* (BI) bem como abordagens integradas e mais interoperáveis. Isto com o objetivo de tornar mais fácil e confiável o acesso às informações da empresa e para um suporte para a tomada de decisões via painéis de produção (por exemplo, (VUKŠIĆ, BACH e POPOVIČ, 2013) (MENDEZ, RABELO, *et al.*, 2018)). Isso é crucial, pois as informações são normalmente distribuídas por muitos sistemas diferentes que usam diferentes tecnologias, interfaces de usuário, formatos e terminologias, o que torna o acesso, a síntese de dados, a compreensão e o uso das informações corretas, às vezes, muito difíceis para os gestores.

Apesar dos benefícios trazidos por esses ambientes, estes por vezes trazem uma maior complexidade adicional para os gestores. A prática tem demonstrado que eles estão tornando os gestores cada vez mais expostos a uma quantidade enorme de informações sobre suas empresas, onde muitas ações de verificação, análise e supervisão, bem como tomadas de decisões críticas, precisam ser realizadas com mais frequência e rapidez (MCDERMOTT, 2017) (RABELO, ZAMBIASI e ROMERO, 2019). Outros problemas incluem a execução de ações repetitivas e/ou por vezes desnecessárias pelos gestores, devido à como o processo de negócios está desenhado (e quando desenhado) bem como a execuções incorretas ou com baixa qualidade devido à pressão por análises e tomadas ágeis de decisão. Ainda, avaliações empíricas demonstraram que, embora os gestores continuem tendo que verificar e fazer muitas coisas relacionadas aos *dashboards* de apoio à gestão, o fazem frequentemente sem uma orientação geral, sem *guidelines* de apoio, e nem sempre de forma eficiente (ABNER, RABELO, *et al.*, 2020).

Algumas abordagens têm sido propostas para tratar dessa problemática, sendo “robôs de software” (*software robots* ou simplesmente *softbots*) uma das consideradas mais promissoras (BRAIN, 2019). Um *softbot* pode ser definido como um sistema virtual, implantado em um determinado ambiente computacional, que automatiza e ajuda os seres humanos na execução de algumas tarefas com níveis variáveis de inteligência, autonomia e proatividade (KIM, 2005).

De acordo com um levantamento realizado, é previsto que 40% das grandes empresas tenham um ou mais assistentes inteligentes ou *chat/softbots* implementados em suas corporações até 2022 (BRAIN, 2019). Vários autores (e.g. (FRANZZON, AGOSTINO, *et al.*, 2020)) têm apontado para a necessidade de nível mais simbiótico de relação homem-máquina visando-se maior satisfação e maior produtividade no trabalho para melhor atender aos requisitos da Indústria 4.0, sendo *softbots* um tipo de tecnologia de suporte para o chamado “operador 4.0” (ROMERO, BERNUS, *et al.*, 2016).

(RABELO, ROMERO e ZAMBIASI, 2018) (RABELO, ZAMBIASI e ROMERO, 2019) demonstraram em cenários industriais reais controlados de chão de fábrica que *softbots* podem ajudar a mitigar aqueles problemas de acesso e uso de informação.

1.1 PROBLEMA GERAL E ESPECÍFICO

A problemática geral lidada nesta dissertação é a de que, embora todos aqueles problemas ocorram em basicamente todos os tipos e tamanhos de empresas (ASG, 2020), ele é mais crítico nas PMEs (Pequenas e Médias Empresas). Isto porque a maioria dos gestores de PMEs não estão acostumados a gerenciar as empresas sob a abordagem *data-driven*, e não têm base teórica sólida ou atualizada para gerenciar mais adequadamente as suas empresas de acordo com os mais modernos métodos e técnicas de gestão (MITTAL, KHAN, *et al.*, 2018) (VUKŠIĆ, 2013).

O problema específico tem a ver com um caso real, de uma empresa de pequeno porte de soluções para automação industrial, chamada *Harbor*¹, localizada em Florianópolis.

Com 20 anos de existência e especializada em sistemas do tipo MES (*Manufacturing Execution System*), ela tem centenas de clientes no Brasil. Investindo em inovação e em novos modelos de negócio, a *Harbor* desenvolveu um MES voltado principalmente para o mercado de PMEs, chamado *LiveMES*². Este sistema é hospedado na nuvem, de forma que as empresas-cliente não precisam instalá-lo

1 <https://www.harbor.com.br/>

2 <https://www.harbor.com.br/livemes-a-producao-nas-suas-maos/>

localmente. Dispositivos de IoT (*Internet of Things*) são instalados e configurados para cada equipamento industrial de cada empresa-cliente e tais dispositivos ficam permanentemente enviando em tempo real diversos dados de cada máquina para o *LiveMES*. Internamente, cada empresa tem uma “instância” no *LiveMES*. Os gestores de cada empresa têm acesso ao *LiveMES* a partir de seus computadores e *smartphones*, sob um modelo SaaS (*Software-as-a-Service*). Com isso, permite-se que acompanhem em tempo real os dados de cada máquina, e façam análises e tomem decisões com base em uma série de indicadores de desempenho e de negócio (KPIs - *Key Performance Indicators*), que são gerados e mostrados em *dashboards*.

Apesar de todas essas facilidades, o problema é que os gestores das PMEs têm, uma série de dificuldades. As principais são (ABNER, RABELO, *et al.*, 2020):

- (i) filtrarem quais informações do chão-de-fábrica são realmente relevantes para serem consideradas nas diversas situações de análise;
- (ii) interpretarem adequadamente o real significado dos muitos indicadores de desempenho operacionais e KPIs gerados, mesmo que exibidos em atraentes “painéis digitais de gestão” (*dashboard*);
- (iii) raciocinarem sobre as informações e *dashboards* para terem maior confiança as decisões adequadas para cada situação.

Em resumo, observou-se que apenas a disponibilização de sistemas para os gestores de PMEs não é suficiente para assumir que estes sejam capazes de realizar análises complexas, ou até mesmo tirar dos sistemas os ganhos de qualidade de produção e de produtividade das máquinas pretendidos.

Além disto, e como consequência disto, a *Harbor* passou a inserir no seu modelo de negócios a possibilidade dela mesma fazer análises periódicas para cada empresa se esta desejar (e pagar). Porém, estas análises das várias máquinas de cada empresa são trabalhosas e demoradas de serem realizadas, exige um grande conhecimento técnico e de gestão industrial, e consomem várias horas de pessoas da *Harbor*. Por conseguinte, isto passou a representar um problema adicional.

Nesse contexto, o problema para o qual se pretende apresentar uma contribuição é como que a abordagem e tecnologia de *softbots*, considerando o seu potencial, pode ajudar a mitigar esses problemas dos gestores das empresas-cliente

e dos analistas da *Harbor*. No caso, de acesso à informação, de apoio à análise e tomadas de decisão, etc., sob uma filosofia *data-driven* e, ao mesmo tempo, de se ter um ambiente mais efetivo de execução de ações pelos gestores e mais agradável e simbiótico entre estes e os sistemas de gestão envolvidos (o *LiveMES*, no caso).

Já do ponto de vista de arquitetura de sistemas e de ciclo de vida do *LiveMES*, a estratégia da empresa é que o tipo de suporte a ser dado pelo *softbot* pretendido seja complementar ao trabalho realizado pelos colaboradores e não uma automação completa de todas as análises realizadas por eles. Ou seja, no futuro outras funcionalidades deverão vir a ser adicionados ao sistema e ao *softbot* bem como este deve ser flexível e interoperável para poder se comunicar com outros sistemas.

1.2 JUSTIFICATIVA

Do ponto de vista da gestão da produção, o modelo da Indústria 4.0 e da filosofia *data-driven* impõem uma série de novos requisitos e demandas para empresas e gestores (LIAO, DESCHAMPS, *et al.*, 2017). Sob pressão de inovarem, muitas empresas têm investido na aquisição ou utilização de uma série de tecnologias e sistemas (CNI, 2016), dentre os quais os sistemas MES. Porém, e principalmente no universo das PMEs, tais gestores têm grande dificuldade e limitações técnicas de tirar proveito do que tais sistemas são capazes de propiciar (WMF, 2019). Além disto, os sistemas “clássicos” de MES não são desenhados para eles mesmos fazerem complexas análises em cima dos dados que eles coletam e organizam (ABNER, RABELO, *et al.*, 2020). Portanto, há oportunidades para se desenvolver algo adicional nessa direção.

Do ponto de vista de estado da prática, não se encontrou produtos no mercado que combinem sistemas MES com *softbots* para, tanto ajudar os gestores em tomadas de decisão, como criar um ambiente mais “natural” de interação gestores-MES. Aliás, este é justamente um dos diferenciais desejados pela *Harbor* no mercado.

Do ponto de vista científico, considerando a literatura avaliada, foram encontrados trabalhos que buscam criar modelos de maturidade para a indústria, e trabalhos que buscam problemas em processos produtivos e trabalhos que utilizam *softbots* na indústria. Porém, não foram encontrados trabalhos que utilizem *softbots*

com capacidade de avaliar processos produtivos e indicar pontos de melhoria com base em modelos de maturidade. Além disso, a quase totalidade de trabalhos sobre *softbots* (em seus diversos termos) foca basicamente em conversações básicas e não suportam uma integração efetiva com diversos sistemas empresariais/industriais para atender a requisições mais complexas dos gestores sobre variados aspectos de produção. Portanto, suas arquiteturas são relativamente simples, não atendendo suficientemente a requisitos da Indústria 4.0.

1.3 PERGUNTA DE PESQUISA

Considerando-se o problema e justificativa abordados, esta dissertação busca responder a seguinte pergunta de pesquisa:

Como desenvolver um softbot que se integre a e interopere com sistemas existentes num modo SaaS, que faça automaticamente análises de dados de produção, bem como interaja com os gestores para o fornecimento de informações específicas visando ajudá-los em tomadas de decisão?

1.4 OBJETIVOS

Nas seções abaixo estão descritos o objetivo geral e os objetivos específicos deste trabalho.

1.4.1 Objetivo Geral

Este trabalho tem como objetivo conceber uma arquitetura de *Software as a Service* (SaaS) de softbot aberta, flexível e escalável, que forneça suporte à monitoração, análise de dados e tomada de decisão para plantas de manufatura.

1.4.2 Objetivos Específicos

Os objetivos específicos são partes do objetivo geral que devem ser desenvolvidos ao longo da pesquisa de modo a construir e avaliar o modelo como um todo.

Estes objetivos podem ser agrupados da seguinte forma:

- Revisar literatura;
- Conceber e Especificar uma Arquitetura de Referência;
- Desenvolver protótipo computacional que implemente um assistente virtual com base na Arquitetura de Referência.
- Integração do *softbot* com o sistema comercial *LiveMes*;
- Estruturar um modelo de maturidade para análise de processos ligados ao *LiveMES*;
- Construir algoritmos para avaliação e determinação automática da maturidade dos processos do *LiveMES*;

1.5 CONTRIBUIÇÃO CIENTÍFICA

A contribuição científica pretendida com esta dissertação dá-se em alguns aspectos, em diferentes dimensões, com base em levantamentos do estado-da-prática e do estado-da-arte:

- Adaptação da abordagem de softbots e de modelos de maturidade para Indústria 4.0 para o contexto de sistemas MES;
- Utilização de *softbots* em cenários de gestão industrial / de chão de fábrica.

1.6 PROPOSIÇÃO DE VALOR

Além dos aspectos não funcionais pretendidos de serem suportados pela arquitetura proposta (mencionados no objetivo geral), o *softbot* derivado da arquitetura pretende fazer com que: i) os gestores tenham informações mais amplas, completas e confiáveis para suas tomadas de decisão; e ii) os analistas da *Harbor* consigam realizar suas avaliações de forma mais rápida, assertiva e confiável.

1.7 ADEQUAÇÃO ÀS LINHAS DE PESQUISA DO PROGRAMA

O trabalho aqui descrito se enquadra no contexto da área de Sistemas Computacionais, que é uma das linhas de concentração onde o Programa de Pós-Graduação em Engenharia de Automação e Sistemas realiza atividades de formação e pesquisa. Considerando a problemática exposta e a linha de concentração, este trabalho se adere a um conjunto de tópicos, nomeadamente os de Tempo Real, Sistemas Distribuídos e Redes, Engenharia de Software, Inteligência Artificial, e Informática Industrial.

1.8 ORGANIZAÇÃO DO DOCUMENTO

Além da introdução e contextualização apresentada neste primeiro capítulo, o restante do trabalho está dividido em outros sete capítulos, descritos brevemente a seguir:

O capítulo 2 apresenta os aspectos metodológicos dessa dissertação, o roteiro e as delimitações dessa pesquisa.

O capítulo 3 descreve a fundamentação teórica relacionada a esta pesquisa fazendo um breve resumo sobre os principais conceitos utilizados e apresenta a revisão do estado da arte, assim como a contribuição científica pretendida.

O capítulo 4 introduz a arquitetura proposta para desenvolvimento da solução. Neste capítulo encontram-se detalhes sobre os módulos que compõe o sistema pretendido.

O capítulo 5 descreve o protótipo construído, falando sobre detalhes da implementação.

O capítulo 6 faz a avaliação dos resultados obtidos com o protótipo construído.

No capítulo 7 são apresentados testes, validação dos resultados e feitas comparações entre resultados e objetivos.

Por fim, o capítulo 8 condensa o trabalho por meio das conclusões, contribuições, limitações e trabalhos futuros.

2 ASPECTOS METODOLÓGICOS

Este capítulo apresenta o enquadramento metodológico usado neste trabalho, assim como as etapas e procedimentos definidos visando atingir os objetivos estabelecidos.

2.1 ENQUADRAMENTO METODOLÓGICO

O enquadramento metodológico é um instrumento que permite planejar a execução da pesquisa proporcionando os meios para definir as condições dentro das quais os experimentos de pesquisa são realizados e delimitar o escopo do trabalho (GIL, 2010).

O trabalho de pesquisa realizado foi realizado dentro do seguinte enquadramento metodológico:

- Método de pesquisa: *Pesquisa-ação*
Pretende-se encontrar uma solução a um problema concreto e específico de uma comunidade intervindo diretamente na concepção de uma arquitetura e sistema (*softbot*) voltados a facilitar a análise de gestores de produção;
- Natureza de pesquisa: *Aplicada*
Visa aplicar os estudos ao redor da temática de *softbots* trabalhando com e como gestores de produção, tomando como base fundamentações teóricas básicas e tecnologias já existentes;
- Abordagem do problema: *Qualitativa*
Visa conceber uma arquitetura e software através do qual alguns experimentos podem ser efetuados para complementar a análise de valor da contribuição científica proposta;
- Tipo de Pesquisa: *Dedutivo*
Busca-se obter conclusões por meio de informações colhidas de entrevistas, referências bibliográficas e experimentos orientados a criação, utilização e integração de *softbots*;

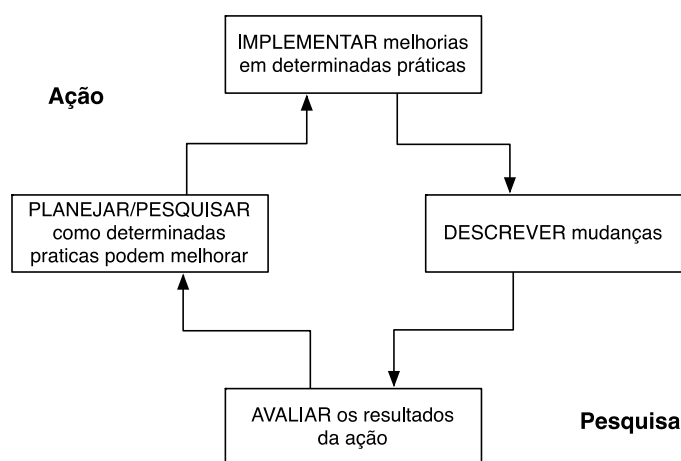
- Paradigma de Pesquisa: *Positivista*
É possível considerar uma realidade dentro da qual o problema está inserido. O problema é tangível, palpável, com algumas possibilidades de predição e controle. O pesquisador e o objeto de estudo conseguem ser delimitados. Parte-se do princípio que se consegue obter informações concretas do chão de fábrica de empresas e se desenvolver um sistema (*softbot*) que consiga processar tais informações e se integrar com um outro sistema (o *LiveMES*) já existente;
- Objetivo de Pesquisa: *Ação*
Visa desenvolver um conjunto de artefatos teóricos e computacionais com o objetivo de responder à pergunta de pesquisa;
- Tempo de Pesquisa: *Longitudinal*
Em alinhamento com a própria metodologia de desenvolvimento de software escolhida (ágil / incremental-iterativa), as informações para o desenvolvimento da arquitetura e do *softbot* foram coletadas, medidas e implementadas ao longo de todo trabalho;
- Observação: *Sistemática*
São realizadas observações no decorrer das diferentes etapas de desenvolvimento da arquitetura e do *softbot* visando-se garantir, tanto que os requisitos funcionais e não funcionais do sistema estavam sendo atingidos ao longo das suas evoluções de desenvolvimento, como a arquitetura atendia aos requisitos gerais idealizados;
- Procedimentos: pesquisas bibliográficas e testes experimentais.
Visa obter as informações sobre o estado da arte e da prática através de leituras de artigos (resultado de revisão sistemática da literatura) e avaliações de tecnologias e ferramentas computacionais, e usar este conhecimento como fundamentação para a concepção do modelo e dos experimentos.

2.2 A PESQUISA-AÇÃO

A Pesquisa-ação (*Action research*) é uma metodologia de base empírica, aplicada para conduzir intervenção, desenvolvimento e mudança no âmbito de grupos, organizações e comunidades de prática. Ela utiliza técnicas de pesquisa consolidadas para desenvolver ações para melhorar práticas de certos grupos (GIL, 2010).

A Pesquisa-ação tem características situacionais, pois procura identificar problemas específicos em situações específicas com a finalidade de atingir um resultado prático de acordo com o problema a se resolver e o cenário que se quer atender (GIL, 2010).

Figura 1 - Quatro fases básicas do ciclo da Pesquisa-ação



Fonte: GIL (2010)

A Pesquisa-ação utiliza um ciclo de atividades, ilustradas na Figura 1. A pesquisa passa pela identificação do problema, o planejamento de uma solução, sua implementação, e a avaliação de solução proposta. A fase de planejamento e pesquisa inicia por uma exploração do tema-objeto da pesquisa. Esta atividade consiste em uma imersão na literatura sobre o tema bem como contatos e/ou observações do grupo para o qual se pretende implementar melhorias de práticas (GIL, 2010). A partir dessa etapa, se procede à etapa de implementação de modificações. Nessa etapa se propõe melhorias que possam sanar ou melhorar as dificuldades encontradas pelo grupo estudado. As mudanças são descritas e

avaliadas pelo grupo ligado ao objeto da pesquisa. Tomando como base as respostas da avaliação feita pelo grupo, se inicia outros ciclos de Pesquisa-ação, planejando-se novas etapas da pesquisa para identificar e corrigir problemas que persistirem.

2.3 ROTEIRO BÁSICO DE PESQUISA

Neste momento são definidas as seguintes principais etapas para a realização da pesquisa. Considerando-se a natureza e os objetivos desse trabalho de pesquisa, em termos gerais a metodologia de desenvolvimento é sequencial, com algumas etapas podendo ser realizadas em paralelo.

- 1. Revisão bibliográfica:** A abordagem adotada para efetuar a primeira parte da pesquisa é o método SLR - *Systematic Literature Review* (KITCHENHAM e CHARTERS, 2007), para assim conseguir obter a maioria dos trabalhos relevantes na área (pelo menos nos últimos 10 anos) e garantir uma boa análise do estado da arte, identificação criteriosa das lacunas nas áreas e da contribuição científica pretendida;
- 2. Projeto da arquitetura:** Baseando-se na revisão do estado da arte e com base nos indicadores mais utilizados de acordo com a base de dados fornecida, foi idealizada uma arquitetura que fosse capaz de buscar os dados, processá-los, e interagir com o usuário, sob um modelo SaaS. Em outras palavras, identificaram-se os requisitos funcionais, não-funcionais e atores do ambiente previsto;
- 3. Seleção dos modelos de referência:** Para nortear a implementação, foram escolhidos um modelo de maturidade de referência e uma arquitetura de referência;
- 4. Seleção de tecnologias e ferramentas:** para a implementação do *softbot* idealizado foram selecionadas TIs e ferramentas de programação, protocolos de comunicação, etc. Além disso, sabia-se que o sistema a ser integrado deveria ser o *LiveMES* (com sua respectiva interface/API), o que incluiu a questão de uso de serviços web (*web services* e serviços *REST*). De forma similar, optou-se por

utilizar a plataforma-como-serviço *ARISA NEST* para geração de instâncias de *softbots*. Sistemas complementares, serviços, uso de bibliotecas, etc., foram usadas de acordo com as referências;

5. **Implementação do softbot:** implementação incremental de versões do protótipo, com a elaboração de testes, refinamentos, etc., com este ciclo se repetindo até que os resultados foram considerados finais;
6. **Construção de cenários de testes:** construção de cenários de testes, verificações, avaliações e validação para análises intermediárias e final da arquitetura e do *softbot* implementado;
7. **Experimentação e documentação:** realização de experimentos com base nos cenários de teste construídos, sendo os resultados obtidos documentados para análises posteriores;
8. **Análise final:** baseado nas informações adquiridas na etapa anterior, foi feita uma análise sobre os resultados obtidos;
9. **Escrita de publicações:** Esta etapa contribuiu para dar maior solidez ao trabalho na medida que artigos científicos possibilitam consultar a comunidade científica sobre os resultados intermediários e final do trabalho. No caso, foi publicado um artigo numa importante conferência internacional da IFIP e um outro artigo com os resultados finais está sendo preparado para ser enviado para uma revista científica internacional.

2.4 DELIMITAÇÃO DA PESQUISA

Considerando a viabilidade temporal deste trabalho de pesquisa, atingir o objetivo geral do trabalho requereu alguns recortes analíticos, a adoção de alguns pressupostos, e algumas simplificações já desde o seu início, a saber:

- O banco de dados do MES utilizado possui informações de diversas empresas, mas para o escopo deste trabalho foi escolhida a empresa com mais dados disponíveis para a criação do protótipo;

- Existem diversos cenários possíveis em um ambiente industrial, mas apenas alguns cenários mais comuns e relevantes no chão de fábrica foram considerados;
- O *softbot* não possui conhecimentos fora do escopo do domínio de aplicação de produção e chão de fábrica, que por sua vez são delimitados pelas funcionalidades do sistema *LiveMES*;
- Apenas comunicação via texto é suportada.

3 FUNDAMENTAÇÃO TEÓRICA

Este capítulo visa dar um panorama das principais fundamentações teóricas de base para a pesquisa. No decorrer do desenvolvimento do trabalho, tais estudos foram bastante aprofundados, em largura e profundidade, de acordo com os avanços e necessidades da pesquisa.

3.1 MES

A origem do conceito MES é datada no começo dos anos 80, com os sistemas de aquisição de dados. Nesta época era comum uma grande quantidade de softwares de aquisição de dados dedicados para cada uma das disciplinas corporativas de negócio, como: planejamento de produção, qualidade e manutenção (KLETTI, 2007).

As definições do que um sistema MES deveria fazer dependiam das vontades e interesses de cada autor. Em 1997 a MESA (*Manufacturing Enterprise Solutions Association*) deu um passo em direção a padronização de sistemas MES propondo uma definição formal do MES:

MES entrega informação que possibilita a otimização de ativos de produção desde a criação da ordem até o bem produzido. Usando dados acurados e atuais, um MES guia, inicia, responde e relata as atividades da planta quando elas ocorrem. O rápido tempo de resposta para mudanças de condições, junto com o foco na redução de atividades que não agregam valor, geram operações e processos efetivos na planta. O MES melhora o retorno dos ativos operacionais, bem como a entrega no prazo, as rotações de estoque, a margem bruta e o desempenho do fluxo de caixa. Um MES fornece informações críticas sobre as atividades de produção na empresa e na cadeia de suprimentos por meio de comunicações bidirecionais. (MESA 1997c)

Para atender às necessidades de vários ambientes de fabricação, a MESA identificou 11 funções principais do MES (KLETTI, 2007):

- Programação das operações: sequenciamento e cronometragem de atividades para desempenho otimizado da planta com base nas capacidades finitas dos recursos.

- Estado e alocação de recursos: orientando o que as pessoas, máquinas, ferramentas e materiais devem fazer e rastreando o que estão fazendo ou acabaram de fazer.
- Expedição de unidades de produção: dando o comando para enviar materiais ou pedidos para certas partes da planta para iniciar um processo ou etapa.
- Controle de documentos: gerenciamento e distribuição de informações sobre produtos, processos, projetos ou pedidos, bem como coleta de declarações de certificação de trabalho e condições.
- Rastreabilidade e genealogia do produto: monitorando o progresso de unidades, lotes ou lotes de saída para criar um histórico completo do produto.
- Análise de desempenho: comparando resultados medidos na planta com metas e métricas definidas pela corporação, clientes ou órgãos reguladores.
- Gerenciamento de mão de obra: rastreando e direcionando o uso de pessoal durante um turno com base em qualificações, padrões de trabalho e necessidades de negócios.
- Gerenciamento de manutenção: planejamento e execução de atividades apropriadas para manter o equipamento e outros ativos de capital da planta em execução.
- Gerenciamento de processos: direcionando o fluxo de trabalho na planta com base nas atividades de produção planejadas e reais.
- Gerenciamento da qualidade: registrando, acompanhando e analisando as características do produto e do processo em relação aos ideais de engenharia.
- Coleta/aquisição de dados: monitoramento, coleta e organização de dados sobre os processos, materiais e operações de pessoas, máquinas ou controles.

Entre essas 11 funcionalidades, algumas estão diretamente vinculadas ao processo, como programação e controle de qualidade, enquanto outras, como

gerenciamento de recursos e rastreabilidade, são melhor descritas como funções cruzadas.

O MES é então um intermediário entre os vários departamentos da empresa. Esse padrão agora é amplamente aceito pelos participantes do mercado para o design de fluxos de informações entre as aplicações no nível da fábrica e as de nível superior. Também permite que a indústria tenha uma terminologia consistente. Todas as funções do MES contribuem para alcançar o desempenho no prazo e a total aderência às programações de pedidos dos clientes. De acordo com a pesquisa da MESA (MESA 1997c), os sistemas MES forneceram às empresas de manufatura alguns dos benefícios mais impressionantes de qualquer software de manufatura, como uma redução média de 45% no tempo do ciclo de manufatura, uma melhoria significativa da flexibilidade para responder às demandas dos clientes, a realização de certos graus de fabricação ágil e satisfação do cliente (KLETTI, 2007).

3.2 ASSISTENTES VIRTUAIS

Baseados em *chatbots*, agentes inteligentes e conversação em linguagem natural, os AVs interagem com pessoas através de linguagem natural e são capazes de executar tarefas para elas (SCHIAFFINO e AMANDI, 2006).

O primeiro *chatbot* se chamava ELIZA e foi desenvolvido por (WEIZENBAUM, 1966) para simular conversas com um psicoterapeuta. Muito do que foi desenvolvido nesse projeto, foi utilizado por muitos anos como base de estudos. Em 2001 foi criada a primeira versão de uma linguagem para *chatbots* desenvolvida com origem na linguagem XML. Essa linguagem, criada por Richard Wallace, se chama AIML, do inglês *Artificial Intelligence Markup Language* (WALLACE, 2003).

O grande diferencial da AIML com relação a programação estruturada utilizada tradicionalmente, foi a abstração entre a linguagem para modelar a conversação e as linguagens usadas para a programação do software que interpreta a conversa, o “motor” do *chatbot*. Isso permite que o software por trás do *chatbot* seja projetado e construído de forma independente de qualquer possível diálogo que o *chatbot* venha a possuir. Da mesma forma, a conversa pode ser construída sem que haja preocupação com detalhes de software e até mesmo que o profissional

responsável pela elaboração da conversação não necessite conhecer linguagem alguma, além de AIML (SHAWAR e ATWELL, 2007).

Segundo (ADAMOPOULOU e MOUSSIADES, 2020) existem várias formas de classificar os *chatbots*, podem ser classificados com base no seu domínio de conhecimento, sendo de *domínio aberto* quando podem falar sobre temas mais gerais e de *domínio fechado* quando são focados em algum conhecimento específico. Podem ser classificados com base em suas metas primárias, sendo:

- Informativo: quando tem o objetivo de fornecer informações disponíveis em uma base de dados;
- Baseado em chat: quando tem por objetivo conversar da forma mais humana possível;
- Baseado em tarefas: quando seu objetivo principal é voltado para realização de atividades, normalmente dependendo de comunicação com outros sistemas

Existem ainda outras formas de classificação de *chatbots*, nas quais se destaca a classificação baseada na forma de processamento das entradas de dados pelos usuários e nos métodos de geração de respostas. Essa classificação distingue três tipos de *chatbots* (ADAMOPOULOU e MOUSSIADES, 2020):

- Baseados em regras: são o tipo de chatbot mais antigo e possuem um grande número de implementações atualmente. Os diálogos são escolhidos com base em regras fixas predefinidas. Seu reconhecimento de texto se dá baseado em padrões textuais sem a preocupação com o significado das sentenças;
- Baseado em recuperação: ligeiramente diferente do anterior, esse tipo de chatbot consulta e analisa recursos disponíveis usando APIs e seleciona candidatas a respostas de forma flexível, antes de fazer o casamento (*matching*) com a pergunta.
- Gerador: utiliza mensagens atuais e passadas para produzir uma resposta mais humana para as perguntas que são feitas. Normalmente é baseado em técnicas de inteligência artificial.

Hoje em dia, existem plataformas que disponibilizam as mais variadas funcionalidades para facilitar a criação, controle, manutenção, disponibilização, integração e até comercialização de *chatbots*. Muitas dessas plataformas são suportadas pelas maiores empresas de software da atualidade e por isso possuem incríveis capacidades de processamento e integração, além disso, normalmente disponibilizam funcionalidades para aprendizado do *chatbot* com utilização de técnicas de IA (ABDUL-KADER e WOODS, 2015).

3.3 ANÁLISE DE NEGÓCIO

Business Analytics refere-se a métodos e técnicas usadas para medir o desempenho de uma organização que explora seus dados (geralmente com base em análises estatísticas) para obter visibilidade e impulsionar o planejamento e a tomada de decisões dos negócios (POCHIRAJU, 2019).

Segundo (POCHIRAJU, 2019), existem cinco tipos de análise:

- *Análise de planejamento: qual é o nosso plano?;*
- *Análise descritiva: o que aconteceu?;*
- *Análise de diagnóstico: por que isso aconteceu?;*
- *Análise preditiva: o que acontecerá a seguir?;*
- *Análise prescritiva: o que deve ser feito sobre isso?*

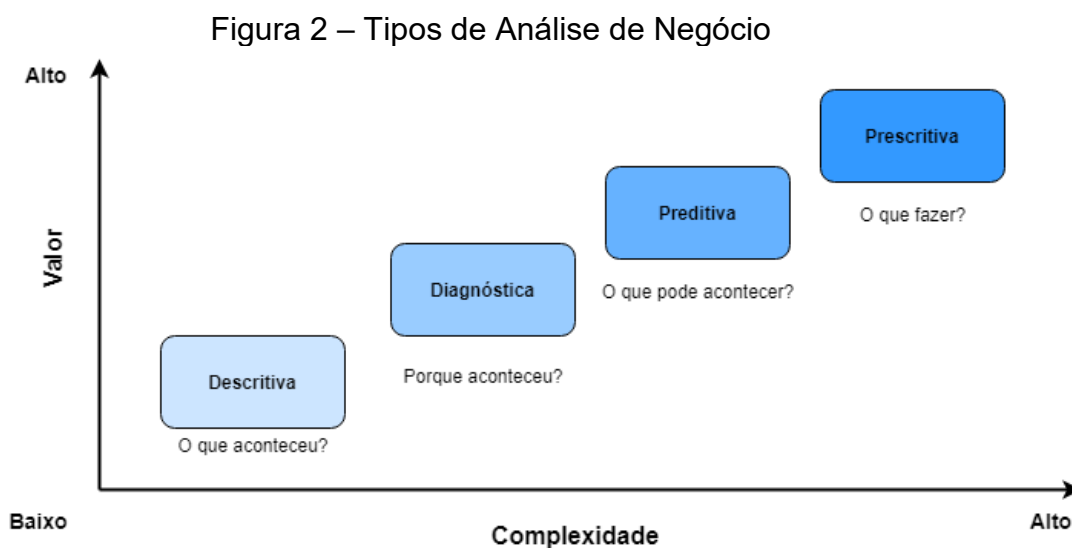
Dentre estes, é importante destacar os tipos de análise descritiva, preditiva e prescritiva, que são foco do presente trabalho.

A análise descritiva manipula dados brutos de várias fontes de dados para fornecer informações valiosas sobre o passado. No entanto, essas descobertas simplesmente sinalizam que algo está errado ou certo, sem explicar o porquê. Por esse motivo, não é recomendado que empresas altamente orientadas a dados se contentem apenas com análises descritivas, o ideal é combiná-las com outros tipos de análise de dados (POCHIRAJU e SESHADRI, 2019).

A análise preditiva pertence a tipos avançados de análise e traz muitas vantagens, como análise sofisticada baseada em máquina ou aprendizado profundo

e abordagem proativa que as previsões permitem. No entanto, a previsão é apenas uma estimativa, cuja precisão depende muito da qualidade dos dados e da estabilidade da situação, por isso requer tratamento cuidadoso e otimização contínua (POCHIRAJU e SESHADRI, 2019).

A análise prescritiva utiliza ferramentas e tecnologias avançadas, como aprendizado de máquina, algoritmos híbridos em dados estruturados e não estruturados, além das *regras de negócio (business rules)* que são as regras, muitas vezes descritas com lógica “SE-ENTÃO”, que estabelecem as condições e consequências esperadas (por exemplo, Se a performance da máquina está baixa, então é bom verificar motivos de paradas de máquina). Isto faz com que a este tipo de análise se torne sofisticada para implementar e gerenciar. Além disso, esse tipo de análise de dados de última geração requer não apenas dados internos históricos, mas também informações externas devido à natureza dos algoritmos nos quais se baseia (POCHIRAJU e SESHADRI, 2019).



Fonte: Elaborada pelo autor (2020)

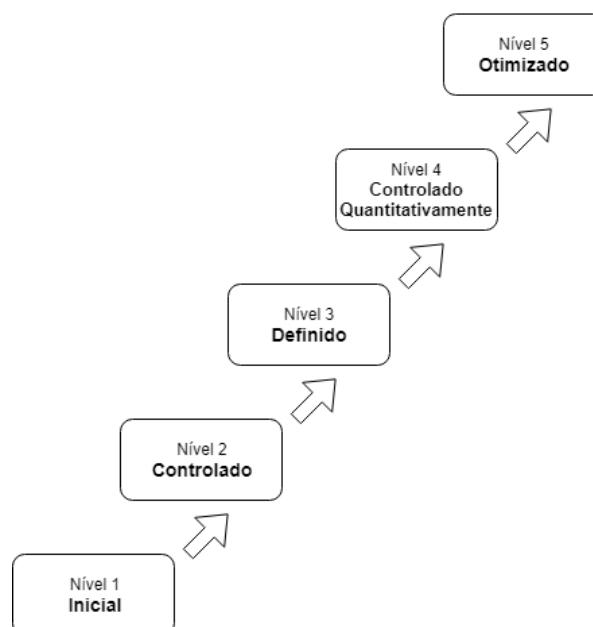
3.4 MODELOS DE MATURIDADE

Em diferentes domínios, modelos de maturidade têm sido propostos como uma forma de organizar um corpo de conhecimento e avaliar as organizações.

A base para a maioria destes modelos foi o *Capability Maturity Model* (CMM) – atualmente evoluído para o *Capability Maturity Model Integration* (CMMI) – desenvolvido para avaliar a maturidade ou capacidade dos processos de desenvolvimento ou manutenção de software das organizações. O CMMI funciona como um *framework* que organiza um conjunto de práticas básicas de engenharia de software para guiar os esforços de melhoria de processos. Estas práticas devem ser aplicadas de forma sistemática para se atingir um determinado padrão de qualidade nos produtos e serviços (CHRISISS, KONRAD e SHRUM, 2011).

Segundo (KLETTI, 2007), “Como na aviação, onde cada acidente aéreo faz com que voar seja mais seguro, em uma fábrica cada erro deve fazer a empresa melhor”

Figura 3 – Níveis de Maturidade CMMI
Níveis de Maturidade CMMI



Fonte: Elaborada pelo autor (2020)

A Figura 3 ilustra como são organizados os níveis de maturidade CMMI. O nível Inicial é o primeiro, seguido de Controlado, Definido, Controlado Quantitativamente e Otimizado como último nível.

3.5 SERVIÇOS WEB

Um serviço é um componente de software que possui uma funcionalidade de negócios bem definida e uma interface de comunicação capaz de ser publicada e descoberta por consumidores de serviços. O agrupamento desses componentes permite a comunicação entre eles através de trocas de mensagens permite a criação de diferentes aplicações e processos de negócios com baixo acoplamento (O'BRIEN, MERSON e BASS, 2007).

A implementação dos serviços pode ocorrer com a utilização de diversas técnicas e tecnologias, porém a implementação de serviços web é a mais amplamente utilizada devido a sua padronização (GESSER, 2006). De forma geral, um serviço web pode ser definido como um sistema de software projetado para suportar a interação entre máquinas conectadas em uma rede de comunicações (W3C, 2004). Existem dois grandes grupos de tecnologias para a implementação de *web services*: serviços web baseados no protocolo *SOAP* e serviços web *RESTful*.

3.5.1 Serviços *RESTful*

Os serviços *RESTful* são voltados a recursos e utilizam a arquitetura *REST*, do inglês (*REpresentational State Transfer*). A abstração de qualquer conceito bem definido pode ser um recurso, por exemplo, um centro de trabalho, um histórico de produção, um sinal vindo de uma máquina, uma ordem de produção, uma pergunta de um usuário, etc. Os recursos podem ser detectados por clientes de forma única por meio de *identificadores* chamados URIs (*Uniform Resource Identifier*) e podem ser acessados por uma *interface unificada* como o protocolo HTTP (FIELDING, 2000) (PAPAZOGLU, 2012) (ROTEM-GAL-OZ, BRUNO e DAHAN, 2012).

A representação de um recurso se dá através do encapsulamento de suas características, informações e estado da interação com o servidor. Para a

comunicação entre cliente e servidor, esse encapsulamento é serializado em um formato (JSON, XML, HTML, RDF) que tanto o cliente quanto o servidor entendam (ALLAMARJU, 2010).

Uma das características mais marcantes do estilo de arquitetura REST é a transferência de estado (*State Transfer*), em que a cada requisição são transmitidos todos os dados necessários para a contextualização do recurso. Sem que seja mantido qualquer dado de estado no servidor (*Stateless*) ou que seja necessária a criação de uma sessão (W3C, 2004).

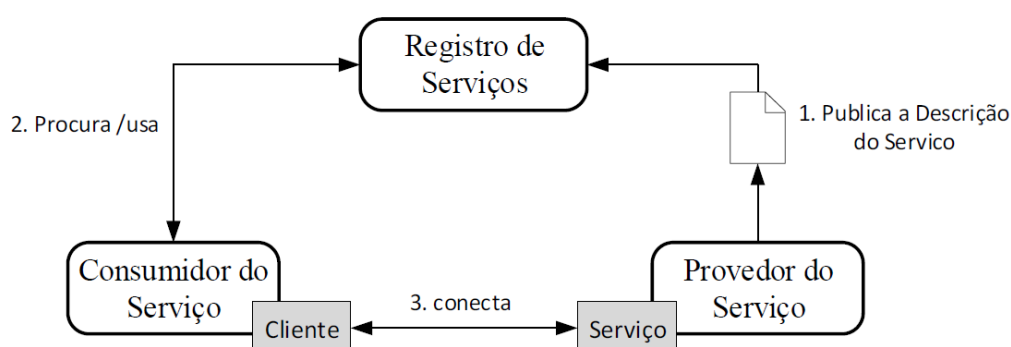
3.6 ARQUITETURA ORIENTADA A SERVIÇOS

A Arquitetura Orientada a Serviços ou SOA, do inglês *Service Oriented Architecture*, é um estilo de arquitetura de software que surgiu com a proposta de compartilhamento de lógica e dados entre aplicações. Para isso, esse estilo arquitetônico é baseado em componentes de software chamados de serviços. (ROTEM-GAL-OZ, BRUNO e DAHAN, 2012).

Visando superar os problemas de integração fortemente acoplada das aplicações monolíticas tradicionais, o paradigma SOA foi proposto oferecendo baixo acoplamento, flexibilidade e interoperabilidade permitindo compor dinamicamente diferentes tecnologias independentemente da plataforma tecnológica utilizada. Este paradigma incentiva a reutilização de software, reduz o tempo de implementação e de manutenção, mais do que isso, é possível publicar, descobrir e compor um conjunto de serviços através da internet de forma dinâmica (JARDIM-GONCALVES; GRILO; STEIGER-GARCAO, 2006).

Segundo Booth et al. (2004), a arquitetura SOA é composta por três componentes básicos: o provedor, o registro e o consumidor. Sendo o Provedor de Serviços que disponibiliza as aplicações de softwares e processos de negócio na forma de serviços; o Consumidor de Serviços (usuário, aplicação ou outro serviço) que localiza e invoca os serviços conforme sua necessidade; e o Registro de Serviços que fornece uma forma de localização dos serviços por meio de suas descrições e interfaces

Figura 4 - Blocos da Arquitetura Orientada a Serviços



Fonte: Adaptada de (JARDIM-GONCALVES; GRILO; STEIGER-GARCAO, 2006)

A Figura 4 ilustra como os componentes da arquitetura orientada a serviços se relacionam. A interação inicia quando um provedor disponibiliza a descrição de um serviço no registro de serviços para que este possa ser descoberto e invocado pelos consumidores. Quando precisa invocar um serviço, o cliente procura no registro que retorna uma lista de descrições relacionadas com os parâmetros de busca utilizados pelo cliente. De posse das descrições dos serviços encontrados, o cliente escolhe uma dessas descrições para obter o endereço do serviço, detalhes das mensagens e assim conseguir invocá-lo (CHEN, 2012).

3.7 SOFTWARE COMO SERVIÇO

Softwares como serviços (SaaS) é um modelo de negócios (e não uma arquitetura de software) para exploração comercial de softwares onde a rentabilidade do software não vem através de licenciamento clássico de produto e sim por meio de pagamentos periódicos pela utilização temporária (ESTEFAN, LASKEY, *et al.*, 2006). Ao invés de a empresa fornecedora obter a receita de forma singular e completa, essa receita é recebida de forma parcial e constante reduzindo as grandes variações em suas arrecadações (SUSARLA, BARUA e WHINSTON, 2009) (KINDSTRÖM, 2010).

De forma geral, esse modelo possui as características de não necessitar de um grande investimento inicial do cliente, retirar as responsabilidades de manutenção e infraestrutura do cliente, ser mais facilmente atualizado e ser uma solução mais padronizada entre clientes (COHEN e WHANG, 1997) (CHOUDHARY, 2007).

Embora o conceito de produto “*as-a-Service*” tenha se originado nos softwares, este modelo de “algo” como serviço pode ser expandido e incorporado a outros produtos e trabalhos. A música, por exemplo, é um produto que atualmente é consumido como serviço ao invés de comprado em alguma mídia permanente. Da mesma forma, grandes empresas oferecem a utilização de seus recursos computacionais para a utilização como serviço. Isso permite que os utilizadores de tais serviços, não necessitem de investimentos em infraestrutura ou manutenção e possam focar seus esforços em seu próprio negócio. (VANDERMERWE e RADA, 1988) (KASTALLI e LOOY, 2013).

3.8 GERENTE DE PRODUÇÃO COMO SERVIÇO

Esse mesmo tipo de abstração pode ser aplicado ao trabalho de um gerente de produção. O gerente de produção é um profissional caro e especializado para uma linha de produção. Suas atribuições são gerenciais, avaliando os processos produtivos, identificando problemas, elaborando soluções para falhas no processo, otimizando tempos produtivos e reduzindo ineficiências. Enfim, é um profissional para tornar o processo produtivo mais eficiente (NEELY, 2008).

Dadas as restrições financeiras das PMEs, o investimento em profissionais que possuam valor agregado de forma indireta, como é o caso dos gerentes de produção, é negligenciado em prol de profissionais que apresentem um ganho de forma direta e visível, como é o caso de operadores. (ATKINSON, 1984) (NEELY, 2008).

Usando a filosofia SaaS a empresa provedora contrata esses profissionais e disponibiliza um serviço de gerente de produção de forma remota para empresas clientes. Estes gerentes de produção atuam como profissionais terceirizados acompanhando os indicadores e dando suporte quando solicitados, desta forma esse profissional pode ser encarado como um consultor em gerenciamento de produção.

As empresas clientes usufruem desse serviço de forma parcial, ao invés de ser em período integral, porém a uma fração do custo e sem incorrer a encargos trabalhistas (NEELY, 2008) (KASTALLI e LOOY, 2013).

3.9 INTELIGÊNCIA ARTIFICIAL

Ao contrário da inteligência “natural” que é observada em pessoas e animais, a inteligência *artificial* (IA) é uma simulação da inteligência natural, mas observada em máquinas. De forma genérica, a IA pode ser definida como o estudo de agentes inteligentes em sistemas capazes de perceber o seu ambiente e tomar ações que maximizem as chances de sucesso de dada tarefa (RUSSEL e NORVIG, 2013).

A premissa de que “a inteligência humana pode ser descrita tão precisamente que uma máquina pode ser construída para simulá-la” serviu como base para a criação desse campo de estudo (CREVIER, 1993).

O primeiro trabalho reconhecido hoje em dia como sendo do campo de estudo de IA foi o estudo (MCCULLOCH e PITTS, 1943), onde foi criado um neurônio artificial “Turing-completo” já em 1943.

O estudo de IA evoluiu muito desde então diversificando os estudos tanto quanto aos algoritmos utilizados quanto nos paradigmas. Podemos dividir o estudo da IA em dois grandes grupos de estilos, a IA Simbólica e a IA Conexionista.

Na IA Simbólica busca-se fazer uma relação com a forma de pensar humana. Este tipo de estilo de IA se popularizou com os Sistemas Especialistas. Estes são os exemplos clássicos de IA simbólica, pois em geral, necessitam que o conhecimento sobre o problema seja definido no sistema de forma manual para que assim ele possa tomar as decisões. (AUFFARTH, 2020)

Já a IA conexionista, ao invés de se basear na forma de pensar humana, se baseia em simular os componentes do cérebro e tem como principal exemplo as Redes Neurais. Inicialmente, com as limitações impostas pelas capacidades computacionais do século passado, o avanço nessa linha de pesquisa da IA não teve muitas contribuições, se comparado aos dias de hoje. Com o avanço na capacidade computacional dos *hardwares*, aprimoramentos das redes neurais, como as redes

profundas (*deep learning*) se tornaram viáveis mesmo com o alto custo computacional. (AUFFARTH, 2020)

3.9.1 Sistemas Especialistas

Um Sistema especialista é um Sistema desenvolvido de forma que simule os conhecimentos e tomadas de decisão de um especialista. Este tipo de sistema é desenvolvido a fim de resolver problemas complexos com um modelo do conhecimento. O primeiro sistema especialista data de 1970, mas foi na década de 1980 que os sistemas especialistas se popularizaram. Os sistemas especialistas foram as primeiras formas de IA a serem realmente bem-sucedidas. (RUSSEL e NORVIG, 2020)

Eles podem ser divididos em dois subsistemas: o motor de inferência e a base de conhecimento. A base de conhecimento representa os fatos e regras, já o motor de inferência, utiliza essas regras e fatos. (RUSSEL e NORVIG, 2020)

Na base de conhecimento, as regras são descrições das consequências de determinadas condições. As regras traduzem de forma quase literal o conhecimento dos especialistas. O motor de inferência é um sistema automatizado de racionalização que avalia o estado atual da base de conhecimento, aplica as regras relevantes e infere novos conhecimentos na base de conhecimentos. Existem dois modos de funcionamento dos motores de inferência: *forward chaining* e *backward chaining*. No primeiro, o raciocínio segue a ordem usual, partindo das premissas e chegando a uma conclusão, já no segundo o raciocínio é inverso, partindo das conclusões e se chegando às premissas. (JACKSON, 1998).

3.10 REVISÃO SISTEMÁTICA DA LITERATURA

Para a revisão do estado da arte foi utilizado o método SLR – *Systematic Literature Review* (KITCHENHAM e CHARTERS, 2007). Para o método SLR a revisão é dividida em fases.

Primeiramente é feito um planejamento com a intenção de levantar os objetivos e justificar a revisão da literatura para o problema que se está tentando resolver. No capítulo 1 esses tópicos são abordados.

Para a próxima fase é estabelecido o processo que é efetivamente executado na revisão da literatura. Desta forma foram selecionadas as bases de dados para pesquisa, foram definidas *strings* de busca geral com palavras-chaves relacionadas aos temas do trabalho e os critérios de exclusão para refinamento das buscas foram definidos com algumas iterações de buscas.

Tendo como base as áreas de pesquisa que englobam o objetivo desta dissertação, as bases de dados científicos mais relevantes para temas como engenharia, software, computação e produção são: *IEEEExplore*, *Google Scholar*, *Scimedirect*, *SpringerLink* e *ACM* (BRERETON, KITCHENHAM, *et al.*, 2007).

Com as bases de dados selecionadas, uma *string* de busca geral deve ser elaborada considerando os termos (em inglês) relacionados com Modelos de Maturidade, Análise de Negócios, *Softbots*, Assistentes Virtuais, Sistema de Execução de Manufatura, Sistemas como Serviços, Arquitetura e Pequenas e Médias Empresas, assim como os seus sinônimos e variações. Essas buscas foram aplicadas em cada uma das bases observando-se as particularidades dos filtros de busca avançada de cada uma das bases de dados selecionada para a pesquisa.

Quadro 1 - Trabalhos retornados na SLR

Bases de dados	Trabalhos retornados
IEEEExplore	206
Google Scholar	264
Scimedirect	126
SpringerLink	208
ACM	167
Total	971

Fonte: Elaborada pelo autor (2020)

Por se tratar de um tema que engloba várias frentes de pesquisa, os resultados obtidos utilizando uma *string* única de busca geral foram poucos ou nulos para algumas bases de dados. Com isso se decidiu dividir a *string* de busca geral em *strings* de buscas específicas relacionadas a partes do problema e não ao todo. Desta forma houve um agrupamento das palavras-chave em grupos relacionados a “modelos de maturidade na manufatura”, “*softbots* ajudando na indústria”, “*softbots* distribuídos” e “gerência de produção como serviço”.

Nem todos os resultados obtidos com a aplicação das *strings* de busca se adequavam ao tema desta dissertação. Para filtrar os resultados de forma a se adequarem mais ao tema dessa pesquisa, alguns critérios de exclusão foram adotados. Os critérios definidos para eliminar alguns trabalhos dessa revisão foram:

1. Artigos que não eram escritos em inglês ou Português
2. Artigos que não tratavam de manufatura
3. Artigos duplicados do mesmo trabalho (neste caso o artigo com maior nível de detalhamento do mesmo trabalho foi considerado)
4. Artigos com menos de 4 páginas
5. Artigos informativos

Os trabalhos restantes foram avaliados considerando-se seu título, palavras-chave, *abstract* e conclusões. Após esta avaliação os trabalhos relevantes foram estudados com uma maior profundidade chegando-se assim ao resultado da revisão sistemática da literatura.

Os trabalhos considerados mais fortemente relacionados a essa dissertação que foram resultantes desse processo de revisão estão identificados no quadro a seguir.

Quadro 2 – Trabalhos selecionados na SLR

	Título do trabalho	Referência
1	Hybrid Teams: Flexible Collaboration between Humans, Robots and Virtual Agents	(SCHWARTZ, ZINNIKUS, <i>et al.</i> , 2016)

2	Applying Chatbots to the Internet of Things: Opportunities and Architectural Elements.	(KAR e HALDAR, 2016)
3	The Social Factory: Connecting People, Machines and Data in Manufacturing for Context-Aware Exception Escalation	(KASSNER, HIRMER, <i>et al.</i> , 2017)
4	Real world smart chatbot for customer care using a software as a service (SaaS) architecture	(D'SILVA, THAKARE, <i>et al.</i> , 2017)
5	Smart Operators in Industry 4.0: A Human-centered Approach to Enhance Operators' Capabilities and Competencies within the New Smart Factory Context	(LONGO, NICOLETTI e PADOVANO, 2017)
6	ChatPy: Conversational agent for SMEs	(PÉREZ, DE-LA-CRUZ, <i>et al.</i> , 2019)
7	Softbots Supporting the Operator 4.0 at Smart Factory Environments	(RABELO, ROMERO e ZAMBIASI, 2018)
8	Collaborative Softbots: Enhancing Operational Excellence in Systems of Cyber-Physical Systems	(RABELO, ZAMBIASI e ROMERO, 2019)
9	A critical review of smart manufacturing & Industry 4.0 maturity models: Implications for small and medium-sized enterprises (SMEs)	(MITTAL, KHAN, <i>et al.</i> , 2018)

Fonte: Elaborada pelo autor (2020)

Para concluir o processo de revisão sistemática da literatura, uma última fase contendo um resumo com os pontos principais de cada trabalho selecionado deve ser

elaborada de modo que seja possível realizar uma análise comparativa entre os trabalhos e este próprio texto.

HYBRID TEAMS: FLEXIBLE COLLABORATION BETWEEN HUMANS, ROBOTS AND VIRTUAL AGENTS

(SCHWARTZ, ZINNIKUS, *et al.*, 2016) Propuseram um ambiente híbrido entre humanos e máquina aonde o humano é o elemento central. Este ambiente ciber-físico é criado com a utilização realidade aumentada, em que sensores detectam as ações e posições dos humanos, assim como os dados dos robôs e máquinas.

Sua arquitetura modular foi implementada em quatro módulos distintos, com responsabilidades bem definidas. Sendo que a utilização de *softbots* como interface homem-máquina parte do problema global abordado.

Na arquitetura proposta, *softbots* são usados para aumentar a colaboração entre humanos, equipamentos e software. Os *softbots* são voltados a tarefas específicas como coleta de dados de sensores, rastreamento de movimento, entre outros. Como exemplo complexo de *softbot* o artigo cita um módulo que identifica e aprende as matérias-primas associadas entre si e no futuro pergunta se alguma outra matéria-prima é necessária quando uma é escolhida pelo operário.

Existe um módulo de planejamento de diálogos que é utilizado para que o sistema possa “falar” com o humano de forma natural. Este planejador utiliza um módulo de Text-To-Speech (TTS) para a conversão e ontologias para elaboração semântica utilizando o padrão OWL.

A organização das tarefas é feita através de um *Blackboard*, em que as tarefas são organizadas em filas. Essas tarefas são visíveis a qualquer agente no sistema e são disponibilizadas para os humanos através de representação gráfica.

Um *Middleware* baseado em eventos foi desenvolvido para intermediar a comunicação entre os agentes que compõem o ambiente híbrido. Sua implementação é baseada em *Apache's cross-language services-framework*.

A validação da prova de conceito implementada se deu em ambiente bem definido e controlado.

APPLYING CHATBOTS TO THE INTERNET OF THINGS: OPPORTUNITIES AND ARCHITECTURAL ELEMENTS.

(KAR e HALDAR, 2016) propuseram um conceito de arquitetura de alto nível baseado em nuvem para a comunicação entre *chatbots* e dispositivos IoT. O trabalho levanta as diferentes classificações entre os *chatbots*, assim como os dispositivos IoTs, buscando discutir o cenário atual de ambas tecnologias e apontar os desafios desses dispositivos.

Os desafios dos IoTs são classificados pelos autores em “Desafios centrados em tecnologia” e “Desafios centrados em humanos”. Esses desafios são vistos como oportunidades para os *chatbots*, em especial, *chatbots* baseados em IA com capacidade de aprendizado.

A utilização de *chatbots* em conjunto com IoTs é justificada pela facilidade de desenvolvimento, facilidade de *deployment* e a padronização de protocolos WEB. Estas três características estão presentes em ambas as tecnologias.

Como conceito de arquitetura, o trabalho apresenta uma Plataforma IoT conectada aos dispositivos IoT ou seus integradores, que juntamente com uma API de comunicação com a Plataforma de *chatbot*, possui a função de integrar, unificar e sobrepujar os problemas de interoperabilidade entre os dispositivos. Já a Plataforma de *chatbot* é responsável pelo motor do *chatbot* e pela integração com aplicação mobile.

O trabalho possui uma abordagem conceitual e exploratória do problema, sendo assim, não faz nenhuma implementação ou prova de conceito do tema.

THE SOCIAL FACTORY: CONNECTING PEOPLE, MACHINES AND DATA IN MANUFACTURING FOR CONTEXT-AWARE EXCEPTION ESCALATION

(KASSNER, HIRMER, *et al.*, 2017) propuseram o conceito de “*Social Factory*” com a utilização de rede social atrelada a informações do chão-de-fábrica. De forma simplificada, o conceito consiste na criação de perfis de usuário em rede social para as máquinas e utilizar *chatbots* para a troca de mensagens entre humanos e máquinas.

A arquitetura proposta possui uma base de conhecimento. Esta base se comunica com um *Middleware* integrador. Este *Middleware* possui analisador de texto, tratador de exceções, integrador com máquinas e ainda engloba as funcionalidades de *chatbot*. O *Middleware* se comunica com o *front-end* da rede social ou diretamente com o usuário em caso de exceção.

Foi utilizado MySQL para a implementação da base de conhecimento e utilizado Java para a programação. Toda a comunicação é baseada em troca de mensagens implementada com o padrão REST.

Para validação da arquitetura, foi implementada a interação com uma única máquina em ambiente simulado, levando em conta o tratamento de exceções.

Como trabalhos futuros os autores esperam aumentar as capacidades de julgamento automático e fazer testes no mundo real.

REAL WORLD SMART CHATBOT FOR CUSTOMER CARE USING A SOFTWARE AS A SERVICE (SAAS) ARCHITECTURE

(D'SILVA, THAKARE, *et al.*, 2017) Desenvolveram um *softbot* voltado para a verificação de mensagens em redes sociais, classificação dessas mensagens quanto a sua necessidade de ação a ser tomada e automatização de respostas para início de diálogos com clientes.

A arquitetura proposta utiliza a nuvem pública AWS e consiste em cinco componentes principais: servidor Ejabberd, *gateway* API, *chatbot*, lambda e ML lambda. O servidor Ejabberd é um servidor de mensagens instantâneas escrito em erlang. O *gateway* API da Amazon é um serviço que facilita a criação, publicação, manutenção, monitoramento e segurança das APIs para os desenvolvedores. O lambda é uma plataforma voltada a eventos capaz de executar códigos e escalar automaticamente. O *chatbot* é uma aplicação com capacidade de dialogar com humanos. E o ML lambda é o componente de aprendizado de máquina.

O *chatbot* utiliza serviços inteligentes para detecção de padrões como o LUIS (*Language Understanding Intelligent Service*). Não é detalhada a técnica exata de aprendizado de máquina utilizada no módulo ML lambda.

A implementação foi validada utilizando um ambiente simulado e controlado onde usuário seguia o comportamento esperado do sistema criando uma sala inicial.

Os autores concluem com os objetivos propostos alcançados, isto é, com a arquitetura capaz de auxiliar o atendimento ao consumidor via redes sociais se comportando da forma esperada.

SMART OPERATORS IN INDUSTRY 4.0: A HUMAN-CENTERED APPROACH TO ENHANCE OPERATORS' CAPABILITIES AND COMPETENCIES WITHIN THE NEW SMART FACTORY CONTEXT

(LONGO, NICOLETTI e PADOVANO, 2017) propuseram um *framework* para suporte à interação entre humanos e máquinas usando um ambiente virtual. Combina realidade aumentada com um *chatbot* para ampliar as percepções do operador melhorar a curva de aprendizado em treinamentos de novos operários. O *chatbot* proposto possui capacidades informativas respondendo perguntas sobre equipamentos, tarefas e procedimentos. Além de *chatbot*, o *framework* utiliza realidade aumentada e virtualização para uma maior imersão do operador.

O trabalho faz um estudo detalhado das curvas de aprendizado e tempo de *setup*, comparando os métodos tradicionais de treinamento com a utilização do *framework* proposto. Esta utilização se dá com intermédio de *gadgets* para visualização de ambientes virtuais e realidade aumentada.

O estudo apresenta um ganho na curva de aprendizado dos operários treinados com o *framework* proposto além de melhorias de performance em suas tarefas. Parte do sucesso é atribuído ao suporte efetivo proporcionado pelo assistente pessoal digital implementado.

CHATPY: CONVERSATIONAL AGENT FOR SMES

(PÉREZ, DE-LA-CRUZ, *et al.*, 2019) propuseram uma arquitetura para *chatbots* voltada ao setor comercial de pequenas e médias empresas. O processamento de linguagem natural é realizado por redes neurais recorrentes por sua eficiência.

A implementação se fez com a utilização de PSP (*Personal Software Process*), que se assemelha ao CMM, porém voltado a pessoas.

Diversas plataformas para criação de *chatbot* foram comparadas e o Dialogflow foi escolhida como a plataforma mais adequada às necessidades da pesquisa. O Dialogflow foi conectado ao Facebook Messenger para interação com o usuário. Os webhooks para funcionalidades personalizadas foram implementados em Node e hospedados no Heroku. Já as APIs de terceiros em FireBase.

A validação envolveu testes controlados simulando cenários reais de pequenas e médias empresas. Através da validação o modelo se mostrou promissor e atendendo aos objetivos da pesquisa.

O artigo considerou o aspecto limitado de pequenas e médias empresas, mas o seu foco foi o setor comercial e não de produção, além disso a inteligência da arquitetura é utilizada apenas para reconhecimento de linguagem natural.

SOFTBOTS SUPPORTING THE OPERATOR 4.0 AT SMART FACTORY ENVIRONMENTS

(RABELO, ROMERO e ZAMBIASI, 2018) Investigam como *softbots* podem ajudar operadores na interação homem-máquina, utilizando o conceito de Operador 4.0 como abordagem.

Utiliza a plataforma de referência ARISA, que é uma implementação da arquitetura de referência ARISA, para a implementação de uma prova de conceito do tema. A arquitetura da ARISA é organizada no estilo SOA.

Através de simulação de situações reais, foram estabelecidos três casos de usos envolvendo o auxílio a realização de ordens de compra da cadeia de suprimentos de forma automática. O *softbot* implementado permite a comunicação através de interface própria utilizando Gtalk ou através de Twitter e gera relatórios de forma proativa.

O trabalho explica que apesar de simples, a implementação é bem representativa do ponto de vista técnico. Além disso, os autores cometam sobre melhorias em análises e integrações para trabalhos futuros.

COLLABORATIVE SOFTBOTS: ENHANCING OPERATIONAL EXCELLENCE IN SYSTEMS OF CYBER-PHYSICAL SYSTEMS

(RABELO, ZAMBIASI e ROMERO, 2019) propôs a utilização dos fundamentos de Redes Colaborativas para suporte a *softbots* colaborativos em cenários de pequenas e médias empresas. *Softbots* colaborativos em sistemas ciberfísicos são frutos de um processo racional que envolve não só a comunicação direta entre *softbots*, como também a colaboração com sistemas externos a fim de atingir objetivos.

A plataforma Arisa Nest foi utilizada na implementação dos casos de uso para testes juntamente com uma planta para fins acadêmicos da FESTO disponível no laboratório dos autores.

Os casos de uso selecionados englobam a comunicação de *softbots* especializados em tarefas diferentes, interação por voz e comportamento proativo dos *softbots*. Embora os casos de uso tenham sido escolhidos de forma a validar os objetivos propostos, os testes foram realizados em ambiente simulado e controlado.

O artigo não utiliza modelos de maturidade e nem integração com sistemas externos, porém sua arquitetura permita a inclusão de um *softbot* colaborador para cada um dos itens mencionados.

A CRITICAL REVIEW OF SMART MANUFACTURING & INDUSTRY 4.0 MATURITY MODELS: IMPLICATIONS FOR SMALL AND MEDIUM-SIZED ENTERPRISES (SMES)

(MITTAL, KHAN, *et al.*, 2018) Faz uma revisão dos modelos de maturidade disponíveis para a Indústria 4.0 e *Smart Industry* aplicados a PMEs. Demonstra as diferenças de mentalidade entre PMEs e Grandes Empresas destacando as deficiências dos modelos quando usados em empresas menores. Não apresentar um “nível 0” de conectividade básica e não considerar o grande esforço necessário para sair deste nível são algumas dessas deficiências. Além disso o artigo explica que é melhor que PMEs criem seus próprios modelos de maturidade, a fim de capturarem

melhor as suas realidades e com isso reduzir os riscos de transição e mudança de cultura.

Poucos trabalhos foram encontrados na literatura combinando *softbots* e Indústria 4.0 com foco no PM. (SCHWARTZ, 2016) propuseram *softbots* para apoiar equipes híbridas para aumentar a colaboração entre humanos, equipamentos e software. (KAR, 2016) propôs uma arquitetura de sistema baseada em nuvem para *softbots* para lidar com a comunicação entre humanos e ambientes IIoT. (KASSNER, 2017) propuseram uma arquitetura geral para os *softbots* interagirem com uma única máquina para ilustrar seus benefícios em fábricas inteligentes. (D'SILVA, THAKARE, *et al.*, 2017) propuseram uma arquitetura para *chatbot* baseada em nuvem, porém atrelada as funcionalidades disponíveis na AWS. (LONGO, 2017) implementaram uma estrutura para apoiar a interação de humanos com equipamentos físicos e seus gêmeos digitais. (PÉREZ, DE-LA-CRUZ, *et al.*, 2019) propuseram uma arquitetura em nuvem para suporte comercial de PMEs.

(RABELO, ROMERO e ZAMBIASI, 2018) (RABELO, ZAMBIASI e ROMERO, 2019), implementaram cenários em que um *softbot* ajudou os operadores de máquinas em algumas tarefas por meio de uma interação de alto nível e realizada por voz. Na segunda etapa deste trabalho, eles implementaram um cenário de *softbots* colaborativos em cima de um grupo de sistemas ciberfísicos para melhorar a excelência da operação em uma loja.

Baseado nos trabalhos citados, pode-se evidenciar que a abordagem de interface com o usuário através de um *softbot* é eficaz e atende as necessidades desse trabalho. Em termos de arquitetura, a criação de módulos bem definidos e utilização da nuvem são destaques. Quase sempre é presente a necessidade de criação de um software intermediário, desacoplando as soluções e servindo como um núcleo de processamento. A adaptação de modelos de maturidade, tornando-os mais simples e direcionados, geram um bom engajamento e rápida adaptação. Contudo, é comum observar que embora diferentes, esses modelos possuem padrões de construção comuns. Por fim, a análise dos trabalhos evidenciou que as diferenças de mentalidades entre as grandes empresas e as PMEs faz com que modelos e softwares desenvolvidos voltados as grandes empresas, sejam difíceis de serem usados em

PMEs e necessitem de alterações para se adequarem à realidade das empresas menores.

Neste capítulo foram apresentados os conceitos principais para a construção da proposta deste trabalho, assim como o estado da arte através de trabalhos relacionados ao tema. Apesar das importantes contribuições desses trabalhos, não foi encontrado nenhum trabalho que possua todo o escopo deste trabalho, ou seja, usando *softbots* para o suporte de gerente de produção, combinando “*bate-papo*” conforme a situação, informações e alarmes em tempo real, o uso de níveis de maturidade para análise sobre o *status* de chão de fábrica das empresas, e algumas análises de negócios para orientar os gerentes em seus processos de tomada de decisão contra problemas identificados.

4 ARQUITETURA PROPOSTA

Este capítulo apresenta detalhes da arquitetura proposta para responder à pergunta de pesquisa e atingir as contribuições pretendidas.

Para a definição de uma arquitetura alvo, foi realizada uma análise das demandas de utilização do sistema de MES pelos clientes, assim como do nível de aprofundamento da informação alcançado na maioria dos atendimentos realizados por “gerentes de produção remotos”.

Além da análise dos atendimentos realizados, foram analisadas as APIs prontas para coleta dos dados, o nível de maturidade médio das possíveis empresas clientes e a forma de comunicação predileta dos clientes.

Essas análises foram somadas ao conhecimento e experiência dos gerentes de produção remotos, juntamente com as estratégias de produtos e negócios da provedora do sistema MES em nuvem para que com isso fossem levantados os requisitos do projeto.

4.1 CENÁRIO PRETENDIDO

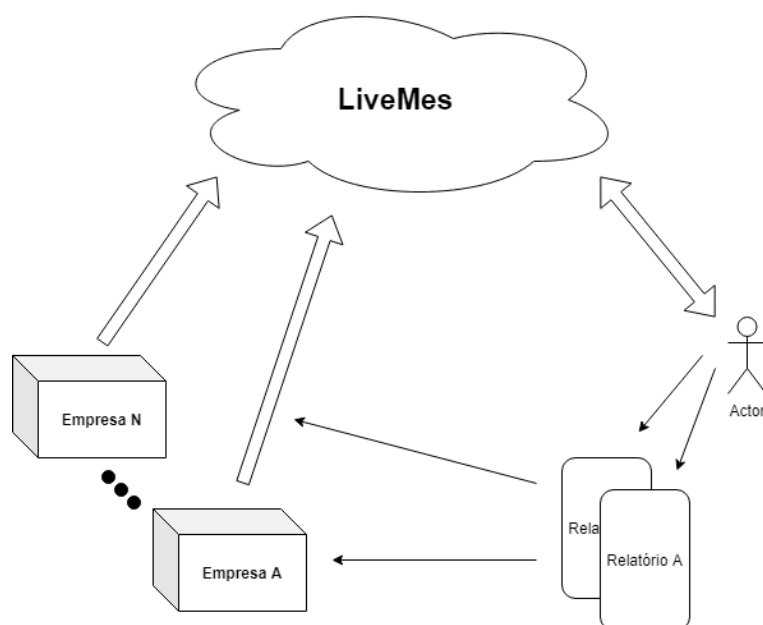
Atualmente, o serviço de gerente de produção é oferecido de forma clássica e não otimizada. Um profissional qualificado é designado para buscar por todo o sistema por possíveis problemas, desvios de comportamento, causas raízes de problemas e oportunidades de melhorias. Todo esse processo leva horas e deve ser feito para cada estação de trabalho monitorada. Essa é a forma como a análise da produção é feita atualmente nas indústrias, seja por um profissional contratado pela empresa cliente, ou por um profissional terceirizado como um serviço (NEELY, 2008).

A principal vantagem de uma pessoa realizar esse trabalho é o contato direto com o cliente de forma pessoal. Como principal desvantagem, temos o tempo necessário para as análises e interações com clientes. Já que este tipo de trabalho requer conhecimentos especializados e específicos, existe a problemática da contratação de profissionais menos capacitados para suprir a demanda.

A Figura 5, exemplifica o modelo atual de trabalho. As empresas cliente, enviam os dados de produção para o sistema de MES na nuvem através de um coletor

devidamente configurado. Estes coletores geram dados com informações simples como a corrente elétrica da máquina ou contagem de peças com auxílio de um sensor de presença, possibilitando assim que mesmo equipamentos antigos possam ser integrados. Os dados disponíveis no sistema são acessados pelo profissional responsável pela análise através do portal web do sistema. O profissional precisa conferir visualmente todos os indicadores, agrupar as informações de acordo com seu conhecimento, de forma instintiva e de forma repetitiva, para cada dia, semana, centro de trabalho, produto, turno ou da maneira que julgar melhor.

Figura 5 – Cenário atual



Fonte: Elaborada pelo autor (2020)

Com o intuito de não eliminar a interação pessoal com o cliente, mas sim, aumentar a capacidade de comunicação com os clientes diminuindo o tempo gasto com análises. Este trabalho pretende criar uma camada de aquisição dos dados e pré-processamento para o suporte ao profissional.

Este pré-processamento é responsável por coletar todos os dados necessários para análise de um período especificado e fazer um levantamento dos problemas encontrados e suas prováveis causas. Esse levantamento está ordenado utilizando-se de critérios estabelecidos pelos profissionais em acordo com os

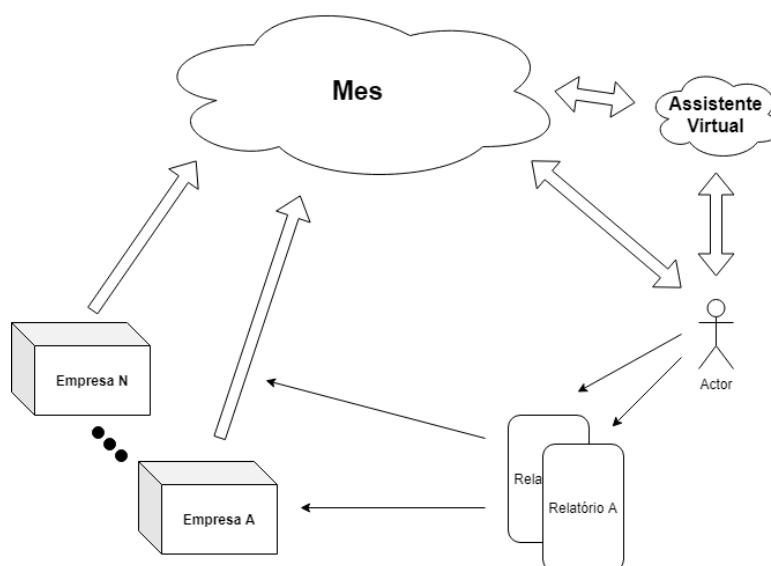
encontrados na literatura, de forma a classificar o grau de importância das ocorrências.

Da mesma forma como é feito no cenário atual, os dados são coletados automaticamente e enviados ao sistema de MES na nuvem. Porém é desejado que o profissional não faça a análise através do portal web do sistema e sim que interaja com o assistente virtual da forma como o próprio cliente faria. Usando linguagem natural, este assistente deve relatar a análise gerada conforme solicitado.

Em termos de arquitetura, o *softbot* deve ser capaz de avaliar todos os indicadores preestabelecidos e deve poder ser validado pelos especialistas antes de ser disponibilizado para os clientes de forma direta. Esse *softbot* deve ficar em nuvem e se comunicar utilizando protocolos padrões de troca de mensagens.

Os pontos de atenção do período solicitado são explicitados pelo *softbot*, juntamente com as suas possíveis e principais causas. Essa análise prévia será validada pelo profissional especializado e este profissional irá dar continuidade ao fluxo da informação gerando o seu relatório da mesma forma como no cenário atual.

Figura 6 – Cenário pretendido



Fonte: Elaborada pelo autor (2020)

A Figura 6 exemplifica o modelo comentado com a inclusão do AV intermediando as informações e apresentando ao especialista.

4.2 REQUISITOS

Embora o levantamento de requisitos em Engenharia de Software possa ser extremamente detalhado e minucioso na categorização dos diferentes tipos de requisitos, para efeitos de simplificação, os requisitos listados neste documento foram agrupados apenas em Requisitos Funcionais e Requisitos Não Funcionais.

4.2.1 Requisitos Funcionais

Os requisitos funcionais são os requisitos que adicionam alguma funcionalidade na utilização que seja entendida pelo usuário como uma nova habilidade do sistema (SOMMERVILLE, 2007).

Partindo do exposto na fundamentação teórica, na revisão da literatura e no problema ao qual esta Pesquisa-Ação tenta solucionar, são definidos os requisitos funcionais e não-funcionais apresentados no Quadro 3 e Quadro 4, respectivamente.

Quadro 3 – Requisitos Funcionais

ID	Requisito	Descrição
RF1	Interface através de <i>softbot</i>	O sistema deve disponibilizar as informações por meio de chat de conversação com um bot
RF2	Reconhecimento de linguagem natural	O bot deve ser capaz de reconhecer linguagem natural do usuário
RF3	Capacidade de exibir os dados mais relevantes	O sistema deve ser capaz de reconhecer o grau relevância das informações e exibir as mais relevantes
RF4	Capacidade de alertar sobre problemas no processo produtivo	O sistema deve ser capaz de monitorar a produção de forma proativa alertar sobre anormalidades

RF5	Capacidade de sugerir ações	O sistema deve ser capaz de sugerir diferentes ações para o usuário considerando os dados de produção
RF6	Personalizar a experiência do usuário	O sistema deve ser capaz de armazenar informações e preferências do usuário
RF7	Validar informações do MES	Os dados do MES devem ser avaliados quanto a sua integridade antes de serem usados em alguma análise

Fonte: Elaborada pelo autor (2020)

4.2.2 Requisitos Não Funcionais

Os requisitos não funcionais são os requisitos que não estão diretamente relacionados a adição de uma nova funcionalidade ao sistema, mas descrevem regras e condições que devem ser respeitadas pelo sistema de modo que este seja considerado em seu funcionamento normal e aceito como um protótipo de produto comercial (ALFF, 2018) (SOMMERVILLE, 2007).

O Quadro 4 mostra os requisitos não funcionais da arquitetura proposta e apresenta uma breve descrição de cada requisito.

Quadro 4 – Requisitos Não Funcionais

ID	Requisito	Descrição
RNF1	Facilidade de Aprender	Deve ser possível aprender com treinamento da ordem de minutos
RNF2	Facilidade de Uso	A utilização deve ser rápida e intuitiva. Além disso a documentação de ajuda deve ser acessível e rápida de ser consultada
RNF3	Modularidade	O sistema deve ser construído em módulos para fácil manutenção e inclusão de novas funcionalidades

RNF4	Disponibilidade	O sistema deve estar disponível em mais de 99% do tempo
RNF5	Desempenho	Para comunicação direta o sistema deve ter tempo de resposta inferior a 30 segundos e para comunicações indiretas (ex. envio de e-mail) o sistema deve responder em menos de 5 minutos. O tráfego de dados pela plataforma de interface com o usuário deve ser baixo, evitando conteúdos “pesados” (ex: imagens, áudios, vídeos).
RNF6	Portabilidade	A solução deve ser capaz de ser executada em ambientes Windows, Linux e MacOS, tanto localmente quanto em nuvem
RNF7	Reusabilidade	O sistema deve reutilizar as APIs existentes para aquisição de dados. Caso alguma nova API seja desenvolvida, ela deve ser disponibilizada para utilização por outros sistemas
RNF8	Segurança	Toda a comunicação entre cliente e servidor deve ser criptografada. Além disso, apenas pessoas que tenham sido autenticadas podem ter acesso às informações
RNF9	Escalabilidade	Deve ser possível escalar a solução para que seja utilizada por muitos usuários conforme a demanda exigir

Fonte: Elaborada pelo autor (2020)

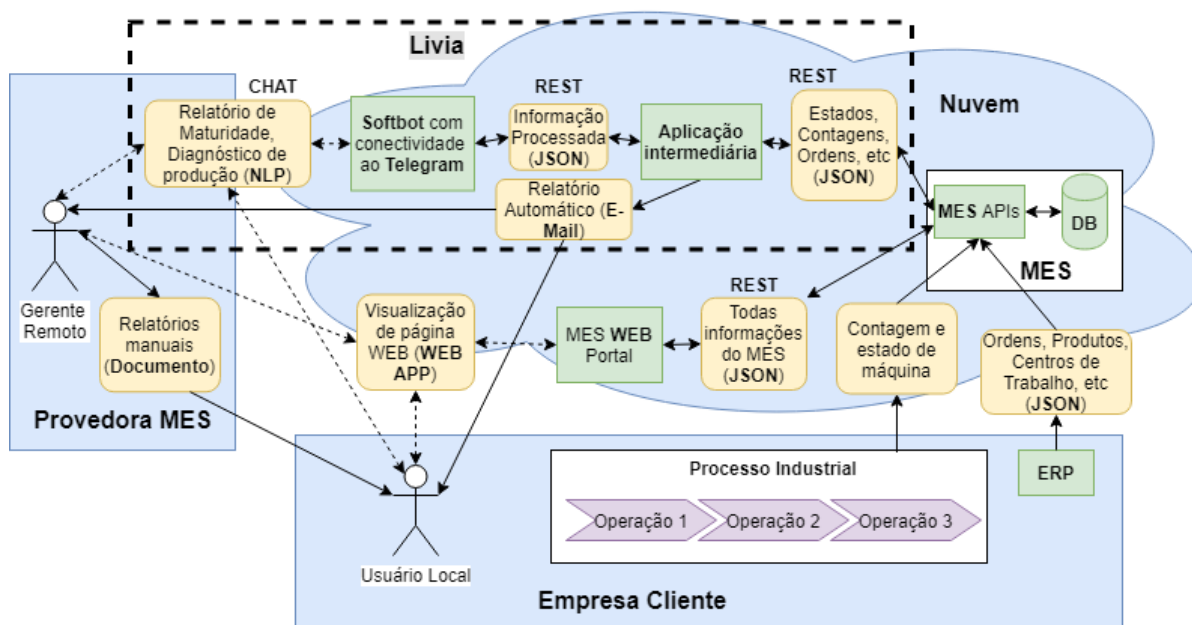
4.3 GERENTE DE PRODUÇÃO COMO SERVIÇO (PMAAS)

A Figura 7 mostra uma visão geral da arquitetura do ambiente PMaaS em que o trabalho desenvolvido é inserido. Sendo o quadrado tracejado o trabalho

desenvolvido nesta dissertação e o restante da imagem o sistema atual já instalado em fábricas que utilizam o software MES de referência (LiveMES). Os quadros em verde representam softwares que podem estar distribuídos em servidores independentes e se comunicam entre si conforme ilustrado. Os quadros amarelos representam as informações enviadas entre esses softwares identificando o formato principal no qual essa informação é transmitida (ex.: JSON, E-mail, relatório). A arquitetura segue o modelo de cinco camadas ISA-95.

Diferentemente de um software de MES instalado localmente em uma indústria, um software de MES em nuvem é capaz de se conectar a várias empresas simultaneamente, porém para efeitos de simplificação a Figura 7 mostra a integração entre os sistemas de apenas uma empresa cliente com apenas um usuário do sistema, assim como apenas um gerente de produção remoto.

Figura 7 – Representação Macro



Fonte: Elaborada pelo autor (2020)

Cada “Empresa Cliente” possui usuários do sistema MES que estão identificados como “Usuário Local”. O processo produtivo da indústria referente as camadas 0, 1 e 2 do modelo ISA-95, está representado por “Processo Industrial” na figura. Os dados, por exemplo contagem de produção e estados de máquinas,

gerados e coletados no processo produtivo por máquinas, controladores locais, redes industriais, sensores, IoT etc, são enviados para o sistema MES através de APIs para comunicação.

O *backend* do sistema MES está representado como “MES” na figura e representa a camada 3 do modelo ISA-95. É composto pelo banco de dados que armazena todas as informações do sistema MES e da aplicação, que por sua vez é composta de diversas APIs. Algumas dessas APIs fornecem dados para o *frontend* do sistema que são disponibilizados para qualquer usuário com as devidas permissões através de um portal web, seja esse usuário um operador na linha de produção, um diretor da fábrica ou até mesmo um gerente de produção remoto acessando os dados de alguma empresa cliente. Com as informações de produção um gerente de produção remoto, identificado na figura como “Gerente Remoto”, pode gerar um relatório de análise de maturidade da empresa de forma manual e enviar para um usuário da empresa cliente.

A camada 4 é representada por “ERP” dentro do escopo da empresa cliente. O ERP pode enviar para o MES informações sobre ordens de produção, produtos, etc, assim como pode receber informações do MES sobre o andamento da execução dessas ordens, entre outras. Como o software MES de referência usado nessa dissertação é voltado para pequenas e médias empresas, a geração de dados de produção das camadas 0, 1 e 2, assim como a geração das informações de negócio da camada 4, podem ser geradas manualmente. Essa possibilidade se dá devido a realidade de algumas empresas que não possuem ERPs ou até mesmo possuem uma baixa conectividade em alguns equipamentos.

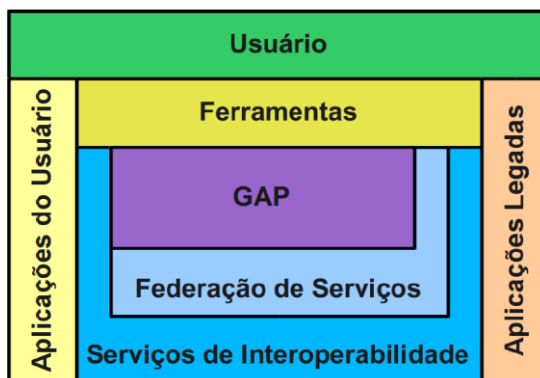
4.4 ARQUITETURA BASE

A arquitetura proposta tem como base a arquitetura de referência para a criação de assistentes pessoais descrita por (ZAMBIASI, 2012). Esta arquitetura utiliza o estilo arquitetural SOA para ser distribuída e independente de tecnologias de implementação. Pontos esse que são objetivos deste trabalho.

A Figura 8 ilustra a arquitetura base e exemplifica a interação entre cada elemento arquitetural. Um ponto importante é que a camada “Usuário” não é

exatamente um elemento da arquitetura, mas sim a visão do *softbot* da perspectiva do usuário, ou seja, dos elementos da arquitetura da qual devem servir como interface entre o usuário e o *softbot* (ZAMBIASI, 2012).

Figura 8 - Arquitetura Base



Fonte: (ZAMBIASI, 2012)

Os elementos arquiteturais descritos na Figura 8 são (ZAMBIASI, 2012):

- **Ferramentas:** são uma composição de IDEs (*Integrated Development Environment*), ambientes/software de configuração ou outras ferramentas que servem para desenvolver ou configurar elementos da arquitetura;
- **GAP:** o Gerenciador de Assistente Pessoal é responsável pela coordenação da execução das ações e suas informações, sua responsabilidade é fazer a coordenação e invocação dos serviços uma vez que determinadas condições sejam satisfeitas, já que o estilo arquitetural SOA é utilizado;
- **Federação de Serviços:** engloba o conjunto de serviços que estão distribuídos na Internet e/ou no repositório local de serviços. Cada serviço possui sua própria funcionalidade e sua forma de implementação em linguagem da escolha do desenvolvedor, mas deve manter o padrão SOA (MacKenzie et al., 2006).

- **Serviços de Interoperabilidade:** é a camada utilizada para que os elementos de interação com o usuário ou outros serviços não compatíveis sejam disponibilizados ao GAP;
- **Aplicações do Usuário:** aplicações que devem ser utilizadas para interagir com o *softbot*. Pode ser uma GUI específica, e-mails, mensagens de Twitter, sistemas de *Instant Messaging*, ou SMS;
- **Aplicações Legadas:** Softwares de desenvolvedores específicos e softwares que podem ser desenvolvidos e acoplados com a criação de novos serviços.

Esta arquitetura permite que o comportamento do *softbot* seja definido pelo subconjunto de serviços selecionados e configurados, além da totalidade de serviços disponíveis e distribuídos na Internet que seguem os padrões abertos de conectividade e comunicação estabelecidos pela arquitetura, ou que utilizem um serviço de interoperabilidade (ZAMBIASI, 2012).

Essa característica da arquitetura permite que o *softbot* possa possuir praticamente qualquer tipo de comportamento, uma vez que a Federação de Serviços abre margem para a implementação de qualquer tipo de funcionalidade e a integração através das interfaces de serviços (ZAMBIASI, 2012).

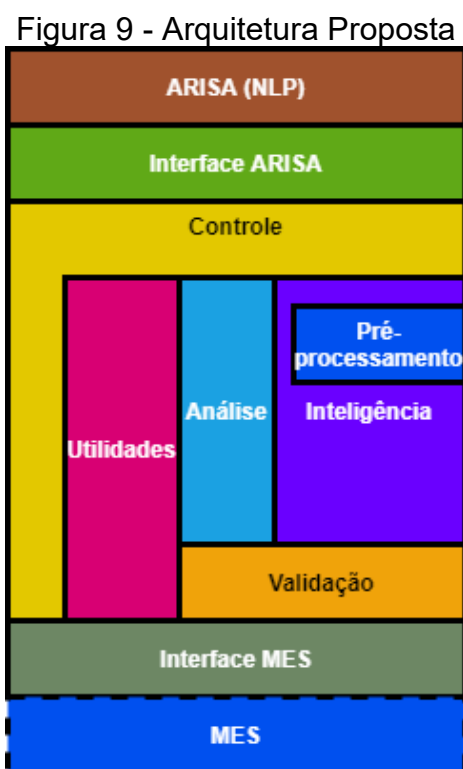
4.5 ARQUITETURA PROPOSTA

A arquitetura proposta no presente trabalho visa expandir as capacidades estabelecidas na arquitetura base. Essa expansão descreve como a integração entre a arquitetura base e um sistema externo, que no caso deste trabalho trata-se de um MES, deve ser realizada.

Mais do que uma expansão da arquitetura base, a arquitetura distribuída proposta se encaixa como um módulo adicional ao software LiveMES de forma que a sua utilização seja totalmente opcional pelos usuários e que não interfira em nenhuma funcionalidade externa a esta solução proposta.

Atendendo aos requisitos funcionais de possuir interface em *chatbot* (RF1), reconhecimento de linguagem natural (RF2) e aos requisitos não-funcionais facilidade

de aprender (RNF1) e facilidade de uso (RNF2) a fim de atender aos objetivos desta pesquisa, a interface com o usuário se dá através de um assistente virtual por meio de chat de conversação. Para este agente, foi dado o nome de *Livia*³, de forma que toda a solução proposta nesta dissertação e representada na figura pelo conteúdo interno ao quadro pontilhado integra a solução *Livia*.



Fonte: Elaborada pelo autor (2021)

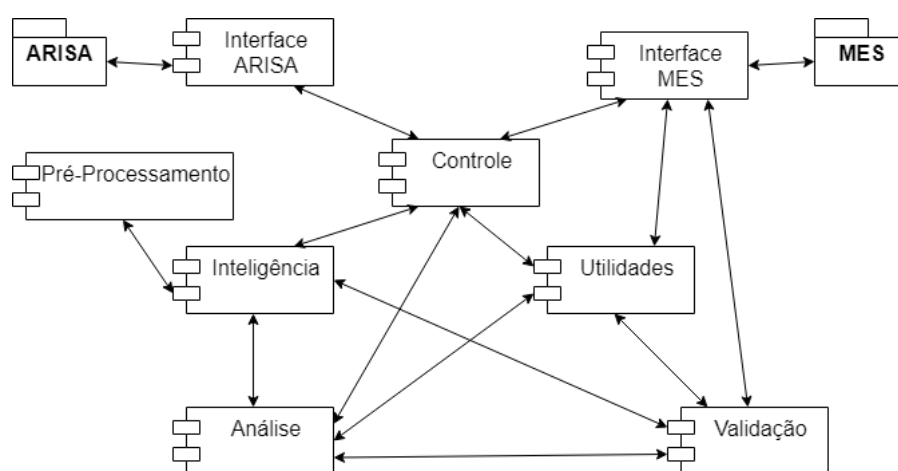
Ainda na figura é mostrado que a arquitetura proposta considera a utilização de no mínimo dois servidores distintos para a implementação de todo o fluxo necessário. Em um servidor é criado um *softbot* capaz de entender requisições descritas utilizando linguagem natural e responder ou agir da forma adequada para atender a essas requisições. Para garantir o atendimento ao requisito não-funcional de reusabilidade (RNF7), uma outra aplicação composta de diversos módulos de processamento foi concebida com o intuito de se conectar diretamente a APIs já

³ [LiveMES](#) + [intelligent analysis](#)

existentes no *backend* do MES, realizar um desacoplamento entre o MES e o bot e retirar a carga de processamento dos dados do servidor dedicado ao *softbot*.

A Figura 9 ilustra a arquitetura proposta por este trabalho, onde as camadas entre as duas interfaces (com as interfaces incluídas), fazem parte da aplicação intermediária. A camada de processamento de linguagem (NLP) é uma instância de *softbot* criada na plataforma Arisa Nest e a camada MES é a aplicação MES que se comunica com esta solução.

Figura 10 - Diagrama de Arquitetura



Fonte: Elaborada pelo autor (2021)

A Figura 10 apresenta a formalização da arquitetura proposta. A arquitetura base é representada pelo pacote Arisa. Os módulos da aplicação intermediária e suas interações presentes na expansão de arquitetura proposta estão detalhados. O MES também é representado como um pacote neste diagrama. A forma de aquisição de dados da planta pelo MES não é representada pois é irrelevante do ponto de vista desta arquitetura.

As requisições geradas pela interação com o usuário são enviadas da Arisa para a Interface Arisa e chegam ao Controle. Neste módulo ficam as lógicas que estabelecem quais são as informações necessárias e como deve ser feito o processamento dessas informações para atender a requisição do usuário. Deste modo, o Controle pode gerar qualquer quantidade de requisições ao MES até que obtenha todos os dados que precisa. Dependendo de qual tenha sido a requisição do

usuário, o fluxo de processamento toma caminhos diferentes. Enviar um e-mail, por exemplo, utiliza o módulo de Utilidades, enquanto fazer a análise completa de uma empresa, utiliza os módulos de Inteligência e Análise várias vezes, mas não utiliza o módulo de Utilidades.

4.6 MÓDULO DE NLP

O módulo de Processamento de Linguagem Natural (NLP) é capaz de transformar os desejos e dúvidas do usuário, expressos por meio de linguagem natural, em gatilhos para ativações de funções e direcionamento do fluxo de conversação. Essa camada da arquitetura proposta ilustra de forma simplificada toda a arquitetura base.

É considerada apenas a comunicação de forma escrita como meio de comunicação entre o usuário e o assistente virtual para fins desta arquitetura, porém é importante ressaltar que dispositivos móveis possuem conversores de fala para escrita de forma nativa. Isto implica que a utilização do assistente virtual pode ser realizada de forma indireta com auxílio destes mecanismos de conversão, mas tais mecanismos estão fora do escopo deste trabalho.

Este módulo é capaz de abstrair as diferentes formas de escrever um mesmo conteúdo, para que independentemente da forma que a requisição seja feita, o entendimento da informação seja o mesmo para um mesmo conteúdo. Em outras palavras o assistente virtual deve ser capaz de compreender gírias e expressões coloquiais assim como linguagem informal até certo ponto. Embora capaz de compreender tipos diferentes de comunicação, o assistente virtual se preocupa com o significado da requisição que foi dada a ele e não com sua forma. Deste modo uma requisição formal e uma requisição informal que possuam o mesmo significado lógico, são interpretadas da mesma forma pelo assistente virtual.

Para a construção dos diálogos utiliza-se o conceito de contexto. Esse conceito tem como base o IML e é largamente utilizado para o desenvolvimento de conversação de acordo com princípios de *conversation design*.

A base de conversação proposta leva em consideração o desenvolvimento de conversação em árvore. A organização dos contextos em forma de árvore possibilita

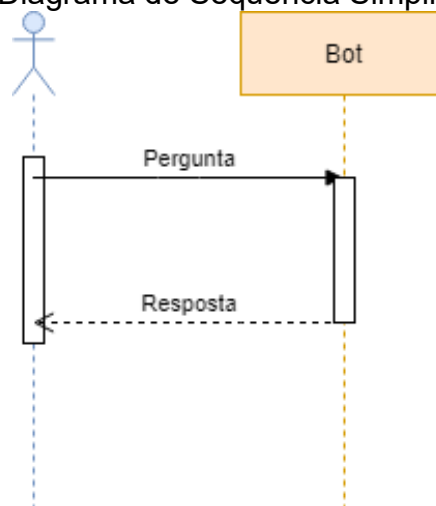
uma fácil e rápida implementação, assim como um entendimento lógico mais natural do fluxo de conversa e de suas variâncias.

O *softbot* é dotado de capacidade de aprendizado possuindo uma base de crenças. Com isso o *softbot* é capaz de personalizar suas respostas para cada usuário assim como personalizar suas preferências. As informações simples relacionadas ao usuário são armazenadas nessa base de crenças e possuem uma baixa latência em seu uso, já as informações referentes a produção, dados de centros de trabalho, etc., são armazenadas em um outro sistema de arquivos que será abordado mais à frente.

A base de crenças difere de uma base de conhecimento utilizada em sistemas especialistas pois não armazena fatos e regras, seu funcionamento é como uma base de dados, armazenando preferências e informações individuais de cada usuário.

Para uma comunicação com o assistente virtual em que não haja necessidade da busca por dados do MES a comunicação entre usuário e *softbot* se dá conforme representado no diagrama de sequência da Figura 11. Figura 11 – Diagrama de Sequência Simplificado.

Figura 11 – Diagrama de Sequência Simplificado



Fonte: Elaborada pelo autor (2020)

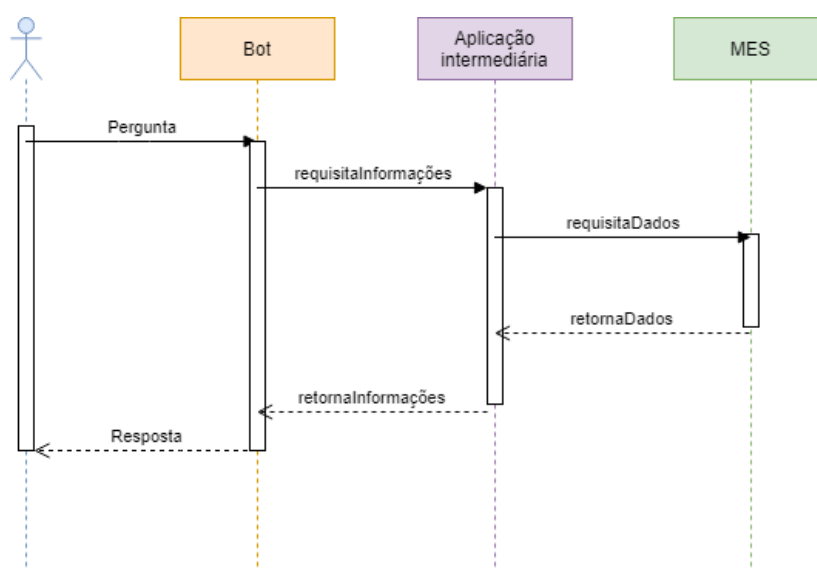
A Figura 11 simplifica a comunicação entre usuário e *softbot* abstraindo a utilização de um aplicativo móvel de comunicação e generalizando qualquer requisição do usuário como uma pergunta. Para cada pergunta do usuário tem apenas uma resposta e todas as perguntas do usuário são respondidas. O diagrama da Figura

11 exemplifica a comunicação usual entre qualquer usuário e qualquer *softbot* quando este não necessita de informações externas para responder ao usuário.

Para ter acesso a informações externas a sua base de crenças, o *softbot* possui capacidade de comunicação com serviços web, sendo esse serviço SOAP ou Rest. Desta forma quando requisitado tem informações externas o *softbot* se conecta a um serviço web disponível recebe as informações requisitadas e as interpreta gerando assim uma resposta em linguagem natural para o usuário.

A fim de respeitar os requisitos de segurança e manter um isolamento entre os dados do MES e dados de conversação do usuário, o acesso às APIs do MES se dá através de uma aplicação intermediária. A arquitetura desta aplicação intermediária representa a principal contribuição deste trabalho e será explicado em detalhes ao longo deste capítulo.

Figura 12 – Diagrama de Sequência Simplificado Completo



Fonte: Elaborada pelo autor (2020)

Para uma comunicação entre usuário e o *softbot*, em que são necessários o uso de dados provenientes do sistema MES, o fluxo da informação ocorre tal como representado no diagrama de sequências da Figura 12.

Para este tipo de comunicação o usuário interage com o *softbot* e esse, por sua vez, realiza uma chamada a uma aplicação web disponível na aplicação

intermediária. Essa aplicação intermediária é responsável por buscar os dados relevantes no MES através de outra chamada web realizada entre a aplicação de intermediária e o MES.

Uma vez que os dados são retornados pelo MES a aplicação intermediária realiza o processamento necessário, condensa essa informação e envia de volta para o *softbot*. O *softbot* interpreta essa informação e gera uma resposta em linguagem natural para ser compreendida pelo usuário. Este é o fluxo natural da utilização desta arquitetura e se aplica há qualquer requisição feita pelo usuário que exija dados provenientes do sistema MES. Contudo, diferentes perguntas ou requisições do usuário podem exigir várias ações desse fluxo representado no diagrama de sequências da figura acima. Para algumas requisições pode ser necessário que o assistente virtual colete mais informações do usuário antes de fazer uma chamada web para adquirir dados do sistema MES, como por exemplo, dados referentes a login. Da mesma forma a aplicação intermediária pode realizar mais chamadas web para as APIs do MES com o objetivo de coletar todos os dados necessários para gerar uma informação que possa ser enviada de volta ao *softbot*. Um diagrama de sequências mais detalhado será discutido no capítulo seguinte de implementação.

Por fim, a arquitetura proposta leva em consideração a utilização de um assistente virtual de forma direta através de uma interface própria ou da utilização de um aplicativo de bate-papo em dispositivo móvel para a comunicação entre o usuário e o *softbot*. Como aplicativo de bate-papo considera-se a utilização de aplicativos populares (WhatsApp, Telegram, etc). A escolha por aplicativos populares como forma de comunicação é motivada pelos requisitos funcionais e requisitos não funcionais.

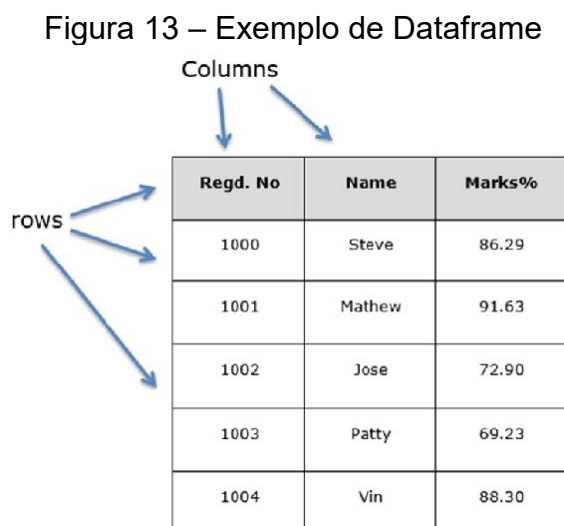
4.7 MÓDULOS DE INTERFACES

Existem dois módulos de interface no sistema, um deles trata da interface entre o assistente virtual e a aplicação intermediária. O segundo deles trata da interface entre aplicação intermediária e o MES. Os módulos de interface garantem a interoperabilidade entre os sistemas de forma que asseguram tanto a interoperabilidade sintática quanto a interoperabilidade semântica.

Para todas as comunicações é adotado o padrão JSON. Esse padrão garante grande parte da interoperabilidade sintática entre os sistemas envolvidos. Contudo a informação pode ser codificada de formas diferentes mesmo com auxílio do padrão de JSON. Sendo assim estes módulos são responsáveis pela conversão dos dados para que se adéquem a um modelo de referência estabelecido para interpretação dos dados.

Este modelo de referência, para escopo desse projeto, nada mais é do que um conjunto de tabelas pré-definidas comumente chamadas de *dataframe* em análise de dados. Essas tabelas de referência formam a estrutura de dados interna a aplicação intermediária e levam em consideração todas as informações necessárias para o atendimento completo dos requisitos levantados para esta arquitetura.

A Figura 13 exemplifica o que é um *dataframe* e como é estruturalmente organizado. A escolha por adotar esse padrão de representação de dados internamente na aplicação intermediária, busca facilitar o processamento de dados necessário para diagnósticos, predições e prescrições.



Fonte: PANDAS (2020)

A análise semântica dos dados enviados pelo MES também é de responsabilidade destes módulos. Essa análise semântica pode se dar de forma direta ou de forma indireta. Como forma direta entende-se a simples interpretação dos dados de uma coluna para a conversão destes dados em uma única outra coluna. Já como

forma indireta entende-se a interpretação de dois ou mais dados provenientes de colunas ou tabelas distintas para a formulação de um único dado compreendido pelo sistema intermediário.

4.8 MÓDULO DE INTELIGÊNCIA

O módulo de inteligência é um conjunto de classes e métodos responsáveis pelas classificações, predições e prescrições necessárias. Todas as tarefas de classificação, predição e prescrição, são padronizadas como *DataFrames* de entrada, um algoritmo de inteligência apropriado e um resultado da execução desse algoritmo como saída. A interpretação dessa saída pode ou não envolver uma nova iteração em um algoritmo de inteligência e fica a cargo do módulo de Análise discutido a seguir.

Figura 14 - Modelo caixa preta



Fonte: Elaborada pelo autor (2021)

A Figura 14 ilustra o modelo caixa preta de inteligência utilizado. Essa abordagem apresenta vantagens em uma primeira implementação, assim como prepara a solução para mudanças nas técnicas de inteligência artificial utilizadas e estudos relacionados em trabalhos futuros.

Em termos práticos, todas as classes e métodos utilizados possuem a mesma interface de chamada. Isso se estende até mesmo para algoritmos usuais sem a implementação de qualquer técnica de inteligência artificial, mas capazes de realizar classificações através de parâmetros pré-fixados.

Uma outra característica dessa abordagem é a possibilidade de decisão da técnica utilizada para computação dos dados ser tomada apenas durante a implementação da solução. Não sendo assim uma decisão arquitetural.

Por esse motivo, mais detalhes sobre técnicas utilizadas são discutidos mais no capítulo 5.6.3.

4.8.1 Pré-processamento

Dependendo do tipo de algoritmo utilizado, algum tipo de pré-processamento pode se fazer necessário para uma melhor performance do algoritmo. Esses pré-processamentos são, em geral, normalizações das *features* de entrada.

A fim de manter a padronização e independência da técnica de inteligência artificial utilizada no processamento dos dados, um método de pré-processamento sempre deve ser invocado antes da invocação do modelo. Em casos em que não seja necessário um pré-processamento, este método simplesmente retorna uma saída igual à entrada.

4.9 INTEGRIDADE DE DADOS

Um grande problema relacionado ao big data e análise de negócio é a consistência dos dados (KWON, LEE e SHIN, 2014). Para garantir o que os dados analisados são corretos em representativos essa arquitetura propõem um módulo de avaliação da integridade dos dados logo após a sua coleta do sistema MES. Desta forma todo dado coletado deve passar por um módulo avaliador que classifica cada dado obtido conforme a sua qualidade e discrimina dados considerados incorretos como sendo inconsistências.

A qualidade do dado é medida baseando-se nas métricas adotadas pelo modelo de maturidade considerado. Essas métricas são passadas como pesos para um classificador e podem ser customizadas para cada centro de trabalho de cada empresa ou usadas com um valor padrão pré-definido, conforme explicado anteriormente.

Essa abordagem modular visa obter vantagens relacionadas à facilidade de implementação e também facilidades relacionadas à substituição de técnicas de classificação. Com isso é possível a criação de um classificador utilizando um

algoritmo comum e simplificado para uma implementação inicial e posteriormente a sua fácil substituição por algoritmos de inteligência artificial.

4.10 ANÁLISE DE DADOS

Uma vez que os dados tenham sido importados, traduzidos, classificados e avaliados quanto a sua integridade é possível realizar análise de dados ou análise de negócio. Uma questão importante que deve ser mencionada é que a decisão de inclusão ou exclusão de dados considerados com problemas fica a cargo do usuário do sistema e não do sistema em si. Antes de efetivamente analisar os dados, o sistema expõe ao usuário o percentual de dados considerados íntegros, ou seja, dados em que não foram encontradas irregularidades de forma automática. Esta decisão binária entre usar ou não usar o volume completo dos dados é de responsabilidade do usuário, que deve considerar que o MES não é capaz de filtrar e tampouco de reconhecer a integridade dos dados que coleta. Isso implica que a utilização de um filtro para que apenas os dados íntegros fossem utilizados pelo assistente virtual, embora gere uma informação mais correta, geraria também uma informação diferente da disponível no MES. Para que os dois sistemas não se contradigam a utilização desse filtro deve ser opcional.

Como todos os dados são processados e classificados quanto a sua integridade, não há nenhum impacto para o sistema independente da decisão tomada pelo usuário.

A análise dos dados se faz de forma incremental considerando-se o modelo de maturidade adotado como referência. Deste modo uma vez feita a escolha do nível de maturidade que se deseja avaliar, cada um dos parâmetros que devem ser avaliados para este determinado nível será então avaliado, até que todos sejam avaliados.

Essa Análise se assemelha a análise realizada pelos especialistas que atualmente realizam a função de gerente de produção remoto. Para isso este módulo usou como referência a bibliografia disponível sobre produção industrial em conformidade com as regras de análise estabelecidas por estes especialistas. A

obtenção dessas regras se deu por meio de diversas rodadas de entrevistas a quatro especialistas que desempenham um papel de gerente de produção remoto.

4.11 MÓDULO DE UTILIDADES

Algumas funções de suporte que não se encaixam em nenhum dos outros módulos são agrupadas em um módulo de utilidades. Este módulo realiza tarefas de inicialização do sistema, ajustes de parâmetros ou algumas conversões de unidade secundárias que não fazem parte do fluxo principal.

A funcionalidade de envio de e-mail também é agrupada a este módulo já que é uma funcionalidade de suporte.

4.12 MODELO DE MATURIDADE

Para auxiliar o gerente de produção remoto essa arquitetura propõem o uso de um modelo de maturidade simplificado voltado à analisar a maturidade individual de cada centro de trabalho de uma empresa, assim como a maturidade desta empresa como um todo. Inspirado no CMMI, esse modelo considera apenas os dados disponíveis no MES para análise.

A utilização do modelo de maturidade simplificado proposto nesta dissertação possui duas finalidades distintas, porém fortemente interligadas. A primeira delas consiste na efetiva avaliação da empresa considerando-se a busca pelo acesso a indústria 4.0. Já a segunda é servir como indicativo auxiliador na tomada de decisão para a geração das sugestões de ações fornecidas pelos gerentes de produção.

- **Avaliação:** Serve como critério avaliativo e qualitativo para estabelecer uma métrica para que se possa criar uma comparação entre a maturidade de um processo de uma determinada empresa a uma referência.
- **Suporte à decisão:** Empresas em diferentes níveis de maturidade podem apresentar problemas com sintomas semelhantes, porém com causas distintas, sendo assim o nível de maturidade de uma empresa

é considerado na decisão da sugestão que melhor se aplicam ao processo em avaliação desta empresa a fim de se obter um melhor resultado para a solução de problemas.

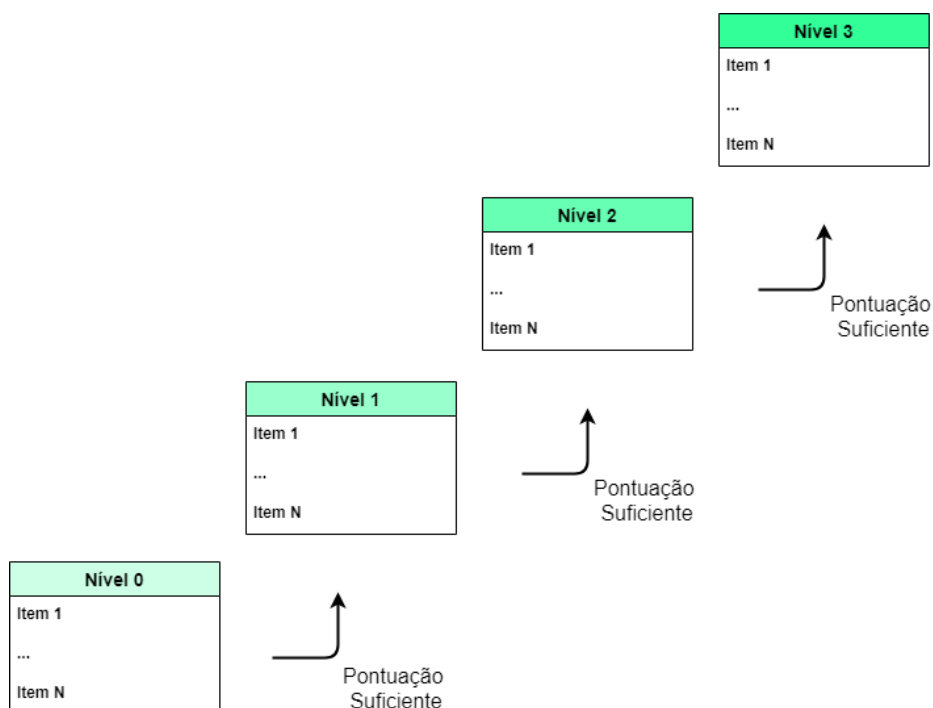
Um outro benefício proporcionado pela adoção desse modelo de maturidade simplificado para avaliação dos processos de uma empresa se dá na sistematização do processo avaliativo. Desta forma garante que os dados sejam analisados do mesmo modo e sob a mesma ótica todos os dias. Isso gera um novo conjunto de dados que contam a história evolutiva da maturidade do processo na empresa.

Modelo de maturidade é apresenta 4 níveis:

- **Nível 0:** neste nível a empresa deve se preocupar com a coleta dos dados garantir disponibilidade dos mecanismos de coleta. É impossível fazer uma boa análise em cima de dados incorretos e não representativos de um processo, por isso este é o nível de maturidade mais simples e inicial de um processo.
- **Nível 1:** uma vez que os dados são coletados de forma correta e constante já é possível ampliar os horizontes da análise de um processo. Mesmo que para este nível os dados já estejam sendo coletados de forma correta isso não implica que os dados estejam corretos. Neste nível de maturidade há um enfoque na correção dos dados mestre cadastrados no MES.
- **Nível 2:** O enfoque dos dois primeiros níveis se dava na aquisição e correção dos dados. O enfoque deste nível passa a ser efetivamente o processo produtivo. Para este nível são relevantes os dados de produção, performance e qualidade.
- **Nível 3:** Este é o último nível no modelo de maturidade. Neste nível busca-se o enfoque da melhoria contínua. Para isso diversos fatores externos ao sistema MES são utilizados. Por esse motivo arquitetura proposta não considera este nível de maturidade como parte integrante desta solução.

A Figura 15 ilustra como se dá a evolução de um processo segundo o modelo proposto.

Figura 15 – Níveis de Maturidade do Modelo Proposto



Fonte: Elaborada pelo autor (2020)

Cada nível de maturidade possui diferentes critérios avaliativos. Para cada um desses critérios é dado uma nota de zero à 5, sendo zero inexistente e 5 perfeito. Para a definição dos limiares de cada nota, são considerados os critérios estabelecidos pelo *World Class Manufacturing (WCM)*, assim como a experiência profissional dos especialistas consultados.

Uma média aritmética simples é feita para obter uma nota final da maturidade. Notas finais acima de 4 por mais de 3 semanas indicam que a empresa já pode começar a ser avaliada em um nível superior no modelo. Da mesma forma, notas abaixo de 3 indicam que a empresa ainda não está consolidada no nível atual e volta a ser avaliada em um nível inferior até que esteja apta a subir novamente. Os períodos

avaliativos são subjetivos e dependem das capacidades de adaptação de cada empresa, como também dos processos em si.

4.13 CASOS DE USO

O presente trabalho considera a utilização de modelos de maturidade como um meio de quantificar a maturidade de uma empresa e assim sugerir ações mais adequadas para o melhoramento contínuo desta empresa. Tal abordagem implica em um grande número de funções relacionadas à busca de dados para cada um dos indicadores presentes no modelo de maturidade.

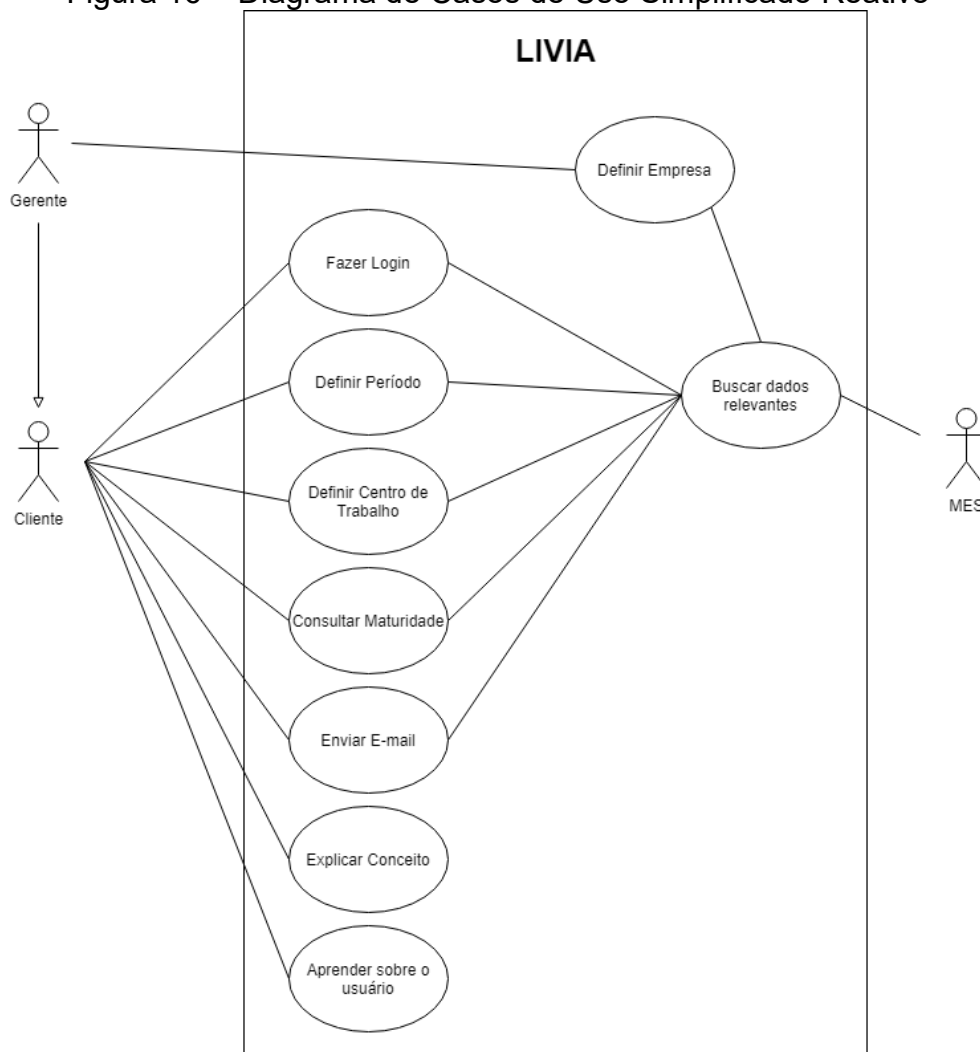
Além de diferentes tipos de busca de dados o sistema também deve ser capaz de validar e avaliar diferentes dados de formas diferentes. Com isso a quantidade de casos de uso para descrever o sistema de forma completa torna-se bastante elevada dificultando a compreensão do sistema. A fim de simplificar a visualização dos possíveis casos de uso, todos os casos de uso relativos a buscas de dados para análise de maturidade de uma empresa foram agrupados em um caso de uso. Da mesma forma houve uma simplificação no acesso aos dados provenientes do MES de forma a agrupar em apenas um caso de uso. Essa simplificação facilita o entendimento do uso do sistema, mas esconde a sua verdadeira complexidade. Fluxogramas foram desenvolvidos para abordar essa complexidade e detalhar parte da lógica que foi simplificada por ora.

4.13.1 Casos de uso do comportamento reativo

Observando-se o comportamento reativo do assistente virtual, ou seja, nas situações em que o assistente virtual responde a requisições iniciadas por um usuário humano, são identificados três atores e o sistema Lívia como um sistema único. O ator primário “cliente” representa um usuário de uma empresa cliente que esteja utilizando o assistente virtual de forma direta a fim de tirar suas próprias conclusões sobre o sistema. O ator “gerente” tem acesso a todas as funcionalidades que um “cliente” possui, além disso um gerente também é capaz de alternar entre as empresas clientes desde que possua permissão de acesso a outras empresas. O ator

secundário MES é o sistema MES disponibilizado pela provedora MES. Neste diagrama o MES é representado como um ator secundário pois responde a requisições e se trata de um sistema externo ao sistema em desenvolvimento.

Figura 16 – Diagrama de Casos de Uso Simplificado Reativo



Fonte: Elaborada pelo autor (2020)

A Figura 16 representa os casos de uso de forma simplificada para uma situação reativa. Conforme mostrado na figura, são identificados os seguintes casos:

- **Definir empresa:** este caso de uso fornece a funcionalidade de definição da empresa que se está analisando. Essa funcionalidade

engloba a busca pelas empresas disponíveis, a validação de acesso do usuário, o registro da escolha selecionadas (RF6).

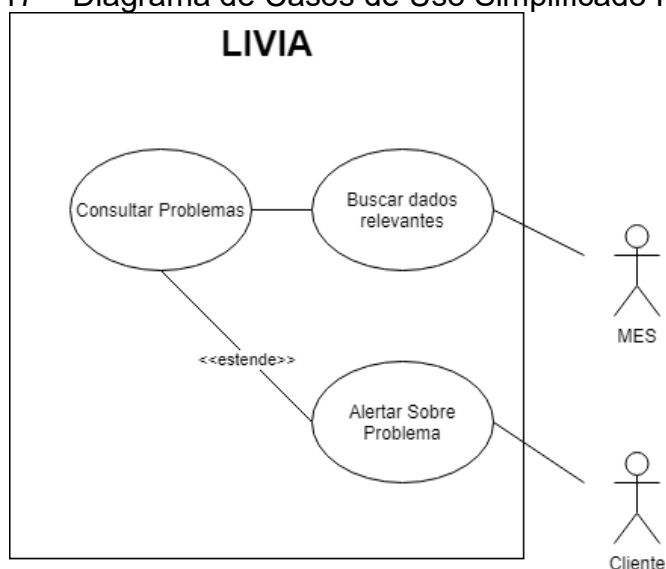
- **Fazer login:** refere-se à funcionalidade de *logar* o usuário ao sistema MES.
- **Definir período:** É a funcionalidade te permite com que o usuário descreva de forma natural o período que tempo que será utilizado para fazer a análise de maturidade desejada (RF6).
- **Definir centro de trabalho:** refere-se à funcionalidade que permite que o usuário selecione o centro de trabalho que deseja analisar baseado nos centros de trabalho disponíveis para aquela empresa (RF6).
- **Consultar maturidade:** este caso de uso engloba diversos casos de uso os para a avaliação da maturidade de uma empresa (RF7).
- **Enviar e-mail:** Este caso de uso representa a funcionalidade que assistente virtual possui de enviar um relatório completo com os dados da maturidade por e-mail para um ou vários usuários (RF4) (RF7).
- **Explicar conceito:** refere-se ao caso de uso em que o usuário pede por explicações de conhecimentos gerais ou conceitos utilizados nas respostas do assistente virtual. Para esta funcionalidade não há necessidade de conexão com o MES.
- **Aprender sobre o usuário:** refere-se à funcionalidade de personalização das respostas para os usuários. Esses conhecimentos são usados para gerar uma aproximação com o usuário em aumentar a agilidade em acessos futuros. São exemplos: nome, idade, empresa, etc. (RF6).
- **Buscar dados relevantes:** estão agrupados aqui, todas as coletas de dados no sistema MES que alimentam a aplicação intermediária para dar prosseguimento ao processamento desses dados (RF3).

Cada caso também engloba todos os diálogos necessários para a realização de todos os processos envolvidos em cada caso (RF2).

4.13.2 Casos de uso do comportamento proativo

Além de possuir o comportamento reativo, a solução proposta também é capaz de reproduzir um comportamento proativo. Em outras palavras o assistente virtual é capaz de iniciar diálogos chamando atenção do usuário para alguma questão específica. O gatilho para o início desse diálogo pode-se dar baseado em eventos ou em tempo, porém ambos os casos são semelhantes do ponto de vista de funcionalidade sob a ótica do usuário. Por isso, ambos os casos são representados através do mesmo diagrama de casos de uso simplificado mostrado na Figura 17. Figura 17 – Diagrama de Casos de Uso Simplificado Proativo.

Figura 17 – Diagrama de Casos de Uso Simplificado Proativo



Fonte: Elaborada pelo autor (2020)

Diferentemente do caso reativo, no caso proativo existem apenas dois atores, sendo que os dois atores são atores secundários. Como o início do caso de uso se dá através de um gatilho no sistema, não existe ator primário para este caso. O ator MES, assim como no caso reativo, representa o sistema externo MES. As funcionalidades compreendidas na arquitetura proposta ativadas de forma proativa não distinguem um usuário normal de um gerente de produção remoto, uma vez que essas funcionalidades sempre estão associadas a uma única empresa. Por isso para o

diagrama de caso de uso representado na figura, apenas é exibido o ator “cliente” que neste caso pode ser também um gerente de produção remoto.

Para este diagrama também foi usada a simplificação em prol da legibilidade e entendimento da solução. O agrupamento “buscar dados relevantes” é uma abstração da mesma forma como foi feita no diagrama de caso de uso para comportamento reativo exibido na Figura 16.

Para o caso proativo foram identificados os seguintes casos de uso simplificados:

- **Consultar problemas:** refere-se ao agrupamento de funcionalidades em que o sistema busca por possíveis problemas de forma proativa ou gera relatórios automáticos de forma proativa. A complexidade pode variar desde a busca e análise de um único parâmetro como o “tempo de máquina parada”, até na busca de todas as informações para todos os níveis de maturidade em todos os centros de trabalho de uma empresa (RF4) (RF7).
- **Alertar sobre o problema:** caso seja necessário o sistema possui a funcionalidade de alertar sobre um problema específico identificado de forma autônoma (RF4).
- **Buscar dados relevantes:** estão agrupados aqui, todas as coletas de dados no sistema MES que alimentam a aplicação intermediária para dar prosseguimento ao processamento desses dados (RF3).

5 PROTÓTIPO IMPLEMENTADO

Neste capítulo é apresentada a implementação do *softbot* construído com o intuito de melhorar uma série de limitações existentes no processo geral de gerenciamento da produção de empresas, conforme explicado na Introdução. O *softbot* é uma instância da especialização da arquitetura de referência já comentada. De um lado, procurou-se utilizar de uma arquitetura de referência já existente voltada para *softbots*, e que atendesse aos requisitos necessários, tais como orientação a serviços, escalabilidade e fácil interoperabilidade. De outro lado, derivou-se uma instância de *softbot* e a especializou para o cenário particular desejado, voltado à realidade da empresa Harbor. Trata-se de um protótipo computacional que, embora já tenha um bom nível de implementação e desenvolvido, já seja direcionado a torná-lo um módulo do produto/software LiveMES da empresa, deve aqui nesta dissertação ser visto como uma prova de conceito, sobre como o uso de *softbots* pode ajudar empresas na gestão da produção em relação a certos aspectos de chão de fábrica.

5.1 ELEMENTOS DE IMPLEMENTAÇÃO

Para a implementação podem-se definir dois grandes grupos de atividades. O primeiro está relacionado a criação de uma instância de *softbot*. O segundo engloba as atividades de amplificação das capacidades do *softbot* através de serviços externos e comunicação com outros sistemas.

Os objetivos específicos, para fins de implementação, podem ser traduzidos em:

- Criação de um fluxo de conversa dinâmico, capaz de compreender mudanças bruscas nos contextos das conversas;
- Tradução do conhecimento dos especialistas para programação. Este conhecimento é tanto de lógica de negócio quanto de regras de decisão;
- Interoperabilidade com serviços existentes já disponibilizados pelo MES;
- Ação visa conceber um protótipo de produto e por isso quase todo o modelo de maturidade precisa ser implementado, para que os usuários consigam

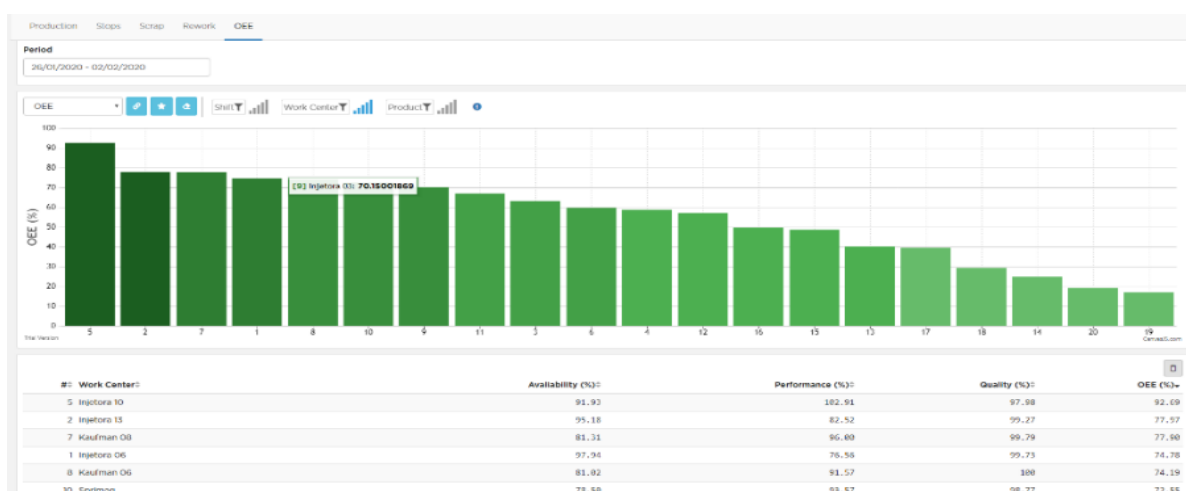
ter uma base avaliativa da dos benefícios gerados por um assistente com capacidades prescritivas.

5.2 LIVEMES

Embora possua muitas vantagens, de forma geral, a implementação de um MES é lenta e exige um grande investimento em hardwares e softwares. Uma forma de solucionar estes problemas é a utilização de SaaS (*Software as a Service*). O LiveMES é um MES no modelo SaaS desenvolvido pela empresa Harbor e atende centenas de empresas no Brasil atualmente.

Para se adequar a PMEs o LiveMES simplifica diversos conceitos do MES e usa dispositivos IoT para coleta de dados. A Figura 18 exemplifica uma tela para análise da eficiência dos equipamentos.

Figura 18 - *Dashboard* geral de OEE



Fonte: LiveMES (2020)

Todo o sistema é hospedado em nuvem e não exige servidor local. Os dispositivos IoT são de baixo custo e podem ser instalados em qualquer equipamento de forma rápida. Com isso, o sistema pode ser totalmente instalado e configurado em um ou dois dias para um cenário com poucos equipamentos conectados (menos do

que 5). Já softwares convencionais MES, costumam levar meses ou anos para a mesma implementação.

5.3 MATURIDADE HARBOR

A empresa Harbor já utiliza o conceito de modelo de maturidade simplificado exposto no capítulo 4.12. A Harbor define o modelo de maturidade simplificado que se denominou de *RA-RE-RI-RO-RU*.

O nível 0 é representado pelo Recurso (RE). Neste nível a conectividade da empresa cliente é avaliada e medida. São avaliados aqui as conexões entre os coletores de dados e o LiveMES. É feita a verificação de cada um dos pontos de coleta automática de cada estação de trabalho. De forma geral, esses pontos são: estado da máquina, contagem de peças boas, contagem de retrabalhos e contagem de sucatas. Mas cada estação pode ou não possuir cada um desses pontos, dependendo de sua automação. Também faz parte deste nível a visibilidade das telas do MES para os usuários e considera a quantidade de acessos a cada uma das telas como critério avaliador. Essa quantidade de acessos foi obtida com a ajuda do Google Analytics já que o LiveMES ainda não possui um registro próprio de acessos por telas.

O nível 1 é representado pelo Rigor (RI). Neste nível os cadastros do sistema são o principal objeto de avaliação. Como exemplo temos as taxas de produção padrão de cada estação, sendo que essas taxas podem ser definidas pelo produto que está sendo produzido ou pelo processo. A quantidade de usuários cadastrados e a frequência de utilização de cada nível de acesso é avaliada aqui a fim de garantir que o MES está sendo usado em toda a fábrica e por todos os níveis hierárquicos. Neste nível também é considerada a utilização de ordens de produção e avaliado o respeito ao planejamento proposto pelas ordens. As implementações relacionadas a esse nível usam dados provenientes de serviços WEB disponíveis no LiveMES.

O nível 2 é representado pela Rotina (RO). Neste nível são avaliados os indicadores de produção básicos que compõem o cálculo do OEE. Os indicadores são performance, disponibilidade e qualidade. Já o OEE é obtido através de multiplicação direta entre esses três indicadores. As implementações relacionadas a esses indicadores usam dados provenientes de serviços WEB disponíveis no LiveMES.

O nível 3 é representado pelo Rumo (RU). Este nível avalia a capacidade da empresa em se manter em melhoria contínua. Muitas informações são indiretas e poucas empresas clientes já estão neste nível de maturidade. Esses fatores levaram a não implementação deste nível de maturidade para este protótipo inicial e está fora do escopo pretendido neste trabalho.

Todos esses níveis têm suas informações compiladas em um gráfico de Radar (RA), formando o assim chamado modelo “RA-RE-RI-RO-RU”, como esclarece a Figura 19.

Figura 19 – Níveis de maturidade do modelo Harbor

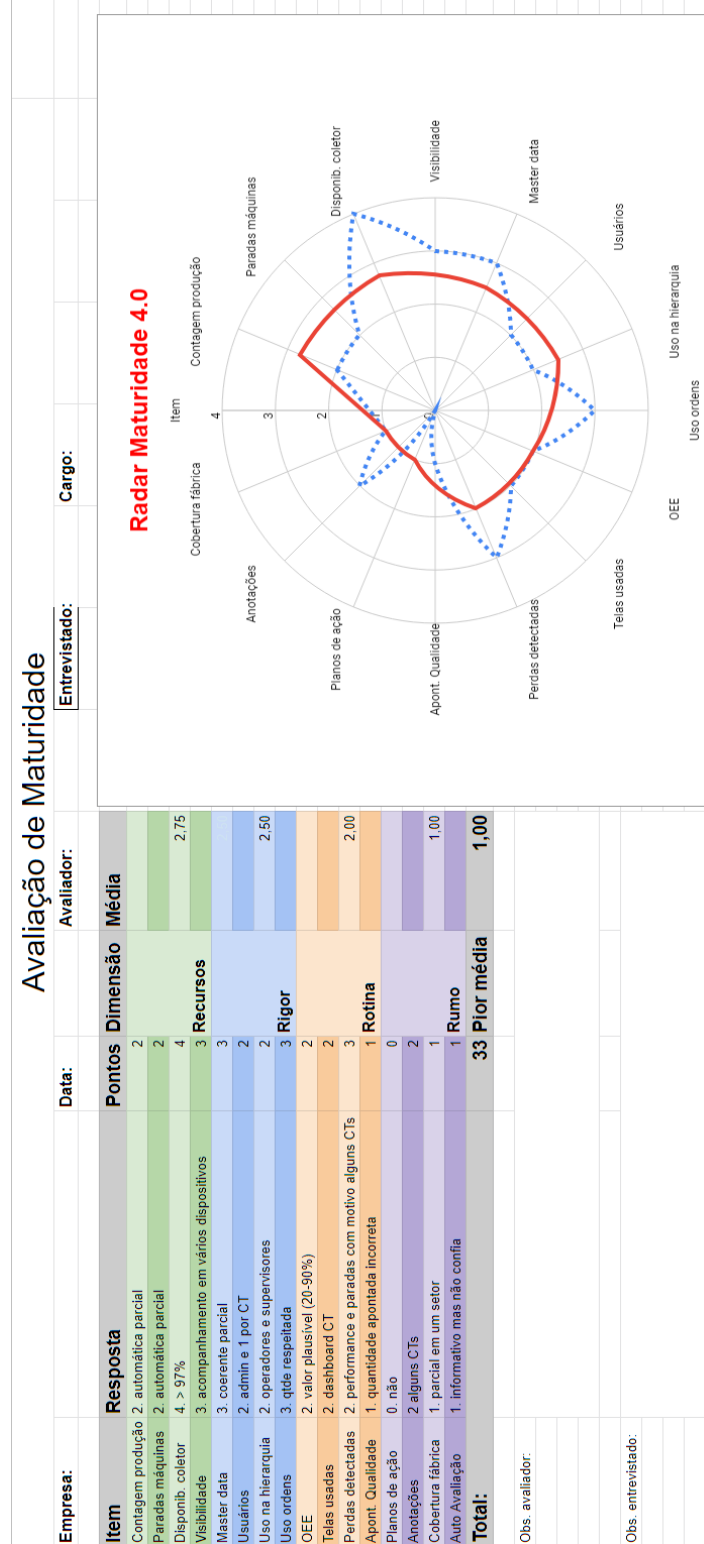
1 Radar	O gráfico de radar define as etapas que devem ser completadas, na sequência
2 Recursos	Os recursos necessários para a coleta automática devem ser disponibilizados, validados e certificados
3 Rigor	Os cadastros devem ser feitos com rigor , assim como a cobrança inicial de uso do sistema por todos
4 Rotina	A rotina da fábrica deve depender e apoiar-se na ferramentas oferecidas
5 Rumo	As informações entregues devem dar o rumo para as ações de melhoria

Fonte: Elaborada pelo autor (2020)

A Figura 15 mostra um exemplo do resultado da aplicação do modelo de maturidade em uma empresa cliente. Essa análise leva em conta os dados de uma semana de produção nesta empresa cliente.

Este trabalho não pretende alimentar essa planilha, mas sim criar a possibilidade de receber esses dados através de bate-papo ou e-mail para que o gerente de produção apenas complemente a análise automatizada com as análises não implementadas neste trabalho.

Figura 20 – Exemplo da interface de maturidade



Fonte: Elaborada pelo autor (2020)

5.4 ESCOLHA DAS TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO

O *softbot* é implementado como uma instância da plataforma Arisa Nest (ZAMBIASI e RABELO, 2020), que se baseia numa arquitetura de referência (ZAMBIASI e RABELO, 2012). Essa plataforma foi escolhida por ser acadêmica e de rápida aprendizagem e derivação. Muito pouco conhecimento é necessário para criar um *softbot* básico e a plataforma proporciona todas as condições necessárias para a integração com outros sistemas. Por ser uma *Plataform-as-a-Service (PaaS)*, os *softbots* derivados já o são com as primitivas essenciais para funcionarem sob o modelo *SaaS*.

A aplicação intermediária foi toda desenvolvida na linguagem *Python*. O *softbot* derivado aceita teoricamente qualquer linguagem de programação. A escolha por essa linguagem de programação se dá devido a sua grande utilização em análises de negócio e inteligência artificial. Por ser amplamente usada nesse campo de conhecimento, a linguagem *Python* apresenta um grande número de bibliotecas, facilitadoras para a implementação ágil e incremental da proposta deste trabalho.

Para as interações entre os serviços WEB que compõe a solução completa deste trabalho, foi utilizado o padrão aberto REST de estilo de arquitetura de software. Este padrão foi implementado com HTTP e usando JSON. Além das suas claras vantagens em sistemas distribuídos, a escolha destes padrões também se deu para manter os padrões já utilizados no sistema LiveMES em sua implementação.

A segurança é implementada usando o mesmo padrão utilizado pelo LiveMES. Este padrão é o OAuth 2.0⁴ e é usado para garantir que o usuário possui acesso e os privilégios necessários para receber as informações requisitadas. Esta validação é feita internamente no sistema LiveMES e a sua implementação foge do escopo pretendido no presente trabalho.

Como artifício facilitador da implementação, a solução foi disponibilizada como um *container* Docker⁵. Implementado em um *container*, a solução pode

4 É o protocolo padrão para autorização na autenticação do protocolo REST. Mais informações estão disponíveis no site <https://oauth.net/2/>

5 É a solução de containerização mais popular do mercado. É amplamente utilizada pelas maiores empresas de software do mundo. Mais informações estão disponíveis em <https://www.docker.com/>

facilmente ser migrada entre ambientes heterogêneos sem necessidades de alteração no código. É importante ressaltar que este *container* engloba apenas a aplicação intermediária já que a instância do assistente virtual fica localizada nos servidores da Arisa Nest.

5.5 LIVIA: INSTÂNCIA DA ARISA NEST

Uma forma de personificar um *softbot* é dando um nome a ele, nesta implementação foi escolhido o nome Livia *para* representar a instância do *softbot* criado. Do ponto de vista do usuário, a solução inteira é única e pode ser entendida como *Livia*. Já do ponto de vista arquitetural, a *Livia* é uma instância de *softbot* implementada na plataforma Arisa Nest e as suas capacidades analíticas foram implementadas na aplicação intermediária.

5.5.1 Arisa nest

De acordo com (ZAMBIASI e RABELO, 2020), a plataforma *Arisa Nest* é uma ferramenta acadêmica desenvolvida como um ambiente baseado em PaaS que permite o gerenciamento de diversos assistentes virtuais e uma complexa estrutura de orquestração e coreografia algoritmos e serviços web.

Esta plataforma permite a criação de instâncias de *softbots* de forma rápida e simples, mais do que isso, esta plataforma disponibiliza recursos e meios para a comunicação com serviços personalizados que adicionam funcionalidades ao *softbot*.

5.5.2 Integração com o Telegram

Para a interface de comunicação móvel com o usuário foi escolhido o aplicativo Telegram. A escolha por este aplicativo se deu por ser um aplicativo popular e gratuito. O aplicativo Telegram está disponível em várias plataformas permitindo que a comunicação seja feita em iOS, Android, Windows e Linux. Além disso a integração entre Telegram e ARISA é facilmente configurável na plataforma de configuração do assistente virtual.

Os detalhes da comunicação entre a ARISA e Telegram ficam a cargo da plataforma de desenvolvimento do assistente virtual e por isso não são explicados em detalhes neste trabalho.

5.5.3 Conversação

A criação de uma conversa ágil e eficiente entre usuário e AV é ponto chave para o sucesso da ação proposta. A conversa precisa conter todas as informações necessárias, mas ao mesmo tempo precisa ser sucinta o suficiente para não se tornar uma conversa cansativa para o usuário de bate-papo. A ARISA NEST organiza a base de conversação em contextos. Cada contexto pode possuir qualquer quantidade de diálogos. Além disso a plataforma também disponibiliza sinônimos, comportamentos e a utilização de *scripts* para uma personalização das capacidades do AV.

5.5.3.1 Contextos

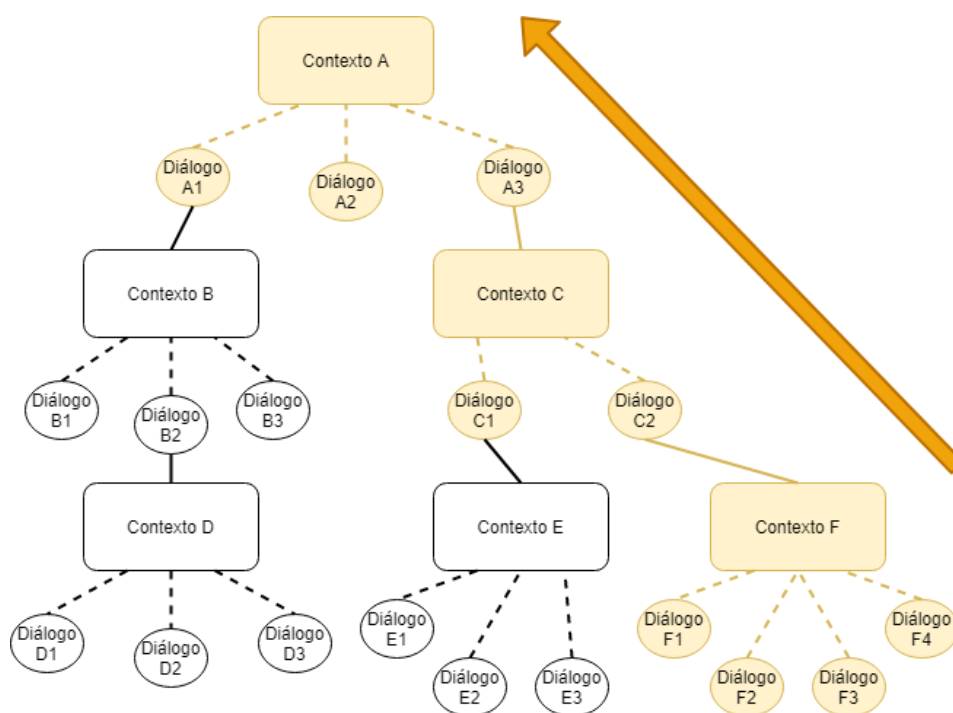
Um contexto é um agrupamento de ideias de assuntos relacionados. Na ARISA os contextos são organizados em árvore. Quando uma nova entrada é dada ao AV, a plataforma busca dentro do contexto atual por todos os diálogos até encontrar um que se encaixe nos padrões do texto de entrada. Caso nenhum diálogo seja compatível, a busca continua no “contexto pai” do contexto atual. Caso nenhum diálogo seja compatível a busca prossegue da mesma forma até chegar no contexto raiz. Se ainda assim não for encontrado nenhum diálogo compatível, a plataforma busca por um diálogo sem nenhum padrão de entrada e assume este como sendo um diálogo válido para esta situação. Este é chamado de diálogo de “fuga” e deve possuir respostas declarativas para alertar que nenhum diálogo conhecido pelo AV foi acionado. São exemplos de respostas: “não entendi” ou “Não consegui entender, se precisar de ajuda para falar comigo é só dizer ajuda”.

Em situações onde um diálogo não é encontrado e o sistema recorre a um diálogo do tipo fuga, a plataforma não altera qual é considerado o contexto atual. Já no caso onde um diálogo é encontrado em um contexto diferente do atual, a

plataforma também não altera qual é o contexto atual, a menos que esse diálogo encontrado leve a conversa especificamente para outro contexto.

A Figura 21 exemplifica a árvore de contextos e o método de busca descrito. Tomando a figura como referência e partindo do contexto F como sendo o atual, caso o diálogo A2 seja encontrado como sendo compatível com os padrões do texto de entrada, a resposta definida no diálogo A2 é retornada, mas o contexto atual permanece sendo o Contexto F. Já caso o diálogo A1 seja encontrado como compatível, então a resposta do diálogo A1 é retornada. Além disto, o contexto atual muda para o Contexto B. Isso ocorre porque o Contexto B é explicitamente definido como próximo contexto na saída do diálogo A1.

Figura 21 – Método de busca dos contextos ARISA



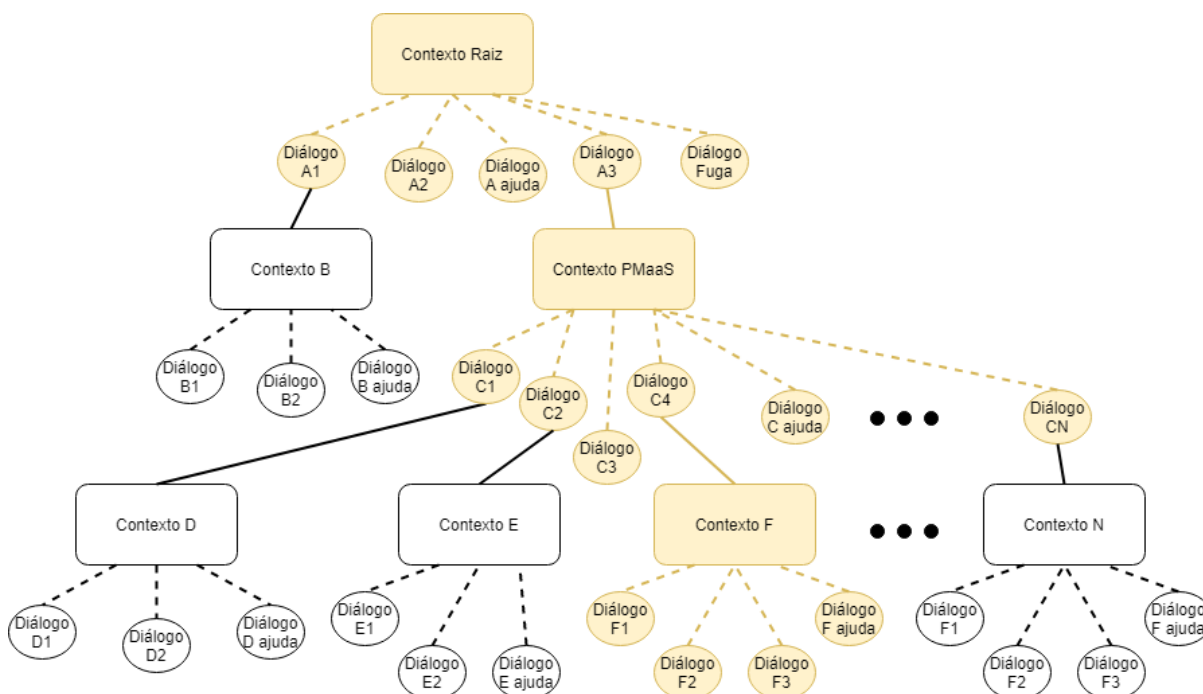
Fonte: Elaborada pelo autor (2020)

Ao longo das etapas de refinamento do projeto, usando as opiniões dos usuários, chegou-se a um modelo de árvore de contextos mais horizontal. Com isso, a implementação é capaz de compreender mudanças abruptas nos contextos e “saltar” entre os itens de cada nível de maturidade de forma direta, sem que seja

necessário que usuário tenha que entender como a árvore de contextos está organizada.

A Figura 22 demonstra a implementação final do modelo de árvore horizontal utilizado na árvore de contextos. O Contexto Raiz é o contexto inicial da conversa. O AV é capaz de manter qualquer conversa não relacionada aos dados provenientes do MES através deste contexto geral e contextos filhos. Os dados informativos disponibilizados pelo AV são implementados no contexto raiz para que a qualquer momento da conversa seja possível esclarecer alguma dúvida conceitual sem que o AV entenda isto como uma mudança de contexto. Como exemplo, considerando o contexto atual como sendo o Contexto F e considerando o diálogo A2 como sendo um diálogo informativo (por exemplo: “O que é OEE?”), o AV é capaz de esclarecer a dúvida do usuário sem prejuízo ao fluxo da conversa. Mais do que isso, essa abordagem permite que o AV mantenha as suas capacidades informativas mesmo sem conexão com o MES.

Figura 22 – Modelagem de contextos Implementada

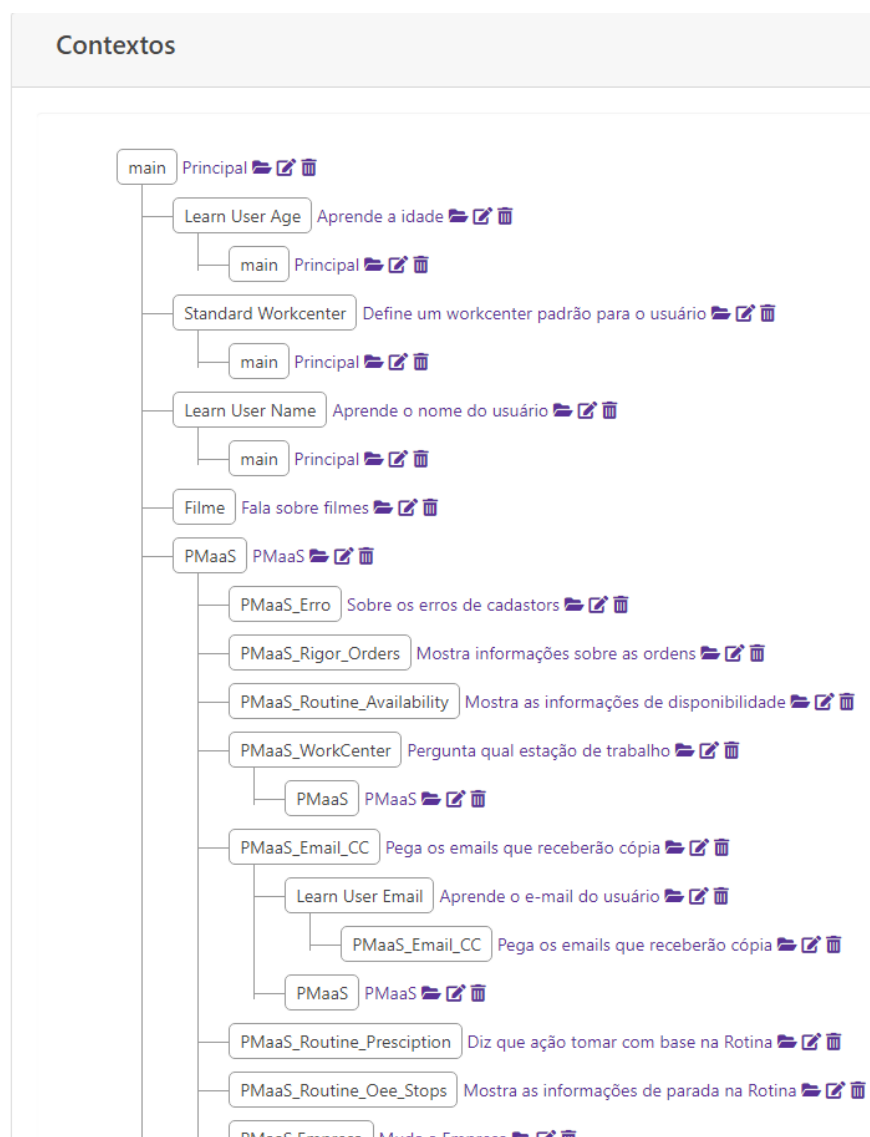


Fonte: Elaborada pelo autor (2020)

Todos os contextos relacionados ao fluxo de conversação, onde é necessária a comunicação com o MES, são implementados abaixo de um contexto genérico para assuntos relacionados a PMaaS. Este contexto é identificado como Contexto PMaaS na figura e possui os diálogos que levam aos contextos específicos de cada item de análise do modelo de maturidade.

Um outro aspecto relevante é que cada contexto possui um diálogo “ajuda”. Isso possibilita que a ajuda fornecida pelo AV seja contextualizada com o que está em foco na conversa.

Figura 23 – Parte da árvore de contextos implementada



Fonte: Elaborada pelo autor (2020)

A Figura 23 apresenta parte da árvore de contextos utilizada na implementação do modelo proposto. Esta é a visualização da árvore de contextos pela interface da plataforma Arisa Nest.

5.5.3.2 Diálogos

Dentro de cada contexto, um ou mais diálogos podem ser configurados conforme desejado. Um diálogo pode conter uma pré-condição para que seja ativado. Também existe a possibilidade de configurar diversas formas de padrões de entrada para um mesmo diálogo. Padrões de entrada são radicais de palavras que devem ser encontrados na sentença de entrada e com isso ativar o diálogo em questão. Essa capacidade é essencial para desenvolvimento de sistemas reais os usuários podem utilizar palavras e construções de frases diferentes para um mesmo significado.

De forma similar, o retorno de cada diálogo pode ser escrito de diversas formas diferentes, tornando assim para conversação com o assistente virtual menos repetitiva. Para o caso de haver mais de uma opção de resposta configurada em um diálogo, a plataforma escolhe uma aleatoriamente dentre as opções disponíveis.

Durante a configuração de um diálogo também é possível estabelecer um novo contexto para qual a conversa será direcionada no caso vê este diálogo ser acionado. Essa funcionalidade que permite a movimentação entre os contextos da árvore.

Por fim a configuração de um diálogo permite até a inclusão de imagens e links como forma de ampliar a capacidade comunicativa do assistente virtual. No presente trabalho estes recursos foram utilizados apenas para o caso de uso onde o usuário requisita explicações de conceitos.

A Figura 24 exibe a tela de interface onde os diálogos são configurados. Nesta figura é possível ver apenas a configuração de um único diálogo de um único contexto, porém também é possível observar todos os outros diálogos utilizados nesse mesmo contexto. O contexto ilustrado é o Contexto PMaaS, usado como exemplo na sessão anterior.

Figura 24 – Interface de criação de diálogos

(312) PMaaS + Adicionar

Afluentes: [main](#) [PMaaS_Date_Range](#) [PMaaS_Email_CC](#) [PMaaS_Empresa](#) [PMaaS_WorkCenter](#)

Ajuda [Continuar Centro de Trabalho → PMaaS_WorkCenter](#) [Continuar Começa nova análise](#) [Continuar Data → PMaaS_Date_Range](#) [Continuar Empresa → PMaaS_Empresa](#) [Continuar Erro → PMaaS_Erro](#)
[Continuar Fim da análise](#) [Continuar Rigor Ordens → PMaaS_Rigor_Orders](#) [Continuar Rigor Taxa de produção](#) [Continuar Rotina Disponibilidade → PMaaS_Routine_Availability](#) [Continuar Rotina OEE → PMaaS_Routine_Oee](#)
[Continuar Rotina Paradas → PMaaS_Routine_Oee_Stops](#)

Descrição (2215) Contexto

Condições

Padrões:

Respostas:

Imagem:

Link Ir para

[Continuar Rotina Performance → PMaaS_Routine_Performance](#) [Continuar Rotina Prescrição](#) [Continuar Rotina Qualidade → PMaaS_Routine_Quality](#) [Direto para Erro → PMaaS_Erro](#)
[Direto para Rigor Ordens → PMaaS_Rigor_Orders](#) [Direto para Rotina Disponibilidade → PMaaS_Routine_Availability](#) [Direto para Rotina OEE → PMaaS_Routine_Oee](#) [Direto para Rotina Paradas → PMaaS_Routine_Oee_Stops](#)
[Direto para Rotina Prescrição → PMaaS_Routine_Prescription](#) [Email → PMaaS_Email_CC](#) [Enviar meu Email → PMaaS_Email_CC](#) [Enviar meu Email - não sabe → Learn User Email](#) [Fim](#) [Lista Centros de trabalho](#)
[Lista Empresas disponíveis](#) [Mudar centro de trabalho → PMaaS_WorkCenter](#) [Mudar data → PMaaS_Date_Range](#) [Mudar empresa → PMaaS_Empresa](#) [Mudar para Centro de Trabalho especi...](#)
[Mudar para Empresa especifica](#) [Não](#) [Qual é o período atual → PMaaS_Date_Range](#)

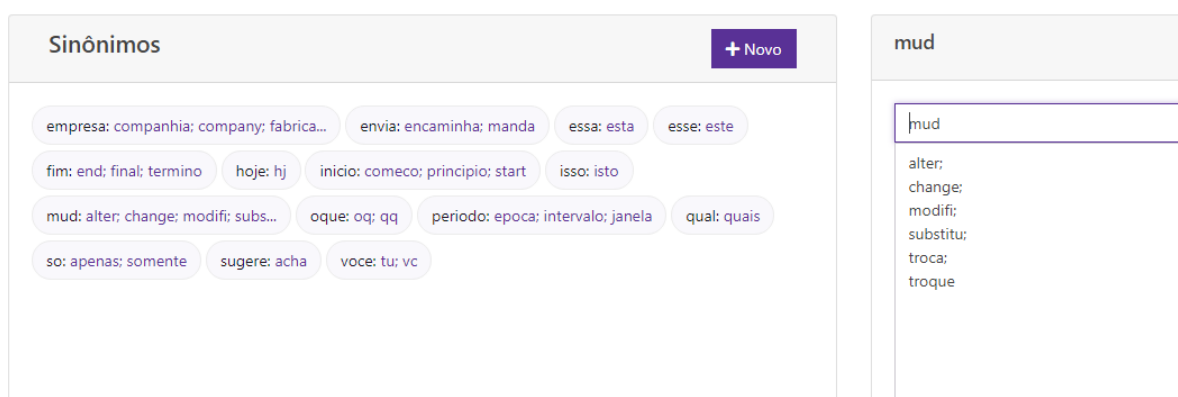
Fonte: Elaborada pelo autor (2020)

Considerando-se o tamanho da árvore de contextos, a quantidade de diálogos por contexto e a quantidade de variações de ativação de cada diálogo, é possível ter uma ideia da complexidade de desenvolvimento de uma conversação adequada considerando arquitetura proposta neste trabalho.

5.5.3.3 Sinônimos

Além as possibilidades de múltiplas definições de ativação de diálogo, a plataforma também proporciona a capacidade de criação de sinônimos. A criação de sinônimos facilita o processo de descrição de diálogos e adiciona uma camada extra de personalização à conversação. A Figura 24 mostra a interface de criação de sinônimos na plataforma ARISA NEST.

Figura 25 – Interface de configuração de sinônimos



Fonte: Elaborada pelo autor (2020)

5.5.4 Crenças

As crenças representam o conhecimento adquirido do AV. Este conhecimento pode ser adquirido através de conversas ou pode ser previamente configurado durante a etapa de implementação. Os dois casos citados são utilização na implementação desta solução.

Crenças Globais representam conhecimento global do AV, que pode ser acessado por qualquer usuário e não dependem de quem está acessando essa informação. Conhecimentos como o nome do AV e sua data de criação são exemplos deste tipo de crença. Além dos casos citados, as crenças globais foram utilizadas para armazenar conhecimento técnico que não é disponibilizado para os usuários de forma direta. Os diferentes IPs dos servidores da aplicação intermediária são exemplos deste tipo de conhecimento. Não há nenhum diálogo que informe ao usuário qual é o IP do servidor da aplicação intermediária, porém existe um diálogo capaz de alterar qual servidor deve ser utilizado. Seja o servidor de desenvolvimento, seja o servidor de QA ou até mesmo o servidor de produção. Isso gera a possibilidade de mudança de servidor em tempo real.

As Crenças Locais são utilizadas para armazenar conhecimentos diretamente associados ao usuário que está conversando com o AV. Conhecimentos como o nome do usuário, a empresa que ele trabalha (ou utiliza como padrão), o período de tempo que foi usado na última análise, entre outros. Além dos casos já citados as crenças locais são utilizadas para armazenar qual é o servidor da aplicação intermediária que se está acessando. Deste modo, a funcionalidade de troca de servidor em tempo real explicada anteriormente pode ser configurada para cada usuário de forma independente. Em outras palavras, é possível o usuário A realizar testes na aplicação intermediária no ambiente de desenvolvimento, enquanto o usuário B utiliza o AV com a aplicação intermediária no ambiente de produção sem nenhuma interferência entre os dois. Uma vez que o usuário A tenha terminado o desenvolvimento, ele pode colocar as modificações no ambiente de QA e avisar ao AV que agora quer utilizar o ambiente de QA. Mesmo sem sair do bate-papo o usuário A passa se conectar a aplicação intermediária já no novo ambiente e o usuário B continua sua utilização normal no ambiente de produção.

Um ponto que deve ser destacado é que a alteração dinâmica de ambientes se refere apenas à aplicação intermediária e que por sua vez se reflete ao ambiente do software MES que fornece as informações. Toda a conversa é configurada em um ambiente único na plataforma ARISA NEST e por isso qualquer alteração, seja no fluxo dos contextos ou nos diálogos, independem de qual servidor está sendo usado para comunicação com a aplicação intermediária.

5.5.5 Interface com serviços externos

A comunicação entre ARISA e aplicação intermediária se dá por meio de chamadas REST. Para a codificação dessas chamadas ou para a adição de qualquer capacidade computacional adicional, a plataforma proporciona a possibilidade de criação de *scripts* que podem ser chamados como parte das respostas dos diálogos. Esses *scripts* utilizam a linguagem Lua e Python.

A linguagem lua foi projetada, implementada e desenvolvida no Brasil por uma equipa na PUC-Rio (Pontifícia Universidade Católica do Rio de Janeiro). Possui

muitas utilizações em aplicações de processamento de imagens e em jogos (IERUSALIMSKY, 2015).

Figura 26 – Exemplo de programação na linguagem Lua

```

1  Authorization = getLocal('LM_User_Token')
2  CompanyId = getLocal('LM_User_CompanyId')
3  WorkCenterId = getLocal('LM_PMaas_WorkcenterId')
4  Server_path = getGlobal('Server_path')
5  apiKey = getGlobal('APIKEY')
6  DEV = getLocal('DEV')
7  if DEV == 'true' then
8      Server_path = getGlobal('DEV_Server_path')
9  end
10
11
12 response = callRest(Server_path, '/WorkCenter/list', 'GET',
13 {
14     'Authorization: ' .. Authorization,
15     'CompanyId: ' .. CompanyId,
16     'apiKey: ' .. apiKey
17 },
18 null, null, null)
19
20 ok, data = pcall(function() return json.decode(response) end)
21
22 if ok then
23     list = string.char(10)
24     if data.restStatus == 200 then
25         for i, workcenter in ipairs(data.WorkCentersList) do
26             if #data.WorkCentersList == i then
27                 list = list .. tostring(workcenter)
28             else
29                 list = list .. tostring(workcenter) .. ',' .. string.char(10)
30             end
31         end
32         return 'Aqui está a lista com todos os centros de trabalho: ' .. list
33     end
34     return 'Eu tentei te trazer a lista completa dos centros de trabalho disponíveis mas tive o erro ' .. data.restStatus ..
35 end
36
37 return 'Eu tentei te trazer a lista completa dos centros de trabalho disponíveis mas infelizmente não consegui me comunica
38

```

Fonte: Elaborada pelo autor (2020)

A Figura 26 exemplifica um *script* utilizando a linguagem lua para a chamada de um serviço WEB disponibilizado pela aplicação intermediária. O serviço em questão retorna a lista de todos os centros de trabalho disponíveis para a escolha em uma dada empresa. Ainda no código exemplo, é possível observar como foi codificada a troca dinâmica de servidores da aplicação intermediária, citada anteriormente.

5.5.6 Comportamento proativo agendado

A ARISA também proporciona a opção de criação de comportamentos proativos, os quais chama apenas de comportamentos. Assim como outros *scripts* da plataforma, os comportamentos são programados utilizando a linguagem lua.

O gatilho de ativação dos comportamentos se dá sempre baseado em tempo, sendo que é possível configurar um intervalo de tempo para que esse comportamento seja ativado ou estabelecer um comportamento contínuo que fica em loop infinito de execução.

Os comportamentos da plataforma são capazes de executar todas as funções presentes nos *scripts* em lua ou python. Com isso, são capazes de realizar chamadas REST de forma direta ou podem chamar outros *scripts* desenvolvidos em Lua, que por sua vez realizam as chamadas REST.

Figura 27 – Exemplo de codificação de comportamento em linguagem Lua

```

1  -----gera mensagens
2  function GenerateMessage (AlertType, friendId, WorkCenterName, AlertValue)
3  msg = 'Acabo de encontrar um problema na estação '.. WorkCenterName
4  if AlertType == 'paradaLonga' then
5  msg = 'Acabo de notar que o centro de trabalho '.. WorkCenterName .. ' está parado a mais de ' .. AlertValue .. ' minutos'
6  end
7  if AlertType == 'paradaPendente' then
8  msg = 'O centro de trabalho '.. WorkCenterName .. ' está com parada pendente a mais de 10 minutos'
9  end
10 return msg
11 end
12 -----substitui a chamada de script
13 function scriptFunc (friendId)
14 Authorization = tostring(getLocal(friendId, 'LM_User_Token'))
15 CompanyId = tostring(getLocal(friendId, 'LM_User_CompanyId'))
16 WorkCenterId = getLocal(friendId, 'LM_PMaaS_WorkcenterId')
17 Server_path = getGlobal('Server_path')
18 DEV = getLocal(friendId, 'DEV')
19 if DEV == 'true' then
20 Server_path = getGlobal('DEV_Server_path')
21 end
22
23 response = callRest(Server_path, '/WorkCenter/analysis', 'GET',
24 {
25   'Authorization: '.. Authorization,
26   'CompanyId: '.. CompanyId,
27   'WorkCenterId: '.. WorkCenterId
28 },
29 null, null, null)
30
31 if response then
32 data = json.decode(response)
33 if data.RestStatus == 200 then
34   setLocal(friendId, 'LM_Alert_PendentRC', tostring(data.Analysis.PendentAlert))
35   if data.Analysis.PendentAlert == 'false' then
36     setLocal(friendId, 'LM_Alert_PendentRC_Previous', 'false')
37   end
38   return 'foi'
39   else
40     return data.RestStatus
41   end
42 else
43   return 'noConnection'
44 end
45 end
46 -----comportamento
47 friends = json.decode(friendList())
48
49 for key, friendId in pairs(friends) do
50 Notifications = getLocal(friendId, 'User_Notifications')
51 if Notifications == 'true' then
52   WorkCenterName = getLocal(friendId, 'LM_PMaaS_WorkCenter')
53   workcenterId = getLocal(friendId, 'LM_PMaaS_WorkcenterId')
54   scriptRet = scriptFunc(friendId)
55   if getLocal(friendId, 'LM_Alert_PendentRC') == 'true' and getLocal(friendId, 'LM_Alert_PendentRC_Previous') ~= 'true' then
56     msg = GenerateMessage ('paradaPendente', friendId, WorkCenterName, 0)
57     setLocal(friendId, 'LM_Alert_PendentRC_Previous', 'true')
58     sendMessage(friendId, msg)
59   end
60 end
61 end
62

```

Fonte: Elaborada pelo autor (2020)

A Figura 26 ilustra como um comportamento é codificado na plataforma ARISA utilizando a linguagem Lua ou Python. Este é o código utilizado para o envio de mensagens de forma proativa através de bate-papo. Esta mensagem avisa aos usuários sobre paradas longas na linha de produção e “aprende” que já informou ao usuário sobre a parada longa em questão. Isto evita que o usuário receba diversas notificações sobre o mesmo problema na linha. O comportamento descrito distingue cada usuário, centro de trabalho e parada, de modo que cada usuário decide se deseja ou não receber notificações e que cada parada longa gere apenas uma notificação por usuário.

A criação de comportamentos baseados em tempo possibilita a simulação de comportamentos baseados em eventos, pois é possível criar uma rotina de constante verificação de determinado parâmetro para detecção de eventuais alterações. A frequência dessa detecção pode ser pequena ou suficiente de forma a simular o acionamento através de um evento externo.

5.6 IMPLEMENTAÇÃO DA APLICAÇÃO INTERMEDIÁRIA

Até este momento foi discutida a implementação relacionada a instância de um novo *softbot* utilizando a plataforma ARISA NEST. No capítulo de arquitetura a Figura 7 dá uma visão macro de onde a aplicação intermediária se localiza no desenho geral da solução, já a Figura 9 ajuda a dar uma visão interna da mesma. Na seção de integração foi demonstrado como é realizada a comunicação com serviços externos do ponto de vista da plataforma, quando é utilizado um script Lua para a realização de chamadas REST e para a decodificação do retorno em JSON desta chamada. Embora tecnicamente a plataforma seja capaz de chamar diretamente os serviços WEB disponibilizados pelo MES, as implicações de se utilizar um sistema real em utilização, geram algumas barreiras que impõem a implementação de uma aplicação intermediária.

O primeiro grande fator é a questão da segurança. A comunicação direta entre a ARISA e o MES, implicaria que as informações de acesso ao MES seriam transmitidas por bate-papo através de vários servidores. Qualquer interceptação desta

comunicação representaria em uma falha grave de segurança para o MES. Mais do que isto, por estar diretamente no bate-papo, qualquer pessoa com acesso ao celular do usuário poderia olhar o histórico de conversa entre usuário e Livia e literalmente ler as informações de usuário e senha para acesso ao MES.

Outro fator relevante para a criação da aplicação intermediária é consequência direta do objetivo desta dissertação. A ação proposta prevê análise prescritiva, entre outras, possuindo um certo grau de inteligência. Como já foi visto, a plataforma ARISA utiliza a linguagem Lua para a criação de *scripts* e embora esta linguagem possua muitas qualidades, ela não é a mais indicada para análise de negócio e inteligência artificial. Criar toda a lógica necessária em Lua sem auxílio de bibliotecas prontas levaria muito mais tempo de implementação e iria contra o objetivo de utilizar bibliotecas prontas e padrões existentes suportados pela comunidade.

Relacionado a este fator existe o custo computacional do processamento das informações. Não é de responsabilidade dos servidores da plataforma ARISA NEST o processamento de análise de negócio, execução de algoritmos de inteligência artificial ou de validação de integridade de dados do MES.

Desta forma, a aplicação intermediária agrupa todas as funcionalidades externas à plataforma ARISA e necessárias para o escopo pretendido. São disponibilizados serviços WEB específicos para cada necessidade.

Durante as várias fases de refinamento, são feitas alterações na aplicação intermediária para adaptar as opiniões dos usuários aos objetivos propostos. Além disso, os refinamentos permitem um incremento gradual na complexidade dos algoritmos envolvidos no processamento dos dados.

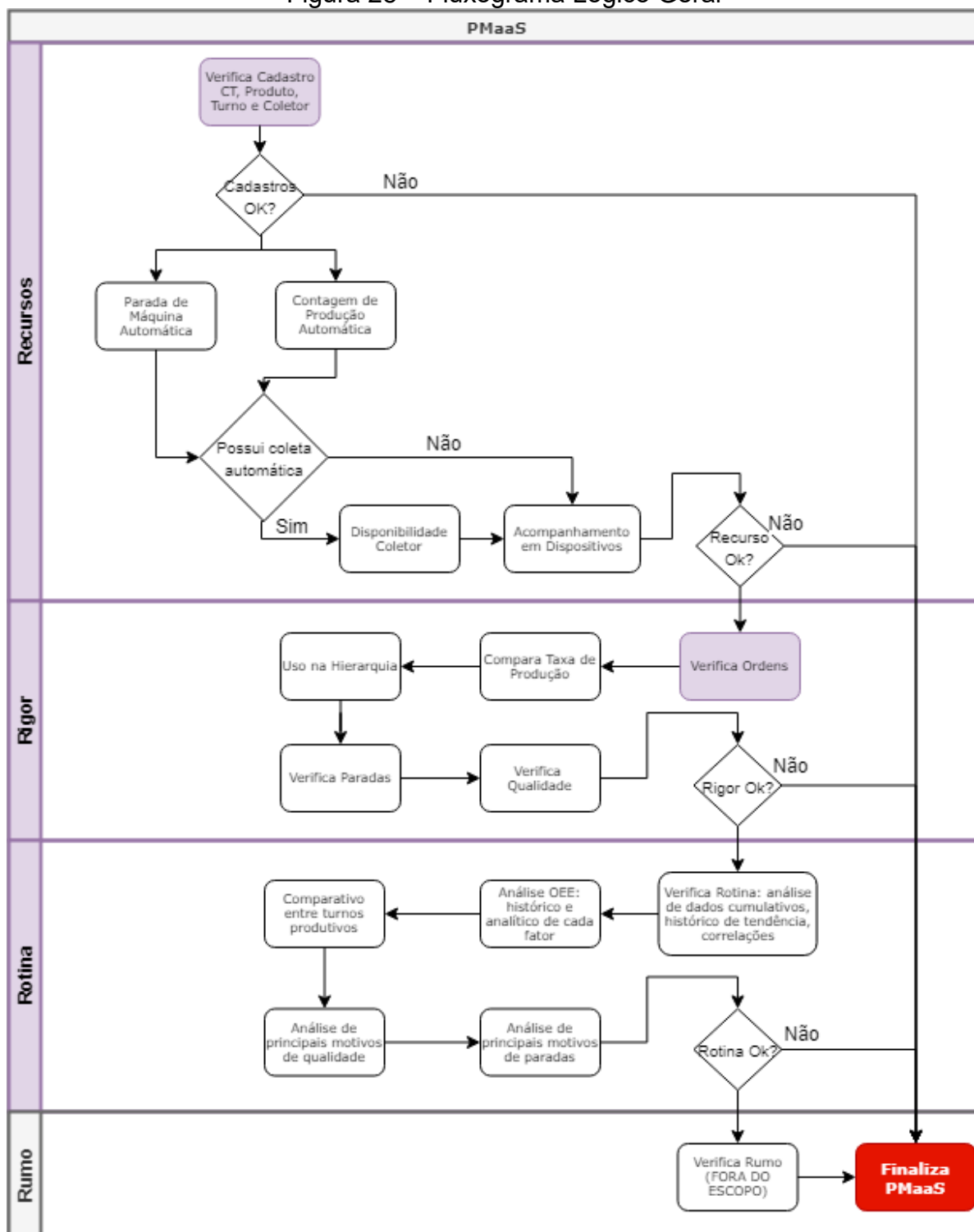
A implementação desta aplicação intermediária não se traduz puramente em programação, mas sim no transporte do conhecimento prático dos especialistas, associados ao conhecimento teórico da literatura, para linguagem de programação.

5.6.1 Análise de maturidade

Mesmo com um modelo de maturidade simplificado bem definido, devido às características heterogêneas das inúmeras empresas clientes, é usual que nem todas as análises sejam executadas para todas as empresas. Para que o processo de

análise de maturidade seja metódico e bem delimitado, foi definido um fluxo natural para a realização das análises.

Figura 28 – Fluxograma Lógico Geral



Fonte: Elaborada pelo autor (2020)

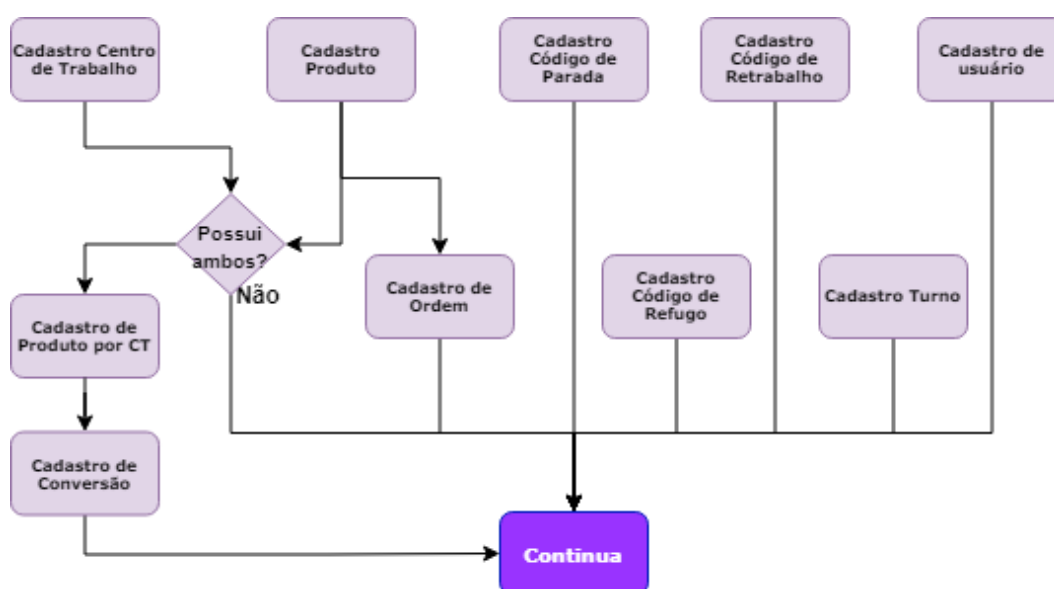
A Figura 28 apresenta um fluxograma lógico geral da implementação do agrupamento de casos de uso “Consultar Maturidade” exposto na Figura 16. Cada processo neste fluxograma é uma caixa fechada contendo um próprio fluxograma lógico interno. Este fluxograma representa o fluxo completo de análise da maturidade quando feito de forma linear e contínua. Porém, como já visto anteriormente, a implementação do AV possui conversação dinâmica e o usuário pode interromper o fluxo pré-definido a qualquer momento. Esta interrupção pode ser uma interrupção definitiva, abortando a continuidade da análise, ou pode ser apenas pulando determinadas análises que não são de interesse do usuário.

Para um maior detalhamento da lógica envolvida internamente aos processos ilustrados no fluxograma geral, foram escolhidos dois casos exemplos. Ambos os casos estão destacados no fluxograma geral e serão ilustrados nas Figuras 29 e 30.

5.6.1.1 Verifica Cadastros

A Verificação de Cadastros compreende todo o fluxo necessário para a validação dos “Dados Mestre” do MÊS (Figura 29).

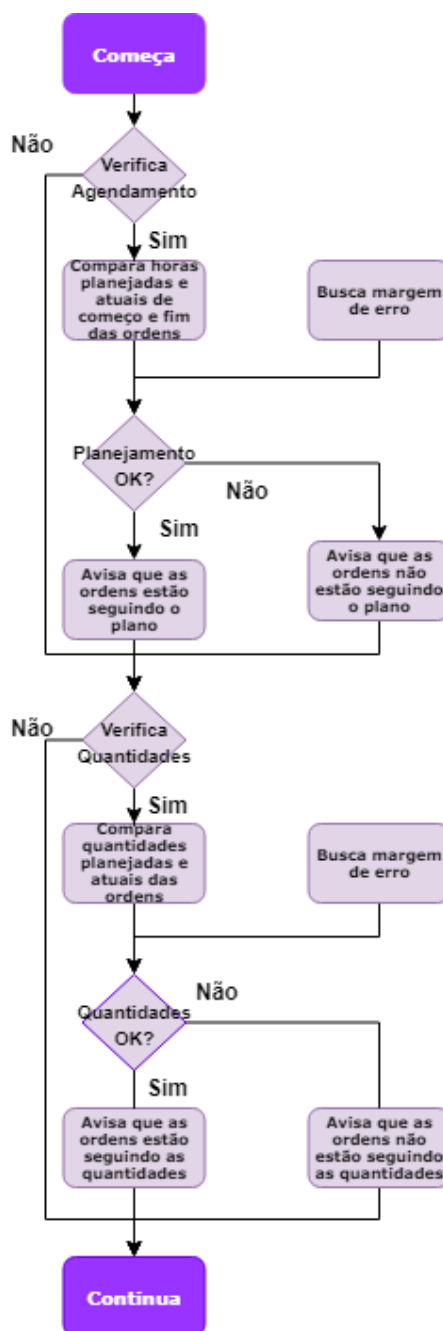
Figura 29 – Fluxograma Lógico da Verificação de Cadastros



Fonte: Elaborada pelo autor (2020)

Para cada um dos cadastros ilustrados na figura, o ciclo de transmissão de dados é realizado. Ou seja, a informação é solicitada para o MES, traduzida, avaliada quanto a sua integridade e processada.

Figura 30 – Fluxograma Lógico da Verificação de Ordens



Fonte: Elaborada pelo autor (2020)

5.6.1.2 *Verifica Ordens*

Como um segundo exemplo, é exibido o fluxograma da Verificação de Ordens na Figura 30.

Antes de cada verificação, o sistema avalia se a verificação deve ser executada ou se deve ser ignorada. A possibilidade de pular qualquer verificação é necessária em sistemas reais heterogêneos, pois determinadas empresas simplesmente não coletam alguns dados. Algumas vezes esses dados não existem por particularidades do processo produtivo da empresa. Por exemplo, um processo pode não possuir nenhum dado de retrabalho, pois toda peça ruim é considerada sucata e não pode ser retrabalhada. Outras vezes a ausência do dado se dá pela incapacidade técnica de coletar o dado. Porém, existem casos em que a empresa não coleta o dado porque não considera relevante.

Para qualquer que seja o caso, uma análise levando em conta dados que não são coletados, proporcionariam uma má avaliação da empresa de forma global. Assim como no exemplo anterior, cada processo deste fluxograma possui um ciclo de transmissão de dados completo.

5.6.2 **Sequência de comunicação**

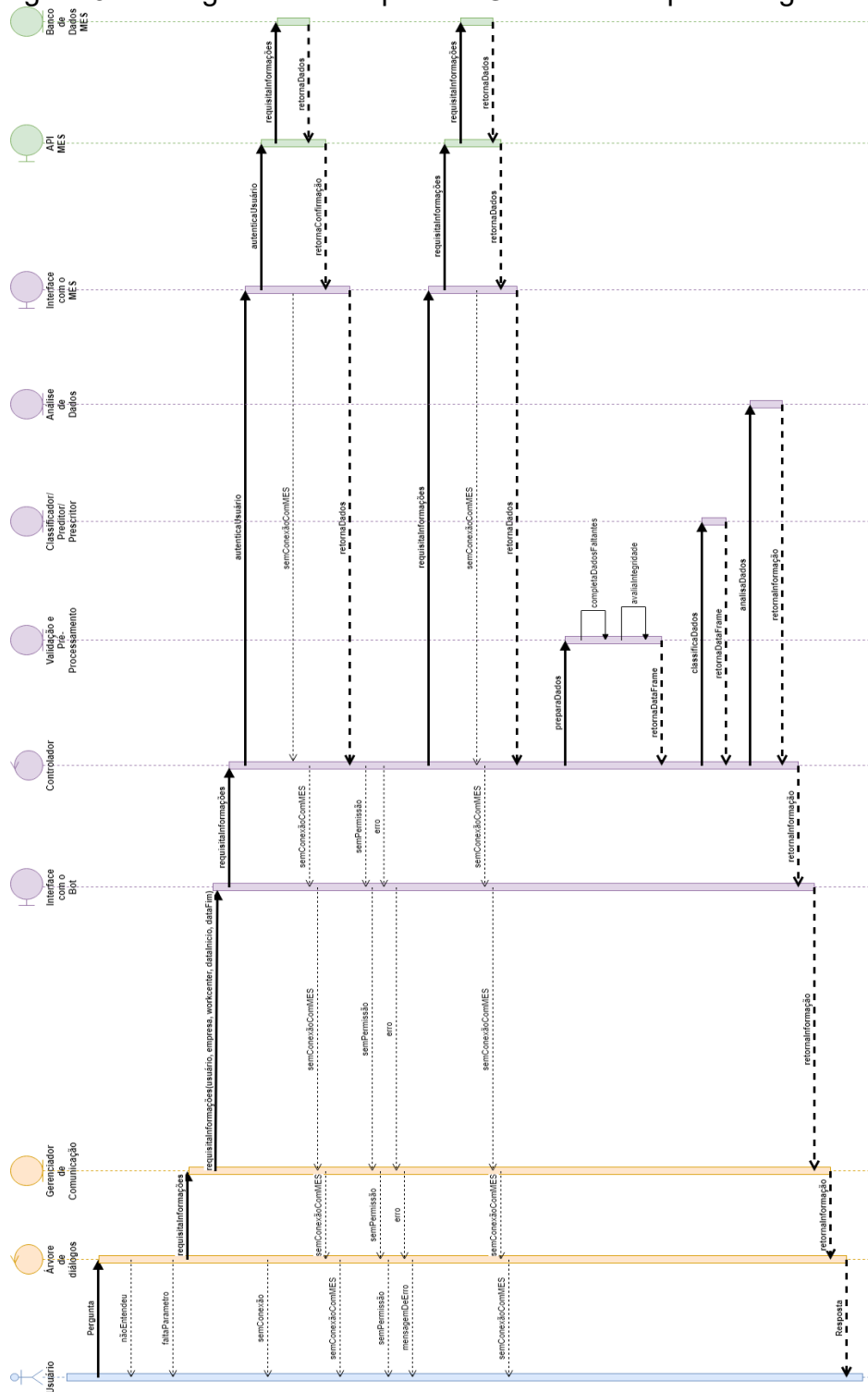
Para cada item analisado, a informação segue um fluxo através dos módulos do sistema até que uma resposta seja devolvida ao usuário. Toda a comunicação implementada é síncrona. Alguns itens da análise de maturidade exigem mais de uma comunicação entre a aplicação intermediária e o MES, já outros itens exigem mais de uma rodada análise de negócio, porém de forma geral todos os itens podem ser representados pelo diagrama de sequência exposto na Figura 31 – Diagrama de Sequência Generalizado para Diagnóstico.

Os elementos em amarelo fazem parte da plataforma ARISA NEST, os elementos em roxo compõem a aplicação intermediária e os elementos em verde integram o LiveMES. O fluxo desejado está destacado, porém o diagrama também

mostra o fluxo de exceções. Não está representado neste diagrama a utilização do Telegram.

O fluxo do comportamento reativo do AV inicia com uma sentença do usuário. A plataforma ARISA NEST encontra o diálogo adequado, este diálogo executa um *script* LUA, que por sua vez, faz uma chamada REST HTTP GET de um serviço WEB da aplicação intermediária. O módulo de interface de comunicação com a ARISA recebe os parâmetros de entrada, faz as devidas conversões e envia para o controlador da aplicação intermediária. O controlador solicita ao módulo de interface com o MES que chame determinado serviço disponibilizado pelo MES. O módulo de interface faz uma chamada REST HTTP GET de um serviço WEB do MES. O MES recebe a requisição faz os processamentos internos necessários e retorna um objeto JSON com os dados solicitados à interface da aplicação intermediária. O módulo de interface faz as alterações sintáticas e semânticas necessárias e retorna um *dataframe* ao módulo controlador. O módulo controlador continua requisitando dados da mesma forma até que todos os dados necessários para a análise desejada estejam disponíveis. Uma vez que todos os dados já estejam disponíveis, o controlador envia um *dataframe* com esses dados para validação e pré-processamento. Os dados são classificados quanto a sua integridade, preparados para utilização de técnicas de IA e retornam ao controlador. Os dados são enviados ao módulo de inteligência para processamento e retornam juntamente com os resultados. Uma análise desses resultados é feita pelo módulo de análise e um objeto JSON com as informações dessa análise retornam ao controlador. As informações são enviadas ao módulo de interface entre a aplicação intermediária e a ARISA, onde são enviados como resposta ao *script* LUA, que retorna para o diálogo. Por fim uma resposta é retornada ao usuário.

Figura 31 – Diagrama de Sequência Generalizado para Diagnóstico



Fonte: Elaborada pelo autor (2020)

Por ser um ambiente real, cada possível exceção precisa ser tratada para que o usuário receba uma resposta amigável do problema ocorrido.

5.6.3 Implementações da inteligência

Utilizando os conceitos de separação entre controle e conhecimento (BRACHMAN e LEVESQUE, 2004) e visando uma fácil substituição entre técnicas de inteligência artificial para a solução de problemas, a implementação da inteligência da Livia foi modular. Isto permitiu a evolução gradual na técnica utilizada para simular a inteligência.

Existiram três tipos básicos de algoritmos utilizados neste módulo:

- Algoritmo procedural
- Sistema especialista
- Outras técnicas de IA

O algoritmo procedural é apenas um código de programação comum construído de forma a receber um *dataframe* com as características (*features*) de entrada e métodos iguais aos utilizados em técnicas de IA. Com os ciclos de refinamento da implementação foi possível substituir alguns destes códigos por sistemas especialistas construídos de forma similar. Para a utilização de outras técnicas de IA é necessária apenas a importação da biblioteca que contenha a técnica desejada e alteração da invocação do método “*predict*”.

A Figura 32 apresenta um trecho de código da classe que faz a invocação da prescrição utilizando um sistema especialista desenvolvido. Assim como os módulos de bibliotecas Python voltadas a IA, este módulo “Inteligência” desenvolvido, espelha os métodos comuns utilizados por padrão na maioria das classes. São os métodos *fit* e *predict*. Porém o método *fit* não faz nada, já que não é necessário treinar o sistema especialista desenvolvido.

As implementações práticas de IA em Python, normalmente seguem o padrão usado neste código. Um objeto do tipo desejado é instanciado, normalmente um classificador, em seguida este objeto é alimentado com dados de entrada para a geração de um modelo com o método *fit* e esse modelo é usado pelo método *predict* para obter os resultados. Essa padronização faz com que a alteração entre técnicas de IA se dê com a simples alteração do objeto instanciado e em alguns casos algum

pré-processamento adicional, já que algumas técnicas exigem normalização das entradas para evitar erros.

Figura 32 – Prescrição usando sistema especialista

```

1 import ExpertSystem as es
2
3 def PMaaSRotinePrescriptiveAction(Authorization, CompanyId, WorkCenterId, StartDate, EndDate):
4     RestStatus, Oee_hst_df = RI.Cube_Oee_History_get(Authorization, CompanyId, StartDate, EndDate)
5     if RestStatus == 200:
6         Oee_hst_df = vd.Oee_History_validade(Oee_hst_df)
7         FullAnalysis_df = Pr.Cube_Oee_History_prepare(Oee_hst_df)
8         intel_db = db.IntelligenceDB()
9         analist = da.DataComprehension(intel_db)
10
11         # Carrega o modelo do Sistema Especialista
12         prescriptor = es.PrescriptAction()
13
14         FullAnalysis_df['PrescriptiveAction'] = prescriptor.predict(FullAnalysis_df)
15         action = analist._actionJSON(FullAnalysis_df[])
16     else:
17         action = errorJSON(RestStatus)
18     return action

```

Fonte: Elaborada pelo autor (2020)

A Figura 33 mostra a implementação do mesmo código, porém agora carregando um modelo armazenado em arquivo e utilizando esse modelo. Neste caso em específico o modelo havia sido obtido com o algoritmo Randon Forest que possui um bom desempenho para prescrições. Porém o código utilizado para a implementação com Rede Neural Recorrente, KNN, ou qualquer outra técnica, utiliza exatamente o mesmo código. Mudando apenas a classe chamada em outro trecho do código onde é realizado o treinamento.

Figura 33 – Geração diária de relatório por empresa

```

1 import Pickle as pk
2
3 def PMaaSRotinePrescriptiveAction(Authorization, CompanyId, WorkCenterId, StartDate, EndDate):
4     RestStatus, Oee_hst_df = RI.Cube_Oee_History_get(Authorization, CompanyId, StartDate, EndDate)
5     if RestStatus == 200:
6         Oee_hst_df = vd.Oee_History_validade(Oee_hst_df)
7         FullAnalysis_df = Pr.Cube_Oee_History_prepare(Oee_hst_df)
8         intel_db = db.IntelligenceDB()
9         analist = da.DataComprehension(intel_db)
10
11         # Carrega o modelo treinado salvo em arquivo
12         prescriptor = pk.load(open('modeloSalvo.sav', 'rb'))
13
14         FullAnalysis_df['PrescriptiveAction'] = prescriptor.predict(FullAnalysis_df)
15         action = analist._actionJSON(FullAnalysis_df[])
16     else:
17         action = errorJSON(RestStatus)
18     return action

```

Fonte: Elaborada pelo autor (2020)

O treinamento das técnicas é feito em outro trecho do código para que o sistema não precise treinar a cada vez que for usado.

5.6.3.1 Técnicas de IA

Como já mencionado, todas as classificações são implementadas na forma de classificadores caixa preta, que podem ser implementados de qualquer forma. Ao todo, para essa primeira implementação da solução, existem 57 chamadas a classificadores, 1 a preditor e 1 a prescritor, totalizando 59 caixas pretas. Esse número de 59 caixas pretas considera a execução de cada parte implementada da solução uma única vez. Isso implica que mesmo para essa solução inicial, é possível utilizar 59 técnicas distintas, gerando 59 modelos distintos para realizar todas as classificações necessárias.

Embora modelos distintos sejam necessários, uma vez que as características (*features*) de entrada sejam diferentes, os algoritmos que geram e utilizam esses modelos não precisam ser diferentes.

Inicialmente todos os classificadores foram implementados utilizando algoritmos procedurais de decisões, como já explicado anteriormente. Porém para fins de validação da proposta e testes de precisão, alguns classificadores foram implementados utilizando técnicas diferentes.

Para a realização de testes e validação do desacoplamento entre técnicas de IA e a arquitetura em si, foram utilizados os algoritmos conforme descritos a seguir. Já que este trabalho não tem o objetivo de definir qual é a melhor e mais precisa técnica de IA, a escolha dos algoritmos se deu apenas a caráter de potencialidades.

Sistemas especialistas: a migração para sistema especialista é a mais natural nesse caso, uma vez que as regras implementadas na forma de algoritmo procedural, já são regras obtidas através dos especialistas no sistema. Uma interface foi criada para que o sistema especialista receba as informações (fatos) na forma de *dataframes* e que aplica as regras pertinentes utilizando o motor de inferência de forma progressiva (*forward chaining*). Não foi construído um sistema especialista

completo, mas sim, foi utilizada a biblioteca PyKe e um sistema foi construído utilizando as suas facilidades.

Randon Forest: Utilizada apenas para testes em um caso e o algoritmo utilizado foi o disponível na biblioteca *sklearn*, utilizando o parâmetro de entrada *n_estimators* igual a 10 para testes com esse mesmo número de árvores.

KNN: Utilizada em apenas um caso e apenas para testes. Esta técnica foi implementada com um algoritmo próprio que utiliza a biblioteca *scipy* para fazer o cálculo das distâncias entre os vizinhos e determinar o vizinho mais próximo.

RNN: Utilizada apenas para testes em um caso. Foi criada uma rede com 3 camadas, função de ativação degrau e otimizador adam. A rede foi criada com a utilização da biblioteca *keras* que faz parte do *tensorflow*.

XGBoost: Utilizado na predição para testes e validação, este algoritmo possui a sua própria biblioteca (*xgboost*). Os testes realizados com este algoritmo obtiveram os melhores resultados com os parâmetros *max_depth* igual a 4, *eta* igual a 0,3, *objective* igual a *multi:softmax*, e *num_class* igual a 3.

5.6.3.2 Envio de e-mail

A aplicação intermediária também implementa o envio de e-mail para os usuários. Foi implementado um envio de e-mails externo a plataforma ARISA NEST para não sobrecarregar os servidores da ARISA. A implementação utiliza o protocolo SMTP.

5.7 DEPLOY

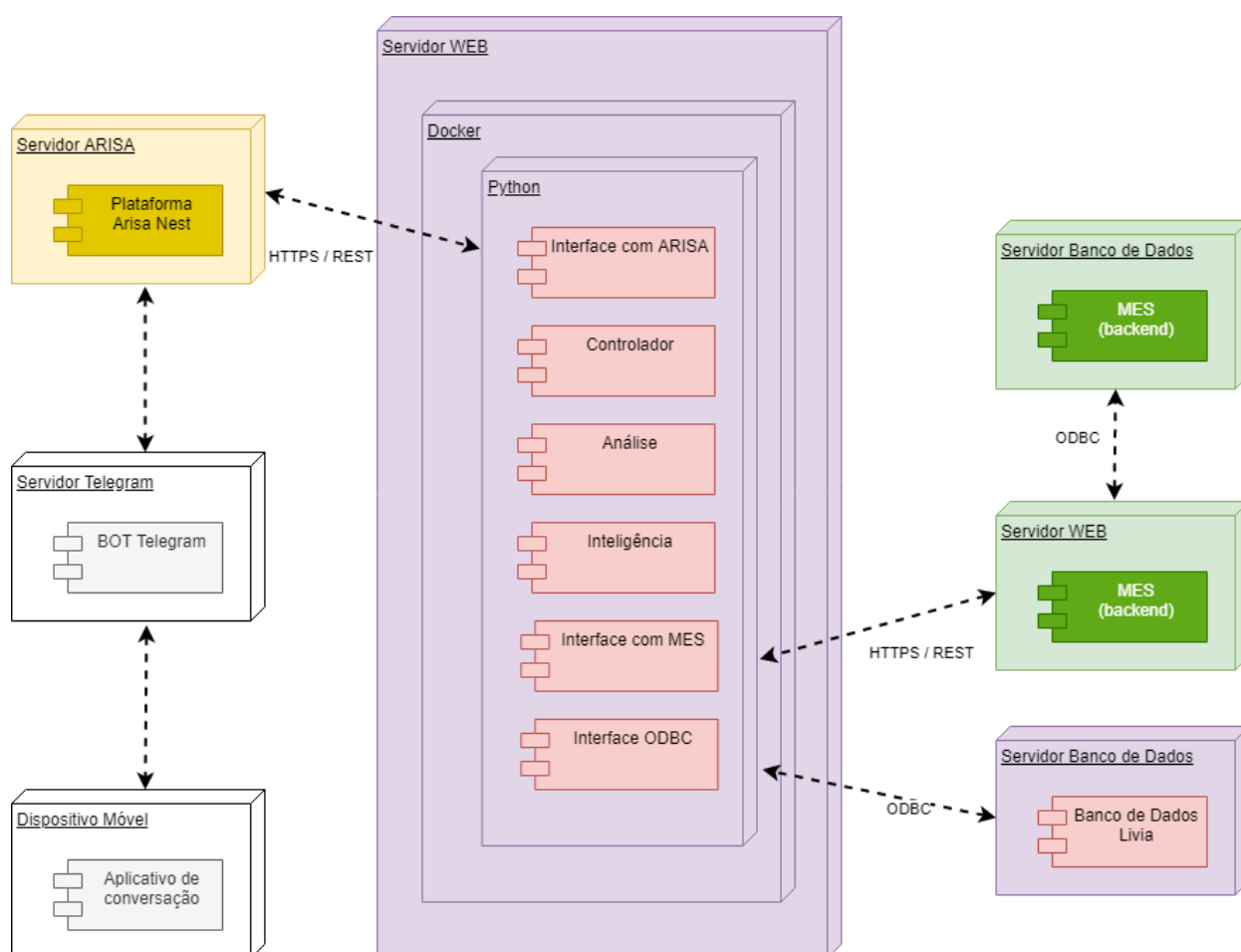
O controle de versão através do Git foi utilizado para gerenciar o envio do código para outros ambientes. Sempre que uma *branch* específica é atualizada, o servidor do repositório encapsula todo o código em um *container* de forma automática utilizando Docker e envia esse código para servidor destino. Lá o novo *container* é carregado e executa a nova aplicação intermediária.

Embora o Python já possibilite uma virtualização natural, essa abordagem utilizando *container* isola o ambiente de forma completa e adiciona uma camada extra de segurança e portabilidade para a solução.

O ambiente de desenvolvimento da aplicação intermediária foi um laptop com processador intel Core i7 de 2,8GHz e 16GB de memória RAM usando sistema operacional Windows 10. Já para os ambientes de QA e produção, foram utilizados servidores na nuvem hospedados pela Amazon com possibilidade de incremento de memória e de processamento conforme a demanda usando sistema operacional Linux.

Como já mencionado, a instância do *softbot* foi projetada, desenvolvida e executada nos servidores da plataforma ARISA NEST.

Figura 34 – Diagrama de Implementação



Fonte: Elaborada pelo autor (2020)

A Figura 34 ilustra como os módulos da solução proposta são distribuídos entre os servidores. O dispositivo móvel possui a aplicação utilizada pelo usuário para comunicação. Essa aplicação, que para este caso é o aplicativo Telegram, se comunica com o servidor do próprio aplicativo e com o serviço de bot disponibilizado para integração com aplicações terceiras. Este bot do Telegram se comunica com a Plataforma Arisa Nest que está instalada em um servidor próprio. A aplicação intermediária é encapsulada em um *container* (Docker) que é executado em um servidor na nuvem. A título de curiosidade esse servidor está localizado na AWS, onde também estão localizados os servidores do LiveMes. Embora ambas as partes da solução estejam localizadas na AWS por conveniência, estão desacopladas e em servidores distintos. Como é possível ver pela figura, elas se comunicam via HTTPS/REST.

6 ASPECTOS DE VALIDAÇÃO E TESTES

Neste capítulo são apresentados os métodos e procedimentos utilizados para validar o sistema implementado.

6.1 VALIDAÇÃO

A validação visa comprovar que a implementação criada atende os requisitos que motivaram o seu desenvolvimento. Como métodos de validação utilizados, são exibidos exemplos reais do funcionamento da solução, resultados de questionário aplicado em entrevistas (*surveys*) e artigo publicado em conferência internacional da área.

6.1.1 Utilização em casos selecionados

Os dois tipos de comportamentos realizados pela Livia são exemplificados a seguir. Primeiramente um exemplo do comportamento reativo contendo grande parte das análises realizadas pela Livia. Em seguida, são feitas considerações sobre o comportamento proativo. Por fim, a mudança de contextos dinâmica é comentada.

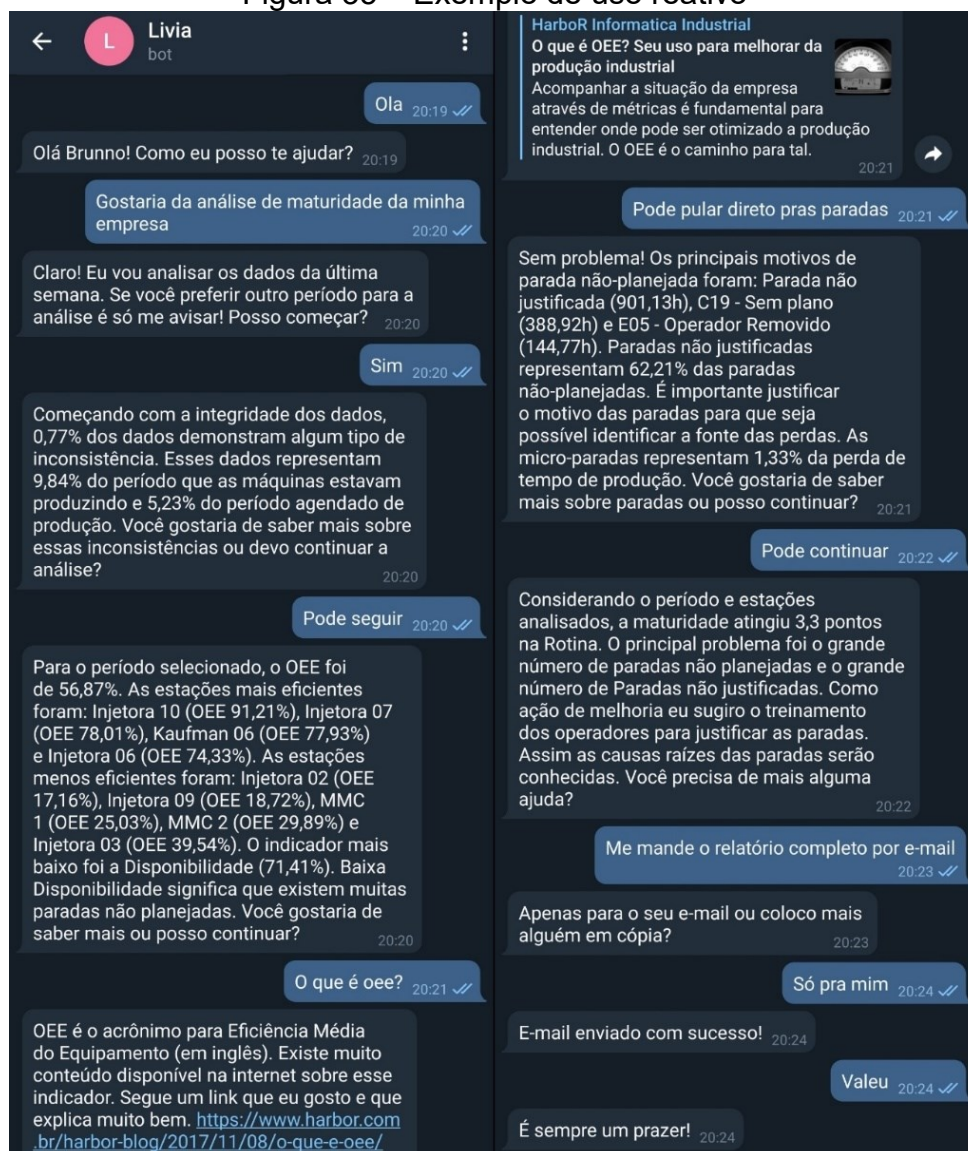
A utilização da Livia é exemplificada, mas não limitada, aos casos exibidos aqui. Todo o desenvolvimento realizado e funcionalidades implementadas, foram avaliados por analistas e desenvolvedores de softwares da Harbor através de utilização real do *softbot*, tanto em ambientes de QA, quanto em ambiente de produção. As opiniões dos usuários serão abordadas no questionário mais à frente.

6.1.1.1 Validação do comportamento reativo

Para validação do comportamento reativo foi utilizado um exemplo de funcionamento da Livia. Este exemplo foi obtido com dados de uma empresa cliente no ambiente de produção do software LiveMES. A Figura 35 foi obtida através *screenshots* realizados em aparelho celular. A conversa com a Livia foi através do

aplicativo Telegram. A imagem original foi dividida em duas partes que estão lado-a-lado na figura. Esta divisão foi feita apenas para uma melhor visualização da figura.

Figura 35 – Exemplo de uso reativo



Fonte: Elaborada pelo autor (2020)

O caso reativo começa com um estímulo do usuário. Neste caso o usuário inicia o diálogo dizendo “Olá”. A Livia identifica um diálogo em seu contexto raiz, que contenha este padrão de entrada e responde, com uma das opções de respostas cadastradas.

Esta resposta contém o nome do usuário “Brunno”, que havia sido informado à Livia em conversa prévia e estava armazenado em uma crença local. A conversa segue com o usuário solicitando a análise de sua empresa. Novamente aqui a Livia já possuía conhecimento prévio de qual era a empresa do usuário. Essa medida foi utilizada para preservar o nome da empresa utilizada no exemplo. Como padrão, a Livia sugere o período da última semana para a análise, mas oferece a opção de troca de período de busca caso desejado. A última semana inicia no penúltimo domingo e termina no último sábado, ambos anteriores ao dia atual.

Quando o usuário solicita a análise de maturidade da empresa e a Livia já conhece esse usuário, além de já saber de qual empresa se trata, a primeira chamada à serviços é realizada. Neste momento a plataforma ARISA chama um serviço disponibilizado pela aplicação intermediária. Esta por sua vez, realiza uma chamada a um serviço disponibilizado no LiveMES. Estas chamadas são feitas para autenticação do usuário. Quando a Livia envia a sua resposta sugerindo o período de busca, o contexto PMaaS se torna o contexto atual.

O usuário concorda com a busca e autoriza a continuidade da análise. Toda vez que o usuário dá prosseguimento a análise a partir deste contexto, é iniciada a sequência de informações de forma similar a exibida na Figura 31.

A análise começa com uma avaliação geral da integridade dos dados obtidos e de quanto esses dados representam no período planejado de produção e no real de produção.

O usuário pede a continuidade da análise e a Livia retorna as informações de OEE da empresa. São listados os melhores e piores centros de trabalho com base nesse indicador e qual dos fatores que compõe o OEE se encontra mais baixo, seguido de uma breve sugestão de um possível problema.

O usuário então pede informação a respeito do significado de OEE. A Livia esclarece a dúvida do usuário com uma breve explicação e envia um link para mais informações.

Já na segunda coluna da imagem o usuário continua a conversa com o pedido para “pular direto para as paradas”, onde é utilizada linguagem informal. A Livia interrompe o fluxo “natural” da análise e vai direto para a Análise de Paradas de linha. Nesta análise são exibidos os principais motivos de paradas da empresa. Além de

valores a Livia realiza uma análise de qual é o principal problema e indica uma forma de mitigar o problema encontrado.

O usuário pede a continuidade da análise e por já se tratar da última análise, ela parte para suas considerações finais. É informada a pontuação alcançada no nível atual de maturidade da empresa, que neste caso é o nível de “Rotina” conforme o modelo implementado. Dentre todos os problemas encontrados, incluindo os que foram “pulados” pelo usuário, a Livia informa o principal problema considerado por ela. Neste caso é o excesso de paradas não-planejadas que não foram justificadas. Em seguida, ela orienta o usuário com uma ação para diminuir o problema encontrado.

Por fim o usuário pede para receber o relatório completo por e-mail sem colocar ninguém “em cópia”.

O e-mail é enviado ao usuário, que agradece. Terminando a conversa com o a despedida da Livia.

6.1.1.2 Validação do comportamento proativo

Para validação do comportamento proativo, foi criado um comportamento ARISA com agendamento diário para envio de relatório para os usuários cadastrados para o recebimento. A análise de cada empresa disponível no MES que o usuário tenha acesso, gera um relatório do dia anterior. A execução dessa rotina foi agendada para as 06:00 horas.

Figura 36 – Geração diária de relatório por empresa

• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.
• livia@livemes.com	★ Relatório diário da empresa [REDACTED] (2020-09-10)	Bom dia, aqui vai uma breve análise sobre a sua...	Entrada	11 de set.

Fonte: Elaborada pelo autor (2020)

A Figura 36 mostra a caixa de entrada de e-mails do usuário e comprova o recebimento dos relatórios.

Este relatório diário automático foi utilizado por alguns dias em caráter de validação e foi desativado. Para fins de validação eles foram criados com os mesmos textos presentes nos diálogos usados no bate-papo.

6.1.1.3 Validação da mudança de contexto dinâmica

Como evidenciado na Figura 35, o usuário é capaz de alterar dinamicamente entre os contextos das análises de maturidade. No exemplo o usuário deu um “salto” para frente no fluxo da maturidade, porém qualquer contexto pode ser acessado a qualquer hora dentro do escopo da maturidade.

De fato, houve mais de uma ocasião onde houve esse “salto” entre contextos. Além do evento já mencionado, o usuário pede o envio de e-mail para a Livia. Envio de e-mails é um contexto separado da análise prescritiva que estava sendo feita, ou seja, neste ponto a mudança dinâmica de contexto foi realizada.

6.1.2 Questionário

De acordo com (GÜNTHER, 2003), questionar pessoas sobre determinado assunto é uma forma de conduzir estudos empíricos de observação, experimento e *survey* (avaliação). A opinião condensada dos entrevistados é de grande valor na análise do objeto de estudo de uma pesquisa.

Para a elaboração de um questionário é importante analisar o objetivo da pesquisa e qual é a população-alvo (GÜNTHER, 2003). Através de perguntas, conectadas aos objetivos do trabalho, este método de estudo é caracterizado por ser uma das mais rápidas e eficientes formas de coletar dados (GIL, 2010).

O tipo das respostas possíveis escolhido para aplicação do questionário do presente trabalho é baseado na escala de (LIKERT, 1932), em que é medido o nível de concordância à afirmação. Normalmente são utilizados cinco níveis de respostas (Não concordo fortemente, não concordo parcialmente, indiferente, concordo parcialmente e não concordo fortemente). Além das respostas normalmente

utilizadas, a resposta "Não se aplica / não sei" foi adicionada para o caso de o entrevistado não se sentir apto a responder.

6.1.2.1 Perguntas do questionário

O Quadro 5 – Perguntas do questionário aplicado apresenta as perguntas utilizadas no questionário apresentado aos entrevistados.

Quadro 5 – Perguntas do questionário aplicado

ID	Pergunta
1	Na sua opinião, a interface com o usuário através de <i>chatbot</i> é adequada para auxiliar gerentes de produção em suas análises?
2	Você enxerga essa solução como um diferencial importante para análises da produção?
3	Você se sente confortável em interagir com <i>softbots</i> ?
4	Você considera a abordagem " <i>as a service</i> " (sob demanda) mais adequada do que a abordagem de licenciamento de produto para um assistente de gerenciamento de produção?
5	Você acredita que essa solução pode ser usada comercialmente para auxiliar os gerentes de produção a tornar suas análises mais rápidas?
6	Você acredita que essa solução pode ser usada diretamente por usuários do MES que não tenham formação na área de gerência de produção?
7	Na sua opinião, utilizar um Software Assistente Virtual para fazer análises de maturidade de empresas ajuda na inclusão para a Indústria 4.0?
8	Na sua opinião, a proposta apresentada pode ser considerada inovadora?
9	Você acredita um Assistente Virtual como serviço pode ser capaz de realizar a grande maioria das análises, liberando os analistas para os casos mais complexos ou para outras atividades?
10	Na sua opinião, o Assistente Virtual deve ser capaz de indicar que ação deve ser tomada e não apenas exibir os problemas encontrados?

11	Você considera necessário que o Assistente Virtual possua um comportamento proativo, ou seja, que ele notifique o usuário sobre alguma condição observada por conta própria?
12	Sendo um analista de produção, as suas análises se tornaram mais direcionadas e rápidas usando a solução proposta?

Fonte: Elaborada pelo autor (2020)

6.1.2.2 Respostas dos entrevistados

As respostas obtidas com a aplicação do questionário foram analisadas e são mostradas a seguir. Ao todo, 48 pessoas foram entrevistadas. Elas foram classificadas em 3 grupos. São eles: PMaaS (verde), Colaboradores da Harbor (azul) e Total (amarelo).

O grupo Total são todas as pessoas entrevistadas, dentro deste grupo existe o grupo de Colaboradores da Harbor e dentro deste outro grupo existe o grupo PMaaS, que leva em conta apenas os Gerentes de Produção como Serviço e são a comunidade alvo desta pesquisa. Das 48 pessoas entrevistadas, 16 são Colaboradores da Harbor e dentre eles 5 são PMaaS.

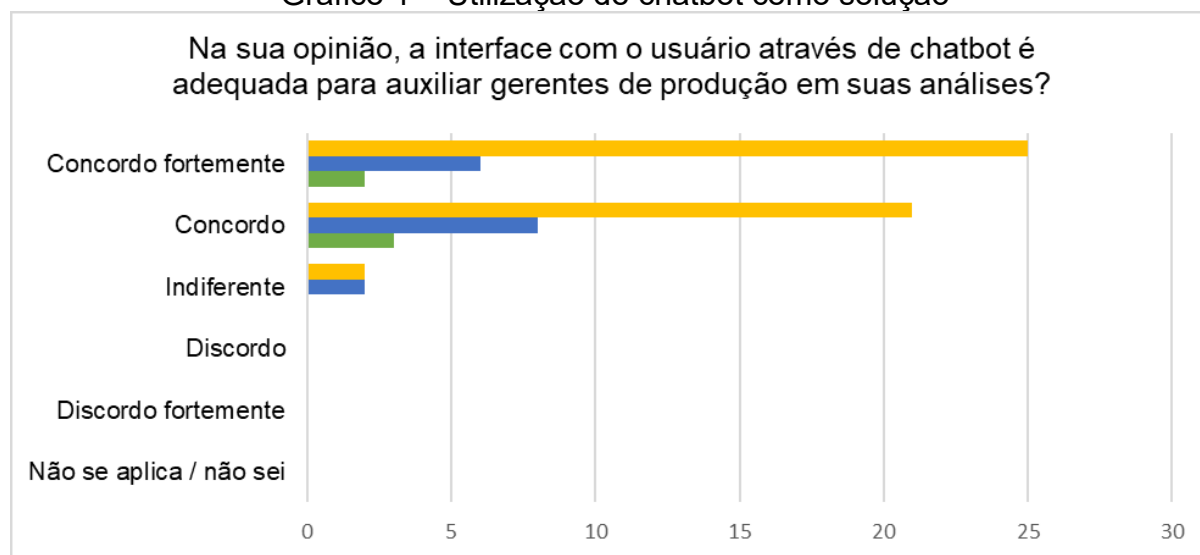
Para preservação da identidade dos entrevistados, eles foram numerados de #1 a #48, sendo que de #1 a #5 são os entrevistados PMaaS, de #6 a #16 são entrevistados Colaboradores da Harbor que não são PMaaS e de #17 a #48 são pessoas externas à Harbor.

Os grupos foram separados desta forma para estratificar os diferentes níveis de conhecimento sobre a solução proposta. Entrevistados PMaaS participaram ativamente das etapas da pesquisa, realizaram testes e utilizaram a Livia em suas análises. Outros colaboradores da Harbor, puderam fazer alguns testes, tinham acesso às informações fornecidas pela Livia e acompanharam o projeto de perto dando algumas opiniões ao longo das iterações. Já as pessoas externas à Harbor, não puderam utilizar a Livia por questões de privacidade dos dados das empresas as quais ela está conectada. Estas pessoas receberam uma apresentação sobre a solução, tiveram acesso ao artigo publicado e puderam ver o funcionamento informativo que não exige comunicação com o LiveMES.

Todas as pessoas tiveram acesso a todas as perguntas realizadas, com exceção da última pergunta que é dirigida especificamente ao grupo PMaaS.

O Gráfico 1 – Utilização de chatbot como solução é referente a relevância de se utilizar chatbots para a obtenção do resultado esperado pela pesquisa.

Gráfico 1 – Utilização de chatbot como solução

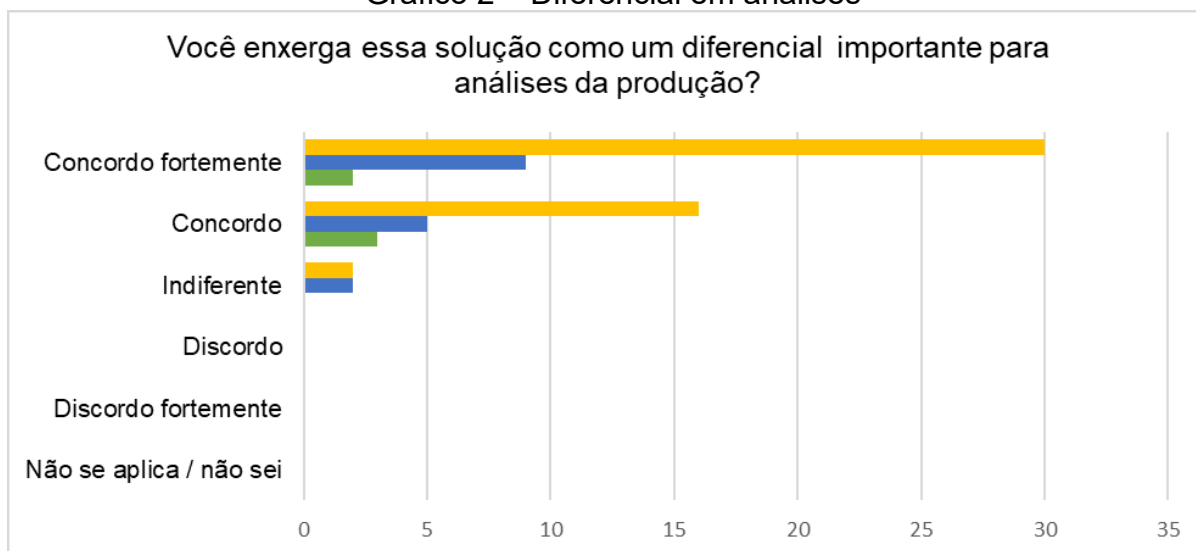


Fonte: Elaborada pelo autor (2020)

Quanto a esta questão, não houve nenhum entrevistado que tenha discordado. Além disso, 100% dos PMaaS concordam com a adequação da abordagem proposta.

O Gráfico 2 – Diferencial em análises refere-se à agregação de valor perceptível que essa solução proporciona ao ser adotada.

Gráfico 2 – Diferencial em análises

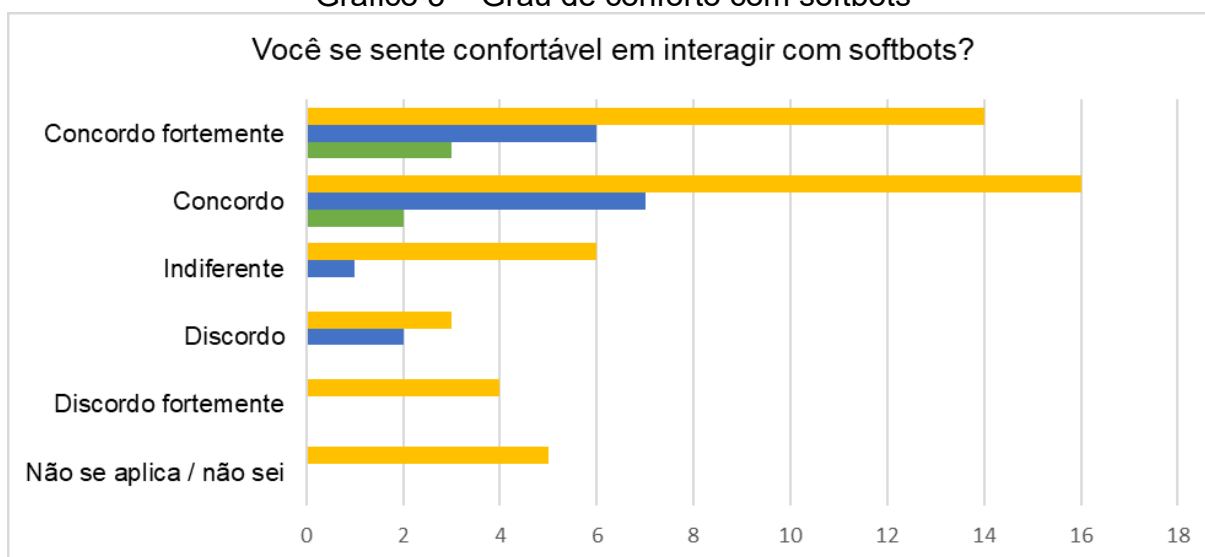


Fonte: Elaborada pelo autor (2020)

Novamente não houve nenhum entrevistado que tenha discordado e novamente 100% dos PMaaS consideram a solução proposta como sendo um diferencial competitivo no mercado.

O Gráfico 3 – Grau de conforto com *softbots* apresenta a influência que o preconceito e as experiências prévias em interações com *softbots* exercem sobre a aceitação da solução.

Gráfico 3 – Grau de conforto com softbots



Fonte: Elaborada pelo autor (2020)

Nesta questão, como esperado, houve vários entrevistados que discordaram. Com isso, é possível estabelecer um marco zero nas opiniões dos entrevistados e entender se as suas respostas são baseadas no seu entendimento e uso da solução proposta, ou se estão fortemente influenciadas em opiniões preestabelecidas anteriores a este trabalho. Alguns entrevistados deixaram comentários sobre o motivo de discordarem. O entrevistado #9 comenta:

Eu não gosto de chatbots, mas é opinião pessoal. – Entrevistado #9

Já o entrevistado #33, explica:

As vezes os chatbots respondem umas coisas que não tem nada a ver com o que eu tô perguntando, é frustrante na maioria das vezes – Entrevistado #33

Os comentários expostos pelos entrevistados demonstram como a má implementação de *softbots* afeta a aceitação desta tecnologia causando um sentimento negativo em algumas pessoas. O comentário do entrevistado #33 mostra esse sentimento negativo adquirido por ele e exemplifica a razão por este trabalho propor a modelagem de trocas abruptas de contextos.

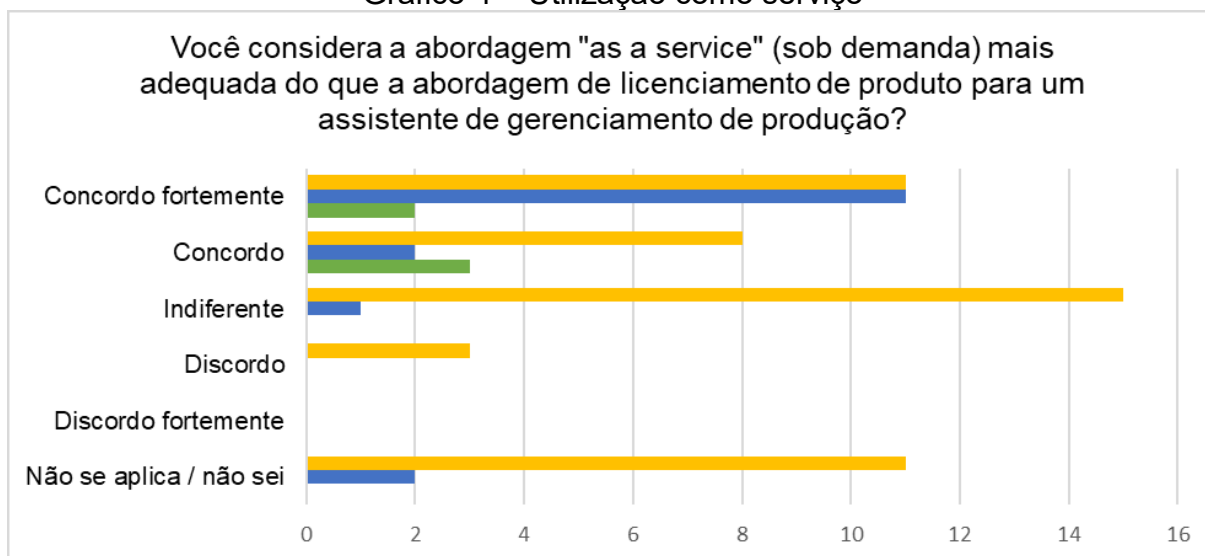
Dentre os entrevistados que concordam, há um comentário deixado pelo entrevistado #2 que vale ser destacado. Ele comenta:

Não tenho problema de desconforto quando uso chatbots. Normalmente preferia não estar usando, mas não tenho escolha. Em geral, o bot só precisa responder algo aleatório uma vez pra eu já não querer mais usar. – Entrevistado #2

Mesmo concordando que se sente confortável com a utilização de *chatbots* o usuário expõe a fragilidade que um AV pode apresentar quando interpreta de forma errada a sentença do usuário e responde diferente do esperado.

O Gráfico 4 – Utilização como serviço considera os benefícios de uma abordagem “como serviço” frente a uma abordagem clássica de licenciamento de produto para a distribuição da solução proposta.

Gráfico 4 – Utilização como serviço



Fonte: Elaborada pelo autor (2020)

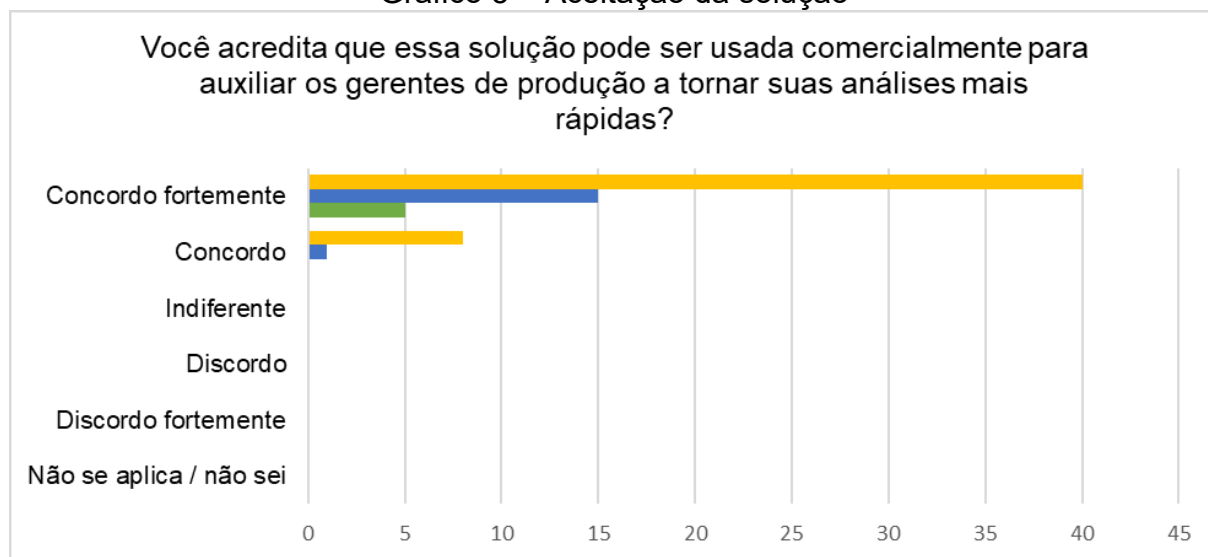
As respostas para essa questão são entendidas de forma positiva. Dentre os entrevistados, 11 pessoas não souberam responder e 15 pessoas consideram a forma como a solução vai ser licenciada indiferente. Apenas 8,1% dos entrevistados que souberam opinar discordam que a estratégia SaaS não seja o ideal. Somente o entrevistado #41 deixou um comentário, dentre os entrevistados que discordam. Ele comenta:

Os serviços sob demanda parecem mais baratos porque pagamos pouco por mês, mas a longo prazo são mais caros. Prefiro comprar os programas porque pago uma vez só – Entrevistado #41

A opinião do entrevistado #41 reflete apenas o gasto com o software em si e desconsidera os gastos relacionados a manutenção, infraestrutura e atualizações do software. Além disso esclarece que se trata de gosto pessoal do entrevistado.

O Gráfico 5 – Aceitação da solução expõe a opinião dos usuários quanto à aceitação do protótipo desenvolvido já possuir condições de ser usado como um produto.

Gráfico 5 – Aceitação da solução

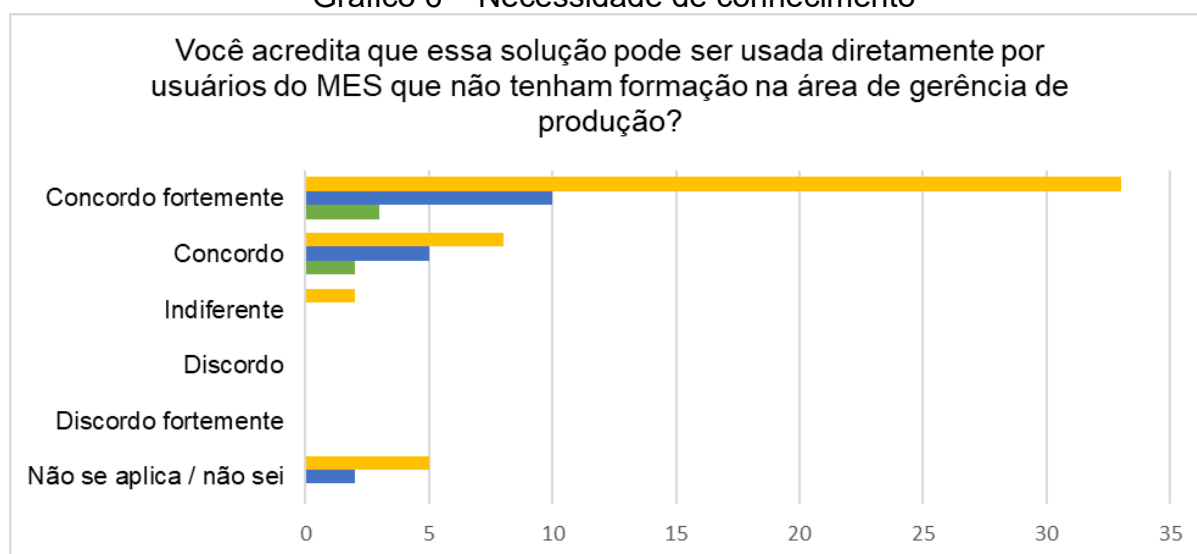


Fonte: Elaborada pelo autor (2020)

Todos os entrevistados concordaram que o protótipo inicial concebido já possui as características e requisitos suficiente para que possa se tornar um produto. Vale destacar que 83,3% dos entrevistados “concordam fortemente”, assim como 100% dos PMaaS.

O Gráfico 6 – Necessidade de conhecimento mostra a opinião dos entrevistados quanto a necessidade de conhecimento teórico prévio para a utilização do protótipo construído.

Gráfico 6 – Necessidade de conhecimento

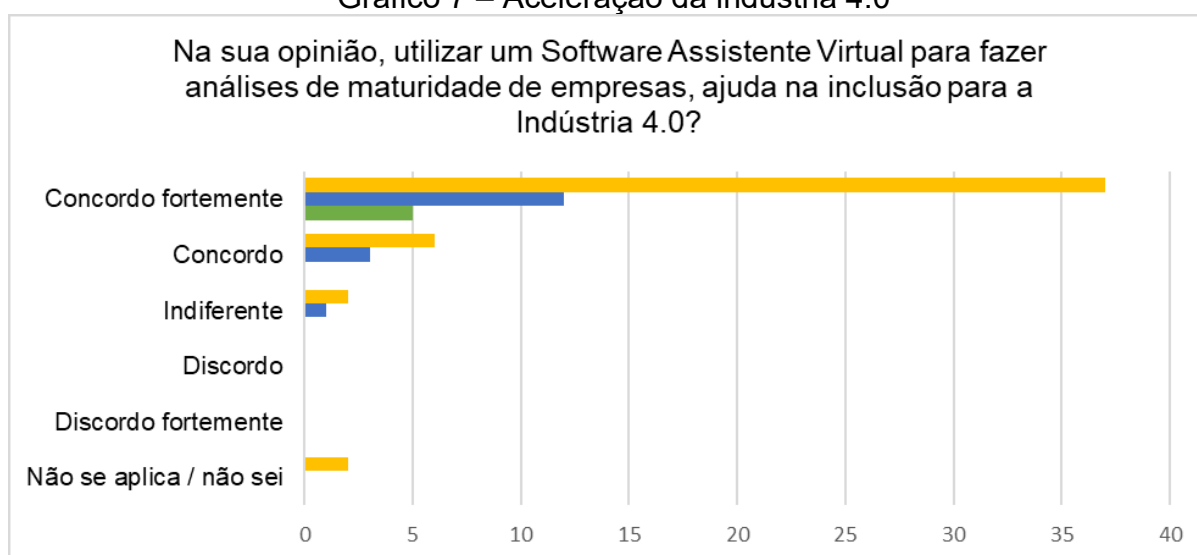


Fonte: Elaborada pelo autor (2020)

Não houve nenhum entrevistado que tenha discordado da capacidade da solução apresentada em ser utilizada diretamente pelos usuários nas empresas clientes e não apenas por pessoas com conhecimento formal sobre o tema. É considerado aqui como um sucesso de implementação. Dado pela construção dos diálogos da Livia é feita de forma informal, por permitir mudanças abruptas de contextos e por oferecer ajudas contextualizadas. Não houve nenhum comentário relacionado a esta questão.

O Gráfico 7 – Aceleração da indústria 4.0 considera a opinião dos entrevistados sobre um dos pontos focais desta dissertação.

Gráfico 7 – Aceleração da indústria 4.0



Fonte: Elaborada pelo autor (2020)

Não houve nenhum entrevistado que tenha discordado que a abordagem utilizada impacta positivamente na velocidade em atingir os objetivos da indústria 4.0. Considerando apenas os usuários que souberam opinar, 80,4% dos entrevistados “concordam fortemente” com o questionamento.

O Gráfico 8 – Inovação considera a opinião dos entrevistados a respeito da inovação da solução proposta frente a qualquer outra solução conhecida por eles.

Gráfico 8 – Inovação



Fonte: Elaborada pelo autor (2020)

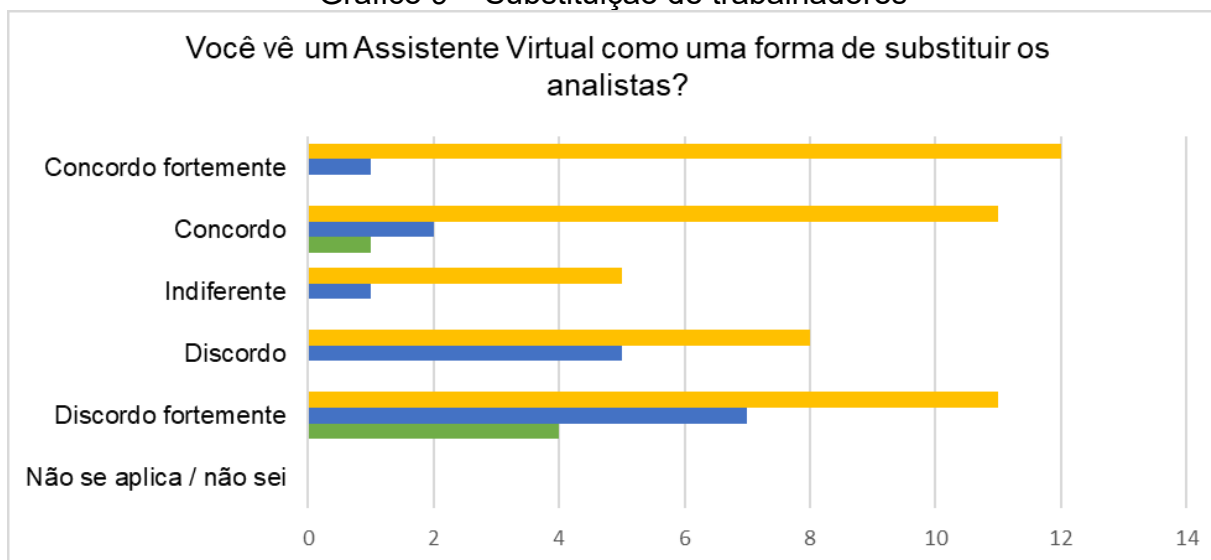
A interpretação desses resultados é bastante significativa, pois deve se levar em conta que os colaboradores da Harbor e os PMaaS trabalham diariamente com o tema e constantemente buscam novas soluções no mercado para traçar estratégias e criar novos produtos. Mais do que os entrevistados da Harbor, 100% do total de entrevistados considerou a Livia como um projeto inovador.

O Gráfico 9 – Substituição de trabalhadores considera as perspectivas futuras de substituição de mão de obra por AVs do ponto de vista dos entrevistados.

Dentre todas as perguntas do questionário, esta foi a que mais apresentou divergências entre os entrevistados e não houve nenhum entrevistado que tenha se considerado incapaz de opinar. Analisando o grupo de entrevistados PMaaS, notamos que apenas 1 entre os 5 não “discorda fortemente” do questionamento, porém este entrevistado não deixou nenhum comentário relacionado. Dentre os que discordam, o entrevistado #1 revela o seu ponto de vista sobre o tema. Ele explica:

A ideia do chatbot como um auxiliar de analista de processo me parece super inovadora e útil, porém não acredito que vá substituir grande parte das atividades do analista, mas sim agilizar ou direcionar para lugares importantes. – Entrevistado #1

Gráfico 9 – Substituição de trabalhadores



Fonte: Elaborada pelo autor (2020)

Esta opinião é reforçada pelo comentário do entrevistado #5, que diz:

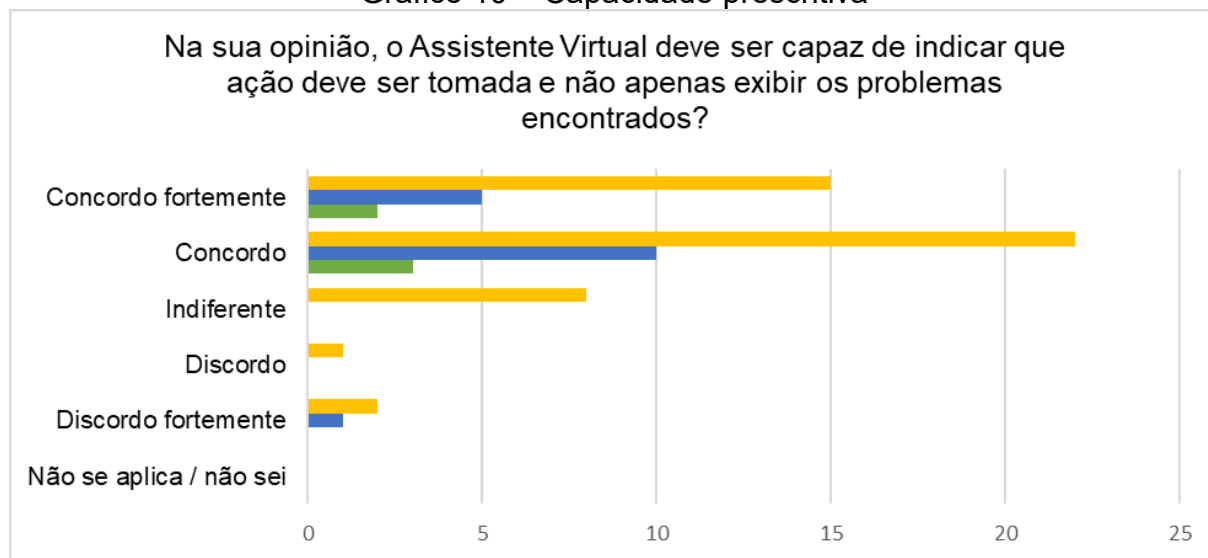
Não acho que um bot consiga substituir um analista porque muito do que é analisado vem de informações obtidas através de reuniões com os clientes e do 'feeling' do analista. Além disso o bot não é capaz de notar que análise é mais valiosa para cada cliente específico. Mas acho que o bot ajuda a agilizar as análises. – Entrevistado #5

Ambos entrevistados possuem pontos de vista semelhantes sobre a substituição dos analistas. Eles concordam em avaliar que a análise envolve mais do que os dados medidos pelos MES.

Um outro ponto relevante sobre os resultados desta questão, é a observação que a tendência dos colaboradores da Harbor está no sentido contrário a tendência de entrevistados externos a mesma empresa. Isto é entendido como uma falta de conhecimento dos trabalhos necessários para a realização de uma análise pelos PMaaS da Harbor. Pessoas externas ao trabalho realizado pelos PMaaS tendem a pensar que é um trabalho meramente informativo, ou seja, apenas são informados os dados que já estão disponíveis no sistema. De fato, essa análise informativa é parte do trabalho realizado pelos PMaaS, assim como parte das análises realizadas pela Livia, porém a interpretação dos dados é um grande diferencial e fica evidente no questionamento a seguir.

O Gráfico 10 – Capacidade prescritiva considera a opinião dos entrevistados a respeito da capacidade de análise prescritiva ser uma necessidade em um AV que se proponha a auxiliar PMaaS, ou ser uma funcionalidade adicional e dispensável para tal fim.

Gráfico 10 – Capacidade prescritiva



Fonte: Elaborada pelo autor (2020)

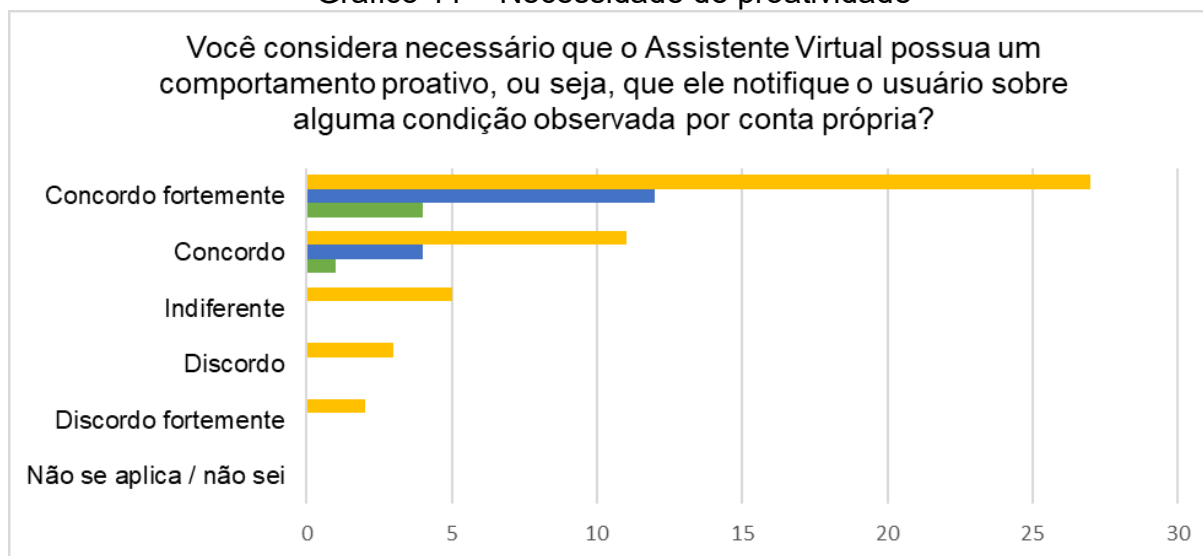
Nesta questão, 77% dos entrevistados acreditam que seja necessário que o AV possua análise descritiva dos dados coletados para que seja capaz de auxiliar os PMaaS de forma consistente. Esses resultados complementam a argumentação feita no questionamento anterior sobre a importância de outros tipos de análises realizadas. Dentre os entrevistados que discordam, o entrevistado #10 comenta:

Acredito que a capacidade de indicar problemas é bem-vinda e desejada. Certamente agrega muito valor ao assistente virtual. Mas não vejo como uma obrigação e sim como um grande diferencial. – Entrevistado #10

O entrevistado #10, não vê a capacidade de realização de análises prescritivas como uma necessidade para auxiliar os PMaaS e sim como um “grande diferencial”. Embora discorde do questionamento feito, este entrevistado demonstra dar um grande valor para a automatização deste tipo de análise.

O Gráfico 11 – Necessidade de proatividade demonstra a importância do comportamento proativo do ponto de vista dos usuários.

Gráfico 11 – Necessidade de proatividade



Fonte: Elaborada pelo autor (2020)

Os resultados desta questão são interpretados de forma positiva quanto a necessidade de comportamentos proativos. Todos os PMaaS concordam sobre a necessidade de um comportamento proativo. Dentre eles, o entrevistado #3 comenta:

Acredito que o modelo tem que ser proativo para dar mais resultado. Uma vigilante da lei e da ordem no mundo da fábrica - regras desrespeitadas devem ser apontadas. – Entrevistado #3

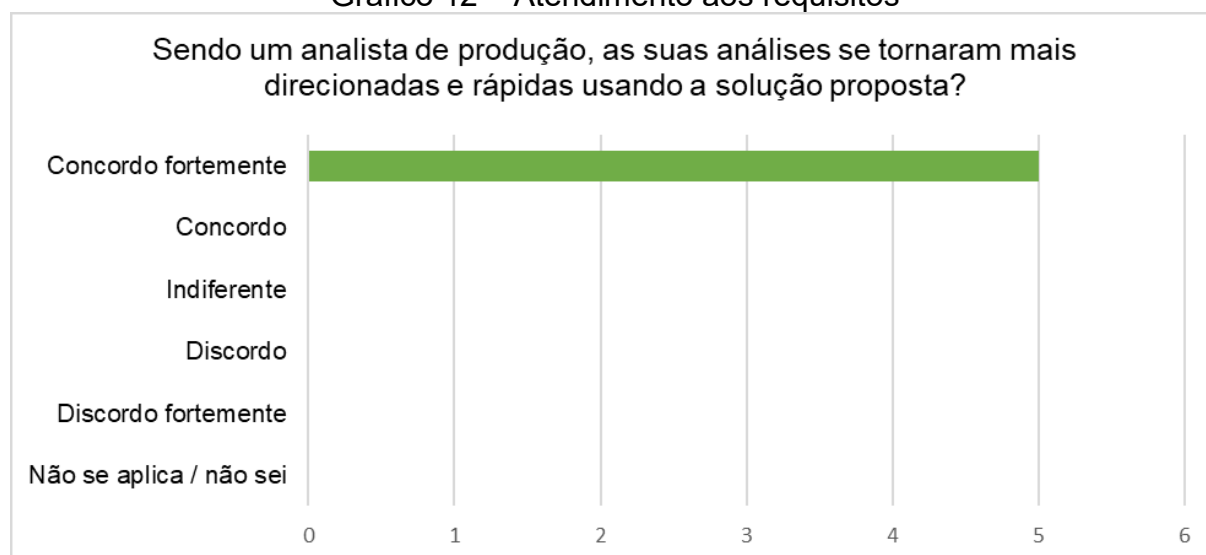
A visão do entrevistado #3 é de que o AV deve controlar ativamente os apontamentos realizados de forma manual pelos operadores do chão de fábrica. Neste cenário o AV monitora todas as variáveis de produção e alerta os operadores e seus superiores sobre falta de apontamentos e justificativas manuais, assim como falhas e seguimento do plano de produção.

Embora seja uma abordagem interessante, ela representa um trabalho futuro ao realizado no presente trabalho. Para a criação do comportamento “policia” sugerido, é necessário primeiramente validar a utilização de AV neste contexto, construir um AV capaz de realizar todos os tipos de “policia” necessários

considerando alguns casos inicialmente. Esta é, de certo modo, a definição do escopo desta dissertação, com as ressalvas de que o comportamento proativo na Livia não implementado para policiar e sim para avisar uma única vez, e de que as funcionalidades consideradas para este escopo inicial são direcionadas aos PMaaS ao invés de serem focadas nos operadores, como sugerido pelo entrevistado #3.

O Gráfico 12 – Atendimento aos requisitos consolida as opiniões dos entrevistados sobre a utilização da solução disponibilizada considerando as necessidades e requisitos que motivaram o presente estudo. Esta foi uma pergunta dirigida especificamente ao PMaaS que utilizaram a solução na Harbor e por isso apenas as repostas dos 5 entrevistados são exibidas.

Gráfico 12 – Atendimento aos requisitos



Fonte: Elaborada pelo autor (2020)

Todos os PMaaS “concordam fortemente” que a utilização da Livia foi capaz de auxiliar nas análises das empresas e indicar soluções para problemas encontrados de forma automática. Além dos dados, um comentário do entrevistado #1 conclui:

Muito bom o trabalho, acho que será super importante na i4.0. – Entrevistado #1

A opinião final do entrevistado #1 expressa sua avaliação final sobre a Livia. Este é considerado o resultado mais importante do questionário e talvez o mais relevante do estudo como um todo.

A interpretação deste gráfico representa o sucesso da pesquisa-ação proposta, pois deixa claro que ação realizada, utilizando a solução proposta, concebida por meio de uma pesquisa, que pretende solucionar um problema de uma dada comunidade, resolveu o problema proposto de acordo com a opinião da própria comunidade.

6.2 TESTES

A realização de testes também é uma forma de validação de um conceito. Neste trabalho, os testes realizados visam validar os aspectos arquiteturais da solução proposta. Dado que esta dissertação pretende agir em um ambiente real e ativo, não foram realizados testes que pudessem causar prejuízo na experiência de utilização do MES pelas empresas clientes.

6.2.1 Teste unitário

Os testes unitários foram realizados com auxílio do *framework* de testes unittest disponível para Python. Apenas versões do código que não apresentaram nenhuma falha nos testes unitários, foram enviadas para o ambiente de QA. Esta é uma medida de garantia de qualidade exigida pelo ambiente de QA disponibilizado pela empresa Harbor.

6.2.2 Teste de portabilidade

A portabilidade testada é apenas referente à aplicação intermediária, já que a instância do *softbot* fica sempre no servidor da plataforma ARISA NEST.

O desenvolvimento foi feito utilizando 2 *laptops* distintos, ambos com sistema operacional Windows. Como primeiro teste, a aplicação foi migrada para um servidor

em nuvem hospedado na *Amazon* com sistema operacional Ubuntu. Não houve nenhuma implicação na troca de ambiente.

Em seguida foi implementado a troca dinâmica de ambientes e a solução conseguiu funcionar normalmente nos dois ambientes de forma simultânea.

Por fim foi realizado um teste utilizando o aplicativo QPython para android. Foi possível executar a aplicação intermediária no celular. Contudo esta abordagem é bastante limitada pois nem todas as bibliotecas conseguem executar de forma correta neste motor Python para *Android*.

6.2.3 Teste de desempenho

Os testes de desempenho realizados visam verificar a viabilidade desta solução. Atualmente o tempo levado para a realização da análise de uma empresa pelos especialistas é maior que duas horas por empresa e a ordem de grandeza da execução das análises pela Livia é de segundos. Mais do que isso, a interação entre usuário e sistema se faz por meio de bate-papo, o que permite alguns segundos de tolerância para a obtenção da resposta. Desta forma, os tempos de execução coletados foram gerados pela utilização real dos usuários.

Tabela 1 – Tempos de Execução da Aplicação Intermediária

Caso de Uso	Mínimo (s)	Máximo (s)	Média (s)
Autenticação	0,1	0,8	0,3
Integridade	1,0	1,5	1,3
OEE	2,3	3,1	2,5
Completa	3,4	10,7	8,1
Proativo Completo*	62,9	193,4	131,3

Fonte: Elaborada pelo autor (2020)

A Tabela 1 – Tempos de Execução da Aplicação Intermediária apresenta os tempos de execução de alguns casos. Com exceção do caso Proativo Completo, todos os outros casos possuem tempos inferiores a 10s e sendo assim atendem os requisitos. O caso proativo utilizado, realiza a análise para todas as empresas de forma contínua e extrapola o tempo de execução limite, porém se trata de um caso

proativo, ou seja, não é um tempo percebido pelo usuário, já que este apenas recebe os e-mails gerados com o término da execução. Deste modo este caso não afeta o desempenho do sistema.

6.2.4 Teste de escalabilidade

A Tabela 1 – Tempos de Execução da Aplicação Intermediária mostra os dados obtidos pela execução caso proativo. Como pode ser observado na Figura 36 – Geração diária de relatório por empresa, a execução do caso proativo utilizado dispara a realização completa do relatório para 16 empresas diferentes na mesma execução. Além disso, ainda acrescenta o custo computacional de envio de e-mail de cada empresa, para cada usuário cadastrado para ser notificado para aquela empresa. Para 16 empresas se espera que um sistema escalável aumente o tempo de execução de forma linear, ou seja, o tempo esperado é próximo de 129,6 segundos. De acordo com a tabela, o sistema levou em média 131,3 segundos, um valor próximo ao esperado e proporcional ao incremento de carga.

6.2.5 Testes dos algoritmos de IA

Embora não seja objetivo deste trabalho avaliar as melhores técnicas de IA para a implementação do *softbot* pretendido, alguns testes foram realizados a fim de provar o desacoplamento entre as técnicas de inteligência e o restante do sistema. A realização desses testes se conduziu com substituição de algumas chamadas a classificadores implementados de forma clássica (se-então) por técnicas de IA. A escolha dessas técnicas se deu de forma subjetiva, sem compromisso em ser a melhor escolha possível.

Nas situações testadas, todos os algoritmos tiveram uma precisão superior a 99% nos casos reais. Embora esses resultados pareçam excelentes a primeira vista, em uma investigação foi identificado que o motivo da precisão alta independentemente do algoritmo utilizado se deu devido à característica dos dados nos cenários reais. Foi observado que os padrões de erro são basicamente os

mesmos para cada empresa cliente. Em outras palavras os padrões se repetem, implicando que os padrões avaliados eram os mesmos usados para treinamento.

Embora esses resultados sejam fracos ou até mesmos nulos do ponto de vista de IA, eles possuem uma grande relevância para esse trabalho e serão discutidos no próximo capítulo.

7 ASPECTOS DE AVALIAÇÃO E ANÁLISE DE RESULTADOS

Neste capítulo são apresentadas as avaliações, tanto do aspecto arquitetural, quanto do uso da solução proposta. O protótipo foi refinado ao longo de cinco iterações de desenvolvimento e avaliações dos usuários.

7.1 AVALIAÇÃO DA ARQUITETURA

A avaliação da arquitetura é feita baseada em propriedades. Foram considerados os atributos de qualidade para arquiteturas “clássicas” definidos em (BASS, CLEMENTS e KAZMAN, 2003) e os atributos de qualidades para arquiteturas orientadas a serviços definidos em (O’BIEN, MERSON e BASS, 2007). São eles:

1. *Interoperabilidade*: refere-se ao grau em que dois ou mais sistemas podem trocar informações por meio de interfaces em um dado contexto. Tanto a habilidade de trocar dados (interoperabilidade sintática) quanto a habilidade de interpretar esses dados trocados (interoperabilidade semântica) fazem parte da definição desse atributo. Esta propriedade é garantida principalmente pelo módulo de interfaces e pela utilização de padrões abertos como HTTP, REST, JSON, entre outros.

2. *Desempenho*: refere-se ao tempo e a capacidade do sistema em atender aos requisitos temporais, em outras palavras, retornar a resposta esperada a eventos (como interrupções, mensagens, etc.) dentro de limites de tempo estabelecidos como aceitáveis. Segundo testes de desempenho realizados e a experiência de uso dos usuários-chave evidenciada nas respostas ao questionário aplicado, a solução apresenta o desempenho esperado e dentro dos limites aceitáveis de utilização.

3. *Segurança*: refere-se à capacidade do sistema de permitir que pessoas autorizadas tenham acesso aos dados internos ao sistema e de proteger dados e informações contra acessos não autorizados. A utilização de uma aplicação intermediária garante essa propriedade. É utilizado o padrão OAuth 2.0 para autenticação e todas as integrações fazem uso de API-key.

4. *Confiabilidade*: refere-se à capacidade que um sistema possui em se manter operando sem falhas ao longo do tempo. Muito pode ser dito sobre confiabilidade de sistemas abordando diferentes aspectos. Para este trabalho, houve

a preocupação do ponto de vista computacional. Cada troca de mensagens entre serviços é validada antes de ser processada. Casos de falha são identificados e tratados, mantendo o sistema operando em um estado estável.

5. *Disponibilidade*: refere-se à capacidade que um sistema possui de estar pronto para realizar sua tarefa quando for requerido. A solução proposta segue padrões de integração e considera tecnologias amplamente usadas na indústria. Como mostrado nos testes onde múltiplos ambientes foram utilizados simultaneamente, a solução permite escalar horizontalmente por meio de réplicas, aumentando assim sua disponibilidade conforme desejado.

6. *Modificabilidade*: refere-se à capacidade que um sistema possui de sofrer alterações da forma mais rápida e econômica possível. A utilização de um AV baseado em SOA e serviços, assim como a arquitetura modular utilizada na aplicação intermediária garantem essa propriedade. Como visto na alteração de algoritmos de IA para testes. Além disso, as interfaces da aplicação intermediária criam um desacoplamento entre os sistemas envolvidos de forma que mudanças nos sistemas podem ser ajustadas nas interfaces.

7. *Testabilidade*: refere-se à capacidade que um sistema possui quanto à facilidade do estabelecimento de critérios de teste e a realização de testes para determinar se ditos critérios são atendidos. Dada a modularização inerente à utilização de serviços, a realização de testes unitários é facilitada e trivial. O grande desafio se dá nos testes integrados entre os sistemas. Contudo, o softbot se comunica por bate-papo com o usuário e por isso a comunicação entre sistemas é síncrona, ajudando na realização de testes.

8. *Usabilidade*: é uma medida da qualidade da experiência de um usuário na interação com informações ou serviços fornecidos pelo sistema. Normalmente mede a facilidade com que o usuário realiza uma tarefa desejada e o tipo de suporte que o sistema oferece. Conforme evidenciado pelo questionário, a Livia atende as expectativas dos usuários e possui a capacidade de dar suporte contextualizado as dúvidas dos mesmos. Mais sobre a usabilidade será explorado na próxima seção.

9. *Escalabilidade*: refere-se à capacidade que um sistema possui de funcionar sem degradação de outros atributos de qualidade em alterações de tamanho ou de volume de dados do sistema. A utilização de sistemas reais operando 24/7 impedem

a realização de alguns tipos de testes formais (como teste de carga), porém considerando os testes realizados, que mostram que o custo computacional aumenta de forma proporcional ao aumento da carga, o sistema proposto pode ser dito escalável.

7.2 AVALIAÇÃO DO USO

Para avaliar o uso do sistema, foram feitas comparações entre o cenário sem a utilização do AV com o cenário utilizando o AV. Essas comparações são embasadas pelos resultados dos testes e validações expostos no capítulo anterior.

7.2.1 Comparação temporal entre Livia e especialista

Como evidenciado nos testes de desempenho realizados, a Livia responde em 8 segundos aproximadamente, enquanto os analistas levam por volta de 2,5 horas por empresa analisada. Segundo os gerentes, este tempo pode ser dividido em três agrupamentos de tarefas. São eles:

- Identificação do que deve ser analisado
- Análise dos pontos identificados
- Escrita de relatórios

A Tabela 2 – Redução semanal de carga de trabalho por cliente compara os tempos gastos com e sem auxílio da Livia em cada agrupamento de tarefas. A coluna “Atual” apresenta os tempos gastos sem auxílio da Livia e foram retirados do sistema de gerência projetos e controle de horas da Harbor. A coluna “Livia” apresenta o tempo com a utilização do AV para realização de cada agrupamento de tarefas. A coluna “Redução” apresenta a redução percentual na carga de trabalho necessária para a realização do agrupamento de tarefas, ou

Tabela 2 – Redução semanal de carga de trabalho por cliente

Tarefa	Atual (min)	Livia (min)	Redução (%)
Identificação do que deve ser analisado	57	5	91
Análise dos pontos identificados	55	20	64
Escrita de relatórios	23	18	28
Total	135	43	68

Fonte: Elaborada pelo autor (2020)

Fica claro o ganho expressivo proporcionado pela inclusão da Livia no suporte aos PMaaS. A redução de 68% na carga de trabalho por empresa cliente triplica a capacidade de análise dos especialistas ou proporciona uma diminuição na pressão sobre esses profissionais, aumentando assim a qualidade do seu trabalho. É importante ressaltar que essa comparação levou em conta a implementação atual da Livia, onde nem todas as análises são realizadas e os analistas ainda não possuem confiança suficiente para deixarem de realizar uma dupla checagem das análises feitas de forma automática. Espera-se um ganho ainda maior com a continuidade do projeto na Harbor.

7.2.2 Comparação de qualidade da análise

A Livia também apresenta melhorias na qualidade das análises realizadas pelos especialistas.

7.2.2.1 Profundidade da análise

Todos os dados são analisados e avaliados quanto a sua integridade pela Livia. Todos os dados, de todos os centros de trabalho, para todos os produtos, em todos os turnos e associados a todos os motivos. Embora toda essa estratificação seja teoricamente possível de forma manual, é impraticável.

Devido a grande quantidade de centros de trabalhos cadastrados e a produção de uma variedade muito grande de produtos em algumas empresas clientes, os especialistas escolhem até quatro centros de trabalho para realizar a sua análise semanal. A escolha dessas estações se dá de forma empírica, levando em conta o grau de importância dos centros segundo os clientes e a experiência profissional do especialista.

7.2.2.2 Integridade da análise

Ainda que selecionem apenas alguns centros de trabalho, os especialistas não conseguem olhar para todos os dados dessas estações e realizam suas análises agrupando categorias de dados. O problema relacionado a isto é o fato de os especialistas não conseguirem visualizar pontos fora da curva que foram agrupados juntamente com dados “corretos”. Esses pontos fora da curva alteram as médias calculadas de forma aleatória.

A Livia pode realizar a análise nos dados “sujos” originais do sistema para manter a coerência com os dados exibidos no MES, mas também é capaz de filtrar os dados para eliminar todos os pontos fora da curva identificados.

7.3 AVALIAÇÃO DA IA

Conforme descrito no capítulo anterior, os resultados obtidos de precisão total em todos os algoritmos testados se deram devido à repetitividade dos dados nas séries avaliadas pelos algoritmos quando comparadas aos dados de treinamento. Isso ocorreu mesmo com técnicas de randomização dos dados de treinamento.

Ainda assim esse resultado é bastante expressivo para esse trabalho, pois evidencia a frequência com que os mesmos problemas são encontrados nos mesmos clientes, semana após semana. Esse dado deixa claro o quanto cansativo e maçante é o trabalho realizado de forma manual para de identificar os mesmos problemas constantemente. Dessa forma justifica a tentativa de automatização dessas análises conforme a proposta deste trabalho.

Uma outra evidência relevante é que empresas que estão se mantendo constantes em um mesmo nível de maturidade, de forma geral, não dependem de algoritmos sofisticados ou de alto custo computacional para realizar análises, uma vez que tendem a manter os mesmos problemas e algoritmos simples são capazes de gerar bons resultados.

Certamente mais estudos precisam ser realizados, envolvendo mais cenários e mais algoritmos, porém tais estudos serão realizados em trabalhos futuros.

8 CONSIDERAÇÕES FINAIS

Esse trabalho apresentou uma pesquisa de como *softbots* podem ajudar gestores de produção em suas análises rotineiras. A abordagem utilizada considera a criação de um software de suporte a decisão e sua disponibilização na forma software como um serviço (SaaS), para que possa ser usada remotamente por gerentes fora do ambiente de fabril, chamados Gerentes de Produção como Serviço (PMaaS).

O intuito desta pesquisa foi agir junto a uma comunidade de gerentes de produção remotos, solucionando, ou ao menos atenuando, os problemas mencionados. A comunidade alvo é composta por colaboradores da empresa Harbor que atuam como PMaaS.

Como esperado em uma pesquisa com este nível de complexidade, algumas dificuldades precisaram ser resolvidas para dar continuidade aos trabalhos.

8.1 DIFICULDADES ENCONTRADAS

Dada a dimensão e complexidade deste trabalho, dificuldades surgiram e foram superadas ao longo de toda a pesquisa.

Muitas das linguagens e tecnologias utilizadas não eram de conhecimento do autor no começo do projeto e foram aprendidas juntamente com a criação do protótipo. Essa evolução no aprendizado se refletiu no protótipo de modo que, por mais de uma vez, a solução foi remodelada quase que inteiramente, para torná-la mais robusta e aprimorar alguma característica desejada (legibilidade, performance, escalabilidade, etc).

A integração com sistemas reais e em constante desenvolvimento também gera dificuldades. Ao longo da implementação, tanto a plataforma Arisa quanto o sistema LiveMes, tiveram atualizações. Algumas vezes essas atualizações alteraram as funcionalidades, outras vezes protocolos de comunicação ou autenticação e também houveram atualizações que mudaram o modelo de dados e API de comunicação. No último caso, motivou uma remodelação do protótipo para gerar um maior desacoplamento entre o modelo de dados interno do projeto e o modelo de dados do MES, aprimorando e redefinindo o módulo de interface com o MES.

A dificuldade mais inesperada durante o projeto está relacionada a criação de diálogos intuitivos, coerentes, naturais, autoexplicativos e sobretudo práticos. Toda a árvore de conversação foi remodelada diversas vezes em busca de diálogos grandes o bastante para explicar toda a informação necessária de forma natural; porém pequenos o bastante para não serem cansativos, já que deveriam ser enviados por aplicativo de *chat*. Além da dificuldade intrínseca de construir os diálogos e contextos relevantes, uma mudança na forma com que os dados são entregues ao usuário faz com que seja necessária uma mudança nos *scripts* que buscam esse dado, nas APIs que disponibilizam esse dado e algumas vezes, no processamento dos dados, chegando até a consulta dos dados no MES.

Após a solução das dificuldades encontradas, foi possível finalizar o protótipo e realizar os testes de validação da solução. Com os resultados dos testes e juntamente com a opinião dos usuários, foi possível responder a pergunta de pesquisa.

8.2 RESPOSTA À PERGUNTA DE PESQUISA

Esta pesquisa se propôs a responder a seguinte pergunta apresentada no capítulo 1.3:

Como desenvolver um softbot que se integre a e interopere com sistemas existentes num modo SaaS, que faça automaticamente análises de dados de produção, bem como interaja com os gestores para o fornecimento de informações específicas visando ajudá-los em tomadas de decisão?

Considerando o objetivo e desenvolvimento deste trabalho como Pesquisa-Ação, a pergunta foi respondida com a implementação de um protótipo baseado na proposta de arquitetura de referência e que visava atender a um conjunto de requisitos para introduzir melhorias numa situação específica e concreta da empresa Harbor. A implementação seguiu o estilo arquitetural baseado em serviços e o modelo de referência, e se utilizou de um estudo de caso para a criação dos comportamentos do *softbot*.

8.3 CONCLUSÕES

Esse trabalho uniu aspectos teóricos e práticos de gerenciamento de produção, em especial em PMEs. Para quantificar o avanço de uma empresa-cliente na direção da Indústria 4.0 foi proposta a utilização de um modelo de maturidade simplificado. Este modelo considera apenas dados básicos de produção e pontua critérios definidos distribuídos em quatro níveis de maturidade. A quantificação da maturidade da empresa torna os problemas mais visíveis e direciona as ações de correção.

A arquitetura modular proposta foca em adaptabilidade e integração com um software MES real disponibilizado em nuvem. Esta arquitetura utiliza uma plataforma para a instanciação de um *softbot* e uma aplicação intermediária que adiciona capacidades analíticas e cognitivas ao *softbot*, além de adicionar uma camada de segurança na proteção dos dados das empresas clientes. Tanto o comportamento reativo quanto o proativo fazem parte da proposta. Este *softbot* funciona como um módulo opcional ao software MES e realiza diferentes tipos de análises de negócio, assim como faz uso de diferentes técnicas de inteligência artificial para diagnosticar, prever e prescrever.

A utilização de uma arquitetura distribuída e modular baseada em tecnologias conhecidas e consolidadas se mostrou adequada e capaz de dividir o processamento dos dados para garantir uma boa performance e manter a segurança dos dados. A estratégia de implementação foi baseada em ciclos, consultando os usuários a cada iteração, o que se mostrou importante no progresso de cada versão do sistema.. Um desacoplamento entre a coleta e processamento dos dados com interface e diálogos com os usuários reduz de forma significativa o tempo gasto com as alterações necessárias para um ajuste fino. Gestores, em geral, não são programadores e possuem um modo de pensar diferente de quem constrói a solução, isso faz com que os diálogos mais “lógicos” não façam sentido a alguns usuários. Diálogos extensos se tornam cansativos rapidamente, por isso a forma de utilização e a frequência com que os diálogos serão acionados deve ser considerada. Casos para dúvidas ou consultas

pontuais podem ser tratados através de conversação via chatbot, porém para casos de uso em que muitas informações precisem ser transmitidas a uma frequência constante, a utilização de relatórios completos de forma proativa é mais eficaz e melhor aceita pelos usuários.

Foi gerado um fluxo de conversa capaz de realizar trocas rápidas de contextos reduzindo as chances do *softbot* não entender o usuário e com isso diminuindo frustrações em conversas com a Livia.

Apenas padrões abertos foram utilizados no desenvolvimento da aplicação intermediária, de forma a facilitar a interoperabilidade entre componentes e a Livia com outros sistemas. As capacidades analíticas da Livia utilizaram algoritmos procedurais, sistemas especialistas e outras técnicas de IA para validação. Graças a sua modularidade, a solução permitiu a implementação direcionada e o incremento gradual na complexidade dos algoritmos de análise. Um aumento de complexidade dos algoritmos de IA se mostraram não serem necessários nos ambientes reais testados, já que mesmo algoritmos simples possuem alta precisão devido à repetitividade dos dados.

Testes realizados comprovaram que a arquitetura proposta é aberta, flexível e escalável. Após testes de qualidade em ambiente de testes, o protótipo foi integrado ao ambiente produtivo real do LiveMES e a Livia passou a acessar e analisar os dados de dezenas de empresas clientes em tempo real. Para uma validação completa do atendimento dos objetivos da pesquisa, também foram realizadas entrevistas com aplicação de um questionário aos usuários.

Antes da utilização da Livia, as análises realizadas levavam cerca de 2,5 horas por empresa. Com a implementação da Livia, ainda que em um estágio inicial, foi possível uma redução de 68% no tempo necessário para a mesma análise. Mais do que redução no tempo, a Livia também aumenta muito a qualidade da análise, pois a sua capacidade de avaliar a integridade dos dados possibilita a eliminação quase completa da influência de pontos fora da curva que antes não eram detectados.

Ainda como ganho proporcionado pela solução, é considerada a quantidade de centros de trabalho analisados por empresa. Enquanto manualmente são escolhidos por volta de 4 principais centros de trabalho para análise, a Livia analisa *todos* os centros da empresa, proporcionando uma análise mais completa. Dadas as

capacidades proativas da Livia, relatórios podem ser gerados automaticamente, ou problemas no chão de fábrica podem ser alertados em tempo real.

Avaliações feitas pelos usuários foram muito positivas e confirmaram que a ação proposta atende aos objetivos propostos. Os usuários se mostraram bastante empolgados com o projeto e ansiosos para a sua ampliação.

A implementação da Livia tem mostrado resultados bastante promissores, porém ainda é um produto em fase inicial e ainda possui limitações. O fluxo de conversas ainda apresenta dificuldade inicial para usuários não treinados. Algumas análises ainda utilizam algoritmos procedurais e limitados. Nem todas as análises do modelo de maturidade foram implementadas.

8.4 TRABALHOS FUTUROS

Considerando a abrangência do tema e também as limitações e delimitações expostas ao longo do texto, surgem ideias de trabalhos futuros para melhorar a arquitetura proposta. Entre elas:

- Estudo de diferentes técnicas de organização de contextos em casos de uso reais e complexos com o objetivo de compreender melhor os diálogos não-lineares.
- Ampliar os estudos de diferentes técnicas de IA na implementação dos comportamentos preditivo e prescritivo de forma a ampliar as capacidades cognitivas do *softbot*.
- Estudo e implementação de detecção de sentimentos no *softbot*, tentando personalizar ainda mais a experiência do usuário de chão de fábrica e prever um potencial período de desconcentração do usuário, que levaria a uma maior quantidade de problemas na produção.
- Aprofundar o estudo sobre a melhoria de performance dos gestores de produção remotos e aplicar o conceito a outros profissionais, como por exemplo, profissionais do setor comercial.

- Avaliar as implicações da adição da capacidade de ação do *softbot* para correção automática de problemas na linha.

REFERÊNCIAS

ABDUL-KADER, S. A.; WOODS, J. C. Survey on chatbot design techniques in speech conversation systems. **International Journal of Advanced Computer Science and Applications**, v. 6, n. 7, 2015.

ABNER, B. et al. **Production Management as-a-Service: A Softbot Approach**. Towards Smart and Digital Manufacturing, IFIP AICT. [S.l.]: Springer. 2020. p. 19-30.

ALFF, F. R. **Análise de Requisitos**. [S.l.]: [s.n.], 2018.

ALLAMARJU, S. **RESTful Web Services Cookbook**. 1. ed. [S.l.]: O'Reilly, 2010.

ASG. The ASG Technologies 2020 Survey Report – Barriers to Success with Information-Powered Business: Bad Data Burdens the Modern Enterprise. **ASG**, 2020. Disponível em: <<https://content.asg.com/MarketingCollateral/DataIntelligence/whitepaper-asg-data-intelligence-barriers-to-success.pdf>>. Acesso em: 30 Julho 2020.

ATKINSON, J. Manpower strategies for flexible organisations. **Personnel management**, v. 16, n. 8, p. 28-31, 1984.

AUFFARTH, B. **Artificial Intelligence with Python Cookbook: Proven recipes for applying AI algorithms and deep learning techniques using TensorFlow 2.x and PyTorch 1.6**. 1ª. ed. [S.l.]: Packt Publishing, 2020.

BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software architecture in practice**. [S.l.]: Addison-Wesley Professional, 2003.

BITTENCOURT, G. **Inteligência Artificial**. 2ª. ed. Florianópolis: Editora da UFSC, 2001.

BRACHMAN, R.; LEVESQUE, H. **Knowledge Representation and Reasoning**. [S.l.]: Elsevier, 2004.

BRAIN. Chatbot Report 2019: Global Trends and Analysis, 2019. Disponível em: <<https://chatbotmagazine.com/chatbot-report-2019-global-trends-and-analysis-a487afec05b/>>. Acesso em: 30 Julho 2020.

BRERETON, P. et al. **Lessons from applying the systematic literature review process within the software engineering domain**. JSS 80. [S.I.]: [s.n.]. 2007. p. 571-583.

CAMARINHA-MATOS, L. M.; FORNASIERO, R.; AFSARMANESH, H. **Collaborative Networks as a Core Enabler of Industry 4.0**. IFIP AICT. [S.I.]: Springer. 2017. p. 3-17.

CHEN, X. et al. Human-centred Adaptation and Task Distribution utilizing Levels of Automation. **IFAC PapersOnLine**, v. 51, p. 54-59, 2018.

CHRISSIS, M. B.; KONRAD, M.; SHRUM, S. **CMMI for development: guidelines for process integration and product improvement**. [S.I.]: Pearson Education, 2011.

CNI. **CNI**, 2016. Disponível em: <http://www.ipt.br/download.php?filename=1436-desafios_para_a_industria_40_no_Brasil_CNI_2016.pdf>.

COHEN, M. A.; WHANG, S. Competing in Product and Service: A Product Life-Cycle Model. **Management Science**, v. 43, p. 535-545, Abril 1997.

COULOURIS, G. et al. **Sistemas Distribuídos: Conceitos e Projeto**. 5ª. ed. [S.I.]: Bookman Editora, 2013.

CREVIER, D. **AI: The Tumultuous Search for Artificial Intelligence**. New York: BasicBooks, 1993.

DERSINGH, A.; SRISAKULPINYO, P. **Chatbot and Visual Management in Production Process**. Conference on Electronics, Information and Communication. [S.I.]: [s.n.]. 2017. p. 274-277.

D'SILVA, G. M. et al. **Real world smart chatbot for customer care using a software as a service (SaaS) architecture**. International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud). [S.I.]: [s.n.]. 2017. p. 658-664.

ESTEFAN, J. A. et al. Reference Model for Service Oriented Architecture Version 1.0. **OASIS**, 2006. Disponível em: <<http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>>. Acesso em: 07 Dezembro 2020.

FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. University of California. Irvine. 2000.

FRANZZON, E. M. et al. Manufacturing Networks in the Era of Digital Production and Operations: A Socio-Cyber-Physical perspective. **Annual Reviews in Control**, 49, 2020. 288-294.

GESSER, C. E. **Uma abordagem para a integração dinâmica de serviços web em portais**. Eng. Elétrica - UFSC. Florianópolis. 2006.

GIL, A. C. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: [s.n.], 2010.

GÜNTHER, H. **Série Planejamento de pesquisa nas ciências sociais**. 1. ed. Brasília: Laboratório de Pesquisa Ambiental, 2003.

HERMANN, M.; PENTEK, T.; OTTO, B. **Design principles for industry 4.0 scenarios**. Proceedings of the 49th IEEE Hawaii International Conference on System Sciences. [S.I.]: [s.n.]. 2016. p. 3928–3937.

IERUSALIMSKY, R. **Programando em Lua**. 3. ed. [S.I.]: LTC, 2015. Disponível em: <<https://www.lua.org/portugues.html>>.

JACKSON, P. **Introduction To Expert Systems**. 3^a. ed. [S.I.]: Addison Wesley, 1998.

JÄRVINEN, P. Action Research is Similar to Design Science. **Quality & Quantity**, fev 2007. 37-54.

KAR, R.; HALDAR, R. Applying Chatbots to the Internet of Things: Opportunities and Architectural Elements. In: J., I. **Advanced Computer Science and Applications**. 7(11). ed. [S.I.]: [s.n.], 2016. p. 147-154.

KASSNER, L. et al. **The Social Factory**: Connecting People, Machines and Data in Manufacturing for Context-Aware Exception Escalation. Proceed-ings 50th Hawaii International Conference on System Sciences. [S.I.]: [s.n.]. 2017.

KASTALLI, I. V.; LOOY, B. V. Servitization: Disentangling the impact of service business model innovation on manufacturing firm performance. **Journal of Operations Management**, v. 31, p. 169-180, 2013.

KAUSHIK, S. **Production and Materials Management**. Centrum Press. [S.I.]: [s.n.]. 2014.

KIM, J. H. Ubiquitous Robot. **Computational Intelligence, Theory and Applications**. **Advances in Soft Computing**, v. 33, 2005.

KINDSTRÖM, D. Towards a service-based business model – Key aspects for future competitive advantage. **European Management Journal**, v. 28, p. 479-490, Dezembro 2010.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. [S.l.]: [s.n.], 2007.

KLETTI, J. **Manufacturing Execution System-MES**. Springer Science & Business Media. ed. [S.l.]: [s.n.], 2007.

KUSIAK, A. Smart Manufacturing. **International Journal of Production Research**, v. 56, n. 1-2, p. 508-517, 2018.

KWON, O.; LEE, N.; SHIN, B. Data quality management, data usage experience and acquisition intention of big data analytics. **International Journal of Information Management**, 2014. 387-394.

LAMNABHI-LAGARRIGUE, F.; ANNASWAMY, A.; ENGELL, S. Systems & Control for the Future of Humanity, Research Agenda: Current and Future Roles, Impact and Grand Challenges. **Annual Reviews in Control**, 43, 2017. 1-64.

LEE, J.; BAGHERI, B.; KAO, H. A. **Recent advances and trends of cyber-physical systems and**. Industrial Informatics. IEEE. [S.l.]: [s.n.]. 2014.

LIAO, Y. et al. Past, Present and Future of Industry 4.0 - A Systematic Literature Review and Research Agenda Proposal. **International Journal of Research Production Research**, v. 55, n. 12, p. 3609-3629, 2017.

LIKERT, R. A Technique for the Measurement of Attitudes. **Archives of Psychology**, 1932. 1-55.

LONGO, F.; NICOLETTI, L.; PADOVANO, A. Smart Operators in Industry 4.0: A Human-centered Approach to Enhance Operators' Capabilities and Competencies within the New Smart Factory Context. **Computers & Industrial Engineering**, v. 113, p. 274-277, 2017.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115-133, 1943.

MCDERMOTT, A. Information Overload Is Crushing You. Here are 11 Secrets That Will Help, 2017. Disponível em: <<https://www.workzone.com/blog/information-overload/>>.

MENDEZ, J. D. et al. A Plug and Play Integration Model for Virtual Enterprises. **IFIP AICT**, 534, 2018. 312-324.

MITCHELL, T.; HILL, M. **Machine Learning**. [S.l.]: [s.n.], 1997.

MITTAL, S. et al. Smart manufacturing: characteristics. **J. Eng. Manuf**, 2017. Disponivel em: <<https://doi.org/10.1177/09544054>>.

MITTAL, S. et al. A critical review of smart manufacturing & Industry 4.0 maturity models: Implications for small and medium-sized enterprises (SMEs). **Journal of Manufacturing Systems**, 2018. 194-214.

NEELY, A. Exploring the financial consequences of the servitization of manufacturing. **Operations Management Research**, v. 1, p. 103–118, 2008.

NEELY, A. Exploring the financial consequences of the servitization of manufacturing. **Operations Management Research**, v. 1, p. 103-118, 2008.

O'BRIEN, L.; MERSON, P.; BASS, L. **Quality attributes for service-oriented architectures**. Proceedings of the international Workshop on Systems Development in SOA Environments. [S.l.]: [s.n.]. 2007.

PAPAZOGLU, M. **Web Services & SOA Principles and Technology**. 2. ed. [S.l.]: Pearson Prentice Hall, 2012.

PÉREZ, P. et al. **ChatPy**: Conversational agent for SMEs. 14th Iberian Conference on Information Systems and Technologies (CISTI). [S.l.]: [s.n.]. 2019. p. 1-6.

POCHIRAJU, B.; SESHADRI, S. **Essentials of Business Analytics: An Introduction to the Methodology and its Applications**. [S.l.]: Springer. 2019.

POOLE, D.; MACKWORTH, A. **Artificial Intelligence: Foundations of Computational Agents**. 2^a. ed. [S.l.]: Cambridge University Press, 2017.

RABELO, R. J. . R. D. . Z. S. P. **Softbots Supporting the Operator 4.0 at Smart Factory Environments**. IFIP AICT. [S.l.]: Springer. 2018. p. 456-464.

RABELO, R. J. . Z. S. P. . R. D. **Collaborative Softbots: Enhancing Operational Excellence in Systems of Cyber-Physical Systems**. IFIP AICT. [S.l.]: Springer. 2019. p. 55-68.

RABELO, R. J.; ROMERO, D.; ZAMBIASI, S. P. **Softbots Supporting the Operator 4.0 at Smart Factory Environments**. IFIP AICT. [S.l.]: Springer. 2018. p. 456-464.

RABELO, R. J.; ZAMBIASI, S. P.; ROMERO, D. **Collaborative Softbots: Enhancing Operational Excellence in Systems of Cyber-Physical Systems**. IFIP AICT. [S.l.]: Springer, 2019. p. 55-68.

ROMERO, D. et al. The operator 4.0: human cyber-physical systems and adaptive automation towards human-automation symbiosis work systems. In: NÄÄS, I. . E. A. (.). **APMS**. Cham: Springer, v. 488, 2016. p. 677-686. Disponível em: <https://doi.org/10.1007/978-3-319-51133-7_80>.

ROMERO, D. et al. **Towards an operator 4.0 typology: a human-centric perspective on the fourth industrial**. Proceedings of the International Conference on Computers and. [S.l.]: [s.n.]. 2016. p. 1–11.

ROMERO, D. et al. Social factory architecture: social networking services and production scenarios through the social internet of things, services and people for the social operator 4.0. In: LÖDDING, H. . R. R. . T. K.-D. . V. C. G. . K. D. (.). **APMS 2017. IAICT**. Cham: Springer, v. 513, 2017. p. 265–273.

ROMERO, D.; VERNADAT, F. Enterprise Information Systems State of the Art: Past, Present and Future Trends. **Computer in Industry. No. 79**, 2016. 3-13.

ROTEM-GAL-OZ, A.; BRUNO, E.; DAHAN, U. **SOA patterns**. Shelter Island: Manning, 2012.

RUSSEL, S.; NORVIG. **Inteligência Artificial 3ª Ed. Tradução da terceira edição**. Rio de Janeiro: Editora Campus / Elsevier, 2013.

RUSSEL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 4ª. ed. [S.l.]: [s.n.], 2020.

SCHIAFFINO, S.; AMANDI, A. Polite Personal Agents. **IEEE Intelligent Systems**, Jan-Fev 2006. 12-18.

SCHULTE, W. R.; NATIS, Y. V. **Service Oriented Architecture, Part 1**. [S.l.]: [s.n.], 1996.

SCHWARTZ, T. et al. Hybrid Teams: Flexible Collaboration between Humans, Robots and Virtual Agents. **LNCS**, v. 9872, p. 131-146, 2016.

SHAWAR, B. A.; ATWELL, E. **Chatbots: are they really useful?** Ldv forum. [S.l.]: [s.n.]. 2007. p. 29-49.

SOMMERVILLE, I. **Engenharia de Software**. 8ª. ed. [S.l.]: Addison-Wesley, 2007.

STOJANOVIC, Z.; DAHANAYAKE, A. **Service-oriented software system engineering: challenges and practices**. Hershey: Idea Group Inc, 2005.

SUSARLA, A.; BARUA, A.; WHINSTON, A. B. A Transaction Cost Perspective of the "Software as a Service" Business Model. **Journal of Management Information Systems**, v. 26, n. 2, p. 205-240, 2009.

TANENBAUM, A. S. **Distributed Systems: Principles and paradigms**. [S.I.]: [s.n.].

VAISHNAVI, V.; KUECHLER, B.; PETTER, S. **Design Science Research in Information Systems**. [S.I.]: [s.n.], 2004.

VANDERMERWE, S.; RADA, J. Servitization of business: Adding value by adding services. **European Management Journal**, v. 6, p. 314-324, 1988.

VUKŠIĆ, V. B. . B. M. P. . P. A. Supporting Performance Management with Business Process Management and Business Intelligence: A Case Analysis of Integration and Orchestration. **International Journal of Information Management**, 2013. 613-619.

VUKŠIĆ, V. B.; BACH, M. P.; POPOVIČ, A. Supporting Performance Management with Business Process Management and Business Intelligence: A Case Analysis of Integration and Orchestration. **International Journal of Information Management**, 2013. 613-619.

W3C. Web Services Architecture. **W3C**, 2004. Disponível em: <<https://www.w3.org/TR/ws-arch/>>. Acesso em: 07 Dezembro 2020.

WALLACE, R. **The Elements of AIML Style**. [S.I.]: A.L.I.C.E. Artificial Intelligence Foundation, Inc., 2003.

WEIZENBAUM, J. Eliza a computer program for the study of natural language communication between man and machine. **Communications of the ACM**, v. 9, n. 1, p. 36-45, 1966.

WMF. WORLD MANUFACTURING FORUM REPORT - Skills for the Future of Manufacturing. **WMF**, 2019. Disponível em: <<https://www.worldmanufacturingforum.org/report-2019>>.

ZAMBIASI, S. P. **Uma arquitetura de referência para softwares assistentes pessoais baseada na arquitetura orientada a serviços**. PGEAS-UFSC. Florianópolis. 2012.

ZAMBIASI, S. P.; RABELO, R. J. **A proposal for reference architecture for personal assistant**. IEEE Latin Am. Trans. [S.l.]: [s.n.]. 2012. p. 1227–1234.

ZAMBIASI, S. P.; RABELO, R. J. Softbots Supporting the Operator 4.0. **IEEE Latin America Transactions**, n. 10(1):1227-1234, 2012.

ZAMBIASI, S. P.; RABELO, R. J. **Arisa Nest – Uma Plataforma Baseada na Nuvem para Desenvolvimento de Assistentes Virtuais**. 13th Workshop-School on Agents, Environments, and Applications. Florianópolis: [s.n.]. 2019.

ZAMBIASI, S. P.; RABELO, R. J. Arisa Nest – A Cloud-Based Platform for Development of Virtual Assistant. **Revista de Informática Teórica e Aplicada - RITA**, v. 27, n. 2, p. 116-126, 2020.