



Sistemas Ciberfísicos orientados para a Produção Aberta

NUNO JOEL DA ROCHA PINTO
dezembro de 2021

SISTEMAS CIBERFÍSICOS ORIENTADOS PARA A PRODUÇÃO ABERTA

Nuno Joel Rocha Pinto
1150569

2021

Instituto Superior de Engenharia do Porto
Engenharia Mecânica

SISTEMAS CIBERFÍSICOS ORIENTADOS PARA A PRODUÇÃO ABERTA

Nuno Joel Rocha Pinto
1150569

Dissertação apresentada ao Instituto Superior de Engenharia do Porto para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob a orientação do Professor Doutor Hélio Alves de Castro e coorientação do Professor Doutor Filipe de Sousa Pereira e do Professor Doutor João de Sousa Bastos.

2021

Instituto Superior de Engenharia do Porto
Engenharia Mecânica

JÚRI

Presidente

Doutor Paulo da Silva Ávila

Professor Coordenador, Instituto Superior de Engenharia do Porto

Orientador

Doutor Hélio Alves de Castro

Professor Adjunto, Instituto Superior de Engenharia do Porto

Coorientadores

Doutor Filipe de Sousa Pereira

Professor Adjunto Convidado, Instituto Superior de Engenharia do Porto

Doutor João de Sousa Bastos

Professor Adjunto, Instituto Superior de Engenharia do Porto

Arguente

Doutor Nuno Ferreira Lopes

Professor Adjunto, Instituto Politécnico do Cávado e do Ave

PALAVRAS-CHAVE

Sistema Ciberfísico; CPS; *Open Design*; Indústria 4.0; *Cloud Computing*; Sistema de Multi-Agentes; *IoT*; *Smart Factory*; *Industrial Internet of Things (IIoT 4.0)*; *Cloud Manufacturing*; *open source*; *low-cost*; Laboratório de Ciência Aberta

RESUMO

O objetivo principal deste trabalho passa por estabelecer um sistema ciberfísico orientado para a comunidade no âmbito da produção aberta, recorrendo, então, ao *Open Design*. Nesse contexto, numa primeira fase, é feita uma revisão bibliográfica sobre estes conceitos, da qual não foram identificados quaisquer trabalhos já realizados que abrangessem tal finalidade. Mesmo assim, e com base no material obtido dessa revisão, juntamente com documentação relativa a trabalhos enquadrados no tema, define-se uma arquitetura, por camadas, que satisfaz o pretendido e enquadra-se na premissa *open-source* e/ou *low-cost*.

Efetivamente, é elaborado um sistema que permite a qualquer utilizador o acesso livre a um repositório onde se encontra disponível uma vasta gama de produtos, personalizados ou não, os quais são facultados pela própria comunidade. Para que o processamento da informação congruente seja realizável sem limitações, a maior parte do mesmo está alocada na *cloud*, o qual será auxiliado por um Sistema de *Multi-Agentes*. De seguida, estabelece-se uma ponte de ligação com o mundo físico, por meio de *interfaces CPS* viabilizadas por dispositivos *IoT*, os quais conectam e controlam recursos físicos simbólicos à Manufatura Aditiva e Subtrativa, com a finalidade de produzir o produto pretendido. Nesta fase, é possibilitado ao operador e à comunidade acompanhar em tempo real todas as ações e operações realizadas pelos recursos físicos referidos. Juntamente com estes últimos intervenientes, existem igualmente máquinas e processos a cooperar e colaborar simbioticamente para o refinamento do processo de produção, através da análise e recálculo dos dados relacionados com o mesmo.

De forma a concretizar esta arquitetura, a implementação da mesma é iniciada, primeiramente a nível físico e, posteriormente, a nível computacional/digital.

KEYWORDS

Cyber-Physical System; CPS; *Open Design*; Industry 4.0; *Cloud Computing*; *Multi-Agent System*; *IoT*; *Smart Factory*; *Industrial Internet of Things (IIoT 4.0)*; *Cloud Manufacturing*; open source; low-cost; Open Science Laboratory

ABSTRACT

The main objective of this work is to establish a cyberphysical system oriented to the community within the scope of open production, resorting, then, to *Open Design*. In this context, in a first phase, a bibliographic review of these concepts is carried out, from which no work that had already been carried out covering this purpose was identified. Even so, and based on the material obtained from this review, together with documentation related to works framed in the theme, an architecture is defined, in layers, which satisfies the intended purpose and falls under the open-source and/or low-cost premise.

In fact, a system is created that allows any user free access to a repository where a wide range of products, personalized or not, are available, which are provided by the community itself. For the processing of congruent information to be feasible without limitations, most of it is allocated in the cloud, a process better known as *Cloud Computing*, which will be supported by a *Multi-Agent System*. Then, a connection bridge with the physical world is established, through CPS interfaces made possible by IoT devices, which connect and control symbolic physical resources to the Additive and Subtractive Manufacturing, to produce the intended product. At this stage, it is possible for the operator and the community to monitor in real time all actions and operations carried out by the aforementioned resources. Along with these last participants, there are also machines and processes to cooperate and collaborate symbiotically for the refinement of the production process, through the analysis and recalculation of the data related to it.

To realize this architecture, its implementation is started, firstly at the physical level and, later, at the computational/digital level.

LISTA DE SÍMBOLOS E ABREVIATURAS

Lista de Abreviaturas

<i>ABS</i>	<i>Acrylonitrile Butadiene Styrene</i>
<i>API</i>	<i>Application Programming Interface</i>
<i>AR</i>	<i>Augmented Reality</i>
<i>BLE</i>	<i>Bluetooth Low Energy</i>
<i>BOM</i>	<i>Bill of Materials</i>
<i>CAD</i>	<i>Computer-Aided Design</i>
<i>CAM</i>	<i>Computer-Aided Manufacturing</i>
<i>CLA</i>	<i>Coalition Leader Agent</i>
<i>CNC</i>	Controlo Numérico Computorizado
<i>CPS</i>	<i>Cyber Physical System</i>
<i>CPS-AMS</i>	<i>Cyber Physical System - Augmented Manufacturing Systems</i>
<i>CPS-VMS</i>	<i>Cyber Physical System - Virtual Manufacturing Systems</i>
<i>DA</i>	<i>Deployment Agent</i>
<i>DBMS</i>	<i>Data Base Management System</i>
<i>DEC</i>	<i>Discrete Event Calculus</i>
<i>EEPROM</i>	<i>Electrically Erasable Programmable Read-Only Memory</i>
<i>FIPA</i>	<i>Foundation for Intelligent Physical Agents</i>
<i>GPIO</i>	<i>General Purpose Input/Output</i>
<i>GUI</i>	Interface Gráfica do Utilizador
<i>HIPS</i>	<i>High Impact Polystyrene</i>
<i>HTTP</i>	<i>Hypertext Transfer Protocol</i>
<i>I-CPS</i>	<i>Intelligent Cyber Physical System</i>
<i>ICSP</i>	<i>In Circuit Serial Programming</i>
<i>IDE</i>	<i>Integrated Development Environment</i>
<i>IIoT</i>	<i>Industrial Internet of Things</i>
<i>I/O</i>	<i>Input/Output</i>
<i>IoT</i>	<i>Internet of Things</i>
<i>I²C</i>	<i>Inter-Integrated Circuit</i>
<i>JADE</i>	<i>Java Agent Development Framework</i>
<i>JPEG</i>	<i>Joint Photographic Experts Group</i>
<i>JSP</i>	<i>JavaServer Pages</i>
<i>LAN</i>	<i>Local Area Network</i>
<i>MA</i>	<i>Multi-Agents</i>
<i>MAS</i>	<i>Multi-Agent System</i>

<i>M-CPS</i>	<i>Mixed Cyber Physical System</i>
<i>MDE</i>	<i>Model-driven Development</i>
<i>MP</i>	<i>MegaPixel</i>
<i>NC</i>	Controlo Numérico
<i>NBLoT</i>	<i>Narrowband Internet of Things</i>
<i>OD</i>	<i>Open Design</i>
<i>OSD</i>	<i>Open Source Definition</i>
<i>OSH</i>	<i>Open Source Hardware</i>
<i>OSS</i>	<i>Open Source Software</i>
<i>PA</i>	<i>Product Agent</i>
<i>PaaS</i>	<i>Platform as a Service</i>
<i>PC</i>	<i>Polycarbonate</i>
<i>PCB</i>	<i>Printed Circuit Board</i>
<i>P&D</i>	Pesquisa e Desenvolvimento
<i>PETG</i>	<i>Polyethylene Terephthalate Glycol</i>
<i>PLA</i>	<i>Polylactic Acid</i>
<i>PLC</i>	<i>Programmable Logic Controller</i>
<i>PP</i>	<i>Polypropylene</i>
<i>PS/MS</i>	<i>Production/Manufacturing Systems</i>
<i>PVA</i>	<i>Polyvinyl Acetate</i>
<i>PVC</i>	<i>Polyvinyl chloride</i>
<i>PWM</i>	<i>Pulse-Width Modulation</i>
<i>RA</i>	<i>Resource Agent</i>
<i>RAM</i>	<i>Random Access Memory</i>
<i>RCS</i>	Sistema de controlo de resiliência
<i>REST</i>	<i>Representational State Transfer</i>
<i>RFID</i>	<i>Radio Frequency Identification</i>
<i>RTC</i>	<i>Real-time Clock</i>
<i>SaaS</i>	<i>Software as a Service</i>
<i>SPI</i>	<i>Serial Peripheral Interface</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>SRAM</i>	<i>Static random-access memory</i>
<i>STL</i>	<i>Standard Triangle Language</i>
<i>TCP/IP</i>	<i>Transmission Control Protocol / Internet Protocol</i>
<i>TPU</i>	<i>Thermoplastic Polyurethane</i>
<i>UART</i>	<i>Universal Asynchronous Receiver/Transmitter</i>
<i>UML</i>	<i>Unified Modeling Language</i>
<i>USB</i>	<i>Universal Serial Bus</i>
<i>VR</i>	<i>Virtual Reality</i>
<i>WIN</i>	<i>Wearable, Implantable, Noninvasive</i>

Lista de Unidades

A	Ampere
C	<i>Celsius</i>
cm	centímetro
dB	Decibel
<i>fps</i>	<i>Frames per second</i>
g	grama
GB	<i>Gigabyte</i>
g·cm	Grama por centímetro cúbico
GHz	<i>Gigahertz</i>
Hz	<i>Hertz</i>
kB	<i>Kilobyte</i>
kg	kilograma
Kg·cm	Kilograma por centímetro cúbico
kHz	<i>Kilohertz</i>
m	metro
mA	miliampere
MB	<i>Megabyte</i>
Mbps	<i>Megabits per second</i>
MHz	<i>Megahertz</i>
mm	milímetro
mm/s	Milímetros por segundo
ms	milisegundo
N·m	<i>Newton</i> por metro
Pa	Pascal
r/min	Rotações por minuto
V	<i>Volt</i>
W	<i>Watt</i>

Lista de Símbolos

∅	Diâmetro
€	Euro
°	Grau

ÍNDICE DE FIGURAS

FIGURA 1 - CPS GENERALIZADO E ALGUMAS DAS SUAS APLICAÇÕES (GUAN ET AL. 2016)	4
FIGURA 2 - INTERPRETAÇÃO DO SISTEMA DE "FEEDBACK" NUM CPS (GUAN ET AL. 2016)	4
FIGURA 3 - COMPARAÇÃO ENTRE A INDÚSTRIA ATUAL E UMA INDÚSTRIA 4.0 (LEE ET AL. 2015)	5
FIGURA 4 - TIPOS DE ERRO, EM CASO DE DESCONEXÃO (DERLER ET AL. 2012)	7
FIGURA 5 - BASE DE FUNCIONAMENTO PARA AS DIFERENTES ABORDAGENS (GUAN ET AL. 2016)	9
FIGURA 6 - MODELAÇÃO DE CPS COM RECURSO A PTOLEMY II (GUAN ET AL. 2016)	10
FIGURA 7 - "5C ARCHITECTURE" PARA IMPLEMENTAR EM CPS (LEE ET AL. 2015)	11
FIGURA 8 - APLICAÇÕES E TÉCNICAS ASSOCIADAS A CADA NÍVEL DA "5C ARCHITECTURE" (LEE ET AL. 2015)	12
FIGURA 9 - DOMÍNIO LINEAR DO ESPECTRO RELATIVO AOS MODELOS/DEFINIÇÕES DE CPS (PUTNIK AND FERREIRA 2019)	14
FIGURA 10 - DOMÍNIO DINÂMICO DO ESPECTRO RELATIVO AOS MODELOS/DEFINIÇÕES DE CPS (PUTNIK AND FERREIRA 2019)	15
FIGURA 11 - AS DEFINIÇÕES E MODELOS COMPLETOS DO CPS, INTEGRANDO AS DUAS PARTES LINEARES E DINÂMICAS DO DOMÍNIO DO ESPECTRO (PUTNIK AND FERREIRA 2019)	16
FIGURA 12 - ARQUITETURA DE UM PS/MS (PUTNIK AND FERREIRA 2019)	17
FIGURA 13 - ARQUITETURA DE UM MODELO CPS ⁰ (PUTNIK AND FERREIRA 2019)	18
FIGURA 14 - ARQUITETURA DE UM MODELO CPS ¹ (PUTNIK AND FERREIRA 2019)	19
FIGURA 15 - SINGLE-LOOP LEARNING, JUNTAMENTE COM PROCESSOS DE SIMULAÇÃO (PUTNIK AND FERREIRA 2019)	20
FIGURA 16 - ARQUITETURA DE UM MODELO CPS ² (PUTNIK AND FERREIRA 2019)	21
FIGURA 17 - DOUBLE-LOOP LEARNING, JUNTAMENTE COM PROCESSOS DE SIMULAÇÃO E DE DECISÃO (PUTNIK AND FERREIRA 2019)	21
FIGURA 18 - ARQUITETURA DA CAMADA "REALIDADE AUMENTADA" (KHALID ET AL. 2015)	23
FIGURA 19 - ARQUITETURA (GERAL) DE UM CPS-VMS (HU ET AL. 2017)	24
FIGURA 20 - ARQUITETURA MULTIFACETADA DE UM CPS-VMS, DE TRÊS CAMADAS (HU ET AL. 2017)	24
FIGURA 21 - TROCA DE "MENSAGENS" ENTRE MA'S (HU ET AL. 2017)	25
FIGURA 22 - DESDOBRAMENTO DE UM CPS-VMS (1-AGENTES VR; 2-AGENTES FISIOLÓGICOS; 3-AGENTES DE MOVIMENTO; 4-AGENTES DE RASTREAMENTO) (HU ET AL. 2017)	25
FIGURA 23 - FRAMEWORK (TEORÉTICO) DO OPEN INNOVATION (CHIARONI ET AL. 2011)	28
FIGURA 24 - FASES DE UM PROCESSO RELATIVO AO OPEN DESIGN (REBENS DORF ET AL. 2015)	32
FIGURA 25 - POSSÍVEIS VARIAÇÕES DE UM PROCESSO DE OPEN DESIGN (TOOZE ET AL. 2014)	33
FIGURA 26 - SINGLE-LOOP LEARNING (FORRESTER, SWARTLING, AND LONSDALE 2008)	36
FIGURA 27 - DOUBLE-LOOP LEARNING (FORRESTER ET AL. 2008)	36
FIGURA 28 - ARQUITETURA PROPOSTA	40
FIGURA 29 - TECNOLOGIAS CENTRAIS DO CLOUD COMPUTING, ADAPTADO DE (LUO ET AL. 2016)	43
FIGURA 30 - ARQUITETURA BASEADA EM NUVEM (FERREIRA ET AL. 2017)	44
FIGURA 31 - VISÃO HIERÁRQUICA DA COMPUTAÇÃO EM NUVEM, ADAPTADO DE (TSAI ET AL. 2010)	45
FIGURA 32 - ARQUITETURA CORRESPONDENTE À CLOUD DO SISTEMA	45

FIGURA 33 - ARQUITETURA DO REPOSITÓRIO <i>BPCM</i> (GAO AND KROGSTIE 2010)	46
FIGURA 34 - ARQUITETURA DO REPOSITÓRIO <i>DSPACE</i> (TANSLEY ET AL. 2003)	47
FIGURA 35 - ARQUITETURA CLIENTE-SERVIDOR DE DOIS NÍVEIS (OLUWATOSIN 2014)	48
FIGURA 36 - ARQUITETURA CLIENTE-SERVIDOR DE TRÊS NÍVEIS (OLUWATOSIN 2014)	48
FIGURA 37 - DIAGRAMA DE FLUXO AVANÇADO (NAGAPPAN AND VAYIRAVAN 2015)	49
FIGURA 38 - ARQUITETURA COMUNICACIONAL, ADAPTADO EM FERREIRA ET AL. (2017)	50
FIGURA 39 - EXEMPLO DE UM <i>IDE</i> (WIELENGA 2016)	51
FIGURA 40 - SOFTWARE CAD/CAM (4IENG 2019)	52
FIGURA 41 - MANUFATURA ADITIVA (IMPRESSÃO 3D) (ALL3DP 2021A)	53
FIGURA 42 - <i>SOFTWARE SLICER</i>	53
FIGURA 43 - EXEMPLO DE UM <i>SOFTWARE</i> DE <i>HOST</i> E DE CONTROLO (IMPRESSÃO 3D) (GUJAR ET AL. 2019)	54
FIGURA 44 - EXEMPLO DE UMA BASE DE DADOS SQL (SULZBACH 2018)	55
FIGURA 45 - EXEMPLO DE UMA INTEGRAÇÃO DO SISTEMA MULTIAGENTE NA ARQUITETURA PROPOSTA	57
FIGURA 46 - FLUXOGRAMA DO AGENTE PRODUTO, ADAPTADO DE (RAMIÃO ET AL. 2017)	58
FIGURA 47 - DIAGRAMA DE ATIVIDADES DO AGENTE DE RECURSO (RAMIÃO ET AL. 2017)	60
FIGURA 48 - DIAGRAMA DE ATIVIDADE DE UM CLA (PARTE 1) (RAMIÃO ET AL. 2017)	61
FIGURA 49 - DIAGRAMA DE ATIVIDADE DE UM CLA (PARTE 2) (RAMIÃO ET AL. 2017)	61
FIGURA 50 - DIAGRAMA DE ATIVIDADES DO DA (RAMIÃO ET AL. 2017)	63
FIGURA 51 – ARQUITETURA DO SISTEMA FÍSICO	64
FIGURA 52 - COMPONENTE CIBERFÍSICO	66
FIGURA 53 – REPRESENTAÇÃO DO SISTEMA DE INFORMAÇÃO E CONTROLO	66
FIGURA 54 - SISTEMA DE PRODUÇÃO, CONCEITUAL (RAMIÃO ET AL. 2017)	68
FIGURA 55 - FASE 1 DO CASO DE EXEMPLO	68
FIGURA 56 - FASE 2 DO CASO DE EXEMPLO	69
FIGURA 57 - FASE 3 DO CASO DE EXEMPLO	69
FIGURA 58 - FASE 4 DO CASO DE EXEMPLO	70
FIGURA 59 - FASE 5 DO CASO DE EXEMPLO	70
FIGURA 60 - FASE 6 DO CASO DE EXEMPLO	71
FIGURA 61 - REPRESENTAÇÃO DO SISTEMA DE PRODUÇÃO	71
FIGURA 62 - MOTOR PASSO-A-PASSO (SOLECTRO 2020)	72
FIGURA 63 - <i>MICRO-SWITCH</i> (DISTRELEC 2021)	73
FIGURA 64 - SISTEMA DE COMUNICAÇÃO E VIGILÂNCIA	73
FIGURA 65 - MODELO REPRESENTATIVO DA IMPLEMENTAÇÃO (FÍSICA) PRETENDIDA	74
FIGURA 66 - DIAGRAMA DE BLOCOS/FLUXO CORRESPONDE AO SISTEMA FÍSICO	74
FIGURA 67 - ARDUINO NANO (ARDUINO LLC ET AL. 2012)	79
FIGURA 68 - ARDUINO UNO REV3 (FARNELL 2013)	80
FIGURA 69 - ARDUINO MEGA 2560 REV3 (MELLIS 2011)	81
FIGURA 70 - ARDUINO UNO WIFI REV2(WIFI N.D.)	82
FIGURA 71 - ESP8266 (FIREBEETLE ET AL. N.D.)	83
FIGURA 72 – <i>CONTROLLINO MAXI</i> (ABDUL AZIZ AHMED SIYAD 2019)	84
FIGURA 73 - <i>CONTROLLINO MEGA</i> (CONTROLLINO 2017)	84

FIGURA 74 - LAYOUT DA DISPOSIÇÃO DO SISTEMA DE PRODUÇÃO	87
FIGURA 75 - <i>ROBOLINK® RL-DP 5 DOF</i> (IGUS 2021B)	88
FIGURA 76 - DORNA 2 (AVE AND CA N.D.)	89
FIGURA 77 - <i>BCN3D MOVEO</i> (BCN3D TECHNOLOGIES 2016A)	91
FIGURA 78 - <i>BEEVERYCREATIVE B2X300</i> (BEEVERYCREATIVE 2019)	94
FIGURA 79 - <i>CREALITY ENDER 3</i> (CREALITY 2020)	96
FIGURA 80 - <i>ANET A8</i> (COMPONENTES 2018)	97
FIGURA 81 - <i>SAINSMART 3018 CNC PRO</i> (BANGOOD 2020)	98
FIGURA 82 - <i>MILLRIGHT CNC M3</i> (ALL3DP 2021B)	99
FIGURA 83 - <i>MICROSWITCH MS5-L</i> (ELETRÓNICA 2021A)	101
FIGURA 84 - D3V-16-1C24 (ELETRÓNICA 2021B)	102
FIGURA 85 - <i>SW015</i> (MAUSER.PT 2021)	102
FIGURA 86 - ARDUCAM MINI (OV2640) (ARDUCAM 2019)	104
FIGURA 87 - ESP32 CAM (MAUSER.PT 2018)	105
FIGURA 88 - <i>RASPBERRY PI 400</i> (FOUNDATION 2020)	106
FIGURA 89 - MONITOR ASUS VS197DE (PCDIGA 2021)	107
FIGURA 90 - <i>EWENT EM3563</i> (VELLEMAN 2016)	108
FIGURA 91 - LOGÓTIPO DO <i>NETBEANS</i> (LENZ 2017)	109
FIGURA 92 - EXEMPLO DE UM PROJETO EM <i>NETBEANS</i> (WIELENGA 2016)	110
FIGURA 93 - LOGÓTIPO DO <i>ECLIPSE IDE</i> (FOUNDATION 2012)	110
FIGURA 94 - EXEMPLO DE UM PROJETO EM <i>ECLIPSE</i> (JAVA TUTORIAL 2018)	111
FIGURA 95 - LOGÓTIPO DO <i>ARDUINO</i> (ARDUINO 2015)	111
FIGURA 96 - EXEMPLO DE UM PROJETO <i>ARDUINO</i>	112
FIGURE 97 – OPEN SCIENCE TAXONOMY (PONTIKA ET AL. 2015).	132
FIGURE 98 - ARCHITECTURE OF THE PROPOSED CPS SUPPORTED BY OD.	134
FIGURE 99 – MANUFACTURING SYSTEM LAYOUT (3D, IN LEFT SIDE, AND 2D, IN RIGHT SIDE, MODELS)	135
FIGURA 100 - ARQUITETURA PROPOSTA	145

ÍNDICE DE TABELAS

TABELA 1 - OS TRÊS ARQUÉTIPOS DO OPENNESS, BASEADO EM (KLAASSEN ET AL. 2018)	27
TABELA 2 - CLOSED VS OPEN INNOVATION, BASEADO EM (GRECO, LOCATELLI, AND LISI 2017)	29
TABELA 3 - AS DIFERENTES CAMADAS DO OPEN DESIGN, BASEADO EM (KLAASSEN ET AL. 2018)	31
TABELA 4 - OPEN DESIGN VS CLOSED DESIGN, ADAPTADO DE (KLAASSEN ET AL. 2018)	34
TABELA 5 - COMANDOS PRINCIPAIS DE CÓDIGO G	52
TABELA 6 - COMUNICAÇÃO ENTRE AGENTES, BASEADO EM (RAMIÃO ET AL. 2017)	63
TABELA 7 - LISTA DE COMPONENTES DO SISTEMA DE INFORMAÇÃO E CONTROLO	78
TABELA 8 - ORÇAMENTO PRIMORDIAL DO SISTEMA DE INFORMAÇÃO E CONTROLO	78
TABELA 9 – ESPECIFICAÇÕES TÉCNICAS DO ARDUINO NANO, ADAPTADO DE (ARDUINO LLC ET AL. 2012)	79
TABELA 10 – ESPECIFICAÇÕES TÉCNICAS DO ARDUINO UNO REV3, ADAPTADO DE (FARNELL 2013)	80
TABELA 11 - ESPECIFICAÇÕES TÉCNICAS DE ARDUINO MEGA 2560 REV3, ADAPTADO DE (MELLIS 2011)	81
TABELA 12 - ESPECIFICAÇÕES TÉCNICAS DO ARDUINO UNO WIFI REV2, ADAPTADO DE (WIFI N.D.)	82
TABELA 13 - ESPECIFICAÇÕES TÉCNICAS DO ESP8266, ADAPTADO DE (ESPRESSIF SYSTEMS 2020; FIREBEETLE ET AL. N.D.; KODALI AND MAHESH 2016)	83
TABELA 14 - ESPECIFICAÇÕES TÉCNICAS DO CONTROLLINO MAXI, ADAPTADO DE (DEGREE 2017)	85
TABELA 15 - ESPECIFICAÇÕES TÉCNICAS DO CONTROLLINO MEGA, ADAPTADO DE (CONTROLLINO 2017)	85
TABELA 16 - LISTA DE COMPONENTES DO SISTEMA DE PRODUÇÃO	87
TABELA 17 - ORÇAMENTO PRIMORDIAL DO SISTEMA DE PRODUÇÃO	88
TABELA 18 - ESPECIFICAÇÕES TÉCNICAS DO ROBOLINK® RL-DP 5 DOF (TECHNICAL N.D.)	89
TABELA 19 - ESPECIFICAÇÕES TÉCNICAS DO DORNA 2 (AVE AND CA N.D.)	90
TABELA 20 - ESPECIFICAÇÕES TÉCNICAS DO BCN3D MOVEO (BCN3D TECHNOLOGIES 2016A)	91
TABELA 21 - LISTA DE MATERIAIS NECESSÁRIOS À MONTAGEM DO BRAÇO ROBÓTICO BCN3D MOVEO, ADAPTADO DE (BCN3D TECHNOLOGIES 2016B)	92
TABELA 22 - LISTA DE MATERIAIS DE IMPRESSÃO PARA O BRAÇO ROBÓTICO BCN3D MOVEO	94
TABELA 23 - ESPECIFICAÇÕES TÉCNICAS DA BEEVERYCREATIVE B2X300, ADAPTADO DE (BEEVERYCREATIVE 2019)	95
TABELA 24 - ESPECIFICAÇÕES TÉCNICAS DA CREALITY ENDER 3, ADAPTADO DE (ALL3DP 2020)	96
TABELA 25 - ESPECIFICAÇÕES TÉCNICAS DA ANET A8, ADAPTADO DE (KOSLOW 2018)	97
TABELA 26 - ESPECIFICAÇÕES TÉCNICAS DA SAINSMART 3018 CNC PRO, ADAPTADO DE (SAINSMART 2020)	98
TABELA 27 - ESPECIFICAÇÕES TÉCNICAS DA MILLRIGHT CNC M3, ADAPTADO DE (MILLRIGHT 2017)	100
TABELA 28 - ESPECIFICAÇÕES TÉCNICAS DO MS5-L, ADAPTADO DE (VELLEMAN 2021)	101
TABELA 29 - ESPECIFICAÇÕES TÉCNICAS DO D3V-16-1C24, ADAPTADO DE (GAP 2021)	102
TABELA 30 - ESPECIFICAÇÕES TÉCNICAS DO SW015, ADAPTADO DE (MAUSER.PT 2021)	102
TABELA 31 - LISTA DE COMPONENTES DO SISTEMA DE COMUNICAÇÃO E VIGILÂNCIA	103
TABELA 32 - ORÇAMENTO PRIMORDIAL DO SISTEMA DE COMUNICAÇÃO E VIGILÂNCIA	103
TABELA 33 - ESPECIFICAÇÕES TÉCNICAS DA ARDUCAM MINI, ADAPTADO DE (ARDUCAM 2015)	104

TABELA 34 - ESPECIFICAÇÕES TÉCNICAS DA ESP32 CAM, ADAPTADO DE (AI-THINKER 2017; WORKSHOP 2020)	105
TABELA 35 - ESPECIFICAÇÕES TÉCNICAS DO RASPBERRY PI 400, ADAPTADO DE (PI 2020)	106
TABELA 36 - ESPECIFICAÇÕES TÉCNICAS DO ASUS VS197DE, ADAPTADO DE (PCDIGA 2021)	107
TABELA 37 - ESPECIFICAÇÕES TÉCNICAS DO EWENT EM3563, ADAPTADO DE (MAUSER.PT 2017; VELLEMAN 2016)	108
TABELA 38 - ESPECIFICAÇÕES TÉCNICAS DO MICROFONE, ADAPTADO DE (MAUSER.PT 2017; VELLEMAN 2016)	108
TABLE 39 – BOM OF THE OSLAB4MAN PHYSICAL SYSTEM	136

ÍNDICE

1	INTRODUÇÃO	1
1.1	Contextualização	1
1.2	Objetivos	1
2	REVISÃO BIBLIOGRÁFICA.....	3
2.1	Sistemas Ciberfísicos	3
2.1.1	O presente e o futuro	3
2.1.2	Desafios na modelação	6
2.1.3	Abordagens à modelação	8
2.1.4	Outros tipos de arquiteturas	13
2.1.5	O “espectro” de Modelos.....	14
2.1.5.1	Domínio Linear.....	14
2.1.5.2	Domínio Dinâmico	15
2.1.5.3	Análise dos dois domínios.....	16
2.1.6	Descrição dos Modelos	17
2.1.6.1	PS/MS.....	17
2.1.6.2	CPS ⁰	17
2.1.6.3	CPS ¹	19
2.1.6.4	CPS ²	20
2.1.6.5	CPS-AMS	22
2.1.6.6	CPS-VMS.....	23
2.2	Open Design	26
2.2.1	Openness.....	26
2.2.2	Open Innovation.....	27
2.2.3	Open Source	29
2.2.4	Open Design	30
2.3	Encadeamento dos Sistemas Ciberfísicos no Open Design.....	35
3	ARQUITETURA PROPOSTA DE UM SISTEMA CIBERFÍSICO ORIENTADO À PRODUÇÃO ABERTA	38
3.1	Estrutura global da arquitetura proposta	38
3.2	Sistema computacional em Nuvem (<i>cloud</i>)	42
3.2.1	Repositório	46
3.2.2	Servidor (es)	47
3.2.2.1	Aplicação.....	49

3.2.2.2	IDE's	51
3.2.2.3	Software gerador de código-G	51
3.2.2.4	Software de Host e Controlo (NC)	54
3.2.3	Base de dados.....	54
3.2.4	Segurança, Documentação e Gestão	55
3.3	Sistema Multiagente	56
3.3.1	Product Agent.....	57
3.3.2	Resource Agent	59
3.3.3	Coalition Leader Agent	60
3.3.4	Deployment Agent	62
3.3.5	Comunicação entre Agentes	63
3.4	Sistema Físico	64
3.4.1	Sistema de Informação e Controlo.....	65
3.4.2	Sistema de Produção	67
3.4.3	Sistema de Automação	72
3.4.4	Sistema de Comunicação e Vigilância	73
3.4.5	Encadeamento dos subsistemas	73
4	IMPLEMENTAÇÃO DA ARQUITETURA PROPOSTA	76
4.1	Sistema Físico	77
4.1.1	Sistema de Informação e Controlo.....	77
4.1.1.1	Lista de Componentes e Orçamento Primordial	77
4.1.1.2	Propostas de componentes	78
4.1.1.3	Análise, Escolha e Orçamento Final dos Componentes	86
4.1.2	Sistema de Produção	86
4.1.2.1	Lista de Componentes e Orçamento Primordial	86
4.1.2.2	Propostas de componentes	88
4.1.2.3	Análise e Orçamento Final	100
4.1.3	Sistema de Automação.....	101
4.1.3.1	Propostas	101
4.1.4	Sistema de Comunicação e Vigilância	103
4.1.4.1	Lista de Componentes e Orçamento Primordial	103
4.1.4.2	Propostas	103
4.1.4.3	Escolha de Componentes e Orçamento Final	108
4.2	Sistema computacional em Nuvem (<i>cloud</i>)	109
4.2.1.1	IDE's	109
5	CONCLUSÕES.....	114
5.1	Conclusões	114

5.2	Trabalhos futuros	115
6	BIBLIOGRAFIA E OUTRAS FONTES DE INFORMAÇÃO.....	119
6.1	Referências bibliográficas	119
7	APÊNDICES	131
7.1	Apêndice A – Artigo intitulado “ <i>Cyber-Physical Systems using Open Design: an approach towards an Open Science Lab for Manufacturing</i> ” (no âmbito da conferência internacional <i>CENTERIS - International Conference on ENTERprise Information Systems</i>)	131
7.2	Apêndice B - Artigo intitulado: “ <i>Open Science Laboratory for Manufacturing: an education tool to contribute to sustainability</i> ” (no âmbito da conferência internacional ACM)	138
7.3	Apêndice C – Artigo intitulado “ <i>Sistemas Ciber-Físicos recorrendo ao Open Design: uma abordagem a um Laboratório de Ciência Aberta destinado à Produção</i> ” (no âmbito do Encontro Virtual de Engenharia Gestão Industrial (EVEGI) 2021)	144

INTRODUÇÃO

1.1	Contextualização	1
1.2	Objetivos	1

1 INTRODUÇÃO

1.1 Contextualização

Ao longo da última década assistimos a uma evolução muito significativa e positiva da indústria. De facto, o avanço tem sido tanto que é denominado de Indústria 4.0. Com ela têm surgido novos e revolucionários recursos e funcionalidades, como é o caso dos Sistemas Ciberfísicos, que, por sua vez, é um sistema composto por elementos computacionais colaborativos com o intuito de controlar entidades físicas de, neste caso de estudo, uma produção.

Além do fenómeno anterior, também se sucedeu o Rise of Open-X, que consiste na disponibilização de quaisquer conteúdo à comunidade de forma relativamente gratuita e aberta, sendo, igualmente neste caso, esse conteúdo alusivo à produção (Open Design).

No entanto, estes dois conceitos ainda não se encontram interligados, ou melhor dizendo, ainda não surgiu um sistema que tenha a capacidade de controlar uma produção e, posteriormente, disponibilizar todo esse processo de uma forma “aberta”. Com isto, este trabalho surge de maneira a colmatar tal entrave, diga-se.

1.2 Objetivos

Um dos objetivos passa por fazer uma revisão do estado da arte relativamente aos sistemas ciberfísicos e à produção aberta, com o âmbito de recolher o máximo de informação no que toca às características e faculdades de ambos os temas. Feito isto, o outro objetivo, e principal, é estabelecer um modelo ideal que vincule as duas matérias, recorrendo à informação referida, e, posteriormente, comprová-lo através de uma análise prática e respetivos resultados.

REVISÃO BIBLIOGRÁFICA

- 2.1 Sistemas Ciberfísicos
- 2.2 Open Design
- 2.3 Encadeamento dos Sistemas Ciberfísicos no Open Design

2 REVISÃO BIBLIOGRÁFICA

2.1 Sistemas Ciberfísicos

2.1.1 O presente e o futuro

Inicialmente, de uma forma geral, o CPS (Cyber Physical System) consistia simplesmente na integração de processos computacionais e físicos, nos quais computadores interligados por redes monitorizam e controlam esses mesmos processos físicos, dando posteriormente respostas cíclicas em que ambos os processos em questão se afetam entre si (Putnik and Ferreira 2019).

No entanto, devido à não tão descoberta ligação e coordenação entre o mundo computacional e o mundo físico, os sistemas de controlo associados ao CPS estão sobre uma perseverante e dinâmica reorganização e reconfiguração, mais propriamente com novos e altos níveis de automatização em diversas dimensões espaciais e temporais (Guan et al. 2016). Com isto, foram surgindo novas definições, ou pelo menos foi sendo adicionada nova informação às definições “primordiais”. Além da razão descrita anteriormente, a constante investigação e pesquisa relativamente a este tipo de sistema, bem como o surgimento de tecnologias melhoradas e novos serviços também foram determinantes para que tal acontecesse.

Em seguimento ao referido anteriormente, Trappey et al. (2016), aborda os CPS's como pontos de perceção do mundo físico através do (mundo) computacional, onde essas noções são tidas como dados que, à semelhança da maior parte das definições, serão processados e calculados. Dessa análise, resultarão ações a tomar para que o resultado final do processo seja melhorado.

Os mesmos autores defendem ainda que os conceitos de computação, comunicação e controlo, juntamente com a informação recolhida, são a base fundamental para que os Sistemas Ciber-Físicos possam chegar a resultados dinâmicos, tais como a deteção em tempo real, controlo dinâmico e proporcionar informação a sistemas de maior calibre (Trappey et al. 2016). Efetivamente, os dados detetados nos processos físicos têm de ser traduzidos no mundo *cyber*, bem como as estratégias de controlo definidas por este último também necessitam de ser transferidas para o mundo real. Para que tal seja possível é necessário recorrer aos sensores e atuadores, que juntamente com as infraestruturas de comunicação em rede, operam como conectores do mundo físico e do computacional (Guan et al. 2016), tal como é ilustrado na Figura 1 e Figura 2.

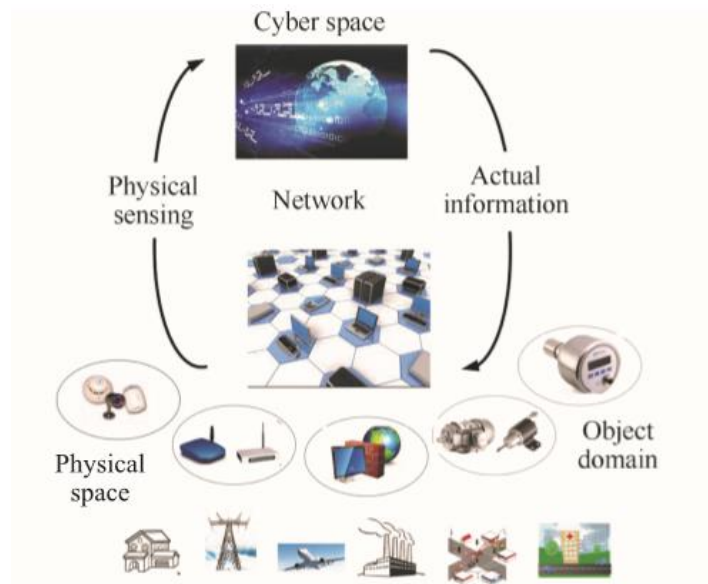


Figura 1 - CPS generalizado e algumas das suas aplicações (Guan et al. 2016)

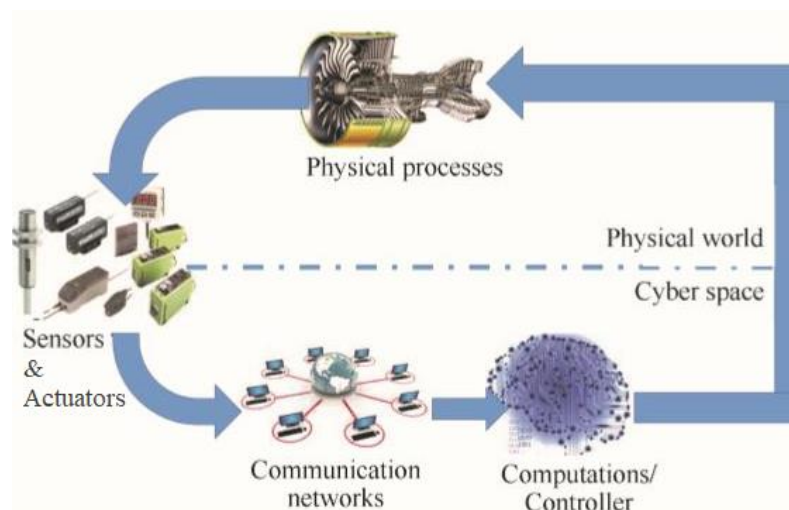


Figura 2 - Interpretação do sistema de "feedback" num CPS, adaptado de (Guan et al. 2016)

Um outro ponto de realçar é aquele referido por Guan et al. (2016), no qual defende que a integração dos mundos físicos e computacionais não é meramente uma convergência de ambos, mas sim uma interação profunda dos componentes pertencentes aos dois “mundos”. Por esse motivo, a análise e o design de CPS é baseada no entender das ligações dinâmicas entre os processos físicos, computadores, *software* e redes.

O CPS apresenta características como a oportunidade, distribuição, confiabilidade, tolerância ao erro, segurança, escalabilidade e operação autónoma, que combinadas resultam numa ferramenta que auxilia na superação de certos obstáculos/desafios, como é o caso do controlo do custo dos produtos que uma empresa fabrica. Além do mais, tais atributos oferecem micro-inteligência ao equipamento de forma a que os produtos de curto ciclo de vida de hoje atinjam uma customização em massa no futuro (Trappey et al. 2016). Com tudo isto, é seguro dizer que os sistemas ciberfísicos são a marca da transição da Indústria 3.0 para a 4.0.

Por outro lado, o conceito de CPS continua a ser um paradigma visto que ainda há vários aspetos do mesmo a explorar. Quanto a este assunto, Putnik and Ferreira (2019), fundamenta que os sistemas ciberfísicos não se tratarão apenas de sistemas de controlo resilientes com interfaces geradas automaticamente, mas sim sistemas de controlo ricos em contexto que oferecem configurações inovadoras para mudanças corretivas oportunas. Os autores acrescentam ainda que o enquadramento associado ao CPS enfrentará um conjunto de unidades mais autónomas, ao invés de encararem simplesmente um maior número de sensores e atuadores. Efetivamente, estas “coisas” (como são generalizadas), apesar de não estarem propriamente integradas, cooperam entre si, permitindo assim uma reconfiguração e programação dinâmica, criando também um ecossistema de redes forte e adaptável.

Em seguimento a este pensamento, Lee, Bagheri, and Kao (2015), alegam que o constante desenvolvimento de sensores, atuadores e das tais unidades autónomas referidas anteriormente, bem como o dos engenhos interligados por redes, tem resultado numa contínua geração de grandes quantidades de informação, conhecida por Big Data. Tendo isto em conta, os mesmos defendem que os CPS's podem ser futuramente desenvolvidos para esse propósito, ou seja, gerir Big Data, assim como permitir uma melhor interconectividade entre máquinas de forma a que elas se tornem inteligentes, flexíveis e auto-adaptáveis.

Da mesma forma, ao equacionar os sistemas ciberfísicos com a produção, logísticas e serviços na atual indústria, esta tornar-se-ia numa Indústria 4.0 com enorme potencial económico (Lee et al. 2015). Aliás, não será apenas a economia a crescer, mas sim também a quantidade, a estrutura e as competências das equipas que desenvolvem e implementam CPS's. Isto trará, igualmente, como consequência uma maior competitividade global entre nacionalidades que se tornaram tecnologicamente líderes neste tipo de sistema em questão (Rajkumar et al. 2010).

	Data source	Today's factory		Industry 4.0	
		Attributes	Technologies	Attributes	Technologies
Component	Sensor	Precision	Smart sensors and fault detection	Self-aware Self-predict	Degradation monitoring & remaining useful life prediction
Machine	Controller	Producibility & performance	Condition-based monitoring & diagnostics	Self-aware Self-predict Self-compare	Up time with predictive health monitoring
Production system	Networked system	Productivity & OEE	Lean operations: work and waste reduction	Self-configure Self-maintain Self-organize	Worry-free productivity

Figura 3 - Comparação entre a indústria atual e uma indústria 4.0 (Lee et al. 2015)

Este último artigo dá ainda realce a um ponto bastante interessante, em que refere que da mesma maneira que a *Internet* transformou como os humanos interagem e comunicam entre si, revolucionou como e onde a informação está disponível, e até mesmo mudou a forma de como as pessoas compram e vendem produtos; os sistemas ciberfísicos modificarão o modo como os humanos interagem e controlam o mundo físico envolvente.

2.1.2 Desafios na modelação

A aplicação dos padrões comuns para a criação de modelos encontra-se, atualmente, distante da mais adequada, tendo em conta a inerente e característica complexidade dos emergentes panoramas relativos aos CPS's. De facto, incluir sistemas físicos, computacionais, dinâmicos e baseados em humanos que necessitam de ser representados e correlacionados, onde também é crucial considerar o contexto, campo de informação, (re)organização dinâmica e respostas cíclicas em tempo real, torna a modulação em si um desafio. Posto isto, é requerida a inclusão de engenharia de controlo, engenharia de software, redes de sensores, etc (Putnik and Ferreira 2019).

Os processos computacionais tratam-se de eventos sequenciais, ao contrário dos processos físicos, que por sua vez são conjuntos de coisas de ocorrem simultaneamente, ou, se quisermos, composições de vários processos em paralelo. Assim sendo, avaliar e controlar as dinâmicas destes últimos processos, instrumentando ações que os influenciam, são as principais tarefas dos sistemas ciberfísicos. Aliás, os maiores desafios técnicos passam por isso mesmo, isto é, criar uma ponte entre uma semântica inerentemente sequencial e um mundo físico intrinsecamente concomitante (Putnik and Ferreira 2019).

No entanto, o mundo físico não é inteiramente previsível. Consequentemente, tal como já defendia Derler, Lee, and Sangiovanni-vincentelli (2011), e visto não estar a operar em ambientes controlados, os CPS's devem ser resistentes e adaptáveis tendo em conta possíveis condições inesperadas e falhas de subsistemas. Para anexar este problema, o projeto de componentes previsíveis e confiáveis seria, teoricamente, suficiente, porém, nenhum componente é perfeitamente confiável e tais condições inesperadas também acabariam por ludibriar a previsibilidade. Com isto, a questão passa por saber o quão um designer se deve focar nesses fatores aquando da modelação deste tipo de sistemas, bem como evitar sistemas frágeis, onde pequenas variações das operações espectadas leva a falhas catastróficas.

De forma a colmatar as questões anteriores, o princípio a seguir é relativamente simples: os componentes de qualquer nível de abstração têm de ser previsíveis e confiáveis, se isso for tecnologicamente exequível. Caso não o seja, então o próximo nível de abstração acima desses mesmos componentes devem compensar com robustez.

Mesmo usando este estratagema, esta situação complica-se quando passamos de um simples programa para sistemas de *software*, em particular para os sistemas ciberfísicos. Efetivamente, nestes casos os componentes que, teoricamente, são perfeitamente previsíveis e confiáveis acabam por não o ser em dimensões importantes. A isto denomina-se falha de abstração (Derler et al. 2011).

Khaitan and McCalley (2015) vão mais longe e consideram os CPS's como um sistema de sistemas, complexos e heterogéneos, que interagem entre si de uma forma contínua. Os mesmos autores contrapõem que as dificuldades na modelação de CPS's

surtem devido a essa mesma heterogeneidade, à simultaneidade dos processos físicos e às exigências do “a tempo real”. Entrando mais em detalhe, diferentes níveis e tipos de abstrações são requeridos para os componentes físicos e computacionais e respectivas interações. Contudo, essas abstrações diferem das pretendidas em virtude da física subjacente, granularidade e detalhes solicitados, o que torna as abordagens, estruturas e arquiteturas de modelagem inapropriadas, assim como já foi referido no início do tópico.

Com o propósito de atenuar todos os problemas atrás descritos, os projetistas recorrem a *model-based design* e *model-driven development* visto se tratar de modelos que permitem a simulação e a análise, possibilitando assim uma identificação prévia de possíveis defeitos no mesmo. Todavia, Lee E., num artigo seu mais recente (Derler, Lee, and Sangiovanni Vincentelli 2012), menciona que continuam a surgir obstáculos tais como: modelos com comportamento dependente de um Solver, indeterminado ou *Zeno*; prevenir a desconexão de componentes; e interações (de modelagem) de funcionalidade e implementação.

Melhor explicando, os modelos de CPS's que trabalham a “tempo contínuo” estão associados a *Solvers* que aproximam numericamente as soluções das equações diferenciais. No entanto, esta integração ainda se encontra em exploração, o que torna os sistemas ainda muito dependentes destes *Solvers*. Outro problema que pode surgir é quando o comportamento definido por um modelo pode ser indeterminado, mesmo quando o sistema subjacente modelado é determinado, significando que o modelo define uma multiplicidade de comportamentos, em vez de um único comportamento (Derler et al. 2012).

Ainda dentro do tema, outra adversidade é a prevenção da desconexão dos componentes pertencentes ao sistema, isto porque quanto maior o modelo, maior é a dificuldade em verificar essas mesmas conexões. Em caso de existência de desconexões, são três os tipos de erros que podem acontecer: *unit errors*, *semantic errors* and *transposition errors* (Derler et al. 2012), tal como exemplifica a Figura 4.

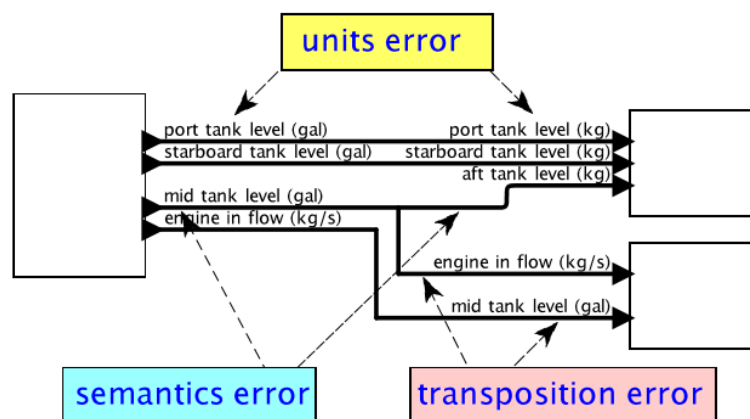


Figura 4 - Tipos de erro, em caso de desconexão (Derler et al. 2012)

Em relação à última contrariedade, é necessário ter uma maior atenção às interações entre a funcionalidade e a implementação, uma vez que a implementação é amplamente ortogonal à funcionalidade, e daí não deve ser parte integrante de um modelo de funcionalidade. O último permite a exploração do espaço de design, enquanto o primeiro apoia o design de estratégias de controlo. Os modelos combinados permitem a avaliação das interações entre esses domínios (Derler et al. 2012).

2.1.3 Abordagens à modelação

Aprofundando agora o que diz respeito às abordagens, recentemente estas têm surgido em número significativo, juntamente com ferramentas para CPS's, de forma a combater a heterogeneidade dos mesmos, bem como modelar conceitos e requisitos de domínio de aplicação específicos. No que concerne aos modelos, os investigadores exploraram conceitos como a semântica multiagente, semântica baseada em eventos e *actor-oriented design*. A isto podemos associar os modelos de computação, nomeadamente o “a tempo-contínuo”, máquinas de estados finitos, eventos discretos e redes de processo (Khaitan and McCalley 2015).

Efetivamente, Yue et al, em (Niz 2010), propôs um método que usa o modelo adaptável de eventos discretos, o qual incorpora o *discrete event calculus* (DEC). O uso do DEC permite manipular eventos do ambiente, evitando inconsistências na especificação das regras de domínio. Além disso, para melhorar a adaptabilidade do aplicativo CPS, os autores introduziram um conjunto de raciocínio anormal, que permite ao CPS lidar com eventos imprevistos. Os investigadores demonstraram a utilidade desta abordagem através de um CPS de rastreamento de movimento de objetos.

Tan, Vuran, and Goddard (2009) apresentaram uma arquitetura para o CPS, baseada nas propriedades temporais e espaciais dos eventos. Os autores representaram um evento como uma função das condições de evento baseadas em atributos, espaciais e temporais. Ademais, usando os operadores lógicos, diferentes tipos de condições de eventos são combinados para capturar eventos compostos, o que permite capturar relacionamentos complexos dos sistemas ciberfísicos. Dessa forma, seu quadro permite a realização de análises espaciais e temporais formais de CPS's.

Kong, Jiang, and Wu (2010) expuseram uma metodologia de projeto baseada em modelo para CPS's. Eles decompuseram o *design* baseado em modelo em dez etapas e propuseram uma metodologia de *design* iterativa. Os mesmos demonstraram a eficácia de sua abordagem de projeto com o exemplo de um dispositivo de bola de tunelamento, o que requer interação estreita de computação embarcada em tempo real e movimento de hardware com deteção e atuação de alta precisão.

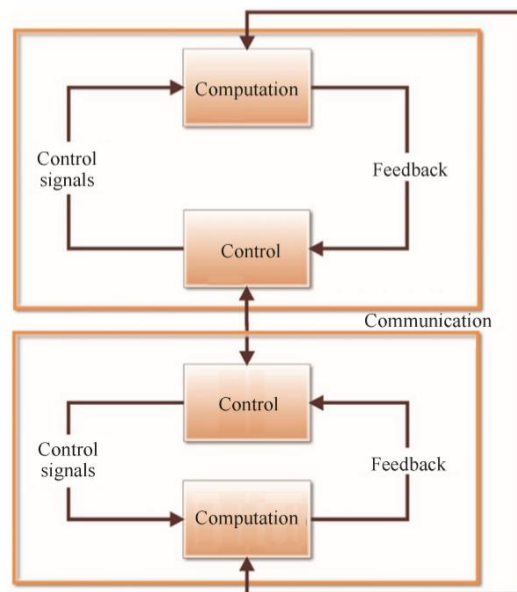


Figura 5 - Base de funcionamento para as diferentes abordagens (Guan et al. 2016)

(Derler et al. 2012) estudou duas abordagens complementares para o design de CPSs. A primeira abordagem, chamada *“cyberizing the physical”*, refere-se a agrupar abstrações de software em torno de subsistemas físicos. A segunda abordagem, denominada *“physicalizing the cyber”*, refere-se à dotação de abstração de componentes de software e rede, adequados para subsistemas físicos. À medida que a complexidade dos CPSs aumenta, os desafios de modelar a interação dos sistemas cibernético e físico também aumentam e, portanto, o uso dessas abordagens ajuda a diminuir as diferenças para obter uma modelagem mais realista.

Jha et al. (2010) discutiram uma abordagem para modelar um CPS com dinâmica multimodal. Este sistema opera em modos diferentes, onde a dinâmica de cada modo é bem conhecida, e para garantir a segurança, é importante alternar adequadamente entre os diferentes modos. Posto isto, os autores propuseram um método para sintetizar a lógica de comutação fornecida na dinâmica modificada. Nestes casos, o projetista pode auxiliar o processo de síntese, fornecendo aproximações iniciais das proteções de permutação.

Ehyaie et al, em (Properties and Mucosa 2015), exploraram o problema de agregar as leituras do sensor em um CPS. Essas leituras são obtidas de um grande número de sensores que estão densamente localizados e se comunicam em um único campo de transmissão (ou seja, a qualquer momento, no máximo um sensor pode comunicar). Os investigadores propuseram um método eficiente para a aquisição de dados com base na interpolação de todas as leituras. O método deles também modela a dinâmica do sistema físico, o que permite realizar a aquisição de dados em um momento independente do número de nós.

Conforme já foi referido anteriormente neste tópico, o *model-driven development* com suporte à base de conhecimento desempenha um papel importantíssimo no design de software, tal como defende Guan et al. (2016). De facto, a abordagem MDE é

considerada como o facilitador para modelar CPS's. A título exemplificativo, *Ptolemy II*, baseado em MDE, fornece *actor-oriented modelling* que suporta modelagem de vários domínios no mesmo espaço de trabalho de design, assim como está representado na Figura 6.

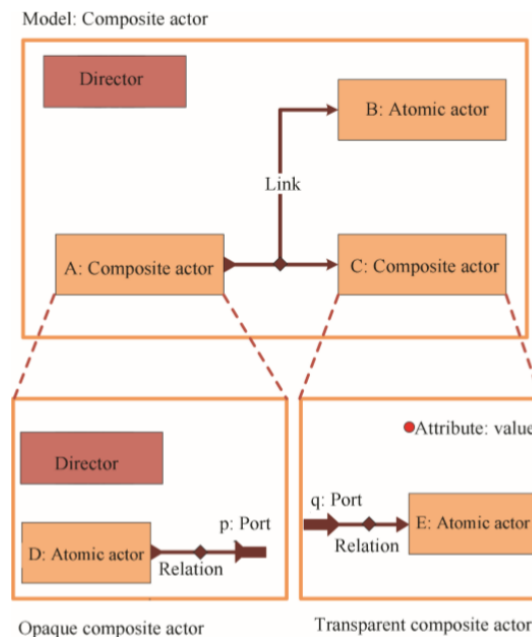


Figura 6 - Modelação de CPS com recurso a Ptolemy II (Guan et al. 2016)

O mesmo autor argumenta ainda que o *agent-based modeling* é um método promissor para lidar com as interdependências entre as infraestruturas cibernéticas e as físicas, visto que os *software agents* agem como componentes flexíveis, autônomos e inteligentes para a tomada de decisões (Guan et al. 2016). Para modelar sistemas multi-agente, a unified modeling language (UML) é uma escolha intuitiva. Esta linguagem consiste em vários tipos de diagramas estruturados e elementos gráficos que são montados para representar um modelo.

Um outro fator na modelagem aqui descrito é o comportamento dos sistemas. Uma abordagem para verificar formalmente o comportamento do sistema é usar o *Simplex Architecture* para lógica de controlo não verificada, que por sua vez é combinada com um controlador verificado. A lógica de controlo não verificada normalmente tem melhor desempenho, enquanto o controlador verificado é projetado apenas com aspetos críticos de segurança. Quando o sistema estiver em estado anormal, o sistema passará para o controlador verificado automaticamente. Perante isto, o principal desafio de projetar um sistema baseado na *Simplex Architecture* é criar corretamente a lógica de comutação (Johnson et al. 2016).

No entanto, com o avanço das técnicas de deteção, comunicação e computação em nuvem, o tamanho do CPS está em contínuo aumento, bem como a informação gerada por este, ou seja, *Big Data*. Em consequência de tal surgem novos desafios: como melhorar a precisão dos modelos analisando dados massivos; como transformar os dados recolhidos em informação útil para a modelação; e como utilizar informações

massivas obtidas por sensores para controlo e tomada de decisão de alto nível. Com isto, Lee et al. (2015) propôs uma arquitetura de cinco camadas para a implementação do CPS com *Big Data* industrial, mais conhecido por “*5C Level Architecture*”. Pormenorizando cada nível (Figura 7), temos:

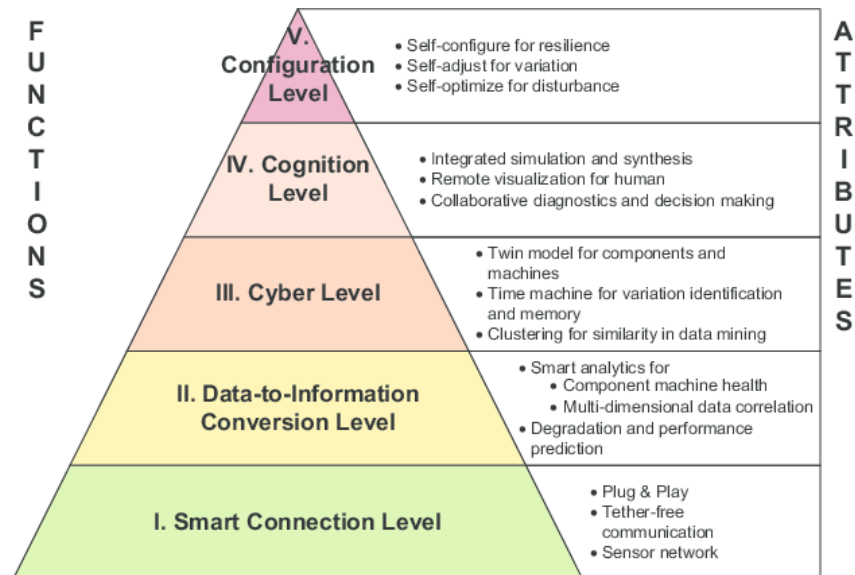


Figura 7 - “5C architecture” para implementar em CPS (Lee et al. 2015)

- **Connection:** *Smart connection* - Dois fatores importantes neste nível devem ser considerados. Primeiro, tendo em conta vários tipos de dados, um método sem interrupções e sem fio para gerenciar o procedimento de aquisição de dados e transferir dados para o servidor central é necessário onde protocolos específicos, como o *MTConnect* e etc. são efetivamente úteis. Por outro lado, a seleção de sensores adequados (tipo e especificação) é a segunda consideração importante para o primeiro nível;
- **Conversion:** *Data-to-information conversion* - Informações relevantes devem ser deduzidas dos dados. Nos últimos anos, um amplo foco tem sido aplicado para desenvolver esses algoritmos especificamente para prognósticos e aplicativos de gestão da saúde das máquinas. De facto, o calcular o valor da saúde, a vida útil estimada restante, etc., o segundo nível da arquitetura do CPS traz autoconsciência às máquinas;
- **Computation:** *Cyber* - O nível cibernético atua como um *hub* central de informações nesta arquitetura, onde as informações são enviadas de todas as máquinas conectadas para formar a rede das máquinas. Tendo sido recolhida toda essa informação, análises específicas são feitas para extrair informações adicionais que fornecem uma melhor visão sobre o *status* de máquinas individuais entre as restantes. Essas análises fornecem às máquinas a capacidade de auto-comparação, onde o desempenho de uma única máquina pode ser comparado e classificado entre a frota. Em contrapartida, as semelhanças entre o desempenho da máquina e os ativos anteriores

(informações históricas) podem ser medidas para prever o comportamento futuro do maquinário;

- **Cognition:** A apresentação adequada do conhecimento adquirido a usuários especializados suporta a decisão correta a ser tomada. Como as informações comparativas e o *status* individual da máquina estão disponíveis, é possível tomar uma decisão sobre a prioridade das tarefas para otimizar o processo de manutenção. Para este nível, são necessários gráficos de informação adequados para transferir completamente o conhecimento adquirido aos usuários.;
- **Configuration:** O nível de configuração é o *feedback* do espaço cibernético para o espaço físico e atua como controle de supervisão para tornar as máquinas auto-configuráveis e auto-adaptáveis. Esta etapa opera como um sistema de controle de resiliência (RCS) para aplicar as decisões corretivas e preventivas, que foram tomadas em nível de cognição, ao sistema monitorado.

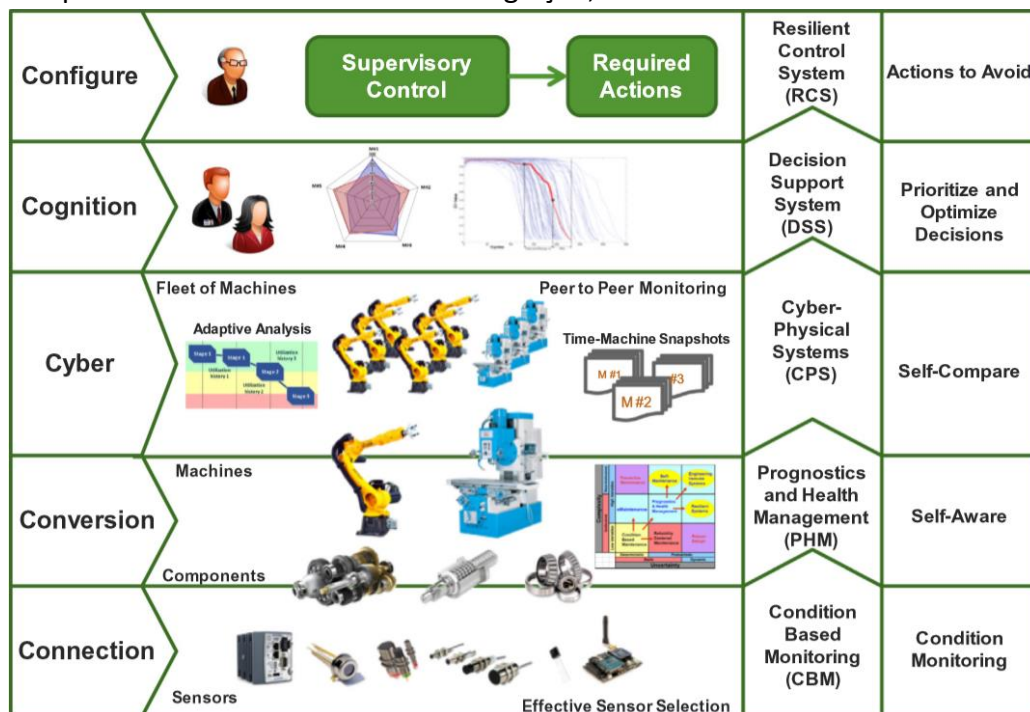


Figura 8 - Aplicações e técnicas associadas a cada nível da "5C architecture" (Lee et al. 2015)

Em forma de resumo, as abordagens anteriormente descritas, juntas, estabelecem uma base teórica para representar e descrever os CPS's e os seus diferentes aspetos, tais como a concorrência e a comunicação (Khaitan and McCalley 2015). Em contrapartida, a natureza heterogênea da maioria das aplicações de CPS requer o uso de misturas heterogêneas de modelos de computação, assim como será desenvolvido posteriormente no trabalho.

Contudo, apesar das inúmeras soluções para maior parte dos problemas, o cenário futuro dos sistemas ciberfísicos não parece ficar por aqui. Aliás, a atenção e preocupações emergentes em torno da Comunicação, Integração, Digitalização, Miniaturização, Personalização, Sustentabilidade (entre outros) apoiam a lucratividade

de investimentos (humanos e materiais), a reutilização de equipamentos legados, a eficiência energética, a eficiência energética, a reorganização de processos (e outros), que por sua vez são pilares essenciais na relevância atual dos CPS's (Putnik and Ferreira 2019).

Além disso, a capacidade computacional e comunicacional das infraestruturas e sistemas existentes (e previstos), força novos desafios e promove novos paradigmas de integração e colaboração, bem como a necessidade de melhorar a adaptabilidade e a autonomia. Assim sendo, uma linguagem de modelagem exclusiva é altamente desejada para o design deste tipo de sistema (Putnik and Ferreira 2019). *Ptolomy II* fornece certo nível de abstração e encapsulamento para o CPS, mas nem todos os domínios de aplicativo podem ser cobertos, ainda. Além do mais, essa nova linguagem de modelagem também deverá conectar o controlo em tempo real com cálculos e comunicações baseados em eventos (Guan et al. 2016).

2.1.4 Outros tipos de arquiteturas

Apesar desta dissertação/pesquisa se debruçar com mais ênfase nos modelos de CPS, existem outro tipos de arquitetura, nomeadamente a Meta-Arquitetura/Meta-Programação, Semânticas e *Codesign* (Khaitan and McCalley 2015). Em relação à primeira, esta surgiu devido à necessidade de métodos formais para especificar e verificar propriedades de componentes heterogêneos envolvidos no sistema. Um exemplo de uma Meta-arquitetura é a sugerida por Hang, Manolios, and Papavasileiou (2011), que propõe técnicas para permitir a síntese algorítmica de modelos arquitetónicos ciberfísicos com restrições em tempo real. Efetivamente, eles utilizaram uma linguagem de especificação meta-arquitetural para permitir que os designers especifiquem as propriedades desejadas do modelo de arquitetura sem exigir que esclareçam como alcançá-las. Além disso, eles também desenvolveram um solucionador de teorias de módulos de programação linear inteira, juntamente com um solucionador de teoria de agendamento e implantaram-no para sintetizar modelos arquitetónicos ciberfísicos com restrições rigorosas em tempo real.

No que toca às “Semânticas”, estas estruturas têm sido abordadas de diferentes modos, particularmente de forma denotacional, axiomática, operacional ou um híbrido destes. De facto, e servindo como exemplo, Bujorianu, Bujorianu, and Barringer (2009) sugeriram uma abordagem formal denominada Método Formal *Hibertian* para fornecer uma semântica denotacional para CPS's. Eles combinaram semântica denotacional com um modelo algébrico de processos físicos para modelar causalidade e observabilidade físicas.

A respeito do *Codesign*, Miller, Vahick, and Givargis (2011) apresentaram uma estrutura de código para o desenvolvimento de dispositivos ciberfísicos, onde dispositivos ou protótipos reais são conectados a modelos em tempo real que simulam o ambiente de interação. A estrutura proposta suporta código de hardware/software para permitir que modelos de velocidade e precisão variáveis sejam implementados

num processador incorporado. Usando esta abordagem, uma plataforma específica do aplicativo para testar o CPS pode ser gerada automaticamente.

2.1.5 O “espectro” de Modelos

Em consequência das inúmeras abordagens relativas aos sistemas ciberfísicos, descritas anteriormente, existem também várias definições, as quais, em parte, surgiram através da “mistura” dessas mesmas abordagens. Subsequentemente, a estas definições estão associados modelos de CPS, que por sua vez podem ser agrupados em classes pelas suas características qualitativas e ordenados pela sua complexidade e aplicabilidade, criando assim um “espectro”, assim como Putnik and Ferreira (2019) defendem. Na verdade, o espectro está dividido em duas secções: o domínio linear e o domínio dinâmico.

2.1.5.1 Domínio Linear

Quanto ao primeiro, este inclui os modelos/definições PS/MS, CPS⁰, CPS¹ e CPS², em que podem ser ordenados linearmente devido à natureza aditiva da incorporação de novos módulos ou funcionalidades (Putnik and Ferreira 2019), tal como está representado na Figura 9.

É de realçar que os modelos do tipo CPS¹ e, especialmente, CPS², também podem ser denominados de CPS’s “Inteligentes”, ou I-CPS, em virtude da inclusão de capacidades de aprendizagem. Assim sendo, considerando o *single loop* no CPS¹ e o *double loop* no CPS², as denominações passariam a I-CPS¹ e I-CPS², ou, alternativamente, I¹-CPS e I²-CPS. Claro está que os modelos sem aprendizagem, e de forma a manter a consistência, podem ser intitulados de I-CPS⁰ ou I⁰-CPS (Putnik and Ferreira 2019). Esta nomenclatura alternativa também está retratada na Figura 9.

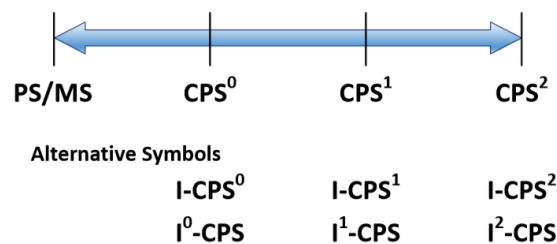


Figura 9 - Domínio linear do espectro relativo aos modelos/definições de CPS (Putnik and Ferreira 2019)

Na extremidade esquerda encontra-se a definição mais clássica de sistema ciberfísico, o PS/MS, que será mais tarde abordado com maior detalhe. De seguida, estão os modelos sem capacidade de aprendizagem (CPS⁰), como é o caso da defendida por Griffor et al. (2017), o qual refere que CPS’s são sistemas inteligentes que incluem redes de interação projetadas de componentes físicos e computacionais.

Num ponto intermédio do domínio apresentam-se os modelos com aprendizagem em *single loop* (CPS¹), representando a funcionalidade da parte física da reconfigurabilidade de um CPS, tal como Tomiyama and Moyon (2018) alegam.

Adicionalmente, e analisando as abordagens/modelos até agora referenciadas, é concluinte que a maioria pertence a este tipo de modelos, segundo o domínio linear.

Na outra extremidade, a direita, situam-se definições que requerem conexões não fixas e em constante mudança, aliás, nestas situações, arquiteturas com conexões fixas não são consideradas CPS's. Um exemplo disso está presente em Lee, E. A., & Seshia (2017), onde os autores argumentam que um sistema ciberfísico é uma integração da computação com processos físicos, cujo comportamento é definido pelas partes cibernética e física do sistema, e que os computadores e redes interligados monitorizam e controlam os processos físicos, com *loops* de *feedback* nos quais os processos físicos afetam os cálculos e vice-versa. Contudo, uma abordagem/modelo, do tipo CPS², que incorpora a alteração do sistema computacional pelo sistema físico, ainda não foi desenvolvido.

2.1.5.2 Domínio Dinâmico

Posto isto, também é afirmável que o domínio linear pode ser traçado simplesmente pelas definições, ao contrário do dinâmico. Efetivamente, esta última é regida pela capacidade de crescimento e adaptação ao meio envolvente, ou seja, cria uma ordem dentro dos CPS's com capacidade de aprendizagem. Além disso, é igualmente tido em conta os sistemas industriais existentes, dando origem à seguinte disposição (Putnik and Ferreira 2019) (Figura 10): CPS (*Classical Cyber Physical Systems*), CPS-AMS (*CPS Augmented Manufacturing Systems*), M-CPS (*Mixed CPS*); CPS-VMS (*CPS Virtual Manufacturing Systems*) e I-CPS (*Intelligent CPS*).

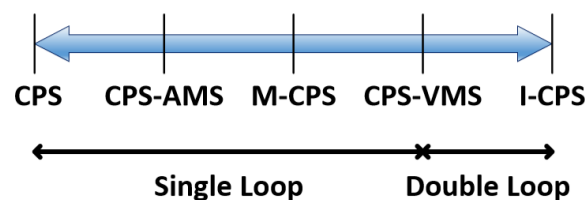


Figura 10 - Domínio dinâmico do espectro relativo aos modelos/definições de CPS (Putnik and Ferreira 2019)

O domínio do espectro em questão começa com modelos de sistemas ciberfísicos em torno do clássico, isto é, CPS⁰ e CPS¹, que são, resumidamente, componentes físicos, e respetivos sistemas de controlo, que podem reconfigurar o módulo de controlo do sistema de fabricação. Por outro lado, os CPS-AMS exploram ativos de realidade aumentada, usando dados em tempo real provenientes de sensores integrados (reconhecimento de contexto), enquanto que o CPS-VMS explora a realidade virtual em cenários de simulação. Existe ainda o M-CPS, onde a realidade mista, tecnologia e sistemas de controlo autónomo são explorados, em direção a medidas corretivas (Putnik and Ferreira 2019).

No entanto, todos estes modelos são baseados no *single loop learning*, ignorando, assim, as causas reais à grande escala. De forma a ultrapassar este problema, existem os, ainda hipotéticos, I-CPS, que por sua vez apresentam aprendizagem contínua, o que torna o CPS inteligente, eficaz, colaborativo e sistêmico, com ativos abstratos ou

intangíveis, preparado para domínios de grande escala (Putnik and Ferreira 2019). Isto deve-se ao significativo *double loop learning*, cuja aprendizagem efetiva resultante provoca um aumento de capacidade individual de tomar decisões eficazes (Kantamara and Ractham 2014).

2.1.5.3 Análise dos dois domínios

Apesar dos conceitos, que servem como base para os modelos do espectro, já serem avançados, estes não conseguem lidar com cenários de complexidade, incerteza e dinamismo que a realidade gera (Sousa, Nunes, and Lopes 2015). Tendo isto em consideração, bem como a não tão simples relação entre os dois domínios, Putnik et al (Putnik and Ferreira 2019) alega a necessidade de estender o (domínio) linear para CPS^N , permitindo assim “reprojetar” ou reprogramar continuamente os algoritmos de suporte dos sistemas de controlo e, portanto, uma nova instância do CPS, potencializando modelos de aprendizagem de *n-loop*.

Além disso, os mesmo autores (Putnik and Ferreira 2019) acreditam que os *n-loops learning* suportam um espectro síncrono de CPS, onde novas categorias mais eficientes de CPS são geradas, tal como está ilustre na Figura 11. Efetivamente, cada uma delas apresenta uma caracterização linear (PS , CPS^0 , CPS^1 , CPS^2) e um ténue limite dinâmico que possibilita a adaptação a diferentes situações, explorando a realidade virtual, aumentada e de inteligência para aprender, sendo assim capaz de evoluir. De facto, a capacidade ágil de aprender contra a incerteza ou o contexto imprevisível enquadra-se em soluções claramente exponenciais de modelos de crescimento, cuja implementação terá de seguir linhas híbridas de algoritmos clássicos e quânticos (Putnik and Ferreira 2019).

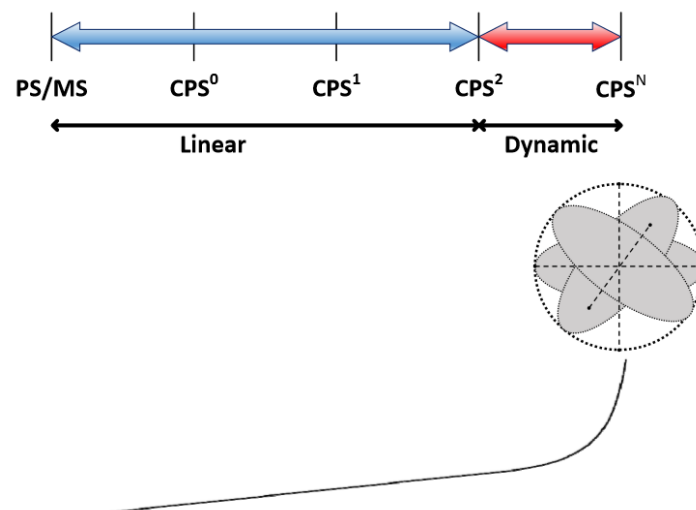


Figura 11 - As definições e modelos completos do CPS, integrando as duas partes lineares e dinâmicas do domínio do espectro (Putnik and Ferreira 2019)

Na Figura 11 também está representado, figurativamente, o “crescimento” linear e exponencial, em que este último é proveniente da complexidade das definições e

modelos, acabando com um “quântico”, e respetivo “crescimento”, hipoteticamente, extremo.

2.1.6 Descrição dos Modelos

Quanto aos modelos, e de forma a dar continuidade à ilação tirada, tratar-se-á primeiramente daqueles que constituem o domínio linear do espectro de sistemas ciberfísicos. Sendo assim, temos então:

2.1.6.1 PS/MS

Nos Sistemas de Produção (PS) ou Sistemas de Manufatura (MS), simplificado na Figura 12, as infraestruturas físicas são geridas e controladas por diversos sistemas de informação, mais concretamente por ERP, BOM, MES, etc. nos níveis mais “altos” de controlo e por funções de controlo incorporadas a “níveis” mais baixos, como por exemplo o servo-controlo, formando assim a parte digital do PS/MS. No entanto, estes tipos de sistema de controlo, tradicionais e convencionais, não são considerados CPS, uma vez que os *feedback loops* nos tais “níveis altos” são realizados *offline*, ao contrário do que acontece nos “baixos”, o que é, atualmente, visto como uma limitação (Putnik and Ferreira 2019). Aliás, a referência a estes sistemas serve também para mostrar as diferenças paradigmáticas do antes e depois do termo “ciberfísico”.

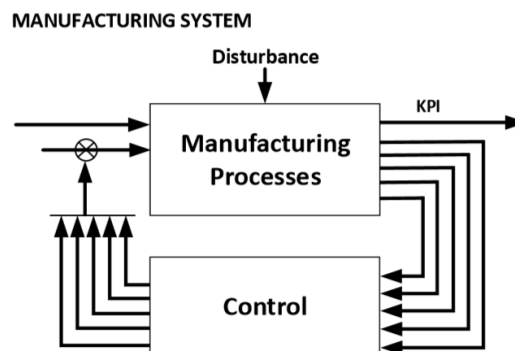


Figura 12 - Arquitetura de um PS/MS (Putnik and Ferreira 2019)

2.1.6.2 CPS⁰

Os Sistemas Ciber-Físicos Clássicos tratam-se de sistemas onde infraestruturas físicas e sistemas digitais estão relacionados e controlados ciclicamente (*feedback loop*), fornecendo adaptabilidade, reconfigurabilidade, capacidade de resposta rápida e robustez (Putnik and Ferreira 2019). Estas características são algumas vantagens de um CPS, visto que os sistemas (de controlo) tradicionais não podem responder com rapidez suficiente a um sistema em larga escala devido à falta de agilidade e comportamento dinâmico, que por sua vez são necessários para lidar com ambientes de sistemas complexos e variáveis (Jiang et al. 2017). Por outras palavras, a principal diferença entre os sistemas de produção tradicionais sem e com CPS é que o último tem a capacidade de controlar, ciclicamente, em tempo real.

Para atingir tamanhos atributos, o CPS⁰ recorre a uma nova geração de tecnologias de computação emergentes, tais como computação em “nuvem” e onipresente, *Internet of Things*, *Web 2.0*, *3.0 e 4.0*, *Fast Internet* e similares (Putnik and Ferreira 2019). Na verdade, todas essas tecnologias computacionais “emergentes” requerem, quase sem exceção, serviços recorrentes “externos”, enquanto que os restantes *loops* de controlo de “nível inferior” continuam embutidos “internamente” nos módulos de controlo “tradicionais” PS/MS. Por esta mesma razão, a parte computacional intrínseca do CPS é dada como separada, sendo então representada na Figura 13 como “DIGITAL PROCESSES 0 LAYER”, em concordância com a denominação de CPS⁰. Em contrapartida, a “camada física”, ou se quiser, o PS/MS, ainda mantém uma parte do controlo tradicional (Putnik and Ferreira 2019), retratada como “PHYSICAL PROCESSES LAYER”, também na Figura 13.

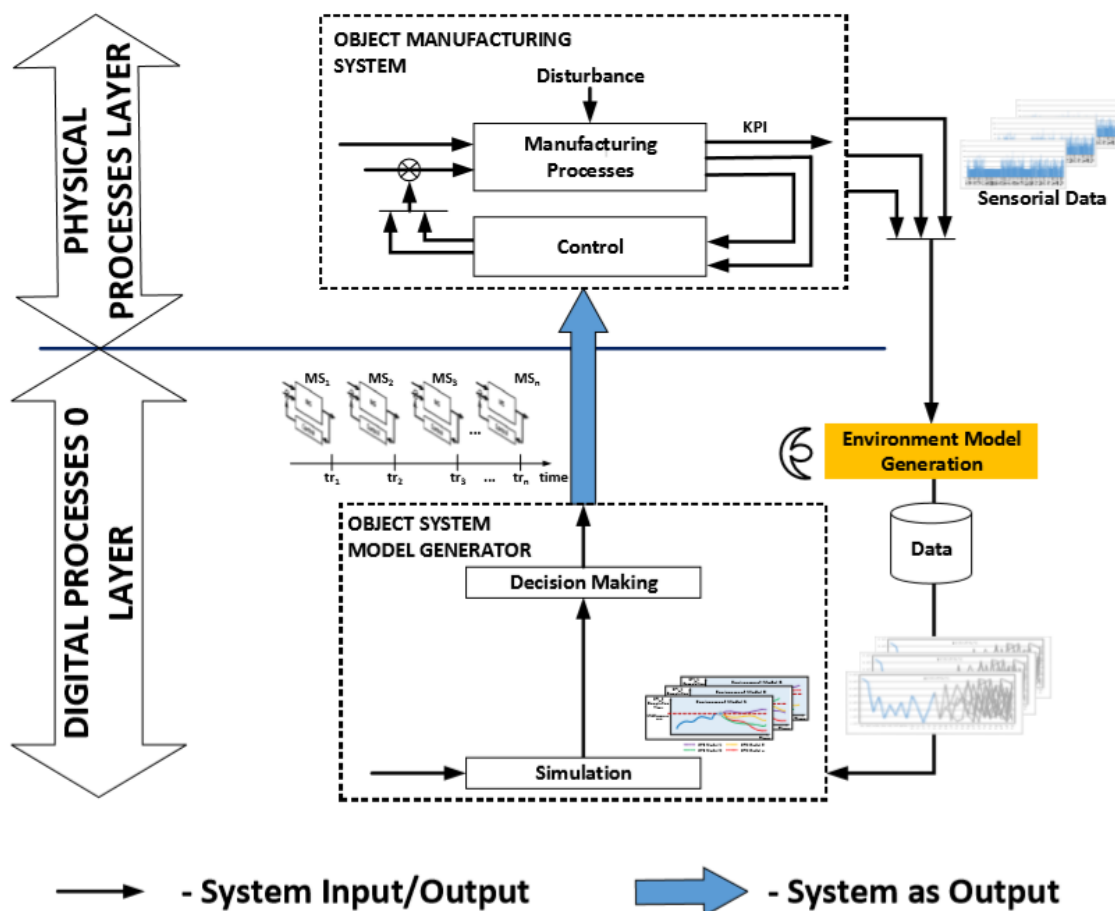


Figura 13 - Arquitetura de um modelo CPS⁰ (Putnik and Ferreira 2019)

A “Simulação” (Figura 13), que, basicamente, ao reagir aos *input states*, permite a simulação de alterações aplicadas a todos os CPS, bem como a análise do impacto das alterações do ambiente (dados de processamento anterior, distúrbios, etc.) no processo de fabricação. A agregação de dados passados, presentes e previstos e a exploração de algoritmos avançados apoiarão a capacidade de reconfigurabilidade da “parte física” do CPS. Desta maneira, a reconfiguração da mesma e a reprogramação

dos seus serviços principais podem ser exploradas e, consequentemente, os resultados da produção realinhados e previstos (Putnik and Ferreira 2019).

A necessidade de simulações contínuas vem da avaliação permanente dos resultados ao longo do tempo, em que cada estado deve ser visto como uma nova instância de todo o sistema, tendo o estado anterior como input, bem como as novas condições do ambiente, pretendidas (variações forçadas) ou não (perturbações). Isto provocará alterações funcionais na camada física que, por seu lado, só são possíveis através das ferramentas avançadas de simulação orientadas aos requisitos do CPS e da Indústria 4.0, como é o caso das *i-machines*, as quais permitem que os sistemas lidem com isso de forma contínua e dinâmica, reconfigurando-o em qualquer altura do processo de fabricação (Putnik and Ferreira 2019).

2.1.6.3 CPS¹

Este é um sistema ciberfísico com a parte física e a parte digital integradas, onde o *single-loop learning* pode afetar (ou alterar) a parte física (configurações, agendas, etc.). A arquitetura lógica desse tipo de modelo do CPS é uma extensão da arquitetura do CPS⁰ com a incorporação do módulo *Learning*, fornecendo reconfiguração estrutural do módulo de controlo "Object Manufacturing System". Com esta incorporação, é formado um segundo tipo de camada computacional/digital (Putnik and Ferreira 2019), mais propriamente a "DIGITAL PROCESSES 1 LAYER", novamente em concordância com a nomenclatura de CPS¹, tal como está ilustre na Figura 14.

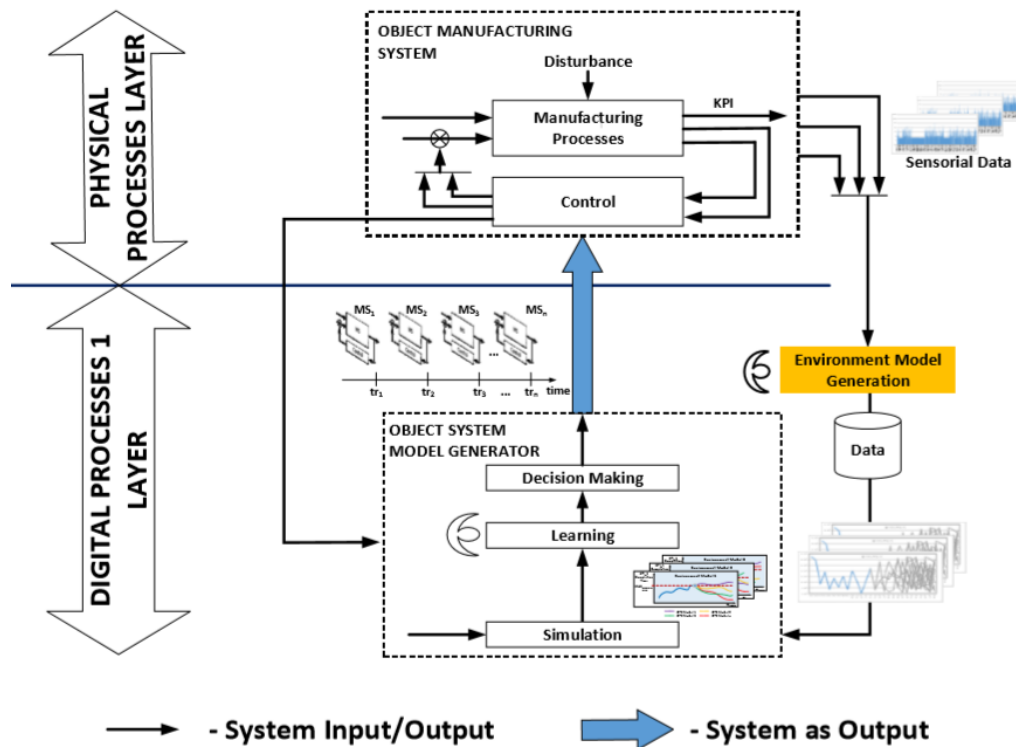


Figura 14 - Arquitetura de um modelo CPS¹ (Putnik and Ferreira 2019)

O “*Learning*”, da mesma forma que o modelo CPS⁰, agrega dados passados, presentes e previstos e explora algoritmos avançados de estratégias de aprendizagem de interfaces de inteligência artificial, classificação de lógica difusa, decisão com vários critérios, aprendizagem profunda e outros, o que engradecerá a capacidade do próprio *Learning*. Igualmente à semelhança do CPS⁰, existe uma necessidade constante de “aprender”, exatamente pelas mesmas razões, o que irá, posteriormente, causar alterações estruturais dentro do “componente” em questão (Putnik and Ferreira 2019).

Assim sendo, juntamente com os processos de Simulação, o *Learning* dará forma a um novo módulo de controlo e considerar-lho-á como um *output*, o que origina a substituição do módulo antigo. Isto repetir-se-á sequencialmente, devido à mudança de ambiente e contextos (Putnik and Ferreira 2019), como descrito na Figura 15.

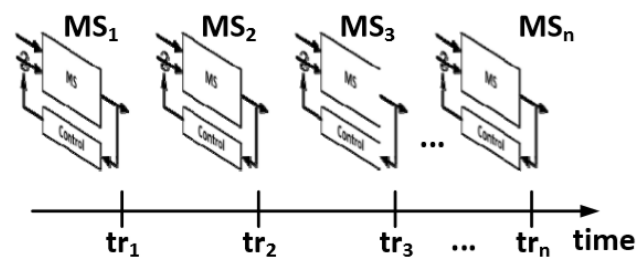


Figura 15 - Single-loop learning, juntamente com processos de Simulação (Putnik and Ferreira 2019)

2.1.6.4 CPS²

Concerne a Sistemas Ciberfísicos com *double loop learning*, onde a parte digital pode ser alterada (ajustada) para controlar melhor a parte física. Neste modelo, é introduzido outro nível digital, com o segundo módulo *Learning*, que, por sua vez, tem a capacidade de aprender sobre o de nível inferior (“*Learning about learning*”) e também está incorporado no módulo denominado “*Computational System Model Generator*” (Figura 16). Ambos os módulos “*Learning*”, juntos, originam um terceiro, e o mais complexo, tipo de parte computacional (Putnik and Ferreira 2019), representado como “DIGITAL PROCESSES 2 LAYER” na Figura 16, estando mais uma vez de acordo com a definição de CPS².

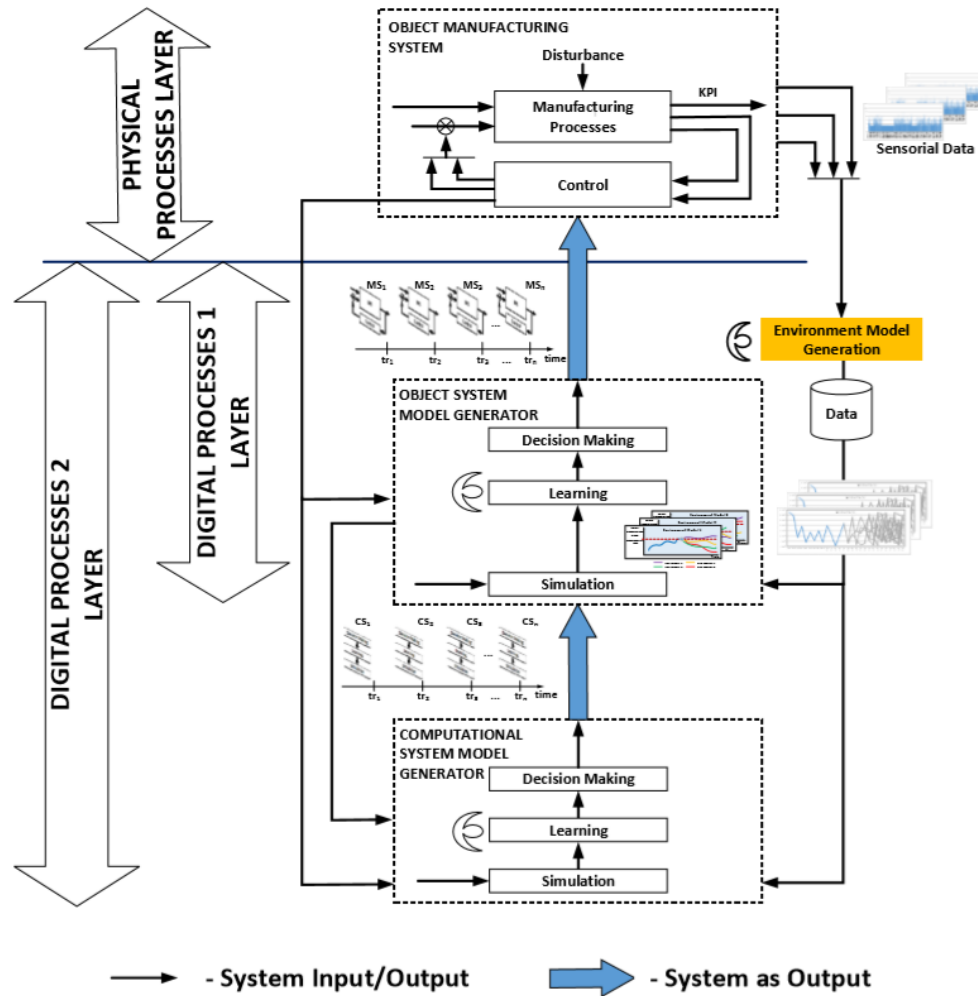


Figura 16 - Arquitetura de um modelo CPS² (Putnik and Ferreira 2019)

No nível digital em foco serão tratados, da mesma forma que os níveis inferiores, os dados passados, presentes e previstos, bem como a exploração dos mesmos conceitos de tais níveis. Além disso, a essencialidade de uma aprendizagem contínua persiste, proveniente de causas similares, resultando em modificações estruturais no *second loop learning* análogas ao primeiro (*loop*). De facto, este comportamento ocorrerá, novamente, de modo sequencial, em virtude das mudanças de ambiente e contextos (Putnik and Ferreira 2019), tal como é visível na Figura 17.

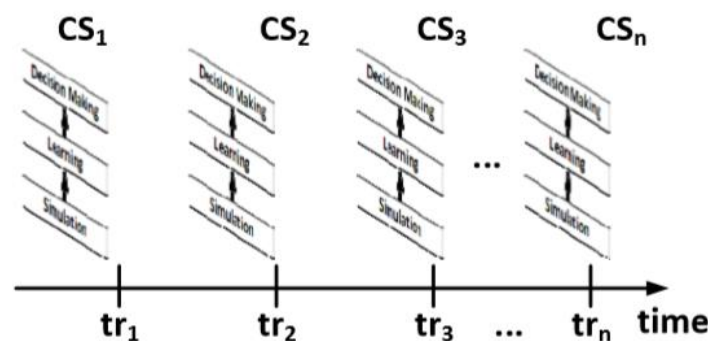


Figura 17 - Double-loop learning, juntamente com processos de Simulação e de Decisão (Putnik and Ferreira 2019)

É ainda de notar que esta arquitetura/modelo satisfaz perfeitamente os requerimentos e visões em relação aos sistemas ciberfísicos, os quais referem que um CPS é um “sistema com feedback loops onde os processos físicos afetam os computacionais, e vice-versa”, assim como já se tinha defendido em tópicos anteriores.

2.1.6.5 CPS-AMS

CPS Augmented Manufacturing Systems resultam da integração da Realidade Aumentada nos sistemas ciberfísicos. A Augmented Reality (AR), também assim chamada, é um sistema que complementa o mundo real com objetos/informações virtuais que parecem coexistir no mesmo espaço que o mundo real (Khalid, Fathi, and Mohamed 2015).

No entanto, para que esta integração seja possível, é necessário que tanto o CPS como a AR apresentem componentes tecnológicos chave. No primeiro caso os componentes tratam-se de:

- Virtual Prototyping: A prototipagem virtual é a exploração de processos de simulação para teste, avaliação e modificação de protótipos em ambientes de design virtual. Envolve, também, a criação de modelos digitais em 3D, 4D, nD e VR que aprimoraram as instalações para novos atributos e comportamentos de exploração e a investigação de cenários de 'e se', permitindo a identificação e a correção de quaisquer problemas antes da construção (Khalid et al. 2015).
- Wireless Sensors: A tecnologia de identificação por radiofrequência (RFID) é uma tecnologia de sensor sem fio que se baseia na deteção de sinais eletromagnéticos por *tags* e leitores (Khalid et al. 2015).
- Mobile Devices: Estes dispositivos são capazes de acomodar vários tipos de informações que podem ser usadas para navegar pelos modelos e permitir a entrada rápida de dados de construção. Além disso, a maioria dos dispositivos móveis possui recursos externos para captura de dados, como scanners de código de barras e leitores RFID (Khalid et al. 2015).
- Communication Networks: A rede de comunicação é uma das tecnologias mais importantes para aperfeiçoar a coordenação bidirecional, a fim de transferir e trocar informações entre dispositivos móveis e fixos. Algumas dessas redes a serem utilizadas incluem a Internet, a rede local sem fios e a rede pessoal sem fios (Khalid et al. 2015).

Por outro lado, em relação à Realidade Aumentada, os componentes chave são:

- Media Representation: Os sistemas de AR usam representações de *media* mais variadas, incluindo texto, símbolo e indicador, imagem 2D, 3D *wireframe*, dados e modelo 3D e também animação, cada uma representando uma classe de informações com características comuns (Khalid et al. 2015).
- Input Mechanism: Nos sistemas de que se fala, a informação virtual é apresentada ao usuário para adicionar à cena do mundo real e, portanto, o

conteúdo da informação é o que importa. Por esse motivo, a maioria das tarefas no uso de um sistema de AR é aceder informações, em vez de manipular e editar conteúdo virtual (objeto) como em um ambiente típico de VR ou 3D CAD (Khalid et al. 2015).

- *Output Mechanism Continuum*: O mecanismo de saída refere-se aos dispositivos e quase todo o trabalho em AR concentrou-se no sentido visual: gráficos virtuais e sobreposições. Contudo, os monitores em AR não precisam de ser apenas visuais, mas também de áudio (Khalid et al. 2015).
- *Trackers*: O registo e o posicionamento precisos de objetos virtuais no ambiente real exigem precisão no rastreamento, bem como na deteção da localização de objetos reais no ambiente. Por tal, o maior obstáculo para a construção de sistemas eficazes de AR é a exigência de sensores e rastreadores precisos e de longo alcance (Khalid et al. 2015).

Tendo todos estes elementos emparelhados, o subsistema de Realidade Aumentada realizará então uma série de tarefas, tal como está esquematizado na Figura 18. Efetivamente, a primeira delas é a captura de imagem, onde o posicionamento do “objeto” é realizado pelas técnicas de processamento de imagem, sendo que a imagem capturada pela câmara servirá como fonte para esta tarefa (Khalid et al. 2015).

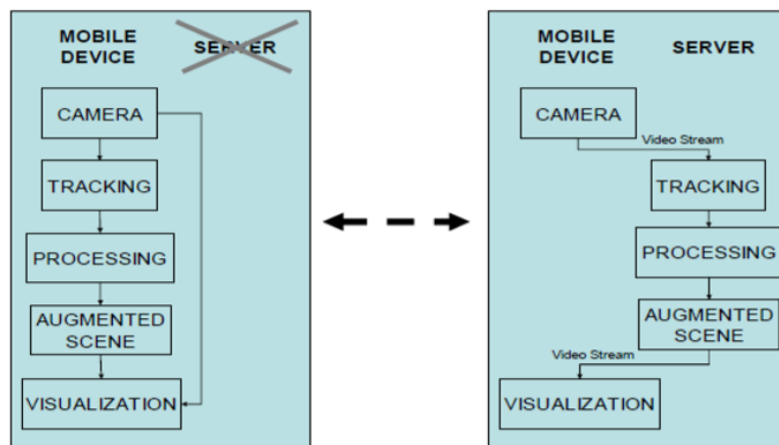


Figura 18 - Arquitetura da camada "Realidade Aumentada" (Khalid et al. 2015)

Posteriormente realiza-se o rastreamento da posição da peça, seguido do processamento e o *rendering* da Realidade Aumentada, acabando com a visualização no dispositivo de saída (Khalid et al. 2015).

2.1.6.6 CPS-VMS

À semelhança do modelo anterior, este também consiste numa integração, mas desta vez da Realidade Virtual com deteção multifacetada (*wearable, implantable, noninvasive* (WIN)). Servindo como auxílio, os dispositivos médicos implantáveis e a identificação por radiofrequência (RFID) também serão integrados ao sistema, podendo ser de três tipos: dispositivos de interação em jogos; sensores médicos; e Mini-leitor RFID (Hu et al. 2017).

Os sinais transferidos entre os dispositivos WIN incluem *inputs* e *outputs*, em que os dados resultantes de tal são recolhidos e enviados para um server de forma a efetuar aprendizagem *symptom-of-interest* (SoI), tal como é mostrado na Figura 19. Além disso, este sistema utiliza ferramentas *machine-learning* para obter fusão de dados, projeção de sinais (para visualizar sinais complexos no subespaço de baixa dimensão), análise de correlação de sinais e reconhecimento de padrões (Hu et al. 2017).

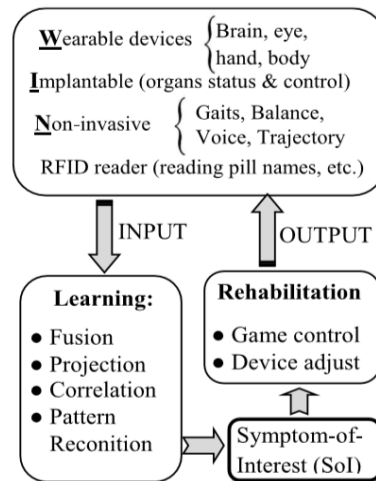


Figura 19 – Arquitetura (geral) de um CPS-VMS (Hu et al. 2017)

Na camada mais baixa da plataforma (Figura 20) é necessário garantir que todos os drivers de *hardware/software* funcionem como o esperado. De forma a assegurar isso, os dispositivos do mesmo subsistema devem funcionar perfeitamente juntos e alguns *threads* de *software* devem poder migrar de um dispositivo para outro no mesmo subsistema.

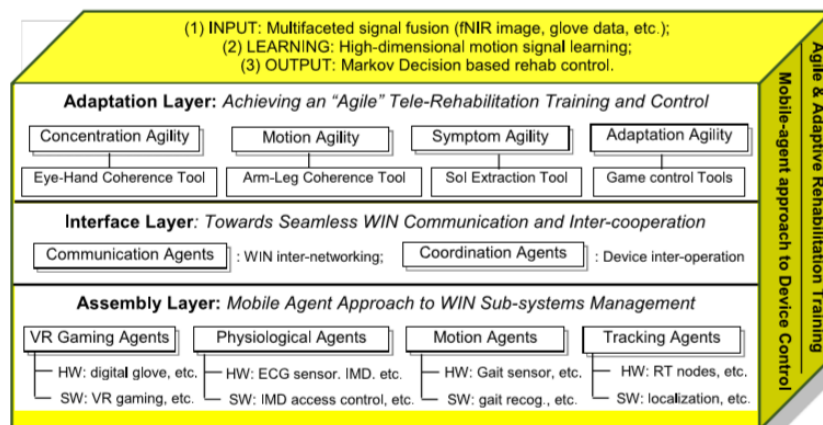


Figura 20 - Arquitetura multifacetada de um CPS-VMS, de três camadas (Hu et al. 2017)

Para uma maior conveniência do gerenciamento de subsistemas e da migração de *threads* de *software*, recorre-se ao uso de uma abordagem MA (*Multi-Agents*) para coordenar todos os dispositivos de deteção e atuação. A troca de “mensagens” e a migração de encadeamentos entre MA’s são implementadas por meio de objetos de mensagem com base num procedimento de interação com o agente de entrega, gravação e escuta (Hu et al. 2017), assim como está exposto na Figura 21.

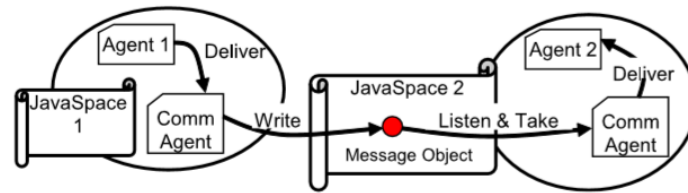


Figura 21 - Troca de "mensagens" entre MA's (Hu et al. 2017)

Nesta arquitetura são quatro as principais categorias de *Multi-Agents*, mais propriamente:

- *VR Gaming Agents*: controlam as funções básicas dos dispositivos de interação baseados em VR, além de serem responsáveis pela recolha de dados (Hu et al. 2017);
- *Physiological Monitoring Agents*: correm nos sensores do corpo para reunir sinais fisiológicos em tempo real (Hu et al. 2017);
- *Motion-Disorder Detection Agents*: funcionam com sensores não invasivos para monitorar o distúrbio de movimento do paciente e o status de desequilíbrio corporal (Hu et al. 2017);
- *Patient Tracking Agents*: verificam as operações dos leitores/tags RFID (Hu et al. 2017).

Agrupando todos estes elementos descritos anteriormente, obtemos então um CPS-VMS prático, em que o seu “desdobramento” está ilustre na Figura 22.

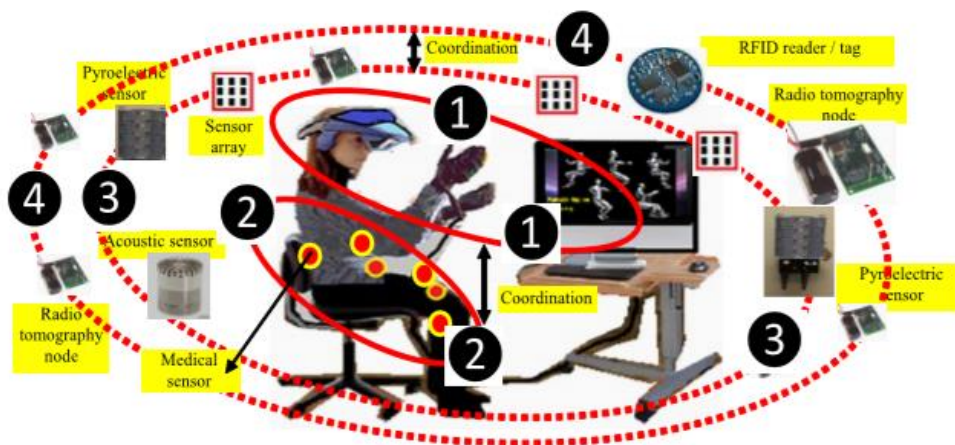


Figura 22 - Desdobramento de um CPS-VMS (1-Agentes VR; 2-Agentes fisiológicos; 3-Agentes de movimento; 4-Agentes de rastreamento) (Hu et al. 2017)

2.2 Open Design

2.2.1 Openness

Openness é uma característica relativa que se refere ao grau em que algo é acessível para visualizar, modificar e usar, sendo estas as três operações fundamentais implícitas na acessibilidade. Efetivamente, a capacidade de visualizar refere-se ao compartilhamento de conteúdo e à disponibilidade de informações detalhadas sobre o assunto, enquanto que a de modificar entende-se como a partilha de trabalho e capacitação de mudanças, melhorias e extensões do assunto. Por último, a competência de usar refere-se ao compartilhamento da propriedade e à possibilidade de reutilização, semi ou irrestrita, do objeto ou de partes dele (Klaassen, Evers, and Troxler 2018).

Tooze et al. (2014) defendem que para um trabalho ser considerado “aberto” terá de cumprir uma série de regras/requisitos, mais concretamente, ser acessível a toda sociedade; redistribuível; reutilizável; atribuível; integrável; não ser discriminatório a pessoas ou grupos; não ser discriminatório contra campos de atuação; e as suas respetivas licenças não devem ser restritas. Resumidamente, não devem existir restrições que limitem quem e como uma obra é usada e, se existir uma licença para a mesma, ela não deve afetar as outras condições que definem um trabalho como “aberto”.

Do ponto de vista social, o *Openness* é uma característica central de uma infraestrutura que transmite e reforça o compartilhamento, a reciprocidade, a colaboração, a tolerância, a equidade, a justiça e a liberdade (Klaassen et al. 2018). No entanto, isto não significa que seja de domínio público, visto que este último conceito apresenta diversas formas de propriedade intelectual e direitos que poderiam ser assumidos (Castro, Putnik, Castro, and Fontana 2019).

Já em termos sociopolíticos, Dahlander and Gann (2010) argumentam quatro razões que justificam a união monetária que o tema em questão apresenta, sendo elas:

- reflete mudanças sociais e económicas nos padrões de trabalho, onde os profissionais procuram carreiras em carteira, em vez de um emprego vitalício com um único empregador;
- a globalização expandiu a extensão do mercado, o que permite uma divisão maior do trabalho;
- instituições de mercado aprimoradas, como Direitos de Propriedade Intelectual (IPR), Capital de Risco (VC) e padrões de tecnologia, permitem que a organização troque ideias;
- novas tecnologias possibilitam novas maneiras de colaborar e coordenar através de distâncias geográficas.

Desta forma, a aplicação do *Openness*, conforme implícita em vários recursos de acessibilidade, a um número crescente de práticas ubíquas centrais que impulsionam o empreendimento humano, transformou-se numa megatendência que pode ser

rotulada como “*Rise of Open-X*” (Klaassen et al. 2018). De facto, são várias as variações de tal: *Open Design*, *Open Data*, *Open Source*, *Open City*, *Open Innovation*, *Open Government*, *Open Education*, *Open Science*, etc (Castro, Putnik, Castro, and Bosco Fontana 2019).

O fenómeno o qual o *Open-X* está associado são tendências generalizadas que têm um grande impacto e provavelmente afetarão todos os níveis - indivíduos, organizações, mercados, países e sociedade civil - por um longo período. Compreender as megatendências e os seus efeitos contínuos pode fornecer informações valiosas para o desenvolvimento de cenários futuristas e, posteriormente, ajudar a moldar as ações atuais em antecipação a esse futuro (Klaassen et al. 2018).

Com a finalidade de corroborar a ideia anterior, uma análise de artigos relacionados indica que o modificador “aberto” geralmente indica melhor, mais barato e mais rápido. Aparentemente, as qualidades inerentes à abertura materializaram-se como facilitadores subjacentes que pavimentam o caminho para a criatividade, a inovação e a prosperidade (Klaassen et al. 2018). Até ao momento, conforme descrito na Tabela 1, o *Open-X* efetivou-se em várias configurações que podem ser classificadas de acordo com três arquétipos: *open innovation*, *open source* e *open design* (Klaassen et al. 2018). A mesma tabela (Tabela 1) mostra-nos também as suas diferentes propostas de valor e impulso, orientação central e principais atores envolvidos.

Tabela 1 - Os três arquétipos do Openness, baseado em (Klaassen et al. 2018)

	<i>Open Innovation</i>	<i>Open Source</i>	<i>Open Design</i>
<i>Propostas de Valor e Impulso</i>	Conhecimento Distribuído	Desenvolvimento Distribuído	Produção Distribuída
<i>Orientação Central</i>	Visualizar	Modificar	Usar
<i>Principais Atores envolvidos</i>	Organizações	Comunidades de desenvolvedores	Consumidores

2.2.2 Open Innovation

No que diz respeito ao primeiro arquétipo, e de acordo com a doutrina tradicional, os líderes da indústria criam mais e melhores ideias, o que dita que a inovação deve, portanto, ser promovida pelas equipas internas de desenvolvimento atrás de altos muros organizacionais e protegida como segredo comercial. Todavia, o *open innovation* baseia-se na ideia de que os líderes da indústria fazem o melhor uso de ideias internas e externas para desenvolver melhores modelos de negócios (Klaassen et al. 2018).

De facto, a premissa básica desta variante de *Open*, defendida por Huizingh (2011), é abrir o processo de inovação, que, por sua vez, trata-se do uso de entradas e saídas propositais de conhecimento para acelerar a inovação interna e expandir os mercados para uso externo da inovação, respetivamente. Por outras palavras, um resultado

superior deve ser esperado com limites permeáveis entre uma empresa e o seu ambiente, que permitam o fluxo de ideias, a troca de conhecimentos e o comércio de propriedade intelectual, tal como diz Klaassen et al. (2018).

Para um melhor entendimento do *open innovation*, recorreu-se a Chiaroni, Chiesa, and Frattini (2011), mais propriamente à Figura 23, disponível nesse mesmo artigo. Primeiramente, o conceito é dividido em duas dimensões básicas, nomeadamente o *outside-in* (de fora para dentro) e o *inside-out* (de dentro para fora). De seguida, o processo de transição de *Closed* para *Open* é interpretado como um procedimento de mudança organizacional, o que permite identificar as diferentes etapas pelas quais uma empresa se conforma ao novo paradigma de inovação. Por último, também podemos ver que são introduzidas e discutidas as principais alavancas nas quais os gerentes intervêm para realizar a transição de um modelo de inovação fechado para aberto (Chiaroni et al. 2011).

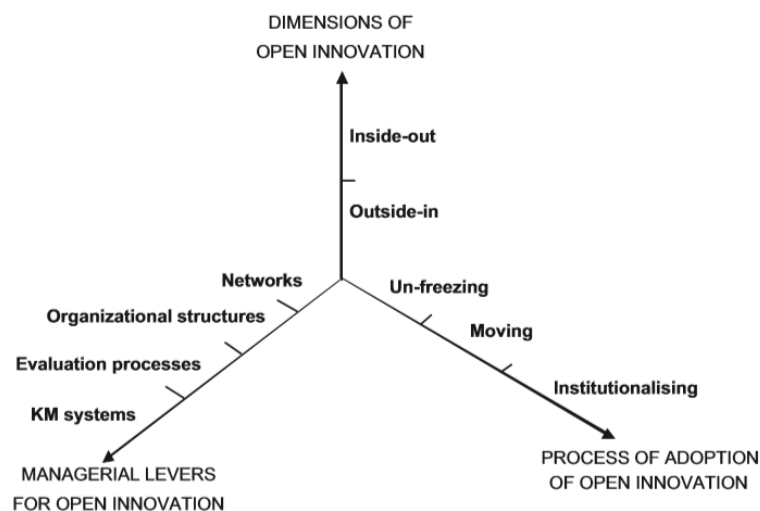


Figura 23 – Framework (teórico) do Open Innovation (Chiaroni et al. 2011)

A proposição de valor e o impulso da *inovação aberta* são processos de "conhecimento distribuído" que enfatizam os recursos de "abertura" relacionados à visão, em que os principais atores são as organizações. Ao alcançar e explorar recursos externos de conhecimento, este tipo de inovação estende os recursos geradores e inovadores de uma empresa, promovendo também a proliferação de comunidades de prática e o lançamento das bases do *crowdsourcing* (Klaassen et al. 2018).

A Tabela 2 resume as principais características do paradigma do *Open Innovation* em relação ao modelo do *Closed*.

Tabela 2 - *Closed vs Open innovation, baseado em (Greco, Locatelli, and Lisi 2017)*

<i>Closed Innovation</i>	<i>Open Innovation</i>
O conhecimento externo tem um papel complementar em relação ao conhecimento intraempresa;	O conhecimento externo é tão importante quanto o conhecimento intraempresa;
A contratação e o financiamento de talentos de classe mundial levarão às inovações necessárias à empresa;	Modelos de negócios precisam ser desenvolvidos para converter esforços de P&D em criação de valor;
As repercussões são potencialmente prejudiciais e o vazamento de conhecimento deve ser evitado;	As empresas precisam gerenciar propositadamente os fluxos de saída de conhecimento e tecnologia;
A P&D interna pode fornecer a vantagem competitiva sem a necessidade de interagir com o ecossistema de inovação externo;	A maior parte do conhecimento desenvolvido em quase todas as disciplinas fica fora dos limites de uma empresa;
A gestão da propriedade intelectual é entendida como uma ferramenta defensiva.	O gerenciamento da propriedade intelectual pode ter um papel proativo e diferenciado.

2.2.3 Open Source

Em relação ao *open source*, o conceito teve origem na indústria de software e, segundo a doutrina tradicional, este é desenvolvido em empresas comerciais (de software) por pessoal profissional, protegido por medidas legais e técnicas e, em seguida, licenciado por uma taxa. Em contraste, e de acordo com o modelo de negócios de *open source*, o software é desenvolvido por meio de produção coordenada por voluntários independentes (Klaassen et al. 2018).

Com esta ideologia, está claro que os principais atores do são os desenvolvedores. Por tal, o Open Source Definition (OSD) implica que esse software possuía uma licença que permite modificações e deve exigir redistribuição gratuita do software sob a mesma licença. Além disso, estas mesmas licenças também desempenham a função, e talvez a mais importante, de garantir acesso não exclusivo à propriedade intelectual (Team 2014). Subsequentemente, todos podem aceder livremente o código-fonte e modificá-lo e redistribuí-lo sob os mesmos termos, alimentando assim ciclos contínuos de melhoria, adaptação e extensão de maneira distribuída (Klaassen et al. 2018).

Derivado do *Open Source Software* (OSS), o *Open Source Hardware* (OSH) é um artefacto físico cuja documentação é, à semelhança do OSS, liberada sob uma licença que concede a qualquer pessoa direitos de produção e distribuição (Bonvoisin and Mies 2018). No entanto, Kim and Shin (2016) defendem que, fazendo uma comparação entre OSS e OSH, as características do *Open Source Hardware* parecem ser extremamente desvantajosas. Na verdade, a comunidade tem sido cética em relação ao OSH, porque é relativamente mais difícil copiar e modificar produtos físicos do que

em software. Outro motivo é o custo e a estrutura de distribuição, na qual o OSS ganha vantagem, comparativamente ao concorrente (Kim and Shin 2016).

Em jeito de resumo, a proposição de valor e o impulso do *open source* são processos de "desenvolvimento distribuído" que enfatizam os recursos de "abertura" relacionados à modificação, abrangendo e explorando recursos externos de desenvolvimento, o que amplia os recursos geradores e inovadores de um projeto principal. Com isto, as doutrinas do modelo de desenvolvimento, licenciamento e distribuição de "código aberto" promoveram a proliferação de projetos de todos os tipos relativos a este código (Klaassen et al. 2018).

2.2.4 Open Design

O conceito de *Open Design* é baseado na configuração do *open-source* e na redefinição dos direitos de propriedade intelectual e industrial, através da Internet, para facilitar o desenvolvimento colaborativo de hardware e sistemas (Castro, Putnik, Castro, and Fontana 2019), apoiado por uma comunidade que compartilha informações (Castro, Putnik, Castro, and Bosco Fontana 2019). Segundo Castro, Putnik, Castro, and Bosco Fontana (2019), com esta variante de Open estamos também a redefinir duas dimensões, mais especificamente o processo e os direitos. A primeira significa que o desenvolvimento do produto é feito de uma forma colaborativa e partilhada, enquanto que a segunda indica que os modelos de propriedade intelectual são mais diversos e, num caso mais "extremo", até podem banir os direitos e tornar-se domínio público (Castro, Putnik, Castro, and Bosco Fontana 2019).

De forma a dar seguimento à ideia anterior, o *open design* implica que os modelos estejam disponíveis ao público, compartilháveis, licenciados sob termos de acesso aberto e distribuídos digitalmente num formato de especificação geral de design, como, por exemplo, "dxf" e "dwg" (Klaassen et al. 2018). Efetivamente, qualquer pessoa pode usar o "design aberto" como elemento funcional de um sistema proprietário ou não proprietário, desde que o elemento deste design seja claramente identificado. No entanto, qualquer indivíduo ou organização que use ou modifique um design aberto deve concordar com os termos especificados na licença do mesmo (Vallance, Kiani, and Nayfeh 2001), que, por sua vez, são:

- a documentação de um design está disponível gratuitamente;
- qualquer pessoa é livre para usar ou modificar o design alterando a documentação do design;
- qualquer pessoa é livre para distribuir os designs originais ou modificados (mediante taxa ou gratuitamente);
- modificações no design devem ser devolvidas à comunidade (se redistribuídas).

Estando disponível para qualquer um, o potencial inerente de reconfiguração e extensão de um open design orientado ao usuário reforça os recursos geradores e inovadores dos consumidores, tornando-os, assim, numa das peças principais. De facto, embora os designers tenham um papel fundamental na promoção do "design aberto", produzindo e compartilhando projetos de design adequados, a verdadeira

razão do open design existir são os consumidores que se envolvem na produção distribuída (Klaassen et al. 2018).

Explicando melhor este último ponto, de acordo com a doutrina tradicional, o design é sobretudo um estágio preliminar antes da fabricação e distribuição comercial. Porém, o *open design* é direcionado aos consumidores envolvidos na fabricação, ignorando os canais convencionais de fabricação e distribuição (Klaassen et al. 2018). Com isto, o modelo relativo ao design em questão diminui a cadeia de valor vertical tradicional formada pelas relações designer-fabricante-distribuidor-consumidor e oferece uma rede aberta alternativa entre designers e consumidores. Os relacionamentos breves, transitórios e não hierárquicos resultantes criam matrizes dinâmicas e flexíveis de projetos que não são apenas centrados no usuário, mas também orientados ao usuário (Klaassen et al. 2018).

Além das características atrás descritas, Klaassen et al. (2018) também referem que o “design aberto” abrange uma infinidade de considerações: por exemplo, especificação do projeto, fabricação, ação colaborativa, gestão da cadeia de abastecimentos e respetivos valores, modelos de negócios, aspetos legais, infraestrutura tecnológica e valores normativos. Os mesmos autores argumentam que a complexidade dessa ecologia pode ser decomposta, em certa medida, classificando os problemas subjacentes em quatro camadas conceituais interdependentes, detalhadas na Tabela 3.

Tabela 3 - As diferentes camadas do Open Design, baseado em (Klaassen et al. 2018)

<i>Practise Layer</i>	As práticas de trabalho permitem e restringem a conceção dos processos de design. Esta camada abrange a cultura de <i>open design</i> , ou seja, a nomenclatura relacionada, padrões profissionais, artesanato, regras de comércio, código de conduta, rituais e valores normativos;
<i>Object Layer</i>	<i>Blueprints</i> configuráveis e extensíveis que estão disponíveis sob licença de acesso aberto em repositórios públicos online;
<i>Process Layer</i>	Os meios de produção que permitem e restringem a produção dos objetos de design;

Infrastructure Layer

Os fundamentos institucionais e técnicos que permitem e restringem a vitalidade das práticas de design, bem como o sistema jurídico relacionado, a estrutura de mercado e a arquitetura técnica que governam as atividades de design aberto e o crescimento futuro.

Tendo também em conta a complexidade abordada, Tooze et al. (2014) apresentam novos termos e definições relacionados com o design colaborativo e *openness*, a fim de falar e escrever sobre o open design com uma melhor clareza. Efetivamente, os termos denotados são:

- ***Open design solution***: um conjunto de planos e instruções que permitem que outras pessoas utilizem as informações de design para criar ou modificar a 'solução de design' sem restrição. Uma solução de design é algo que pode ser atuado diretamente e, no contexto de coisas físicas, ser feito;
- ***Open design contribution***: qualquer contribuição, em qualquer formato, para um processo de design que seja disponibilizado para uso irrestrito;
- ***Open design process***: o desenvolvimento de uma solução ou soluções de design através de contribuições (*open design contributions*);
- ***Open designing***: envolver-se em qualquer tipo de design por meio de um processo (*open design process*);
- ***Open design project***: qualquer projeto que tenha como base um *open design process*.

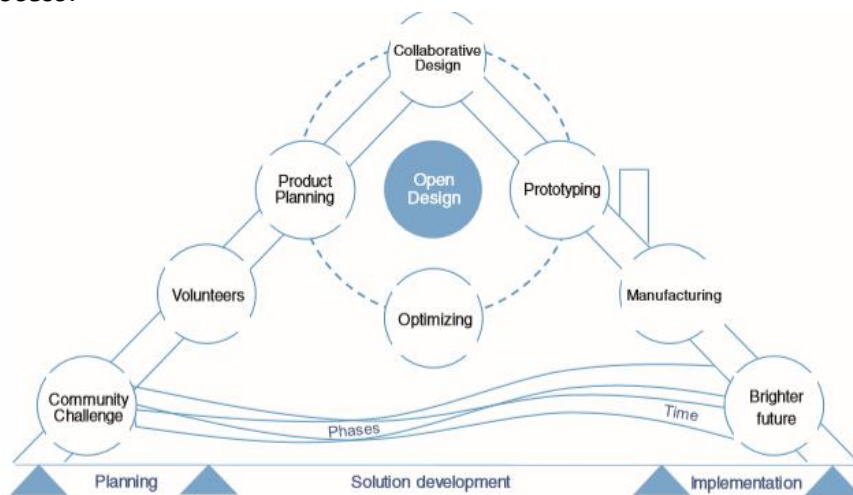


Figura 24 - Fases de um processo relativo ao Open Design (Rebensdorf et al. 2015)

No que diz respeito aos processos de design aberto, qualquer um deste tipo envolve três fases – planeamento, desenvolvimento de soluções e implementação (Rebensdorf et al. 2015), tal como está esquematizado na Figura 24. Além disso, para ilustrar os componentes abertos de um processo de design é útil visualizar as várias partes e as

possíveis variações. Assim sendo, a Figura 25 mostra modelos simplificados de como soluções de design abertas e não abertas podem ser geradas por contribuições abertas e não abertas (Tooze et al. 2014).

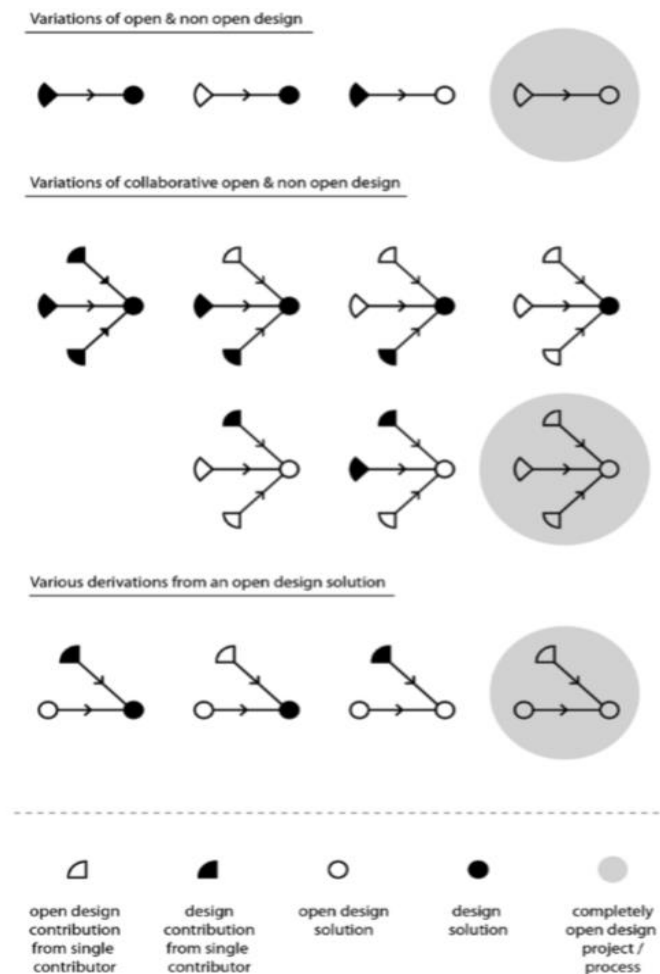


Figura 25 - Possíveis variações de um processo de Open Design (Tooze et al. 2014)

Um outro aspeto positivo do *open design*, e realçado por Klaassen et al. (2018), é o facto de este estar a abrir caminho para a próxima iteração na grande mudança de modelos de negócios *push* para *pull*. Efetivamente, de um modo geral, os modelos de negócios *push* são baseados em cadeias de valor descendentes, onde uma linha de produtos produzidos em massa é amplamente distribuída, por meio de técnicas de marketing *downstream* direcionadas ao valor. Por outro lado, os modelos *pull* apoiam-se em cadeias de valor ascendentes, onde uma linha de produtos configurados pelo cliente é distribuída individualmente por meio de técnicas de marketing *upstream* orientadas a recursos. Comparando ainda as duas faces desta transição, enquanto os modelos *push* são baseados em economias de escala e enfatizam a eficiência de custos, os *pull* debruçam-se na produção flexível e evidenciam a personalização em massa (Klaassen et al. 2018).

Em jeito de resumo, realça-se a relação direta com a comunidade, em parte permitida pelas licenças de “projeto aberto”, as quais foram projetadas para permitir que os subsistemas “proprietários” e “gratuitos” sejam integrados harmoniosamente

(Vallance et al. 2001). De facto, a rapidez com que o *open design* cresce e transforma o seu estado, criando simultaneamente algumas vantagens para promover a comunidade e a plataforma de OD, modela uma possível transição de fase, a partir da qual o sistema alcança a sustentabilidade (Castro, Putnik, Castro, and Fontana 2019).

Relativamente ao futuro do tema em estudo, o caminho a percorrer é auspicioso. Uma prova disso mesmo são as novas abordagens de design que estão a ser desenvolvidas, orientadas para fornecer soluções no ambiente turbulento e complexo atual, tentando responder às necessidades reais, como sustentabilidade, competitividade e inovação. Além do mais, o open design é, ultimamente, apresentado como uma abordagem que compreende um conjunto de características intrínsecas que oferecem um potencial para transformar a visão tradicional de design e produção (Castro, Putnik, Castro, and Bosco Fontana 2019).

Tabela 4 - *Open Design vs Closed Design, adaptado de (Klaassen et al. 2018)*

	Open Design	Closed Design
Acesso	Disponível, compartilhável, licenciado sob termos de acesso aberto;	Oculto, protegido, licenciado por uma taxa;
Blueprints	Especificadas pela linguagem comum de notação digital;	Especificado pela linguagem proprietária de notação digital;
Derivados	Reconfigurável e extensível;	“Opaca” e fixa;
Exclusividade	Reproduzível;	Limitado a uma série finita ou única;
Meios de Produção	Fabricado por máquinas comerciais, “prontas para uso” ou multifuncionais;	Fabricado com mão de obra artesanal, máquinas ou moldes personalizados
Processo de Produção	Sujeito a produção distribuída e escalável;	Sujeito a produção em lote controlada centralmente e predefinida;
Potencial	Generativo.	Fechado.

Considerando as diferenças, evidenciadas na Tabela 4, entre o *open design* e o design tradicional (*closed design*), é importante promover o primeiro para a produção. Na verdade, esta necessita de “experimental” e “viver” o design aberto, a fim de encontrar uma nova maneira de reinventar e ampliar novos modelos de negócios (Castro, Putnik, Castro, and Fontana 2019), bem como abrir novas perspectivas e novas

oportunidades, sendo provável que gere uma maior apreciação do papel dos designers (Klaassen et al. 2018). Com esta simbiose está também implicada a exploração de áreas não tão bem desenvolvidas, como é o caso das quatro camadas associadas ao *open design*, já descritas anteriormente, o que seria uma mais valia para ambas as partes (Klaassen et al. 2018).

2.3 Encadeamento dos Sistemas Ciberfísicos no Open Design

Primeiramente, é de realçar que ainda não existe qualquer trabalho desenvolvido nesta matéria na literatura. À vista disso, será agora feito o vínculo dos Sistemas Ciberfísicos para o *Open Design*, tendo em conta as características principais destes dois temas.

Efetivamente, este último é um conceito, tal como o próprio nome indica, muito aberto, indefinido e de vasta abrangência. Melhor dizendo, o *design* tanto pode ser bastante simples como de enorme complexidade. Com isto, é possível afirmar que para a produção aberta necessitamos, em teoria, de um sistema ciberfísico com essas mesmas características, isto é, com capacidade de se adaptar ao meio envolvente e que também consiga aprender com tais circunstâncias, levando então a um maior crescimento por parte do CPS.

Outro aspeto já aqui mencionado é o “nível” de aprendizagem que um sistema pode adotar, mais propriamente *single-loop learning* e *double-loop learning*. Aprofundando melhor estes temas, o primeiro ocorre quando erros ou problemas são detetados e corrigidos e as organizações continuam com suas políticas e objetivos. Nestes casos, as atividades para corrigir erros e resolver os problemas aumentam a base de conhecimento ou as competências ou rotinas específicas da empresa, sem alterar a natureza fundamental das atividades da organização (Kantamara and Ractham 2014). Por outro lado, o *double-loop learning* ocorre quando, além da deteção e correção de erros, a organização está envolvida no questionamento e modificação das normas, procedimentos, políticas e objetivos subjacentes de uma organização, o que requer alterar as competências ou rotinas da base de conhecimento ou específicas da empresa (Kantamara and Ractham 2014).

Relacionando estes dois conceitos (Figura 27 e Figura 26), o último mostra-se mais adequado para consumir a turbulência existente e as mudanças invariáveis no ambiente; capacitado para a mudança e adaptação ao meio caótico; e também capaz de desenvolver novo conhecimento organizacional, que é consequentemente integrado ao modelo de aprendizagem da organização (Kantamara and Ractham 2014). Por outras palavras, *single-loop learning* pode ser considerado como um entrave à aprendizagem organizacional, enquanto que o *double-loop learning* engrandece essa mesma aprendizagem. Desta forma, e embora este critério tenha um menor peso que o anterior, chega-se à conclusão, em tese, que o modelo ideal para o *Open Design* é um pertencente ao domínio dinâmico e que, simultaneamente, apresente *double-loop learning*.

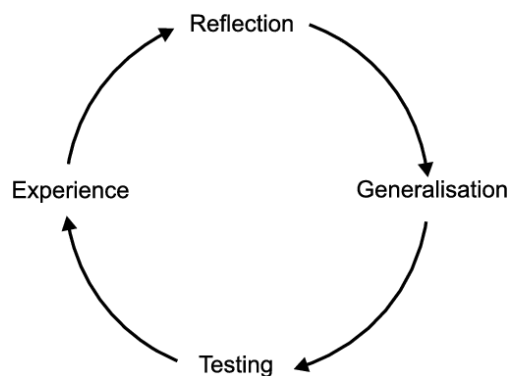


Figura 26 - Single-Loop Learning (Forrester, Swartling, and Lonsdale 2008)

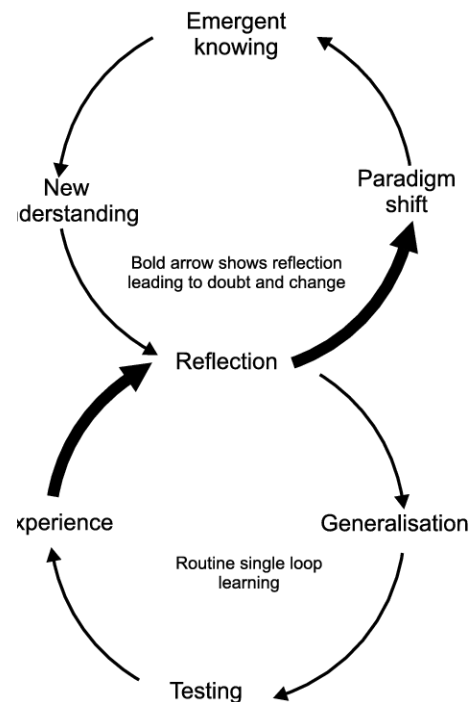


Figura 27 - Double-Loop Learning (Forrester et al. 2008)

Numa fase futura, esse mesmo modelo ideal poderá ser disponibilizado seguindo os princípios do *Open Source Software*, fundamentados e justificados no tópico “Open Source”, de forma a serem igualmente estudados e, posteriormente, desenvolvidos ainda mais por alguém da comunidade.

Em jeito de ilação, impor o nexos em estudo, de uma forma mais enfática, poderia ser um paradigma transformador que poderia levar a um conceito verdadeiramente novo na indústria. Além disso, conseguiria aprimorar sistemas avançados de produção, como empresas distribuídas, virtuais e ubíquas, devido à abordagem progressiva que leva a um acesso realmente aberto a dados, produtos, recursos e, provavelmente, a uma produção totalmente aberta.

DESENVOLVIMENTO

3	ARQUITETURA PROPOSTA DE UM SISTEMA CIBERFÍSICO ORIENTADO À PRODUÇÃO ABERTA	38
3.1	Estrutura global da arquitetura proposta	38
3.2	Sistema computacional em Nuvem (<i>cloud</i>)	42
3.3	Sistema Multiagente	56
3.4	Sistema Físico	64
4	IMPLEMENTAÇÃO DA ARQUITETURA PROPOSTA	76
4.1	Sistema Físico	77
4.2	Sistema computacional em Nuvem (<i>cloud</i>)	109

3 Arquitetura proposta de um sistema ciberfísico orientado à produção aberta

Dando seguimento ao tópico anterior, a arquitetura proposta apresenta um conjunto de características que torna o sistema ciberfísico flexível, dinâmico, inteligente e robusto. Efetivamente, a ideia exposta procura o equilíbrio entre o *software* e o *hardware*, de forma a fornecer uma reconfiguração intuitiva nas linhas de produção em massa, bem como a viabilização do uso de diferentes tecnologias e aplicações em vários propósitos diferentes (Ramião, Oliveira, and Rocha 2017).

Além disso, a arquitetura baseia-se num sistema multiagente de modo a maximizar a capacidade do mesmo, o que significa que este estará preparado para a adaptação em caso de distúrbios e mudanças inesperadas, devido à autonomia das entidades que constituem o subsistema.

A abordagem descrita aproxima-se das exploradas em alguns trabalhos, como é o caso de, por exemplo, João Ricardo Cardoso, Marcelo Götz, and João Netto (2018); Ramião et al. (2017); e em alguns segmentos do livro Shcherbakov, Glotov, and Cheremisinov (2020). Desta forma, estes artigos (entre outros), funcionam como uma excelente base para a elaboração da arquitetura pretendida.

No entanto, e visto que a proposta está orientada para o *Open Design*, as ideias referidas são complementadas com conceitos que permitem uma aprendizagem colaborativa dentro dessa comunidade, uma vez que o nosso sistema estará capacitado para agregar toda a informação disponível sobre a mesma, tirar conclusões através dos resultados obtidos no decorrer da produção e, posteriormente, divulgar abertamente essas ilações, no mesmo meio. Com isto, é fortemente viabilizado o conhecimento comunitário, que é uma das grandes premissas do tema em questão.

Claro está que para que tudo seja possível, estes conceitos necessitam de estar alinhados e a trabalhar em conjunto, diga-se. Nesse âmbito, é detalhado e justificado o funcionamento do sistema num seu todo.

3.1 Estrutura global da arquitetura proposta

Começando pelo fator que molda a disposição do sistema, o produto é transposto para o mesmo através do repositório Open Design. Juntamente com o produto, estará também incluída toda a informação necessária para o seu fabrico, mais propriamente, uma lista interna de operações, que, por sua vez, será enviada para o Sistema *Multi-Agente* (MAS).

Efetivamente, o MAS, também referido como o cérebro do sistema global e responsável pela sua gestão, processará a informação referida e organizá-la-á com o auxílio da virtualização de todos os elementos do sistema de Produção. Desta forma

são conseguidas tomadas de decisão mais precisas, bem como as ações necessárias sem afetar mais recursos do que os necessários (Ramião et al. 2017).

No entanto, as tarefas acarretadas requerem um poder de processamento e uma capacidade de armazenamento acima dos limites dos de um dispositivo IoT, o que os torna num impasse nesse aspeto. Por essa razão, propõe-se o recurso ao *Cloud Computing*, uma vez que apresenta essas características como praticamente ilimitadas, permitindo também um acesso conveniente, escalável e passivo a um conjunto de recursos computacionais configuráveis (Diaz, M., Martin, C., Rubio 2016).

Com este caminho, significa que a parte de maior toma de decisão do Sistema *Multi-Agente* encontra-se alocado na nuvem, devido às razões descritas. Contudo, isto não retira a utilidade dos dispositivos IoT. Aliás, estes têm uma igual importância relativamente ao funcionamento do sistema, visto que estes serão os únicos com capacidade de interagir diretamente com o hardware.

Esta simbiose trata-se da integração da IoT na nuvem, também conhecida por *Cloud of Things*, e possibilita, acima de tudo, tirar uma maior vantagem do uso deste tipo de dispositivos. Além disso, resolve as limitações quanto à computação e ao acesso e análise de dados (Diaz, M., Martin, C., Rubio 2016), oferecendo assim uma plataforma que dispõe alta escalabilidade, armazenamento e processamento uniforme em tempo real (Tsai, Sun, and Balasooriya 2010).

A integração em ênfase acaba por servir como a ponte de ligação entre o sistema ciber e o físico. Efetivamente, e tal como já foi referido, os dispositivos IoT têm como principal função comunicar aos recursos físicos quais as operações a executar, que é conseguido através da parte do sistema multiagente responsável pela devida distribuição da informação processada.

Relativamente aos recursos físicos, estes podem ser impressoras 3D, braços mecânicos, mecanismos de perfuração (geralmente CNC's), tapetes de transporte ou até mesmo outro dispositivo/ferramenta, o que significa que não há propriamente restrições quanto ao tipo de recursos em foco. Tal é justificado pelo maior leque de habilidades que o conjunto enumerado oferece ao sistema, estando assim preparado para uma, também, vasta variedade de produtos.

Aqui entra um outro aspeto característico da arquitetura proposta, nomeadamente a modularidade, que consiste na ideia de cada componente do sistema ser um módulo independente. Isto permite que a gestão realizada pelo MAS redirecione o produto para outro módulo de produção capaz de realizar a habilidade pretendida. Além disso, oferece ainda a vantagem de uma fácil e rápida integração e remoção de recursos de produção sem interromper o sistema de produção, sendo, no entanto, necessário que o módulo informe o sistema da sua habilidade e da sua posição no mesmo, bem como o sistema tem de ser capaz de detetar automaticamente a integração ou remoção do módulo (Ramião et al. 2017).

Em termos práticos, a modularidade é extremamente útil em caso de produção em série em que um recurso esteja ocupado ou em caso de avaria onde o produto tem de ser redirecionado, dado que, geralmente, existe mais do que um recurso a conceder a mesma habilidade. Todavia, para que tal seja possível, é necessário que as interfaces dos componentes sejam genéricas, daí a proposta se debruçar na ligação do sistema multiagente através dos dispositivos *IoT* com os recursos físicos.

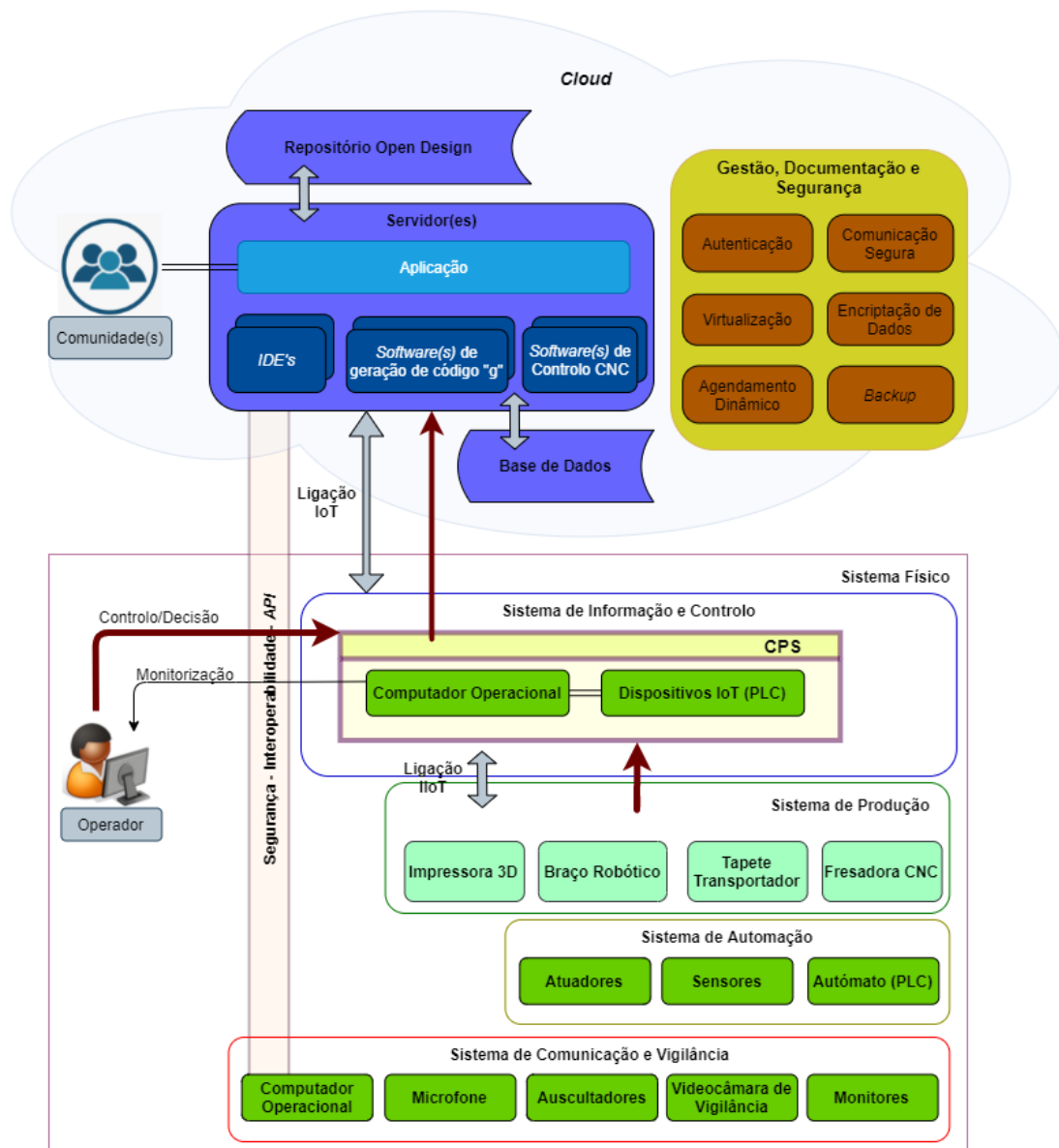


Figura 28 - Arquitetura proposta

O referido anteriormente encontra-se representado na Figura 28, servindo também como um resumo de tal informação. Ou seja, o utilizador escolhe um ou mais produtos a fabricar por meio do repositório *Open Design*, os quais contêm a lista interna de operações realçada, e de seguida o sistema organiza os produtos de acordo com as habilidades necessárias e as disponíveis fisicamente. De forma a atribuir as devidas funções aos recursos físicos, os dispositivos *IoT* usam as suas interfaces para se ligarem aos últimos, que por sua vez terão habilidades diferentes até que sejam capazes de

garantir a construção pretendida. Atendendo à adoção da modularidade na arquitetura proposta, é possível existirem vários desses dispositivos, o que significa que os mesmos podem estar a produzir vários produtos em simultâneo, recorrendo ao MAS para garantir que os recursos físicos são distribuídos entre as várias tarefas a serem feitas (Ramião et al. 2017).

Até agora descreveu-se o funcionamento do sistema unicamente num sentido, isto é, começando no lançamento do produto no repositório e acabando nos recursos físicos. Contudo, o projeto apresentado viabiliza uma nova vertente, sendo esta regida pelo controlo humano. Efetivamente, o utilizador consegue definir o quão automático o sistema é, sendo assim possível acompanhar a produção em tempo real e fazer ajustes/alterações assim que se pretender, auxiliado pelo operador. Tal flexibilidade permite ainda que se altere completamente o sentido, ou seja, existe a capacidade de fabricar um produto de autoria própria, registar todos os processos e etapas envolventes e, posteriormente, disponibilizá-las abertamente no repositório *Open Design*, estando, assim, acessível a toda comunidade.

Ao abranger todo este panorama de diferentes possibilidades quanto à dinâmica do funcionamento do sistema, surge simultaneamente um novo caminho que nos proporciona um conhecimento mais aprofundado dos conceitos congruentes. De facto, esse conhecimento é traduzido, em termos práticos, na capacidade do sistema em tomar nota de todas as ações tomadas, compará-las a ações anteriores ou até mesmo às partilhadas pela comunidade e, filosoficamente falando, conseguir aprender com o passado comum, tornando-se assim num sistema inteligente. Isto acontece uma vez que o *Deep Learning* permite que modelos computacionais compostos por várias camadas de processamento aprendam representações de dados com vários níveis de abstração (Lecun, Bengio, and Hinton 2015).

Além de divulgada (abertamente), esta aprendizagem é também interiorizada no próprio sistema multiagente, servindo como um forte *background* para os agentes responsáveis pelo processamento e organização da informação. Tal como mostra a Figura 28, o trabalho mútuo destes dois componentes tem como fruto uma tomada de decisão, que, à priori, será aquela que obterá melhores resultados.

Todos estes procedimentos associados ao *machine learning* são automáticos e à base de algoritmos, o que torna a solução unicamente tecnológica. No entanto, se atendermos que as pessoas são recursos cada vez mais essenciais, melhores atores e que devem ser considerados utilizadores ativos em todo o sistema, a participação de tais torna-se essencial, bem como a sua eventual co-decisão. Com isto em vista, a arquitetura proposta incorpora um sistema de comunicação, por meio de uma aplicação, que enriquece essas perspetivas tecnológicas com perspetivas sociais, nas quais o utilizador pode interagir facilmente com o sistema e, naturalmente, com outros utilizadores no processo de co-criação, tal como defende Ferreira et al. (2017), bem como o(s) operador(es) encarregues pelo sistema físico.

Resumidamente, o recurso a estas duas vertentes de aprendizagem permite a interoperabilidade entre sistemas, uma vez que a tecnologia presente no sistema é moldada em torno dos conceitos pragmáticos introduzidos, o que pode significar essencialmente a capacidade de comunicação entre seres humanos. Assim, mais uma vez, estamos perante uma característica do sistema que vai ao encontro das premissas da comunidade, sobretudo o conhecimento comunitário, sendo, então, uma mais-valia para a mesma.

Por fim, dá-se foco àquilo que, metaforicamente, é o sangue do nosso organismo, nomeadamente a informação. Efetivamente, durante este tópico foi salientada inúmeras vezes a importância da informação ou, de uma forma geral, dos dados que correm no sistema, fazendo deles o bem mais valioso. Posto isto, a proposta oferece um sistema de informação que garante a segurança desses mesmos dados. Por segurança entenda-se os conceitos necessários para que a informação (dados) esteja disponível, certa e legível apenas a quem de direito. Esta definição aplica-se indistintamente aos dados que residem num sistema bem como aos dados em trânsito, isto é, quando são enviados de um sistema para outro (Laudon 2003).

Para assegurar tal, o sistema de informação subdivide-se em Segurança e Gestão/Documentação. A primeira oferece comunicação segura, encriptação e backup de dados, enquanto que a segunda apresenta agendamento dinâmico, autenticação e virtualização de dados, assim como está ilustre na Figura 28.

3.2 Sistema computacional em Nuvem (*cloud*)

A *cloud*, ou, neste caso de estudo, mais propriamente o *Cloud Computing* é um modelo que permite um acesso, conveniente e “mediante o pedido”, à rede a um conjunto compartilhado de recursos de computação configuráveis (por exemplo: redes, servidores, armazenamento, aplicações e serviços), que, por sua vez, podem ser rapidamente provisionados e lançados com o mínimo esforço de gestão ou interação do provedor de serviços (Dillon, Wu, and Chang 2010). Esse conjunto de recursos apresenta-se ainda como virtualmente ilimitado em termos de capacidade de armazenamento e processamento (Diaz, M., Martin, C., Rubio 2016).

Surgindo como um novo modo de computação, a computação em nuvem é uma integração de várias tecnologias importantes, incluindo *grid computing*, *utility computing*, *automation computing*, virtualização de hardware e serviço de internet (Luo et al. 2016), tal como está representado na Figura 29.

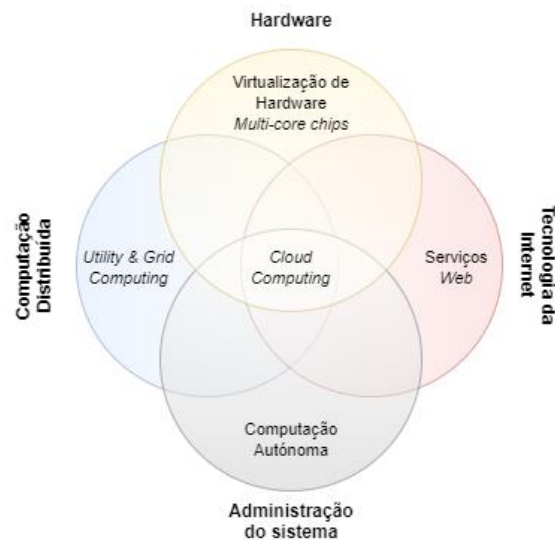


Figura 29 - Tecnologias centrais do *Cloud Computing*, adaptado de (Luo et al. 2016)

Associados a este método de computação, estão cinco elementos essenciais (Dillon et al. 2010), nomeadamente:

- Auto atendimento mediante pedido;
- Amplo acesso à rede;
- Agrupamento de recursos;
- Rápida elasticidade;
- Serviço medido/controlado.

As definições, e respetivas características, referidas neste tópicos agregam-se, formando, assim, arquiteturas baseadas em nuvem e congruentes estratégias de desenvolvimento e segurança, assim como Ferreira et al. (2017) apresentam na Figura 30.

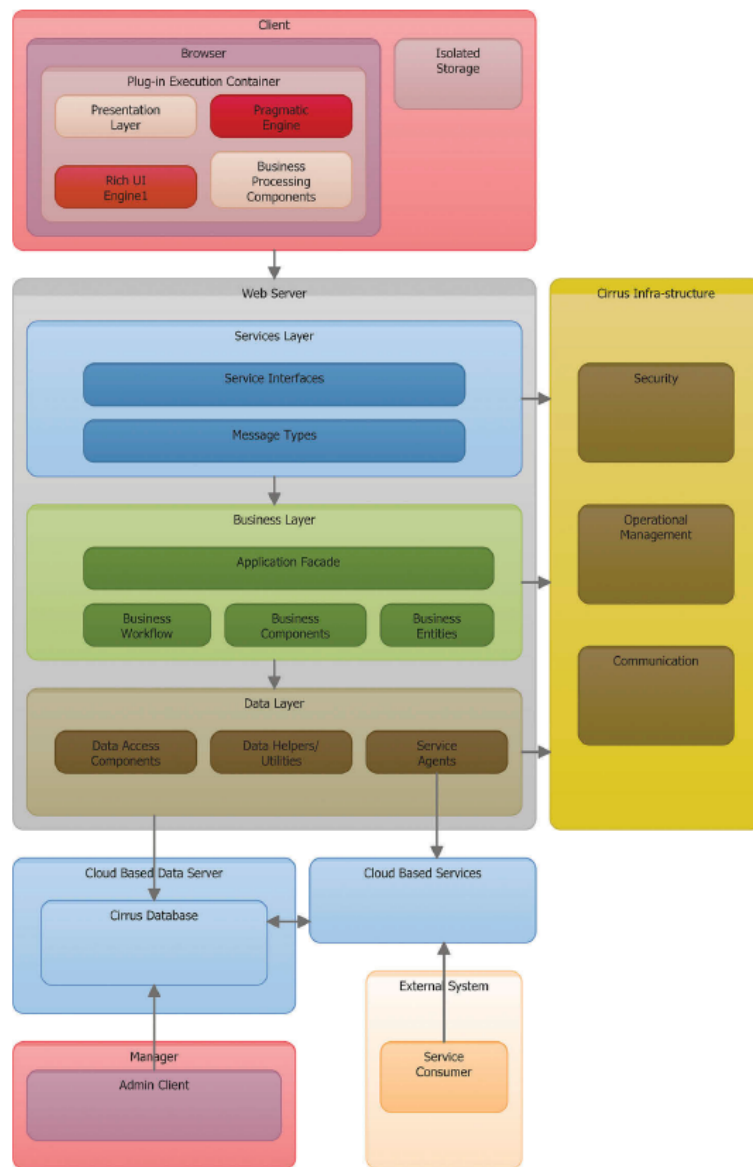


Figura 30 - Arquitetura baseada em nuvem (Ferreira et al. 2017)

No entanto, existem diferentes tipos de categorias na computação em nuvem (Diaz, M., Martin, C., Rubio 2016), hierarquicamente ilustres na Figura 31:

- Infraestrutura como Serviço (IaaS) - a camada mais baixa na infraestrutura da *Cloud* e oferece uma variedade de Máquinas Virtuais para computação e armazenamento;
- Plataforma como Serviço (PaaS) - a camada intermediária que permite o desenvolvimento de aplicações;
- Software como Serviço (SaaS) - a camada superior, a qual oferece aplicações acessíveis aos diversos utilizadores.

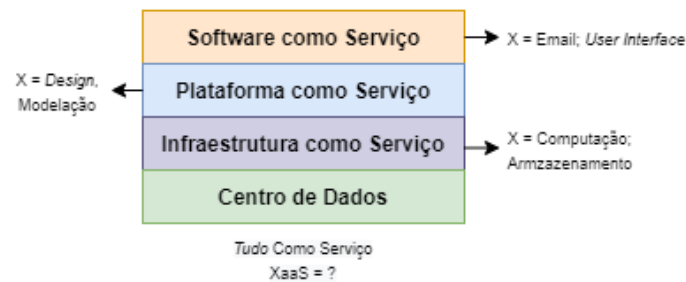


Figura 31 - Visão hierárquica da computação em nuvem, adaptado de (Tsai et al. 2010)

Desta forma, a nível funcional e a fim de suportar os serviços do mundo físico, o sistema recorre, então, à *cloud*, a qual corresponde a uma, e primeira, camada (Figura 32) da arquitetura proposta inicialmente. Efetivamente, o processo dá início nessa mesma camada, mais propriamente num repositório, onde se encontra disponível uma vasta gama de produtos, personalizados ou não, facultados pela comunidade, a qual lhe é garantida acesso livre. Toda a informação agregada é preservada pelo(s) servidor(es) e seus serviços de gestão.

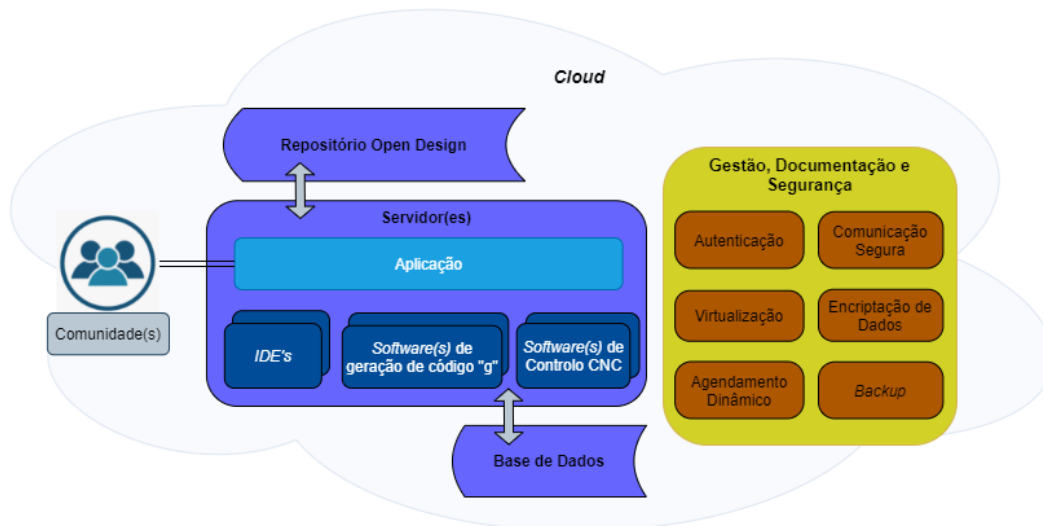


Figura 32 - Arquitetura correspondente à *cloud* do sistema

Posteriormente, dá-se uma ligação utilizador-servidor (*frontend*) suportada por uma aplicação, a qual permite ao primeiro escolher o produto desejado, monitorizar a sua fabricação continuamente e até mesmo comunicar e colaborar com outros usuários e/ou o operador, por meio de canais síncronos de comunicação direta. Para responder ao pedido, o servidor invoca uma série de serviços (*backend*), apoiado por serviços de interoperabilidade (Blackstock and Lea 2014)(Otto et al. 2013) (*REST*, *Microserviços*, entre outros) e, se necessário, recorre a aplicações de geração de código G (Mohamed et al. 2019).

Assim, toda a informação é devidamente processada e organizada, de modo que esteja pronta para ser distribuída para o sistema físico, por intermédio de software de controlo CNC (Erwinski et al. 2013) ou infraestrutura(s) IoT.

3.2.1 Repositório

Um repositório é um banco de dados compartilhado de informações sobre artefactos de engenharia produzidos ou usados por uma empresa, o que leva a uma variedade de requisitos para a estruturação e acesso ao banco de dados. Exemplos de tais artefactos incluem *software*, documentos, mapas, sistemas de informação e componentes/sistemas manufaturados discretos (Bernstein and Dayal 1994).

Relativamente à arquitetura, os repositórios, na sua generalidade, seguem um modelo de três camadas composto por uma camada de apresentação, uma camada de gestão do repositório e uma camada de armazenamento. Para esta última, recorre-se, frequentemente, a meta-modelos, tal como está expresso nas Figura 33 e Figura 34. Por sua vez, os meta-modelos são os dados que descrevem os meta-dados e essa mediação baseada nos primeiros melhora a flexibilidade na escolha de esquema de dados e linguagem de consulta comuns, além de fornecer um alto nível de interoperabilidade (Gao and Krogstie 2010).

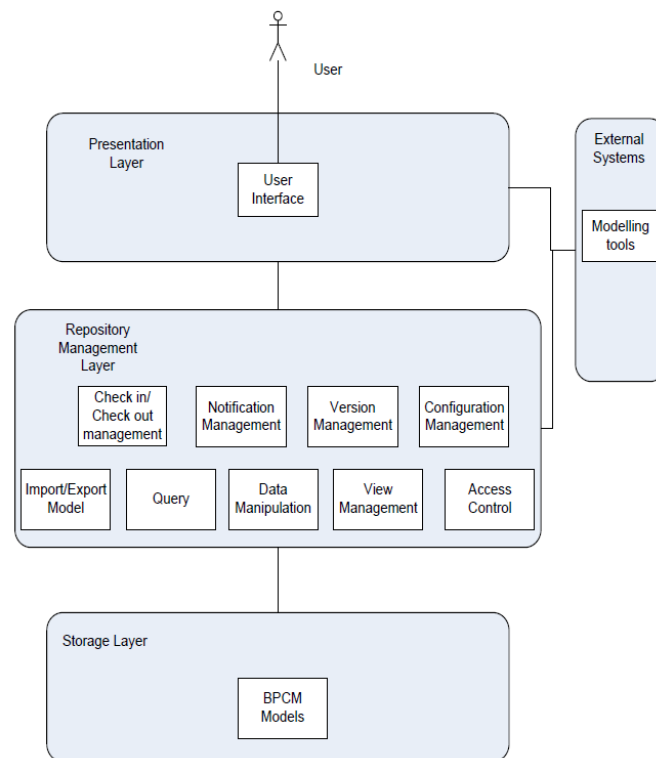


Figura 33 - Arquitetura do repositório BPCM (Gao and Krogstie 2010)

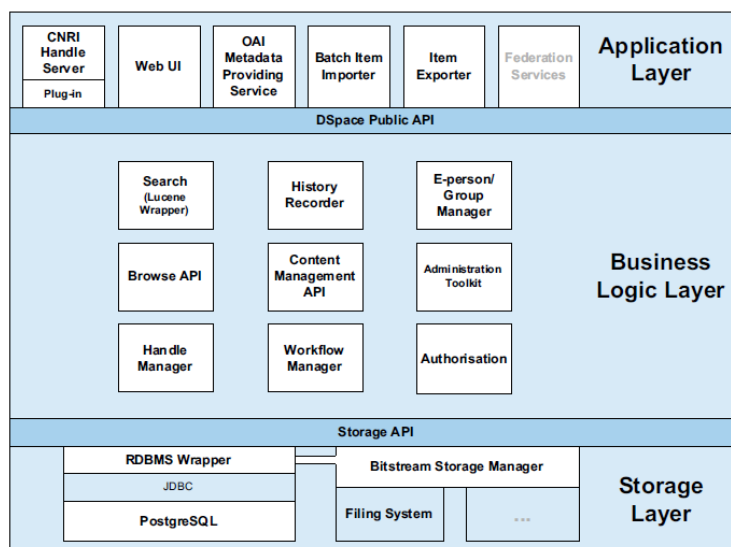


Figura 34 - Arquitetura do repositório *DSpace* (Tansley et al. 2003)

Armazenar modelos num repositório comum normalmente traz vários benefícios. Primeiramente, torna-se fácil para outros utilizadores reutilizarem os modelos relevantes no futuro. Em segundo lugar, os modelos no repositório estão sujeitos a serviços de controlo comuns, o que permite que diferentes direitos de acesso possam ser concedidos a diferentes grupos de utilizadores, configurando controlos de acesso ao repositório. Por fim, fornece uma plataforma com o âmbito da comunidade partilhar entre si informações para que possam se beneficiar uns dos outros (Gao and Krogstie 2010).

Em suma, ao promover o compartilhamento de dados por meio de modelos comuns de dados e informações, e impor um conjunto comum de serviços de controlo, um repositório é a peça central de um ambiente integrado no qual uma coleção dinâmica de ferramentas pode trabalhar em conjunto (Bernstein and Dayal 1994).

É, então, neste tipo de banco de dados e informações, que os ficheiros, associados tanto aos produtos/projetos como aos recursos físicos, serão descarregados, aquando do propósito de fabricar um produto ou da disposição de um novo recurso (físico). Toda essa informação é providenciada pela própria comunidade, à qual lhe é concedido livre acesso ao repositório em foco.

Desta forma, o repositório inerente ao sistema está orientado para toda a comunidade, viabilizando, também, o *Open Design*, uma vez que todos os produtos/projetos, quer a nível documental ou de código-fonte, encontram-se digitalmente abertos a toda a comunidade.

3.2.2 Servidor (es)

De seguida ocorre a interação cliente-servidor, a qual se baseia num modelo de computação onde o servidor hospeda, fornece e gere a maioria dos recursos e serviços a serem consumidos pelo cliente (Nagappan and Vayiravan 2015), entre os quais, aplicações, comunicações, segurança, sistemas e serviços de gestão de rede (Berson

1996). O modelo em questão contém um ou mais computadores-clientes conectados a um servidor central por meio de uma rede ou conexão com a internet, o que o torna igualmente um modelo de computação em rede ou, então, rede cliente-servidor (Nagappan and Vayiravan 2015).

É importante compreender que um servidor é um conceito arquitetónico e não uma descrição de uma implementação física, no sentido em que o mesmo dispositivo físico pode fornecer funções de cliente e servidor. Efetivamente, o sistema cliente-servidor pode ser definido como uma arquitetura de software composta tanto pelo cliente quanto pelo servidor, em que os clientes enviam somente solicitações enquanto o servidor responde às mesmas. Relativamente à referida arquitetura, tal é usualmente constituída por um servidor de aplicação, um servidor de banco de dados e um computador, podendo ser definida em dois ou três níveis (Oluwatosin 2014), assim como mostra a Figura 35 e Figura 36, respetivamente.

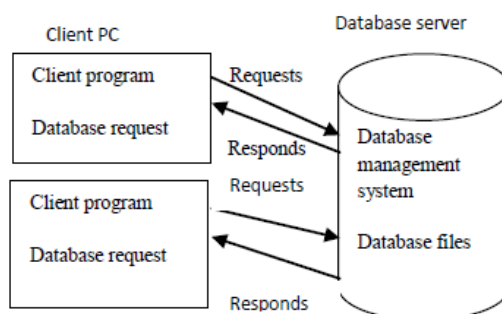


Figura 35 - Arquitetura Cliente-Servidor de dois níveis (Oluwatosin 2014)

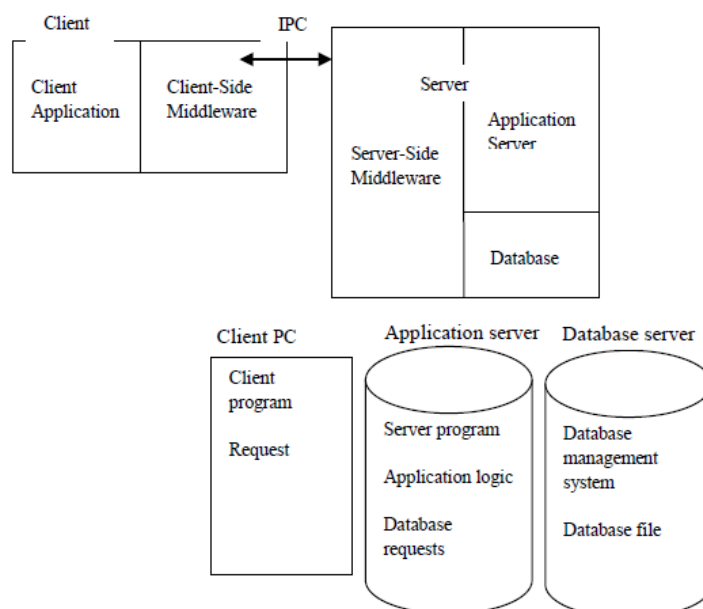


Figura 36 - Arquitetura Cliente-Servidor de três níveis (Oluwatosin 2014)

Para responder ao pedido do cliente, o servidor segue um protocolo generalista, o qual consiste na receção do pedido, processamento do mesmo, recorrendo a serviços *backend*, e o retorno do resultado final, o qual será enviado ao cliente. Deste processo

resulta um fluxo de informação entre os vários recursos, assim como está demonstrado a título exemplificativo na Figura 37.

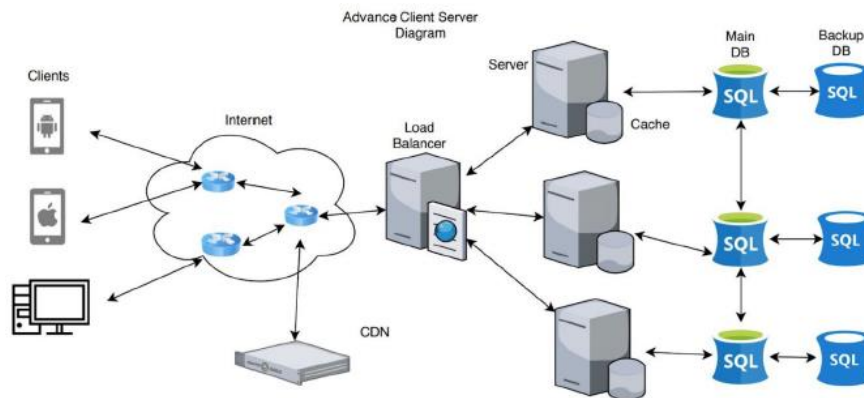


Figura 37 - Diagrama de fluxo Avançado (Nagappan and Vayiravan 2015)

Para que a comunicação entre os vários serviços *backend* seja facilitada, é necessário um nível base de interoperabilidade, que é definida como a capacidade de dois ou mais componentes de software cooperarem, apesar das diferenças de linguagem, interface e plataforma de execução (Wegner 1996). No entanto, em contexto de IoT este conceito ganha ainda mais relevo, uma vez que, para atingir o seu potencial máximo, não é suficiente que as “coisas” estejam apenas conectadas à Internet, elas também precisam de estar disponíveis, acessíveis, geridas e potencialmente conectadas a outras 'coisas'. Por essa razão, neste contexto, é requerido um nível de interoperabilidade mais aprofundado (Blackstock and Lea 2014).

Entre os demais serviços de interoperabilidade, destacam-se o *REST* e os *Microserviços*. O primeiro trata-se de um estilo de arquitetura para sistemas hipermídia distribuídos, como é o caso da *World Wide Web*, em que toda a sua interação resume-se ao acesso a recursos (Haupt et al. 2014). Por outro lado, o segundo refere-se a um estilo arquitetónico inspirado na computação orientada a serviços, o qual consiste num processo coeso e independente que interage por meio de mensagens e que é, geralmente, conceitualmente implementado de forma isolada e equipado com ferramentas de persistência de memória dedicadas (Dragoni et al. 2017).

Neste caso em particular, o sistema defendido serve o pedido do cliente, realizado através de uma aplicação que funciona como *frontend*, recorrendo a *IDE's*, *software* de geração de *código-G* e *software* de controlo *CNC*, que, por sua vez, se tratam dos principais serviços *backend*. Para suportar os últimos, o servidor apoia-se, então, em serviços de interoperabilidade, em que os referidos *REST* e *Microserviços* surgem como exemplos.

3.2.2.1 Aplicação

Dando agora foco ao pedido do cliente, o mesmo é realizado por meio de uma aplicação que funciona como *frontend*. Numa primeira vertente, sendo ela a nível de

ficheiros, documentação, dados, etc., o cliente, ou neste caso, qualquer utilizador da comunidade, comunica ao servidor qual o produto pretendido, o qual se encarregar-se-á de processar devidamente toda a informação anexa. Posteriormente, a comunicação dá-se no sentido contrário, ou seja, é o servidor que dá uma resposta ao cliente. No entanto, o servidor não se limita a uma simples resposta inicial. Efetivamente, o mesmo providencia ao cliente um acompanhamento contínuo e em tempo real do processo de produção do mesmo produto, bem como a sua monitorização, tratando-se, assim, de uma também resposta contínua.

Por outro lado, e para humanizar o sistema, a arquitetura proposta precisa de suportar a colaboração real e síncrona de humano para humano que permite a coexploração do sistema com outros agentes, mais concretamente, humanos. Desta forma, a arquitetura necessita de ser efetivamente comunicacional, o que significa que os serviços de comunicação não serão suportados apenas como transacionais (troca de informações), mas sim para terem também participação e colaboração humana direta em qualquer fase (Ferreira et al. 2014).

Na arquitetura apresentada, a participação ativa do utilizador é essencial para superar problemas de interoperabilidade tecnológica. Além disso, e para garantir a eficácia do sistema, as pragmáticas envolvidas devem ser suportadas, assim como a geração de mecanismos eficazes semelhantes a *middleware* deve ser possível. Isto significa que não é apenas disponibilizado *chat* ou videoconferência como serviços, mas também é providenciada uma ação imediata, se necessário, promovendo, assim, a codecisão e a cocriação ou *codesign* do mecanismo de interoperabilidade (Ferreira et al. 2017).

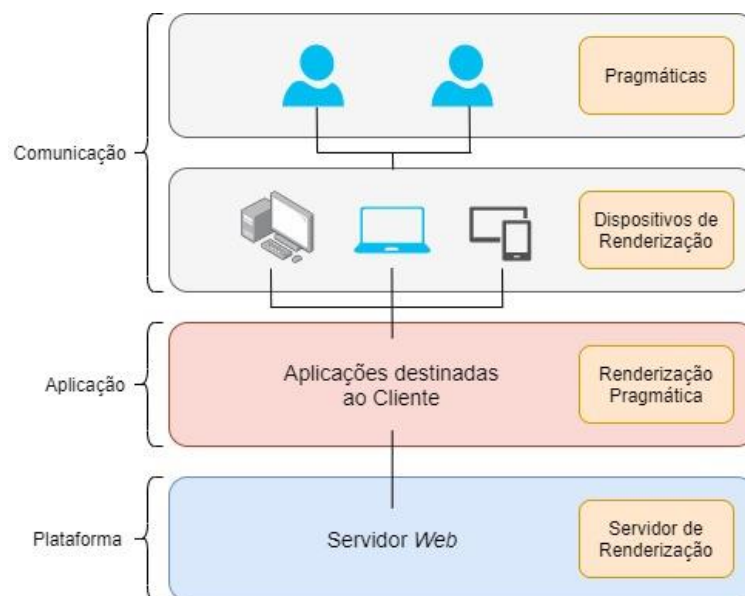


Figura 38 - Arquitetura Comunicacional, adaptado em Ferreira et al. (2017)

Como tal, a aplicação em foco surge, então, como o *middleware* pertencente ao modelo desta vertente comunicacional, ou então, como define Ferreira et al. (2017), da arquitetura comunicacional, assim como mostra a Figura 38.

3.2.2.2 IDE's

Para facilitar o desenvolvimento da aplicação previamente referida, programar muitas das ações executadas pelos vários elementos do sistema físico e viabilizar o desenvolvimento de um sistema *Multi-Agente*, o sistema auxilia-se de Ambientes de Desenvolvimento Integrado (*IDE's*). Efetivamente, um *IDE* é uma área de trabalho baseada em interface gráfica do utilizador (*GUI*) projetada para auxiliar no desenvolvimento de aplicações de *software* com um ambiente integrado, combinado com todas as ferramentas necessárias à mão, entre as quais se destacam o editor, compilador, vinculador e depurador (*debugger*) (Mutiara, Refianti, and Witono 2014). O descrito encontra-se exemplificado na Figura 39.

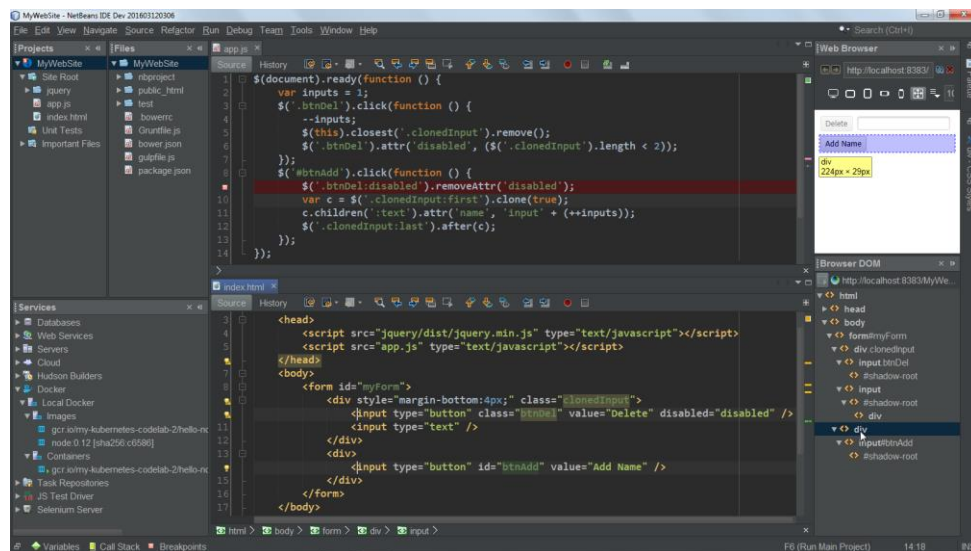


Figura 39 - Exemplo de um *IDE* (Wielenga 2016)

Além dessas, este tipo de ambiente de desenvolvimento fornece ferramentas que automatizam tarefas comuns, como é o caso da reestruturação, preenchimento automático e correção de erros de compilação, as quais têm o objetivo de aumentar a velocidade e reduzir os erros do desenvolvedor (Muşlu et al. 2012).

3.2.2.3 Software gerador de código-G

O *código G* é uma linguagem de programação de baixo nível e surge como linguagem intermediária entre o controlador e a máquina em si. Efetivamente, todos os movimentos feitos pela máquina são instruídos por meio do *código G*, o qual opera a partir das coordenadas e pontos que levam à geração de caminhos da ferramenta (*toolpaths*). Este tipo de controlo é também conhecido como Controlo Numérico Computorizado (CNC) e trata-se de um conceito moderno nas indústrias de manufatura e produção (Mohamed et al. 2019).

A Tabela 5 mostra os principais comandos referentes ao código G.

Código	Função		
G00	Interpolação linear em avanço rápido.	G51	Escala de um contorno.
G01	Interpolação linear em avanço programado.	G50.1	Cancela o espelhamento.
G02	Interpolação circular em sentido horário.	G51.1	Espelhamento de um contorno.
G03	Interpolação circular em sentido anti-horário.		
G10	Deslocamento de zero peça no programa	G54	Definição do zero peça em relação ao zero máquina. (G54 até G59)
G17	Plano de trabalho em X e Y. (ferramenta paralela a Z)	G74	Ciclo de rosqueamento rosca esquerda.
G18	Plano de trabalho em X e Z. (ferramenta paralela a Y)	G80	Desliga os ciclos de furação.
G19	Plano de trabalho em Y e Z. (ferramenta paralela a X)	G81	Ciclo de furação simples.
G20	Programação em polegadas.	G82	Ciclo de Furação com faceamento.
G21	Programação em milímetros.	G83	Ciclo de furação profunda .
G28	Deslocamento da ferramenta até o zero-máquina.	G84	Ciclo de rosqueamento rosca direita.
G40	Usinagem sem compensação do raio da ferramenta.	G90	Sistema de coordenadas absolutas.
G41	Usinagem com compensação do raio da ferramenta à esquerda do contorno.	G91	Sistema de coordenadas relativas ou incrementais.
G42	Usinagem com compensação do raio da ferramenta à direita do contorno.	G94	Avanço da ferramenta de corte em mm/min.
G43	Compensação do comprimento da ferramenta.	G95	Avanço da ferramenta de corte em mm/rot.
G50	Cancela o valor da escala.		

Tabela 5 - Comandos principais de *Código G*

Atualmente, os programas de código G são gerados usando as ferramentas de manufatura auxiliada por computador (CAM) (Figura 40), usando os dados do projeto geométrico das ferramentas de desenho assistido por computador (CAD) como entrada. Isto é, o software CAM analisará e determinará o tipo de ferramenta e criará os caminhos da ferramenta com base nos recursos desejados e no modelo CAD. Para o poder fazer da melhor forma, esse processo requer um pós-processador CAM que gere o dialeto exato do código G usado pela máquina em questão (Mohamed et al. 2019).

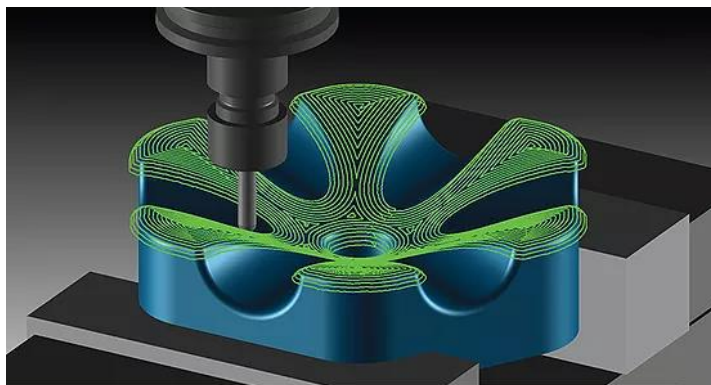


Figura 40 - Software CAD/CAM (4ieng 2019)

Como é de senso comum, as ferramentas CAM estão associadas à manufatura subtrativa, que se trata do corte/rasgo de um pedaço de metal ou plástico com, por exemplo, uma fresadora. No entanto, também é possível se estar perante outro tipo de manufatura: a aditiva. Efetivamente, a igualmente denominada de impressão 3D refere-se a qualquer um dos vários processos em que o material é unido ou solidificado sob controlo computacional para criar um objeto tridimensional, adicionando-se o material, normalmente, camada por camada (Gujar et al. 2019), tal como demonstra a Figura 41.

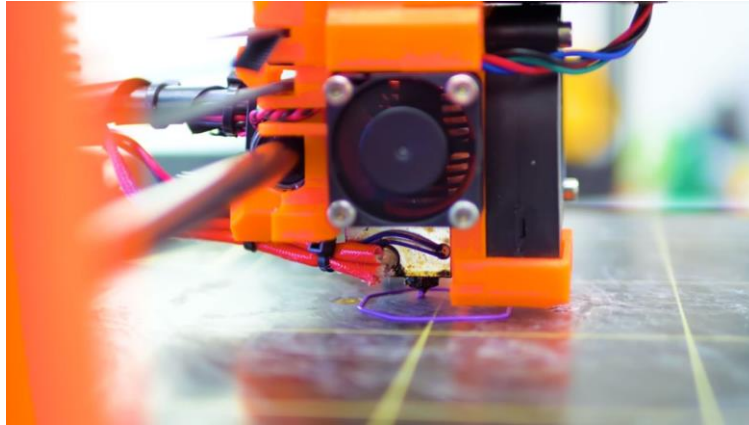


Figura 41 - Manufatura Aditiva (Impressão 3D) (All3DP 2021a)

Como tal, o processo de geração do *código G* é também ele diferente. De facto, nestes casos, recorre-se a *Slicers* (Figura 42), uma vez que são capazes de converter ficheiros *STL* em *código G* (Gujar et al. 2019). Para um melhor entendimento, os ficheiros *STL* são obtidos através de software de modelação 3D e apresentam representação de faceta triangular das superfícies, o que é fundamental para a prototipagem rápida (impressão 3D), visto que cada camada é uma secção transversal 2D da malha 3D correspondente ao *STL* (Szilvsi-Nagy and Matyasi 2003).

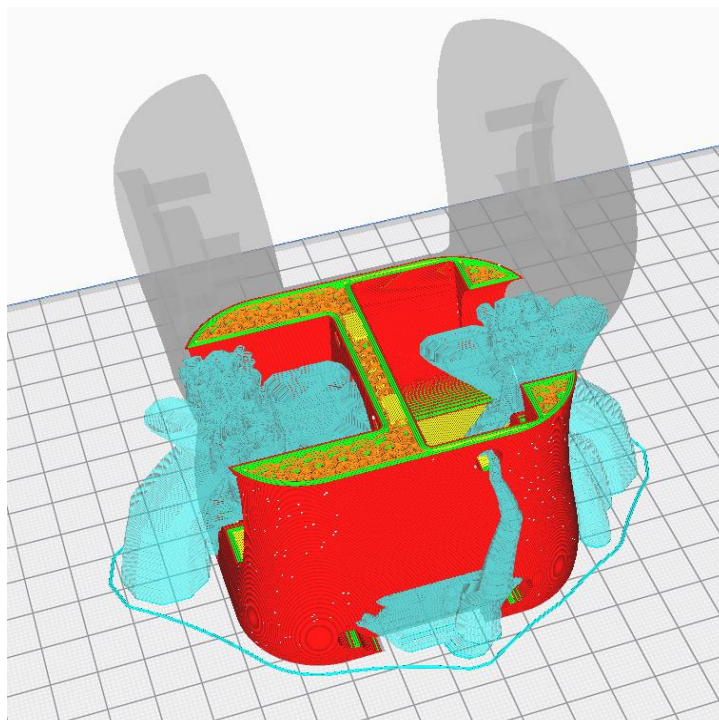


Figura 42 - Software Slicer

Em termos práticos, e visto que o sistema está preparado para os métodos de manufatura mencionados, o mesmo recorre, então, a ferramentas CAM e a *Slicers*, caso o ficheiro relativo ao produto não se encontre no formato pretendido, neste caso, *código G*.

3.2.2.4 Software de Host e Controlo (NC)

Para dar seguimento ao fluxo da informação, o sistema recorre, então, a *software* de *host*, o qual envia o *código G*, gerado ou obtido, para os respetivos recursos físicos. Posteriormente, um analisador de *código G* de alto desempenho, instalado nos mesmos, converte as trajetórias desejadas do próprio código em pulsos discretos para controlar cada motor de passo-a-passo (Peele et al. 2015).

Simultaneamente ao descrito, é feito um controlo do funcionamento dos recursos físicos em execução. Efetivamente, no caso da fresadora CNC, o controlo é realizado em tempo real e em modo multitarefa, sendo elas o controlo multicanal, interpolação multi-eixo, servo-controlo de alta velocidade e alta precisão e o controlo lógico (Yuhan, Jun, and Ye 2003).

Por outro lado, e no que toca a impressoras 3D, existe o *software* de controlo da impressora, exemplificado na Figura 43, que fornece uma interface em tempo real para as funções e configurações da impressora. Além disso, monitoriza o estado do trabalho de impressão, bem como a própria impressora, principalmente a temperatura da extrusora e a da base (Gujar et al. 2019).

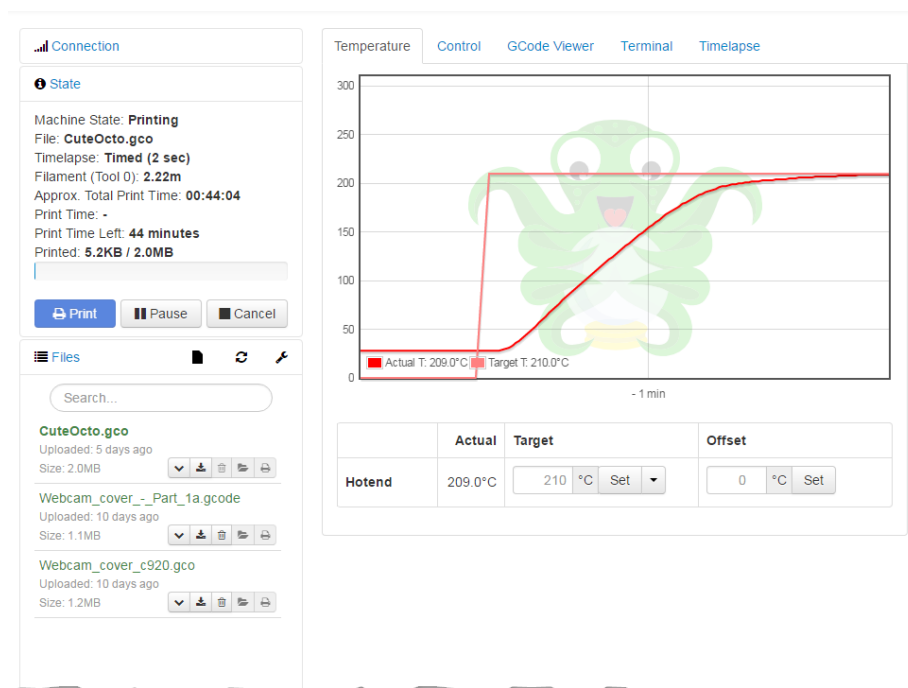


Figura 43 - Exemplo de um *software* de *Host* e de Controlo (impressão 3D) (Gujar et al. 2019)

3.2.3 Base de dados

Em computação, uma base de dados é uma coleção organizada de dados armazenados e acedidos eletronicamente a partir de um sistema de computador. Em contrapartida, o sistema de gestão da base de dados (*DBMS*) é o software que interage com os utilizadores, aplicações e a própria base de dados para capturar e analisar os dados (Hellerstein, Stonebraker, and Hamilton 2007).

Estes dois conceitos tratam-se igualmente dos componentes de um servidor de base de dados (Oluwatosin 2014), que, por sua vez, serve-se de uma aplicação de base de dados que fornece serviços (de base de dados) a outros programas ou computadores, conforme definido pelo modelo cliente-servidor (Berson 1996).

Maioria destas aplicações responde à linguagem query, sendo que a mais conhecida e utilizada é a Structured Query Language (SQL). Desta forma, os servidores de base de dados mais recorridos são os servidores SQL (Figura 44), os quais consistem num Database Engine, um serviço que corre em segundo plano, e num Management Studio, uma ferramenta gráfica para configurar e visualizar as informações na base de dados (Sumathi and Esakkirajan 2007).

```

dvdrental=# select title, release_year, length, replacement_cost from film
dvdrental=#   where length > 120 and replacement_cost > 29.50
dvdrental=#   order by title desc;

```

title	release_year	length	replacement_cost
West Lion	2006	159	29.99
Virgin Daisy	2006	179	29.99
Uncut Suicides	2006	172	29.99
Tracy Cider	2006	142	29.99
Song Hedwig	2006	165	29.99
Slacker Liaisons	2006	179	29.99
Sassy Packer	2006	154	29.99
River Outlaw	2006	149	29.99
Right Cranes	2006	153	29.99
Quest Mussolini	2006	177	29.99
Poseidon Forever	2006	159	29.99
Loathing Legally	2006	140	29.99
Lawless Vision	2006	181	29.99
Jingle Sagebrush	2006	124	29.99
Jericho Mulan	2006	171	29.99
Japanese Run	2006	135	29.99
Gilmore Boiled	2006	163	29.99
Floats Garden	2006	145	29.99
Fantasia Park	2006	131	29.99
Extraordinary Conquerer	2006	122	29.99
Everyone Craft	2006	163	29.99
Dirty Ace	2006	147	29.99
Clyde Theory	2006	139	29.99
Clockwork Paradise	2006	143	29.99
Ballroom Mockingbird	2006	173	29.99

(25 rows)

Figura 44 - Exemplo de uma base de dados SQL (Sulzbach 2018)

Neste sentido, o sistema defendido recorre a este tipo de recursos computacionais para que qualquer utilizador da comunidade possa ficar registado e os seus respetivos dados armazenados. Com o intuito de tornar o sistema inteligente, é igualmente criada e armazenada uma cópia de todo o código processado e gerado, na mesma biblioteca de informação. Isto permite que não seja necessário processar novamente toda a informação em caso de correspondência a um produto previamente executado pelo sistema.

3.2.4 Segurança, Documentação e Gestão

Por fim, dá-se foco àquilo que, metaforicamente, é o sangue do nosso organismo, nomeadamente a informação. Efetivamente, foi salientada inúmeras vezes a importância da informação ou, de uma forma geral, dos dados que correm no sistema, fazendo deles o bem mais valioso. Posto isto, a proposta oferece um sistema de informação que os preserva, isto é, garante que a informação (dados) esteja disponível, certa e legível apenas a quem lhe é de direito. Esta definição aplica-se indistintamente aos dados que residem num sistema bem como aos dados em trânsito, isto é, quando são enviados de um sistema para outro (Laudon 2003).

Para assegurar tal, o sistema de informação subdivide-se em Segurança e Gestão/Documentação. A primeira oferece comunicação segura, encriptação e *backup* de dados, uma vez que criptografa e autentifica os dados para garantir que não possam ser lidos ou interferidos durante a comunicação, o que é alcançado estabelecendo canais seguros, através da proteção pela chave de sessão (Luo et al. 2016). Nesta subcamada, é ainda viabilizada a recuperação de dados, por meio de um sistema de *backup*.

Por outro lado, a segunda apresenta agendamento dinâmico, autenticação e virtualização de dados, no sentido em que implementa os algoritmos de escalonamento dinâmico para a otimização do desempenho do sistema, onde as tarefas computacionais são agendadas entre diferentes recursos computacionais para alcançar eficiência de computação de alto desempenho e otimizar o equilíbrio de carga computacional dos recursos (computacionais). Além disso, executa a virtualização de *hardware* para os recursos computacionais, em que gera dinamicamente as máquinas virtualizadas com diferentes escalas e configurações, sustentando os ciclos de vida das mesmas e desempenhando tarefas de agendamento de baixo nível (Luo et al. 2016).

3.3 Sistema Multiagente

O sistema multiagente é um conjunto de vários tipos de agentes responsáveis pelo processamento, organização e distribuição da informação/dados relativa ao produto(s) a desenvolver, funcionando, então, como o cérebro do sistema global, assim como já foi realçado. De facto, este sistema gere todo o sistema de produção, desde qual produto deve ir para um determinado recurso a qual recurso deve estar a ser executado num determinado produto num momento específico, tendo como base o plano de produção do produto (Ramião et al. 2017).

Para que tal aptidão seja atingida, o sistema em foco virtualiza todos os elementos do sistema de produção, isto é, o produto a produzir dá origem a um Agente de Produto (PA), um recurso/ferramenta do sistema lança um Agente de Recursos (RA), além de outros agentes com funcionalidades distintas, sendo os dois destacados aqueles que apresentam maior relevância.

A Figura 45 esquematiza, então, a implementação da arquitetura proposta. Efetivamente, nela observa-se uma distribuição equilibrada quanto à disposição dos agentes, uma vez que aqueles que requerem um maior poder de processamento e armazenamento encontram-se alocados na nuvem. Melhor explicando, os dispositivos *IoT* alojam agentes do tipo RA, DA e CLA, enquanto que os restantes estão designados ao *cloud computing*. Salienta-se ainda a possibilidade de DA e CLA estarem também afetos à nuvem, visto que o primeiro funciona como “mensageiro” e o segundo pode surgir como líder de agentes do mesmo tipo, o que solicita um maior poder de computação.

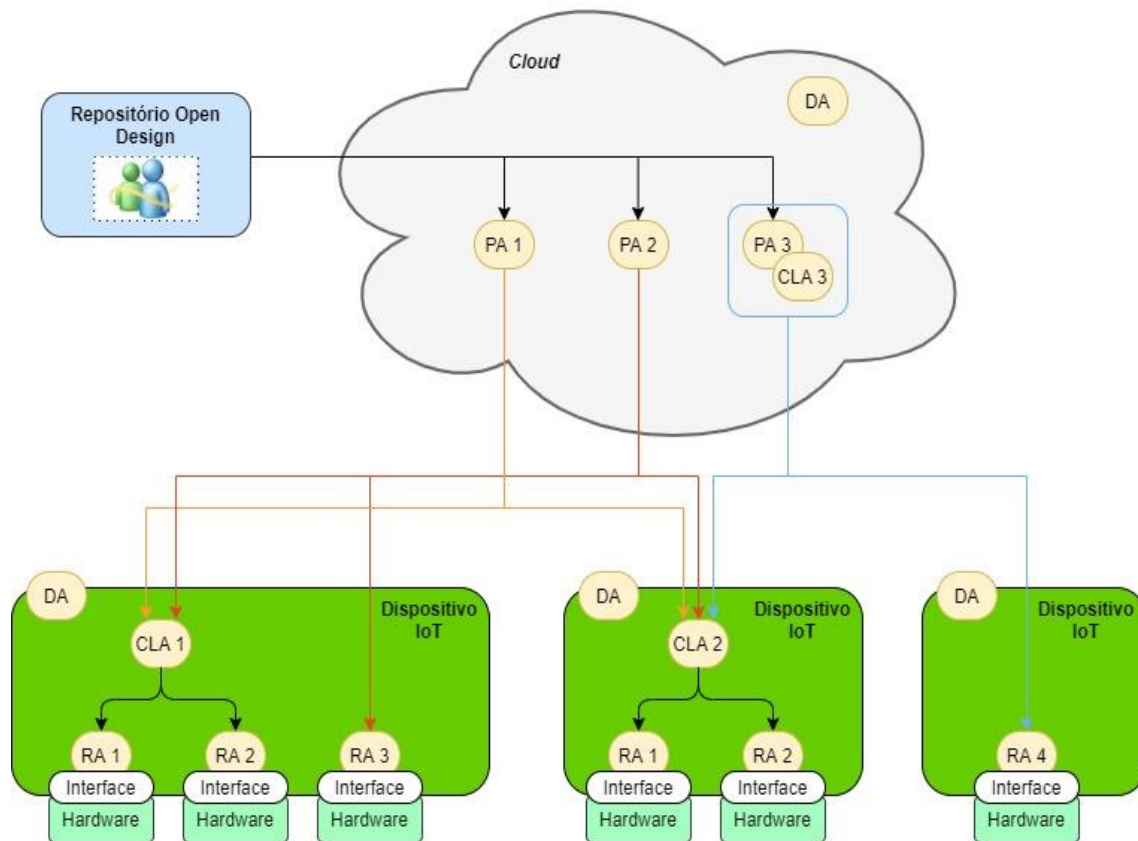


Figura 45 - Exemplo de uma integração do sistema multiagente na arquitetura proposta

Os diversos agentes podem assim ser representados por breves fluxogramas onde se pode observar o comportamento individual de cada um deles bem como a ligação existente entre os agentes (Ribeiro et al. 2013). As subsecções seguintes irão apresentar uma vista de alto nível dos principais agentes do sistema.

3.3.1 Product Agent

O Agente do Produto, ou PA, é uma representação virtual de um objeto a ser produzido e, dessa forma, representa o nível mais alto da abstração do sistema, visto que o mesmo funciona todo em prol da produção de um dado produto. Para que tal seja possível, cada PA tem a capacidade de recorrer ao plano de execução do produto e verificar se o sistema permite a sua realização. Se isso se verificar, o agente em foco está habilitado para controlar esse mesmo plano, bem como tomar decisões sobre o local onde cada etapa do processo é realizada, que, por sua vez, têm por base a ação coletiva existente entre si e os agentes do sistema de transporte (Ribeiro et al. 2013). Todos estes procedimentos estão ilustres no fluxograma presente na Figura 46.

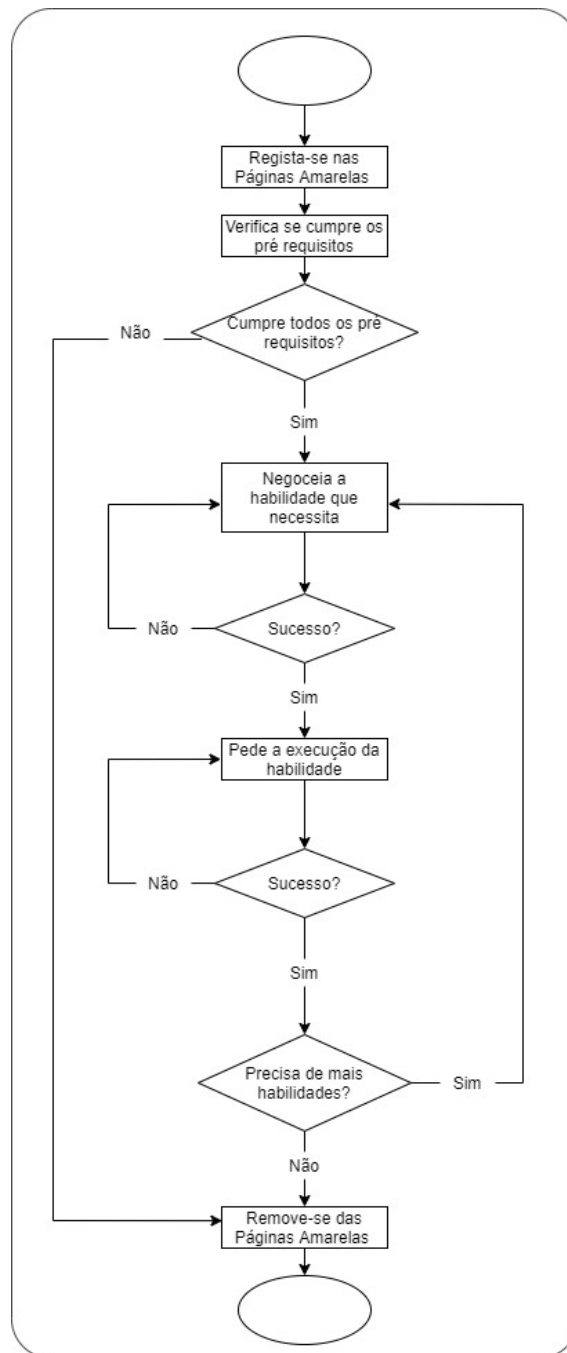


Figura 46 - Fluxograma do Agente Produto, adaptado de (Ramião et al. 2017)

Efetivamente, e analisando o fluxograma de forma mais detalhada, notamos que o agente que virtualiza o produto, ao ser criado, regista-se no serviço de páginas amarelas e certifica se, no sistema, existem recursos capazes de oferecer todas as habilidades necessárias para cumprir a execução do plano. Caso não existam, este remove-se das páginas amarelas e é terminado com a mensagem de impossibilidade de produção devido à falta de habilidades para a realização do plano. Por outro lado, se estas habilidades se encontrarem disponíveis, o agente em causa inicia o processo de produção.

O processo de produção consiste em negociar todas as habilidades necessárias, uma a uma, que constam no respetivo plano de produção. Isto significa que o agente do produto estuda qual recurso é o mais indicado para a produção da habilidade em questão e, em seguida, executá-la. A negociação da habilidade é um processo que passa por procurar nas páginas amarelas todos os RA e CLA que disponibilizem habilidade dada e verificar qual é o mais favorável para a execução da habilidade tendo em conta fatores como o estado do recurso, ou seja, quantos produtos já estão em lista de espera para aquele recurso e o valor associado à produção do mesmo (Ramião et al. 2017).

Como outro qualquer tipo de negociação, esta pode ser bem-sucedida ou não. Caso não o seja, o agente tentará novamente a fazer, no máximo de três tentativas. Se, dentro destes parâmetros, a negociação continuar a não ter êxito, o PA remove-se das páginas amarelas. Em caso contrário, o mesmo agente segue o seu processo para o estado de execução.

Na execução, o agente do produto pede ao agente associado ao recurso que ganhou a negociação para executar a habilidade em questão. À semelhança da negociação, o RA terá três tentativas para validar este processo, em que, mais uma vez, se tal não acontecer, o agente remove-se das páginas amarelas. De outro modo, caso seja sucedido o agente procede para a verificação do plano, ou seja, verifica se existem mais habilidades no plano de produção que ainda não foram negociadas e executadas e, caso existam, volta ao estado de negociação, repetindo o procedimento para as habilidades que faltam (Ramião et al. 2017).

Assim que todas as habilidades tenham sido efetuadas com sucesso, o agente do produto dá como fechado o processo de produção e remove-se das páginas amarelas.

3.3.2 Resource Agent

O Agente do Recurso, ou RA, é a entidade de menor abstração possível do sistema multiagente, em que o mesmo simula um recurso/ferramenta do sistema físico. Diante de tal, estes agentes interagem diretamente com os controladores da linha e têm como função principal traduzir habilidades para código nativo e garantir a sua execução e sincronização com a plataforma de agentes. Além disso, os RA's informam ainda o sistema de transporte sobre a localização física onde as suas habilidades podem ser executadas (Ribeiro et al. 2013). De forma a simplificar a informação acima, surge na Figura 47 o diagrama de atividades associado a este tipo de agente.

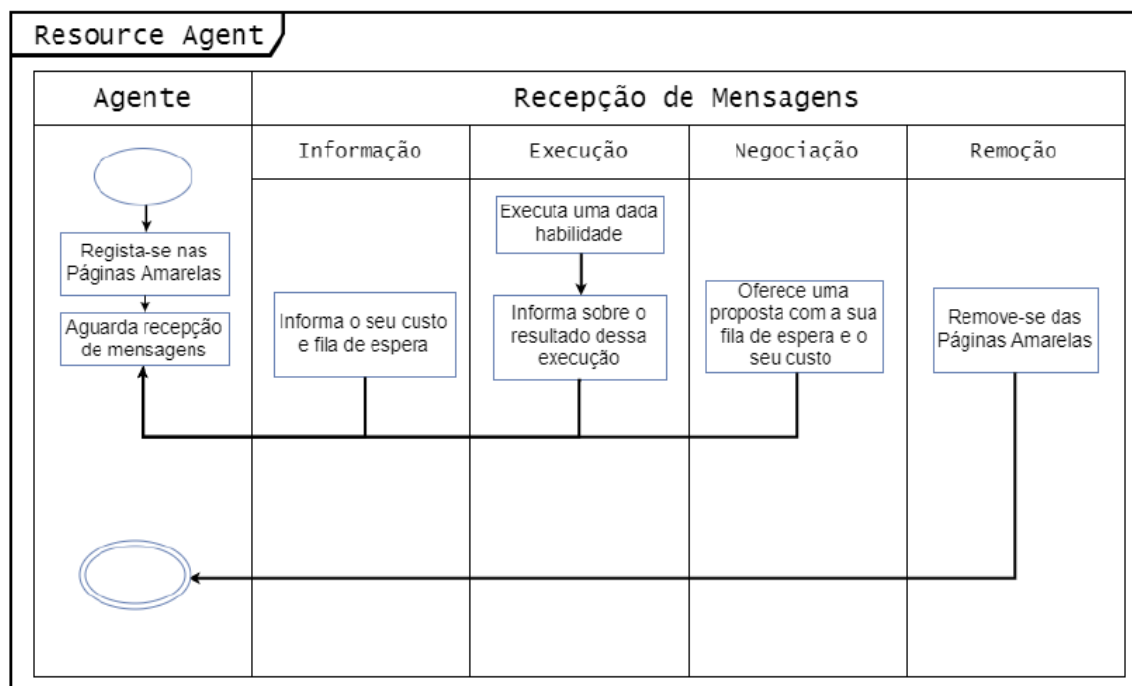


Figura 47 - Diagrama de atividades do Agente de Recurso (Ramião et al. 2017)

A primeira observação a tirar desse diagrama é que o agente, após originado e registrado no serviço de páginas amarelas, onde registra as habilidades capazes de realizar, apenas responde a mensagens (pedidos ou negociações) de outros agentes, nomeadamente do PA, CLA e DA. Perante uma comunicação de negociação o agente do recurso oferece uma proposta com o valor associado à execução de uma habilidade e o tamanho da sua lista de espera. Caso a sua oferta seja aceite, o agente que iniciou essa mesma negociação será colocado na lista de espera de execução (Ramião et al. 2017).

Ao receber uma comunicação de execução, ele verifica se o iniciador da comunicação é o próximo na sua lista e, caso o seja, essa execução é iniciada. Nesta situação, o RA comunica com o hardware para a execução da habilidade e aguarda a resposta com o resultado, que será posteriormente passada ao agente que iniciou a comunicação, seja uma mensagem de sucesso ou não. Caso não seja o próximo na lista, o agente em ênfase faz com que o iniciador fique à espera da sua vez (Ramião et al. 2017).

A comunicação de informação é semelhante à da negociação, no entanto, esta só informa o valor associado ao recurso e o tamanho da sua lista de espera. Por fim, o agente do produto remove-se das páginas amarelas, assim que receber uma comunicação de remoção, e conclui o seu ciclo.

3.3.3 Coalition Leader Agent

Os CLA's são utilizados para oferecer habilidades mais complexas recorrendo a uma agregação de habilidades de agentes já existentes. Para que tal seja possível, o agente em destaque está capacitado para a execução e coordenação de CLA e RA, não interferindo diretamente com qualquer controlador da linha. À semelhança do agente

de recurso, o CLA também informa o sistema de transporte sobre a localização física onde as habilidades estão disponíveis (Ribeiro and Barata 2013). Com o âmbito de melhor perceber o funcionamento destes agentes, apresentam-se as Figura 48 e Figura 49, que apontam as várias etapas do mesmo.

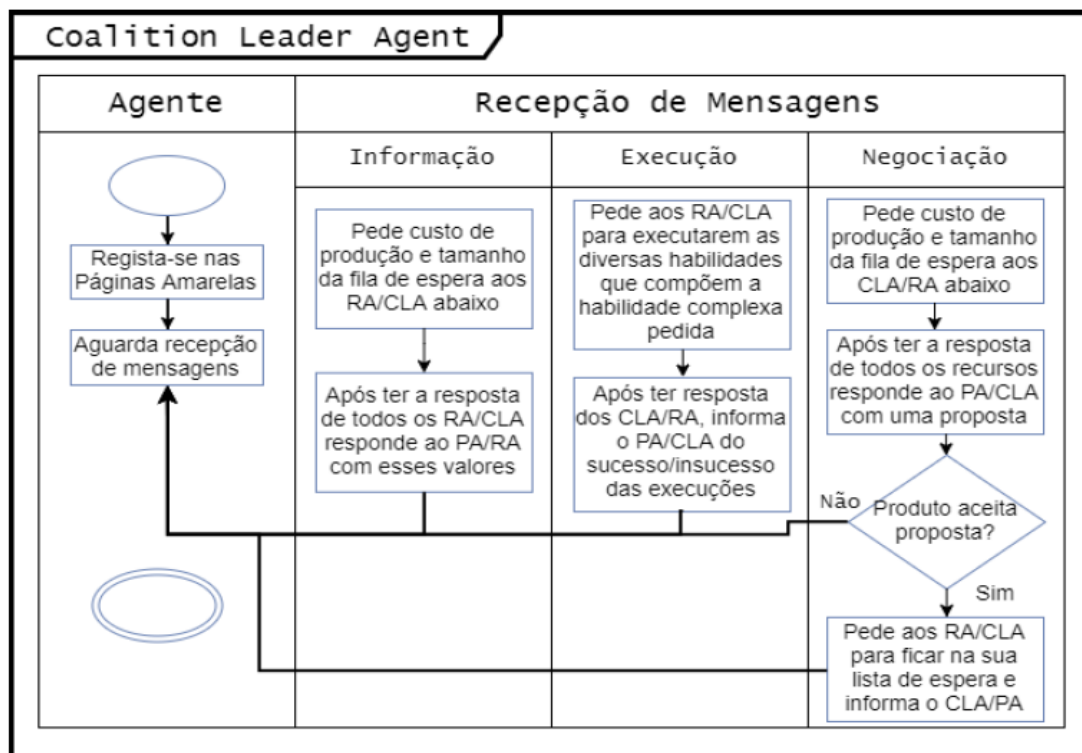


Figura 48 - Diagrama de atividade de um CLA (Parte 1) (Ramião et al. 2017)

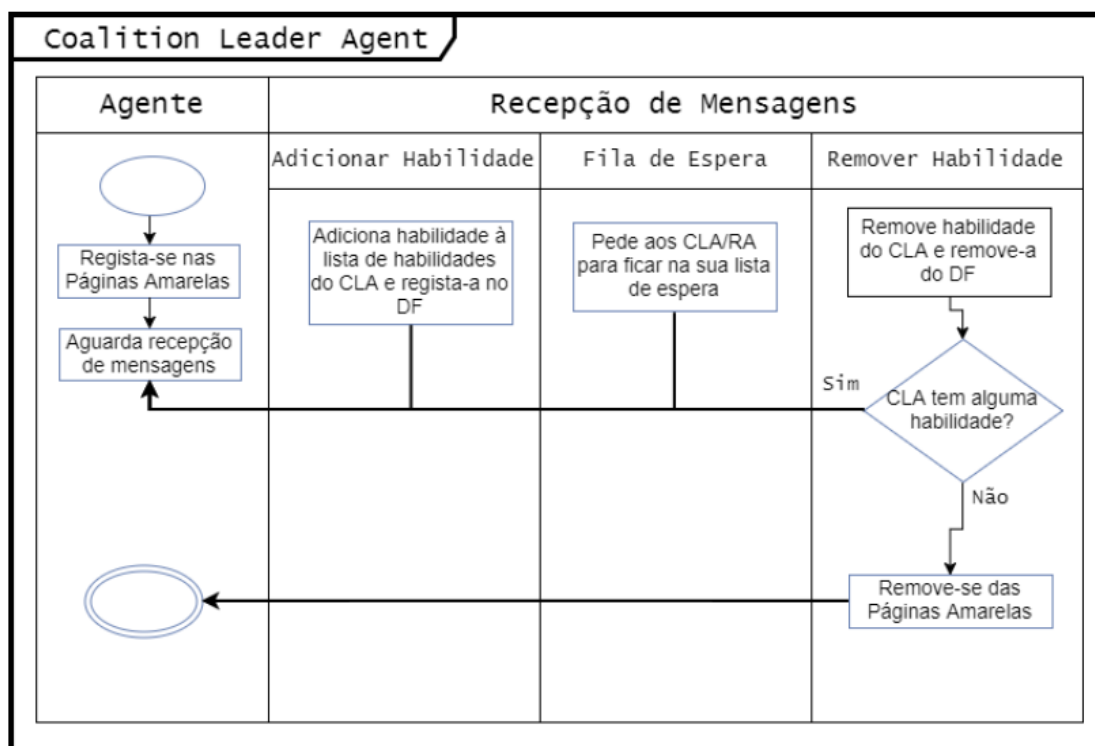


Figura 49 - Diagrama de atividade de um CLA (Parte 2) (Ramião et al. 2017)

O CLA, após ser iniciado, regista-se nas páginas amarelas e, em seguida, fica a aguardar quatro tipos de mensagem correspondentes a quatro ações distintas referentes à negociação, execução, adição e remoção de habilidades.

Neste cenário, a negociação é semelhante à do RA, contudo, o CLA ao receber uma negociação pedirá o valor de produção e o tamanho da fila de espera aos CLA e RA controlados pelo próprio. Deste jeito, será feita uma proposta à habilidade complexa pretendida, juntando, assim, os valores das diversas habilidades simples que a compõem, em que, caso seja aceite, o agente em questão informa os CLA e RA que rege, de modo a esses poderem colocá-lo na lista de espera (Ramião et al. 2017).

A execução de habilidade consiste em pedir a todos os RA e CLA, controlados pelo CLA “líder”, que executem uma determinada habilidade e, após receber a confirmação de todos, informa o produto do resultado. Além disso, sempre que o *coalition leader agent* receber uma mensagem de que uma habilidade foi adicionada, o mesmo regista esse serviço nas páginas amarelas. Em caso contrário, ou seja, de remoção de habilidade, o CLA remove a habilidade das páginas amarelas e, se não se verificar mais nenhuma, dá por terminado o seu ciclo.

3.3.4 Deployment Agent

Este tipo de agente tem como principal função servir de intermediário entre a plataforma agente e os controladores da linha. De facto, após iniciar, ele carrega a informação referente aos recursos e às habilidades complexas e, com base nessa informação, vai correndo um conjunto de condições definidas para um controlador específico, mais concretamente processos de execução e transporte (Ribeiro et al. 2013).

Desta forma, é possível lançar ou remover RA, lançar CLA e adicionar novas habilidades a um CLA ou até mesmo remover uma que deixou de ser exequível devido a algum recurso se ter desconectado, por exemplo. A Figura 50 mostra, de uma forma geral, o funcionamento do agente em foco, realçando igualmente que o mesmo só encerra os seus serviços quando o sistema for desligado.

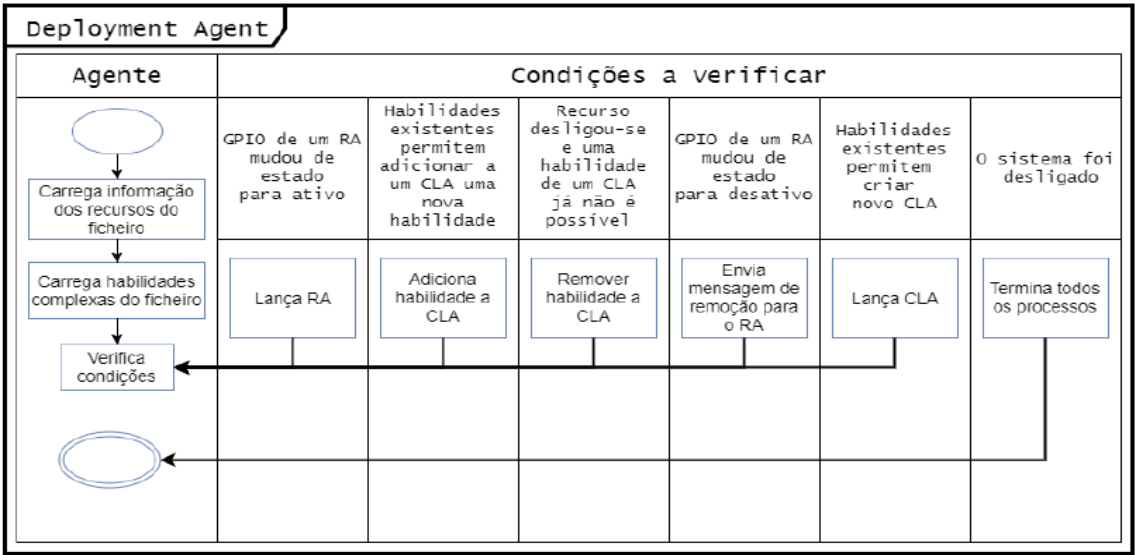


Figura 50 - Diagrama de atividades do DA (Ramião et al. 2017)

3.3.5 Comunicação entre Agentes

Como em todos os trabalhos em equipa, a comunicação entre todos os pertencentes é de enorme importância. O mesmo acontece na solução apresentada, uma vez que sem essa comunicação o sistema simplesmente não seria funcional. Posto isto, uma característica fundamental deste sistema multiagente é a viabilização dessa mesma comunicação entre os diversos agentes, destacando-se as dos PA, CLA E RA. Efetivamente, estas são a base do sistema permitindo, assim, que os diversos agentes comuniquem entre si para realizar as diversas operações necessárias.

A Tabela 6 mostra quais os agentes iniciantes e quais os participantes nas diversas comunicações entre os referidos.

Tabela 6 - Comunicação entre agentes, baseado em (Ramião et al. 2017)

Iniciantes	Participantes
PA	RA; CLA
RA	-----
CLA	RA; CLA

3.4 Sistema Físico

O objetivo principal passa por obter um sistema de produção altamente flexível e acessível a toda a comunidade, recorrendo, então, ao *Open Design*. Isto é, o utilizador consegue definir o quão automático o sistema é, sendo assim possível acompanhar a produção em tempo real e fazer ajustes/alterações assim que se pretender, sendo auxiliado por um sistema computacional à base de multiagentes. Aliás, a proposta é ainda mais aprofundada, no sentido em que se pretende dar uma utilidade mais definida e inovadora à arquitetura defendida. De facto, esta debruça-se na Manufatura Aditiva, mais conhecida por impressão 3D, com recurso a componentes físicos *low-cost*. Sem esquecer que é igualmente complementada pela manufatura mais usual, a subtrativa, para uma maior e melhor aptidão a uma vasta diversidade de produtos.

Posto isto, o sistema terá de ser capaz de ligar os dois mundos, o computacional e o físico, assim como está representado na Figura 51. Para que isso seja possível, o sistema físico apresenta quatro subsistemas interligados, nomeadamente o sistema de informação e controlo, o sistema de produção, o sistema de automação e o sistema de comunicação e vigilância, que, por sua vez, funcionam em conjunto para que se atinja essa finalidade. O referido encontra-se representado por camadas de forma a traduzir a relevância de cada um deles, tal como é visível na Figura 51.

Para um melhor entendimento de cada um destes subsistemas, os tópicos seguintes são acompanhados da descrição das suas respetivas funções, bem como a definição dos seus componentes e congruente funcionamento e orçamento.

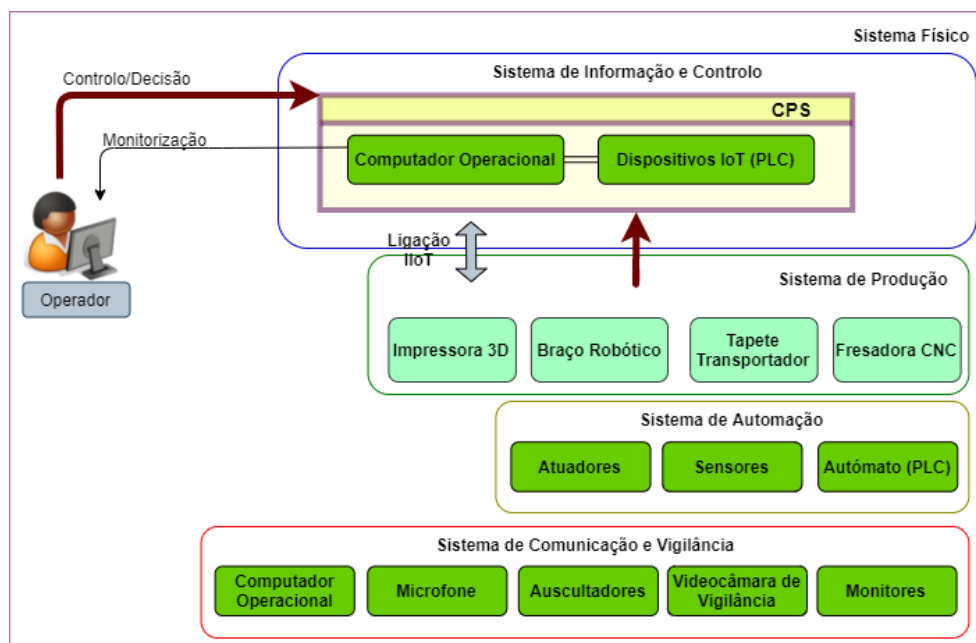


Figura 51 – Arquitetura do Sistema Físico

3.4.1 Sistema de Informação e Controlo

Na introdução deste capítulo referiu-se que a adoção de sistemas *IoT* e de computação em *Cloud* pode superar as ferramentas existentes de software assistido por computador ao lidar com a complexidade, dinâmica e incertezas nas aplicações empresariais. Tal afirmação deve-se, em parte, à informação derivada da, geralmente, elevada quantidade de recursos existentes num sistema de produção, que, por sua vez, será determinante nas decisões empresariais. Uma outra razão recai no facto de os novos produtos tenderem a ser mais inteligentes, versáteis e sofisticados, criando desafios nos processos de produção. Desta forma, o desempenho do sistema (lucro, tempo de execução, qualidade e custo) irá depender consideravelmente da eficácia do modelo de montagem (Wang, Bi, and Xu 2014).

Em termos práticos, e de modo a servir este tipo de produtos, é importante a existência dum sistema flexível e dinâmico capaz de comunicar e operar em conjunto, tal como acontece na arquitetura proposta. De facto, o *IoT* faz a ligação entre todos os recursos físicos, independentemente de como e onde possam estar distribuídos geograficamente, e possibilita a monitorização e a recolha de dados em tempo real dos mesmos, permitindo que, por exemplo, incertezas passem ser identificadas e usadas para suportar decisões de otimização. Além disso, a implementação da *IoT* no sistema terá um impacto significativo na disponibilidade de dados para o planeamento da fabricação do produto bem como no nível de dificuldade ao modelar um sistema de montagem (Ramião et al. 2017).

Efetivamente, foi estabelecido que um dispositivo *IoT* é aquele que apresenta melhores características para desempenhar as funções referentes ao subsistema em questão, sendo elas: a comunicação com um sistema computacional (I); a atuação no mundo físico (II); e a recolha de dados e controlo (III).

- I. Primeiramente, o dispositivo *IoT* liga-se, via *Wi-fi* ou *LAN*, à *cloud* do sistema, onde está alocada a maior parte do processamento da informação. Fruto dessa ligação, cria-se uma ponte que permite que toda essa informação circule entre o mundo *ciber* e o mundo físico. Além disso, o dispositivo terá de apresentar alguma capacidade de processamento e armazenamento, ou seja, de correr agentes (Figura 52), uma vez que têm autoridade para realizar algumas ações predispostas face alguns fatores. Essa capacidade está diretamente dependente do software adotado para a implementação dos agentes, o que terá de ser posto em consideração na escolha do dispositivo, juntamente com o custo associado a ambos.

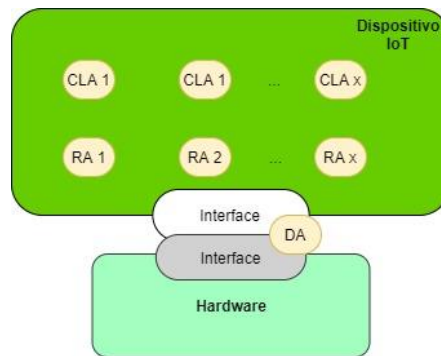


Figura 52 - Componente ciberfísico

- II. O fator mais importante na ponte de ligação talvez seja a capacidade do dispositivo de interagir diretamente com o *hardware*. De facto, este dispositivo é responsável por transmitir e coordenar toda a informação processada anteriormente para os recursos físicos, bem como receber mensagens dos mesmos no que toca à sua disponibilidade e existência e também das suas habilidades. Para que isto aconteça tecnicamente da melhor forma, o segredo está nas interfaces tanto do dispositivo como dos restantes recursos físicos, sendo elas genéricas, tal como está proeminente na Figura 51.
- III. Uma vez que o dispositivo se encontra conectado a todos os recursos físicos, é possível acompanhar em tempo real todas as ações e operações realizadas pelos mesmos. Isto permite, assim, ter um controlo e monitorização total de todo o processo envolvente à produção, bem como a recolha de dados referentes à mesma. No âmbito associado a este trabalho, isto é, à produção aberta, os dados reunidos poderão ser disponibilizados abertamente a toda a comunidade, o que facilita, por exemplo, a otimização do processo, ou então a replicação do mesmo.

No sistema proposto definiu-se que esta camada (Figura 53) é constituída por dois dispositivos *IoT*, mais propriamente, um (micro)computador operacional e um controlador lógico programável (*PLC*) ou um microcontrolador, visto que a junção das características e propriedades oferecidas por estes dispositivos garantem um melhor funcionamento do subsistema e, consequentemente, do sistema geral. Efetivamente, o primeiro assegura a ligação à *cloud* e recebe a informação proveniente da mesma, bem como possibilita a comunicação direta com a impressora 3D. Por outro lado, o segundo, além de conectado ao primeiro, trata-se de uma placa microcontroladora, em que a sua principal função é comunicar com os restantes elementos do sistema de produção e, claro está, controlá-los.

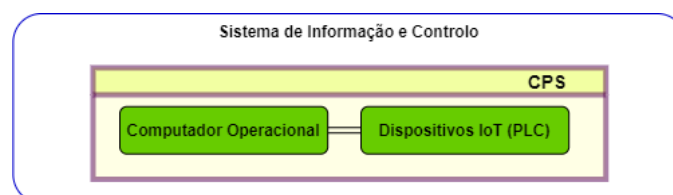


Figura 53 – Representação do Sistema de Informação e Controlo

Focando na ponte de ligação entre os dois “mundos”, esta é alcançada através de uma infraestrutura de interface de comunicação (sob protocolos de comunicação apropriados (Al-Sarawi et al. 2017): *BLE*, *NBIoT*, entre outros), a qual explora uma infraestrutura industrial local *IoT* (*IIoT*) para transmitir e coordenar toda a informação previamente processada para a camada de produção.

É neste contexto que o nível de computação cibernética, correspondente a um CPS⁰, é relevante, dando alguma capacidade cognitiva a todo o processo, dado que existem máquinas, pessoas e processos a cooperar e colaborar simbioticamente. De facto, um conjunto de processos recebe continuamente dados do processo da produção atual (de cada unidade de produção), da análise do operador e das métricas esperadas, e reage ao refinamento dos algoritmos envolvidos, ao realinhamento das máquinas envolvidas, bem como, se necessário, efetua o redesenho ou reengenharia dos produtos propostos. Além do mais, programação orientada a eventos para dinâmica síncrona, *Edge Computing* (Shi et al. 2016) em dispositivos *IoT* e *Message-Oriented Middleware* (Rausch 2018), para suporte de interoperabilidade, irão sustentar o comportamento distribuído e em tempo real necessário a tal nível de computação. Desta forma, a reconfiguração dinâmica e a integração entre o mundo físico e o computacional serão asseguradas.

3.4.2 Sistema de Produção

No capítulo 2 conclui-se que os sistemas de produção tradicionais são incapazes de, por si só, acompanhar a procura de produtos no mercado atual. Como tal, estes sistemas têm de ser adaptados ou substituídos por abordagens de arquitetura descentralizada e reconfigurável.

Assim sendo, propõe-se a divisão do sistema de produção em subsistemas com capacidade de interação entre si, isto é, módulos, tirando partido das diversas vantagens associadas ao conceito de modularidade. Efetivamente, a modularidade permite a diminuição do investimento e os tempos do mesmo, uma vez que os módulos podem ser reutilizados. Além disso, em situações de avaria de um sistema, os módulos avariados podem ser substituídos por suplentes, diminuindo assim os custos de reparo bem como o tempo de inatividade do sistema global (Lameche et al. 2017). Aliás, neste caso de estudo, esse tempo pode chegar a ser nulo, devido ao sistema ser inteligente o suficiente para procurar alternativas por via de outros módulos instalados.

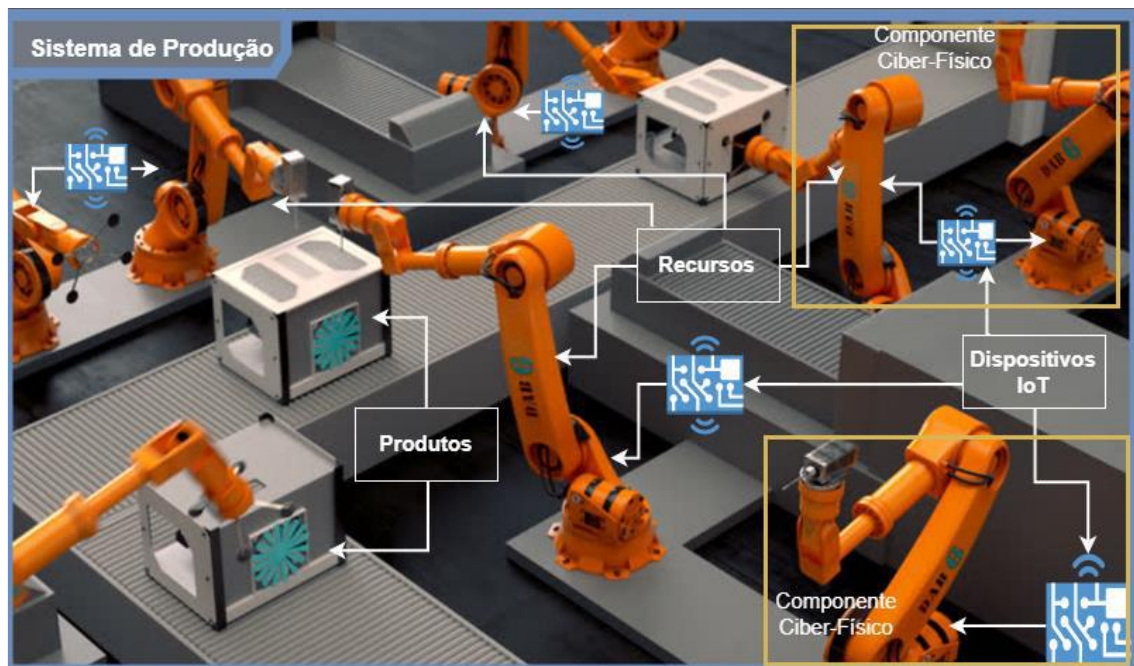


Figura 54 - Sistema de Produção, conceitual (Ramião et al. 2017)

Por conseguinte, considera-se um recurso ou ferramenta do sistema de produção como um módulo. No entanto, os componentes ciberfísicos também são considerados módulos, ou seja, um componente ciberfísico é um módulo composto por diversos módulos, nomeadamente o dispositivo IoT e os recursos/ferramentas ligadas a este, assim como ilustra a Figura 54.

A fim de um melhor entendimento do descrito anteriormente, segue-se o seguinte exemplo:

- Inicia-se o sistema numa nuvem (Figura 55).

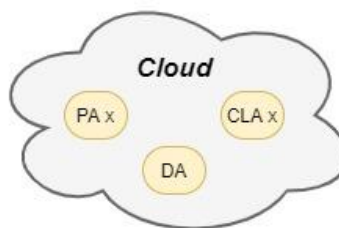


Figura 55 - Fase 1 do caso de exemplo

- Um dispositivo IoT liga-se, sendo automaticamente conectado ao sistema principal (Figura 56).

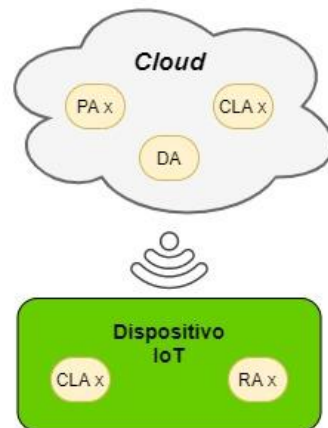


Figura 56 - Fase 2 do caso de exemplo

- O sistema reconhece que outro módulo se ligou a esse mesmo dispositivo, neste caso, um recurso físico (Figura 57). Assim, este recurso e as suas habilidades passam a ser identificadas pelo sistema através do dispositivo IoT.

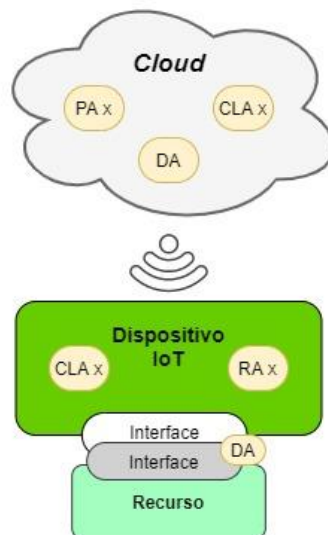


Figura 57 - Fase 3 do caso de exemplo

- Dois novos dispositivos IoT foram ligados, conectando-se, novamente, ao sistema. Um dos novos dispositivos verificou que não tinha nenhum módulo ligado a si enquanto que o outro reconheceu dois recursos e integrou-os no sistema (Figura 58), disponibilizando as suas habilidades no mesmo.

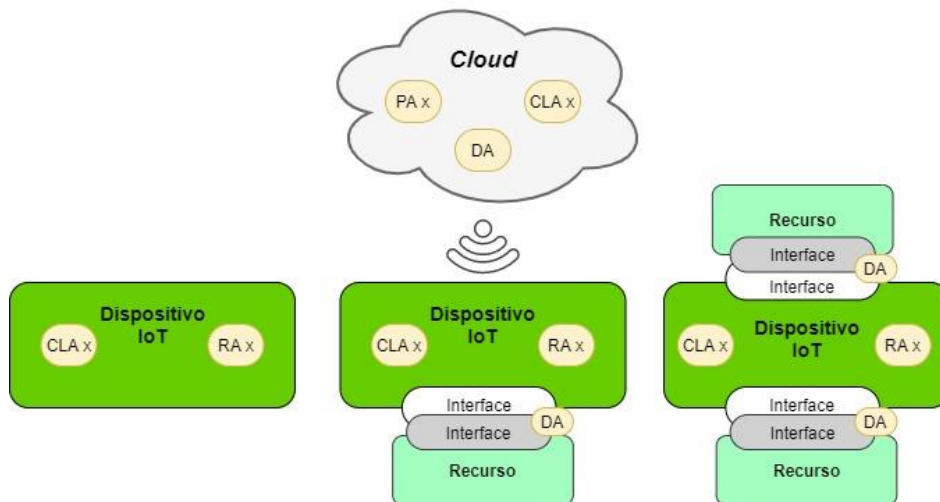


Figura 58 - Fase 4 do caso de exemplo

- Todavia, um dos dispositivos notou que um dos módulos que tinha ligado a si desligou-se e, como tal, desconectou-o do sistema (Figura 59). Em função disso, as habilidades disponibilizadas por esse módulo deixaram de estar disponíveis no sistema. Ainda assim, estas habilidades podem ser oferecidas por outro(s) módulos que estejam conectados e que, no seu conjunto, tenham as mesmas habilidades.

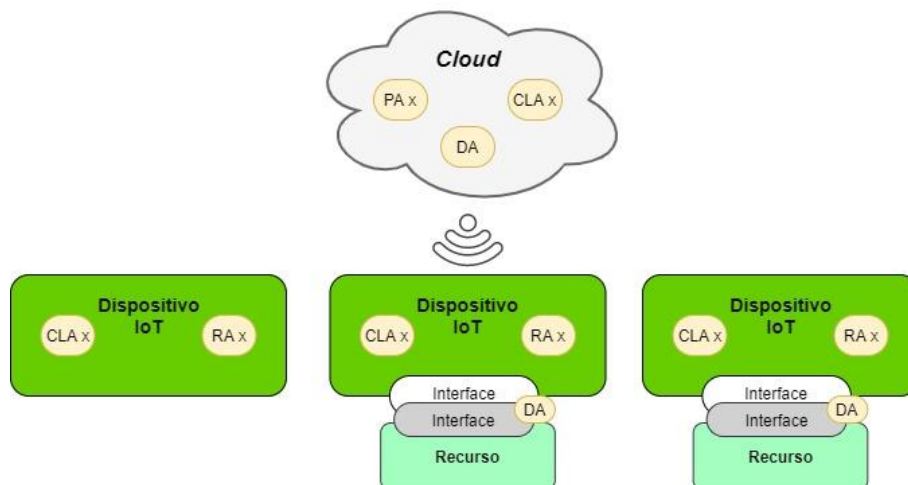


Figura 59 - Fase 5 do caso de exemplo

- Por último, um dos dispositivos IoT desliga-se e, consequentemente, o mesmo e todos os módulos ligados a ele deixam de fazer parte do sistema (Figura 60).

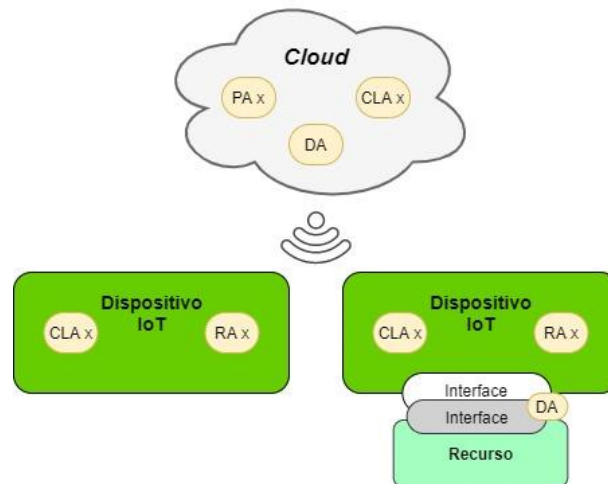


Figura 60 - Fase 6 do caso de exemplo

Em síntese, qualquer módulo composto por um dispositivo IoT existente no sistema é capaz de se conectar automaticamente ao mesmo, forçando-o a se reorganizar de acordo com as necessidades do produto a produzir e as funcionalidades que cada módulo oferece. Caso ocorra o contrário, isto é, algum módulo se desconecte do sistema, o procedimento é o mesmo, ou seja, este irá reorganizar-se consoante os módulos disponíveis. Destaca-se ainda que os recursos físicos são apenas reconhecidos no sistema através dos dispositivos IoT, que, por sua vez, usam interfaces para comunicar com tais recursos (Ramião et al. 2017).

Em contrapartida e de modo a simplificar tanto as representações como as definições, o sistema de produção reduz-se a apenas um módulo, o qual se apresenta como responsável pela fase final de todo o processo, isto é, pelo fabrico do produto. De facto, nele estão representados os dois tipos de manufatura anteriormente referidos, nomeadamente a aditiva, através da(s) impressora(s) 3D, e a subtrativa, por meio da(s) fresadora(s) CNC. Deste jeito, estamos, realmente, preparados e adaptados para uma maior diversidade de produtos, visto que o sistema está apto tanto para produtos personalizados e de geometrias complexas, como para a produção de peças mais rigorosas mecanicamente, dependendo da finalidade do produto em questão.

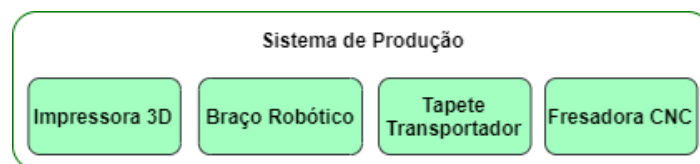


Figura 61 - Representação do Sistema de Produção

Além dos já mencionados, os recursos físicos necessários à produção em si (Figura 61) são igualmente um braço mecânico e um tapete de transporte, sendo todos eles na vertente *low-cost*. Neste caso é apenas utilizada uma impressora 3D e uma fresadora CNC por se tratar de um número não muito ambicioso, mas suficiente. No entanto ele pode variar, uma vez que o sistema está apto para tal.

3.4.3 Sistema de Automação

O sistema físico em análise pode ser considerado igualmente como um sistema mecatrónico, uma vez que a produção depende da integração de componentes mecânicos, eléctricos e electrónicos. Nesse sentido, os atuadores tornam-se elementos essenciais, mais precisamente, neste caso, os motores passo-a-passo (Figura 62). Efetivamente, este tipo de motores é amplamente implementado em sistemas que exigem alta precisão combinada com resposta rápida, onde, no começo, eram utilizados principalmente para tarefas simples de posicionamento “ponto a ponto”, nas quais eram controlados em circuito aberto (Elsodany, Rezeka, and Maharem 2011).

No entanto, atualmente, devido aos avanços feitos tanto na eletrónica de potência como no processamento de dados, os motores passo a passo são mais frequentemente controlados em circuito fechado e empregues em muitas aplicações para controlo de posição, em virtude da baixa inércia do rotor, fácil implementação e controlo e alta precisão de posicionamento (Kumar 2018).



Figura 62 - Motor passo-a-passo (Solectro 2020)

Para além dos atuadores, o tipo de sistema referido também recorre a sensores sob controlo de computador para realizar tarefas específicas no processo de produção, o que, em conjunto, consiste na automação programável. No que toca aos sensores, estes podem ser de contacto ou de não contacto. Os primeiros variam entre deteção por contato de força, binário e toque, e podem ser combinados de forma útil com deteção visual para muitas tarefas de manuseamento de materiais e montagem. Por outro lado, as principais categorias de sensores de não contacto são os sensores de proximidade, sensores eletro-óticos de imagem e sensores de alcance de imagem (Rosen and Nitzan 1977).

Desta última classe de sensores mencionada, realça-se os *micro-switches* (Figura 63), visto que, neste caso, irão delimitar os movimentos do braço robótico, que, por sua vez, é o único componente (físico) articulado. Para um melhor entendimento, um *micro-switch* típico é construído a partir de dois eléktodos condutores, um fixo e outro móvel, em que o eléktrodo móvel superior irá flexionar para baixo, para o eléktrodo base, devido à atração eletrostática após a aplicação de uma diferença de tensão entre os dois eléktodos (Lin and Zhao 2008). Quando o *micro-switch* fecha, dá-se um contacto entre ambos os eléktodos por meio de um par de saliências na superfície

inferior, originando, simultaneamente, um sinal elétrico (Majumder et al. 2001). Neste cenário, o sinal será reconhecido pelo sistema, o que significa que é estabelecido, então, o(s) limite(s) para os movimentos do braço robótico.



Figura 63 - Micro-switch (Distrelec 2021)

3.4.4 Sistema de Comunicação e Vigilância

O subsistema em pauta representa o aspeto humano do sistema ciberfísico proposto, dado que é assistido por um operador. Efetivamente, o mesmo está capacitado para comunicar com outros utilizadores e operadores, auxiliado por uns auscultadores e um microfone, tirando proveito do sistema comunicacional disponível na aplicação oferecida pela *cloud*.

Por outro lado, o sistema também oferece meios para a vigilância e monitorização a tempo real de todo o processo de produção, quer seja a nível visual, através de câmaras de vigilância, ou a nível computacional. De facto, toda essa informação é refletida nos monitores presentes no subsistema em ênfase e, desse modo, o operador poderá detetar possíveis anomalias e atuar perante elas.

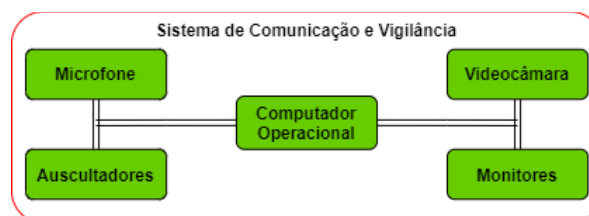


Figura 64 - Sistema de Comunicação e Vigilância

Por fim surge o computador operacional que viabiliza o devido desempenho das funções referentes aos componentes mencionados, uma vez que todos eles se encontram conectados ao primeiro (Figura 64), que, por sua vez, está ligado à *cloud* e ao outro dispositivo *IoT*, assim como já previamente mencionado.

3.4.5 Encadeamento dos subsistemas

De um ponto de vista mais realista e prático, surge a Figura 65 que demonstra o funcionamento do sistema físico. De facto, é possível observar que tanto o braço mecânico como a impressora 3D e a fresadora CNC estão conectados ao dispositivo *IoT*, desempenhando em conjunto todas as operações e movimentos necessários para

que se atinja o produto final, o qual será colocado no tapete transportador para que se dê o seu armazenamento. É de realçar que o exposto na Figura 65 trata-se de apenas um módulo pertencente ao sistema, uma vez que os recursos físicos representados estão a associados a um único dispositivo. Ou seja, um módulo corresponde a um dispositivo IoT e todos os componentes ciberfísicos (dispositivo + recurso físico) congruentes.

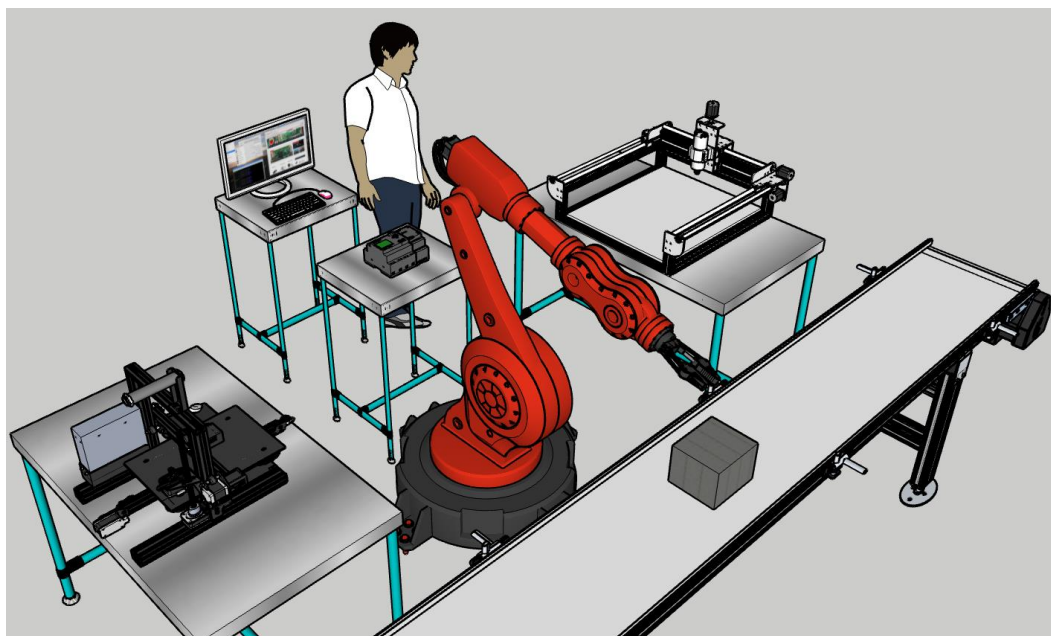


Figura 65 - Modelo representativo da implementação (física) pretendida

Em jeito de resumo, o fluxo de informações entre os vários componentes do sistema físico está representado na Figura 66.

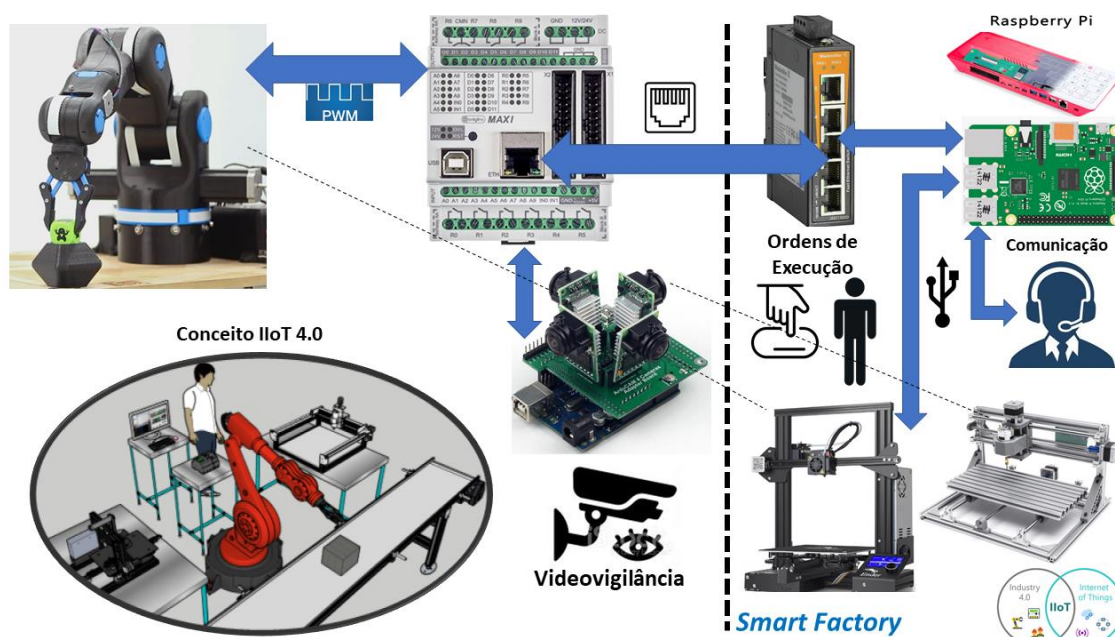


Figura 66 - Diagrama de blocos/fluxo corresponde ao Sistema Físico

4 Implementação da arquitetura proposta

De forma a transpor todos os conceitos abordados anteriormente para a componente prática, surge o presente capítulo, o qual apresenta uma implementação possível para arquitetura proposta.

Efetivamente, a parte cibernética, nucleada pelo sistema multiagente (MAS), foi desenvolvida em Java com o auxílio de uma plataforma de desenvolvimento e execução de MAS nesta tecnologia denominada *Java Agent Development Framework* (JADE). O recurso a este programa justifica-se pela sua flexibilidade, boa documentação e sua capacidade de atender aos padrões de comunicação entre os agentes, o que o torna também numa plataforma abundantemente utilizada em diversas aplicações. Além destas vantagens, Carrasco et al. (2014) enumera outras, mais concretamente:

- Fornece uma camada estável para desenvolver aplicativos distribuídos tornando a complexidade da camada inferior transparente para o programador;
- Facilita questões como coordenação de agentes, segurança, comunicação, mobilidade, redundância e outros serviços básicos em uma arquitetura distribuída;
- É *open-source*. Assim, muitas pessoas contribuem para desenvolvê-lo e mantê-lo;
- É compatível com comunicação *Foundation for Intelligent Physical Agents* (FIPA), como tal, permite comunicar com agentes desenvolvidos em outros ambientes que também implementem FIPA;
- A sua API é intuitiva, fácil de aprender e simples de usar, reduzindo assim o tempo de desenvolvimento.

No que toca à parte física, e com o âmbito de concretizar a arquitetura proposta, dando-lhe uma utilidade mais definida, esta debruça-se na Manufatura Aditiva, mais conhecida por impressão 3D, com recurso a componentes físicos *low-cost*. Desta forma, estamos perante um sistema que apoia a Indústria 4.0, uma vez que nele estão englobados conceitos como IoT (*Internet of Things*), *Industrial Internet of Things* (*IIoT 4.0*) (Boyes et al. 2018), *Cloud Manufacturing* (Xu 2012), *Smart Factory* (Chen et al. 2017) e a própria manufatura aditiva e respetivos recursos associados. Além de ajudar no desenvolvimento da indústria, a proposta apresentada direciona abertamente e facilita os seus serviços para uma toda comunidade envolvida no desenvolvimento e

produção de novos produtos e projetos, ideologia esta também denominada de *Open Design*.

De facto, ao utilizar o nosso sistema ciberfísico, a comunidade em questão é capaz de produzir rapidamente, e a um baixo custo unitário, produtos personalizados ao gosto e necessidades próprias, podendo estes ser de geometrias mais complexas, uma vez que o tipo de manufatura realçado permite tal. Uma outra grande vantagem é sua a sustentabilidade, dado que usa menos material e gera menos resíduos de produção e consome pouca energia elétrica (Wang et al. 2019).

No entanto, e de forma a mostrar a diversidade, flexibilidade e amplitude do sistema proposto, o mesmo é complementado com a manufatura mais convencional, ou seja, a subtrativa. Com isto, o sistema está, de facto, preparado para uma maior diversidade de produtos, visto que também está apto para a produção de peças mais rigorosas mecanicamente, dependendo da finalidade do produto em questão.

Tendo em conta o referido anteriormente, o propósito paralelo passa por implementar este sistema na comunidade universitária, inicialmente o Instituto Superior de Engenharia do Porto, para que os seus alunos e professores possam dar asas à sua imaginação e, possivelmente, trabalhar em conjunto para o desenvolvimento de produtos inovadores, ou simplesmente para desejo pessoal.

Além disso, outro aspeto bastante positivo é a sustentabilidade que o sistema fornece, isto é, o mesmo poderá ser também usado para consumo próprio. Efetivamente, o instituto ao invés de recorrer a meios externos para a compra de peças/material para os seus laboratórios e afins, produzem-nas internamente a um custo muito reduzido. Neste caso em concreto, é igualmente possível que outros institutos/faculdades recorram ao ISEP para essa mesma produção, o que traria um lucro para o último.

Nesse sentido, é, então, descrita a implementação da arquitetura proposta, a qual se encontra em concordância com a respetiva ordem de implementação dos sistemas associados.

4.1 Sistema Físico

4.1.1 Sistema de Informação e Controlo

4.1.1.1 Lista de Componentes e Orçamento Primordial

No que toca aos componentes deste subsistema, estipulou-se que o mesmo seria constituído por dois dispositivos *IoT*, tal como referido. Numa primeira fase efetuou-se um trabalho de pesquisa, a nível industrial, relativamente às dimensões e preços do(s) componente(s) de forma a obter valores próximos do pretendido, servindo também como uma forte base para a continuação da elaboração e estabelecimento da implementação (desenhos, modelos, representações, orçamentos, etc.). Posto isto, o mencionado anteriormente encontra-se exposto nas Tabela 7 e Tabela 8.

Tabela 7 - Lista de componentes do Sistema de Informação e Controlo

Lista de Componentes		
Componentes	Quantidade	Dimensões aproximadas (mm)
(Micro)Computador Operacional	1	64x56x12
PLC	1	90x60x20

Tabela 8 - Orçamento primordial do Sistema de Informação e Controlo

Orçamento primordial	
Componentes	Preço estimado (€)
(Micro)Computador Operacional	130
Dispositivo IoT	400
Total	400

4.1.1.2 Propostas de componentes

Salienta-se que a abordagem às propostas de componentes debruça-se na plataforma *open-source Arduino*, o que se justifica pelo imenso feedback por parte da comunidade, na medida em que tem sido o cérebro de milhares de projetos, desde objetos do quotidiano a complexos instrumentos científicos (Arduino.cc 2015).

Além disso, trata-se de (Arduino.cc 2015):

- I. uma plataforma relativamente barata em comparação com outras plataformas de microcontrolador;
- II. uma *cross-platform*, uma vez que o *Arduino Software* (IDE) é executado nos sistemas operacionais Windows, Macintosh OSX e Linux, quando a maioria dos sistemas de microcontroladores está limitada ao Windows;
- III. um ambiente de programação simples e claro. De facto, é fácil de usar para iniciantes, dada a sua vasta biblioteca de código de programação disponível abertamente, mas flexível o suficiente para usuários avançados aproveitarem;
- IV. um software de código aberto e extensível. Efetivamente, o software *Arduino* é publicado como ferramentas de código aberto, disponíveis para extensão por programadores experientes.

4.1.1.2.1 Arduino Nano

O *Arduino Nano* (Figura 67) é uma placa pequena, completa e compatível com a placa de ensaio, baseada no *ATmega328* (*Arduino Nano 3.x*). Possui mais ou menos a mesma funcionalidade do *Arduino Duemilanove*, mas num pacote diferente. Efetivamente, o mesmo apenas não possui um conector de alimentação DC e funciona com um cabo USB Mini-B em vez do padrão (Arduino 2019).

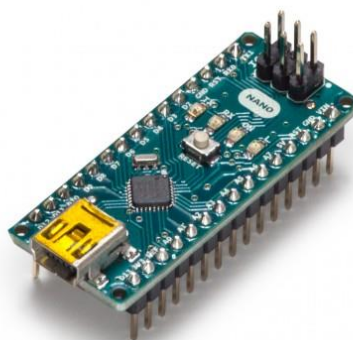


Figura 67 - Arduino Nano (Arduino LLC et al. 2012)

As especificações técnicas associadas estão expostas na Tabela 9.

Tabela 9 – Especificações técnicas do Arduino Nano, adaptado de (Arduino LLC et al. 2012)

Arduino Nano	
Microcontrolador	ATmega328
Arquitetura	AVR
Tensão Operacional	5 V
Memória Flash	32 kB, dos quais 2 kB são usados pelo <i>bootloader</i>
SRAM	2 kB
Velocidade do Relógio	16 MHz
Pinos Analógicos <i>IN</i>	8
EEPROM	1 kB
Corrente DC por pino I/O	40 mA (Pino I/O)
Tensão de entrada	7-12V
Pinos I/O digitais	22 (dos quais 6 são PWM)
Saídas PWM	6
Consumo de energia	19 mA
Tamanho PCB	18 x 45 mm
Peso	7 g
Preço PVP	20,00€
Preço Revenda	3,00€

4.1.1.2.2 Arduino Uno Rev3

Arduino Uno (Figura 68) é uma placa microcontrolada baseada no *ATmega328P*. Possui 14 pinos de entrada / saída digital (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um ressonador de cerâmica de 16 MHz, uma conexão USB, um conector de força, um conector ICSP e um botão de *reset*, contendo, dessa forma, tudo o que é fundamental para dar suporte ao microcontrolador (Farnell 2013).

A placa *Uno* e a versão 1.0 do *Arduino Software* (IDE) foram as versões de referência do *Arduino*, agora evoluídas para versões mais recentes. Além disso, a placa em foco é

a primeira de uma série de placas *Arduino USB* e o modelo de referência para a plataforma *Arduino* (Farnell 2013).



Figura 68 – Arduino Uno Rev3 (Farnell 2013)

A Tabela 10 mostra as especificações técnicas correspondentes a esta placa microcontroladora.

Tabela 10 – Especificações técnicas do Arduino Uno Rev3, adaptado de (Farnell 2013)

Arduino Uno Rev3	
Microcontrolador	ATmega328P
Tensão Operacional	5 V
Memória Flash	32 kB, dos quais 0,5 kB são usados pelo <i>bootloader</i>
SRAM	2 kB (ATmega328P)
Velocidade do Relógio	16 MHz
Pinos de entrada Analógicos	6
EEPROM	1 kB (ATmega328P)
Corrente DC por pino I/O	20 mA (Pino I/O)
Tensão de entrada	7-12V
Pinos I/O digitais	14 (dos quais 6 são PWM)
Pinos I/O PWM	6
LED_BUILTIN	13
Comprimento	68,6 mm
Largura	53,4 mm
Peso	25 g
Preço PVP	20,00€
Preço Revenda	5,00€

4.1.1.2.3 *Arduino MEGA 2560 Rev3*

O *Arduino MEGA 2560 Rev3* (Figura 69) é uma placa microcontroladora baseada no *ATmega328*, à semelhança da proposta anterior. Aliás, esta versão trata-se, basicamente, de uma extensão da mesma, uma vez que varia na quantidade de pinos

que possui, na capacidade de memória e no seu tamanho. Salienta-se que a placa *Mega 2560* é compatível com a maioria das blindagens projetadas para o *Uno* e as placas anteriores *Duemilanove* ou *Diecimila* (Mellis 2011).



Figura 69 - Arduino MEGA 2560 Rev3 (Mellis 2011)

As suas especificações técnicas encontram-se na Tabela 11.

Tabela 11 - Especificações técnicas de Arduino MEGA 2560 Rev3, adaptado de (Mellis 2011)

Arduino MEGA 2560 Rev3	
Microcontrolador	ATmega2560
Tensão Operacional	5 V
Memória Flash	256 kB, dos quais 8 kB são usados pelo <i>bootloader</i>
SRAM	8 kB
Velocidade do Relógio	16 MHz
Pinos de entrada Analógicos	16
EEPROM	4 kB (ATmega328P)
Corrente DC por pino I/O	20 mA (Pino I/O)
Tensão de entrada	7-12V
Pinos I/O digitais	54 (dos quais 15 são PWM)
Pinos I/O PWM	15
LED_BUILTIN	13
Comprimento	101,52 mm
Largura	53,3 mm
Peso	37 g
Preço PVP	35,00€
Preço Revenda	10,00€

4.1.1.2.4 Arduino Uno WiFi Rev2

O *Arduino UNO WiFi Rev.2* (Figura 70) é o ponto de entrada mais fácil para a IoT básica, apresentando o formato padrão da família *UNO*. A placa em ênfase é funcionalmente igual ao *Arduino Uno Rev3*, mas com a adição de WiFi/Bluetooth e alguns outros aprimoramentos. De facto, incorpora o novo microcontrolador *ATmega4809* de 8 bits

da Microchip e possui uma IMU (Unidade de Medição Inercial) integrada. Além do mais, o módulo Wi-Fi é um SoC independente com pilha de protocolo TCP/IP integrada que pode fornecer acesso a uma rede Wi-Fi ou atuar como um ponto de acesso (Wifi n.d.).

Esta placa de desenvolvimento da *Arduino* contém 14 pinos de entrada/saída digital, dos quais 5 podem ser usados como saídas PWM; 6 entradas analógicas; uma conexão USB; um conector de alimentação; um conector ICSP e um botão de reinicialização, assim como está detalhado na Tabela 12. Desta forma, contém tudo o que é fundamental para dar suporte ao microcontrolador (Wifi n.d.).



Figura 70 - Arduino Uno WiFi Rev2(Wifi n.d.)

Tabela 12 - Especificações técnicas do Arduino Uno WiFi Rev2, adaptado de (Wifi n.d.)

Arduino Uno WiFi Rev2	
Microcontrolador	ATmega4809
Tensão Operacional	5 V
Memória Flash	48 kB (ATmega4809)
SRAM	6,144 Bytes (ATmega4809)
Velocidade do Relógio	16 MHz
Pinos de entrada Analógicos	6
EEPROM	256 Bytes (ATmega4809)
Corrente DC por pino I/O	20 mA (Pino I/O)
Tensão de entrada	7-12V
Pinos I/O digitais	14 (dos quais 5 são PWM)
Pinos I/O PWM	15
LED_BUILTIN	25
Módulo de Rádio	Incluído
Elemento de Segurança	Incluído
Unidade de Medição Inercial	Incluído
Comprimento	68,6mm
Largura	53,4 mm
Peso	25 g
Preço PVP	38,90€

Preço Revenda	50,00€
----------------------	--------

4.1.1.2.5 ESP8266

O módulo em destaque é mais adequado para *Internet of Things* (IoT) visto o seu baixo custo; capacidade de baixo consumo de energia, uma vez que requer apenas 3,3 V; módulo WiFi integrado; pilha de protocolo TCP/IP integrada; *firmware* fácil de atualizar e apagar e é alimentado por USB (Kodali and Mahesh 2016).

Com os recursos de rede Wi-Fi completos e independentes, o *ESP8266EX* pode funcionar como um aplicativo independente ou como escravo de um *MCU host*. Além das funcionalidades de Wi-Fi, o *ESP8266EX* também integra uma versão aprimorada do processador de 32 bits da série *L106 Diamond da Tensilica* e SRAM no chip. A placa pode ser conectada a sensores externos e outros dispositivos por meio dos GPIOs (Espressif Systems 2020). Equipada com memória flash SPI externa de 16 MB, o *ESP8266* está disponível para armazenamento de programas e *firmware* (Firebeetle et al. n.d.).

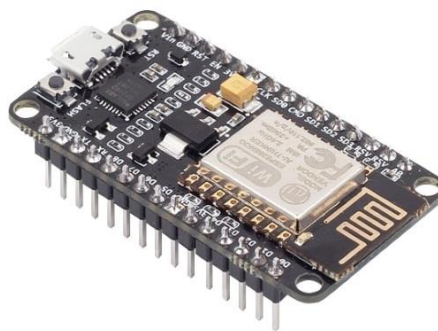


Figura 71 - ESP8266 (Firebeetle et al. n.d.)

Outro importante fator é a compatibilidade com a programação do *Arduino*. Isto permite que o *Firebeetle ESP8266* reduza a barreira da programação, dado que o operador pode implementar códigos de programação do mesmo tipo diretamente no *ESP8266* para reduzir a dificuldade de operação e aumentar a estabilidade da placa (Firebeetle et al. n.d.).

Todas as especificações técnicas enumeradas estão realçadas na Tabela 13

Tabela 13 - Especificações técnicas do ESP8266, adaptado de (Espressif Systems 2020; Firebeetle et al. n.d.; Kodali and Mahesh 2016)

ESP8266	
Microcontrolador	Tensilica L106 32-bit
Tensão Operacional	2,5 V ~ 3,6 V
Memória Flash Externa	16 MB (4 MB já incluídos)
SRAM	50 kB
Velocidade do Relógio	80 MHz (máx 160 MHz)
Pinos de entrada Analógicos	1

UART's	1
Corrente DC por pino I/O	80 mA (Pino I/O)
Tensão de entrada	7-12V
Pinos I/O digitais	10
WiFi	IEEE802.11 b/g/n (2.4 GHz~2.5 GHz), 5GHz WiFi não suportado
SPI Interface	Incluído
I²C Interface	Incluído
IR Interface	Incluído
Programação	Compatível com <i>Arduino</i>
Comprimento	58 mm
Largura	29 mm
Peso	24 g
Preço PVP	-
Preço Revenda	~5€

4.1.1.2.6 Controllino

O *Controllino* é um controlador lógico programável (PLC) que possui a facilidade da sua linguagem ser aberta e compatível com outros dispositivos. De facto, este dispositivo é um pequeno computador (microcontrolador) com periféricos baseados na plataforma *Open Source Arduino*, o que o torna ideal para aplicações nas áreas de automação, monitorização e controlo (Abdul Aziz Ahmed Siyad 2019).

A nível industrial, e devido à necessidade crescente do mercado, idealizou-se uma solução flexível para a construção e automação da indústria. A mesma concentra-se em módulos de controlo completo com adaptabilidade, escalabilidade e disposição a qualquer momento (Abdul Aziz Ahmed Siyad 2019).



Figura 72 – *Controllino Maxi* (Abdul Aziz Ahmed Siyad 2019)



Figura 73 - *Controllino Mega* (Controllino 2017)

Por outras palavras, pode-se afirmar que o *Controllino* se trata de um *Arduino* industrializado. Efetivamente, ele combina a flexibilidade do ecossistema *Arduino* e a natureza de código aberto com a segurança, robustez e confiabilidade do PLC de nível industrial. Além disso, foi desenvolvido com os mais altos padrões da indústria e a segurança eletrónica em mente, o que o torna ideal não apenas para cenários de prototipagem, mas também para produtos finais (Abdul Aziz Ahmed Siyad 2019).

Perante isso, são propostos dois modelos de *Controllino*: o *Maxi* (Figura 72) e o *Mega* (Figura 73). Quanto às especificações técnicas, as relativas ao primeiro são indicadas na Tabela 14 e as do segundo na Tabela 15.

Tabela 14 - Especificações técnicas do *Controllino Maxi*, adaptado de (Degree 2017)

<i>Controllino Maxi</i>	
Microcontrolador	ATMega2560
Memória Flash Externa	248kB
SRAM	8 kB
Velocidade do Relógio	16 MHz
Entradas analógicas/digitais	10
Entradas digitais	2
Saídas digitais	12 (Todas PWM)
Saídas analógicas	0
Saídas Relés	10
Tensão de entrada	12-24V
USB	USB-B, 2.0
Ethernet	RJ45, 10/100Mbps
RTC	1
SPI Interface	1
I²C Interface	1
Serial TTL Interface	2
RS485 Interface	1
Programação	Compatível com <i>Arduino</i>
Dimensões (C x L x A)	72x90x62mm
Peso	250 g
Preço	209€

Tabela 15 - Especificações técnicas do *Controllino Mega*, adaptado de (Controllino 2017)

<i>Controllino Mega</i>	
Microcontrolador	ATMega2560
Memória Flash Externa	248kB
SRAM	8 kB
Velocidade do Relógio	16 MHz

Entradas analógicas/digitais	16
Entradas digitais	5
Saídas digitais	24 (15 PWM)
Saídas analógicas	0
Saídas Relés	16
Tensão de entrada	12-24V
USB	USB-B, 2.0
Ethernet	RJ45, 10/100Mbps
RTC	1
SPI Interface	1
I²C Interface	1
Serial TTL Interface	2
RS485 Interface	1
Programação	Compatível com <i>Arduino</i>
Dimensões (C x L x A)	107x90x62mm
Peso	250 g
Preço	279€

4.1.1.3 Análise, Escolha e Orçamento Final dos Componentes

Atendendo à relação entre custo e aptidão técnica das várias propostas de microcontroladores, o *Arduino MEGA 2560 Rev3* é aquela que a satisfaz melhor. No entanto, e visto que se pretende obter um laboratório que sirva igualmente como um protótipo a nível industrial, a solução recai nos *Controllino*, uma vez que, lá está, tratam-se de microcontroladores industrializados, oferecendo a mesma segurança, robustez e confiabilidade do *PLC* de nível industrial. Das duas opções realçadas, o *Controllino Mega* surge como solução final devido à maior aptidão técnica relativamente à outra alternativa.

Embora também pertença ao presente sistema, o computador operacional será analisado somente no tópico 4.1.4, que diz respeito ao sistema de comunicação e vigilância.

Mantendo a mesma lógica da análise, o orçamento é apenas relativo ao microcontrolador, tendo, então, o valor de 279€.

4.1.2 Sistema de Produção

4.1.2.1 Lista de Componentes e Orçamento Primordial

Quanto à disposição dos componentes, é proposta a visível na Figura 74, acompanhada da lista de componentes associadas à mesma, presente na Tabela 16. Tanto o modelo como as dimensões dos componentes podem não corresponder à realidade, mas, no entanto, tratam-se de aspetos muito próximos de tal, servindo,

assim, como uma forte base para a continuação da elaboração e estabelecimento da implementação. Salienta-se ainda a centralização do braço robótico, dado que este é o responsável tanto por recolher o material de fabrico (do tapete de transporte) para a fresadora CNC, como transferir o produto no seu estado final para o tapete de transporte.

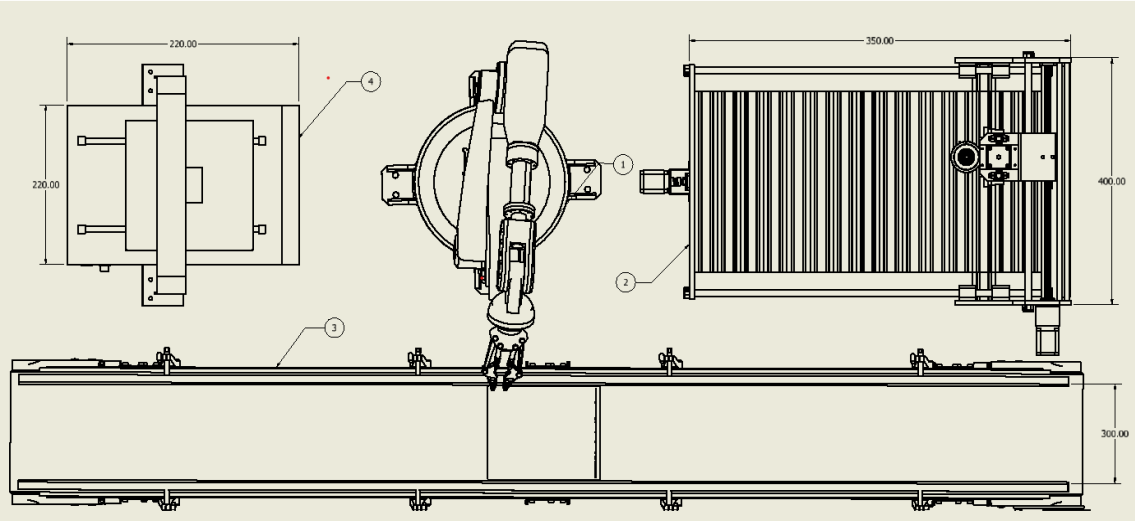


Figura 74 - Layout da disposição do Sistema de Produção

Tabela 16 - Lista de componentes do Sistema de Produção

Lista de Componentes			
Balão	Componentes	Quantidade	Dimensões aproximadas (mm)
1	Braço Robótico	1	700 (Alcance do braço)
2	Fresa CNC	1	400x350x250
3	Tapete Transportador	1	1500x350
4	Impressora 3D	1	220x220x250

À semelhança do subsistema anterior, realizou-se uma pesquisa, novamente a nível industrial, com o intuito de obter valores próximos do presumido no que diz respeito às dimensões e preços dos componentes. De facto, os mesmos estão patentes nas Tabela 16 e Tabela 17, servindo, mais uma vez, de sustentação para o seguimento da implementação.

Tabela 17 - Orçamento primordial do Sistema de Produção

Orçamento primordial	
Componentes	Preço estimado (€)
Braço Robótico	6800
Fresa CNC	250
Tapete Transportador	300
Impressora 3D	150
Total	7500

4.1.2.2 Propostas de componentes

4.1.2.2.1 Braço Robótico - Robolink® RL-DP - módulo completo de 5 graus de liberdade

Com os braços robóticos *Robolink*, os processos monótonos e perigosos podem ser facilmente automatizados sem custos adicionais, dispondo, então, de braços articulados standard completos e sem as garras. Mesmo assim, os braços em destaque podem ser instalados com uma garra ou uma ventosa de sucção, bem como ser configurados individualmente, através do software intuitivo associado ao configurador/controlador industrial adequado (igus 2021a).

Uma vez que os componentes *robolink* se baseiam num princípio de peças para construir modulares, podem ser combinados com componentes especiais, podem ser adquiridos individualmente e funcionam com um controlador auto-programável. Assim, o *Robolink RL-DP* (Figura 75) é não só especialmente versátil, mas também pode ser adaptado, de forma flexível, às mudanças no processo de automação (igus 2021a), estando aptos para atuar com precisão de 1mm e cargas até 3kg em quase todos os setores e indústrias (igus 2021b), tal como está descrito na Tabela 18.



Figura 75 - Robolink® RL-DP 5 DOF (igus 2021b)

Tabela 18 - Especificações técnicas do *Robolink® RL-DP 5 DOF* (Technical n.d.)

<i>Robolink® RL-DP 5 DOF</i>	
Eixos	5
Precisão	±1 mm
Alcance máximo	790 mm
Carga útil máxima	3 kg
Alcance nominal de trabalho	450 mm
Dinâmicas com 500 g	Mín. 7
Massa	21,4 kg
Custo	~6750€

4.1.2.2.2 Braço Robótico - Dorna 2

O motor de planeamento de movimento de alto desempenho da *Dorna 2* (Figura 76) é capaz de atravessar linhas, círculos e caminhos complexos, dado que o controlador pode processar um novo comando de movimento aproximadamente a cada 2 ms. Além disso, *Dorna 2* possui um sistema de controle de *loop* totalmente fechado que garante a precisão da posição, repetibilidade e consistência do robô em qualquer velocidade, fornecendo também deteção de colisão e fácil recuperação de colisões (Robotics 2021).

No que toca aos eixos, além dos cinco do robô, o controlador *Dorna 2* permite aos usuários controlar até três eixos de *loop* fechado adicionais, os quais podem ser usados para mecanismo de preensão, rastreamento de trilhos, controlo de câmara ou qualquer outro mecanismo de automação que requeira movimento preciso e sincronizado com os eixos do robô (Robotics 2021).



Figura 76 - Dorna 2 (Ave and Ca n.d.)

Para interagir com o braço robótico recorre-se a uma interface gráfica de usuário interativa avançada baseada na web, a *Dorna Lab*. Efetivamente, o *software* não requer instalação e permite a programação, simulação e monitoramento do robô

através de um navegador web e em qualquer plataforma, incluindo PC, celular ou tablet (Robotics 2021).

As especificações técnicas estão expostas na Tabela 19.

Tabela 19 - Especificações técnicas do Dorna 2 (Ave and Ca n.d.)

Dorna 2	
Eixos	5
Precisão	$\pm 0,02$ mm
Alcance máximo	500 mm
Carga útil máxima	1 kg
Portas I/O	16
Comunicação	Servidor TCP/IP (Ethernet e Wi-Fi)
Programação	Compatível com <i>Arduino</i>
Massa	5,5 kg
Custo	~2880€

4.1.2.2.3 Braço Robótico - BCN3D MOVEO

De forma a colmatar o elevado custo dos braços robóticos industriais, a *BCN3D* procurou uma solução *open-source* e verdadeiramente *low-cost*. Efetivamente, a estrutura é totalmente impressa usando tecnologia de manufatura aditiva (BCN3D Technologies 2016a), mais propriamente material ABS, que, por sua vez, é uma excelente escolha para a criação de produtos finais que exigem propriedades mecânicas e térmicas (BCN3D 2021). Além do mais, o software *Arduino*, também *open-source*, controla os seus componentes eletrónicos.

Sendo *open-source*, outra característica de realce, os ficheiros *BCN3D Moveo* estão disponíveis para todos, o que significa que qualquer pessoa pode obter todas as informações necessárias para imprimir e montar o seu próprio braço robótico (Figura 77). De facto, os utilizadores podem encontrar a lista de materiais (*BOM*), onde vêm detalhados todos os componentes necessários para a montagem do braço, assim como os desenhos CAD, para que qualquer pessoa possa modificar o design do *BCN3D Moveo* como desejar. Os usuários também encontrarão os ficheiros STL para a impressão da estrutura e os manuais de montagem, afinação e upload de *firmware* (BCN3D 2021).



Figura 77 - BCN3D MOVEO (BCN3D Technologies 2016a)

As especificações técnicas do BCN3D MOVEO são apresentadas na Tabela 20.

Tabela 20 - Especificações técnicas do BCN3D MOVEO (BCN3D Technologies 2016a)

BCN3D MOVEO	
Eixos	5
Precisão	~0,03º
Alcance máximo	~750 mm
Carga útil máxima	2 kg
Portas I/O	Dependente do microcontrolador associado
Comunicação	Servidor TCP/IP (Ethernet e Wi-Fi)
Programação	Compatível com <i>Arduino</i>

Embora o braço robótico não tenha propriamente um preço fixo, é necessário elaborar uma lista de materiais com base naquela proveniente dos ficheiros referidos anteriormente, para obtermos um valor total estimado, tal como está patente na Tabela 21 e Tabela 22.

Tabela 21 - Lista de Materiais necessários à montagem do braço robótico *BCN3D MOVEO*, adaptado de (BCN3D Technologies 2016b)

Lista de Materiais – Braço Robótico				
Componente	Referência	Qtd.	Preço/un	Orçamento
Nema 17 Largo - Motor passo-a-passo "SM42HT47-1684 1,8º step, 1,6 A, 3600 g·cm"	096-6487	1	18,33 €	18,33 €
Driver do motor passo-a-passo "tb6560"	PTR007669	6	11,87 €	71,22 €
Fonte de energia 24V 320W	FAI24V165AC	1	33,83 €	33,83 €
Conversor de energia 24V para 12V	076-2121	1	33,11 €	33,11 €
Servo-motor 180º 55G TORQUE=13KG/CM	096-6268	1	9,20 €	9,20 €
Motor passo-a-passo "Ratio 5:1 Planetary Gearbox Nema 17 Stepper 17HS19-1684S-PG5 (step 0,35º, 2Nm, 1,68A)"	17HS19-1684S-PG5	1	30,16 €	30,16 €
Motor passo-a-passo "Nema 23, 112mm, Ø8mm flat shaft SM57HT112-3004A, step 1,8º, 3A, 28Kg·cm"	096-6486	2	81,10 €	162,20 €
Nema 14 - Motor passo-a-passo "SM35HT36-1004A, 1,8º step, 1A, 1400g·cm"	B089Y48SWK	1	13,76 €	13,76 €
Motor passo-a-passo "Nema 17 17HS13-0404S1, 1,8º step, 1,33A, 2200 g·cm"	PTR005782	1	15,87 €	15,87 €
Barra lisa de aço cromado - 134mm	PTR008342	1	6,52 €	6,52 €
Barra lisa de aço cromado - 114mm		1		0,00 €
Barra lisa de aço cromado - 80mm		1		0,00 €
Barra de haste M8 L 42mm		1		0,00 €
Rolamento de esferas 608ZZ 8mm x 22mm x 7mm	PTR003382	10	1,05 €	10,50 €
Rolamento de esferas 625ZZ 5mm x 16mm x 5mm	625zz	8	0,90 €	7,20 €
Rolamento de esferas 624ZZ 4mm x 13mm x 5mm	624zz	9	1,00 €	9,00 €
Rolamento de esferas 623ZZ 3mm x 10mm x 4mm	623zz	3	0,65 €	1,95 €

Porca de inserção em latão M4	B07YSV66Y5	3	0,22 €	0,66 €
Porca de inserção em latão M3	B08BCRZZS3	20	0,11 €	2,20 €
Polia T5, calibre 8mm, 14 dentes, 16mm largura	Aliexpress1	3	4,97€	14,91 €
Polia T5, calibre 5mm, 10 dentes, 16mm largura	Aliexpress2	2	4,47 €	8,94 €
Acoplador em aço 5 a 8mm - rígido	PTR007441	1	2,21 €	2,21 €
Ventilador axial DC 24V 80x80mm	V248S	1	18,45 €	18,45 €
Ventilador axial DC 24V 50x50mm	V245L	1	8,92 €	8,92 €
Cabo de alimentação IEC 1,8m	CUSAFSM1.8	1	4,43 €	4,43 €
Cabo USB 2.0 AM/BM 1,8m	PTR000700	1	1,85 €	1,85 €
Abraçadeiras	PTR000359	10	0,02 €	0,20 €
Rolamento <i>selfoil</i> 5mm x 8mm x 10mm	VTR	8	1,18 €	9,44 €
Rolamento <i>selfoil</i> 8mm x 12mm x 20mm	VTR	2	1,37 €	2,74 €
Base de madeira - 550mm x 550mm x 16mm		1		0,00 €
Cinto <i>breco</i> T5, 16mm largura, 180cm comprimento	aliexpress1	1	13,56 €	13,56 €
Parafuso M3 x 8mm	092-0038	4	0,05 €	0,20€
Parafuso M3 x 10mm	82582805	30	0,07 €	2,00 €
Parafuso M3 x 12mm	82582807	6	0,07 €	0,42 €
Parafuso M3 x 16mm	82582809	13	0,07 €	0,91 €
Parafuso M3 x 20mm	82582810	4	0,07 €	0,28 €
Parafuso M3 x 25mm	82582812	5	0,08 €	0,40 €
Parafuso M3 x 30mm	82582764	2	0,08 €	0,16 €
Parafuso M3 x 35mm	B08JC7Q1NZ	4	0,25 €	1,00 €
Parafuso M3 x 40mm	092-0230	7	0,15 €	1,05 €
Parafuso M4 x 10mm	82582756	2	0,10 €	0,20 €
Parafuso M4 x 12mm	VTR	2	Fornecidos	-
Parafuso M4 x 16mm	VTR	8	Fornecidos	-
Parafuso M4 x 20mm	VTR	3	Fornecidos	-
Parafuso M4 x 25mm	VTR	4	Fornecidos	-
Parafuso M4 x 30mm	82582774	8	0,13 €	1,04 €
Parafuso M4 x 40mm	82582775	4	0,13 €	0,52 €
Parafuso M4 x 45mm	82231809	6	0,06 €	0,36 €
Parafuso M4 x 60mm	82582778	4	0,20 €	0,80 €
Parafuso M5 x 14mm	M5X14/D912-A2	120	0,03 €	3,60 €
Parafuso M5 x 20mm	AlpeMotor	8	0,08 €	0,64 €
Parafuso M8 x 65mm	AlpeMotor	1	0,25 €	0,25 €
Porca M3	82583002	30	0,01 €	0,30 €

Porca M4	82583003	4	0,02 €	0,08 €
Porca autoblocante M4	AlpeMotor	18	0,03 €	0,54 €
Porca autoblocante M5	AlpeMotor	16	0,03 €	0,48 €
Porca autoblocante M8	AlpeMotor	2	0,04 €	0,08 €
Anilha M3	AlpeMotor	10	0,02 €	0,20 €
			Total	597,41 €

Tabela 22 - Lista de materiais de impressão para o braço robótico *BCN3D MOVEO*

Lista de Materiais de Impressão – Braço Robótico				
Componente	Referência	Qtd.	Preço/un	Orçamento
Rolo de Filamento	096-6919	3	17,99 €	53,97€

4.1.2.2.4 Impressora 3D – BeeveryCreative B2X300

O elevado volume de impressão permite imprimir objetos grandes (até dezoito litros) ou vários objetos de uma vez, sem a necessidade de intervenção humana. Além disso, é também possível imprimir dois tipos diferentes de materiais com temperaturas de extrusão diferentes.

A impressora em questão (Figura 78) permite evitar a confusão de ajustar o desfaseamento do nivelamento da mesa. Para tal, foi desenvolvido um método simples que não requer várias impressões de teste, alcançando excelentes resultados numa única etapa (BeeveryCreative 2019).

Outra característica forte da *B2X300* é a capacidade de detetar a ausência de filamento, reagindo de acordo, bem como a aptidão de o trabalho de impressão poder ser retomado após uma perda de energia.

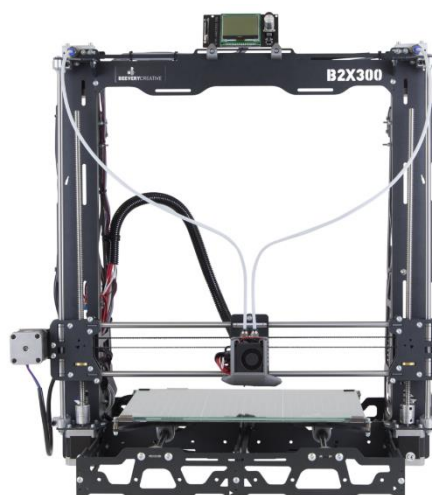


Figura 78 - BeeveryCreative B2X300 (BeeveryCreative 2019)

A solução em foco está ainda preparada para futuras atualizações e personalizações, para melhor atender às suas necessidades específicas e imprimir de forma silenciosa e

reduzir significativamente as vibrações sem perder a velocidade de impressão (BeeveryCreative 2019).

A Tabela 23 destaca as especificações técnicas da impressora em ênfase.

Tabela 23 - Especificações técnicas da *BeeveryCreative B2X300*, adaptado de (BeeveryCreative 2019)

<i>BeeveryCreative B2X300</i>	
Tecnologia de Modelação	FDM (<i>Fused Deposition Modeling</i>)
Nº de Extrusores	2
Volume de impressão	300 x 200 x 300 mm
Dimensões da impressora	560 x 375 x 600 mm
Filamento	PLA/ABS/PETG/ <i>Nylon</i> /PVA/PC/HIPS
Entrada	AC 110-240V 50-60Hz
Saída	360W
Resolução da camada	0,05mm a 0,3mm
Diâmetro do filamento	1,75 mm
Espessura do bico	0,4 mm
Temperatura do bico	300°C
Temperatura máxima de “cama quente”	130°C
Conectividade	Leitor de cartões SD; Cabo USB
Placa-mãe	MKS Gen v1.4
Ecrã LCD	Contém
Peso	12 kg
Preço	~650€

4.1.2.2.5 Impressora 3D – *Creality Ender 3*

O primeiro aspeto que salta à vista é o seu preço extremamente baixo, tendo em conta a qualidade de impressão que se consegue obter desta máquina (Figura 79). A outra propriedade de destaque é a oferta de recursos que não se encontram em impressoras 3D mais caras: um volume de construção decente de 220 x 220 x 250 mm, uma cama magnética, um modo de recuperação de energia e um caminho de filamento estreito que torna mais fácil trabalhar com materiais flexíveis (All3DP 2020).

Por outro lado, comparado a uma placa de construção de metal firme, mas flexível, o leito magnético do *Ender 3 Pro* é positivamente frágil. Além da sua adição ter sido um fracasso, ele não se encaixa numa posição predeterminada, aderindo ao local onde pousa (All3DP 2020).



Figura 79 - *Creality Ender 3* (Creality 2020)

Mesmo assim, a impressora em realce é capaz de fornecer excelente qualidade de impressão, muito graças à sua tecnologia de extrusão avançada. Efetivamente, a extrusora atualizada reduz muito o risco de entupimento e extrusão ruim, assim como o formato em V com rodas *POM* torna o movimento silencioso, suave e durável (Creality 2020).

No que toca às especificações técnicas, as mesmas encontram-se tabeladas, mais concretamente, na Tabela 24.

Tabela 24 - Especificações técnicas da *Creality Ender 3*, adaptado de (All3DP 2020)

<i>Creality Ender 3</i>	
Tecnologia de Modelação	FDM (<i>Fused Deposition Modeling</i>)
Volume de impressão	220 x 220 x 250 mm
Dimensões da impressora	440 x 410 x 465 mm
Filamento	PLA/ABS/TPU
Entrada	AC 100-265V 50-60Hz
Saída	DC 24V 15A 360W
Precisão	±0,1 mm
Espessura da impressão	0,1-0,4 mm
Velocidade máxima de impressão	180 mm/s
Espessura do bico	0,4 mm
Temperatura máxima de “cama quente”	110°C
Conectividade	Leitor de cartões SD
Ecrã LCD	Contém
Peso	8 kg
Preço	~130€

4.1.2.2.6 Impressora 3D – Anet A8

Sendo compatível com vários materiais, incluindo ABS, PLA, pasta de madeira, *Nylon* PVA, PP e outros, a *Anet A8* (Figura 80) apresenta uma moldura de acrílico preto cortado a laser e volume de impressão de 220 x 220 x 240 mm. Além disso, a impressora possui mais recursos de destaque, nomeadamente polias totalmente metálicas para funcionalidade e desempenho aprimorados; engrenagens de alimentação de libertação rápida para extrusão de filamento rápida e eficiente; e hastes de trilho, engrenagens, rolamentos e conectores de aço inoxidável (Koslow 2018).

Em contrapartida, o principal obstáculo é encontrar tempo e paciência para montar e otimizar a impressora 3D. De facto, não só o processo de montagem é relativamente complexo (especialmente para iniciantes), mas calibrar a *Anet A8* também exige precisão e um certo nível de estoicismo. Contudo, tendo em conta o preço, a capacidade de modificação e a qualidade de impressão surpreendentemente boa, estamos perante uma impressora 3D desejável e atrativa (Koslow 2018).



Figura 80 - *Anet A8* (Componentes 2018)

As especificações técnicas da impressora em questão são apresentadas na Tabela 25.

Tabela 25 - Especificações técnicas da *Anet A8*, adaptado de (Koslow 2018)

Anet A8	
Tecnologia de Modelação	FDM (<i>Fused Deposition Modeling</i>)
Volume de impressão	220 x 220 x 240 mm
Dimensões da impressora	500 x 400 x 450 mm
Filamento	PLA/PVA/ABS/TPU/PP/ <i>Nylon</i> /Madeira
Entrada	110-220V
Saída	250W
Precisão	±0,1-0,2 mm
Espessura da impressão	0,1-0,3 mm
Velocidade máxima de impressão	100 mm/s

Espessura do bico	0,2/0,3/0,4/0,5/0,6 mm
Temperatura máxima de “cama quente”	100°C
Conectividade	Leitor de cartões SD
Ecrã LCD	Contém
Peso	9 kg
Preço	~270€

4.1.2.2.7 Fresa CNC – 3018 PRO CNC Router

Uma fresadora ou “roteador” CNC baseia-se na manufatura subtrativa, no sentido em que é semelhante à perfuração, mas tem a capacidade de cortar horizontalmente, bem como verticalmente. Como tal, a *CNC 3018* e a *CNC 3018 PRO* são máquinas genéricas de esculpir e trabalhar com as mesmas oferece uma visão aprofundada de como as máquinas CNC funcionam a um nível mecânico e de programação (MT360 2020).

Efetivamente, a *CNC 3018* (Figura 81) trata-se de um *kit*, que, após algumas horas de montagem, é transformado numa mesa CNC relativamente sólida que pode esculpir uma variedade de materiais, até alumínio macio. Dispõe de uma pinça *ER-11* que oferece alguma versatilidade nas fresas de topo utilizadas, e o kit é fornecido com um pequeno conjunto de brocas em forma de V. Como o nome sugere, apresenta uma área de trabalho de 300 x 180 mm nos eixos X e Y, e o fuso (fornecido) pode “mergulhar” até 45 mm (All3DP 2021c).

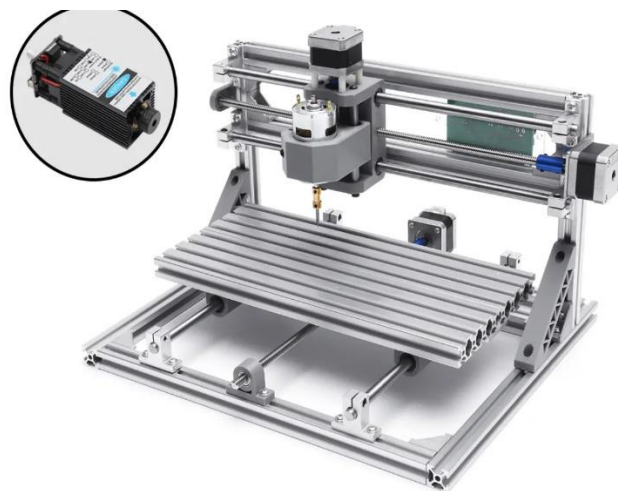


Figura 81 - *SainSmart 3018 CNC Pro* (Bangood 2020)

As especificações técnicas da fresadora proposta expõem-se na Tabela 26.

Tabela 26 - Especificações técnicas da *SainSmart 3018 CNC Pro*, adaptado de (SainSmart 2020)

<i>SainSmart 3018 CNC Pro</i>	
Área de Trabalho	300 x 180 x 45 mm

Área da máquina	330 x 400 x 240 mm
Materiais	Madeiras, plásticos, PVC's, alumínio leve (6061),
Motor “passo-a-passo”	Motor “passo-a-passo” 42; 12V; 1,3A; 0,25 N.m
Motor <i>spindle</i>	Motor <i>spindle</i> 775; 12-36V; 24V 7000r/min; 36V – 9000r/min
Software de controlo	<i>GrbIControl</i>
Fonte de energia	DC 24V 5,6A
Peso	7,5Kg
Preço (kit)	~150€

4.1.2.2.8 Fresa CNC – Millright CNC M3

MillRight CNC M3 (Figura 82) é uma fresadora CNC de mesa que possui um kit de 3 eixos, incluindo a combinação de energia e corte de custos, que é prático e de orçamento acessível. Adicionalmente apresentam motores de passo *premium*, design de fibra de vidro de 9 mm, que, por sua vez, se encontram ligados ao sistema da placa de controlo do *Arduino UNO*, o que permite deslocar a mesa através de um eixo e *spindle* nos outros (Cutter 2020).

Além disso, este kit de fresadora inclui corte a laser, anodizado com placas de alumínio, bem como a praticabilidade do parafuso em todos os eixos, ao invés de cintos nos eixos bidimensionais. De facto, a precisão e a aplicação direta estão muito bem projetadas para isso (Cutter 2020).



Figura 82 - *Millright CNC M3* (All3DP 2021b)

A Tabela 27 mostra as respetivas especificações técnicas.

Tabela 27 - Especificações técnicas da *Millright CNC M3*, adaptado de (MillRight 2017)

Millright CNC M3	
Área de Trabalho/Movimento	260 x 260 x 51 mm
Área da máquina	483 x 483 x 419 mm
Materiais	Madeiras, plásticos, alumínio, fibra de carbono, placas folheadas de cobre FR4, etc.
Ferramenta de corte	<i>DeWalt DWP611 Router</i> , 1.25 HP, 16,000-27,000 RPM, ¼" collet
Motor	Nema 17, 76 oz/in
Software de controlo	<i>Grb1Control 0,9</i>
Fonte de energia	24V; 120W
Peso	11,34 Kg
Preço (kit)	~450€

4.1.2.3 Análise e Orçamento Final

Em relação ao braço robótico, e após uma breve análise da relação entre as premissas de *open-source* e *low-cost*, chega-se facilmente à conclusão de que o *BCN3D MOVEO* é aquele que as melhor preenche. De facto, este braço robótico é controlado por uma plataforma completamente open-source, mais propriamente a *Arduino*, sendo que esta controla igualmente um dispositivo *IoT*, o que facilita a comunicação entre os dois componentes. O único senão desta proposta é a necessidade da sua montagem, o que requer alguma perícia no assunto, mas quando atendemos ao valor associado, este passa a ser um risco que compensa tomar.

No que toca à impressora 3D, o critério não se baseou somente nas premissas previamente referidas, uma vez que as sugestões apresentam sensivelmente o mesmo nível de open-source e baixa discrepância nos custos análogos. Efetivamente, foi também tido em conta a qualidade das impressoras, principalmente da impressão em si, concluindo-se que a mais adequada seria a *BeeveryCreative B2X300*.

Relativamente à fresadora CNC, como ambas as propostas são compatíveis com o mesmo controlador, analisou-se qual é que satisfaz de melhor forma o conceito inerente e que, ao mesmo tempo, acarreta um menor custo. Com isto, a 3018 PRO CNC Router mostra-se como a solução.

Assim sendo, o orçamento total, e final, do sistema em ênfase corresponde a 1450€ (Tabela 28).

Tabela 28 - Orçamento Final do Sistema de Produção

Orçamento Final		
Componentes	Quantidade	Preço(€)
Braço Robótico	1	650
Fresa CNC	1	150
Impressora 3D	1	650
Total		1450

4.1.3 Sistema de Automação

Atendendo ao que foi retratado em 3.4.3, surge, então, o tópico seguinte, no qual serão propostos diferentes tipos de *microswitches*, seguido de uma análise dos mesmos com o intuito de concluir qual o mais adequado para o caso de estudo.

Embora tenham sido igualmente referidos em 3.4.3, os motores “passo-a-passo” não serão debatidos, uma vez que fazem parte da composição dos respetivos recursos físicos, estando, assim, dependentes dos mesmos.

4.1.3.1 Propostas

O *microswitch MS5-L* (Figura 83) trata-se de um interruptor elétrico de (longa) alavanca, em que as suas especificações técnicas encontram-se na Tabela 29.



Figura 83 - Microswitch MS5-L (Eletrónica 2021a)

Tabela 29 - Especificações técnicas do MS5-L, adaptado de (Velleman 2021)

MS5-L	
Resistência de contacto	Início máximo: 30mohm
Resistência de isolamento	Mínimo: 100Mohm a 500V
Rigidez dielétrica	1000V p/min
“Vida” mecânica	1000000 operações
“Vida” elétrica	200000 operações
Dimensões	10,2 x 19,8 x 6,4mm
Energia	5A/125V; 3A/250V

Preço**1,30€**

O *D3V-16-1C24* é igualmente um interruptor elétrico, mas diferente do anterior, visto que não apresenta alavanca. As suas especificações técnicas também se encontram tabeladas, mais propriamente, na Tabela 30.



Figura 84 - D3V-16-1C24 (Eletrónica 2021b)

Tabela 30 - Especificações técnicas do D3V-16-1C24, adaptado de (Gap 2021)

<i>D3V-16-1C24</i>	
Resistência de contacto	Início máximo: 30mohm
Resistência de isolamento	Mínimo: 100Mohm a 500V
Rigidez dielétrica	1000V p/min
“Vida” mecânica	10000000 operações
“Vida” elétrica	100000 operações
Dimensões	27,8 x 10,3 x 15,9mm
Energia	16A/250V; 10A/30V
Preço	1,95€

O SW015 (Figura 85) também se trata de um interruptor elétrico, mas o que o diferencia dos anteriores é a inclusão de uma patilha e um rolamento. As especificações técnicas associadas estão expostas na Tabela 31.



Figura 85 - SW015 (Mauser.pt 2021)

Tabela 31 - Especificações técnicas do SW015, adaptado de (Mauser.pt 2021)

<i>SW015</i>	
Resistência de contacto	Início máximo: 30mohm
Resistência de isolamento	Mínimo: 100Mohm a 500V

Comprimento da patilha	12 mm
“Vida” mecânica	10000000 operações
“Vida” elétrica	100000 operações
Dimensões	28 x 10,3 x 16 mm
Energia	10A/250V
Preço	1,12€

4.1.4 Sistema de Comunicação e Vigilância

4.1.4.1 Lista de Componentes e Orçamento Primordial

Tendo em conta o descrito em 3.4.4, tabelou-se (Tabela 32), então, a lista de componentes pertencentes ao subsistema em causa, na qual são também apresentadas as dimensões aproximadas inerentes aos mesmos, resultantes de uma consulta a nível industrial. De tal alcançaram-se igualmente os preços relativos aos componentes, os quais se encontram tabelados do mesmo modo na Tabela 33.

Tabela 32 - Lista de componentes do Sistema de Comunicação e Vigilância

Lista de Componentes		
Componentes	Quantidade	Dimensões aproximadas (mm)
Vídeocâmara	8	80x80x120
(Micro)Computador Operacional	1	64x56x12
Monitores	3	50x30x5
Auscultadores	1	-
Microfone	1	-

Tabela 33 - Orçamento primordial do Sistema de Comunicação e Vigilância

Orçamento primordial	
Componentes	Preço estimado (€)
Vídeocâmara	20
(Micro)Computador Operacional	Já contabilizado
Monitores	90
Auscultadores	15
Microfone	5
Total	450

4.1.4.2 Propostas

4.1.4.2.1 Vídeocâmara – ArduCAM Mini (OV2640)

ArduCAM Mini (Figura 86) é uma câmara *SPI* de alta-definição, o que reduz a complexidade da interface de controlo da câmara. A mesma integra o sensor de imagem *CMOS 2MP OV2640* e dispõe de um tamanho em miniatura, bem como uma interface de *hardware* fácil de usar e uma biblioteca de código *open-source*.

Além disso, a *ArduCAM Mini* pode ser usada em qualquer plataforma, desde que tenha interface *SPI* e *I2C*, com destaque na boa acoplação em placas *Arduino*, dada a compatibilidade do seu *shield* nas mesmas. A opção em estudo não só oferece a capacidade de adicionar uma *interface* de câmara, o que não existe em alguns microcontroladores de baixo custo, mas também oferece a capacidade de adicionar várias câmaras a um único microcontrolador (ArduCAM 2015).

As especificações técnicas da *ArduCAM Mini* (OV2640) encontram-se expostas na Tabela 34.



Figura 86 - ArduCAM Mini (OV2640) (ArduCAM 2019)

Tabela 34 - Especificações técnicas da *ArduCAM Mini*, adaptado de (ArduCAM 2015)

<i>ArduCAM Mini</i>	
Sensor de Imagem	OV2640
Resolução	1600x1200
Lente	¼ polegadas
Velocidade SPI	8 Mhz
Tamanho do Frame Buffer	384 kB
Dimensões	34 x 24 mm
Peso	20 g
Formato de saída da imagem	JPEG
Energia	5V
Preço	25€

4.1.4.2.2 Videocâmara – ESP32 CAM (OV2640)

O módulo *ESP32* é um microcontrolador poderoso, porém barato, com recursos avançados, como *Bluetooth*, *WiFi* e portas *GPIO* multiuso. Em contrapartida, o *ESP32-CAM* (Figura 87) trata-se desse microcontrolador completo, mas com uma câmara de vídeo integrada e uma entrada para cartão *microSD*, o que o torna perfeito para dispositivos *IoT* que requerem uma câmara com funções avançadas como rastreamento e reconhecimento de imagem. Outra característica a favor desta opção é

a sua compatibilidade com *Arduino*, quer a nível de placa, quer a nível de programação (Workshop 2020).

A Tabela 35 mostra as suas especificações técnicas.



Figura 87 - ESP32 CAM (Mauser.pt 2018)

Tabela 35 - Especificações técnicas da ESP32 CAM, adaptado de (AI-Thinker 2017; Workshop 2020)

ESP32 CAM	
Memória flash	32Mbit
RAM	520KB SRAM + 4M PSRAM
Bluetooth	Bluetooth 4.2 BR/EDR
Wi-Fi	802.11 b/g/n
Interfaces	UART/SPI/I ² C/PWM
Portas IO	9
Câmara (OV2640)	Sensor 2 MP; UXGA 1622x1200; 15 a 60fps
Formato de saída da imagem	JPEG; BMP; GRAYSCALE
Energia	5V
Dimensões	27 x 40,5 x 4,5 mm
Preço	11,30€

4.1.4.2.3 Computador – RaspberryPi 400

O *Raspberry Pi 400* (Figura 88) é um computador pessoal completo, integrado num teclado compacto e possui um processador *quad-core* de 64 *bits*, rede *wireless*, saída para dois monitores e reprodução de vídeo 4K. Este computador é ideal para navegar na internet, criar e editar documentos, assistir a vídeos, aprender a programar usando o ambiente de desktop *Raspberry Pi OS* e a gerir comunicações (Pi 2020), como é o caso da sua principal função na arquitetura proposta.

Mais uma vez, as especificações técnicas encontram-se tabeladas, mais concretamente, na Tabela 36.

Figura 88 - *Raspberry Pi 400* (Foundation 2020)Tabela 36 - Especificações técnicas do *Raspberry Pi 400*, adaptado de (Pi 2020)

Raspberry Pi 400	
Processador	Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
RAM	4GB LPDDR4-3200
Conectividade	Dual-band (2.4GHz and 5.0GHz) IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 and 1 × USB 2.0 ports
GPIO	Entrada horizontal GPIO de 40 pins
Som e Vídeo	2 portas micro HDMI (suporta até 4Kp60)
Multimédia	H.265; H.264; Gráficas integradas <i>OpenGL ES 3.0</i>
Armazenamento	<i>Slot</i> para cartão MicroSD para sistema operacional e armazenamento de dados
Energia	5V DC via USB
Dimensões	286 x 122 x 23 mm (máximo)
Preço	~76€

4.1.4.2.4 Monitor – Monitor Asus 18.5"

O monitor *Asus VS197DE* (Figura 89) retro iluminado por *LED's*, com alta relação de contraste de 500000:1, é otimizado para proporcionar as melhores imagens e

qualidade de cor, uma vez que a resistente base e o perfil fino que não comprometem o estilo e garantem estabilidade e durabilidade. Além disso, melhora o contraste do ecrã dinamicamente ajustando a iluminação da luz de fundo para atingir o preto mais escuro e o branco mais claro, oferecendo imagens realistas; e aumenta o espaço visual em configurações de dois ecrãs para melhorar a eficiência multitarefa (PCDIGA 2021).

Na Tabela 37 estão apontes as especificações técnicas do monitor em questão.



Figura 89 - Monitor Asus VS197DE (PCDIGA 2021)

Tabela 37 - Especificações técnicas do Asus VS197DE, adaptado de (PCDIGA 2021)

Asus VS197DE	
Tamanho do Ecrã	<i>Wide Screen</i> 18.5" (47 cm) 16:9
Resolução	1366x768
Tempo de resposta	5 ms
Dimensões	447 x 346 x 185 mm
Peso	2,6 kg
Preço	~80€

4.1.4.2.5 Auscultadores – Ewent EM3563

Os Ewent EM3563 (Figura 90) são auscultadores de ouvido estéreo de alta qualidade com microfone flexível, ajustável para comunicação mãos-livres e ideal para aplicações de *voice chat* e *video chat*. Dispõem também de controlo de volume integrado no cabo e um design confortável com esponjas de ouvido macias e haste de cabeça ajustável (Velleman 2016).



Figura 90 - Ewent EM3563 (Velleman 2016)

Tabela 38 - Especificações técnicas do Ewent EM3563, adaptado de (Mauser.pt 2017; Velleman 2016)

Ewent EM3563	
Jack	3.5mm (4 pin) stereo
Dimensões	Ø30 mm
Sensibilidade	110dB ± 3dB
Frequência	20Hz – 20kHz
Comprimento do cabo	1,8m
Preço	~9€

4.1.4.2.6 Microfone

Trata-se do microfone incorporado nos auscultadores previamente definidos, daí não apresentar custos adicionais.

Tabela 39 - Especificações técnicas do microfone, adaptado de (Mauser.pt 2017; Velleman 2016)

Microfone (Incorporado nos auscultadores Ewent EM3653)	
Dimensões	Ø6 x 5mm
Sensibilidade	0dB – 1V/Pa a 1kHz – 58dB

4.1.4.3 Escolha de Componentes e Orçamento Final

A escolha dos componentes deste sistema é unidirecional, visto que só é apresentada uma proposta para os vários elementos, à exceção das videocâmaras (de vigilância), onde são feitas duas propostas. Efetivamente, a primeira, tal como o nome indica, está tecnicamente adaptada para as placas *Arduino*, o que facilita a sua conexão ao restante sistema, mas o seu preço mostra-se como um entrave. Por outro lado, a segunda apresenta um valor mais acessível, mas requer elementos de ligação

secundários. Mesmo assim, esta opção, nomeadamente a *ESP32-CAM*, é a mais indicada.

Posto isto, o orçamento total, e final, do sistema em realce é de 329€, assim como está exposto na Tabela 40.

Tabela 40 - Orçamento Final do Sistema de Comunicação e Vigilância

Orçamento Final		
Componentes	Quantidade	Preço (€)
Vídeocâmara	4	10
(Micro)Computador Operacional	1	100
Monitores	2	90
Auscultadores	1	9
Microfone	1	0
Total		329

4.2 Sistema computacional em Nuvem (*cloud*)

4.2.1.1 IDE's

4.2.1.1.1 Proposta - *NetBeans IDE*

O projeto *NetBeans* (Figura 91) consiste num ambiente de desenvolvimento integrado *open-source* e uma plataforma aplicativa que permite aos desenvolvedores criar rapidamente aplicações web, corporativas, de desktop e móveis, tal como mostra a Figura 92. Além dessa plataforma, o IDE em questão também permite a utilização de *JavaFX*, *PHP*, *JavaScript*, *Ajax*, *Ruby*, *C/C++*, etc. (Hield 2012).



Figura 91 - Logótipo do *NetBeans* (Lenz 2017)

Em termos comparativos aos demais *IDE's*, Myatt (2007) listou dez razões que diferenciam o *NetBeans* dos restantes:

- Designer intuitivo e fácil de usar do *GUI Matisse* para desenvolvimento *Swing*;
- Forte suporte de reestruturação;
- Um dos melhores *profilers*;
- Suporte a projetos *UML*;
- Integração *Ant*;
- Suporte a aplicações móveis *J2ME*;
- Ferramentas de colaboração para desenvolvedores;
- Centro de atualização fácil de usar;

- Suporte *JSP* e *Tomcat* pronto para uso;
- Monitor *NetBeans HTTP*.

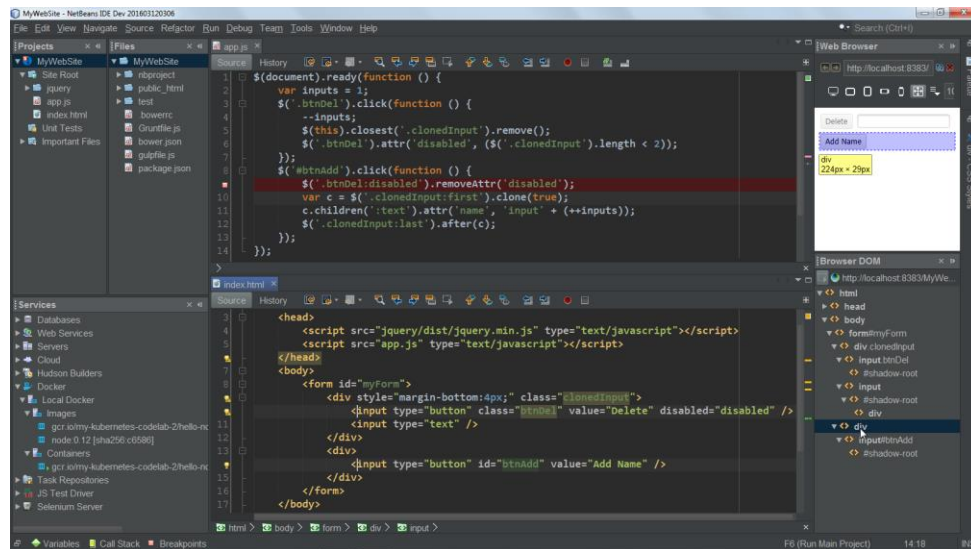


Figura 92 - Exemplo de um projeto em *NetBeans* (Wielenga 2016)

4.2.1.1.2 Proposta – Eclipse

Eclipse (Figura 93) é uma comunidade *open-source* cujos projetos são focados em fornecer uma plataforma de desenvolvimento extensível e estruturas aplicativas para, lá está, o desenvolvimento de *software* (Wahid et al. 2017).



Figura 93 - Logótipo do *Eclipse IDE* (Foundation 2012)

Efetivamente, o *Eclipse IDE* (Figura 94) fornece ferramentas e estruturas extensíveis que abrangem o ciclo de vida de desenvolvimento de *software*, incluindo suporte para modelação, ambientes de desenvolvimento de linguagem para *Java*, *C/C++* e outros, testes e desempenho, inteligência de negócios, aplicações *rich-client* e desenvolvimento integrado (Wahid et al. 2017).

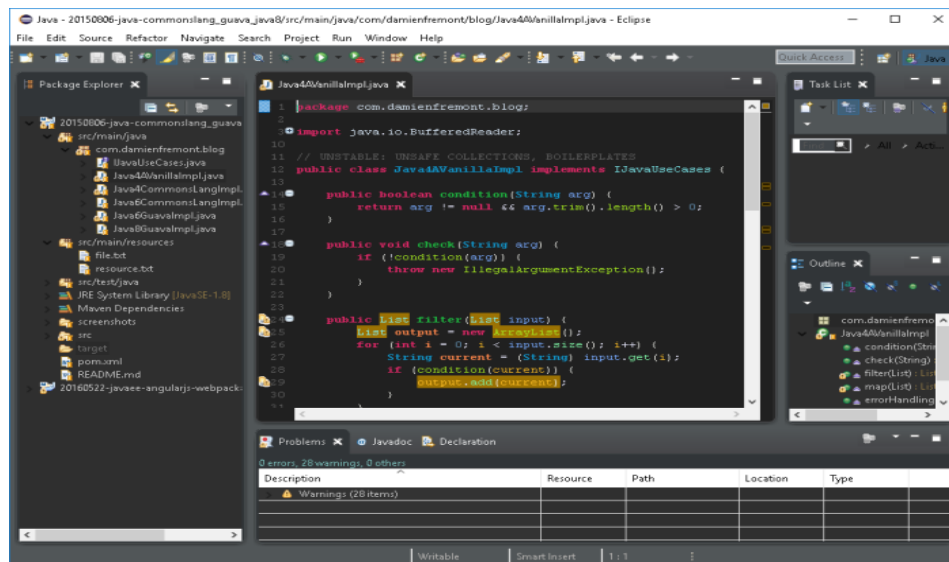


Figura 94 - Exemplo de um projeto em Eclipse (Java Tutorial 2018)

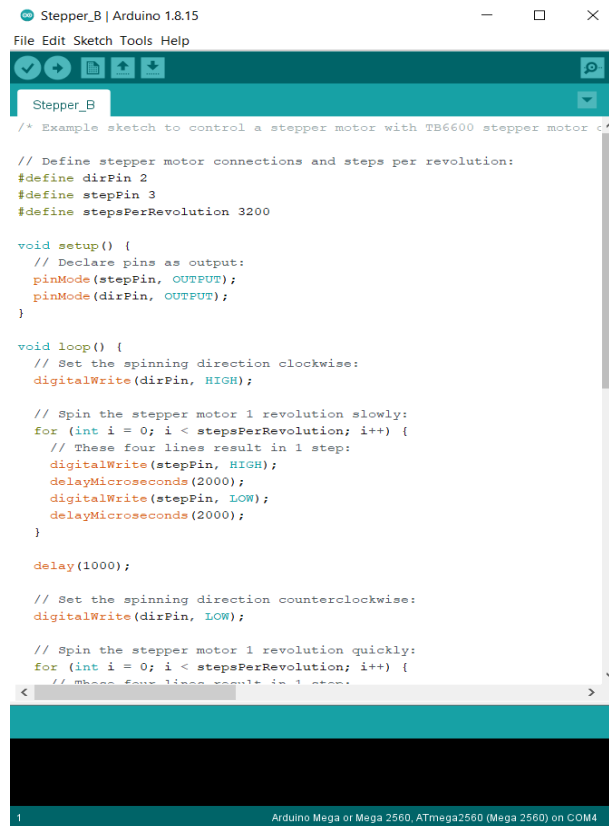
4.2.1.1.3 Proposta – Arduino IDE

Arduino IDE (Figura 95) é um software *open-source* usado principalmente para escrever e compilar código, tendo, na generalidade, como base uma vasta biblioteca que fornece muitos procedimentos comuns baseados em sinais de entrada e saída, o que o torna extremamente acessível, mesmo a alguém sem nenhum conhecimento técnico prévio (Badamasi 2014).



Figura 95 - Logótipo do Arduino (Arduino 2015)

O ambiente em realce na Figura 96, além de suportar as linguagens *C* e *C++*, contém principalmente duas partes básicas: Editor e Compilador, onde o primeiro é usado para escrever o código necessário e o último é usado para compilar e enviar o código para um módulo *Arduino*. Esses módulos podem ser, por exemplo, o *Arduino Uno* e o *Arduino Mega*, que, por sua vez, contêm um microcontrolador na placa que está verdadeiramente programado e aceita as informações em forma de código (Fezari and Dahoud 2018).



The screenshot shows the Arduino IDE interface with a sketch titled "Stepper_B". The code is a C++ program for controlling a stepper motor using a TB6600 driver. It defines two pins: dirPin (2) and stepPin (3). The stepsPerRevolution is set to 3200. The setup function configures both pins as outputs. The loop function alternates between clockwise and counter-clockwise rotation. For each direction, it spins the motor one revolution slowly (with 2000 microsecond delays between steps) and then one revolution quickly (with 1000 microsecond delays between steps). The status bar at the bottom indicates the board is "Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM4".

```
Stepper_B | Arduino 1.8.15
File Edit Sketch Tools Help

Stepper_B
/* Example sketch to control a stepper motor with TB6600 stepper motor driver

// Define stepper motor connections and steps per revolution:
#define dirPin 2
#define stepPin 3
#define stepsPerRevolution 3200

void setup() {
  // Declare pins as output:
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin, OUTPUT);
}

void loop() {
  // Set the spinning direction clockwise:
  digitalWrite(dirPin, HIGH);

  // Spin the stepper motor 1 revolution slowly:
  for (int i = 0; i < stepsPerRevolution; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(2000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(2000);
  }

  delay(1000);

  // Set the spinning direction counterclockwise:
  digitalWrite(dirPin, LOW);

  // Spin the stepper motor 1 revolution quickly:
  for (int i = 0; i < stepsPerRevolution; i++) {
    // These four lines result in 1 step:
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(1000);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(1000);
  }
}
```

1 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM4

Figura 96 - Exemplo de um projeto *Arduino*

CONCLUSÕES

5.1	Conclusões	114
5.2	Trabalhos futuros	115

5 Conclusões

5.1 Conclusões

Após a elaboração de este trabalho, conclui-se que o objetivo principal foi, de facto, atingido. Efetivamente, estabeleceu-se uma arquitetura que integra um sistema ciberfísico orientado para a produção aberta, a qual recorre ao Open Design e outros serviços/recursos para que tal seja exequível. É de realçar que tais serviços/recursos podem ser facilmente obtidos, dada a vasta variedade dos mesmos disponível aberta e gratuitamente (*open-source*). Isto significa que o sistema computacional, que suporta o sistema físico, não pode ser visto como um impedimento para os possíveis trabalhos futuros.

Ao direcionar o CPS para a comunidade, humaniza-se e enriquece-se o próprio sistema, visto que é suportada a colaboração real e síncrona de humano para humano, o que permite a coexploração do sistema com os restantes agentes. Esta participação ativa do utilizador é, igualmente, essencial para superar problemas de interoperabilidade tecnológica.

A concretização da arquitetura permite tirar uma outra ilação, uma vez que é possível alcançar o sistema proposto recorrendo a esses serviços/recursos na vertente *low-cost* e *open-source*, o que torna a implementação muito mais acessível, quer a nível financeiro, quer a nível prático. De facto, com cerca de 2100€ obteve-se um laboratório que serve igualmente como um protótipo a nível industrial, uma vez que apresenta componentes eletrónicos preparados para tal.

Neste contexto, o sistema multiagente agregado apoia a modularidade, o que incrementa a exploração de um maior número de módulos de produção. Tal observação corrobora a ideia de que o sistema proposto se evidencia como uma forte base para uma (pequena) indústria viável e sustentável.

Além de industrial, este laboratório também pode ser considerado um de reprodutibilidade e replicabilidade (R&R) para fins educacionais, diferenciando-se pela perspetiva educacional para a sustentabilidade, o que contribui, simultaneamente, para o cumprimento dos 17 *SDG's*.

Embora o laboratório em destaque tenha sido definido de forma específica, o presente trabalho também demonstra que existem diversas alternativas para a mesma finalidade, o que é um argumento bastante a favor para a adoção de esta abordagem no futuro empresarial. Claro está que optando por diferentes componentes, requer uma igual readaptação a nível técnico, mais concretamente, ligações e conexões.

Com a referida abordagem, o sistema adquire ainda mais características de relevância, nomeadamente, ser social (baseado na comunidade), escalável, replicável, ubíquo, de

acesso remoto e aberto e providenciar novos modelos de negócio, produção e educação.

Agregando as últimas com os conceitos inerentes à Indústria 4.0 que estão igualmente englobados na arquitetura, dos quais se destacam o *Cloud Computing*, *Cloud Manufacturing*, *Industrial Internet of Things (IIoT 4.0)* e *Smart Factory*, as questões da obrigatoriedade de grandes investimentos em laboratórios e da necessidade da disseminação do conhecimento entre as comunidades de pesquisa são também colmatadas.

5.2 Trabalhos futuros

O próximo passo passa por finalizar a implementação a nível físico, visto que os seus componentes já foram analisados e definidos, tal como é descrito em 4.1.

Posteriormente, claro está que também será necessária uma implementação a nível computacional, sendo ela, numa primeira fase, à base de controlo remoto, servindo igualmente como fase teste. Após o sucesso da mesma, o próximo objetivo recai na recriação da arquitetura proposta, isto é, alocar o sistema na *cloud*, conjugado com os respetivos recursos/serviços.

Outro aspeto fulcral a explorar é a questão da modularidade. Efetivamente, a implementação (física) primordial corresponde apenas a um módulo e, nesse sentido, pretende-se que haja uma extensão quanto ao número dos mesmos. Para que tal seja exequível, a capacidade funcional do sistema terá de ser proporcional a essa expansão, o que requer um estudo aprofundado do sistema multiagente e dos algoritmos associados, uma vez que estes elementos são o cérebro do sistema global.

BIBLIOGRAFIA E OUTRAS FONTES DE INFORMAÇÃO

6.1	Referências bibliográficas	119
-----	----------------------------------	-----

6 BIBLIOGRAFIA E OUTRAS FONTES DE INFORMAÇÃO

6.1 Referências bibliográficas

- 4ieng. 2019. "CAD, CAE E CAM: Qual a Diferença Entre Eles?" Retrieved (<https://www.4ieng.com.br/single-post/2017/03/16/cad-cae-e-cam-qual-a-diferenca-entre-eles>).
- Abdul Aziz Ahmed Siyad, Controllino Maxi. 2019. "DESIGN AND DEVELOPMENT OF IP BASED AUTOMATIC BOTTLE INSPECTION SYSTEM USING ve r Ay a ve r Si."
- AI-Thinker. 2017. "ESP32-Cam Module." *AI-Thinker Technology* 1–4.
- Al-Sarawi, Shadi, Mohammed Anbar, Kamal Alieyan, and Mahmood Alzubaidi. 2017. "Internet of Things (IoT) Communication Protocols: Review." *ICIT 2017 - 8th International Conference on Information Technology, Proceedings* (October):685–90.
- All3DP. 2020. "Creality Ender 3 Pro Review: Great 3D Printer Under \$300." Retrieved (<https://all3dp.com/1/creality-ender-3-pro-3d-printer-review/>).
- All3DP. 2021a. "3D Printing First Layer Problems: How to Make It Perfect." Retrieved (<https://all3dp.com/2/3d-printing-first-layer-problems-how-to-make-it-perfect/>).
- All3DP. 2021b. "BEST CNC ROUTERS (DIY CNC ROUTER KITS & DESKTOP CNC MACHINES)." Retrieved (<https://all3dp.com/1/best-diy-cnc-router-kit/#millright-cnc-m3>).
- All3DP. 2021c. "SainSmart Genmitsu 3018 Pro." Retrieved (<https://all3dp.com/1/best-cnc-router-desktop-cnc-machine/#sainsmart-genmitsu-3018-pro>).
- ArduCAM. 2015. "ArduCAM-M-5MP Camera Shield." 0–7.
- ArduCAM. 2019. "Arducam Shield V2." Retrieved (<https://www.arducam.com/docs/spi-cameras-for-arduino/hardware/arducam-shield-v2/>).
- Arduino.cc. 2015. "Arduino – Introduction." Retrieved (<https://www.arduino.cc/en/Guide/Introduction>).
- Arduino. 2015. "Getting Started with Arduino Products." Retrieved (<https://www.arduino.cc/en/Guide>).
- Arduino. 2019. "Arduino Nano."
- Arduino LLC, Ahmad Faishal, Maun Budiyo, Program Diploma, Teknik Elektro, H. Max, and Sugeng T. R. I. Wibowo. 2012. "Arduino Nano." 2010(semnasIF):1.
- Ave, Monte Vista and Upland Ca. n.d. "Dorna 2." 1(979):1–3.

- Badamasi, Yusuf Abdullahi. 2014. "The Working Principle of an Arduino." *Proceedings of the 11th International Conference on Electronics, Computer and Computation, ICECCO 2014*.
- Bangood. 2020. "3 Axis Mini DIY CNC Router 2500mW Laser Module Milling Engraver Machine Wood Engraving Cutting." Retrieved (https://www.banggood.com/3-Axis-Mini-DIY-CNC-Router-2500mW-Laser-Module-Milling-Engraver-Machine-Wood-Engraving-Cutting-p-1263082.html?rmmds=detail-left-hotproducts&cur_warehouse=CZ).
- BCN3D. 2021. "BCN3D MOVEO: A Fully Open Source 3D Printed Robot Arm." *BCN3D Blog* 1.
- BCN3D Technologies. 2016a. "BCN3D MOVEO: A Fully Open Source 3D Printed Robot Arm." Retrieved (<https://www.bcn3d.com/bcn3d-moveo-the-future-of-learning/>).
- BCN3D Technologies. 2016b. "BCN3D Moveo BOM.Xlsx." 3–4.
- BeeveryCreative. 2019. "B2X300 DIY KIT." Retrieved (https://shop.beeverycreative.com/produto/b2x300_kit/?lang=pt-pt).
- Bernstein, Philip a and Umeshwar Dayal. 1994. "An Overview of Repository Technology." *Proceedings of the 20th International Conference on Very Large Data Bases* (Section 6):705–13.
- Berson, Alex. 1996. *Client/Server Architecture*. McGraw-Hill, Inc.
- Blackstock, Michael and Rodger Lea. 2014. "IoT Interoperability: A Hub-Based Approach." *2014 International Conference on the Internet of Things, IOT 2014* 79–84.
- Bonvoisin, Jérémy and Robert Mies. 2018. "Measuring Openness in Open Source Hardware with the Open-o-Meter." *Procedia CIRP* 78:388–93.
- Boyes, Hugh, Bil Hallaq, Joe Cunningham, and Tim Watson. 2018. "The Industrial Internet of Things (IIoT): An Analysis Framework." *Computers in Industry* 101(March):1–12.
- Bujorianu, Marius C., Manuela L. Bujorianu, and Howard Barringer. 2009. "A Formal Framework for User-Centric Control of Multi-Agent Cyber-Physical Systems Marius Bujorianu , Manuela Bujorianu and Howard Barringer February 2009 Manchester Institute for Mathematical Sciences School of Mathematics The University of Manchester A F." *Secretary* (February).
- Carrasco, A., Ma Dolores Hernandez, Ma Del Carmen Romero-Ternero, Francisco Sivianes, David Oviedo, and Jose Ignacio Escudero. 2014. "PeMMAS: A Tool for Studying the Performance of Multiagent Systems Developed in JADE." *IEEE Transactions on Human-Machine Systems* 44(2):180–89.
- Castro, Hélio, Goran Putnik, Alrenice Castro, and Rodrigo Dal Bosco Fontana. 2019. "Open Design Initiatives: An Evaluation of CAD Open Source Software." *Procedia CIRP* 84:1116–19.
- Castro, Hélio, Goran Putnik, Alrenice Castro, and Rodrigo Dal Bosco Fontana. 2019. "Open Design Initiatives: An Evaluation of CAD Open Source Software." *Procedia CIRP* 84:1116–19.

- Castro, Hélio, Goran Putnik, Alrenice Castro, and Rodrigo Dal Bosco Fontana. 2019. "Could Open Design Learn from Wikipedia?" *Procedia CIRP* 84:1112–15.
- Chen, Baotong, Jiafu Wan, Lei Shu, Peng Li, Mithun Mukherjee, and Boxing Yin. 2017. "Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges." *IEEE Access* 6:6505–19.
- Chiaroni, Davide, Vittorio Chiesa, and Federico Frattini. 2011. "The Open Innovation Journey: How Firms Dynamically Implement the Emerging Innovation Management Paradigm." *Technovation* 31(1):34–43.
- Coccia, Mario. 2019. "Metabolism of Public Research Organizations: How Do Laboratories Consume State Subsidies?" *Public Organization Review* 19(4):473–91.
- Componentes, PC. 2018. "Anet A8 Impressora 3D." Retrieved (<https://www.pccomponentes.pt/anet-a8-impresora-3d-montada>).
- Controllino. 2017. "CONTROLLINO MEGA." Retrieved (<https://www.controllino.com/product/controllino-mega/>).
- Creality. 2020. "Ender 3 3D Printer." Retrieved (<https://www.creality3dofficial.com/products/official-creality-ender-3-3d-printer>).
- Cunha, Maria M., Goran D. Putnik, Angappa Gunasekaran, and Paulo Avila. 2005. "Market of Resources as a Virtual Enterprise Integration Enabler." Pp. 145–65 in *Virtual Enterprise Integration*. IGI Global.
- Cutter, The Edge. 2020. "MillRight CNC M3 Review: Comprehensive Detailed Guide." Retrieved (<https://theedgecutter.com/millright-cnc-m3-review/>).
- Dahlander, Linus and David M. Gann. 2010. "How Open Is Innovation?" *Research Policy* 39(6):699–709.
- Degree, Pollution. 2017. "Datasheet Controllino Maxi." 100–101.
- Derler, Patricia, Edward a. Lee, and Alberto L. Sangiovanni-vincentelli. 2011. "Addressing Modeling Challenges in Cyber-Physical Systems." 1–27.
- Derler, Patricia, Edward A. Lee, and Alberto Sangiovanni Vincentelli. 2012. "Modeling Cyber-Physical Systems." *Proceedings of the IEEE* 100(1):13–28.
- Diaz, M., Martin, C., Rubio, B. 2016. "Challenge in Integration IOT and CC 2016.Pdf." 99–117.
- Díaz, Manuel, Cristian Martín, and Bartolomé Rubio. 2016. "State-of-the-Art, Challenges, and Open Issues in the Integration of Internet of Things and Cloud Computing." *Journal of Network and Computer Applications* 67:99–117.
- Dillon, Tharam, Chen Wu, and Elizabeth Chang. 2010. "Cloud Computing: Issues and Challenges." *Proceedings - International Conference on Advanced Information Networking and Applications, AINA* 27–33.
- Distrelec. 2021. "V-165-1C5 - Micro Switch V, 16A, 1CO, 2.35N, Short Hinge Roller Lever, Omron Industrial Automation." Retrieved (<https://www.distrelec.biz/en/micro-switch-16a-1co-35n-short-hinge-roller-lever->

- omron-industrial-automation-165-1c5/p/30069511).
- Dragoni, Nicola, Saverio Giallorenzo, Alberto Lluch Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, and Larisa Safina. 2017. "Microservices: Yesterday, Today, and Tomorrow." *Present and Ulterior Software Engineering* 195–216.
- Eletrónica, Aquario. 2021a. "Microswitch 5a Actuador Longo." Retrieved (<https://www.aquario.pt/product/velleman-microswitch-5a-actuador-longo-ms5-l>).
- Eletrónica, Aquario. 2021b. "Microswitch Sem Alavanca 16a." Retrieved (<https://www.aquario.pt/product/etm-microswitch-sem-alavanca-16a-d3v161c24>).
- Elsodany, Nehal M., Sohair F. Rezek, and Noman A. Maharem. 2011. "Adaptive PID Control of a Stepper Motor Driving a Flexible Rotor." *Alexandria Engineering Journal* 50(2):127–36.
- Erwinski, Krystian, Marcin Paprocki, Lech M. Grzesiak, Kazimierz Karwowski, and Andrzej Wawrzak. 2013. "Application of Ethernet Powerlink for Communication in a Linux RTAI Open CNC System." *IEEE Transactions on Industrial Electronics* 60(2):628–36.
- Espressif Systems. 2020. "Datasheet ESP8266EX."
- European Commission. 2016. "EU Open Innovation, Open Science, Open to the World." *European Comission* (June 2015):108.
- Farnell. 2013. "Arduino Uno Datasheet." *Datasheets* 1–4.
- Ferreira, Luís, Goran Putnik, Maria Manuela Cruz-Cunha, Zlata Putnik, Hélio Castro, Catia Alves, and Vaibhav Shah. 2014. "Dashboard Services for Pragmatics-Based Interoperability in Cloud and Ubiquitous Manufacturing." *International Journal of Web Portals* 6(1):35–49.
- Ferreira, Luís, Goran Putnik, Maria Manuela Cruz Cunha, Zlata Putnik, Hélio Castro, Catia Alves, Vaibhav Shah, and Leonilde Varela. 2017. "A Cloud-Based Architecture with Embedded Pragmatics Renderer for Ubiquitous and Cloud Manufacturing." *International Journal of Computer Integrated Manufacturing* 30(4–5):483–500.
- Fezari, Mohamed and Ali Al Dahoud. 2018. "Integrated Development Environment ' IDE ' For Arduino." *ResearchGate* (October):1–12.
- Firebeetle, Dfrobot, Firebeetle Esp, Iot Wifi, and Firebeetle Esp. n.d. "FireBeetle ESP8266 IOT Microcontroller (Supports Wi-Fi)." 2–5.
- Forrester, John, Åsa Gerger Swartling, and Kate Lonsdale. 2008. "Stakeholder Engagement and the Work of SEI: An Empirical Study." *Environment* (May 2014).
- Foundation, Eclipse. 2012. "Eclipse Logos and Artwork." Retrieved (<https://www.eclipse.org/org/artwork/>).
- Foundation, Raspberry Pi. 2020. "Raspberry Pi 400." Retrieved (<https://www.raspberrypi.org/products/raspberry-pi-400/>).
- Gao, Shang and John Krogstie. 2010. "A Repository Architecture for Business Process

- Characterizing Models.” *Lecture Notes in Business Information Processing* 68 LNBIP(Idi):162–76.
- Gap, Contact. 2021. “D3V.” 21(8):1–13.
- Greco, Marco, Giorgio Locatelli, and Stefano Lisi. 2017. “Open Innovation in the Power & Energy Sector: Bringing Together Government Policies, Companies’ Interests, and Academic Essence.” *Energy Policy* 104(January):316–24.
- Griffor, Edward R., Chris Greer, David A. Wollman, and Martin J. Burns. 2017. “Framework for Cyber-Physical Systems: Volume 1, Overview.” *Special Publication (NIST SP)-1500-201*. 2017. 1(May):266.
- Guan, Xinping, Bo Yang, Cailian Chen, Wenbin Dai, and Yiyin Wang. 2016. “A Comprehensive Overview of Cyber-Physical Systems: From Perspective of Feedback System.” *IEEE/CAA Journal of Automatica Sinica* 3(1):1–14.
- Gujar, Yogita Shinde, Ruksar Madaki, and Smina Nadaf. 2019. “IRJET- IoT Based 3D Printer.” *IRJET Journal*.
- Hang, Christine, Panagiotis Manolios, and Vasilis Papavasileiou. 2011. “Synthesizing Cyber-Physical Architectural Models with Real-Time Constraints.” *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6806 LNCS:441–56.
- Haupt, Florian, Markus Fischer, Dimka Karastoyanova, Frank Leymann, and Karolina Vukojevic-Haupt. 2014. “Service Composition for REST.” *Proceedings . IEEE 18th International Enterprise Distributed Object Computing Conference 2014- Decem(December)*:110–19.
- Hellerstein, Joseph M., Michael Stonebraker, and James Hamilton. 2007. “Architecture of a Database System.” *Foundations and Trends in Databases* 1(2):141–259.
- Hield, Christopher. 2012. “Object-Oriented Software Development.” 1–33.
- Hu, Fei, Qi Hao, Qingquan Sun, Xiaojun Cao, Rui Ma, Ting Zhang, Yogendra Patil, and Jiang Lu. 2017. “Cyberphysical System with Virtual Reality for Intelligent Motion Recognition and Training.” *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47(2):347–63.
- Huizingh, Eelko K. R. E. 2011. “Open Innovation: State of the Art and Future Perspectives.” *Technovation* 31(1):2–9.
- igus. 2021a. “Braços Robóticos Completos.” Retrieved (<https://www.igus.pt/roboLink/robot-arms>).
- igus. 2021b. “RoboLink® RL-DP - Módulo Completo de 5 Graus de Liberdade.” Retrieved (<https://www.igus.pt/product/20239>).
- Java Tutorial. 2018. “Choose Your JAVA IDE: Eclipse, NetBeans, and IntelliJ IDEA.” Retrieved (<https://javatutorial.net/choose-your-java-ide-eclipse-netbeans-and-intellij-idea>).
- Jha, Susmit, Sumit Gulwani, Sanjit A. Seshia, and Ashish Tiwari. 2010. “Synthesizing Switching Logic for Safety and Dwell-Time Requirements.” *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems, ICCPS ’10* 22–31.

- Jiang, Zengqiang, Yang Jin, E. Mingcheng, and Qi Li. 2017. "Distributed Dynamic Scheduling for Cyber-Physical Production Systems Based on a Multi-Agent System." *IEEE Access* 6:1855–69.
- João Ricardo Cardoso, Marcelo Götz, and João Netto. 2018. "Implantação De Uma Arquitetura De Software Baseada Em Tecnologias E Conceitos Da Indústria 4.0." *Proceedings XXII Congresso Brasileiro de Automática*.
- Johnson, Taylor T., Stanley Bak, Marco Caccamo, and Lui Sha. 2016. "Real-Time Reachability for Verified Simplex Design." *ACM Transactions on Embedded Computing Systems* 15(2):1–11.
- Kantamara, Pornkasem and Vichita Vathanophas Ractham. 2014. "SINGLE-LOOP vs . DOUBLE-LOOP LEARNING : AN OBSTACLE OR A SUCCESS FACTOR FOR ORGANIZATIONAL." *International Journal of Education and Research* 2(7):55–62.
- Khaitan, Siddhartha Kumar and James D. McCalley. 2015. "Design Techniques and Applications of Cyberphysical Systems: A Survey." *IEEE Systems Journal* 9(2):350–65.
- Khalid, Che Mohd Lukman, Mohamad Syazli Fathi, and Zainai Mohamed. 2015. "Integration of Cyber-Physical Systems Technology with Augmented Reality in the Pre-Construction Stage." *Proceedings of 2014 2nd International Conference on Technology, Informatics, Management, Engineering and Environment, TIME-E 2014* 151–56.
- Kim, Taeyang and Dong Hee Shin. 2016. "Social Platform Innovation of Open Source Hardware in South Korea." *Telematics and Informatics* 33(1):217–26.
- Klaassen, R., L. Evers, and P. Troxler. 2018. "The Generative Bedrock of Open Design OPEN DESIGN."
- Kodali, Ravi Kishore and Kopulwar Shishir Mahesh. 2016. "A Low Cost Implementation of MQTT Using ESP8266." *Proceedings of the 2016 2nd International Conference on Contemporary Computing and Informatics, IC3I 2016* (October):404–8.
- Kong, Linghe, Dawei Jiang, and Min You Wu. 2010. "Optimizing the Spatio-Temporal Distribution of Cyber-Physical Systems for Environment Abstraction." *Proceedings - International Conference on Distributed Computing Systems* (January):179–88.
- Koslow, Tyler. 2018. "Anet A8 Review: Most Popular 3D Printer in 2018." Retrieved (<https://all3dp.com/1/anet-a8-3d-printer-review-diy-kit/>).
- Kumar, Puranam Revanth. 2018. "Position Control of a Stepper Motor Using Labview." *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, RTEICT 2018 - Proceedings* (May 2018):1551–54.
- Lameche, Khaled, Najib M. Najid, Pierre Castagna, and Khalid Kouiss. 2017. "Modularity in the Design of Reconfigurable Manufacturing Systems." *IFAC-PapersOnLine* 50(1):3511–16.
- Laudon, Kenneth C. 2003. *Management Information Systems: Managing the Digital Firm*. Vol. 7.
- Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning." *Nature*

- 521(7553):436–44.
- Lee, E. A., & Seshia, S. A. 2017. *Introduction to Embedded Systems. A Cyber-Physical Systems Approach. Second Edition*. Vol. 195.
- Lee, Edward. 2008. “Cyber Physical Systems: Design Challenges.”
- Lee, Jay, Behrad Bagheri, and Hung An Kao. 2015. “A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems.” *Manufacturing Letters* 3(December):18–23.
- Lenz, Christian. 2017. “NetBeans Logo.” Retrieved (<https://cwiki.apache.org/confluence/display/NETBEANS/NetBeans+Logo>).
- Lin, Wen Hui and Ya Pu Zhao. 2008. “Pull-in Instability of Micro-Switch Actuators: Model Review.” *International Journal of Nonlinear Sciences and Numerical Simulation* 9(2):175–83.
- Luo, Fengji, Junhua Zhao, Zhao Yang Dong, Yingying Chen, Yan Xu, Xin Zhang, and Kit Po Wong. 2016. “Cloud-Based Information Infrastructure for Next-Generation Power Grid: Conception, Architecture, and Applications.” *IEEE Transactions on Smart Grid* 7(4):1896–1912.
- Majumder, Sumit, N. E. McGruer, George G. Adams, P. M. Zavracky, Richard H. Morrison, and Jacqueline Krim. 2001. “Study of Contacts in an Electrostatically Actuated Microswitch.” *Sensors and Actuators, A: Physical* 93(1):19–26.
- Mauser.pt. 2017. “Auscultadores Headset c/ Microfone e Jack 3.5mm 4pin - Velleman - Velleman.” Retrieved (https://mauser.pt/catalog/product_info.php?cPath=145_346_3024&products_id=091-0172).
- Mauser.pt. 2018. “Módulo ESP32 Com Câmara OV2640 - ESP32-CAM.” Retrieved (https://mauser.pt/catalog/product_info.php?cPath=1667_2889_2619&products_id=096-7762).
- Mauser.pt. 2021. “Microswitch Com Patilha 12mm e Rolamento (IP40) SPDT 250VAC 10A.” Retrieved (https://mauser.pt/catalog/product_info.php?cPath=324_1401_707&products_id=010-0401).
- Mellis, Da. 2011. “Arduino Mega 2560.” Retrieved November (cc):2560.
- Miller, Bailey, Frank Vahick, and Tony Givargis. 2011. “Application-Specific Codesign Platform Generation for Digital Mockups in Cyber-Physical Systems.” *2011 Electronic System Level Synthesis Conference, ESLsyn 2011*.
- MillRight. 2017. “MillRight CNC Model M3 Specifications.” 850.
- Mohamed, S. B., M. Minhat, T. M. S. T. Sulaiman, M. S. M. Kasim, A. S. Mohamed, A. R. Mohamed, and J. Ismail. 2019. “Integrated Interface System for Tool Path Generation of STEP File.” *AIP Conference Proceedings* 2129(July).
- MT360. 2020. “CNC 3018 (Pro) Buyer’s Guide.” Retrieved (<https://www.mt360conference.com/cnc-3018-pro-buyers-guide/>).
- Muşlu, Kvanc, Yuriy Brun, Reid Holmes, Michael D. Ernst, and David Notkin. 2012.

- "Speculative Analysis of Integrated Development Environment Recommendations." *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications, OOPSLA* 669–82.
- Mutiara, A. B., R. Refianti, and B. A. Witono. 2014. "Developing a SAAS-Cloud Integrated Development Environment (IDE) for C, C++, And Java." *Journal of Theoretical and Applied Information Technology* 68(1):156–64.
- Myatt, Adam. 2007. *Pro NetBeans™ IDE 5.5 Enterprise Edition*.
- Nagappan, Mei and Achyudh Ram Keshav Ram Aswin Vayiravan. 2015. "Client-Server Architecture."
- Niz, Dionisio De. 2010. "Avicps 2010." *Integration The Vlsi Journal*.
- Oluwatosin, Haroon Shakirat. 2014. "Client-Server Model." *IOSR Journal of Computer Engineering* 16(1):57–71.
- Otto, Boris, Verena Ebner, Ehsan Baghi, and Ran M. Bittmann. 2013. "Toward a Business Model Reference for Interoperability Services." *Computers in Industry* 64(8):887–97.
- PCDIGA. 2021. "Monitor Asus VS197DE TN 18.5" FWXGA 16:9 60Hz." Retrieved (https://www.pcdiga.com/monitor-asus-vs197de-tn-18-5-fwxga-16-9-60hz-90lmf1001t02201c-?gclid=CjwKCAjw1JeJBhB9EiwAV612y1WqHeYQ9kLEbs46KPRxoqZR-vbc4Yh57bOmlL4EwoooCXP9XDzZhoCeKwQAvD_BwE).
- Pearce, Joshua M. 2013. *Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs*. Newnes.
- Peele, Bryan N., Thomas J. Wallin, Huichan Zhao, and Robert F. Shepherd. 2015. "3D Printing Antagonistic Systems of Artificial Muscle Using Projection Stereolithography." *Bioinspiration and Biomimetics* 10(5).
- Pi, N. E. W. Raspberry. 2020. "New! Raspberry Pi 400." (100).
- Pontika, Nancy, Petr Knoth, Matteo Cancellieri, and Samuel Pearce. 2015. "Fostering Open Science to Research Using a Taxonomy and an Elearning Portal." *ACM International Conference Proceeding Series* 21-22-Octo.
- Properties, Mucoadhesive and Human Oral Mucosa. 2015. "Technical Report : Technical Report : " 2–3.
- Putnik, Goran D. and Luis Ferreira. 2019. "What Is a Cyber-Physical System : Definitions and Models Spectrum." 663–74.
- Putnik, Goran D., Luis Ferreira, Nuno Lopes, and Zlata Putnik. 2019. "What Is a Cyber-Physical System: Definitions and Models Spectrum." *FME Transactions* 47(4):663–74.
- Rajkumar, Ragunathan, Insup Lee, Lui Sha, and John Stankovic. 2010. "Cyber-Physical Systems: The next Computing Revolution." *Proceedings - Design Automation Conference* 731–36.
- Ramião, André Lagarto. 2017. "Sistema Ciber-Físico de Produção Modular Usando Raspberry Pi." Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

- Ramião, André Lagarto, José Oliveira, and André Rocha. 2017. "Sistema Ciber-Físico de Produção Modular Usando Raspberry Pi."
- Rausch, Thomas. 2018. "Osmotic Message- Oriented Middleware for the Internet of Things." (April):17–25.
- Rebensdorf, A., A. Gergert, G. A. Oosthuizen, and S. Böhm. 2015. "Open Community Manufacturing - Development Challenge as a Concept for Value Creation for Sustainable Manufacturing in South Africa." *Procedia CIRP* 26:167–72.
- Ribeiro, Luis and José Barata. 2013. "Self-Organizing Multiagent Mechatronic Systems in Perspective." Pp. 392–97 in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*. IEEE.
- Ribeiro, Luis, José Barata, Dean Whittingslow, and Roland Krain. 2013. "Multiagent Mechatronic Systems with Simulation on the Loop." Pp. 3842–47 in *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE.
- Robotics, Dorna. 2021. "Dorna 2." Retrieved (<https://dorna.ai/robot/dorna-2/>).
- Rosen, Charles A. and David Nitzan. 1977. "Use of Sensors in Programmable Automation." *Computer* 10(12):12–23.
- SainSmart. 2020. "SainSmart Genmitsu CNC Router 3018-PRO DIY Kit." Retrieved (<https://www.sainsmart.com/products/sainsmart-genmitsu-cnc-router-3018-pro-diy-kit>).
- Shcherbakov, Maxim V., Artem V. Glotov, and Sergey V. Cheremisinov. 2020. *Proactive and Predictive Maintenance of Cyber-Physical Systems*. Vol. 259.
- Shi, Weisong, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. "Edge Computing: Vision and Challenges." *IEEE Internet of Things Journal* 3(5):637–46.
- Siyad, Abdulaziz Ahmed. 2019. "Design and Development of IP Based Automatic Bottle Inspection System Controllino Maxi Automation." Faculty of Engineering, University of Malaya.
- Solectro. 2020. "Motor PAP (Passo a Passo) Nema 23 19,3kg 57HS762804." Retrieved (<https://solectroshop.com/pt/motores-de-estepe-para-impressoras-3d/619-motor-pap-nema-23-193kg-57hs762804.html>).
- Sousa, Sérgio, E. Nunes, and I. Lopes. 2015. "Measuring and Managing Operational Risk in Industrial Processes." *FME Transactions* 43(4):295–302.
- Stankovic, John A. 2014. "Research Directions for the Internet of Things." *Ieee Internet of Things Journal* 1(1):3–9.
- Sulzbach, Bernardo. 2018. "An SQL Select Statement against a Publicly Available Teaching Database." Retrieved (https://commons.wikimedia.org/wiki/File:DVD_Rental_Query.png).
- Sumathi, S. and S. Esakkirajan. 2007. "Structured Query Language." *Studies in Computational Intelligence* 47:111–212.
- Szilvsi-Nagy, Márta and G. Y. Matyasi. 2003. "Analysis of STL Files." *Mathematical and Computer Modelling* 38(7–9):945–60.

- Tan, Ying, Mehmet C. Vuran, and Steve Goddard. 2009. "Spatio-Temporal Event Model for Cyber-Physical Systems." 44–50.
- Tansley, R., M. Bass, D. Stuve, M. Branschovsky, D. Chudnov, G. McClellan, and M. Smith. 2003. "The DSpace Institutional Digital Repository System: Current Functionality." *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries* 2003-Janua:87–97.
- Team, R. C. 2014. "Scholar (3)."
- Technical, Technische Dokumentation. n.d. "DP_version_final."
- Tomiyaama, Tetsuo and Florian Moyen. 2018. "Resilient Architecture for Cyber-Physical Production Systems." *CIRP Annals* 67(1):161–64.
- Tooze, James, Sharon Baurley, Robert Phillips, Paul Smith, Edwin Foote, and Sarah Silve. 2014. "Open Design: Contributions, Solutions, Processes and Projects." *Design Journal* 17(4):538–59.
- Trappey, Amy J. C., Charles V. Trappey, Usharani Hareesh Govindarajan, John J. Sun, and Allen C. Chuang. 2016. "A Review of Technology Standards and Patent Portfolios for Enabling Cyber-Physical Systems in Advanced Manufacturing." *IEEE Access* 4:7356–82.
- Tsai, Wei Tek, Xin Sun, and Janaka Balasooriya. 2010. "Service-Oriented Cloud Computing Architecture." *ITNG2010 - 7th International Conference on Information Technology: New Generations* 684–89.
- Vallance, R., S. Kiani, and S. Nayfeh. 2001. "Open Design of Manufacturing Equipment." ... *on Agile, Reconfigurable Manufacturing*, ... (January 2001):1–12.
- Velleman. 2016. "EWENT - HEADSET WITH MICROPHONE." Retrieved (<https://www.velleman.eu/products/view?id=406558&country=us&lang=pt>).
- Velleman. 2021. "MICROSWITCH 5A, LONG LEVER." Retrieved (<https://www.velleman.eu/products/view?id=12225>).
- Vieira, Gustavo, Jose Barbosa, Paulo Leita, and Lucas Sakurada. 2020. "Low-Cost Industrial Controller Based on the Raspberry Pi Platform." *Proceedings of the IEEE International Conference on Industrial Technology* 2020-Febru:292–97.
- Wahid, Hairunnizam, Sanep Ahmad, Mohd Ali Mohd Nor, and Maryam Abd Rashid. 2017. "Eclipse IDE." *Jurnal Ekonomi Malaysia* 51(2):39–54.
- Wang, Chengen, Zhuming Bi, and Li Da Xu. 2014. "IoT and Cloud Computing in Automation of Assembly Modeling Systems." *IEEE Transactions on Industrial Informatics* 10(2):1426–34.
- Wang, Shiyong, Jiafu Wan, Di Li, and Chunhua Zhang. 2016. "Implementing Smart Factory of Industrie 4.0: An Outlook." *International Journal of Distributed Sensor Networks* 2016.
- Wang, Yuanbin, Yuan Lin, Ray Y. Zhong, and Xun Xu. 2019. *IoT-Enabled Cloud-Based Additive Manufacturing Platform to Support Rapid Product Development*. Vol. 57.
- Wegner, Peter. 1996. "Interoperability." *ACM Computing Surveys* 28(1):285–87.

- Wielenga, Geertjan. 2016. "Lightweight & Fast NetBeans IDE."
- Wifi, Uno. n.d. "UNO WIFI REV2 Add WiFi to Your Devices with the Arduino Uno WiFi . It ' s Basically an Arduino Uno Rev3 with More Kic . " 8–10.
- Workshop, DroneBot. 2020. "Getting Started with the ESP32-CAM." Retrieved (https://dronebotworkshop.com/esp32-cam-intro/#ESP32-CAM_Specifications).
- Xu, Li Da, Eric L. Xu, and Ling Li. 2018. "Industry 4.0: State of the Art and Future Trends." *International Journal of Production Research* 56(8):2941–62.
- Xu, Xun. 2012. "From Cloud Computing to Cloud Manufacturing." *Robotics and Computer-Integrated Manufacturing* 28(1):75–86.
- Yen, Alex, Bryse Flowers, Wenshan Luo, Nitish Nagesh, Peter Tueller, Ryan Kastner, and Pat Pannuto. 2021. "A UCSD View on Replication and Reproducibility for CPS & IoT." *CPS-IoTBench 2021 - Proceedings of the 2021 Benchmarking Cyber-Physical Systems and Internet of Things* 20–25.
- Yuhan, Wang, Hu Jun, and Li Ye. 2003. "Study on a Reconfigurable Model of an Open CNC Kernel." *Journal of Materials Processing Technology* 138(1–3):472–74.

Apêndices

7.1	Apêndice A – Artigo intitulado “Cyber-Physical Systems using Open Design: an approach towards an Open Science Lab for Manufacturing” (no âmbito da conferência internacional CENTERIS - International Conference on ENTERprise Information Systems)	131
7.2	Apêndice B - Artigo intitulado: “Open Science Laboratory for Manufacturing: an education tool to contribute to sustainability” (no âmbito da conferência internacional ACM)	138
7.3	Apêndice C – Artigo intitulado “Sistemas Ciber-Físicos recorrendo ao Open Design: uma abordagem a um Laboratório de Ciência Aberta destinado à Produção” (no âmbito do Encontro Virtual de Engenharia Gestão Industrial (EVEGI) 2021)	144

7 Apêndices

7.1 Apêndice A – Artigo intitulado “*Cyber-Physical Systems using Open Design: an approach towards an Open Science Lab for Manufacturing*” (no âmbito da conferência internacional CENTERIS - International Conference on ENTERprise Information Systems)

CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2021

Cyber-Physical Systems using Open Design: an approach towards an Open Science Lab for Manufacturing

Hélio Castro^{a, b1}, Nuno Pinto^a, Filipe Pereira^a, Luís Ferreira^c, Paulo Ávila^{a, b}, João Bastos^{a, b}, Goran D. Putnik^d, Manuela Cruz-Cunha^c

^aSchool of Engineering, Polytechnic of Porto, Rua Dr. António Bernardino de Almeida, 431, 4249-015 Porto, Portugal

^bINESC TEC, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal

^c2Ai - Applied Artificial Intelligence Lab, School of Technology, IPCA, Barcelos, Portugal

^dUniversity of Minho, School of Engineering, Campus de Azurém, 4800-058 Guimarães, Portugal

Abstract

Cyber-Physical Systems (CPS) are fundamental in Industry 4.0 and manufacturing research communities, and research and technologies organizations (RTO) are keen to implement CPS laboratories. Due to the strong investment in these laboratories, it is unaffordable the investment in CPS laboratories. This paper proposes a CPS architecture based on Open Design (an open-source approach for hardware and software), and the implementation of the laboratory created as an *open-source* and truly *low-cost* solution, designated as Open Science Lab for Manufacturing (OSLab4Man), addressing the reproducibility and replicability (R&R) practices for manufacturing research laboratories. An *open-source* and truly *low-cost* solution of the OSLab4Man and the bill of materials (BOM) of the physical components and some alternative components *open-source* and/or *low-cost* is presented, that enables a ubiquitous manufacturing system.

© 2021 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer-review under responsibility of the scientific committee of the CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2021

Keywords: Cyber-Physical System (CPS); Open Design; Open Science; Open Lab; Manufacturing.

1. Introduction

There is a broad consensus that Cyber-Physical System (CPS) is a critical element in Industry 4.0 (Putnik et al. 2019). One of the major challenges with CPS is the integration of physical equipment processes and computational processes (Lee 2008). Industry 4.0 requires Smart Factories and to achieve

* Corresponding author. Tel.: +351 +351 22 83 40 500; fax: +351 22 83 21 159.
E-mail address: hcc@isep.ipp.pt

this state of evolution it is required to infrastructure support as Smart hardware and software (Wang et al. 2016).

Due to the increase of the engineering problems complexity, it is required a greater need to have research teams and work with multiple physical and computational systems (Xu, Xu, and Li 2018). Dynamic and integrated manufacturing systems are necessary to predict, react and align within complex and turbulent surroundings, and to support an evolutionary system Intelligent CPSs (Putnik et al. 2019) are mandatory.

Many research and technology organizations (RTOs), such as universities, applied universities and research entities, are financially supported by state subsidies and public contracts and, due to the turbulent environment, there is a trend of shrinking budgets (Coccia 2019). RTOs that study in the CPS field are developing many architectural infrastructures and it is essential for the concept proof to verify and validate these infrastructures with testbeds in research laboratories for industrial environment and, consequently, industrialization. Research laboratories may represent heavy investment (Pearce 2013), and, in CPS research, hardware and software infrastructures and services could mean an onerous financial challenge.

To overcome research challenges Open Science (OS) is gaining importance among research communities. According to (European Commission 2016), OS is “a new approach to the scientific process based on cooperative work and new ways of diffusing knowledge by using digital technologies and new collaborative tools”. The trendy concept of open approach (Castro, Putnik, Castro, and Dal Bosco Fontana 2019) is a core principle in OS and multiple variations and ramifications are address and comprises within OS. In Figure 1, an OS taxonomy proposal is presented to better represent the variations and ramifications of OS.

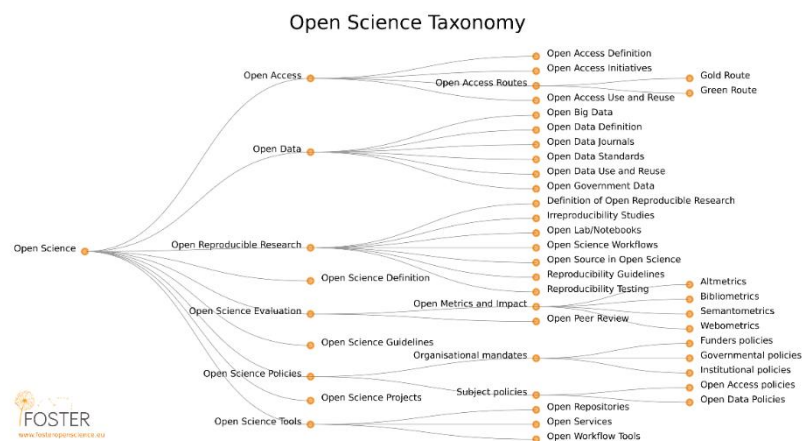


Figure 97 – Open Science Taxonomy (Pontika et al. 2015).

To significantly reduce the high investment in laboratories there is a need to resort to Open Lab (Pearce 2013) by RTOs, in which Open Design (OD) concept must be adopted. Following the open approach, such as OS, OD addresses software and hardware design development based on open source, freely and digital available, and supported by communities of members with common interests. Supported in digital repositories or platforms, OD is considered to be community-based generative process with fast-growing (Castro, Putnik, Castro, and Fontana 2019). According to Yen et al (2021), many communities are oriented to develop research projects adopting reproducibility and replicability (R&R) practices and could be considered as OD a design method for R&R practices.

This work proposes an approach of an Open Science Lab for Manufacturing (OSLab4Man) addressing a R&R and, for this reason, open source and low-cost CPS supported by OD. Relating the OS taxonomy, in Figure 1, the OSLab4Man in this work addresses the Open Reproducible Research branch and the following sub-branches of OS: Open Lab/Notebooks, Open Source in Open Science, and Reproducibility Guidelines.

Section 2 introduces a CPS architecture adopting open-source and low-cost approaches for hardware and software and section 3 proposes a dedicated laboratory using GitHub and low-cost resources, designated Open Science Lab for Manufacturing (OSLab4Man). Finally, section 4 discusses an example of an OSLab4Man implementation.

2. Proposal of a Cyber-Physical System (CPS) architecture supported by Open Design (OD)

The main goal is to create a R&R manufacturing laboratory oriented to the research community, adopting OD practices, whose production dynamics are controlled by a Cyber Physical System (CPS), called Open Science Lab for Manufacturing (OSLab4Man). The proposed CPS architecture aims to be social, scalable, replicable, and with remote and open access to the infrastructures, capable to be considerate an exponential technology and leading to true ubiquitous system. Within this architecture OD repository of digital products are available, in which the OSLab4Man is also available, to customize and replicate these products. It is emphasized that all the resources and services included in the system, whether at a physical or computational/digital level, meet the *open-source* and/or *low-cost* premise (Vieira et al. 2020) and addresses *Cloud Manufacturing* (Xu 2012), the *Industrial Internet of Things (IIoT 4.0)* (Boyes et al. 2018) and the *Smart Factory* (Chen et al. 2017).

Members of a community are allowed to free access to a repository where a wide range of product is available, which are provided by the members of the community itself. To make the processing of congruent information achievable without limitations, most of it is hosted in the cloud, process best known as *Cloud Computing* (Díaz, Martín, and Rubio 2016), which will be supported by a Multi-Agent System (Ramião 2017). Creating a connection bridge with the physical world, through a CPS interface composed by an low cost Operational Computer and an low-cost IoT device, which connects and controls physical resources (Ramião 2017), such as Additive and Subtractive Manufacturing.

A CPS interface communication infrastructure (under appropriate communications protocols (Al-Sarawi et al. 2017): BLE, NBIoT, others) allows the information to circulate between the cloud to the information and control (sub)system in the first layer of the “physical world”. CPS interface explores a local Industrial IoT (IIoT) infrastructure to spread and coordinate the previously processed information to the production layer, which is in charge of manufacturing the intended product, using open-source design manufacturing equipment (e.g., design robotic arm, 3D printer, CNC router and conveyor belt). This communication can be done indirectly, through a complementary layer, namely, the automation system.

After the information has reached its proper destination and the production system has started, the operator and community(ies) are able to monitor in real time all actions and operations performed by the physical resources, since the device is connected to them. For this to be feasible, the global system uses the communication and surveillance system, as it offers means that allow for total control and monitoring of the entire process surrounding the production, as well as the collection of data relating to it. In addition, this layer is also embedded in the application offered by the cloud, which allows the operator to communicate directly with the members of the community(ies), which, in turn, is also granted access to such monitoring and control.

At this level where the cyber computation is relevant towards cognitive capacity, in which a real-time collaborative environment integrates people, equipment and processes. A set of processes continuously receive data from the current manufacturing process (from each production unity), from the operator analysis and the expected metrics, and react towards the refinement of involved algorithms, the realignment (substitution, reconfiguration settings, etc. (Cunha et al. 2005) of involved equipment or any other production unity, as well as, if necessary, redesign or reengineering of proposed products. Event-driven programming for synchronous dynamics, Edge computing (Shi et al. 2016) on IoT devices and Message-Oriented Middleware (MOM) (Rausch 2018) for interoperability support, will sustain the required real-time and distributed behavior of such cyber computation level. The dynamic reconfiguration and integration between the physical and the digital world will be assured.

The Database layer corresponds to a database where community(ies) and members, as well as their respective data. Additionally, to provide an intelligent system, a copy of all the processed and generated code is also created and stored in the same information library. This means that it is not necessary to process all the information again in case of correspondence to a product previously executed by the system.

Like the Database layer, the Management, Documentation and Security layer appears as a protective layer to all information and data that run in the system, ensuring that it is available, legible and true only

to those who are entitled to it (Laudon 2003). In addition, data recovery is made possible through a backup system.

In Figure 2 is shown where all the layers/subsystems mentioned are illustrated, as well as their high-level connections.

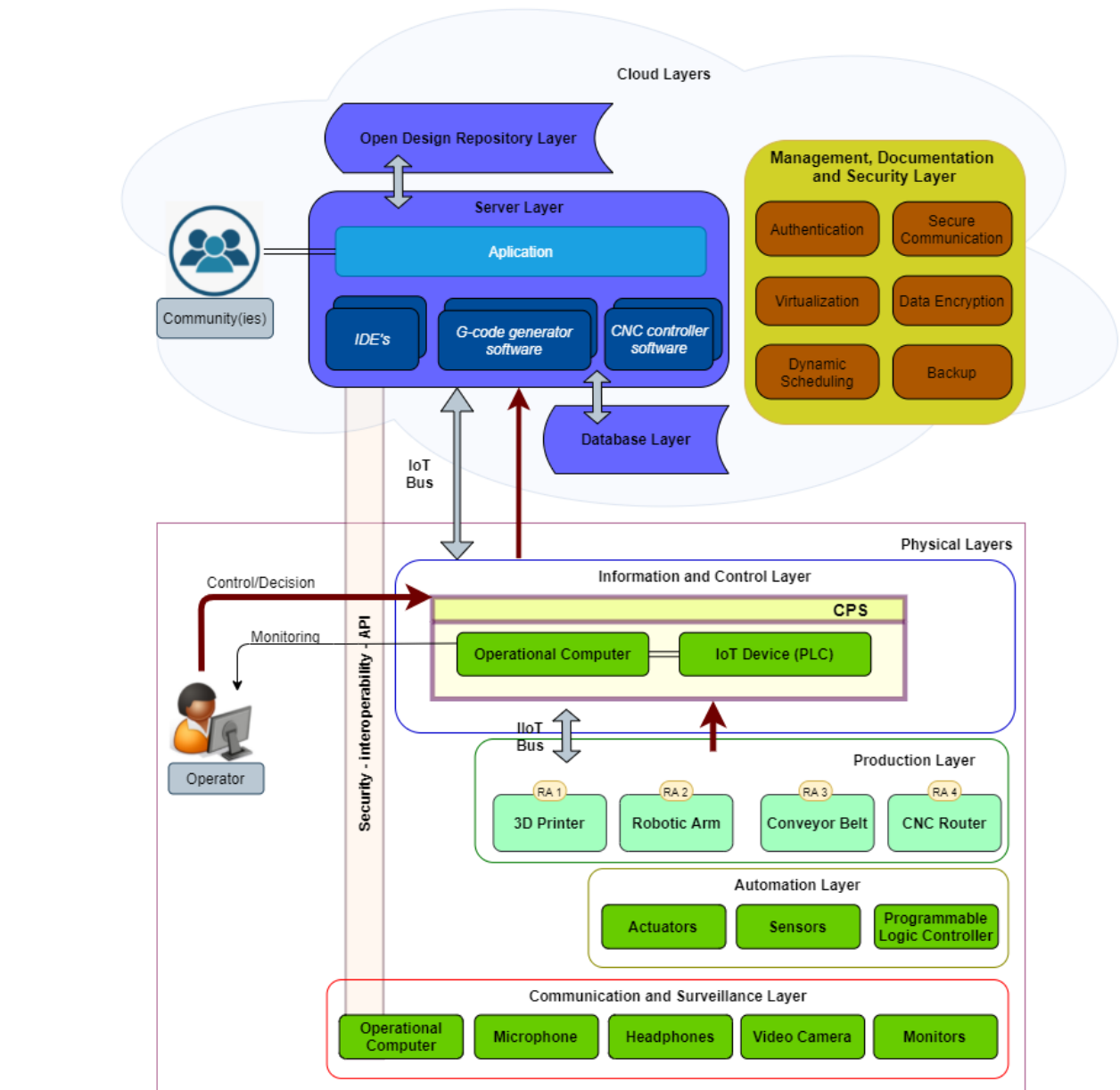


Figure 98 - Architecture of the proposed CPS supported by OD.

3. Implementation of Open Science Lab for Manufacturing (OSLab4Man)

The Open Science Lab for Manufacturing (OSLab4Man) implementation is also established by layers, which are equally distinguished between computational and physical. Effectively, in the cyber world, GitHub is used as a repository, both at a documental level and at a source code level, given its wide variety of files accepted by it. The next layer is the servers, which have as *RunTime Environment (RTE)* the Apache HTTP Server. As a frontend, the open-source application Rocket.Chat is reused, as it enables communication and monitoring, still being added the function of selecting the desired product. As a backend, that is, the server-side code, there are IDE's that allow its elaboration. In fact, NetBeans is an IDE that enables the programming of a MAS, with the help of the *Java Agent DEvelopment framework (JADE)*. The other is the Arduino IDE, where code is written based on a library that provides many common procedures based on input and output signals. In addition, servers can also resort a set of G-code generation software, in case the file acquired in the repository is not yet in the desired format. This software is OctoPrint, in the case of 3D printing, and dxf2gcode, in the case of CNC. Finally, it only remains for the servers to request the services of the software in charge of communicating with the

physical resources, namely, OctoPrint for the 3D printer, Grbl for the CNC router and Pronterface for the robotic arm.

OSLab4Man physical system is centralized by a Raspberry Pi 400 and a Controllino, which, respectively, correspond to the information and control system. The first ensures the connection to the cloud and receives information from it, as well as enabling direct communication with the 3D printer. The second, besides being connected to the first, is basically an industrialized Arduino board, as it offers the security, robustness and reliability of an industrial-grade PLC (Siyad 2019). However, its main function is to communicate with and control the manufacturing system (Figure 3), which is composed of a BeeveryCreative B2X300 (3D printer), a BCN3D MOVEO (robotic arm), a 3018 PRO (CNC router) and a conveyor belt. From this system, the focus is on the specified robotic arm, since it is an *open-source* and truly *low-cost* solution, in which its structure is fully printed using additive manufacturing technology, being "only" necessary to acquire the accessory material. Furthermore, its electronic components are controlled by Arduino (BCN3D Technologies 2016a).

Connected to the Raspberry Pi 400, there is also the communication and surveillance system. Effectively, it consists of two monitors and four ArduCAM video surveillance cameras, given the compatibility of its shield with Arduino boards and its quality/price ratio. As for headphones and microphone, these are at the discretion of each operator. It should also be noted that this layer of materials facilitates direct communication with the members of the community(ies), as well as the monitoring of production.

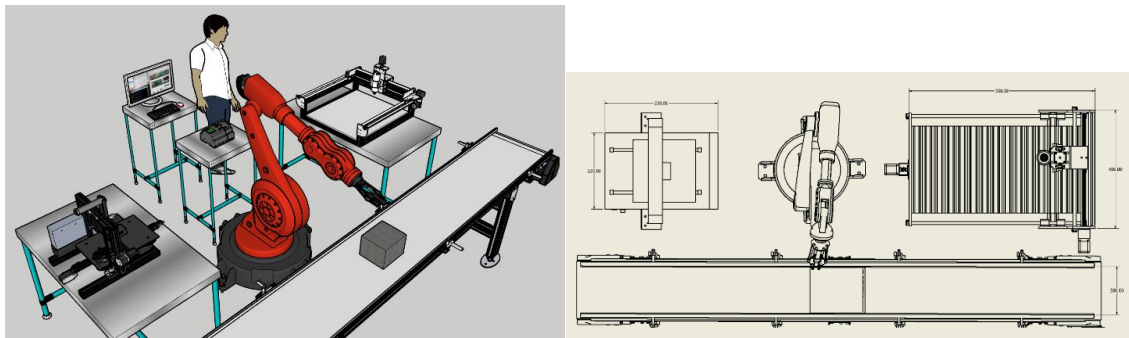


Figure 99 – Manufacturing System layout (3D, in left side, and 2D, in right side, models)

A flow sheet is represented in Figure 4 to better understand the hardware communication flow between the various components of the OSLab4Man physical system.

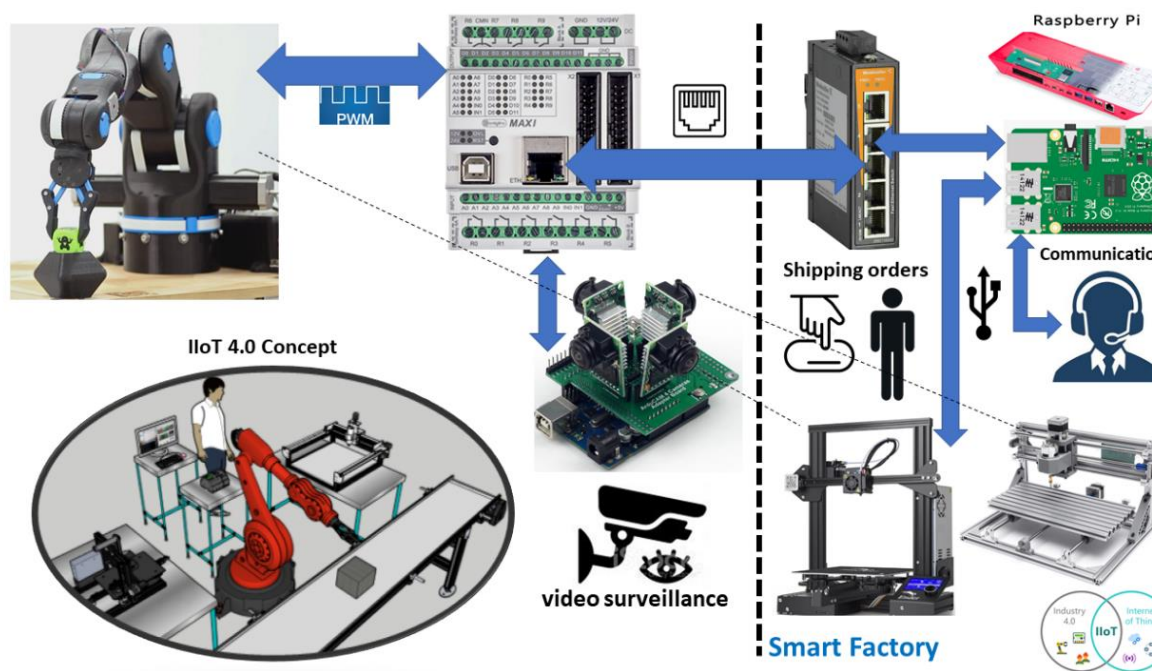


Figure 4 – Flow sheet of the OSLab4Man physical system.

In the following, it is provided a bill of materials (BOM) of the OSLab4Man physical components mentioned above and some alternative components also possible based on *open-source* and/or *low-cost* and listed in Table 1.

Table 41 – BOM of the OSLab4Man Physical System

Component	Component Model in OSLab4Man	Quantity	Some Alternative Open-Source and/or Low-Cost Component Model(s)
Operational Computer	Raspberry Pi 400	1	• Raspberry Pi 4 • Raspberry Pi 5 • Raspberry Pi 3 Model B
PLC	Controllino	1	• PLC Arduino Ardbox & DALI • OpenPLC
3D Printer	BeeveryCreative B2X300	1	• RepRap • Creality Ender 3 • Anet A8 • Original Prusa i3 MK3S
Robotic Arm	BCN3D MOVEO	1	• Dorna 2 • UFactory uArm • Zortrax Robotic Arm
CNC Router	3018	1	• rBotCNC • Carbide 3D Shapeoko 4 • Ooznest Workbee
Conveyor Belt	Customized from spare parts	1	--
Video Camera	ArduCAM (OV9782)	4	• ESP32-CAM • Waveshare OV9655 • SaintSmart Surveillance Arduino Camera
Headphones with Microphone	Tacens AH118 Headset	1	• 1Life Sound One Headset • Ngs Ms103

4. Conclusions

Research and technologies organizations (RTO) to keep at the forefront in their research area need to attract investment, especially from the public sector, to invest in state-of-the-art laboratories, namely a Cyber-Physical System (CPS) laboratory for those RTO that develop in manufacturing area. Due to the trend of less public investment in R&D and the need of dissemination of knowledge among research communities, Open Science (OS) is a new approach to support collaboration and dissemination among researchers, based on digital tools. In the specific case of laboratories, the branch of Open Reproducible Research of OS forwards to Open Lab as a key element to address the reproducibility and replicability (R&R) practices of research laboratories.

Considering the importance for the research on manufacturing field of CPS and, consequently, implementation of a dedicated laboratory for this subject, this paper proposes a CPS architecture based on Open Design (an open-source approach for hardware and software), and the implementation of the laboratory created as an *open-source* and truly *low-cost* solution, designated as Open Science Lab for Manufacturing (OSLab4Man). The CPS architecture is social (community-based), scalable, replicable, and with open access and remote control, aiming for a truly ubiquitous manufacturing system. All physical and computational/digital resources consider the *Cloud Manufacturing*, the *Industrial Internet of Things (IIoT 4.0)* and the *Smart Factory* and compress the *open-source* and/or *low-cost* premise for the implementation of the OSLab4Man.

Based on this work, an example of an OSLab4Man is being implemented. Firstly, the OSLab4Man physical system and then, the computational/digital system. Future works will present the results and new findings of the OSLab4Man implementation.

5. Acknowledgment

This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within project UIDB/50014/2020.

This work also had financial support of *Memória Plural – Associação* to purchase components for the presented laboratory and the authors acknowledge the Department of Mechanical Engineering of the School of Engineering, Polytechnic of Porto, for providing 3D printers to produce parts of the manufacturing equipment of the laboratory.

References

- Putnik GD, Ferreira L, Lopes N, Putnik Z. What is a Cyber-Physical System: Definitions and models spectrum. *FME Trans.* 2019;47(4):663–74.
- Lee EA. Cyber physical systems: Design challenges. *Proc - 11th IEEE Symp Object/Component/Service-Oriented Real-Time Distrib Comput ISORC 2008.* 2008;363–9.
- Wang S, Wan J, Li D, Zhang C. Implementing Smart Factory of Industrie 4.0: An Outlook. *Int J Distrib Sens Networks.* 2016;2016.
- Xu L Da, Xu EL, Li L. Industry 4.0: State of the art and future trends. *Int J Prod Res.* 2018;56(8):2941–62.
- Coccia M. Metabolism of Public Research Organizations: How Do Laboratories Consume State Subsidies? *Public Organ Rev.* 2019;19(4):473–91.
- Pearce JM. Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs. *Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs.* Elsevier Inc.; 2013. 1–271 p.
- European Commission. EU Open Innovation, Open Science, Open to the World [Internet]. European Commission. 2016. 108 p. Available from: http://europa.eu/rapid/press-release_SPEECH-15-5243_en.htm
- Castro H, Putnik G, Castro A, Dal Bosco Fontana R. Open design initiatives: An evaluation of CAD Open Source Software. In: *Procedia CIRP.* 2019. p. 1116–9.
- Pontika N, Knoth P, Cancellieri M, Pearce S. Fostering open science to research using a taxonomy and an elearning portal. *ACM Int Conf Proceeding Ser.* 2015;21-22-Octo.
- Castro H, Putnik G, Castro A, Dal Bosco Fontana R. Could open design learn from wikipedia? In: *Procedia CIRP.* 2019.
- Yen A, Flowers B, Luo W, Nagesh N, Tueller P, Kastner R, et al. A UCSD view on replication and reproducibility for CPS & IoT. *CPS-IoTBench 2021 - Proc 2021 Benchmarking Cyber-Physical Syst Internet Things.* 2021;20–5.
- Vieira G, Barbosa J, Leitao P, Sakurada L. Low-cost industrial controller based on the raspberry pi platform. *Proc IEEE Int Conf Ind Technol.* 2020;2020-Feb:292–7.
- Xu X. From cloud computing to cloud manufacturing. *Robot Comput Integr Manuf [Internet].* 2012 Feb [cited 2011 Sep 12];28(1):75–86. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S0736584511000949>
- Boyes H, Hallaq B, Cunningham J, Watson T. The industrial internet of things (IIoT): An analysis framework. *Comput Ind.* 2018;101(March):1–12.
- Chen B, Wan J, Shu L, Li P, Mukherjee M, Yin B. Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges. *IEEE Access.* 2017;6:6505–19.
- Díaz M, Martín C, Rubio B. State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *J Netw Comput Appl [Internet].* 2016 May;67:99–117. Available from: <https://linkinghub.elsevier.com/retrieve/pii/S108480451600028X>
- Ramião AL. Sistema Ciber-Físico de Produção Modular usando Raspberry Pi [Internet]. Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa; 2017. Available from: <https://www.rcaap.pt/detail.jsp?id=oai:run.unl.pt:10362/36905>
- Al-Sarawi S, Anbar M, Alieyan K, Alzubaidi M. Internet of Things (IoT) communication protocols: Review. *ICIT 2017 - 8th Int Conf Inf Technol Proc.* 2017;(October):685–90.
- Cunha MM, Putnik GD, Gunasekaran A, Avila P. Market of Resources as a Virtual Enterprise Integration Enabler. In: *Virtual Enterprise Integration [Internet].* IGI Global; 2005. p. 145–65. Available from: <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59904-891-8.ch007>
- Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* 2016;3(5):637–46.
- Rausch T. Osmotic Message- Oriented Middleware for the Internet of Things. 2018;(April):17–25.
- Caldas MP. Management information systems: managing the digital firm. Vol. 7, *Revista de Administração Contemporânea.* 2003. 223–223 p.
- Siyad AA. Design and development of IP based automatic bottle inspection system Controllino Maxi automation [Internet]. Faculty of Engineering, University of Malaya; 2019. Available from: <http://studentsrepo.um.edu.my/11449/6/aziz.pdf>
- BCN3D Technologies. BCN3D MOVEO: A fully Open Source 3D printed robot arm [Internet]. 2021 [cited 2021 Jul 2]. Available from: <https://www.bcn3d.com/bcn3d-moveo-the-future-of-learning-robotic-arm/>

7.2 Apêndice B - Artigo intitulado: “*Open Science Laboratory for Manufacturing: an education tool to contribute to sustainability*” (no âmbito da conferência internacional ACM)

Open Science Laboratory for Manufacturing: an education tool to contribute to sustainability

Hélio Castro*

School of Engineering, Polytechnic of Porto, and INESC-TEC, hcc@isep.ipp.pt

Nuno Pinto

School of Engineering, Polytechnic of Porto, 1150569@isep.ipp.pt

Filipe Pereira

School of Engineering, Polytechnic of Porto, fal@isep.ipp.pt

Luís Ferreira

Institute of Cávado e Ave, School of Technology, lufer@ipca.pt

Paulo Ávila

School of Engineering, Polytechnic of Porto, and INESC-TEC, psa@isep.ipp.pt

Goran Putnik

University of Minho, School of Engineering, putnikgd@dps.uminho.pt

Manuel Carlos Felgueiras

School of Engineering, Polytechnic of Porto, mcf@isep.ipp.pt

João Bastos

School of Engineering, Polytechnic of Porto, and INESC-TEC, jab@isep.ipp.pt

Manuela Cunha

Institute of Cávado e Ave, School of Technology, mcunha@ipca.pt

* Corresponding author.

ABSTRACT

This paper presents a Cyber Physical System (CPS) laboratory based on Open Design, called Open Science Laboratory for Manufacturing (OSLab4Man), with educational and research purpose and the potential contribute and (direct and indirect) effects regarding the 17 Sustainable Development Goals (SDGs). A short introduction describing the 17 SDG, the education for sustainability role, and the relevance of the framed CPS. A literature review of CPS laboratories that are addressed to sustainability is presented. The OSLab4Man is describe in the educational purpose lens. Then, the contribute of the OSLab4Man as an education tool for sustainability regarding the 17 SDGs is reflected. Finally, the conclusions of this work and future works are highlighted.

INTRODUCTION

Sustainability become a major global issue for political, economic, social and research reasons. World leaders aimed for a worldwide common framework and, in 2015, an important step was taken through the 17 Sustainable Development Goals (SDGs) of the 2030 Agenda for Sustainable Development (United Nations, 2015).

Education plays a central role in the individual, community, and social awareness towards sustainability. (Scheunpflug & Asbrand, 2006) enhances the relation between global education and education for sustainability.

Concerning manufacturing education and research, CPS is considered to be a critical issue for Industry 4.0 (Putnik et al., 2019), and a requirement system to deal with the increase of manufacturing and engineering problems complexity (Xu et al., 2018) and dynamic environment. Integration of physical equipment processes and computational processes is an important element within CPS (Lee, 2008) and, in the perspective of research work, laboratories for these manufacturing system represents heavy investments for educational and research organizations.

Considered to be a new approach to science (European Commission, 2016) and increasing relevance in the educational and research communities, Open Science (OS) aims to promote collaboration and educational/learning dissemination by using digital technologies and tools. One branch of the OS is the Open Lab, in which a laboratory is developed under Open Design (OD) concept in order to develop educational and research laboratories adopting reproducibility and replicability (R&R) practices, many educational and research communities are embracing these practices for CPS (Yen et al., 2021). OD delivers software and hardware design development in open-source and free available in digital community-based platforms.

An open-source and low-cost CPS supported by OD was developed to address a R&R laboratory for educational and research purposes called Open Science Lab for Manufacturing (OSLab4Man).

Section 2 presents a literature review of CPS laboratories addressing sustainability and section 3 describes the OSLab4Man in the educational perspective. Section 4 reflects the contribute of the OSLab4Man as an education tool for sustainability regarding the 17 SDGs. Finally, discussion of this work and future works is underlined in Section 5.

CPS ADDRESSING SUSTAINABILITY

A literature review regarding CPS addressing sustainability is presented as follows. Lin and Panahi (2010) studied a real-time service-oriented architecture (SOA) architecture to enhance sustainability and predictability in CPS. The purpose of this work was to prove that both SOA and CPS provide useful support for the sustainability of industrial systems, as the first has been adopted in a variety of industrial systems due to its integration flexibility and process composability, and the last is a new technology to bring computational intelligence to physical devices and to make them mission- and situation-aware.

Gupta et al. (2011) approached a model based engineering to verify CPS sustainability. Effectively, they discussed the various categories of the resource management algorithms to deconceptualize the energy consumption of both computing and non-computing components. Z. Wang et al. (2015) provided an overview of water CPS for sustainability from four critical aspects, consequent to the tremendous advancement in science and technology: sensing and instrumentation, communications and networking, computing, and control, and explore opportunities and design challenges of relevant techniques.

Gürdür and Gradin (2018) debated the inclusion of sustainability aspects while improving the interoperability of toolchains in CPS manufacturing. To do that, they resorted to an automobile

manufacturing process as a use case, as well as relevant sustainability metrics for each stage of the process. Life cycle sustainability assessment methodology is used to identify the sustainability metrics, and the use case is employed to exemplify how some of these metrics can be integrated with interoperable toolchains to illustrate the applicability of the approach. To respond on what will the implications of the Cyber-Physical Production Systems (CPPS) be on health, learning and operative performance of human workers, Pinzone et al. (2020) conceptualized a holistic framework and outlined a guideline to analyze how the functionalities of a CPPS relate to operational and social sustainability-related performance impacts at different levels of analysis. In doing so, the study offers first support to researchers and managers of manufacturing companies willing to define suitable operational and social sustainability-related performances for Human-centric Cyber-Physical Production Systems of the future.

The previous works contribute to sustainability but any of previous CPS is presented for educational purpose. The Table 1 offers an analysis relating the presented CPS and their contribution to the 17 SDGs.

Table 42 – CPS proposals contribution to the 17 SDGs

SDGs	CPS proposals				
	(Lin & Panahi, 2010)	(Gupta et al., 2011)	(Wang et al., 2015)	(Gürdür & Gradin, 2018)	(Pinzone et al., 2020)
	9. Industry, Innovation and Infrastructures; 12. Responsible Consumption and Production	9. Industry, Innovation and Infrastructures; 11. Sustainable Cities and Communities; 12. Responsible Consumption and Production	6. Clean Water and Sanitation; 9. Industry, Innovation and Infrastructures; 11. Sustainable Cities and Communities; 12. Responsible Consumption and Production	9. Industry, Innovation and Infrastructures; 12. Responsible Consumption and Production	9. Industry, Innovation and Infrastructures; 11. Sustainable Cities and Communities; 12. Responsible Consumption and Production

OSLab4Man AS AN EDUCATION TOOL

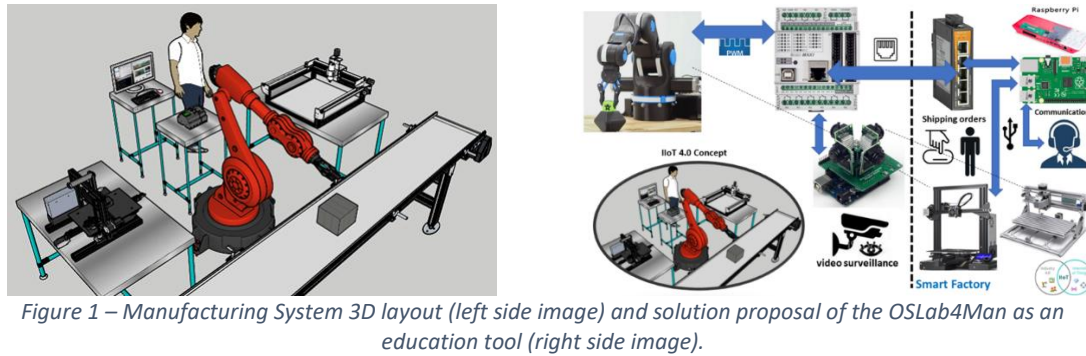
The Open Science Lab for Manufacturing (OSLab4Man) compiles two integrated elements, cyber/computational and physical. These elements create the overall Cyber Physical System (CPS) and both represents educational topics to develop and study.

In the cyber/computational elements requires using digital repositories for Open Design (OD), both hardware and software developed in open-source and free available, that support collaborative development and accept a wide variety of files for documentation and open-source code, such as GitHub. Then, knowledge in servers (signal processing, communication, G-Code generation, ...) is necessary and use of open-source frontend applications are needed to enable remote control, monitoring and communication when the manufacturing physical components is processing the production order.

Concerning the OSLab4Man physical element, it is centralized in two low-cost equipment that requires technological knowledge to integrate. The first low-cost equipment is a low-cost computer (Raspberry Pi 400) that connects with cloud services (Cloud Computing and Cloud Manufacturing) and enables communication with members of the OD community(ies)and authorized direct communication with the manufacturing equipment. And the second low-cost equipment is an industry-grade PLC with open-source software (Controllino) that connects to the low-cost computer and guarantees low-level communicate and automation processes with manufacturing system assuring security, robustness, and reliability.

The manufacturing system equipment studied for the OSLab4Man is composed by an open-source and low-cost 3D printer (BeeveryCreative B2X300), an open-source robotic arm (BCN3D MOVEO), a CNC router (3018 PRO) and a conveyor belt (customized and made by spare parts). For education purpose, the open-source robotic arm could be produced in the 3D printer and the respective assembly of components trains the automation and control subject.

In Figure 1 the manufacturing system layout in 3D and the OSLab4Man as an education tool solution proposal are presented.



Summarizing the concepts that OSLab4Man addresses for manufacturing educational and learning purposes is presented as follows:

Concepts in the Cyber element: CPS, Cloud Computing, Server architecture, Open-source applications.

Concepts in the Physical element: CPS, Internet of Things (IoT), Automation and Control, 3D printer skills.

Concepts in Manufacturing: Industry 4.0, Cloud Manufacturing, Industrial Internet of Things (IIoT 4.0), Smart Factory, Additive and Subtractive Manufacturing.

Organizational Concepts: Open-X (design, source, science, laboratory, ...), Low-cost equipment, Collaborative Tools and Platforms, Community-based design development, Remote hardware and software co-creation, Ubiquitous manufacturing system, Collaborative Engineering, Reproducibility and replicability (R&R).

CONTRIBUTION OF OSLab4Man TO THE 17 SDGs

Open Science Lab for Manufacturing (OSLab4Man) is CPS architecture laboratory, based on Open Design (an open-source approach for hardware and software), created as an open-source and truly low-cost solution. The OSLab4Man is social (community-based), scalable, replicable, and with open access and remote control, aiming for a truly ubiquitous manufacturing system. All physical and computational/digital resources consider the Cloud Manufacturing, the Industrial Internet of Things (IIoT 4.0) and the Smart Factory and compress the open-source and/or low-cost premise for the implementation of the OSLab4Man.

Due to the technological and, mainly, social dimension of the OSLab4Man in the educational perspective, direct and indirect effects are expected by the contribution of this laboratory in the 17 SDGs. For the direct effect contributions, the related SDGs are: 4. Quality Education, 5. Gender Equality, and 17. Partnerships for Goals. And for the indirect effect contributions, the related SDGs are: 1. No Poverty, 2. Zero Hunger, 6. Clean Water and Sanitation, 7. Affordable and Clean Energy, 8. Decent Work and Economic Growth, 9. Industry, Innovation and Infrastructures, 10. Reduce Inequalities, 11. Sustainable Cities and Communities, and 12. Responsible Consumption and Production. In Table 2 the effects descriptions are summarized.

Table 43 – Contribute of the OSLab4Man as an education tool for sustainability regarding the 17 SDGs.

OSLab4Man effects in the contribution to the 17 SDGs				
		Direct Effect(s)	Indirect Effect(s)	Effect(s) Description
SDGs	1 No Poverty		√	Potential development of micro and small-scale manufacturing units for agriculture and food products
	2 Zero Hunger		√	- Low-cost knowledge dissemination dissemination could lead society to better social conditions - Create opportunities (jobs, business, ...) to underdeveloped communities and societies

3 Good Health and Well-Being			---
4 Quality Education	✓		• Training through new and open technologies, equipment, software and systems • New methods of education (remote, networked, multicultural, ...) • Experiential learning • Affordable laboratories for education and research • Equitable education
5 Gender Equality	✓		• Education available to any person • Possibility to promote a gender policy for specific inclusion (girls, women, ...)
6 Clean Water and Sanitation		✓	• Potential development of micro and small-scale manufacturing units for water treatment and sanitation products
7 Affordable and Clean Energy		✓	• Potential development of micro and small-scale manufacturing units for renewable energy products
8 Decent Work and Economic Growth		✓	• Open-source design and low cost manufacturing unit for low-income entrepreneurs • Educational tool suitable in Learning For Life approach • New manufacturing and business models based on open design • Replicable manufacturing units for communities (urban and rural)
9 Industry, Innovation and Infrastructures		✓	• Replicable, scalable, and ubiquitous manufacturing systems • Open-source design and low cost manufacturing for entrepreneurs, RTO, and industries • Customized and digitally open products • Open-source and low-cost resources/services
10 Reduce Inequalities		✓	• Equitable education • Open-source and free products and services • Low-cost resources/services • Open-source design and low cost manufacturing unit for low-income entrepreneurs • Customized and digitally open products • Replicable manufacturing units for communities (urban and rural) • Young and elderly people can learn through this education tool
11 Sustainable Cities and Communities		✓	• Potential development of micro and small-scale urban manufacturing units • New manufacturing and business models based on open design • Replicable manufacturing units for communities (urban and rural)
12 Responsible Consumption and Production		✓	• Customized and digitally open products • Consumers and producers develop their products and services in a sustainable way • Digitalization of products, services, processes
13 Climate Action			---

	14 Life Below Water			---
	15 Life on Land			---
	16 Peace, Justice and Strong Institutions			---
	17 Partnerships for Goals	✓		- Global partnership among education institutions and organizations to promote and develop Open Design (hardware and software) solution for Open Science, such as Open Labs in relevant fields - Global partnership among education institutions and organizations to promote and develop Open Design products and services for specific fields (e.g., agriculture, food, water treatment and sanitation, renewable energy)

CONCLUSIONS

Global education and education for sustainability are important basis to mobilize and create individual, community, and social awareness towards sustainability, and contributes to the fulfillment of the 17 SDGs. The Open Science Lab for Manufacturing (OSLab4Man) is an open-source and low-cost CPS, supported by Open Design (OD) (for software and hardware), developed to address a reproducibility and replicability (R&R) laboratory for educational purposes. According to the literature review, some CPS works delivers a sustainable approach, and this work presents a CPS that is unique due to the educational perspective for sustainability.

The OSLab4Man is educational tool that is social (community-based), scalable, replicable, and with open access and remote control, aiming for a truly ubiquitous manufacturing system and addressing concepts in the cyber element, in the physical element, in manufacturing and organizational concepts.

Based on this paper and the educational approach of the OSLab4Man, specific projects for each SDGs must be developed and implemented.

REFERENCES

- European Commission. (2016). EU Open Innovation, Open Science, Open to the World. In European Commission. http://europa.eu/rapid/press-release_SPEECH-15-5243_en.htm
- Gupta, S. K. S., Mukherjee, T., Varsamopoulos, G., & Banerjee, A. (2011). Research directions in energy-sustainable cyber-physical systems. *Sustainable Computing: Informatics and Systems*, 1(1), 57–74. <https://doi.org/https://doi.org/10.1016/j.suscom.2010.10.003>
- Gürdür, D., & Gradin, K. T. (2018). Interoperable toolchains in cyber-physical systems with a sustainability perspective. 2017 IEEE Conference on Technologies for Sustainability, SusTech 2017, 2018-Janua(SusTech), 1–8. <https://doi.org/10.1109/SusTech.2017.8333471>
- Lee, E. A. (2008). Cyber physical systems: Design challenges. *Proceedings - 11th IEEE Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, ISORC 2008*, 363–369. <https://doi.org/10.1109/ISORC.2008.25>
- Lin, K.-J., & Panahi, M. (2010). A real-time service-oriented framework to support sustainable cyber-physical systems. 2010 8th IEEE International Conference on Industrial Informatics, 15–21.
- Pinzone, M., Albè, F., Orlandelli, D., Barletta, I., Berlin, C., Johansson, B., & Taisch, M. (2020). A framework for operative and social sustainability functionalities in Human-Centric Cyber-Physical Production Systems. *Computers & Industrial Engineering*, 139, 105132. <https://doi.org/https://doi.org/10.1016/j.cie.2018.03.028>
- Putnik, G. D., Ferreira, L., Lopes, N., & Putnik, Z. (2019). What is a Cyber-Physical System: Definitions and models spectrum. *FME Transactions*, 47(4), 663–674. <https://doi.org/10.5937/fmet1904663P>
- Scheunpflug, A., & Asbrand, B. (2006). Global education and education for sustainability. *Environmental Education Research*, 12(1), 33–46. <https://doi.org/10.1080/13504620500526446>
- United Nations. (2015). The Sustainable Development Agenda.
- Wang, Z., Song, H., Watkins, D. W., Ong, K. G., Xue, P., Yang, Q., & Shi, X. (2015). Cyber-physical systems for water sustainability: Challenges and opportunities. *IEEE Communications Magazine*, 53(5), 216–222. <https://doi.org/10.1109/MCOM.2015.7105668>
- Xu, L. Da, Xu, E. L., & Li, L. (2018). Industry 4.0: State of the art and future trends. *International Journal of Production Research*, 56(8), 2941–2962. <https://doi.org/10.1080/00207543.2018.1444806>
- Yen, A., Flowers, B., Luo, W., Nagesh, N., Tueller, P., Kastner, R., & Pannuto, P. (2021). A UCSD view on replication and reproducibility for CPS & IoT. CPS-IoTBench 2021 - Proceedings of the 2021 Benchmarking Cyber-Physical Systems and Internet of Things, 20–25. <https://doi.org/10.1145/3458473.3458821>

7.3 Apêndice C – Artigo intitulado “Sistemas Ciber-Físicos recorrendo ao *Open Design*: uma abordagem a um Laboratório de Ciência Aberta destinado à Produção” (no âmbito do Encontro Virtual de Engenharia Gestão Industrial (EVEGI) 2021)

Sistemas Ciber-Físicos recorrendo ao *Open Design*: uma abordagem a um Laboratório de Ciência Aberta destinado à Produção

Nuno Pinto; Hélio Castro; Filipe Pereira; Paulo Ávila; João Batos; Luís Ferreira

1150560@isep.ipp.pt; hcc@isep.ipp.pt; fal@isep.ipp.pt; psa@isep.ipp.pt; jab@ispe.ipp.pt;
lufer@ipca.pt

Introdução

É fortemente consensual que os sistemas ciberfísicos (CPS) são um elemento crítico na Indústria 4.0, no entanto, um dos seus maiores desafios é a integração de processos de equipamentos físicos e processos computacionais (Putnik and Ferreira 2019). Devido a isso, há uma maior necessidade de existirem equipas de pesquisa e a trabalhar com múltiplos sistemas físicos e computacionais (Xu et al. 2018), o que também implica a cedência de orçamentos, os quais têm sido tendencialmente reduzidos (Coccia 2019).

De forma a ultrapassar estes desafios, os conceitos de *Open Science* e de *Open Lab* têm ganho relevância entre as comunidades de pesquisa. Efetivamente, o primeiro é considerado uma nova abordagem do processo científico, baseada no trabalho cooperativo e em novas formas de difusão do conhecimento (European Commission 2016) e o segundo surge no contexto da *Internet of Things* (IoT), uma vez que a abertura destes sistemas é imprescindível para que seja garantido o acesso às respetivas tecnologias exponenciais, bem como permitir o escalamento do próprio sistema (Stankovic 2014).

Arquitetura proposta

Nesse âmbito, o objetivo principal passa por criar um laboratório aberto destinado à manufatura e orientado para a comunidade, recorrendo ao Open Design, cuja dinâmica de produção é controlada por um Sistema Ciberfísico (CPS), o qual será auxiliado por um Sistema Multi-Agente. Enfatiza-se ainda que todos os recursos e serviços abrangentes ao sistema, seja a nível físico ou computacional/digital, vão ao encontro da premissa *open-source* e/ou *low-cost*.

Efetivamente, definiu-se uma arquitetura por camadas, em que parte delas suporta os serviços oferecidos pelo mundo físico e está alocada na *cloud* (*Cloud Computing*). Funcionalmente, o processo começa na camada da nuvem, a qual consiste num repositório, um servidor e serviços de gerenciamento que permitem preservar as informações dos produtos/projetos, disponibilizadas e gerenciadas pela comunidade, à qual lhe é concedido livre acesso ao repositório. Além disso, existe uma aplicação que suporta a interação utilizador-servidor (*frontend*), a qual permite ao primeiro escolher o seu produto, monitorizar continuamente o respetivo fabrico e ainda comunicar e colaborar com a restante comunidade e/ou o(s) operador(es) através de canais de comunicação direta síncronos. Para responder ao pedido, o servidor invoca uma série de serviços (*backend*) e, se necessário, recorre a aplicações de geração de *código-G*. Assim, toda a informação é devidamente processada e organizada, de modo que esteja pronta para ser distribuída para o sistema físico, por intermédio de software de controlo CNC ou infraestruturas *interface* CPS.

Efetivamente, é estabelecida uma infraestrutura de interface de comunicação CPS que permite que as informações circulem entre a nuvem e o (sub)sistema de informação e controlo na primeira camada do "mundo físico". A interface CPS explora uma infraestrutura industrial local IoT (IIoT) para transmitir e coordenar toda a informação previamente processada para a camada de produção, a qual é responsável pelo fabrico do produto pretendido, com recurso a um braço robótico, uma impressora 3D (Manufatura Aditiva), uma fresadora de mesa CNC (Manufatura Subtrativa) e um tapete de transporte.

Após a informação ter chegado ao seu devido destino e o sistema de produção ter dado início, é possibilitado ao operador acompanhar em tempo real todas as ações e operações realizadas pelos recursos físicos pertencentes ao sistema de produção, recorrendo ao sistema de comunicação e vigilância. Além disso, este subsistema também se encontra embutido à aplicação oferecida pela *cloud*, o que permite ao operador comunicar diretamente com o utilizador, que, por sua vez, também lhe é concedido acesso a essa monitorização.

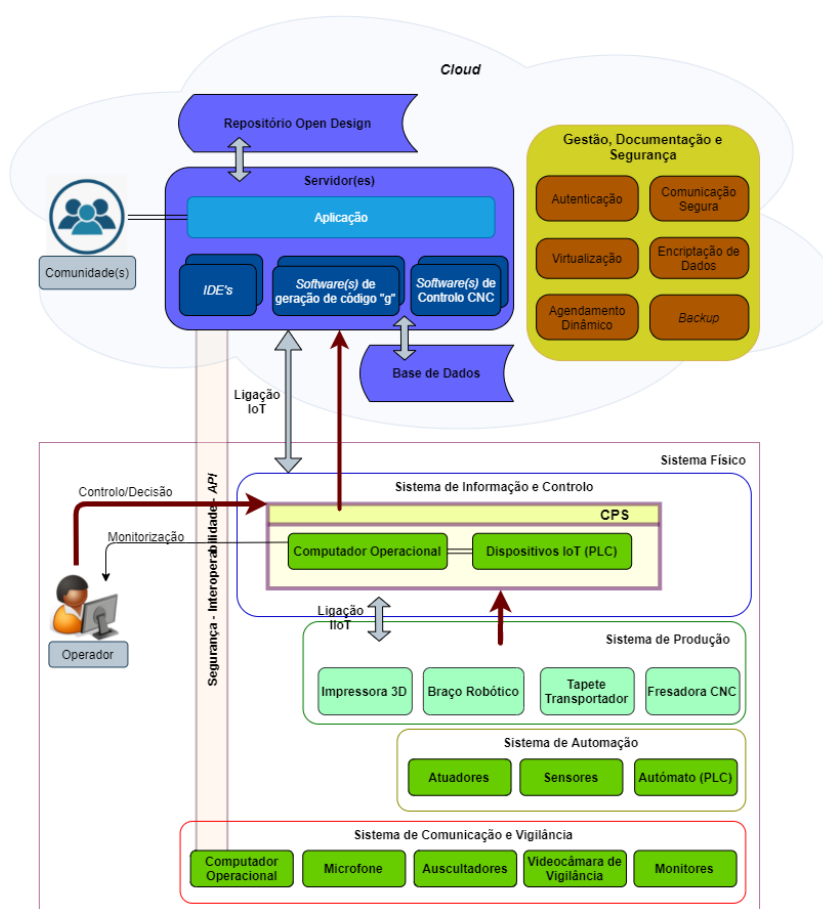


Figura 100 - Arquitetura proposta

É neste contexto que o nível de computação cibernética é relevante, dando alguma capacidade cognitiva a todo o processo, dado que existem máquinas, pessoas e processos a cooperar e colaborar simbioticamente. De facto, um conjunto de processos recebe continuamente dados do processo de manufatura atual (de cada unidade de produção), da análise do operador e das métricas esperadas, e reage ao refinamento dos algoritmos envolvidos, ao realinhamento das máquinas envolvidas, bem como, se necessário, efetua o redesenho ou reengenharia dos produtos propostos.

A título representativo, evidencia-se a Figura 100, onde estão ilustres todas as camadas/subsistemas mencionados, bem como o seu encadeamento, a alto nível.

Conclusões

Este trabalho comprova que é possível satisfazer as necessidades das Organizações de Pesquisa e Tecnologia recorrendo à arquitetura/sistema proposto. Efetivamente, as características do mesmo, mais propriamente, ser baseado na comunidade, escalável, replicável, ubíquo, de acesso remoto e aberto e providenciar novos modelos de negócio, produção e educação; bem como os conceitos inerentes à Indústria 4.0 que estão igualmente englobados (*Cloud Manufacturing*, *Industrial Internet of Things (IIoT 4.0)* e *Smart Factory*), colmatam as questões da obrigatoriedade de grandes investimentos em laboratórios e da necessidade da disseminação do conhecimento entre as comunidades de pesquisa.

Referências

- Coccia, Mario. 2019. "Metabolism of Public Research Organizations: How Do Laboratories Consume State Subsidies?" *Public Organization Review* 19(4):473–91.
- European Commission. 2016. "EU Open Innovation, Open Science, Open to the World." *European Commission* (June 2015):108.
- Putnik, Goran D. and Luis Ferreira. 2019. "What Is a Cyber-Physical System : Definitions and Models Spectrum." 663–74.
- Stankovic, John A. 2014. "Research Directions for the Internet of Things." *Ieee Internet of Things Journal* 1(1):3–9.
- Xu, Li Da, Eric L. Xu, and Ling Li. 2018. "Industry 4.0: State of the Art and Future Trends." *International Journal of Production Research* 56(8):2941–62.

