

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Sistema Integrado de Monitorização da Produção em Ambiente Fabril

Diogo Santos Lopes Simão de Magalhães

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador Interno: Professor Hélio Mendonça

Orientador Externo: Engenheiro Paulo Rocha

23 de Julho de 2016

Abstract

In a manufacturing environment, production monitoring is a critical process which accurate result analysis reveals to be crucial to improve the factory's production process. To perform this analysis it is important that the production data of each machine is effectively collected and properly documented.

At PRETTL Adion Portuguesa, the collection of each machine's production information is performed by the machine's operator and documented on a paper spreadsheet. This method of production control does not reveal to be the most effective, since it implies an additional process of recording data in the factory's information system. Furthermore, this procedure enhances the reporting of incorrect data, either by the machine operator or by the responsible of documenting such data in the information system.

The proposed solution to this problem is to develop a low cost automatic monitoring system, based on *Raspberry Pi* computers, able to effectively collect the machine's production data, process it and register it in a local database accessible by the factory's central database. This solution guarantees the correct registration of the collected data to allow for further analysis of the production performance.

Resumo

Num ambiente fabril, a monitorização da produção é um processo crítico cuja análise correta de resultados se revela determinante na melhoria do processo produtivo da fábrica. Para efetuar esta análise é importante que os dados de operação de cada máquina sejam recolhidos de forma eficaz e devidamente documentados.

Na PRETTL Adion Portuguesa, a recolha da informação de produção de cada máquina é realizada pelo operador da mesma e registada em papel manualmente. Este método de monitorização da produção não se revela o mais eficaz, visto que implica um processo adicional de registo dos dados no sistema de informação. Para além disso, este procedimento potencia a introdução de dados incorretos, quer pelo operador da máquina, quer pelo responsável pela introdução desses dados no sistema de informação.

A proposta de solução para este problema passa por desenvolver um sistema de monitorização automático de baixo custo, baseado em computadores *Raspberry Pi*, capaz de recolher eficazmente os dados de produção da máquina, processá-los e registá-los numa base de dados local acessível à base de dados central da fábrica. Esta solução garante o registo correto dos dados recolhidos para posterior análise da performance da linha de produção.

Agradecimentos

Em primeiro lugar, quero agradecer ao Professor Hélio Mendonça cuja disponibilidade ao longo do semestre e interesse no progresso do trabalho foram chaves importantes no sucesso deste projeto.

Quero também agradecer ao Engenheiro Paulo Rocha por me receber e se disponibilizar para discutir o projeto mesmo quando o tempo lhe era escasso.

Não posso deixar de mencionar os meus amigos Tiago, Daniel, Pi, Dinis, Miguel, Flávio, Renato, Tânia, Gisela, Mariana, Sofia, Catarina e Inês, com quem passei momentos inesquecíveis ao longo destes anos e com quem espero passar muitos mais.

Era impossível não agradecer à minha namorada, Mariana, que conheci durante o meu percurso académico e cuja amizade, amor e apoio emocional são essenciais na minha vida. Sem esquecer a ajuda preciosa que me deu na correção das gralhas linguísticas que ia dando durante a elaboração deste documento.

Quero agradecer à minha irmã porque se não o fizesse, não me perdoava.

Acima de tudo, quero agradecer aos meus pais por tudo o que fizeram por mim, não só ao longo destes anos como ao longo da minha vida. Obrigado pelos sacrifícios que fizeram para que os vossos filhos pudessem ter uma boa educação. Obrigado por fazerem questão de estar sempre presentes. Principalmente, obrigado pelo amor incondicional que sinto da vossa parte, todos os dias. Se há pessoas que merecem tudo de melhor no mundo, são vocês.

Diogo Magalhães

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos	1
1.3	Estrutura do Documento	2
2	Revisão Bibliográfica	3
2.1	Sistemas de Monitorização	3
2.2	<i>Raspberry Pi</i>	4
2.2.1	História	4
2.2.2	Componentes	7
2.3	Linguagem de Programação	10
2.4	Interfaces Gráficas	11
2.5	RFID	13
2.6	Base de Dados	14
2.7	Estudo das Ferramentas de Desenvolvimento	15
3	Tratamento de Sinais Digitais da Máquina de Injeção	19
3.1	Introdução	19
3.2	Conversão de Sinais de Entrada e de Saída	20
3.3	Conversão de Sinais de Alimentação	25
3.4	Conectores	27
3.5	Distribuição dos Pinos GPIO	28
3.6	Caracterização dos Sinais da Máquina	29
4	Sistema de Monitorização e Registo da Produção	31
4.1	Introdução	31
4.2	Levantamento de Requisitos	32
4.3	Estrutura e Funcionamento das Bases de Dados	34
4.3.1	Estrutura da Base de Dados Central da Fábrica	34
4.3.2	Estrutura da Base de Dados Local	35
4.4	Programa de Monitorização	39
4.4.1	Ambiente Gráfico	39
4.4.2	Métodos de Autenticação	44
4.4.3	Processamento da Atividade de Produção	53
4.4.4	Eventos e Subeventos Temporizados	64
4.4.5	Alarmes	68
4.4.6	Função de Atualização das Bases de Dados	69
4.4.7	Temporizadores	71

5	Plataforma de Demonstração e Análise de Dados	73
5.1	Introdução	73
5.2	Demonstração de Dados das Ordens de Fabrico em Curso	73
5.2.1	Dados Atuais das Ordens de Fabrico	73
5.2.2	Eventos em Curso e Eventos Terminados	75
5.3	Processamento das Ordens de Fabrico Concluídas	79
5.3.1	Dados de Identificação	79
5.3.2	Dados da Produção	81
5.3.3	Dados dos Eventos Acionados e Detetados	82
5.3.4	Dados dos Operadores de Substituição	83
5.3.5	Dados de Performance - OEE	83
6	Conclusões e Trabalho Futuro	87
6.1	Conclusões	87
6.2	Trabalho Futuro	88
A	Tipos de Defeitos das Peças Rejeitadas	89
B	Documentos do Circuito Impresso	91
C	Documentos de Registo da Atividade de Produção	95
D	Levantamento de Custos do Material	99
	Referências	101

Lista de Figuras

2.1	<i>Raspberry Pi Model A e Model B</i> [1]	5
2.2	Identificação dos componentes do <i>Raspberry Pi 2 Model B</i>	7
2.3	Oficial <i>Raspberry Pi Touch Display</i>	8
2.4	Esquema da disposição dos pinos no <i>Raspberry Pi 2 Model B</i> [2]	9
2.5	Mapa detalhado dos pinos no <i>Raspberry Pi 2 Model B</i> [3]	9
2.6	Aplicação gráfica que incrementa e decrementa valor numérico desenvolvida em <i>Tkinter</i> e <i>PyQt</i> independentemente	12
2.7	Leitor RFID MFRC522 e respetivas <i>tags</i>	14
2.8	Controlo dos pinos GPIO com um interruptor de 5 posições e um LED RGB	16
2.9	Preparação - Interface gráfica de interação com os pinos GPIO	17
2.10	Fase embrionária do programa utilizado na solução final	18
3.1	Placa de circuitos impressa de aquisição e conversão de sinais	20
3.2	Esquema de um optoacoplador	21
3.3	Circuito de conversão de 24V para 3,3V	22
3.4	Resultados da simulação do circuito da figura	22
3.5	Resultados em laboratório do circuito de conversão de 24V para 3,3V	23
3.6	Circuito de conversão de 3,3V para 24V	24
3.7	Resultados da simulação do circuito da figura	24
3.8	Resultados em laboratório do circuito de conversão de 3,3V para 24V sendo CH1 a entrada e CH2 a saída	25
3.9	Esquema do circuito de um <i>step-down voltage converter</i>	26
3.10	Modo de funcionamento de um <i>step-down voltage converter</i>	27
4.1	Sistema de Monitorização da Produção	31
4.2	Estilo de estruturação do ambiente gráfico no <i>QtDesigner</i>	39
4.3	Diferentes Grupos do ambiente gráfico desenvolvido	40
4.4	Grupo Dados de Produção	40
4.5	Grupo Dados do Operador e Estado da Máquina	41
4.6	Separador "Login"	41
4.7	Teclado Numérico Virtual	42
4.8	Separador "Eventos"	42
4.9	Separador "Subeventos"	43
4.10	Separador "Produção"	43
4.11	Separador "Rejeições"	44
4.12	Estado inicial de autenticação	44
4.13	<i>Grafcet</i> do método de autenticação por indicação do número de operador	45
4.14	Mensagem "Utilizador inexistente" no separador "Login"	46

4.15	Mensagem "Autenticado com sucesso!" no separador "Login"	47
4.16	Entrada na tabela "eventos" do subevento "Operador Planeado" após autenticação	47
4.17	Grafcet do método de autenticação por cartão RFID	48
4.18	Mensagem "Cartão Detetado. Autenticado com sucesso!" no separador "Login"	49
4.19	Grafcet do <i>timeout</i> da autenticação"	49
4.20	Mensagem "Cartão Detetado. Operador inesperado." no separador "Login"	50
4.21	Grupo "Dados do Operador e Estado da Máquina" atualizado com os dados do operador substituto	51
4.22	Entrada na tabela "eventos" do subevento "Operador Substituto"	51
4.23	Botão "Substituição" alterado para "Terminar Substituição"	51
4.24	Tabela "eventos" com entradas dos subeventos "Operador Planeado" e "Operador Substituto"	52
4.25	Separadores "Produção" e "Rejeições" com os <i>widgets</i> bloqueados	53
4.26	Grafcet das ações de processamento de uma ordem de fabrico	54
4.27	Campos "Ordem de Fabrico n ^o " desbloqueados	55
4.28	Campo direito ativado e campo esquerdo bloqueado no separador "Produção"	55
4.29	Mensagem "Ordem de fabrico inexistente." no separador "Produção"	56
4.30	Campo "N ^o cavidades reais" ativo no separador "Produção"	57
4.31	Mensagem "N ^o excessivo de cavidades" no separador "Produção"	58
4.32	Ordens de fabrico iniciadas	59
4.33	Entrada na tabela "eventos" dos dados de arranque da produção	59
4.34	Ordens de fabrico 8 e 9 com a produção em curso	60
4.35	Entrada na tabela "ocorrencias" dos dados das peças produzidas durante as ordens de fabrico 8 e 9	60
4.36	Separador "Rejeições" com a coluna "Esquerda" ativa	62
4.37	Estado de confirmação da quantidade de peças defeituosas detetadas	63
4.38	Tabela "rejeicoes" preenchida com as peças rejeitadas das ordens de fabrico 8 e 9	64
4.39	Entrada na tabela "eventos" da indicação de conclusão de ordens de fabrico	64
4.40	Grafcet do estado "Manutenção" ativado	65
4.41	Estado "Manutenção" ativado	66
4.42	Entrada na tabela "eventos" dos dados de início do evento "Manutenção"	66
4.43	Eventos e Subeventos - Entrada na tabela "eventos" dos dados de conclusão do evento "Manutenção"	66
4.44	Subeventos invisíveis no separador "Subeventos"	67
4.45	Subeventos visíveis do evento "Interrupção de Produção"	67
4.46	Subevento "Periféricos" ativo no separador "Subeventos"	68
4.47	Entrada na tabela "eventos" dos dados de início do subevento "Periféricos"	68
4.48	Entrada na tabela "eventos" dos dados de conclusão do subevento "Periféricos"	68
4.49	Estado de avaria do <i>robot</i>	69
4.50	Entrada na tabela "eventos" dos dados de início da avaria do <i>robot</i>	69
4.51	Entrada na tabela "eventos" dos dados de conclusão da avaria do <i>robot</i>	69
4.52	Entrada ambígua na tabela "eventos" de dados de arranque da produção	70
4.53	Entrada real na tabela "eventos" de dados de arranque da produção	70
5.1	Entradas da tabela "eventos" dos dados de "Arranque Produção" das ordens de fabrico 8, 9, 5 e 6	74
5.2	Entradas da tabela "eventos" dos dados de "Final Produção" das ordens de fabrico 8 e 9	74
5.3	Campo "Ordens de Fabrico" com os dados mais recentes das ordens de fabrico 5 e 6	75

5.4	Entradas da tabela "eventos" com o evento "Avaria" concluído e evento "Falta de Recursos" por concluir	75
5.5	Exemplo da página "Ordens Fabrico em Curso" com o evento "Falta de Recursos" em curso e o evento "Avaria" terminado	76
5.6	Página "Ordens de Fabrico em Curso" com os eventos "Falta de Recursos" e "Avaria" terminados	77
5.7	Página "Ordens de Fabrico em Curso" apenas com a ordem de fabrico 5 ativa	77
5.8	Estado da página "Ordens de Fabrico em Curso" sem ordens de fabrico ativas	78
5.9	Formulário com a lista das ordens de fabrico concluídas	79
5.10	Página "Ordens de Fabrico Concluídas" da ordem de fabrico número 6	80
5.11	Dados da ordem de fabrico número 6	81
5.12	Dados de produção da ordem de fabrico 6	81
5.13	Página "Rejeições" com a lista de defeitos encontrados na ordem de fabrico número 6	82
5.14	Informação relativa aos eventos ocorridos durante a ordem de fabrico 6	82
5.15	Ausência de eventos durante a ordem de fabrico	83
5.16	Informação relativa aos operadores substitutos autenticados durante a ordem de fabrico 6	83
5.17	Ausência de operadores substitutos durante a ordem de fabrico 9	83
5.18	Valor do OEE e indicadores da ordem de fabrico 6	84
5.19	Ausência do resultado do indicador disponibilidade da ordem de fabrico 8	85
B.1	Esquemático do circuito impresso	92
B.2	Pistas das camadas do PCB	93
B.3	Disposição dos componentes nas camadas do PCB	94
C.1	Folha de identificação e registo dos dados da atividade de produção	96
C.2	Folha de registo das peças rejeitadas	97

Lista de Tabelas

2.1	Características dos diferentes modelos <i>Raspberry Pi</i>	6
3.1	Pinos GPIO associados ao leitor RFID MFRC522	28
3.2	Pinos GPIO livres não associados	28
3.3	Pinos GPIO associados como entradas	28
3.4	Pinos GPIO associados como saídas	28
3.5	Sinais emitidos e recebidos pelo <i>Raspberry Pi</i> e respectivos pinos GPIO	29
4.1	Lista de eventos e subeventos definidos na fábrica	33
4.2	Estrutura da tabela "setup_eventos"	35
4.3	Estrutura da tabela "setup_rejeicoes"	35
4.4	Lista de eventos adicionados aos eventos da tabela 4.1	35
4.5	Estrutura da tabela "ordensfabrico"	36
4.6	Estrutura da tabela "ocorrencias"	37
4.7	Estrutura da tabela "rejeicoes"	37
4.8	Estrutura da tabela "eventos"	38
4.9	Respetivo "evento" e "subevento" dos botões "Manutenção", "Mudança de Molde" e "Avaria"	65
5.1	Codificação colorida do indicador "Disponibilidade"	85
5.2	Codificação colorida do indicador "Eficiência"	85
5.3	Codificação colorida do indicador "Qualidade"	85
A.1	Lista de defeitos das peças rejeitadas	90
D.1	Custos discriminados de todos os materiais que compõem um módulo do sistema	99
D.2	Custos discriminados de todos os materiais que compõem dez módulos do sistema	100

Abreviaturas

GPIO	<i>General Purpose Input Output</i>
Grafcet	<i>Grphe Fonctionnel de Commande, Étapes Transitions</i>
I ² C	<i>Inter-Integrated Circuit</i>
LED	<i>Light Emitting Diode</i>
OEE	<i>Overall Equipment Effectiveness</i>
PCB	<i>Printed Circuit Board</i>
RDBSM	<i>Relational Database Management System</i>
RFID	<i>Radio-Frequency Identification</i>
SPI	<i>Serial Peripheral Interface</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
VPN	<i>Virtual Private Network</i>

Capítulo 1

Introdução

1.1 Enquadramento

A fábrica PRETTL Adion Portuguesa desenvolve peças automóveis utilizando plástico como matéria prima. Os dados de produção destas peças são registados manualmente e depois compilados e inseridos no sistema de informação. Este processo manual de recolha e registo de dados é muito propício a incorreções e perdas de informação. Consciente deste problema, a fábrica procurava uma solução de baixo custo capaz de recolher automaticamente e eficazmente os dados de produção e registá-los numa base de dados para facilitar, posteriormente, o seu tratamento e consulta. Deste modo, reduz-se o volume de documentação originado pelo método de monitorização original e garante-se uma maior eficiência na análise dos resultados das atividades de produção. Neste documento é apresentada a solução encontrada que consiste num sistema automático orientado para as necessidades das máquinas de injeção. Especificamente, um módulo do sistema desenvolvido é composto por um *Raspberry Pi* com a função de recolher e processar toda a informação, um ecrã tátil que pretende facilitar a interação entre o programa de monitorização e o operador, um componente de identificação RFID que permite a autenticação automática dos operador no início do seu turno e, finalmente, uma placa de circuito impressa projetada para adaptar os níveis de tensão dos sinais da máquina aos níveis de tensão do *Raspberry Pi* e vice-versa.

1.2 Objetivos

A solução apresentada deverá ser capaz de recolher devidamente os sinais de injeção e avaria emitidos pela máquina. Deve traduzir os primeiros em dados de produção e os segundos em ações de alarme. As ações que não são emitidas pela máquina devem ser acionáveis pelo operador. Para além disso, o sistema deve apresentar o tempo de inatividade e atividade da máquina, o estado da mesma (em produção, parada ou em manutenção) e a identificação do operador responsável pela produção.

Todos a informação recolhida e acionada durante a atividade de produção deve ser inserida numa base de dados local estruturada consoante as necessidades da atividade de produção, que, por sua vez deve ser acessível à base de dados central da fábrica.

1.3 Estrutura do Documento

Depois deste capítulo introdutório, o Capítulo 2 descreve e justifica a escolha das ferramentas e tecnologias utilizadas no projeto. Segue-se o Capítulo 3 onde é apresentada a placa de circuitos impressa projetada para adaptar os níveis de tensão dos sinais emitidos pela máquina aos níveis de tensão do *Raspberry Pi*. O Capítulo 4 entra nos detalhes da solução proposta para o sistema de monitorização da produção. Neste capítulo também está incluída a descrição da estrutura da base de dados construída para acompanhar o sistema. Logo de seguida, o Capítulo 5 apresenta a plataforma que foi desenvolvida para permitir acompanhar a atividade de produção e analisar as ordens de fabrico concluídas. Por fim, a análise dos objetivos alcançados e a forma como o projeto pode ser melhorado e expandido estão incluídas no Capítulo 6.

Capítulo 2

Revisão Bibliográfica

2.1 Sistemas de Monitorização

Numa atividade de produção, a existência de um sistema de monitorização fiável é de elevada importância no controlo do processo produtivo e na análise da sua performance.

No caso da fábrica PRETTL Adion Portuguesa, os dados equivalentes a um turno de produção de cada máquina são documentados manualmente pelo operador do equipamento. Esta situação exemplifica um caso onde a responsabilidade da monitorização incide no próprio operador.

No entanto, a introdução de um *software* capaz de recolher esta informação potencia melhorias no processo de monitorização ao reduzir o número de possíveis erros no registo, o nível de responsabilidade do operador e o volume de documentação em circulação. Além disso, elimina a necessidade de um registo manual adicional no sistema de informação fábrica.

Em Portugal existem várias empresas especializadas no desenvolvimento de *software* que cumpre tais requisitos. Por exemplo, a *Arentia*, empresa sediada em Leiria que atua nas áreas de sistemas e tecnologias de informação, inclui no seu catálogo um sistema de gestão da produção, de nome *aProd*, [4] que através da aquisição dos sinais elétricos dos equipamentos permite monitorizar em tempo real a linha de produção. O *software* inclui uma interface de controlo acessível aos operadores via monitores táteis. No entanto, apesar da empresa afirmar que uma das características deste sistema é a sua adaptabilidade, a existência de um produto com um *software* pré-concebido pode incluir no seu orçamento a existência de funcionalidades desnecessárias para a monitorização em questão e a impossibilidade de configurar outras que sejam necessárias.

Por outro lado, mas no mesmo contexto, a *Critical Software*, empresa de Coimbra envolvida também na área das tecnologias de informação, apresenta um serviço de desenvolvimento de soluções de engenharia adaptadas às necessidades do cliente [5]. Neste caso, a problemática de um sistema com um *software* pré-concebido não se impunha visto que o produto seria totalmente adaptado e configurado tendo em conta os requisitos necessários.

Porém, os orçamentos dos sistemas propostos por ambas as empresas excedem significativamente o valor que a fábrica está disposta a investir na sua implementação. Dessa forma, nasce a

necessidade do desenvolvimento de um sistema de monitorização que para além de ser altamente adaptado às necessidades da fábrica, seja também de baixo custo.

2.2 *Raspberry Pi*

O termo *Raspberry Pi* refere-se a uma série de computadores de baixo custo do tamanho de um cartão de crédito, desenvolvidos pela *Raspberry Pi Foundation* [6]. Ao criar estes componentes, a fundação tinha como objetivo promover a educação em programação e ciências da computação, principalmente no ambiente escolar [7]. No entanto, estas peças de *hardware* não estão limitadas a funções pedagógicas. A sua utilidade pode ir desde o acesso à *internet* no dia-a-dia, até à aplicação em diversos projetos tecnológicos [8], graças à capacidade de transmissão de vídeo em alta definição e inclusão de pinos e vários protocolos de comunicação.

2.2.1 História

Em fevereiro de 2012 foram postos à venda os primeiros dois modelos do *Raspberry Pi* com uma diferença de preço de 10 dólares, o *Model A* e o *Model B* (figura 2.1). Este último tinha aliado ao seu preço acrescido uma porta USB adicional e uma porta *ethernet*, inexistente no *Model A* [9]. Ambos os modelos foram lançados com 256MB de RAM, no entanto, em outubro de 2012, o *Model B* sofreu uma revisão e passou a ser vendido com RAM de 512MB, sem acréscimo de preço [10].

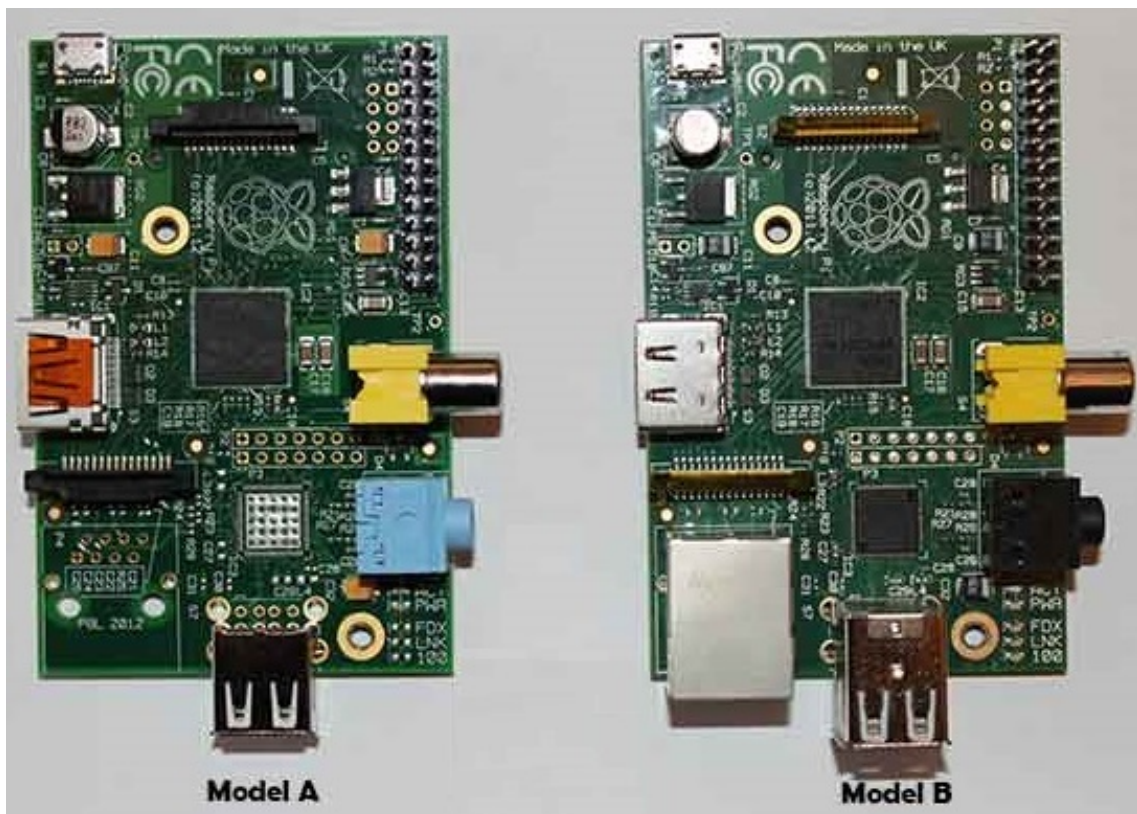


Figura 2.1: *Raspberry Pi Model A e Model B* [1]

Na segunda metade do ano de 2014 foram anunciados mais dois modelos, o *Model A+* e o *Model B+* [11][12]. Ambos apresentavam um aumento do número de pinos de 26 para 40 e o armazenamento passou a ser mais compacto, aceitando apenas cartões *Micro SD* em detrimento dos cartões *Full SD* dos modelos anteriores. Adicionalmente, o *Model B+* apresentou 2 portas USB adicionais em relação ao *Model A+*. O *Model B+* foi lançado ao mesmo preço do seu modelo anterior, enquanto que o *Model A+* chegou ao mercado como o *Raspberry Pi* mais económico até então, a custar 20 dólares.

Em menos de um ano, em fevereiro de 2015, a fundação lançou uma nova geração da placa e o modelo mais potente até então, o *Raspberry Pi 2 Model B* [13]. Comparando com *Raspberry Pi 1 Model B+*, o novo modelo inovou na potência de processamento com o novo processador ARM Cortex-A7 *quad-core* de velocidade de 900MHz. Para acompanhar a capacidade de processamento a memória RAM sofreu um *upgrade* de 512MB para 1GB.

Entretanto, em novembro de 2015, entra no mercado a versão mais económica de sempre do *Raspberry Pi*, o *Raspberry Pi Zero*, com um custo de 5 dólares [14]. De dimensões substancialmente mais reduzidas, esta placa apresenta o mesmo chip dos modelos iniciais e um processador de potência semelhante mas de velocidade superior. Relativamente ao modelo apresentado em 2014, o *Raspberry Pi Zero* chega com um memória RAM de metade do tamanho. Para contribuir para a economia da placa, tanto a saída HDMI como a entrada USB, apresentam respetivamente,

portas "Micro HDMI" e "Micro USB".

Finalmente, no último dia de fevereiro de 2016, chega ao público a terceira geração da placa, o *Raspberry Pi 3 Model B* [15]. Nesta versão, a capacidade de processamento sofre uma nova atualização com o processador ARM Cortex-A53 *quad-core*, cuja velocidade de processamento alcança os 1,2GHz. Outra diferença entre este e o modelo da geração anterior é a integração de um *chip* recetor de sinais *Wi-Fi* e *Bluetooth*, eliminando a necessidade da ligação de antenas externas de receção destes sinais.

Na tabela 2.1 estão detalhadas as principais características dos modelos mencionados.

Características	1 Model A	1 Model B	1 Model B+	2 Model B	Model Zero	3 Model B
Preço Lançamento (USD)	25	35	35	35	5	35
Chip (Boardcom)	BCM2835	BCM2835	BCM2835	BCM2836	BCM2835	BCM2837
CPU	ARM1176JZF-S single core	ARM1176JZF-S single core	ARM1176JZF-S single core	ARM Cortex-A7 quad-core	ARM1176JZ-F single core	ARM Cortex-A53 quad-core
Velocidade do CPU	700Mhz	700Mhz	700Mhz	900MHz	1000MHz Low Power	1,2GHz
RAM	256MB	512MB	512MB	1024MB	512MB	1024MB
Armazenamento	Full SD	Full SD	Micro SD	Micro SD	Micro SD	Micro SD
Ethernet 10/100	Não	Sim	Sim	Sim	Não	Sim
Saída de Vídeo	HDMI/ Vídeo Composto	HDMI/ Vídeo Composto	HDMI/ Vídeo Composto	HDMI/ Vídeo Composto	Micro HDMI/ Vídeo Composto	HDMI/ Vídeo Composto
Saída de Áudio	HDMI/ Conector TRS	HDMI/ Conector TRS	HDMI/ Conector TRS	HDMI/ Conector TRS	HDMI	HDM/ Conector TRS
Portas USB (nº)	1	2	4	4	1 (Micro)	4
Pinos (nº)	26	26	40	40	40	40
Alimentação	5V (DC)	5V (DC)	5V (DC)	5V (DC)	5V (DC)	5V (DC)

Tabela 2.1: Características dos diferentes modelos *Raspberry Pi*

O sistema operativo a correr na placa será o *Raspbian*, o sistema operativo oficial da *Raspberry Pi Foundation*. Este sistema operativo é baseado no *Debian* (sistema operativo *open-source* por sua vez baseado em *Linux*) mas otimizado para o *hardware* da placa.

2.2.2 Componentes

Como o projeto foi iniciado antes do lançamento do *Raspberry Pi 3 Model B*, a versão da placa utilizada no desenvolvimento do sistema foi o *Raspberry Pi 2 Model B*, o modelo com melhor capacidade de processamento na altura.

Na figura 2.2 estão identificados os vários componentes deste modelo.

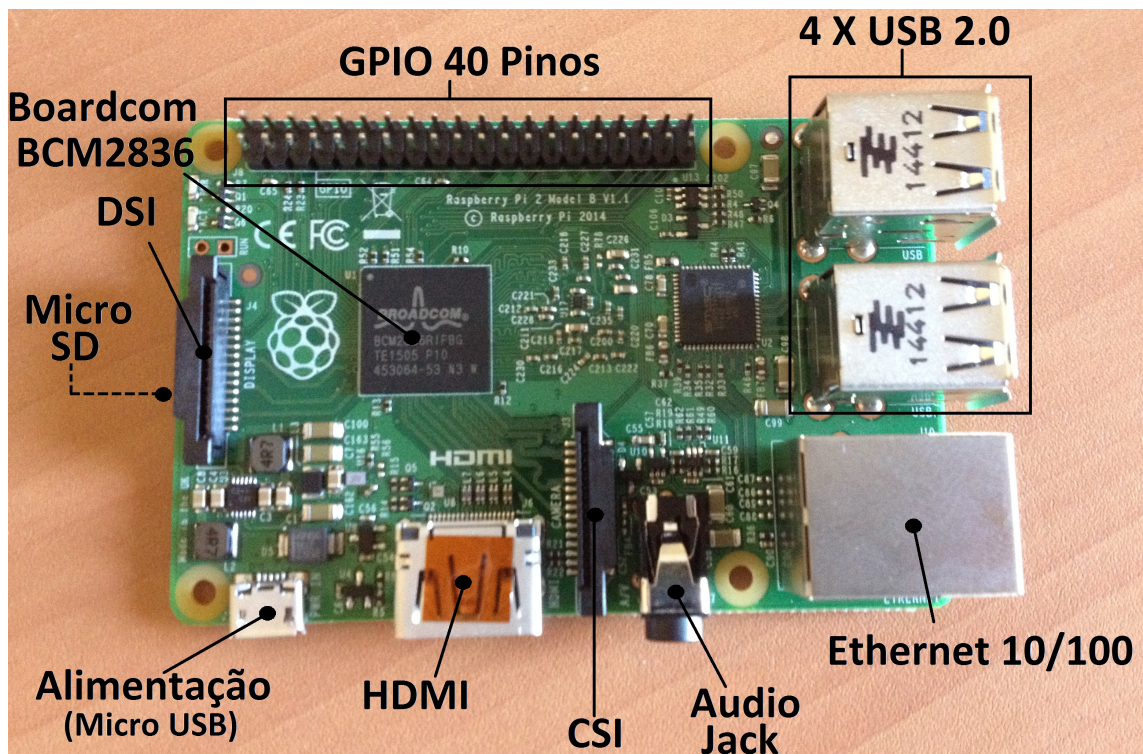


Figura 2.2: Identificação dos componentes do *Raspberry Pi 2 Model B*

2.2.2.1 Display Serial Interface

Um periférico integrante do sistema de monitorização a desenvolver será um ecrã tátil que facilite a interação entre o operador e a máquina. Para tal, esse ecrã teria de satisfazer os seguintes critérios:

- Tecnologia tátil
- Dimensões entre 6 e 10 polegadas
- Suportado pelo sistema operativo *Debian* e pelo *Raspberry Pi*

O *Raspberry Pi Touch Display* (figura 2.3), ecrã tátil oficial da *Raspberry Pi Foundation*, satisfaz todos estes critérios. Para além de ser um ecrã tátil capacitivo, tem 7 polegadas de dimensão, está otimizado para o *hardware* do *Raspberry Pi* e funciona automaticamente com o *Debian*, visto estar pré-instalado nesse sistema operativo. Este monitor permite um ângulo de visão de 70° e está preparado para ser acoplado aos últimos modelos do *Raspberry Pi* [16].



Figura 2.3: Oficial *Raspberry Pi Touch Display*

O ecrã conecta-se à placa via porta DSI, *Display Serial Interface*. Esta porta consiste num protocolo de transmissão de dados a alta velocidade específico da *Mobile Industry Processor Interface* [17], utilizado em controladores de vídeo.

2.2.2.2 Pinos GPIO

Como indicado na secção 2.2.1, os primeiros modelos do *Raspberry Pi* suportavam apenas 26 pinos, 8 de GPIO. O modelo a utilizar neste projeto, o *Raspberry Pi 2 Model B*, suporta no total 40 pinos, 26 deles GPIO.

Por GPIO entenda-se *General Purpose Input/Output*, isto é, são pinos que tanto podem ser definidos como saídas ou como entradas.

Na figura 2.4 é possível observar a disposição dos 40 pinos em modo 2x20. Para além dos 26 GPIO existem 2 pinos de alimentação a 5V, outros 2 de alimentação a 3.3V e 8 pinos ligados à terra, portanto, de 0V. Os pinos GPIO funcionam a uma tensão máxima de 3.3V e os pinos de 5V existem para a alimentação de periféricos ou até mesmo do próprio *Raspberry Pi*.

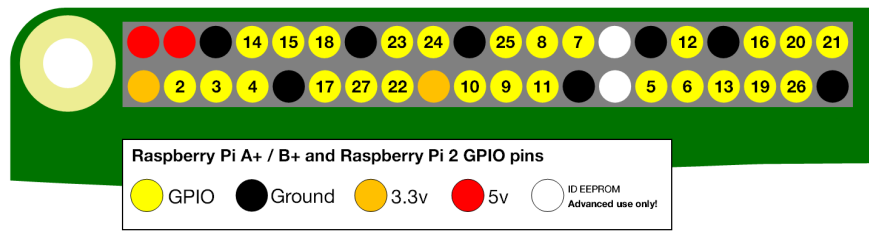


Figura 2.4: Esquema da disposição dos pinos no *Raspberry Pi 2 Model B* [2]

Alguns pinos GPIO podem ser configurados para funcionarem de forma alternativa. O mapa da figura 2.5 indica especificamente qual a função alternativa de cada um desses pinos.

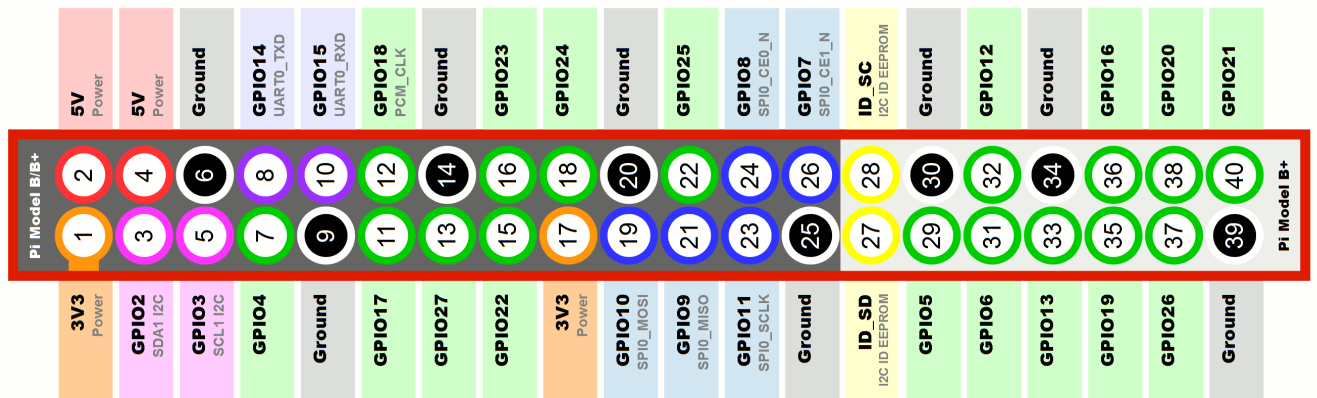


Figura 2.5: Mapa detalhado dos pinos no *Raspberry Pi 2 Model B* [3]

Portanto, dentro dos 26 pinos, para além dos 17 sem qualquer funcionalidade alternativa, temos 2 pinos I²C, 2 pinos UART e 5 pinos SPI.

O protocolo de comunicação I²C permite a comunicação com periféricos compatíveis com este protocolo, como micro-controladores e placas de extensão, e segue o modelo "múltiplos mestres, múltiplos escravos". No entanto, neste caso apenas existe um único mestre, o próprio *Raspberry Pi*. Os pinos GPIO 2 e 3 que suportam este protocolo incluem uma resistência *pull-up* de 1.8kΩ[18].

A porta série UART facilita a comunicação através de protocolos RS-232, RS-422 ou RS-485. No entanto, a limitação de tensão dos pinos a 3.3V implicaria a utilização de um conversor de tensão na utilização de alguns destes protocolos.

Finalmente, o barramento SPI é caracterizado pelo seu modo de comunicação *full-duplex*, ou seja, permite que a comunicação seja bidirecional e simultânea. A sua arquitetura segue o modelo "único mestre, múltiplos escravos" e dos três protocolos mencionados foi o único incorporado no projeto.

2.3 Linguagem de Programação

A placa *Raspberry Pi* suporta inúmeras linguagens de programação, desde que possam ser compiladas ou interpretadas pelo Chip ARM Cortex-A7. [19]

Algumas destas linguagens incluem:

- Scratch
- Python
- Perl
- Java
- Ruby
- C
- C++

Para o desenvolvimento do projeto, a escolha da linguagem incidiu entre *Python* e C ou C++. As duas últimas seriam, à partida, opções bastante viáveis devido à sua presença no programa curricular do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto, pelo que o seu conhecimento prévio seria uma vantagem. No entanto, apesar de existir uma falta de domínio e conhecimento da linguagem, a opção final incidiu no *Python*.

O *Python* é uma linguagem de alto nível. Tem por natureza um aspeto "limpo" e organizado devido à obrigatoriedade de indentação do código e sintaxe simplista, características que contrastam com as linguagens C e C++. Outras vantagens do *Python* relativamente às outras incluem a falta de necessidade de compilação, ou seja, trata-se de uma linguagem interpretada. Não existe a necessidade de alocar memória aquando da criação de novas variáveis, o interpretador executa esta função automaticamente. Contém um bom suporte para o desenvolvimento de aplicações e uma biblioteca mais vasta com um maior número de funções embutidas [18][20][21]. Adicionalmente, alguns destes aspetos transformam o *Python* numa linguagem de desenvolvimento mais rápido do que C/C++, ou seja, o mesmo programa seria concluído mais rapidamente em *Python* do que em C++ ou C.

Em contrapartida, em termos de velocidade de processamento, tanto o C como o C++ são superiores ao *Python*. Exemplos como a ação de interpretação, em vez de compilação do código, e a ação de *garbage collection*¹ reduzem a velocidade de processamento.

Contudo, esta desvantagem pode ser atenuada com o auxílio de técnicas de otimização do algoritmo. Para além disso, a velocidade de desenvolvimento do *Python* foi um fator de grande peso na decisão. Adicionalmente, a falta de domínio na linguagem foi vista como uma oportunidade de aprendizagem, permitindo, assim, a aquisição de um novo conhecimento, valorizando o aspeto pedagógico deste projeto. Porém, não deixa de ser importante ter em conta que o *Python* é a

¹Método automático de libertação de memória de objetos que já não estão a ser utilizados

linguagem recomendada pela *Raspberry Pi Foundation* no desenvolvimento de aplicações para o *Raspberry Pi*. [19]

2.4 Interfaces Gráficas

Este projeto requer que a monitorização da produção das máquinas seja executada num programa apoiado por uma interface gráfica. Portanto, para desenvolver esta interface em *Python*, foi necessário averiguar que ferramentas existiam para tal e qual a mais indicada.

Desse modo, as principais bibliotecas consideradas para o desenvolvimento dessa interface foram as seguintes:

- *TKinter*
- *wxPython*
- *PyQt*
- *Kivy*

Tanto o *TKinter* como o *wxPython* e o *PyQt* são empacotadores de bibliotecas de ferramentas para o desenvolvimento de interfaces gráficas, sendo estas bibliotecas *Tk*, *wxWidgets* e *Qt*, respetivamente.

Entre todos, o *TKinter* é o único cuja biblioteca está integrada no *Python* [22], portanto, a falta de necessidade da sua instalação é sem dúvida uma vantagem. No entanto, o visual "ultrapassado" dos seus *widgets*, pouco apelativos atualmente, e o número reduzido dos mesmos em relação às alternativas, limita a complexidade de projetos a desenvolver com esta ferramenta, o que se pode tornar um problema.. Porém, a simplicidade da sua sintaxe e a facilidade de adaptação a esta biblioteca são razões pelas quais a sua utilização ainda continua relevante.

As alternativas *wxPython* e *PyQt* colmatam as desvantagens do *Tkinter*. Nestas bibliotecas estão incluídas um número de *widgets* bastante superior e de aspeto visual mais atualizado. Nas figuras 2.6a e 2.6b é possível observar esta diferença visual, comparando a mesma aplicação desenvolvida em duas bibliotecas diferentes, *TKinter* e *PyQt*, respetivamente. O resultado no *wxPython* seria semelhante ao do *PyQt*.

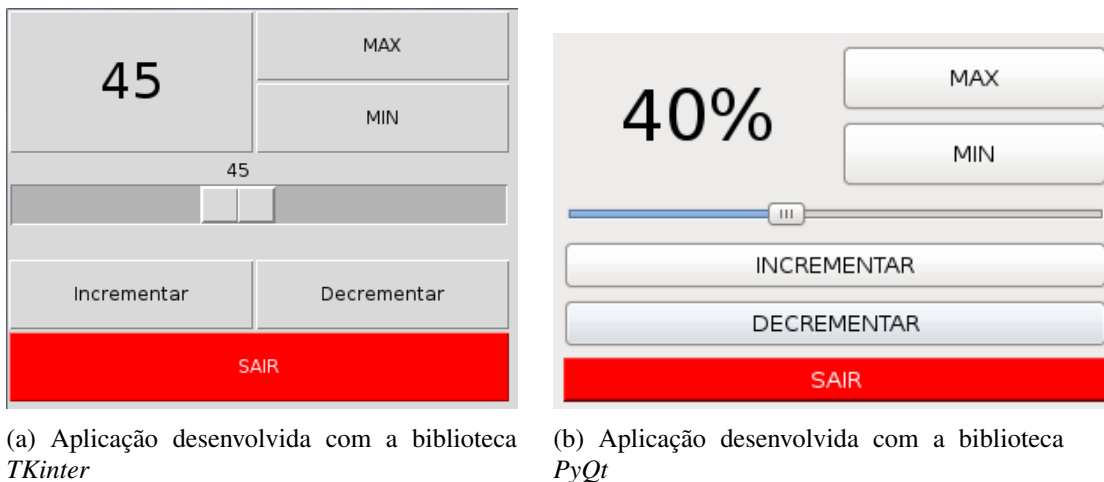


Figura 2.6: Aplicação gráfica que incrementa e decrementa valor numérico desenvolvida em *Tkinter* e *PyQt* independentemente

Porém, comparando o *PyQt* e *wxPython*, são encontradas diferenças nas suas funcionalidades. O primeiro inclui na sua instalação um editor gráfico de aplicações, o *QtDesigner*. Especificamente, este programa permite a criação gráfica de aplicações cujo ficheiro resultante pode ser convertido para código *Python* [20]. Esta funcionalidade é extremamente útil, visto permitir que a interface gráfica a ser desenvolvida no projeto seja mais rapidamente editável no futuro, sem que seja necessário grande conhecimento da linguagem de programação. Apesar de no caso do *wxPython* não existir de raiz uma ferramenta semelhante, foram, entretanto, desenvolvidos alguns editores gráficos para esta biblioteca, como o *wxGlade* e o *wxDesigner*. No entanto, nenhuma destas ferramentas inclui todas as funcionalidades do *wxPython* nem todos os *widgets* existentes.

No que diz respeito às bibliotecas, o *Kivy* é, entre todas, a que apresenta mais funcionalidades. Para além de permitir a customização de *widgets*, ao contrário das alternativas, inclui ainda a capacidade de desenvolvimento para sistemas operativos móveis. Ou seja, uma aplicação desenvolvida para *desktop* não necessita de ser altamente recodificada para ser compatível com uma plataforma *Android*, por exemplo. Para além disto, o suporte da interface de programação *OpenGL* permite incluir animações e efeitos especiais nas aplicações [23]. Estas funcionalidades, apesar de interessantes, são desnecessárias para o projeto em questão visto que a interface a ser desenvolvida não as requer.

Portanto, após estudar as vantagens e de desvantagens de cada uma, foi selecionada a biblioteca *PyQt* para o desenvolvimento da interface gráfica.

2.5 RFID

Como mencionado na secção 1.2, existe a necessidade do sistema identificar o funcionário responsável por operar a máquina. Essa identificação pode ser feita na interface gráfica através da entrada manual dos dados do utilizador ou, idealmente, com o auxílio da tecnologia de identificação automática *Radio Frequency Identification* (RFID).

O RFID é uma tecnologia que permite a transferência de dados via radio frequência, a curtas distâncias, sem a necessidade de fios. O sistema no seu estado mais básico é composto por uma *tag* e um leitor.

A *tag* é um objeto que funciona como dispositivo de identificação. Nesse objeto, que pode ser um cartão ou um porta-chaves, por exemplo, é aplicado um circuito integrado e uma antena, ambos geralmente de dimensões muito reduzidas. O circuito integrado armazena e processa os dados e a antena tem a função de os receber e transmitir.

O leitor é responsável por ler e/ou escrever a informação presente na *tag*. Esta informação é gerada e transmitida ao sistema de tratamento de dados ao qual o leitor está conectado, neste caso o *Raspberry Pi*.

Como alternativa a este sistema de identificação foi estudada a viabilidade da tecnologia de código de barras. No entanto, esta apresenta várias desvantagens em relação ao RFID. A informação presente em cada código de barras é estática, no RFID os dados presentes em cada *tag* podem ser reescritos. Esta possibilidade de escrita não existe no código de barras visto que nesta tecnologia só é possibilitada a leitura. Adicionalmente, um código de barras para ser lido necessita de estar no campo de visão direto do leitor, ao contrário do RFID no qual a proximidade é suficiente para que ocorra troca de dados entre a *tag* e o leitor [24].

Neste projeto o leitor RFID a ser integrado no sistema será o MFRC522 (figura 2.7). Este leitor lê e escreve à frequência de 13.56MHz. Utiliza o protocolo de comunicação SPI à tensão de 3.3V [25]. As *tags* associadas ao leitor, visíveis na figura 2.7, operam à mesma frequência e são passivas, isto quer dizer que não necessitam de alimentação interna, isto é, são energizadas por indução magnética no momento da transmissão de dados entre o leitor e a *tag*. Teoricamente à frequência de 13.56MHz essa transmissão pode ser feita até uma distância máxima de 1.5m [24], porém, na prática a distância é por vezes bastante inferior.

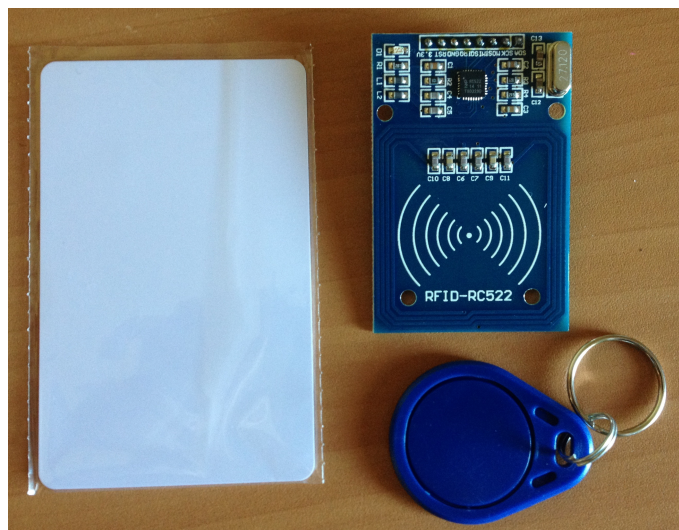


Figura 2.7: Leitor RFID MFRC522 e respectivas tags

2.6 Base de Dados

Neste projeto, é importante que a informação tratada e recolhida das máquinas seja armazenada para consulta e análise futura. Para facilitar estas ações, essa informação será guardada numa base de dados alojada num servidor local do *Raspberry Pi*. Esta bases de dados deverá ser acessível pelo sistema informático da fábrica.

O modelo relacional de base de dados foi selecionado como o ideal para as funções previstas. Este modelo estrutura a base de dados de forma a permitir interligações de informação entre as tabelas, através de um índice de identificação associado a cada entrada [26]. Esta funcionalidade será certamente útil aquando da necessidade de associar dados da máquina com dados dos operários das mesmas, por exemplo. O modelo não relacional não possui estas características e portanto foi considerado inapropriado e excluído.

Adicionalmente, a gestão da base de dados pode ser realizada por inúmeros RDBMS (*Relational Database Management Systems*) diferentes. Para este projeto, a opção pelo RDBMS ideal focalizou-se nos sistemas *SQLite*, *MySQL* e *PostgreSQL*. Estes RDBMS são *open-source*, um critério essencial, e são, também, bastante suportados pela comunidade do *Raspberry Pi*, facilitando a sua integração no projeto.

Dos três apresentados, o *SQLite* é certamente o mais diferenciado. Ao contrário dos outros, este sistema não está estruturado num modelo cliente-servidor. Em vez disso, a sua estrutura baseia-se numa base de dados embutida, ou seja, não corre num processo separado do programa onde está a ser utilizado [27]. Esta funcionalidade impede que os dados presentes no banco de dados sejam acedidos por mais do que um cliente, o que é indesejável para o sistema em desenvolvimento, visto que é esse mesmo o objetivo da implementação de uma base de dados neste projeto.

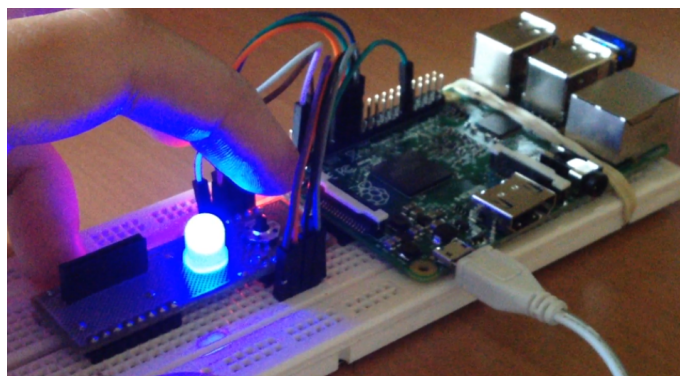
Para além disso, neste sistema é inexistente um método de autenticação, o que compromete a segurança e privacidade dos dados. Portanto, estes aspetos obrigaram à exclusão do *SQLite* como o RDBMS de eleição.

Os dois restantes, *MySQL* e *PostgreSQL* apresentam performances e capacidades muito semelhantes. Ambos são seguros, possuem um sistema de autenticação, são também facilmente escaláveis, mostrando uma boa capacidade de lidar com volumes de dados elevados [28] e são mais funcionais que o *SQLite*, ou seja, suportam um número acrescido de funções em SQL relativamente a este [29]. Portanto, ambos possuem uma estrutura robusta, funcional e confiável. No entanto, de momento, o *PostgreSQL* contém ligeiramente mais funcionalidades que o *MySQL*. Estas funcionalidades são normalmente acrescentadas em atualizações futuras do sistema. Porém, o desenvolvimento do *MySQL* parece estar mais estagnado do que o do *PostgreSQL*, sendo este último atualizado com mais frequência [30]. Este último ponto é um critério importante a ter em conta visto que este projeto visa a desenvolver um sistema que para além de útil seja futuramente escalável a outras máquinas da fábrica e o escalonamento é dificultado quando a tecnologia utilizada é datada. Para além disso, um aspeto que não deve ser ignorado é a experiência adquirida em *PostgreSQL* nas Unidades Curriculares estudadas ao longo do percurso académico, o que pode contribuir para uma aplicação mais eficaz deste sistema no projeto em discussão. Deste modo, a RDBMS eleita para a integração de base de dados no projeto foi o *PostgreSQL*.

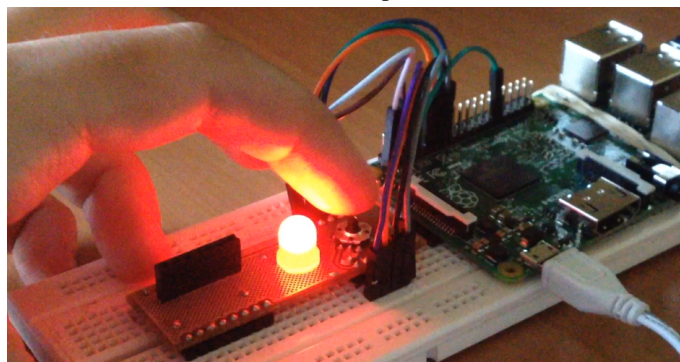
2.7 Estudo das Ferramentas de Desenvolvimento

Antes de avançar com a construção do sistema de monitorização final é importante dominar as ferramentas de trabalho mencionadas para que o processo de desenvolvimento da solução proposta seja o mais eficaz possível. Após a fase de configuração do *Raspberry Pi*, seguiram-se programas de testes de domínio dos pinos GPIO, interação destes com ambientes gráficos e integração adicional com a base de dados.

Numa fase inicial foram desenvolvidos simples programas em *Python* para testar o controlo dos pinos GPIO. Para este fim, foi utilizado um pequeno circuito que comportava um LED RGB e um interruptor de 5 posições. Os pinos destes componentes foram ligados aos pinos GPIO do *Raspberry Pi* e definidos como entradas e saídas usando a biblioteca *RPIO.GPIO* [31]. Os programas usavam rotinas que acendiam os diferentes LED assim que cada botão era acionado. Apesar de pouco complexa, esta fase foi necessária para dominar a aquisição e processamento de sinais



(a) Pino de saída ativo correspondente do LED azul



(b) Pino de saída ativo correspondente do LED vermelho

Figura 2.8: Controlo dos pinos GPIO com um interruptor de 5 posições e um LED RGB

A fase seguinte passou por construir uma aplicação gráfica que interagisse com os sinais recebidos via pinos GPIO e registasse a informação numa base de dados local. O programa, visível na figura 2.9a, estava preparado para ler sinais de três pinos configurados como entradas. A aplicação contabilizava o número de vezes que era detetado cada sinal e atualizava com esse valor uma tabela presente na base de dados local.

Esta experiência permitiu perceber como trabalhar com interfaces gráficas desenvolvidas com o *PyQt* e como integrar a leitura e escrita de uma base de dados *PostgreSQL* em programas baseados nessa biblioteca.



(a) Preparação - Estado inicial da interface



(b) Preparação - Estado da interface após a detecção de algumas variações de sinal

Figura 2.9: Preparação - Interface gráfica de interação com os pinos GPIO

Por último, numa fase inicial de trabalho em conjunto com a fábrica, foi desenvolvido um programa que acabaria por se tornar num ponto de partida para a solução final. Nesta fase, também foi iniciada a construção de uma base de dados mais complexa e orientada às necessidades do sistema. O programa das figuras 2.10a e 2.10b já incluía o tratamento de alguns requisitos como o processamento de ordens de fabrico e eventos acionados pelo operador. Podem ser observados traços deste programa na solução final descrita no Capítulo 4

Esta instância do projeto foi essencial para perceber as potencialidades das ferramentas escolhidas para o desenvolvimento de um sistema industrial de monitorização automática da produção.

Login | **Ordens** | Dados/Eventos | Subeventos

Ordem de Produção: **Ordem de Fabrico Esquerda:** **Ordem de Fabrico Direita:**

Iniciar Produção

OF#1 OF#1
 OF#2 OF#2
 OF#3 OF#3
 OF#4 OF#4
 OF#5 OF#5
 OF#6 OF#6
 OF#7 OF#7
 OF#8 OF#8
 OF#9 OF#9

SAIR

(a) Separador de seleção das ordens de fabrico

Login | **Ordens** | **Dados/Eventos** | Subeventos

Esquerda - Ordem de Fabrico nº x:

Equipamento: -- Molde: --
 Nº de peças a produzir: ---- Qtd por caixa: ----
 Nº peças boas: ---- Nº peças más: ----
 Taxa de Qualidade: **0.00%**

Direita - Ordem de Fabrico nº x:

Equipamento: ---- Molde: --
 Nº de peças a produzir: ----- Qtd por caixa: --
 Nº peças boas: ---- Nº peças más: ----
 Taxa de Qualidade: **0.00%**

EVENTO1 EVENTO2 EVENTO3
 EVENTO4 EVENTO5 EVENTO6
 EVENTO7 EVENTO8 EVENTO9

SAIR

(b) Separador dos dados da produção

Figura 2.10: Fase embrionária do programa utilizado na solução final

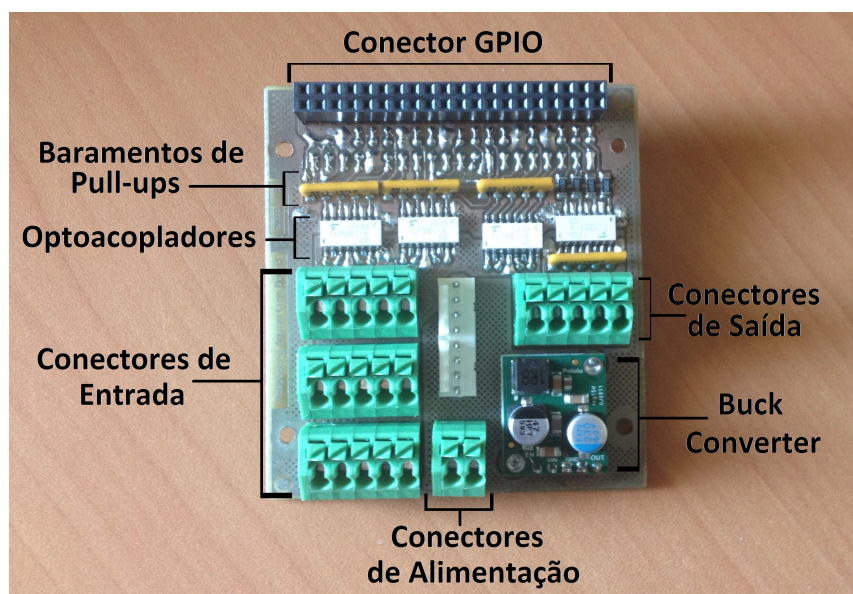
Capítulo 3

Tratamento de Sinais Digitais da Máquina de Injeção

3.1 Introdução

Parte dos dados relevantes para a monitorização da produção provêm da própria máquina de injeção através de sinais digitais. Esta máquina apenas emite e recebe os sinais a 0V ou 24V. No entanto, o *Raspberry Pi*, é alimentado apenas a 5V e os pinos GPIO funcionam a 3,3V. Estas particularidades implicam a necessidade de conversão dos valores de tensão destes sinais antes que sejam emitidos ou recebidos. Para tal, foi projetado e desenhado o PCB da figura 3.1 com o objetivo principal de integrar circuitos de adaptação dos níveis de tensão dos sinais de entrada, saída e alimentação.

Este capítulo explicita os circuitos projetados, a escolha dos componentes incluídos na placa e a distribuição de funcionalidades dos pinos GPIO. O esquemático final da placa, a trajetória das pistas e o desenho da posição dos componentes podem ser consultados no anexo B.



(a) Vista de cima



(b) Vista de baixo

Figura 3.1: Placa de circuitos impressa de aquisição e conversão de sinais

3.2 Conversão de Sinais de Entrada e de Saída

O primeiro desafio passou por converter sinais de 24V em sinais de 3,3V e vice-versa. Para tal, a solução encontrada incidiu na utilização de optoacopladores. Os optoacopladores são componentes integrados compostos por um LED e um fototransistor (3.2) e o seu funcionamento é muito

simples. Quando a corrente passa pelo LED, este acende, sendo a luz detetada pelo fototransistor, que permite que a corrente passe por ele. Desta forma, caso exista uma descarga de tensão muito elevada à entrada, o LED fica danificando, protegendo o circuito de saída ao impedir a transmissão do sinal.

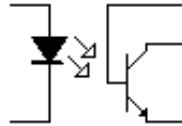


Figura 3.2: Esquema de um optoacoplador

O modelo do optoacoplador utilizado nos testes do circuito de conversão foi o 4N25 cuja tensão de ruptura entre o coletor e o emissor do fototransistor é, no mínimo, de 30V[32]. Visto que o sinal de entrada é no máximo de 24V, enquadra-se nas necessidades do circuito. No entanto, para limitar a corrente no LED, é necessário calcular o valor de uma resistência a colocar entre este e o sinal de entrada.

Segundo a folha de características do integrado, a queda de tensão do LED é tipicamente de 1,3V e a corrente para essa queda de tensão de 50mA[32]. Sendo assim, para um sinal de entrada de 24V, o valor resultante da resistência seria de 454Ω:

$$R_{in} = \frac{24V - 1,3v}{0.05A} = 454\Omega \quad (3.1)$$

Com este valor de resistência para uma corrente de 50mA, obtemos uma potência de 1.135W, o que é um valor demasiado alto. Este nível de potência elevado iria certamente causar danos na resistência, algo que acabou por ser comprovado em testes feitos no laboratório. Por isso, era necessário chegar a um valor de resistência mais elevado para baixar a corrente e, dessa forma, a potência. Tendo em conta o material disponível no laboratório foi calculada a validade da utilização de uma resistência de 6,8kΩ:

$$I_{in} = \frac{24V - 1,3v}{6800\Omega} = 0.0033A \quad (3.2)$$

Com uma resistência de 6,8kΩ, o valor da corrente baixa para 3,3mA e a potência dissipada baixa para 74,1mW, que são valores válidos, quer para a transmissão do sinal ao fototransistor quer, para a durabilidade da resistência.

O circuito projetado à saída do fototransistor segue uma configuração inversa com uma resistência de *pull-up* de 10kΩ. Isto significa que quando a entrada tem um valor lógico "1", a saída tem um valor lógico "0" e vice-versa. Sendo assim, simulando o circuito completo apresentado na figura 3.3, obtemos o resultado da figura 3.4. Assim, é possível confirmar que, quando o sinal de entrada é 24V, a saída é aproximadamente 0V e quando a entrada é de 0V, na saída obtemos 3,3V,

tal como previsto. Montando e testando este circuito, obtêm-se os mesmos resultados. Idealmente, estes resultados deviam ser apresentados gerando, na entrada, uma onda quadrada de 0V a 24V. No entanto, o gerador de sinal presente no laboratório não é capaz de alcançar níveis tão elevados de tensão, por isso, a figura 3.5 apresenta os resultados obtidos com a entrada desligada (figura 3.5a) e ligada (figura 3.5b) a uma fonte de tensão de 24V.

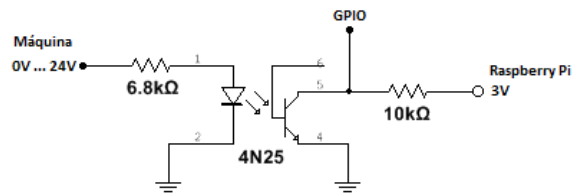


Figura 3.3: Circuito de conversão de 24V para 3,3V

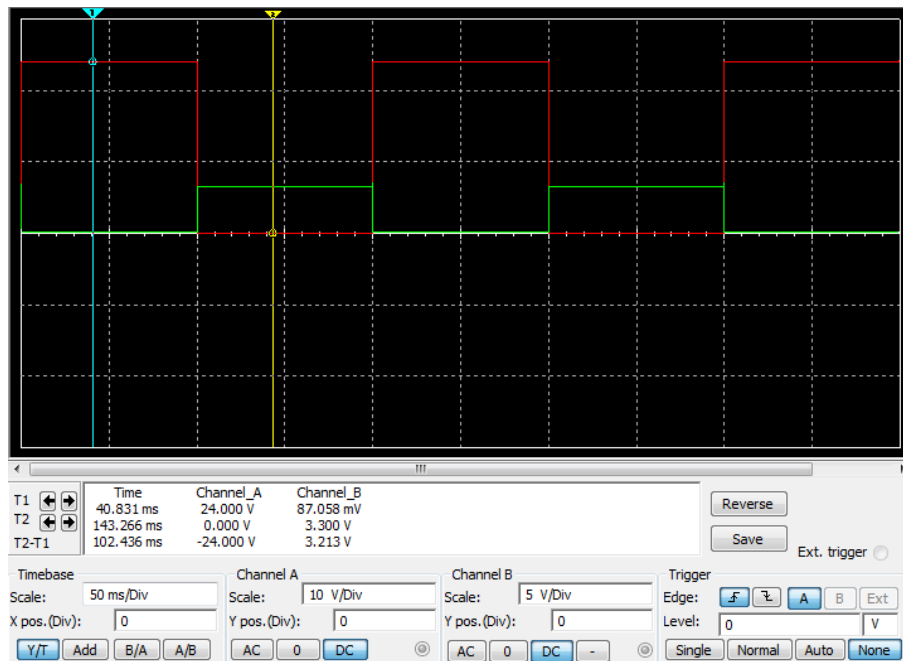
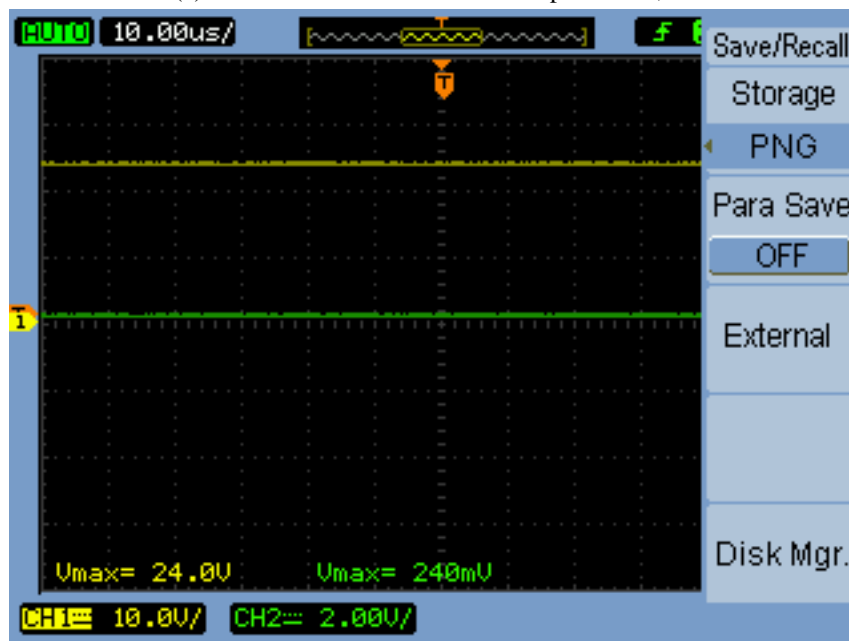


Figura 3.4: Resultados da simulação do circuito da figura 3.3



(a) Entrada CH1 a 0V e saída CH2 perto de 3,3V



(b) Entrada CH1 a 24V e saída CH2 perto de 0V

Figura 3.5: Resultados em laboratório do circuito de conversão de 24V para 3,3V

Esta configuração apenas valida a conversão de 24V para 3,3V, ou seja, apenas permite ler as entradas da máquina. Porém, também é necessário converter os sinais de saída dos pinos GPIO para que possam ser recebidos pela mesma máquina. Isto é, convertê-los de 3,3V para 24V. Para tal, é possível utilizar o circuito anterior desde que a resistência à entrada do LED seja recalculada para o valor de tensão de entrada de 3,3V. Mantendo o mesmo valor de corrente (3,3mA) obtém-se

o valor resistivo de 606Ω :

$$R_{in} = \frac{3,3V - 1,3v}{0.0033A} = 606\Omega \quad (3.3)$$

Substituindo o valor obtido em 3.3 por 680Ω , visto ser o valor de resistência mais próximo ao calculado existente no laboratório, obtém-se uma corrente de $2,9mA$ e uma potência de $5,9mW$. O circuito da figura 3.6 tem os resultados simulados apresentados na figura 3.7 e os resultados reais observáveis na figura 3.8. Assim, pode-se concluir que a conversão de $3,3V$ para $24V$ é feita como pretendido e, portanto, esta projeção é também validada.

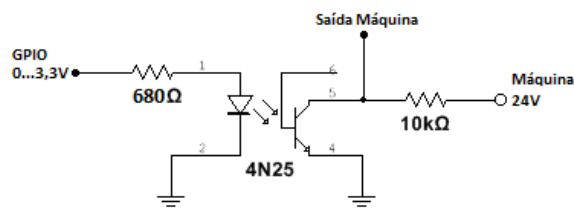


Figura 3.6: Circuito de conversão de $3,3V$ para $24V$

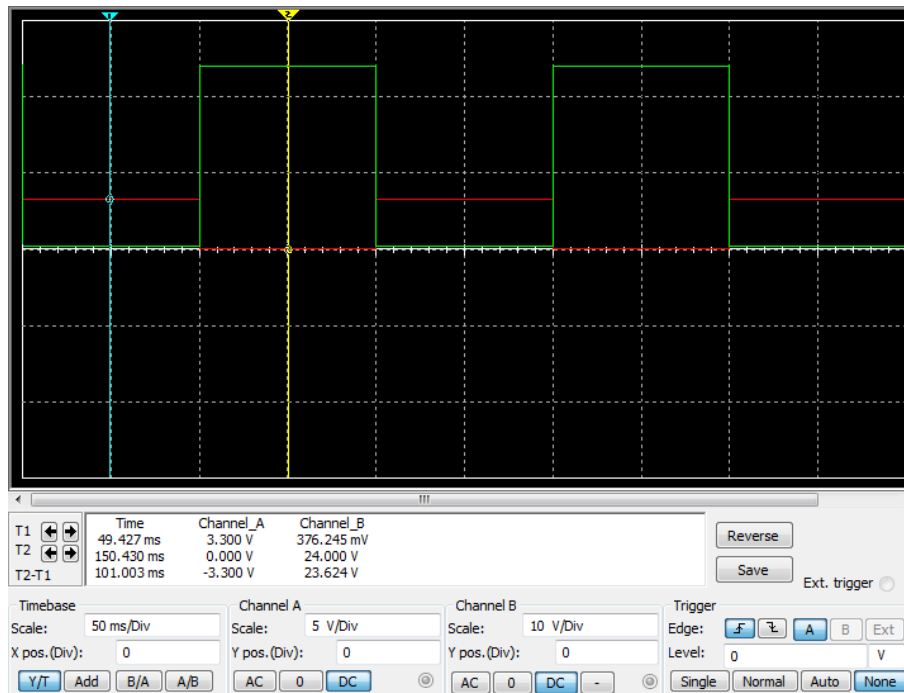


Figura 3.7: Resultados da simulação do circuito da figura 3.6

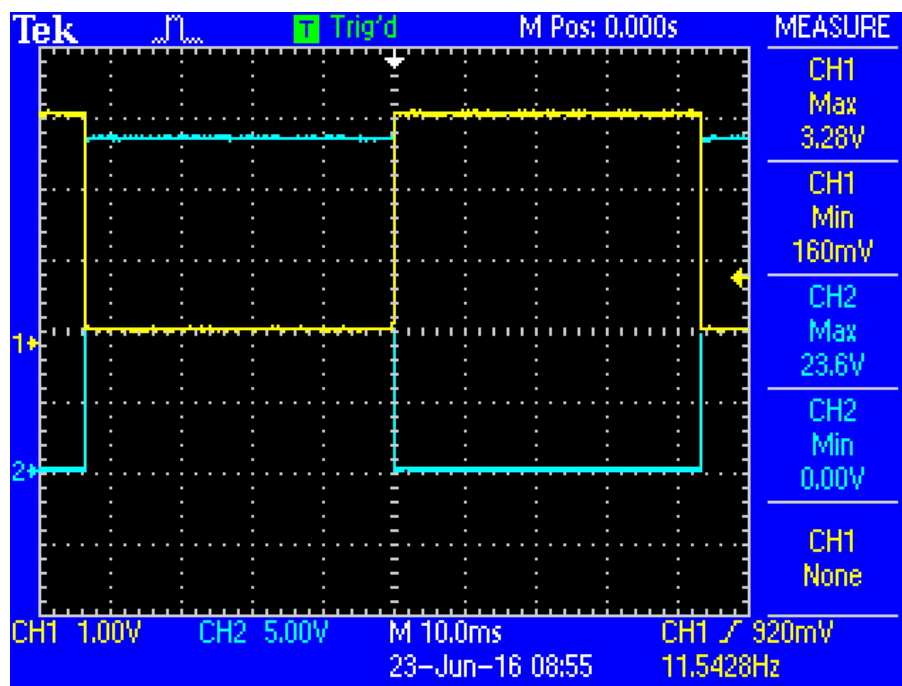


Figura 3.8: Resultados em laboratório do circuito de conversão de 3,3V para 24V sendo CH1 a entrada e CH2 a saída

Um critério tido em conta no desenho do PCB foi a necessidade de aproximar ao máximo as dimensões do mesmo às dimensões do *Raspberry Pi*. Assim, como o tamanho era um fator importante e visto que cada 4N25 é composto apenas por uma entrada e uma saída, este optoacoplador foi substituído pelo equivalente TLP281-4, que num só componente integra quatro entradas e quatro saídas, ou seja, quatro LED e quatro fototransístores [33]. Desta forma, o espaço ocupado pelos optoacopladores na placa é reduzido consideravelmente.

O número de optoacopladores a serem integrados na placa foi definido tendo em mente outros possíveis projetos com as mesmas necessidades de conversão. Portanto, apesar deste sistema trabalhar com sete entradas e uma saída, o PCB está preparado para converter doze sinais de entrada e quatro de saída, ou seja, inclui quatro componentes integrados TLP281-4.

3.3 Conversão de Sinais de Alimentação

Apesar dos pinos GPIO funcionarem a 3,3V, o próprio *Raspberry Pi* é alimentado a 5V, o que implica a necessidade de incluir no PCB um circuito capaz de reduzir o sinal de alimentação de 24V fornecido pela máquina para os 5V necessários no *Raspberry Pi*.

Para pôr em prática esta redução de tensão, foi primeiramente considerado o LM7805, um regulador de tensão linear capaz de devolver um sinal de saída de 5V com uma entrada até 25V [34]. Este tipo de reguladores atua dissipando a potência da diferença entre a tensão de saída e a tensão de entrada. Assim, como o LM7805 devolve apenas uma corrente de 1,5A, a potência

dissipada entre os dois níveis de tensão mencionados seria de 28,5W. Este valor tão elevado leva ao aquecimento excessivo do integrado e, portanto, à necessidade de incorporar um dissipador de calor.

Como esta solução não se revelou muito prática, optou-se por utilizar um *step-down voltage converter*. Este tipo de conversor de tensão é mais eficaz no tratamento da potência dissipada porque se apoia num sistema diferente para regular o valor da tensão de saída.

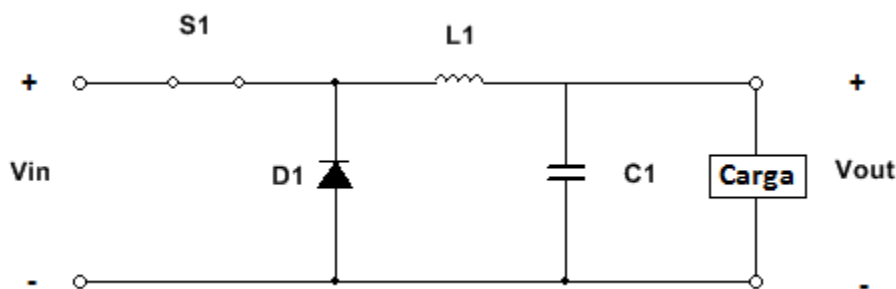


Figura 3.9: Esquema do circuito de um *step-down voltage converter*

Como se pode observar na figura 3.9, um *step-down voltage converter* é composto essencialmente por um interruptor, uma bobine, um condensador e um díodo. Quando o interruptor está fechado, a corrente derivada da fonte flui pela bobine, onde uma parte é armazenada e a restante é distribuída pelo condensador e pela carga no sentido visível na figura 3.10a. Neste estado, o díodo encontra-se no sentido inverso e não conduz. Adicionalmente, a carga do condensador também vai aumentando gradualmente.

Quando o interruptor é aberto (figura 3.10b), a corrente armazenada na bobine é libertada para o resto do circuito e a polaridade deste componente é invertida. Esta corrente continua a fluir pelo circuito através do díodo. Quando a corrente libertada pela bobine se começa a esgotar, o condensador passa a ser a maior fonte de energia da carga, mantendo a corrente a fluir por esta até o interruptor ser fechado novamente.

Desta forma, a corrente na saída e, por consequência a tensão, é mantida a um valor constante mas reduzido relativamente ao valor da entrada porque é distribuída pelos outros componentes do circuito.

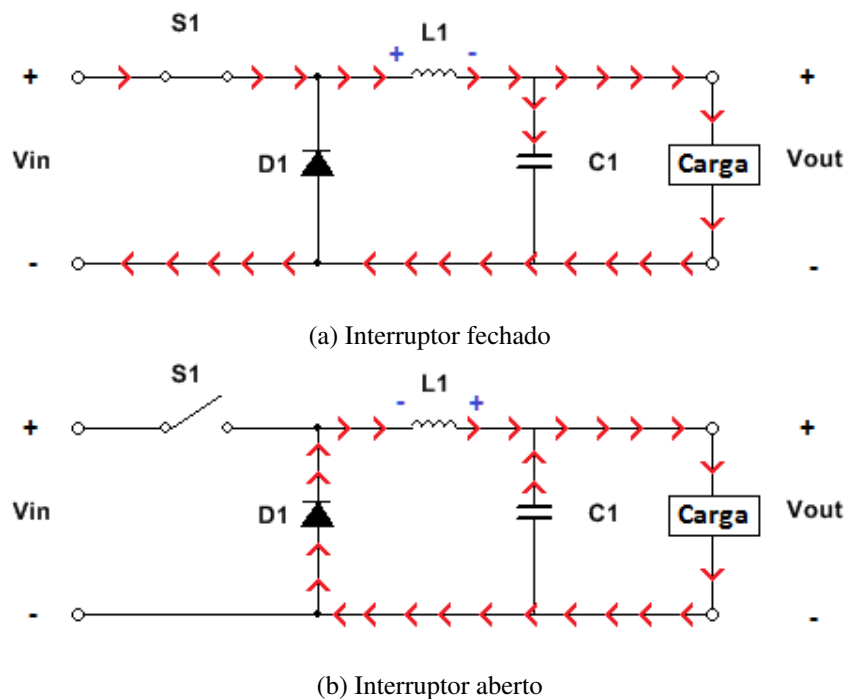


Figura 3.10: Modo de funcionamento de um *step-down voltage converter*

Este tipo de regulador revelou-se mais adequado aos valores de tensão em causa e o modelo escolhido foi o D24V50F5. Este componente suporta uma tensão de entrada até 38V e devolve uma tensão de saída contínua de 5V, a uma corrente de 5A [35], o que é um valor de corrente confortável para alimentar todos os periféricos conectados ao *Raspberry Pi*.

3.4 Conectores

No desenho do PCB, a escolha dos conectores foi também um processo importante para cumprir o critério de ocupação mínima de espaço na placa.

O modelo escolhido para conectar os sinais de entrada, saída e alimentação aos respetivos circuitos foi o PTSA 1,5/ 5-3,5-Z. Este conector de mola tem um ângulo de entrada de 45°, um buraco de entrada de 1mm de diâmetro (os fios dos sinais da máquina têm um diâmetro de 0.977mm) e suporta valores de tensão até 400V [36].

Para além desse conector, foi também incluído um conector macho KK 254 de 8 pinos [37] para conectar ao leitor RFID MFRC552 que, por sua vez, tem soldado no seu terminal um conector fêmea equivalente.

Adicionalmente, o PCB inclui dois conectores fêmea de 40 entradas [38]. Um deles para acoplar a placa aos pinos GPIO do *Raspberry Pi* (3.1b) e o outro para permitir o acesso aos GPIO inutilizados pelos circuitos presentes na placa.

3.5 Distribuição dos Pinos GPIO

Como mencionado na secção 2.2.2.2, alguns pinos GPIO têm funções alternativas associadas a protocolos de comunicação cuja funcionalidade pode ser útil caso exista a necessidade de incluir um periférico que utilize um destes protocolos para comunicar com o *Raspberry Pi*. Portanto, no processo de atribuição da função de cada pino, foi tida em atenção esta particularidade.

Neste projeto, o protocolo SPI foi o único cuja configuração se revelou necessária para permitir a comunicação com o leitor RFID MFRC522. Os pinos GPIO associados a este leitor estão listados na tabela 3.1.

Leitor RFID					
Pino	24	23	19	21	22
GPIO	8	11	10	9	25
	SPI0_CE0_N	SPI0_CLK	SPI0_MOSI	SPI0_MISO	-
Protocolo	SPI				-

Tabela 3.1: Pinos GPIO associados ao leitor RFID MFRC522

Os restantes protocolos, I²C e UART, não iam ser necessários e, portanto, os respetivos pinos ficaram desassociados e livres de serem acedidos pelo conector de 40 pinos presente na face superior do PCB. A tabela 3.2 referencia os pinos e os respetivos protocolos.

Pinos Livres					
Pino	3	5	8	10	18
GPIO	2	3	14	15	24
	SDA1	SCL1	UART0_TXD	UART0_RXD	-
Protocolo	I ² C		UART		-

Tabela 3.2: Pinos GPIO livres não associados

Assim, os pinos GPIO livres de funcionalidades alternativas foram definidos como entradas e saídas do sistema, ou seja, ficaram associados aos respetivos circuitos de conversão descritos na secção 3.2. Os pinos associados a entradas e saídas estão expostos nas tabelas 3.3 e 3.4, respetivamente.

Entradas												
Pino	7	11	12	13	15	16	26	29	31	32	33	35
GPIO	4	17	18	27	22	23	7	5	6	12	13	19

Tabela 3.3: Pinos GPIO associados como entradas

Saídas				
Pino	36	37	38	40
GPIO	16	26	20	21

Tabela 3.4: Pinos GPIO associados como saídas

3.6 Caracterização dos Sinais da Máquina

Definidos os pinos disponíveis como entradas e saídas no *Raspberry Pi*, resta apresentar quais estão associados os sinais emitidos pela máquina de injeção e que sinais são estes.

O sistema desenvolvido recebe e processa sete sinais emitidos pela máquina, no entanto, apenas um sinal é emitido para a mesma. A tabela 3.5 apresenta estes sinais e a que pinos de entrada e saída do *Raspberry Pi* estão associados. O sinal "Automático/Manual" representa em que modo se encontra a máquina. O modo automático, emitido com o sinal em valor lógico "1", sinaliza que a máquina se encontra em estado de produção ou preparada para produzir. O modo manual, emitido com o sinal em valor lógico "0", é ativado na necessidade de uma ação de intervenção na máquina, como uma mudança de molde ou uma atividade de manutenção. O sinal "Injeção no Molde" é emitido quando tal ação ocorre e, no caso desta máquina, tem um tempo de ciclo entre 30 a 60 segundos. O número de peças produzidas por injeção é igual ao número de cavidades existentes no molde. Se após a injeção for emitido o sinal de "Rejeição da Injeção", significa que foi detetado algum defeito nas peças injetadas. Adicionalmente, a máquina comporta a emissão de quatro sinais de avaria com um tempo de ciclo igual ao tempo de duração da avaria. Por último, o único sinal recebido pela máquina, sinal de "Paragem", suporta a ação de interrupção do processo de injeção.

Sinais de Entrada	Pino GPIO
Automático/Manual	18
Injeção no Molde	4
Rejeição da Injeção	17
Avaria da Máquina	6
Avaria do Molde	5
Avaria do Robot	23
Avaria dos Periféricos	7
Sinal de Saída	Pino GPIO
Paragem	16

Tabela 3.5: Sinais emitidos e recebidos pelo *Raspberry Pi* e respetivos pinos GPIO

Capítulo 4

Sistema de Monitorização e Registo da Produção

4.1 Introdução

Durante a monitorização da produção da máquina de injeção, o número de peças produzidas constitui o dado mais crítico a ser contabilizado pelo sistema desenvolvido. Esta informação pode ser adquirida da máquina detetando os sinais descritos na secção 3.6. No entanto, a máquina não devolve toda a informação necessária para caracterizar a atividade da mesma. Eventos como mudança de molde, manutenção e falta de recursos não são sinalizados pela máquina mas precisam na mesma de ser registados. Portanto, foi codificado um programa em *Python* com uma componente gráfica que, para além de demonstrar a informação emitida pelos sinais, permite ao operador indicar estes e outros eventos mencionados. Depois de descrever em maior pormenor os requisitos do sistema, este capítulo descreve também o funcionamento de todos os processos e etapas do programa desenvolvido, incluindo os métodos de autenticação dos operadores e a estrutura da base de dados onde é registada toda a informação relacionada com as atividades de produção.



Figura 4.1: Sistema de Monitorização da Produção

4.2 Levantamento de Requisitos

Os moldes da máquina de injeção apresentam cavidades do lado esquerdo e do lado direito. Cada ordem de fabrico tem indicações de produção de peças orientadas apenas para um dos lados, portanto, o sistema deve ser capaz de processar duas ordens de fabrico em simultâneo, uma para peças injetadas no lado esquerdo do molde e outra para peças injetadas no lado direito. Isto não invalida que a produção não possa ocorrer com uma ordem de fabrico ativa somente para um dos lados. Adicionalmente, o sistema deve ser capaz de, pelo menos, indicar quantas peças aprovadas e rejeitadas estão a ser produzidas por cada ordem de fabrico, a duração do período real de produção e do período de inatividade, o modo de funcionamento da máquina (automático ou manual) e o estado atual da máquina, ou seja, se está em produção, parada, avariada, em manutenção ou em mudança de molde. Estes estados fazem parte de um conjunto de eventos definidos pela fábrica cuja integração no sistema também é necessária. Cada evento da tabela 4.1 divide-se em um ou mais subeventos com um código de identificação. Neste conjunto de subeventos, apenas as avarias são emitidas como sinais da máquina, os restantes devem ser acionados pelos operadores. Todos os dados de monitorização, quer sejam recolhidos da máquina ou acionados pelos operadores, devem ser registados para consulta e posterior análise.

Código	Evento	Subevento
0	Produção	Produção
2	Mudança Molde	Mudança de Molde
3.1	Avaria	Máquina
3.2	Avaria	Molde
3.3	Avaria	Robot
3.4	Avaria	Periféricos
4.1	Ensaio e Testes Planeados	Produção
4.2	Ensaio e Testes Planeados	Projetos Novos
4.3	Ensaio e Testes Planeados	Programação
4.4	Ensaio e Testes Planeados	Qualidade
4.5	Ensaio e Testes Planeados	Compras / Aproveitamentos
5.1	Interrupção de Produção	Periféricos
5.2	Interrupção de Produção	Problemas Qualidade
5.3	Interrupção de Produção	Limpeza
5.4	Interrupção de Produção	Aguarda Decisão
5.5	Interrupção de Produção	Espera Peças Metalizadas
5.6	Interrupção de Produção	Reunião / Formação
5.7	Interrupção de Produção	6S
6.1	Falta de Recursos	Matérias - Primas
6.2	Falta de Recursos	RH (Pessoas)
6.3	Falta de Recursos	Energia
6.4	Falta de Recursos	Componentes
6.5	Falta de Recursos	Ar Comprimido
9.1	Start Up e Set Down	Arranque Produção
9.2	Start Up e Set Down	Estufagem Matéria
9.3	Start Up e Set Down	Final Produção

Tabela 4.1: Lista de eventos e subeventos definidos na fábrica

Para chegar à conclusão de como o registo dos dados deve ser estruturado, foi consultado o sistema de monitorização em vigor. Assim, observou-se que os dados de produção de uma ordem de fabrico eram registados numa folha pelo operador responsável. Esta folha de registo pode ser consultada na imagem C.1 do anexo C. Este registo requer a identificação do número da ordem de fabrico das peças esquerdas e das peças direitas, a identificação da máquina, a data do registo e o operador responsável. Em termos de documentação efetiva de produção, é registado, em intervalos de tempo definidos, o número de peças aprovadas e rejeitadas. Adicionalmente, em caso de ausência temporária da linha de produção, por parte do operador, a identificação do operador substituto também deve ser registada. Paralelamente, o operador deve documentar numa folha adicional, visível na imagem C.2 do anexo C, os tipos de defeitos encontrados nas peças rejeitadas e as quantidades respetivas. Uma vez concluído o turno, as folhas de registo preenchidas são entregues ao departamento de gestão para que os dados sejam inseridos manualmente no sistema de informação. Esta prática origina uma grande acumulação de documentos que por vezes se tornam difíceis de gerir.

4.3 Estrutura e Funcionamento das Bases de Dados

Todos os dados de produção da máquina e eventos acionados pelo operador são registados numa base de dados presente num servidor local do *Raspberry Pi*. Porém, a monitorização da produção não é possível sem os dados das ordens de fabrico, presentes na base de dados central da fábrica. Adicionalmente, o acesso a esta base de dados também é necessário para a recolha de dados dos operadores, das peças, dos moldes e do equipamento. Ao contrário da base de dados local, a base de dados central é apenas acedida para recolher informação, ou seja, nunca sofre qualquer atualização por parte do sistema. Em suma, o programa acede a duas bases de dados diferentes, construídas em DBMS distintos. A base de dados local usa o sistema de gestão *open-source PostgreSQL* e a base de dados da fábrica está construída sobre o sistema *SQL Server*. No programa desenvolvido, o acesso a cada uma é feito segundo as bibliotecas *Psycopg2* [39] e *Pymssql* [40], respetivamente.

A comunicação com o servidor da base de dados central foi estabelecida configurando, em conjunto com o departamento de IT da fábrica, o acesso à rede via VPN. Este método foi útil durante o desenvolvimento do projeto porque permitia estabelecer uma ligação à base de dados a partir de qualquer rede. A configuração da VPN foi realizada segundo o protocolo PPTP com o auxílio do *PPTP Client* para *Debian* [41].

4.3.1 Estrutura da Base de Dados Central da Fábrica

A informação recolhida da base de dados central assenta em dados de identificação do operador, do molde, da máquina, da referência das peças a produzir e das ordens de fabrico.

Da tabela "utilizadores" são recolhidos apenas os dados necessários à identificação do operador autenticado no sistema. Esses dados estão presentes nos campos "numero", "name", "turno" e "rfid". O último campo refere-se ao número de identificação do cartão RFID do respetivo operador. A recolha da informação relativa à referência das peças e à máquina em questão é obtida acedendo às tabelas "referencias" e "equipamentos", respetivamente. Da primeira são recolhidos os dados dos campos "id_referencia" e "referencia" e da segunda os campos "id_equipamento" e "n_equipamento" (nome da máquina). Relativamente ao molde, a informação sobre o mesmo encontra-se na tabela "moldes" e desta são lidos os campos "id_molde", "n_molde" (nome do molde) e "cavidades". O último campo refere-se ao número de cavidades existentes na lateral de cada molde. A recolha de informação das ordens de fabrico é feita copiando na totalidade a tabela "ordensfabrico" para a base de dados local. A presença desta tabela localmente permite a atualização periódica dos seus dados sem que seja necessário escrever na base de dados central. No entanto, é importante que estes dados sejam também registados na tabela homónima da base de dados central, portanto, fica a cargo do administrador da mesma o acesso e recolha desta informação da base de dados local.

4.3.2 Estrutura da Base de Dados Local

Excluindo a tabela "ordensfabrico", todas as tabelas da base de dados local foram criadas especificamente para este projeto. As tabelas nesta base de dados foram divididas em dois grupos, o grupo das tabelas ativas e o grupo das tabelas estáticas.

O termo "tabelas estáticas" refere-se àquelas que apenas servem de consulta e cujos dados se mantêm inalterados. Neste grupo estão incluídas as tabelas "setup_rejeicoes" e "setup_eventos". As estruturas de ambas estão visíveis, respetivamente, nas tabelas 4.2 e 4.3. A primeira alberga toda a informação de todos os tipos de defeitos visíveis na tabela do anexo

Campo	Tipo
id	integer (primary key)
codigo_subevento	character varying (3)
evento	character varying (30)
subevento	character varying (200)

Tabela 4.2: Estrutura da tabela "setup_eventos"

Campo	Tipo
id	integer (primary key)
codigo_defeito	character varying (5)
descricao	character varying (50)

Tabela 4.3: Estrutura da tabela "setup_rejeicoes"

Os eventos adicionais, presentes na tabela 4.4, foram acrescentados à lista inicial por serem essenciais ao sistema desenvolvido. Os eventos "Fim de Turno", "Manutenção" e "Paragem" servem para sinalizar as ações com o mesmo nome e o evento "Login" foi dividido em dois subeventos, "Operador Planeado" e "Operador Substituto". Estes subeventos permitem distinguir se o operador autenticado é o responsável original pelas ordens de fabrico correntes ou se apenas veio substituir o primeiro durante um curto período de tempo. Por último, foi acrescentado um tipo de avaria denominada "Observada". Este subevento é acionado pelo operador quando uma avaria ocorre e o sinal da mesma, por alguma razão, não é detetado pelo sistema, ou quando este observa um tipo de avaria diferente das listadas na tabela 4.1.

Evento	Subevento
Avaria	Observada
Paragem	Paragem
Manutenção	Manutenção
Login	Operador Planeado
Login	Operador Substituto

Tabela 4.4: Lista de eventos adicionados aos eventos da tabela 4.1

As tabelas ativas são aquelas que são constantemente atualizadas com os dados provenientes da atividade de produção. Neste grupo inserem-se as restantes tabelas: "ordensfabrico", "ocorrencias", "rejeicoes" e "eventos".

A tabela "ordensfabrico", como tem origem na base de dados central, mantém a estrutura original apresentada em 4.5. Os campos "qtd_produzida_boas" e "qtd_produzida_mas", das ordens de fabrico em produção, são atualizados de hora em hora na base de dados local e esses dados são lidos no mesmo intervalo de tempo pela base de dados central.

Campo	Tipo
id_ordem_fabrico	integer (primary key)
date_registo	timestamp
registado_por	integer
id Equipamento	integer
id_referencia	integer
id_molde	integer
qtd_a_produzir	integer
tempo_ciclo_previsto	integer
qtd_por_caixa	integer
qtd_produzida_boas	integer
qtd_produzida_mas	integer
data_atualizacao	timestamp
data_prevista_inicio	timestamp
data_prevista_fim	timestamp
tempo_previsto	integer

Tabela 4.5: Estrutura da tabela "ordensfabrico"

No caso da tabela "ocorrencias", cuja estrutura está presente em 4.6, é inserida uma nova linha de dados sempre que é produzido um novo conjunto de peças. Os dados dos campos "id_ordem_fabrico", "id Equipamento", "id_referencia", "id_molde" e "qtd_a_produzir" são retirados da tabela "ordensfabrico". Já os campos "operador_numero" e "turno" são preenchidos com dados provenientes da tabela "utilizadores", presente na base de dados central. A hora e a data da ocorrencia foram separadas em dois campos diferentes para facilitar o tratamento destes dados no caso de apenas ser necessário a consulta de um deles.

Campo	Tipo
id	integer (primary key)
id_ordem_fabrico	integer
id Equipamento	integer
id_referencia	integer
id molde	integer
qtd_a_produzir	integer
qtd_produzida_boas	integer
qtd_produzida_mas	integer
operador_numero	character varying(30)
turno	character
data	character varying(30)
hora	character varying(30)

Tabela 4.6: Estrutura da tabela "ocorrencias"

A tabela "rejeicoes", com estrutura apresentada em 4.7, é atualizada, essencialmente, com o código e a quantidade de peças defeituosas contabilizadas pelo operador. O campo "codigo_defeito" é retirado da tabela "setup_rejeicoes" e o "nr_defeitos" é preenchido com dados recolhidos do sistema. Este processo é detalhado com maior pormenor na secção 4.4.3.

Campo	Tipo
id	integer (primary key)
id_ordem_fabrico	integer
codigo_defeito	character varying(10)
nr_defeitos	integer
operador_numero	character varying(30)
turno	character
data	character varying(30)
hora	character varying(30)

Tabela 4.7: Estrutura da tabela "rejeicoes"

Por último, os eventos são inseridos numa tabela com o mesmo nome, que é preenchida linha a linha sempre que um dos eventos presentes na tabela "setup_eventos" é acionado pelo operador. Os campos "codigo_subevento", "evento" e "subevento" são retirados da tabela "setup_eventos". Os dados dos campos "of_esq" e "of_dir" são preenchidos com os números das ordens de fabrico ativas. Finalmente, o campo "modo" é preenchido pelo termo *on* ou *off*, dependendo do evento ter sido ativado ou desativado. Este campo, no entanto, não se aplica a todos os eventos, algo que é esclarecido na secção 4.4.6.

Campo	Tipo
id	integer (primary key)
codigo_subevento	character varying(5)
evento	character varying(30)
subevento	character varying(30)
of_esq	character varying(5)
of_dir	character varying(5)
modo	character varying(5)
operador_numero	character varying(30)
turno	character
data	character varying(30)
hora	character varying(30)

Tabela 4.8: Estrutura da tabela "eventos"

4.4 Programa de Monitorização

4.4.1 Ambiente Gráfico

Definidos os critérios, a primeira etapa passou por desenhar o ambiente gráfico. No entanto, este componente do sistema foi sofrendo alterações durante o desenvolvimento do projeto conforme foram encontradas soluções mais apropriadas. Como mencionado na secção 2.4, o ambiente gráfico foi desenvolvido com o editor *QtDesigner*, que permite predefinir a posição dos *widgets*, a sua dimensão e folha de estilo. Construído o ambiente gráfico, o *software* gera um ficheiro que é importado no programa onde é codificado todo o sistema de monitorização. A figura 4.2 mostra a estruturação de parte do ambiente gráfico efetuada no *QtDesigner*.

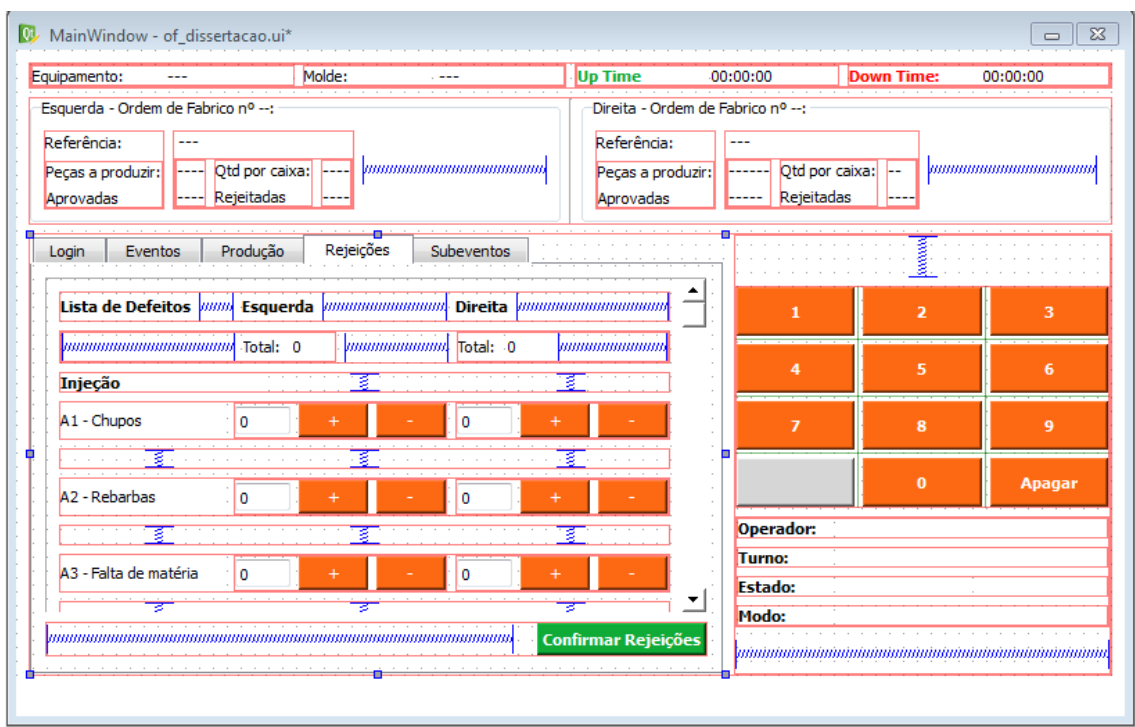


Figura 4.2: Estilo de estruturação do ambiente gráfico no *QtDesigner*

Tendo em conta os requisitos levantados, foi construída a versão final do ambiente gráfico que é apresentado na figura 4.3 e que está dividido em quatro grupos principais:

- 1- Dados da Produção
- 2- Ações de Controlo do Sistema
- 3- Teclado Numérico Virtual
- 4- Dados do Operador e Estado da Máquina

The screenshot displays a production monitoring interface with the following components:

- Callout 1:** Top status bar showing "Equipamento: MO-INJ-01", "Molde: ST0001", "Up Time: 00:01:08", and "Down Time: 00:00:26".
- Callout 2:** "Acções" menu with buttons for "Iniciar Produção", "Substituição", "Fim de Turno", "Parar Máquina", "Mudança de Molde", "Manutenção", "Avaria", and "SUBEVENTOS".
- Callout 3:** A numeric keypad with buttons 1-9, 0, and "Apagar".
- Callout 4:** Operator and machine status: "Operador: Paulo Araujo", "Turno: 1", "Estado: Máquina em Produção.", and "Modo: Automático".

Below the main interface, there is a detailed view of the production data group:

Equipamento: MO-INJ-01	Molde: ST0001	Up Time	00:01:08	Down Time:	00:00:26
Esquerda - Ordem de Fabrico nº 8:			Direita - Ordem de Fabrico nº 9:		
Referência:	1ABM100100	Referência:	1ABM100100		
Peças a produzir:	2000	Qtd por caixa:	50	Peças a produzir:	1000
		Qtd por caixa:	100		
Aprovadas	70	Rejeitadas	25	Aprovadas	70
		Rejeitadas	25		

Figura 4.3: Diferentes Grupos do ambiente gráfico desenvolvido

Como podem ser iniciadas duas ordens de fabrico ao mesmo tempo, uma esquerda e uma direita, no grupo "Dados da Produção" foram criados dois campos de informação para cada uma delas. Essa informação inclui a quantidade de peças a produzir, a quantidade de peças a distribuir por caixa, o número de peças aprovadas e o número de peças rejeitadas. Adicionalmente, na parte superior do grupo, é carregada a informação relativa ao molde e ao equipamento, que são sempre comuns a ambas as ordens de fabrico. Na mesma linha, é contabilizado o tempo de atividade (*uptime*) e o tempo de inatividade (*downtime*). Este grupo está sempre visível para que toda esta informação seja imediatamente consultada.

Equipamento: MO-INJ-01	Molde: ST0001	Up Time	00:01:08	Down Time:	00:00:26
Esquerda - Ordem de Fabrico nº 8:			Direita - Ordem de Fabrico nº 9:		
Referência:	1ABM100100	Referência:	1ABM100100		
Peças a produzir:	2000	Qtd por caixa:	50	Peças a produzir:	1000
		Qtd por caixa:	100		
Aprovadas	70	Rejeitadas	25	Aprovadas	70
		Rejeitadas	25		

Figura 4.4: Grupo Dados de Produção

Outro grupo, cujos dados também se encontram sempre visíveis, é o grupo "Dados do Operador e Estado da Máquina" que indica o nome do operador autenticado no programa, o turno atual, o estado da máquina e o modo em que se encontra (automático ou manual).

Operador: Paulo Araújo
Turno: 1
Estado: Máquina em Produção.
Modo: Automático

Figura 4.5: Grupo Dados do Operador e Estado da Máquina

O grupo "Ações de Controlo do Sistema" é composto por uma tabulação constituída por cinco separadores. A opção de dividir as ações de controlo em vários separadores foi tomada para que todas estas ações fossem contidas no mesmo ambiente gráfico. Cada um destes separadores apresenta funções distintas. No separador "Login" (figura 4.6), ocorrem todas as ações de autenticação por parte do operador. O método preferencial de autenticação deste sistema é via leitura de cartões RFID, no entanto, na ausência desse cartão, a autenticação pode ser feita através da indicação do número de identificação do operador, com o auxílio do teclado numérico virtual da figura 4.7.

Login | Eventos | Produção | Rejeições | Subeventos

Operador: Confirmar

Cartão Detectado.
Autenticado com sucesso!

SAIR

Figura 4.6: Separador "Login"

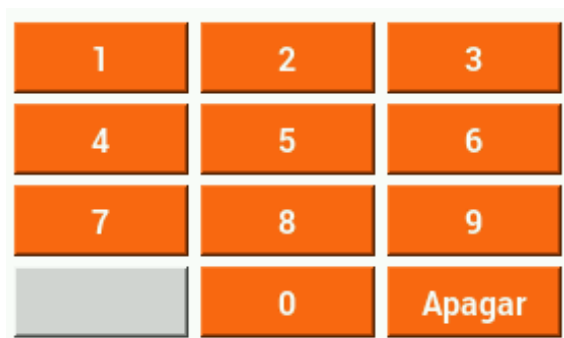


Figura 4.7: Teclado Numérico Virtual

No separador "Eventos" (figura 4.8) encontram-se agrupados os eventos cujo acesso necessita de ser mais imediato. O conjunto "Alarmes" é composto por indicadores de deteção dos quatro tipos de avarias e o conjunto "Acções" suporta os eventos de maior relevância. O botão "Subeventos" direciona para o separador com o mesmo nome (figura 4.9), onde o operador tem acesso aos restantes eventos, "Ensaio e Testes Planeados", "Interrupção de Produção" e "Falta de Recursos".

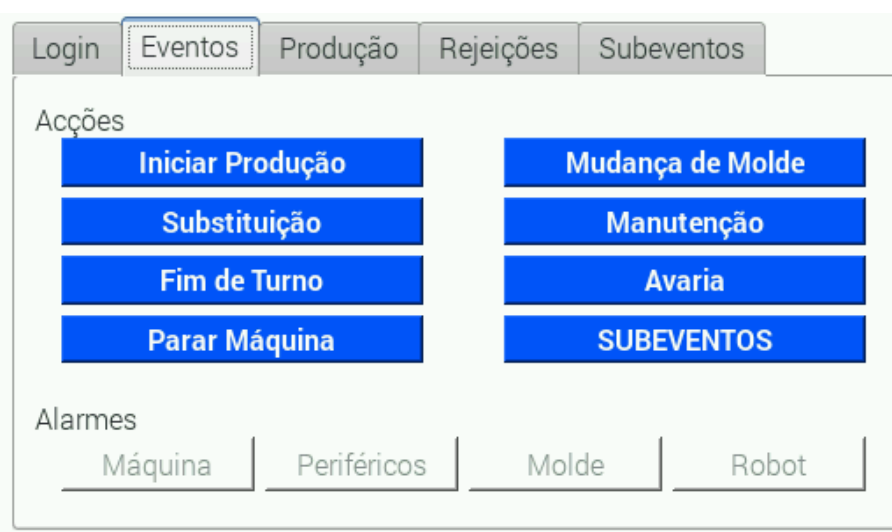


Figura 4.8: Separador "Eventos"

Ensaaios e Testes Planeados

4.1 - Produção OFF

4.2 - Projetos Novos OFF

4.3 - Programação OFF

Figura 4.9: Separador "Subeventos"

Ainda dentro do grupo de "Ações de Controlo do Sistema", a seleção do botão "Iniciar Produção" remete para o separador "Produção" (figura 4.10) onde são definidos os dados necessários para dar início às ordens de fabrico. Por fim, os possíveis defeitos, quer das peças esquerdas, quer das peças direitas, e a respetiva quantidade, são definidos no separador "Rejeições" (figura 4.11). Neste estão listados todos os tipos de defeitos e o utilizador preenche os campos de entrada com o número de peças contabilizadas de cada tipo de defeito observado.

Ordem de Fabrico Esquerda

Ordem de Fabrico nº: OK

Nº cavidades reais: OK

Encerrar Ordem de Fabrico

Ordem de Fabrico Direita

Ordem de Fabrico nº: OK

Nº cavidades reais: OK

Encerrar Ordem de Fabrico

Figura 4.10: Separador "Produção"

Lista de Defeitos	Esquerda		Direita	
	Total: 0		Total: 0	
Injeção				
A1 - Chupos	0	+ -	0	+ -
A2 - Rebarbas	0	+ -	0	+ -
Confirmar Rejeições				

Figura 4.11: Separador "Rejeições"

4.4.2 Métodos de Autenticação

Assim que o programa é iniciado, o utilizador depara-se com o separador "Login", sendo obrigado a autenticar-se. Para impedir que qualquer outra ação seja iniciada sem que o operador se identifique primeiro, todos os restantes separadores mantêm-se bloqueados (figura 4.12).

Equipamento: --- Molde: ---

Esquerda - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ---- Qtd por caixa: ----
Aprovadas ---- Rejeitadas ----

Up Time 00:00:00 **Down Time:** 00:00:51

Direita - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ----- Qtd por caixa: --
Aprovadas ----- Rejeitadas ----

Operador: **Confirmar**

Passe o cartão...

SAIR

1	2	3
4	5	6
7	8	9
	0	Apagar

Operador:
Turno:
Estado: **Máquina Parada.**
Modo: Automático

Figura 4.12: Estado inicial de autenticação

Enquanto o programa se encontra neste separador, o operador tem duas alternativas de autenticação, via aproximação ao leitor RFID de um cartão de identificação ou por inserção do seu número de identificação interno.

Analisando primeiro a segunda alternativa, na figura 4.13 está presente um diagrama de estados que demonstra como foi estruturado este algoritmo de autenticação. No programa, as funções de autenticação recebem um parâmetro que determina o tipo de autenticação a ser feita. Este parâmetro pode ser "login_planeado", "login_substituto" ou "confirmar_operador" e cada um deles refere-se a uma situação distinta. No caso do arranque da aplicação, ou seja, quando o operador se autentica pela primeira vez, o parâmetro transmitido é a expressão "login_planeado". As restantes expressões serão esclarecidas mais à frente.

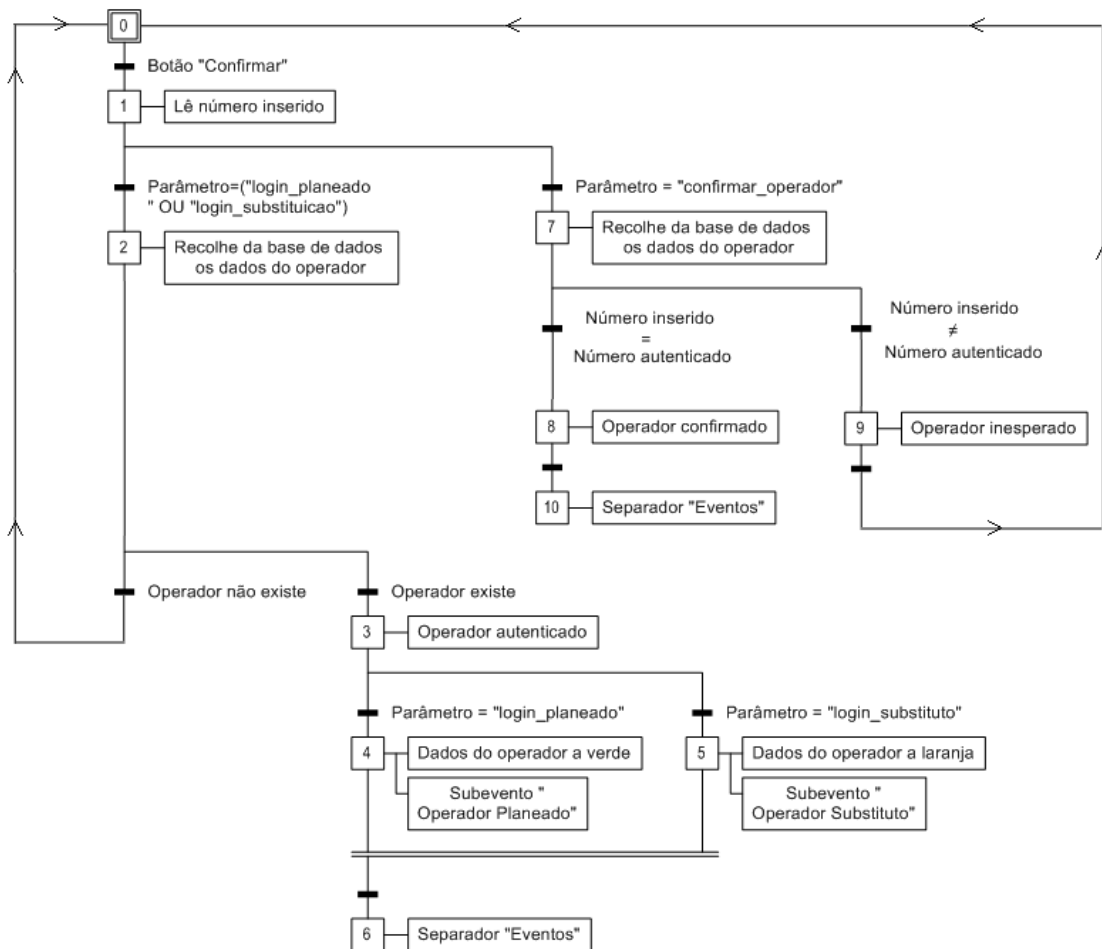


Figura 4.13: *Grafset* do método de autenticação por indicação do número de operador

Assim que o botão "Confirmar" for seleccionado, é guardado o número de identificação do operador inserido no campo de autenticação. De seguida, no caso do parâmetro "login_planeado", é acedida à tabela "utilizadores" da base de dados central e recolhidos os campos "name" e "turno"

da linha da tabela cujo campo "numero" é igual ao número de autenticação inserido pelo operador. Se o número não estiver presente na tabela é devolvida a mensagem "Utilizador Inexistente" (figura 4.14) e o processo de autenticação é interrompido e reiniciado. Caso contrário, é devolvida a mensagem "Autenticado com sucesso!" (figura 4.15).

The screenshot displays a production monitoring interface. At the top, it shows equipment and mold information: "Equipamento: --- Molde: ---". It also displays "Up Time" as 00:00:00 and "Down Time" as 00:00:15. Below this, there are two columns for production orders: "Esquerda - Ordem de Fabrico nº --:" and "Direita - Ordem de Fabrico nº --:". Each column lists "Referência: ---", "Peças a produzir: ---- Qtd por caixa: ----", and "Aprovadas ---- Rejeitadas ----".

The main interface has a navigation bar with tabs: "Login", "Eventos", "Produção", "Rejeições", and "Subeventos". The "Login" tab is active, and a login form is displayed. The form includes a text input field for the operator ID, which contains the value "133", and a green "Confirmar" button. Below the input field, there is a green prompt "Passe o cartão..." and a red error message "Utilizador inexistente." At the bottom left of the login form is a red "SAIR" button.

To the right of the login form is a numeric keypad with buttons for digits 1-9, 0, and "Apagar". Below the keypad, the system status is shown: "Operador:", "Turno:", "Estado: Máquina Parada.", and "Modo: Automático".

Figura 4.14: Mensagem "Utilizador inexistente" no separador "Login"

Equipamento: --- Molde: --- **Up Time** 00:00:00 **Down Time:** 00:02:25

Esquerda - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ---- Qtd por caixa: ----
Aprovadas ---- Rejeitadas ----

Direita - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ---- Qtd por caixa: --
Aprovadas ---- Rejeitadas ----

Login Eventos Produção Rejeições Subeventos

Operador: **Confirmar**

Passa o cartão....
Autenticado com sucesso!

SAIR

1	2	3
4	5	6
7	8	9
	0	Apagar

Operador: Paulo Araújo
Turno: 1
Estado: Máquina Parada.
Modo: Automático

Figura 4.15: Mensagem "Autenticado com sucesso!" no separador "Login"

Confirmada a existência dos dados do operador, tanto o nome do mesmo como o turno de trabalho são carregados no canto inferior direito da aplicação, no grupo "Dados do Operador e Estado da Máquina".

Para finalizar o processo, é chamada a função *update_database_eventos* para inserir na tabela "eventos" uma linha com o evento "Login" e subevento "Operador Planeado" (figura 4.16). A estrutura desta função e de outras semelhantes é esclarecida na secção 4.4.6. O evento "Login" faz parte do grupo de eventos para os quais o campo "modo" não é relevante, portanto, encontra-se preenchido com o termo "NULL".

Ações	id	codigo_subevento	evento	subevento	of_esq	of_dir	modo	operador_numero	turno	data	hora
Editar Deletar	0	---	Login	Operador Planeado	NULL	NULL	NULL	159	1	2016-06-11	16:38:38

Figura 4.16: Entrada na tabela "eventos" do subevento "Operador Planeado" após autenticação

O método de autenticação descrito até agora é iniciado assim que o botão "Confirmar" é premido. No entanto, o método de autenticação por RFID é acionado a cada segundo, ou seja, assim que o utilizador é direcionado para o separador "Login", reinicia um temporizador que a cada segundo procura detetar um cartão RFID. Se nesse instante não for detetado nenhum cartão, a leitura é interrompida e reiniciada no segundo seguinte. Caso um cartão seja detetado, o processo de autenticação é iniciado e no final deste, o temporizador é desligado. Este algoritmo de leitura de *tags* RFID com o auxílio de um controlador de tempo foi utilizado para que a aplicação permitisse

em simultâneo os dois métodos de autenticação. Se o ciclo de leitura dos cartões não fosse interrompido a cada segundo, o botão "Confirmar" só poderia ser premido após a deteção de um cartão RFID, o que era impraticável visto que o primeiro método foi implementado como alternativa ao segundo e, portanto, ambos devem ser acessíveis ao mesmo tempo.

Assim que um cartão é detetado, o método de autenticação, visível no *grafcet* da figura 4.17, segue uma linha muito semelhante à do método alternativo. A única diferença incide no principal agente de autenticação, ou seja, se no método anterior, o dado de autenticação é o número de identificação do operador, neste método esse dado é substituído pelo número de identificação do cartão RFID do operador, isto é, são acedidos aos campos "name" e "turno" da tabela utilizadores da linha cujo campo "rfid" é igual ao número de identificação do cartão detetado. Este número é lido com o auxílio da biblioteca MFRC522-python [42].

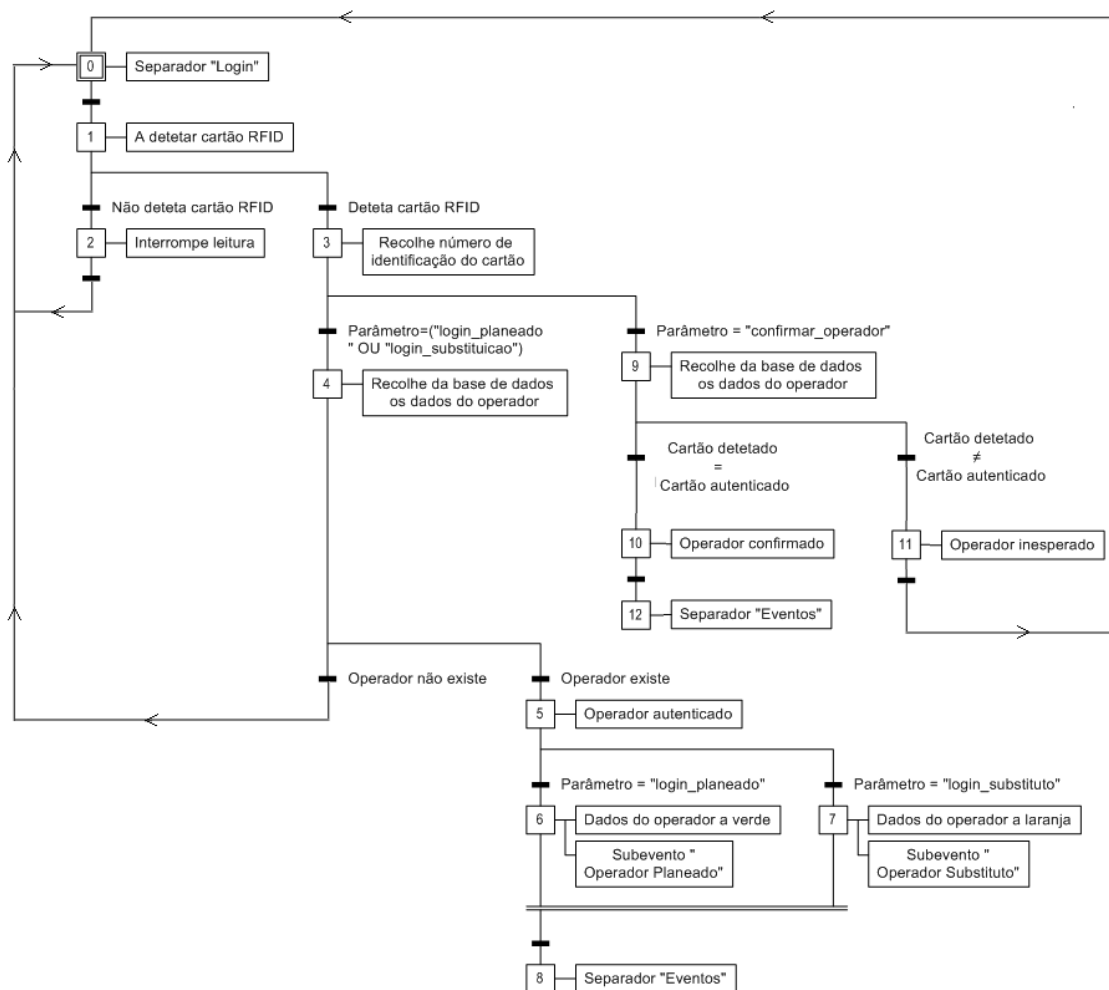


Figura 4.17: Grafcet do método de autenticação por cartão RFID

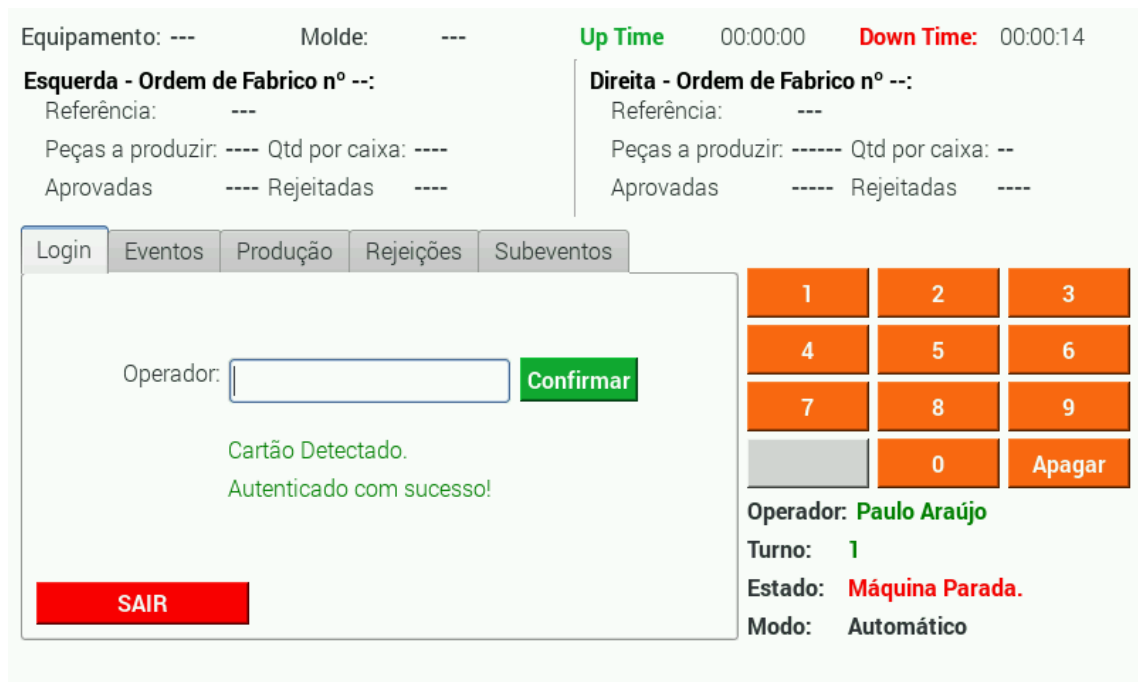


Figura 4.18: Mensagem "Cartão Detetado. Autenticado com sucesso!" no separador "Login"

Adicionalmente, foi implementada uma medida de segurança para impedir que um operador clandestino interaja com o sistema em nome do operador autenticado. Esta medida consiste em requisitar uma nova autenticação ao fim de 3 minutos. Enquanto o operador não confirmar a sua autenticação, o acesso aos restantes separadores encontra-se bloqueado. Assim, sempre que é realizada qualquer tipo de autenticação, é inicializado um temporizador que ao fim de 180 segundos requer a confirmação do operador autenticado.

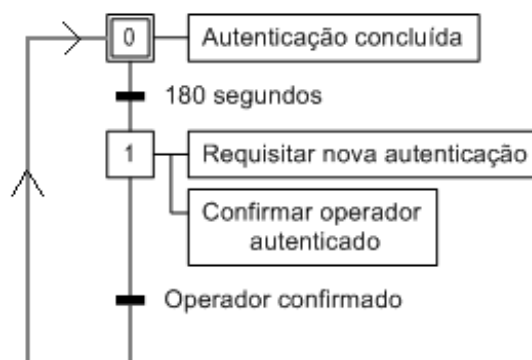


Figura 4.19: Grafcet do "timeout da autenticação"

Neste processo, esquematizado no *grafcet* da figura 4.19, o parâmetro recebido pelas funções de *login* é a supracitada expressão "confirmar_operador", cuja sequência de estados resultante

deste parâmetro pode ser consultada nos *grafcets* dos dois métodos de autenticação já apresentados (figuras 4.13 e 4.17). Neste processo, quando o operador aproxima o cartão RFID do leitor, é confirmado se os dados associados ao cartão de identificação são os mesmos do utilizador já autenticado. Em caso afirmativo é devolvida a mensagem "Autenticado com sucesso!" e os restantes separadores são desbloqueados. Caso contrário, é devolvida a mensagem "Operador Inesperado" (figura 4.20) e os separadores mantêm-se bloqueados com o programa à espera da autenticação correta.

The screenshot displays a production control interface with the following elements:

- Equipment and Mold Info:** Equipamento: MO-INJ-01, Molde: ST0001, Up Time: 00:03:12, Down Time: 00:02:50.
- Production Statistics:**
 - Esquerda - Ordem de Fabrico nº 8:** Referência: 1ABM100100, Peças a produzir: 2000, Qtd por caixa: 50, Aprovadas: 85, Rejeitadas: 5.
 - Direita - Ordem de Fabrico nº 9:** Referência: 1ABM100100, Peças a produzir: 1000, Qtd por caixa: 100, Aprovadas: 85, Rejeitadas: 5.
- Navigation Tabs:** Login (active), Eventos, Produção, Rejeições, Subeventos.
- Login Form:**
 - Operator input field:
 - Confirmar button (green)
 - Message: **Cartão Detectado.** (green), **Operador Inesperado** (red)
 - SAIR button (red)
- Numeric Keypad:** Buttons for digits 1-9, 0, and Apagar.
- Status Summary:**
 - Operador: Paulo Araújo
 - Turno: 1
 - Estado: Máquina em Produção.
 - Modo: Automático

Figura 4.20: Mensagem "Cartão Detetado. Operador inesperado." no separador "Login"

Adicionalmente, é importante mencionar que existem outras ações que podem iniciar processos de autenticação. Isto é o caso da seleção dos botões "Fim de Turno" e "Substituição" do separador "Eventos". A seleção do botão "Fim de Turno" origina a entrada no processo de autenticação com o parâmetro "login_planeado", permitindo a troca do operador atual por outro do turno seguinte. Quando o operador necessita de se ausentar do processo de produção durante um período de tempo, é obrigado a garantir que outro operador o substitua durante essa ausência. Esta substituição tem de ficar registada e portanto o operador substituto deve ativar o botão "Substituição". Neste caso, o processo de autenticação segue a linha do parâmetro "login_substituicao" que origina a atualização dos dados do operador a cor laranja, indicando que o operador autenticado se encontra a substituir o original (figura 4.21). Quando é acionado este parâmetro, a função *update_database_eventos* insere na tabela "eventos" uma linha com o subevento "Operador Substituto" (figura 4.22).

Operador: Paulo Rocha
Turno: N
Estado: Máquina em Produção.
Modo: Automático

Figura 4.21: Grupo "Dados do Operador e Estado da Máquina" atualizado com os dados do operador substituto

Editar Deletar 3 --- Login Operador Substituto 8 9 NULL 326 (substituto) N 2016-06-11 16:49:14

Figura 4.22: Entrada na tabela "eventos" do subevento "Operador Substituto"

Ao mesmo tempo, assim que o botão "Substituição" é pressionado, o seu aspeto visual altera-se para o da figura 4.23 com o objetivo de realçar a importância de indicar o fim da substituição. Assim, aquando do seu regresso, o operador original deve selecionar o botão "Terminar Substituição" e autenticar-se novamente. Esta ação é registada de novo na tabela "eventos" com o subevento "Operador Planeado" (figura 4.24).

Equipamento: MO-INJ-01 Molde: ST0001 **Up Time** 00:00:28 **Down Time:** 00:00:18

Esquerda - Ordem de Fabrico nº 8:
 Referência: 1ABM100100
 Peças a produzir: 2000 Qtd por caixa: 50
 Aprovadas 50 Rejeitadas 10

Direita - Ordem de Fabrico nº 9:
 Referência: 1ABM100100
 Peças a produzir: 1000 Qtd por caixa: 100
 Aprovadas 50 Rejeitadas 10

Login Eventos Produção Rejeições Subeventos

Ações
 Iniciar Produção Mudança de Molde
Terminar Substituição Manutenção
 Fim de Turno Avaria
 Parar Máquina SUBEVENTOS

Alarmes
 Máquina Periféricos Molde Robot

1 2 3
 4 5 6
 7 8 9
 0 Apagar

Operador: Paulo Rocha
Turno: N
Estado: Máquina em Produção.
Modo: Automático

Figura 4.23: Botão "Substituição" alterado para "Terminar Substituição"

Editar	Deletar	3 ---	Login	Operador Substituto	8	9	NULL	326 (substituto)	N	2016-06-11	16:49:14
Editar	Deletar	4 ---	Login	Operador Planeado	8	9	NULL	159	1	2016-06-11	16:50:57

Figura 4.24: Tabela "eventos" com entradas dos subeventos "Operador Planeado" e "Operador Substituto"

4.4.3 Processamento da Atividade de Produção

As ações de controlo relevantes para as ordens de fabrico encontram-se principalmente nos separadores "Produção" e "Rejeições", cujos *widgets* se mantêm desativados enquanto nenhuma ordem de fabrico se encontrar ativa (figura 4.25).

Equipamento: --- Molde: --- **Up Time** 00:00:00 **Down Time:** 00:00:14

Esquerda - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ---- Qtd por caixa: ----
Aprovadas ---- Rejeitadas ----

Direita - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ---- Qtd por caixa: --
Aprovadas ---- Rejeitadas ----

Login Eventos **Produção** Rejeições Subeventos

Ordem de Fabrico Esquerda
Ordem de Fabrico nº:
Nº cavidades reais:

Ordem de Fabrico Direita
Ordem de Fabrico nº:
Nº cavidades reais:

1	2	3
4	5	6
7	8	9
	0	Apagar

Operador: Paulo Araújo
Turno: 1
Estado: Máquina Parada.
Modo: Automático

(a) Separador "Produção" com os *widgets* bloqueados

Equipamento: --- Molde: --- **Up Time** 00:00:00 **Down Time:** 00:01:28

Esquerda - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ---- Qtd por caixa: ----
Aprovadas ---- Rejeitadas ----

Direita - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ---- Qtd por caixa: --
Aprovadas ---- Rejeitadas ----

Login Eventos Produção **Rejeições** Subeventos

A12 - Diferença de cor

A13 - Deformação

Metalização e Pintura:

B1 - Falta de tinta

B2 - Borbotos

1	2	3
4	5	6
7	8	9
	0	Apagar

Operador: Paulo Araújo
Turno: 1
Estado: Máquina Parada.
Modo: Automático

(b) Separador "Rejeições" com os *widgets* bloqueados

Figura 4.25: Separadores "Produção" e "Rejeições" com os *widgets* bloqueados

Os processos da atividade de produção descritos nesta secção estão esquematizados no diagrama de estados da figura 4.26.

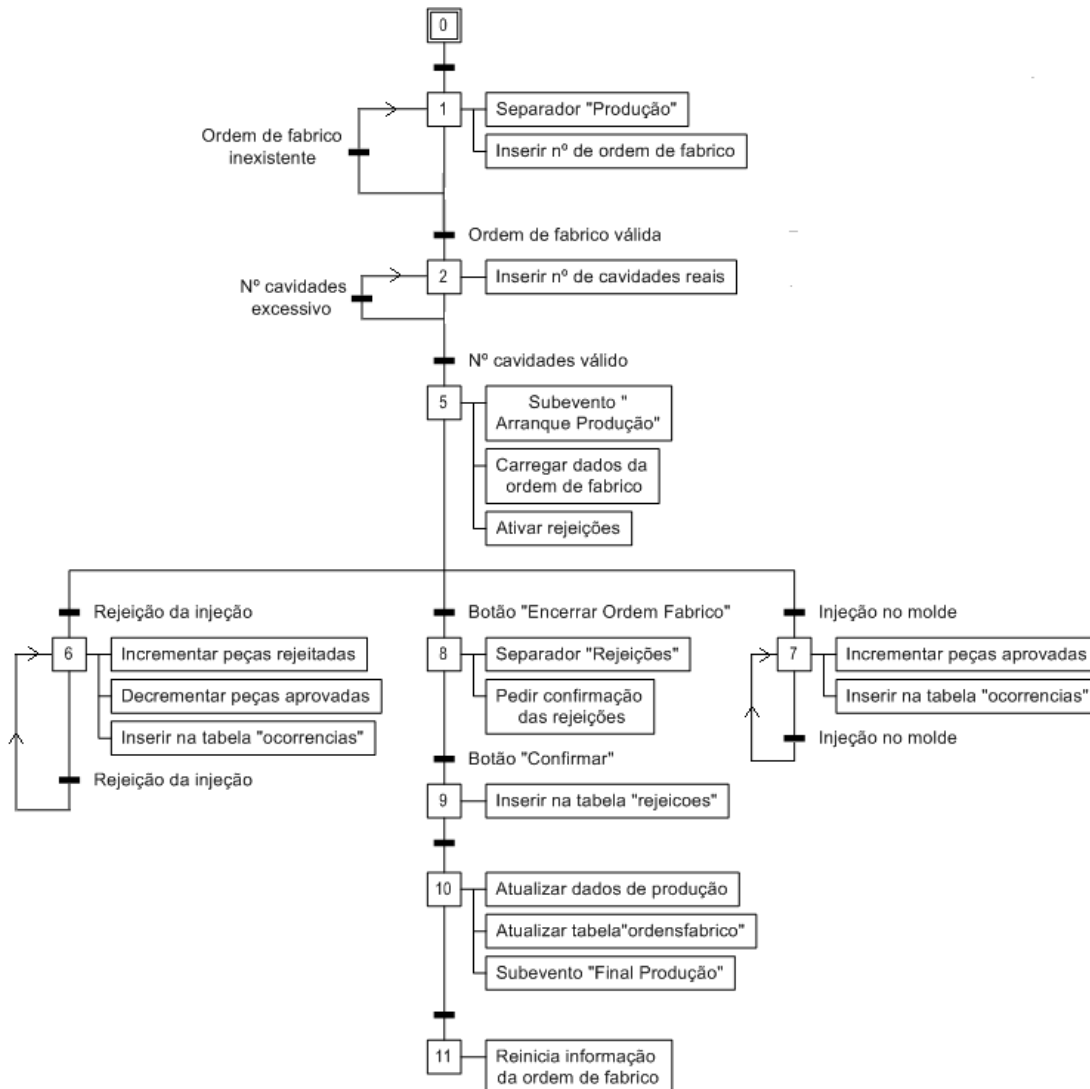


Figura 4.26: Grafset das ações de processamento de uma ordem de fabrico

Para ativar o processo de arranque das ordens de fabrico, o operador precisa de seleccionar o botão "Iniciar Produção" do separador "Eventos". Este botão está conectado à função *iniciar_producao*, que começa por direccionar o operador para o separador "Produção". Imediatamente a seguir, são desbloqueados os campos "Ordem de Fabrico nº" e os respetivos botões de confirmação (figura 4.27). Esta função está preparada para ativar apenas um desses campos caso a ordem de fabrico do lado oposto ainda se encontre ativa. Na figura 4.28 observa-se que o botão "Iniciar Produção" apenas desbloqueou o campo "Ordem de Fabrico Direita" porque a ordem de fabrico esquerda ainda não tinha sido encerrada.

Equipamento: --- Molde: --- **Up Time** 00:00:00 **Down Time:** 00:06:58

Esquerda - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ---- Qtd por caixa: ----
Aprovadas ---- Rejeitadas ----

Direita - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ---- Qtd por caixa: --
Aprovadas ---- Rejeitadas ----

Login Eventos **Produção** Rejeições Subeventos

Ordem de Fabrico Esquerda
Ordem de Fabrico nº: **OK**
Nº cavidades reais: **OK**
Encerrar Ordem de Fabrico

Ordem de Fabrico Direita
Ordem de Fabrico nº: **OK**
Nº cavidades reais: **OK**
Encerrar Ordem de Fabrico

1	2	3
4	5	6
7	8	9
	0	Apagar

Operador: Paulo Araújo
Turno: 1
Estado: Máquina Parada.
Modo: Automático

Figura 4.27: Campos "Ordem de Fabrico nº" desbloqueados

Equipamento: **MO-INJ-06** Molde: **ST0010** **Up Time** 00:01:13 **Down Time:** 00:00:17

Esquerda - Ordem de Fabrico nº 5:
Referência: **1ANJ101100**
Peças a produzir: **4500** Qtd por caixa: ---
Aprovadas **57** Rejeitadas **6**

Direita - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ---- Qtd por caixa: --
Aprovadas ---- Rejeitadas ----

Login Eventos **Produção** Rejeições Subeventos

Ordem de Fabrico Esquerda
Ordem de Fabrico nº: **OK**
Nº cavidades reais: **OK**
Encerrar Ordem de Fabrico

Ordem de Fabrico Direita
Ordem de Fabrico nº: **OK**
Nº cavidades reais: **OK**
Encerrar Ordem de Fabrico

1	2	3
4	5	6
7	8	9
	0	Apagar

Operador: Paulo Araújo
Turno: 1
Estado: Máquina em Produção.
Modo: Automático

Figura 4.28: Campo direito ativado e campo esquerdo bloqueado no separador "Produção"

De seguida, o programa entra na função *ordens_fabrico* que recolhe o campo "id_ordem_fabrico" de todas as linhas da tabela "ordensfabrico", ou seja, regista numa lista todos os números de todas

as ordens de fabrico existentes e conecta o botão "OK", referente ao campo "Ordem de Fabrico nº", à função *cavidades*. Ou seja, assim que o operador indicar, com o auxílio do teclado numérico, o número da ordem de fabrico a iniciar e premir esse botão, o programa processa a função *cavidades*. Esta função começa por ler o número da ordem de fabrico inserido e verifica se este se encontra dentro da lista recolhida. Caso a resposta seja negativa, devolve a mensagem "Ordem de fabrico inexistente." (figura 4.29). Caso seja positiva, ativa o campo "Nº cavidades reais" e o respetivo botão de confirmação "OK" (figura 4.30).

Equipamento: ---	Molde: ---	Up Time 00:00:00	Down Time: 00:00:29												
Esquerda - Ordem de Fabrico nº --:		Direita - Ordem de Fabrico nº --:													
Referência: ---		Referência: ---													
Peças a produzir: ---- Qtd por caixa: ----		Peças a produzir: ----- Qtd por caixa: --													
Aprovadas ---- Rejeitadas ----		Aprovadas ----- Rejeitadas ----													
Login	Eventos	Produção	Rejeições												
Ordem de Fabrico Esquerda		Ordem de Fabrico Direita													
Ordem de Fabrico nº:	<input type="text" value="12"/> OK	Ordem de Fabrico nº:	<input type="text"/> OK												
Nº cavidades reais:	<input type="text"/> OK	Nº cavidades reais:	<input type="text"/> OK												
Encerrar Ordem de Fabrico		Encerrar Ordem de Fabrico													
Ordem de fabrico inexistente.															
		<table border="1"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> <tr><td>7</td><td>8</td><td>9</td></tr> <tr><td></td><td>0</td><td>Apagar</td></tr> </table>		1	2	3	4	5	6	7	8	9		0	Apagar
1	2	3													
4	5	6													
7	8	9													
	0	Apagar													
		Operador: Paulo Araújo Turno: 1 Estado: Máquina Parada. Modo: Automático													

Figura 4.29: Mensagem "Ordem de fabrico inexistente." no separador "Produção"

Equipamento: --- Molde: --- **Up Time** 00:00:00 **Down Time:** 00:01:47

Esquerda - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ---- Qtd por caixa: ----
Aprovadas ---- Rejeitadas ----

Direita - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ---- Qtd por caixa: --
Aprovadas ---- Rejeitadas ----

Login Eventos **Produção** Rejeições Subeventos

Ordem de Fabrico Esquerda	Ordem de Fabrico Direita
Ordem de Fabrico nº: <input type="text" value="8"/> <input type="button" value="OK"/>	Ordem de Fabrico nº: <input type="text"/> <input type="button" value="OK"/>
Nº cavidades reais: <input type="text" value="5"/> <input type="button" value="OK"/>	Nº cavidades reais: <input type="text"/> <input type="button" value="OK"/>
<input type="button" value="Encerrar Ordem de Fabrico"/>	<input type="button" value="Encerrar Ordem de Fabrico"/>

1	2	3
4	5	6
7	8	9
	0	Apagar

Operador: Paulo Araújo
Turno: 1
Estado: Máquina Parada.
Modo: Automático

Figura 4.30: Campo "Nº cavidades reais" ativo no separador "Produção"

A tabela "moldes" da base de dados central tem o registo da quantidade de cavidades presentes em cada lado do molde. Se o número de cavidades de cada lado for três, por exemplo, cada injeção no molde resulta na produção de seis peças, três peças esquerdas e três peças direitas. No entanto, este número de cavidades é apenas um valor teórico. Este valor pode não coincidir com o valor real caso uma das cavidades esteja danificada. Nesta situação, essa cavidade não vai sofrer uma injeção e, portanto, o número de peças produzidas vai ser inferior. Assim, como a base de dados apenas devolve o número de cavidades teórico, é necessário questionar o operador sobre o número real de cavidades do lado referente à ordem de fabrico a iniciar. Este processo começa com a função *cavidades* a recolher o campo "id_molde" referente à "id_ordem_fabrico" indicada da tabela "ordensfabrico". De seguida acede à tabela "moldes" para ler o campo "cavidades" da entrada do "id_molde" recolhido. Posteriormente, carrega o valor do número de cavidades teóricas do molde no campo "Nº cavidades reais" e dá a hipótese ao operador de registar este como o número de cavidades reais ou alterar para outro valor. O botão "OK" referente a este campo é conectado à função *actualiza_dados*. Esta função recebe como parâmetros o número da ordem de fabrico inserido, o número de cavidades reais do molde e qual a orientação das peças a produzir, esquerda ou direita. A primeira ação da função passa por confirmar a validade do número de cavidades reais inserido, ou seja, se o operador indicar erradamente um número de cavidades reais superior ao número de cavidades teóricas é devolvida a mensagem "Nº excessivo de cavidades." (figura 4.31), visto que este número nunca pode ser inferior ao primeiro.

Equipamento: --- Molde: --- **Up Time** 00:00:00 **Down Time:** 00:01:01

Esquerda - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ---- Qtd por caixa: ----
Aprovadas ---- Rejeitadas ----

Direita - Ordem de Fabrico nº --:
Referência: ---
Peças a produzir: ----- Qtd por caixa: --
Aprovadas ---- Rejeitadas ----

Login Eventos **Produção** Rejeições Subeventos

Ordem de Fabrico Esquerda	Ordem de Fabrico Direita
Ordem de Fabrico nº: <input type="text" value="8"/> <input type="button" value="OK"/>	Ordem de Fabrico nº: <input type="text"/> <input type="button" value="OK"/>
Nº cavidades reais: <input type="text" value="9"/> <input type="button" value="OK"/>	Nº cavidades reais: <input type="text"/> <input type="button" value="OK"/>
<input type="button" value="Encerrar Ordem de Fabrico"/>	<input type="button" value="Encerrar Ordem de Fabrico"/>

Nº excessivo de cavidades.

1	2	3
4	5	6
7	8	9
	0	Apagar

Operador: Paulo Araújo
Turno: 1
Estado: Máquina em Produção.
Modo: Automático

Figura 4.31: Mensagem "Nº excessivo de cavidades" no separador "Produção"

Porém, caso esta condição não aconteça, a função *actualiza_dados* ativa o botão "Encerrar Ordem de Fabrico" e desativa todos os campos e botões anteriores para não permitir que o operador os altere enquanto a produção dessa ordem de fabrico decorre. De seguida carrega no campo "Dados da Produção" da aplicação os dados da ordem de fabrico iniciada. Estes dados incluem a designação do equipamento e do molde, a referência da peça a produzir, o número de peças a produzir, a quantidade de peças a inserir por caixa, o número de peças aprovadas e o número de peças rejeitadas. O estado do sistema após este processo é visível na figura 4.32, onde foram iniciadas as ordens de fabrico de número 8 e 9.

Equipamento: MO-INJ-01 Molde: ST0001 Up Time 00:00:00 Down Time: 00:01:07

Esquerda - Ordem de Fabrico nº 8:
Referência: 1ABM100100
Peças a produzir: 2000 Qtd por caixa: 50
Aprovadas 0 Rejeitadas 0

Direita - Ordem de Fabrico nº 9:
Referência: 1ABM100100
Peças a produzir: 1000 Qtd por caixa: 100
Aprovadas 0 Rejeitadas 0

Login Eventos **Produção** Rejeições Subeventos

Ordem de Fabrico Esquerda
Ordem de Fabrico nº: 8 OK
Nº cavidades reais: 5 OK
Encerrar Ordem de Fabrico

Ordem de Fabrico Direita
Ordem de Fabrico nº: 9 OK
Nº cavidades reais: 5 OK
Encerrar Ordem de Fabrico

1	2	3
4	5	6
7	8	9
	0	Apagar

Operador: Paulo Araújo
Turno: 1
Estado: Máquina em Produção.
Modo: Automático

Figura 4.32: Ordens de fabrico iniciadas

Adicionalmente, é chamada a função *update_database_eventos* para inserir na tabela "eventos" o subevento "Arranque Produção" do evento "Start Up e Set Down", indicando na base de dados o início das ordens de fabrico (figura 4.33). Como foi referido anteriormente, a secção 4.4.6 aprofunda a descrição desta função e outras semelhantes.

id	codigo_subevento	evento	subevento	of_esq	of_dir	modo	operador_numero	turno	data	hora
1	9.1	Start Up e Set Down	Arranque Produção	8	---	NULL	159	1	2016-06-14	15:40:36
2	9.1	Start Up e Set Down	Arranque Produção	---	9	NULL	159	1	2016-06-14	15:40:40

Figura 4.33: Entrada na tabela "eventos" dos dados de arranque da produção

Ainda dentro da função *actualiza_dados*, está integrada a deteção dos sinais de injeção no molde e rejeição das peças injetadas. Estas deteções são realizadas com o auxílio do método *add_event_detect* da biblioteca RPIO.GPIO [31] que permite associar uma função de *callback* à deteção de um *rising edge*, *falling edge*, ou ambos, do sinal. Esta função recebe como parâmetros o pino a detetar, o tipo de variação do sinal e que função deve ser iniciada aquando da deteção dessa variação. As funções de *callback* destes sinais são chamadas quando é detetado um *falling edge*, ou seja, assim que é concluído o tempo de ciclo. No caso do sinal de injeção no molde, é chamada a função *counter_pecas_boas* que recolhe o valor do campo "Aprovadas" das ordens de fabrico ativas e incrementa a esse valor o número de cavidades reais associado a cada ordem de fabrico, ou seja, assume inicialmente que todas as peças injetadas estão em boas condições. Como as peças só são consideradas defeituosas quando é emitido o sinal de rejeição, a função de *callback* deste sinal, denominada *counter_pecas_mas*, para além de somar o número de peças rejeitadas ao

campo "Rejeitadas" das ordens de fabrico ativas, também decrementa o mesmo valor ao campo "Aprovadas". Com esta ação elimina deste campo as peças anteriormente assumidas como não defeituosas.

Sempre que estas funções de *callback* processam os valores dos sinais detetados, é chamada a função *update_database_ocorrencias* para inserir uma entrada na tabela "ocorrencias" com os dados atualizados do número de peças aprovadas e rejeitadas. A figura 4.34 demonstra as ordens de fabrico iniciadas anteriormente com peças produzidas entretanto. A figura 4.35 mostra as últimas entradas na tabela "ocorrencias" das peças produzidas das ordens de fabrico 8 e 9. É observável que são produzidas cinco peças de cada vez porque este foi o número de cavidades reais indicado e, portanto, corresponde ao número de peças produzidas a cada injeção.

Equipamento: MO-INJ-01 Molde: ST0001 Up Time 00:03:25 Down Time: 00:14:45

Esquerda - Ordem de Fabrico nº 8:
Referência: 1ABM100100
Peças a produzir: 2000 Qtd por caixa: 50
Aprovadas 290 Rejeitadas 25

Direita - Ordem de Fabrico nº 9:
Referência: 1ABM100100
Peças a produzir: 1000 Qtd por caixa: 100
Aprovadas 290 Rejeitadas 35

Login | Eventos | **Produção** | Rejeições | Subeventos

Ordem de Fabrico Esquerda

Ordem de Fabrico nº:

Nº cavidades reais:

Ordem de Fabrico Direita

Ordem de Fabrico nº:

Nº cavidades reais:

1	2	3
4	5	6
7	8	9
<input type="button" value="OK"/>	0	<input type="button" value="Apagar"/>

Operador: Paulo Araújo
Turno: 1
Estado: Máquina em Produção.
Modo: Automático

Figura 4.34: Ordens de fabrico 8 e 9 com a produção em curso

id	id_ordem_fabrico	id_equipamento	id_referencia	id_molde	qtd_a_produzir	qtd_produzida_boas	qtd_produzida_mas	operador_numero	turno	data	hora
128	9	92	1	1	1000	275	35	159	1	2016-06-14	15:56:47
129	8	92	1	1	2000	275	25	159	1	2016-06-14	15:56:47
130	9	92	1	1	1000	280	35	159	1	2016-06-14	15:56:49
131	8	92	1	1	2000	280	25	159	1	2016-06-14	15:56:49
132	9	92	1	1	1000	285	35	159	1	2016-06-14	15:56:50
133	8	92	1	1	2000	285	25	159	1	2016-06-14	15:56:50
134	9	92	1	1	1000	290	35	159	1	2016-06-14	15:58:00
135	8	92	1	1	2000	290	25	159	1	2016-06-14	15:58:00

Figura 4.35: Entrada na tabela "ocorrencias" dos dados das peças produzidas durante as ordens de fabrico 8 e 9

A deteção de um destes sinais é um indício claro de que a máquina se encontra em produção,

portanto, sempre que ocorre uma injeção no molde, o campo "Estado:" é atualizado com a expressão "Máquina em Produção." A produção pode ser interrompida selecionando o botão "Parar Máquina" do separador "Eventos". Este botão ativa o sinal de saída que provoca a paragem da máquina e o "Estado:" é atualizado para "Máquina Parada."

Outro aspeto importante da monitorização da produção é o tratamento das peças defeituosas. Apesar da máquina devolver um sinal sempre que detetar um defeito nas injeções realizadas, este sinal nem sempre é emitido quando a injeção gera peças defeituosas. Devido a esta situação, o operador deve confirmar sempre o estado das peças produzidas e, ao longo da produção, catalogar a quantidade de defeitos encontrados de cada tipo. Esta ação deve ser realizada no separador "Rejeições" que está dividido em três colunas.

A primeira coluna lista todos os 47 defeitos que o processo de produção pode gerar. A lista completa destes defeitos é visível no anexo A.

Na segunda e terceira colunas do separador "Rejeições" são contabilizadas a quantidade de peças observadas com cada defeito. A segunda coluna está reservada para as peças da ordem de fabrico da esquerda e a terceira para as peças da ordem de fabrico da direita. Cada linha destas duas colunas é composta por uma célula que inclui um campo de entrada, um botão com o símbolo "+" e outro botão com o símbolo "-", com a função de, respetivamente, adicionar e subtrair ao valor do campo de entrada. Se o operador preferir, também pode utilizar o teclado numérico para esse efeito. O operador pode visualizar toda a lista de defeitos do separador deslizando a barra de deslocamento presente no separador. Para além disso, no topo de cada uma destas colunas encontra-se um indicador do número total dos defeitos inseridos.

No programa, assim que é iniciada uma ordem de fabrico esquerda ou direita, a função *atualiza_dados* chama a função *activa_rejeicoes_side* que recebe como parâmetro a orientação da ordem de fabrico. Se a orientação for "esquerda", ativa os *widgets* da coluna "Esquerda", se a orientação for "direita", ativa os *widgets* da coluna "Direita". A figura 4.36 demonstra um exemplo do primeiro caso.

Equipamento: MO-INJ-06	Molde: ST0014	Up Time 00:03:52	Down Time: 00:23:27
Esquerda - Ordem de Fabrico nº 6:		Direita - Ordem de Fabrico nº --	
Referência: AL00100100		Referência: ----	
Peças a produzir: 500 Qtd por caixa: ---		Peças a produzir: ---- Qtd por caixa: ----	
Aprovadas 0 Rejeitadas 0		Aprovadas ---- Rejeitadas ----	

Login	Eventos	Produção	Rejeições	Subeventos
-------	---------	----------	------------------	------------

Injeção						
A1 - Chupos	0	+	-	0	+	-
A2 - Rebarbas	0	+	-	0	+	-
A3 - Falta de matéria	0	+	-	0	+	-
Confirmar Rejeições						

1	2	3
4	5	6
7	8	9
	0	Apagar

Operador: **Paulo Araújo**
 Turno: **1**
 Estado: **Máquina em Produção.**
 Modo: **Automático**

Figura 4.36: Separador "Rejeições" com a coluna "Esquerda" ativa

Como os dados de produção são atualizados de hora a hora, é importante que o operador aceda ao separador "Rejeições" durante a atividade de produção, para contabilizar o número de peças defeituosas reais, visto que no momento de atualização do número de peças produzidas na tabela "ordensfabrico", o programa tem em conta os dados inseridos neste separador.

Alcançados os objetivos de produção de uma das ordens de fabrico ativas, chega o momento de a dar como encerrada. Para tal, o operador deve selecionar o botão "Encerrar Ordem de Fabrico" do separador "Produção". Assim que o faz, ativa a função *confirmar_rejeicoes* que começa por direcionar o programa para o separador "Rejeições" para que confirme o número de peças rejeitadas contabilizado, com o objetivo de garantir o registo dos dados corretos de produção. Nesta instância é ativado o botão "Confirmar" e apresentada a mensagem "Confirme o nº e tipo de peças rejeitadas" (figura 4.37) para salientar essa necessidade.

Equipamento: MO-INJ-06 Molde: ST0014 Up Time 00:07:07 Down Time: 00:25:43

Esquerda - Ordem de Fabrico nº 6:
Referência: AL00100100
Peças a produzir: 500 Qtd por caixa: ---
Aprovadas 498 Rejeitadas 6

Direita - Ordem de Fabrico nº --
Referência: ----
Peças a produzir: ---- Qtd por caixa: ----
Aprovadas ---- Rejeitadas ----

Login Eventos Produção Rejeições Subeventos

Lista de Defeitos	Esquerda	Direita
	Total: 14	Total: 0
Injeção		
A1 - Chupos	4 <input type="button" value="+"/> <input type="button" value="-"/>	0 <input type="button" value="+"/> <input type="button" value="-"/>
A2 - Rebarbas	3 <input type="button" value="+"/> <input type="button" value="-"/>	0 <input type="button" value="+"/> <input type="button" value="-"/>

Confirme o nº e o tipo de peças rejeitadas:

1	2	3
4	5	6
7	8	9
	0	Apagar

Operador: Paulo Araújo
Turno: 1
Estado: Máquina em Produção.
Modo: Manual

Figura 4.37: Estado de confirmação da quantidade de peças defeituosas detetadas

Assim que o operador procede à verificação desses dados, pressiona o botão "Confirmar" que está conectado à função *processa_rejeicoes*. Esta função contabiliza o valor inserido no campo de cada defeito e chama a função *update_database_rejeicoes* para inserir estes dados na tabela "rejeicoes". De seguida, é verificado se o número total de peças defeituosas indicadas é diferente do número de rejeições detetadas. Caso tal se verifique, são atualizados os dados dos campos "Rejeitadas" e "Aprovadas" da ordem de fabrico em processo de encerramento. Se o número de peças defeituosas indicadas pelo operador for superior ao número de peças defeituosas detetadas, retira a diferença ao número de peças aprovadas e vice-versa. Na figura 4.38 é visível a tabela "rejeições", correspondente às ordens de fabrico 8 e 9, preenchida com os tipos de defeitos encontrados e a respetiva quantidade.

id	id_ordem_fabrico	codigo_defeito	nr_defeitos	operador_numero	turno	data	hora
0		9 A1	3	159	1	2016-06-02	17:21:36
1		9 A10	3	159	1	2016-06-02	17:21:36
2		9 A11	2	159	1	2016-06-02	17:21:36
3		9 C2	7	159	1	2016-06-02	17:21:36
4		9 D1	5	159	1	2016-06-02	17:21:36
5		8 A1	14	159	1	2016-06-02	17:26:35
6		8 A2	6	159	1	2016-06-02	17:26:35
7		8 B1	4	159	1	2016-06-02	17:26:35
8		8 B2	4	159	1	2016-06-02	17:26:35
9		8 C3	3	159	1	2016-06-02	17:26:35
10		8 C7	5	159	1	2016-06-02	17:26:35
11		8 C8	4	159	1	2016-06-02	17:26:35
12		8 C9	4	159	1	2016-06-02	17:26:35
13		8 C10	5	159	1	2016-06-02	17:26:35
14		8 C11	3	159	1	2016-06-02	17:26:35

Figura 4.38: Tabela "rejeicoes" preenchida com as peças rejeitadas das ordens de fabrico 8 e 9

Por fim, o programa entra na função *finalizar_of* para retirar a informação da ordem de fabrico encerrada do campo "Dados de Produção". Adicionalmente, é chamada a função *update_database_ordensfabrico* para atualizar, na base de dados local, a tabela "ordensfabrico" com os dados de produção finais. A função *update_database_eventos* também é chamada para sinalizar na tabela "eventos" o encerramento da ordem de fabrico com o subevento "Final Produção".

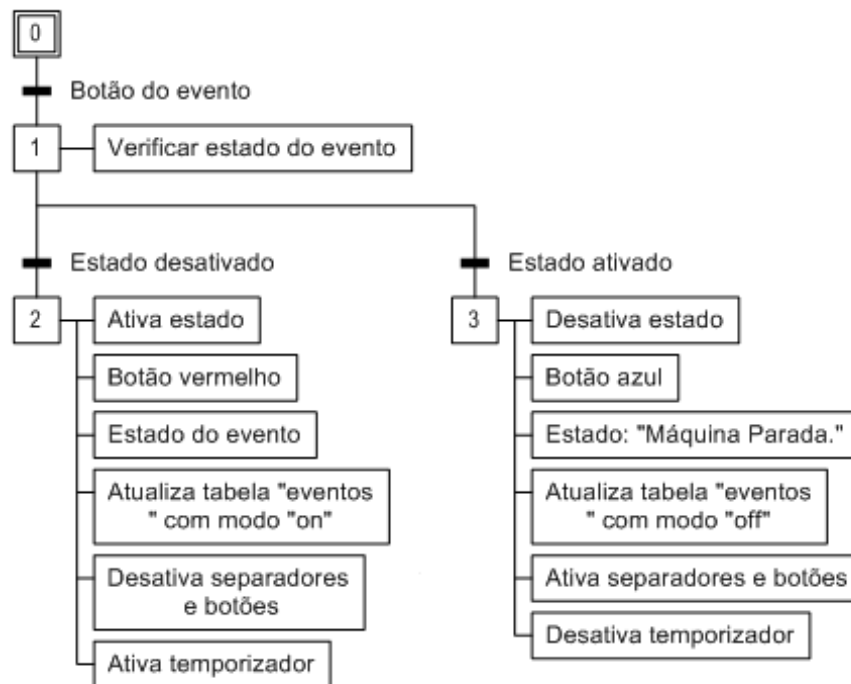
Ações		id	codigo_subevento	evento	subevento	of_esq	of_dir	modo	operador_numero	turno	data	hora
Editar	Deletar	3	9.3	Start Up e Set Down	Final Produção	8	---	NULL	159	1	2016-06-14	16:03:07
Editar	Deletar	4	9.3	Start Up e Set Down	Final Produção	---	9	NULL	159	1	2016-06-14	16:03:11

Figura 4.39: Entrada na tabela "eventos" da indicação de conclusão de ordens de fabrico

4.4.4 Eventos e Subeventos Temporizados

Eventos temporizados são ações ativadas durante um período que pretendem traduzir atividades impeditivas da produção e, portanto, também devem ser registadas.

O separador "Eventos" inclui três destes eventos traduzidos nos botões "Mudança de Molde", "Manutenção" e "Avaria". O processo gerado por cada um destes botões pode ser consultado no *grafcet* da figura 4.40 e o evento e subevento associado a cada um deles pode ser consultado na tabela 4.9.

Figura 4.40: *Grafcet* do estado "Manutenção" ativado

Botão	Evento	Subevento
Manutenção	Manutenção	Manutenção
Mudança de Molde	Mudança Molde	Mudança Molde
Avaria	Avaria	Observada

Tabela 4.9: Respetivo "evento" e "subevento" dos botões "Manutenção", "Mudança de Molde" e "Avaria"

Cada um destes botões tem dois modos, *on* e *off*, ou seja, ligado e desligado. Para além do mais, cada evento dos botões representa um estado da máquina com o mesmo nome, portanto, quando um botão é ligado, ativa o estado respetivo no canto inferior direito do ambiente gráfico.

Os estados que não estão ativos apresentam o botão respetivo a azul e quando um deles é selecionado pelo operador, a sua cor altera-se para vermelho sinalizando a ativação do estado. Quando um destes estados é ativado, inicia-se um temporizador que vai contabilizando a duração da atividade e na mesma instância é invocada a função *update_database_eventos* para inserir na tabela "eventos" uma nova entrada com os dados evento e subevento do respetivo botão. Adicionalmente, na mesma entrada, o campo "modo" é preenchido com a expressão *on*, indicando na base de dados o arranque deste evento. Enquanto um destes estados se encontrar ativo, os outros separadores encontram-se bloqueados para impedir que outro evento seja acionado ou que seja comandada qualquer ação de produção. Apenas os botões "Fim de Turno" e "Substituição" se

mantêm desbloqueados para permitir a troca de operador quando tal situação se aplicar. A figuras 4.41 e 4.42 demonstram este processo usando como exemplo o botão "Manutenção".

Equipamento: --- Molde: --- Up Time 00:00:00 Down Time: 00:04:25

Esquerda - Ordem de Fabrico nº --: Referência: --- Peças a produzir: ---- Qtd por caixa: ---- Aprovadas ---- Rejeitadas ----

Direita - Ordem de Fabrico nº --: Referência: --- Peças a produzir: ---- Qtd por caixa: -- Aprovadas ---- Rejeitadas ----

Logim Eventos Produção Rejeições Subeventos

Acções

Iniciar Produção Mudança de Molde

Substituição Manutenção

Fim de Turno Avaria

Parar Máquina SUBEVENTOS

Alarmes

Máquina Periféricos Molde Robot

1 2 3

4 5 6

7 8 9

0 Apagar

Operador: Paulo Araújo

Turno: 1

Estado: Manutenção 00:04:07

Modo: Automático

Figura 4.41: Estado "Manutenção" ativado

id	codigo_subevento	evento	subevento	of_esq	of_dir	modo	operador_numero	turno	data	hora
14	---	Manutenção	Manutenção	NULL	NULL	on	159	1	2016-06-14	16:26:30

Figura 4.42: Entrada na tabela "eventos" dos dados de início do evento "Manutenção"

Para indicar a conclusão do evento basta desligar o botão respetivo. Este retorna à cor azul, o campo "Estado" é retomado a "Máquina Parada". O temporizador é desligado e os separadores e botões bloqueados são desbloqueados. Para concluir o processo, a função *update_database_eventos* insere na tabela "eventos" uma entrada com os dados do evento desativado e o campo "modo" preenchido com a expressão *off* (figura 4.43).

id	codigo_subevento	evento	subevento	of_esq	of_dir	modo	operador_numero	turno	data	hora
14	---	Manutenção	Manutenção	NULL	NULL	on	159	1	2016-06-14	16:26:30
15	---	Manutenção	Manutenção	NULL	NULL	off	159	1	2016-06-14	16:32:48

Figura 4.43: Eventos e Subeventos - Entrada na tabela "eventos" dos dados de conclusão do evento "Manutenção"

Dentro do mesmo separador, a seleção do botão "SUBEVENTOS" direciona o utilizador para o separador com o mesmo nome. Neste separador, o utilizador tem acesso aos subeventos dos eventos "Ensaio e Testes Planeados", "Interrupção de Produção" e "Falta de Recursos". Como nesta instância apenas é importante o acesso aos subeventos dos eventos mencionados, os botões com os nomes desses eventos apenas têm a função de tornar os subeventos visíveis ou invisíveis.

Esta funcionalidade foi incluída para facilitar a visualização do conteúdo neste separador. A figura 4.44 mostra o estado do separador com todos os subeventos invisíveis e a figura 4.45 demonstra que basta selecionar o botão "Interrupção de Produção" para ter acesso aos subeventos respetivos.

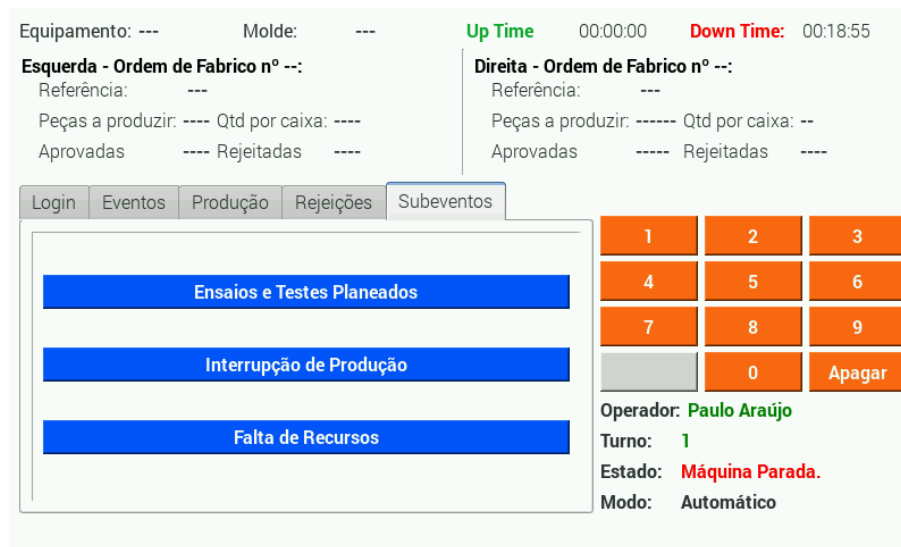


Figura 4.44: Subeventos invisíveis no separador "Subeventos"

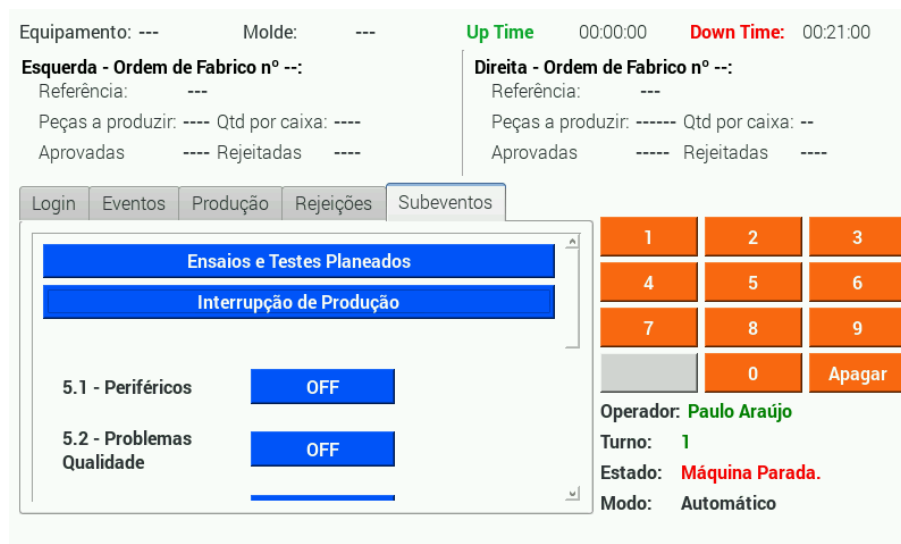


Figura 4.45: Subeventos visíveis do evento "Interrupção de Produção"

De forma semelhante ao eventos do separador "Eventos", cada um destes subeventos tem associado um botão que o permite ativar e desativar. Quando os subeventos se encontram inativos, o botão mantém a cor azul com o texto "OFF". Porém, quando o utilizador prime um destes botões, o subevento é ativado, o botão passa a apresentar uma cor vermelha com o texto "ON" e

um temporizador é iniciado na mesma linha. De formar semelhante ao processo descrito anteriormente, todos os outros separadores e restantes botões são bloqueados, impedindo as suas ações até o subevento ser dado como terminado. Adicionalmente, também é inserida uma entrada na tabela "eventos" com os dados deste subevento e o campo "modo" preenchido com a expressão *on*.

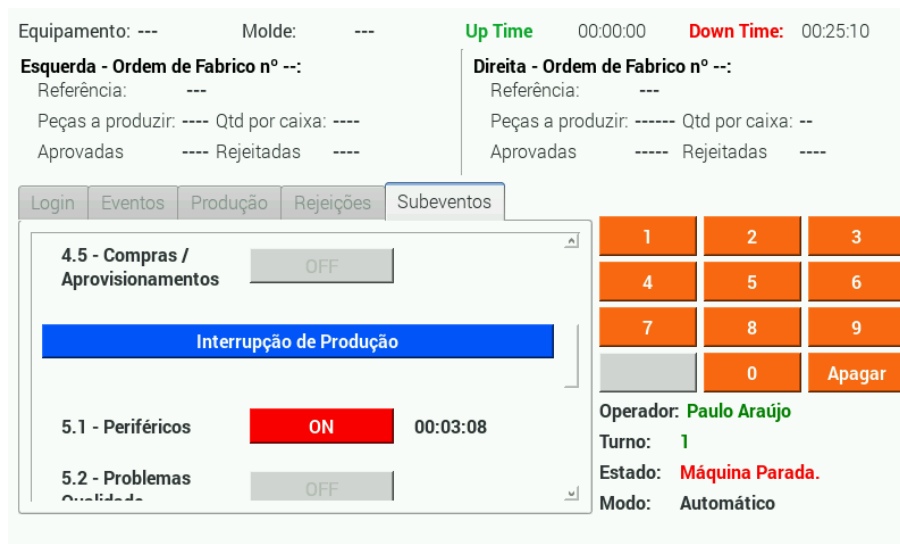


Figura 4.46: Subevento "Periféricos" ativo no separador "Subeventos"

id	codigo_subevento	evento	subevento	of_esq	of_dir	modo	operador_numero	turno	data	hora
16	5.1	Interrupção de Produção	Periféricos	NULL	NULL	on	159	1	2016-06-14	16:48:20

Figura 4.47: Entrada na tabela "eventos" dos dados de início do subevento "Periféricos"

Quando o subevento é desativado, o programa volta ao estado da figura 4.45 e a tabela "evento" é atualizada com uma nova indicando a conclusão deste.

id	codigo_subevento	evento	subevento	of_esq	of_dir	modo	operador_numero	turno	data	hora
16	5.1	Interrupção de Produção	Periféricos	NULL	NULL	on	159	1	2016-06-14	16:48:20
17	5.1	Interrupção de Produção	Periféricos	NULL	NULL	off	159	1	2016-06-14	16:53:47

Figura 4.48: Entrada na tabela "eventos" dos dados de conclusão do subevento "Periféricos"

4.4.5 Alarmes

Existem quatro sinais de alarme de avaria que a máquina consegue emitir. Estes sinalizam independentemente a avaria da máquina, do *robot*, do molde e de periféricos. De forma semelhante aos sinais de injeção e rejeição da injeção, os sinais de avaria foram detetados com o auxílio da função *add_event_detect*. Quando é detetado um *rising edge* de uma destas avarias, por exemplo, do molde, o indicador "Molde" no grupo "Alarmes" do separador "Eventos" é ativado, ou seja, altera a sua cor de fundo para vermelho, com o objetivo de chamar a atenção para esta situação. Para

além disso, o estado da máquina é alterado para "AVARIA" e na mesma linha é iniciado um temporizador. Para registar esta deteção na base de dados, é chamada a função *update_database_eventos* que insere na tabela "eventos" uma entrada com os campos "evento", "subevento" e "modo" preenchidos com "Avaria", "Molde" e "on", respetivamente. As figuras 4.49 e 4.50 ilustram o processo descrito.

Equipamento: MO-INJ-03 Molde: ST0001 Up Time 00:03:14 Down Time: 00:02:21

Esquerda - Ordem de Fabrico nº 6:
Referência: AL00100100
Peças a produzir: 500 Qtd por caixa: ---
Aprovadas 88 Rejeitadas 18

Direita - Ordem de Fabrico nº 8:
Referência: TABM100100
Peças a produzir: 2000 Qtd por caixa: 50
Aprovadas 220 Rejeitadas 50

Login | **Eventos** | Produção | Rejeições | Subeventos

Acções

Iniciar Produção | Mudança de Molde
Substituição | Manutenção
Fim de Turno | Avaria
Parar Máquina | SUBEVENTOS

Alarmes

Máquina | Periféricos | Molde | **Robot**

Operador: Paulo Araújo
Turno: 1
Estado: **AVARIA** 00:00:36
Modo: Automático

Figura 4.49: Estado de avaria do *robot*

Ações		id	codigo_subevento	evento	subevento	of_esq	of_dir	modo	operador_numero	turno	data	hora
Editar	Deletar	31	3.3	Avaria	Robot	6	8	on	159	1	2016-06-14	17:02:18

Figura 4.50: Entrada na tabela "eventos" dos dados de início da avaria do *robot*

No entanto, assim que é detetado o *falling edge* do mesmo sinal, significa que a avaria foi resolvida e portanto o indicador retoma ao aspeto original, o "Estado" é alterado para "Máquina Parada" e o temporizador é desligado. Na tabela "eventos", é inserida uma nova entrada com os mesmos dados, exceto o campo "modo", que é preenchido com a expressão *off*.

Ações		id	codigo_subevento	evento	subevento	of_esq	of_dir	modo	operador_numero	turno	data	hora
Editar	Deletar	31	3.3	Avaria	Robot	6	8	on	159	1	2016-06-14	17:02:18
Editar	Deletar	32	3.3	Avaria	Robot	6	8	off	159	1	2016-06-14	17:02:59

Figura 4.51: Entrada na tabela "eventos" dos dados de conclusão da avaria do *robot*

4.4.6 Função de Atualização das Bases de Dados

As funções de atualização das tabelas ativas da base de dados local são invocadas em várias instâncias descritas nas secções anteriores. Esta secção pretende esclarecer em maior pormenor o seu funcionamento.

Primeiro, a função de atualização da tabela "ordensfabrico", *update_database_ordensfabrico*, recebe como parâmetros a ordem de fabrico, a quantidade de peças aprovadas presente no campo "Aprovadas" e a quantidade de peças rejeitadas presente no campo "Rejeitadas". Esta função insere estes dois últimos dados na linha da tabela "ordensfabrico" cujo campo "id_ordens_fabrico" é igual ao parâmetro ordem de fabrico. O número de peças aprovadas e o número de peças rejeitadas são inseridos, respetivamente, nos campos *qtd_produzida_boas* e *qtd_produzida_mas*. Adicionalmente, também é atualizado o campo *data_atualizacao* com a data/hora do instante da atualização. Esta função é invocada de hora em hora sempre que está uma ordem de fabrico ativa e assim que uma destas ordens de fabrico é encerrada.

Já a atualização da tabela "eventos" está ao cargo da função *update_database_eventos*. Esta função recebe como parâmetros o evento, subevento, ordem de fabrico, orientação da peça da ordem de fabrico (esquerda ou direita), e o modo (*on* ou *off*). Quando esta função é chamada, os campos "of_esq" e "of_dir" são preenchidos com o número das respetivas ordens de fabrico ativas. Se alguma delas não estiver ativa, o respetivo campo fica preenchido com "NULL". No entanto, este processo de preenchimento dos campos "of_esq" e "of_dir", não se aplica a todos os casos. Os subeventos "Arranque Produção" ou "Final Produção" indicam, respetivamente, o início e o encerramento de uma ordem de fabrico. Assumindo o método descrito, se no programa já estiver ativa uma ordem de fabrico esquerda com o número 5, por exemplo, e o operador iniciar uma ordem de fabrico direita com o número 6, é dada a entrada da figura 4.52 na tabela. Esta entrada, só por si, não permite concluir qual das ordens de fabrico iniciou a produção.

id	codigo_subevento	evento	subevento	of_esq	of_dir	modo	operador_numero	turno	data	hora
13	9.1	Start Up e Set Down	Arranque Produção	5	6	NULL	159	1	2016-06-02	17:26:37

Figura 4.52: Entrada ambígua na tabela "eventos" de dados de arranque da produção

Portanto, para evitar esta ambiguidade, caso o parâmetro subevento recebido pela função *update_database_eventos* seja "Arranque Produção" ou "Final Produção", os campos "of_dir" e "of_esq" são preenchidos segundo um método diferente. No arranque de uma ordem de fabrico esquerda com o número 5, por exemplo, o campo "of_esq" é preenchido com esse número e o campo "of_dir" é preenchido com "--", mesmo que exista uma ordem de fabrico ativa com esta orientação. O mesmo acontece caso seja iniciada uma ordem de fabrico direita com o número 6.

id	codigo_subevento	evento	subevento	of_esq	of_dir	modo	operador_numero	turno	data	hora
1	9.1	Start Up e Set Down	Arranque Produção	5	--	NULL	159	1	2016-06-02	17:15:45
2	9.1	Start Up e Set Down	Arranque Produção	--	6	NULL	159	1	2016-06-02	17:15:48

Figura 4.53: Entrada real na tabela "eventos" de dados de arranque da produção

Olhando para figura 4.53 percebe-se que, com este método, a ambiguidade desaparece. A primeira entrada indica claramente que foi iniciada ordem de fabrico esquerda número 5 e a segunda entrada indica que foi iniciada uma ordem de fabrico direita com o número 6. Adicionalmente, o

preenchimento do campo "modo" desta tabela nem sempre é útil para todos os eventos. Quando o parâmetro recebido é "Start Up e Set Down", "Login" ou "Paragem" o campo "modo" é preenchido por "NULL". Isto acontece porque estes eventos não precisam de ser desativados depois de ativados. O "modo" dos restantes eventos é preenchido por *on* ou *off* consoante este tenha sido ativado ou desativado, respetivamente. A função *update_database_eventos* é chamada sempre que surge a necessidade de registar na base de dados um dos eventos da tabela 4.1.

A tabela "ocorrencias" está relacionada com a função *update_database_ocorrencias* e é chamada sempre que é detetado um sinal de injeção ou de rejeição da injeção. Os parâmetros que recebe incluem a ordem de fabrico, os números de identificação da máquina, da peça e do molde, o número de peças a produzir e os valores atuais de peças rejeitadas e peças aprovadas. Esta função foi construída com o objetivo de registar todas as injeções detetadas, em que instância ocorreram e que dados de produção devolveram.

Finalmente, a função *update_database_rejeicoes* é chamada unicamente aquando do encerramento de uma ordem de fabrico e insere na tabela "rejeicoes" os tipos de defeitos encontrados e quantas peças rejeitadas foram contabilizadas com esses mesmos defeitos. Por cada ordem de fabrico, o número de entradas na tabela é equivalente ao número de tipos de defeitos diferentes contabilizados pelo operador. Esta função, por cada linha que insere na tabela, recebe como parâmetros a ordem de fabrico, a orientação da mesma, o código do defeito e a quantidade de peças observadas do defeito respetivo.

4.4.7 Temporizadores

O programa de monitorização recorre a vários temporizadores que, a uma dada frequência específica de cada temporizador, verificam o estado de algumas variáveis globais. Estas variáveis acabam por se traduzir em ações ao longo do programa. Também são utilizados temporizadores para contabilizar o tempo decorrido desde um dado instante.

Um dos critérios deste sistema é o registo dos dados de produção de hora em hora. Para tal, é utilizado o *timer_updatedb*. Este temporizador, iniciado ao mesmo tempo que é iniciada uma ordem de fabrico, chama, de segundo a segundo, a função *timer_update_database_ordensfabrico* que começa por determinar, em cada segundo, a hora atual. Quando os minutos e os segundos são iguais a 0, ou seja, é hora certa, é chamada a função *update_database_ordensfabrico* para atualizar os dados da tabela "ordensfabrico". Nesta atualização também é contabilizada a quantidade de defeitos inseridos no separador "Rejeições", portanto, é importante que o operador registre as peças defeituosas que vai observando ao longo da atividade de produção.

A autenticação por cartão RFID também se baseia num temporizador, tal como é mencionado na secção 4.4.2. O *timer_rfid* é iniciado sempre que o programa é direcionado para o separador "Login". Este método de autenticação utiliza o temporizador para detetar cartões RFID a cada segundo. Assim que um cartão é detetado, o temporizador é encerrado.

Já o *timer_alarmes* foi criado para contornar um problema gerado pela natureza da função *add_event_detect* da biblioteca RPI.GPIO. Todas as funções criadas no programa funcionam numa *thread* que é composta por uma classe onde são iniciados os objetos de todo o ambiente gráfico.

As funções de *callback* de *add_event_detect* funcionam por natureza numa *thread* secundária. Esta *thread* não inclui os objetos do ambiente gráfico e, portanto, nas funções de *callback* não são permitidas linhas de código que alterem o ambiente gráfico. Desta forma, este temporizador, criado na *thread* principal, é iniciado quando o programa arranca e, de segundo a segundo, verifica se foi detetada alguma variação nos sinais dos alarmes. Quando tal acontece, procede às devidas alterações gráficas descritas na secção 4.4.5.

Noutra instância, os temporizadores responsáveis por contabilizar o tempo decorrido depois da ativação de um evento ou subevento, são respetivamente o *timer_eventos* e o *timer_subeventos*. O resultado da ação destes temporizadores é apresentado na secção 4.4.4. A função associada a cada um deles vai contabilizando os segundos decorridos desde a sua ativação e converte-os para o formato HH:MM:SS.

Os tempos de atividade e inatividade contabilizados no topo do programa são determinados segundo o *timer_uptime_downtime*. Na máquina, o tempo de ciclo de injeção no molde nunca é superior a 60 segundos. Caso o número de segundos desde a última injeção detetada ultrapasse este valor, significa que a produção foi interrompida, portanto, a função conectada ao temporizador começa a incrementar ao tempo de inatividade. Por outro lado, assim que uma injeção é detetada, a função interrompe a contagem do tempo de inatividade e retoma a contagem do tempo de atividade.

Por último, o *timer_login* é utilizado para contabilizar o número de segundos desde que foi realizada a última autenticação. Assim que o valor ultrapassar os 180 segundos, é iniciado um processo de autenticação de confirmação do operador. Esta funcionalidade é detalhada na secção 4.4.2.

Capítulo 5

Plataforma de Demonstração e Análise de Dados

5.1 Introdução

Apenas com o sistema de monitorização implementado, apresentado no Capítulo 4, não é possível observar os dados de produção das ordens de fabrico ativas sem ser necessária uma deslocação à máquina onde este está instalado. Também se torna pouco prático analisar e avaliar a produção das ordens de fabrico concluídas acedendo diretamente à base de dados local do módulo. Portanto, para solucionar esses problemas, foi criada uma plataforma em PHP (*back-end*), HTML e CSS (*front-end*), alojada no *Raspberry Pi*, que acede à base de dados local e processa os dados lá presentes. Uma das páginas que constitui a plataforma permite visualizar os dados de produção mais recentes e os eventos ocorridos durante as ordens de fabrico em curso. A outra página faz um levantamento de toda a informação das ordens de fabrico concluídas, incluindo uma avaliação da performance da atividade de produção da mesma. Este capítulo descreve os métodos de processamentos destes dados e demonstra os resultados respetivos.

5.2 Demonstração de Dados das Ordens de Fabrico em Curso

5.2.1 Dados Atuais das Ordens de Fabrico

A página "Ordens de Fabrico em Curso" permite observar os dados de produção das ordens de fabrico correntes, os eventos que se encontram ativos e os que, entretanto, foram concluídos.

Numa primeira instância, a unidade de processamento de dados da página começa por averiguar quais as ordens de fabrico ativas no momento, ou seja, aquelas que ainda não foram dadas como concluídas. Portanto, para verificar quais as ordens de fabrico iniciadas, é realizada uma pesquisa na tabela "eventos" por todas as entradas com o subevento "Arranque Produção". O resultado desta ação, usando como exemplo uma base de dados em processo de construção, pode ser observado na figura 5.1. Portanto, neste caso, é possível perceber que, ao longo da produção,

já foram iniciadas as ordens de fabrico 8, 9, 5 e 6. Os números destas ordens de fabrico ficam registados.

id	of_esq	of_dir	subevento	hora
1	8	---	Arranque Produção	17:15:45
2	---	9	Arranque Produção	17:15:48
14	5	---	Arranque Produção	22:16:15
15	---	6	Arranque Produção	22:16:22

Figura 5.1: Entradas da tabela "eventos" dos dados de "Arranque Produção" das ordens de fabrico 8, 9, 5 e 6

O passo seguinte passa por perceber quais destas ordens de fabrico foram entretanto encerradas. Para tal, o programa pesquisa na tabela por entradas com o subevento "Final Produção". Na figura 5.2 é apresentado o resultado desta pesquisa e é possível concluir que as ordens de fabrico 8 e 9 foram entretanto encerradas e, portanto, são irrelevantes para esta página. Assim, o programa elimina estas ordens de fabrico do registo anterior, resultando numa lista apenas com as ordens de fabrico 5 e 6.

id	of_esq	of_dir	subevento	hora
5	---	9	Final Produção	17:21:38
12	8	---	Final Produção	17:26:37

Figura 5.2: Entradas da tabela "eventos" dos dados de "Final Produção" das ordens de fabrico 8 e 9

De seguida, o programa acede à mesma tabela para recolher a hora de início destas ordens de fabrico e à tabela "ocorrências" para coletar a informação, registada até ao momento, do número de peças produzidas aprovadas e do número de peças produzidas rejeitadas. Estes dados são carregados para o campo "Ordens de Fabrico". Na figura 5.3 estão presentes os dados das ordens de fabrico 5 e 6. Como um dos objetivos é visualizar o resultado atual de produção destas ordens de fabrico, o programa acede sempre à última entrada da tabela "ocorrências". Portanto, sempre que a página é atualizada, são devolvidos os dados de produção mais recentes. A página está programada para atualizar de 3 em 3 minutos. O mesmo efeito pode ser conseguido através de um clique no botão "Atualizar Dados".

Ordens de Fabrico em Curso					
Ordens de Fabrico					
Número	Produzir	Aprovadas	Rejeitadas	Operador	Hora de Início
5	4500	9	0	159	22:16:15
6	500	6	0	159	22:16:22
Actualizar dados					
Eventos em Curso					
(Ausência de Eventos)					
Eventos Terminados					
(Ausência de Eventos)					

Figura 5.3: Campo "Ordens de Fabrico" com os dados mais recentes das ordens de fabrico 5 e 6

5.2.2 Eventos em Curso e Eventos Terminados

Outro objetivo desta página passa por indicar quais os eventos a decorrer durante a atividade de produção e quais os eventos entretanto encerrados. Para fornecer a informação dos eventos ativos, na unidade de processamento da página verifica-se, na tabela "eventos", quais as entradas cujo campo "modo" se encontra preenchido por *on* ou *off*. Desta ação são excluídas todas as entradas que antecedem ao arranque das ordens de fabrico ativas. Continuando com o mesmo exemplo, o resultado deste processo, visível na figura 5.4, devolve dois eventos "Avaria" e "Falta de Recursos". É também possível concluir que, destes dois, apenas o evento "Falta de Recursos" não foi desativado, ou seja, ainda está a decorrer. Portanto, apenas este evento fica registado.

id	codigo_subevento	evento	subevento	modo	hora
19	3.3	Avaria	Robot	on	01:12:23
20	3.3	Avaria	Robot	off	01:43:32
21	6.1	Falta de Recursos	Matérias-Primas	on	02:01:38

Figura 5.4: Entradas da tabela "eventos" com o evento "Avaria" concluído e evento "Falta de Recursos" por concluir

Desta forma, os dados do evento "Falta de Recursos" são carregados no campo "Eventos em Curso". Na mesma linha, também é carregada a informação sobre a hora de início do mesmo (figura 5.5). Como o evento "Avaria" já tinha sido concluído, aparece presente em "Eventos Terminados". Nesta instância, o programa apenas inclui eventos que foram também encerrados, ou seja, que têm uma entrada adicional com o campo "modo" preenchido por *off*.

Ordens de Fabrico em Curso					
Ordens de Fabrico					
Número	Produzir	Aprovadas	Rejeitadas	Operador	Hora de Início
5	4500	582	9	159	22:16:15
6	500	328	16	159	22:16:22
Atualizar dados					
Eventos em Curso					
Código	Evento	Subevento	Hora de Início		
6.1	Falta de Recursos	Matérias-Primas	02:01:38		
Eventos Terminados					
Código	Evento	Subevento	Hora de Início	Duração	
3.3	Avaria	Robot	01:12:23	31 minutos e 9 segundos	

Figura 5.5: Exemplo da página "Ordens Fabrico em Curso" com o evento "Falta de Recursos" em curso e o evento "Avaria" terminado

Entretanto, quando o evento "Falta de Recursos" for desativado, o campo "Eventos em Curso" fica vazio e este faz a transição para o campo "Eventos Terminados" (figura 5.6). Para além da hora de início, este campo também exibe a duração de cada evento. O tempo de duração é obtido calculando a diferença de tempo entre a hora da entrada na tabela "eventos" com o modo *off* e hora da entrada na tabela com o modo *on*.

Ordens de Fabrico em Curso					
Ordens de Fabrico					
Número	Produzir	Aprovadas	Rejeitadas	Operador	Hora de Início
5	4500	621	33	159	22:16:15
6	500	354	32	159	22:16:22
Atualizar dados					
Eventos em Curso					
Código	Evento	Subevento	Hora de Início		
Eventos Terminados					
Código	Evento	Subevento	Hora de Início	Duração	
3.3	Avaria	Robot	01:12:23	31 minutos e 9 segundos	
6.1	Falta de Recursos	Matérias-Primas	02:01:38	15 minutos e 10 segundos	

Figura 5.6: Página "Ordens de Fabrico em Curso" com os eventos "Falta de Recursos" e "Avaria" terminados

Se entretanto, uma das ordens de fabrico for dada como encerrada, desaparece do campo "Ordens de Fabrico". Porém, as informações dos eventos mantêm-se até que a ordem de fabrico restante seja concluída (figura 5.7). Quando não existe qualquer ordem de fabrico em curso, a página não disponibiliza qualquer tipo de informação, ficando a aguardar até que uma nova ordem de fabrico seja iniciada (figura 5.8).

Ordens de Fabrico em Curso					
Ordens de Fabrico					
Número	Produzir	Aprovadas	Rejeitadas	Operador	Hora de Início
5	4500	630	60	159	22:16:15
Atualizar dados					
Eventos em Curso					
Código	Evento	Subevento	Hora de Início		
Eventos Terminados					
Código	Evento	Subevento	Hora de Início	Duração	
3.3	Avaria	Robot	01:12:23	31 minutos e 9 segundos	
6.1	Falta de Recursos	Matérias-Primas	02:01:38	15 minutos e 10 segundos	

Figura 5.7: Página "Ordens de Fabrico em Curso" apenas com a ordem de fabrico 5 ativa



Figura 5.8: Estado da página "Ordens de Fabrico em Curso" sem ordens de fabrico ativas

5.3 Processamento das Ordens de Fabrico Concluídas

A página "Ordens de Fabrico Concluídas" foi criada para permitir a consulta de todos os dados registados durante ordens de fabrico entretanto encerradas. Esta secção explora com profundidade todo o processamento por detrás da informação apresentada.

5.3.1 Dados de Identificação

Primeiro, o código de processamento averigua quais as ordens de fabrico encerradas. Para isso efetua uma pesquisa na tabela "eventos" pelas ordens de fabrico iniciadas e regista os números das ordens de fabrico resultantes. De seguida, pesquisa na tabela por aquelas que já foram concluídas e elimina as restantes da lista. A lista final é carregada no formulário da figura 5.9. Desta forma, o utilizador pode seleccionar a ordem de fabrico que quer consultar e, ao premir o botão "Submeter", o formulário regista e submete o número da ordem de fabrico a consultar. Este dado é submetido para a mesma página e recebido por uma unidade de processamento diferente. Esta unidade processa toda a informação referente à ordem de fabrico submetida e posteriormente carrega todos esses dados nos campos respetivos da página. Na figura 5.10 é possível observar toda a informação que a página processa, usando como exemplo a atividade de produção da ordem de fabrico 6.

Seleccione a Ordem de Fabrico:		<input type="text" value="-----"/>	<input type="button" value="Submeter"/>
		8	
		9	
		6	
Ordem de Fabrico nº ---	Orientação: ---		
Data: ---	Iniciada às ---		Encerrada às ---
ID Peça: ---	ID Máquina: ---		ID Molde: ---
Operadores Responsáveis			
Número do Operador	Turno		Hora de Entrada
-	-		-

Figura 5.9: Formulário com a lista das ordens de fabrico concluídas

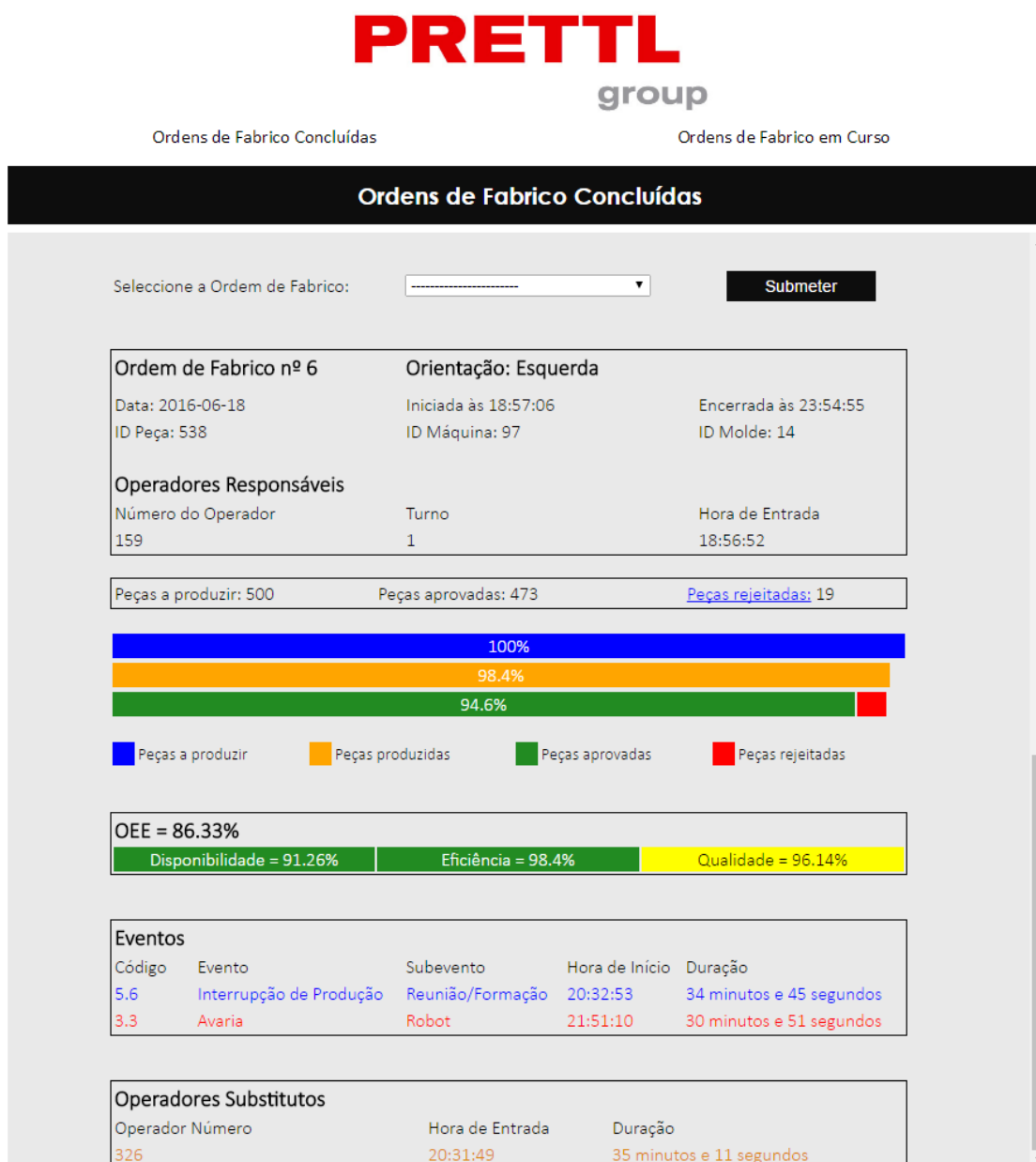


Figura 5.10: Página "Ordens de Fabrico Concluídas" da ordem de fabrico número 6

Os primeiros dados a serem processados são os da figura 5.11. As horas de início e conclusão da ordem de fabrico são retiradas das entradas "Arranque Produção" e "Final Produção" da tabela "eventos". Os números de identificação da peça, da máquina e do molde são recolhidos da entrada na tabela "ordensfabrico", correspondente à ordem de fabrico selecionada. Os números de identificação e turno de trabalho dos operadores responsáveis, durante a atividade da ordem fabrico, são recolhidos da tabela "ocorrências", através do acesso aos campos "operador_numero" e "turno". Adicionalmente, também é acedido à tabela "eventos" para recolher a hora a que é detetada a

primeira autenticação desses operadores durante o turno de trabalho

Ordem de Fabrico nº 6		Orientação: Esquerda	
Data: 2016-06-18	Iniciada às 18:57:06	Encerrada às 23:54:55	
ID Peça: 538	ID Máquina: 97	ID Molde: 14	
Operadores Responsáveis			
Número do Operador	Turno	Hora de Entrada	
159	1	18:56:52	

Figura 5.11: Dados da ordem de fabrico número 6

5.3.2 Dados da Produção

Os dados efetivos da produção são carregados nos campos da figura 5.12, em formato numérico e em formato percentual. Os dados do número de peças a produzir, número de peças produzidas aprovadas e número de peças produzidas rejeitadas são retirados da tabela atualizada "ordensfabrico". Para que estes dados sejam interpretados mais facilmente, foram traduzidos em barras percentuais codificadas por cores. Esta componente visual foi incorporada para permitir, também, uma leitura e análise mais imediata dos níveis de produção.

A barra azul, apresentada sempre a 100%, refere-se à quantidade esperada de peças produzidas. Esta barra foi incluída para ser usada como termo de comparação visual com as restantes. A percentagem de peças produzidas, em relação ao valor de peças a produzir, é indicada pela barra laranja. A quantidade de peças produzidas divide-se entre peças aprovadas e peças rejeitadas. O índice percentual de cada um destes valores é apresentado nas barras verde e vermelha, respetivamente.

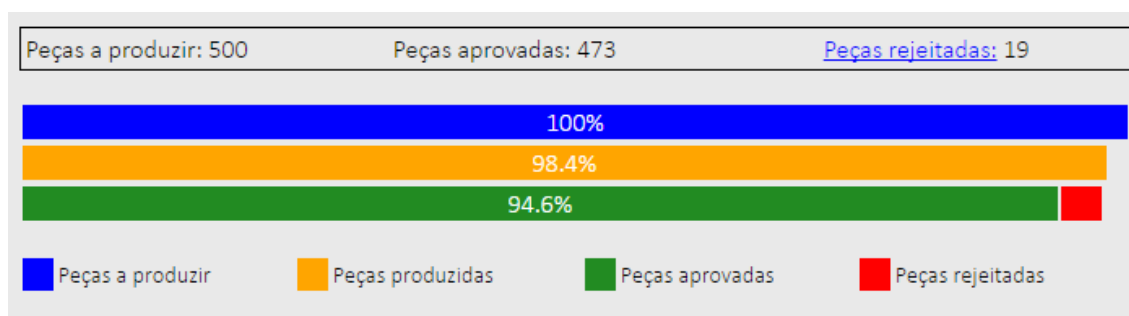


Figura 5.12: Dados de produção da ordem de fabrico 6

Ainda dentro deste grupo, o campo "Peças rejeitadas" está conectado a uma ligação que direciona o utilizador para uma página adicional, enviando como parâmetro a ordem de fabrico selecionada. Nesta página, apresentada na figura 5.13 e de título "Rejeições", são exibidos todos os tipos

de defeitos encontrados no conjunto de peças rejeitadas, bem como as respectivas quantidades. Os dados são retirados da tabela "rejeicoes", mais propriamente dos campos "codigo_defeito" e "nr_defeitos", cujo campo "id_ordem_fabrico" é igual à ordem de fabrico recebida como parâmetro. Como esta tabela não apresenta a descrição associada aos códigos dos defeitos, esta informação é retirada da tabela "setup_rejeicoes"

Rejeições		
Ordem de Fabrico nº 6		Número total de rejeições: 19
<< Voltar		
Código	Descrição	Quantidade
A1	Chupos	7
A2	Rebarbas	4
A3	Falta de Matéria	3
A13	Deformação	1
D2	Peça partida/danificada	4

Figura 5.13: Página "Rejeições" com a lista de defeitos encontrados na ordem de fabrico número 6

5.3.3 Dados dos Eventos Acionados e Detetados

Os dados dos eventos que ocorreram durante a ordem de fabrico são carregados no campo "Eventos", visível na figura 5.14. O método de recolha destes dados é igual ao utilizado no campo "Eventos Terminados" da página "Ordens de Fabrico em Curso", descrito na secção 5.2. As avarias são apresentadas a vermelho porque simbolizam eventos indesejados e inesperados. Os restantes eventos, como representam ações indicadas voluntariamente, são representados pela cor azul. Caso não tenham ocorrido quaisquer eventos durante a ordem de fabrico, é apresentada a mensagem "Ausência de Eventos" (figura 5.15).

Eventos				
Código	Evento	Subevento	Hora de Início	Duração
5.6	Interrupção de Produção	Reunião/Formação	20:32:53	34 minutos e 45 segundos
3.3	Avaria	Robot	21:51:10	30 minutos e 51 segundos

Figura 5.14: Informação relativa aos eventos ocorridos durante a ordem de fabrico 6

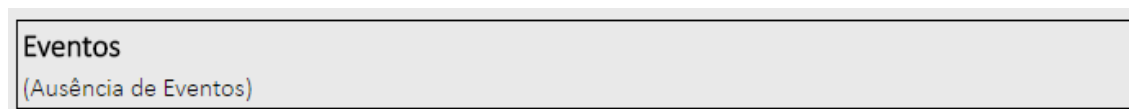


Figura 5.15: Ausência de eventos durante a ordem de fabrico

5.3.4 Dados dos Operadores de Substituição

Outra informação adicional de monitorização da ordem de fabrico indicada pela página são os dados relativos aos operadores de substituição. Para exibir os número de identificação dos operadores e hora de autenticação no sistema, são consultados os campos "operador_numero" e "hora" de todas de todas as entradas da tabela "eventos" cujos campos "subevento" e "id_ordem_fabrico" são iguais a "Operador Substituto" e à ordem de fabrico em causa, respetivamente. Para calcular a duração de cada substituição, é acedido à entrada seguinte com o subevento "Operador Planeado", ou seja, a entrada que simboliza a autenticação do operador original após o retorno deste à responsabilidade da ordem de fabrico. Desta entrada é retirada a hora e calculada a diferença entre as horas das duas entradas. A figura 5.16 apresenta um campo preenchido com estes dados. No caso da ausência de substituições, a página apresenta uma mensagem a indicar isso mesmo (figura 5.17).

Operadores Substitutos		
Operador Número	Hora de Entrada	Duração
326	20:31:49	35 minutos e 11 segundos

Figura 5.16: Informação relativa aos operadores substitutos autenticados durante a ordem de fabrico 6



Figura 5.17: Ausência de operadores substitutos durante a ordem de fabrico 9

5.3.5 Dados de Performance - OEE

A unidade de processamento da página também processa os dados recolhidos com o objetivo de avaliar a performance da produção da ordem de fabrico. A performance é traduzida no valor percentual do OEE resultante da relação entre três indicadores: disponibilidade, eficiência e qualidade.

O indicador de disponibilidade é calculado segundo o quociente entre o tempo de trabalho efetivo e o tempo de trabalho planeado. O tempo de trabalho efetivo refere-se ao tempo total

de atividade de produção com exclusão dos períodos de não produção. Portanto, para obter o tempo total de atividade de produção, é consultada a tabela "ocorrencias", onde são recolhidas as horas da primeira e última entrada referente à ordem de fabrico em causa, ou seja, as horas de primeira e última injeção no molde. O período de não produção resulta da soma da duração total de todos os eventos impeditivos da produção que ocorreram durante a ordem de fabrico. O tempo de trabalho planeado é retirado do campo "tempo_previsto" da entrada referente à ordem de fabrico em questão, na tabela "ordensfabrico".

$$\text{Disponibilidade} = \frac{\text{tempo de trabalho efetivo}}{\text{tempo de trabalho planeado}} = \frac{\text{duração da produção} - \text{duração dos eventos}}{\text{tempo de produção previsto}} \quad (5.1)$$

A métrica da eficiência é calculada através do quociente entre a produção efetiva e a produção esperada no período de trabalho efetivo. A produção efetiva refere-se ao número de peças produzidas, ou seja, ao total de peças aprovadas e rejeitadas. A produção esperada no período de trabalho efetivo refere-se ao número de peças a produzir. Estes três dados são retirados dos campos "qtd_produzida_boas", "qtd_produzida_mas" e "qtd_a_produzir" da tabela "ordensfabrico" atualizada.

$$\text{Eficiência} = \frac{\text{produção efetiva}}{\text{produção esperada no período de trabalho efetivo}} = \frac{\text{peças aprovadas} + \text{peças rejeitadas}}{\text{quantidade de peças a produzir}} \quad (5.2)$$

Já o indicador de qualidade resulta simplesmente do número de peças aprovadas sobre o número total de peças produzidas.

$$\text{Qualidade} = \frac{\text{peças aprovadas}}{\text{peças aprovadas} + \text{peças rejeitadas}} \quad (5.3)$$

Finalmente, o valor do OEE é o resultado do produto entre os três indicadores.

$$\text{OEE} = \text{Disponibilidade} \times \text{Eficiência} \times \text{Qualidade} \quad (5.4)$$

Na figura 5.18 é observável que, para além do valor do OEE, também são apresentados os resultados dos três indicadores. Na ausência de algum dado que impeça o cálculo de um dos indicadores, é exibida uma mensagem que indica isso mesmo. A figura 5.19 mostra o exemplo da ordem de fabrico 8 cujo campo "tempo_previsto" da tabela "ordensfabrico" não se encontra preenchido e, portanto, impossibilita o cálculo da disponibilidade e, conseqüentemente, do OEE. No entanto, isto não impede que sejam calculados e exibidos os resultados dos restantes indicadores.

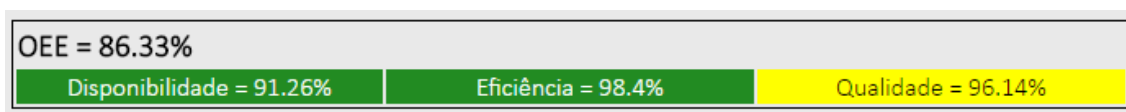


Figura 5.18: Valor do OEE e indicadores da ordem de fabrico 6

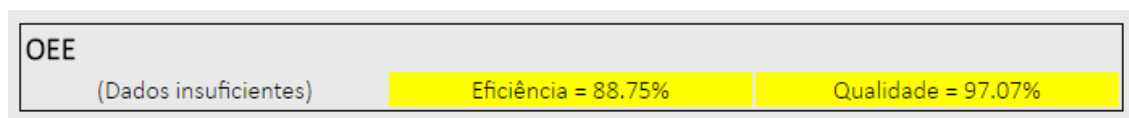


Figura 5.19: Ausência do resultado do indicador disponibilidade da ordem de fabrico 8

Contudo, para facilitar a interpretação dos resultados dos indicadores, foi implementado um sistema de codificação colorido. Quanto mais próximas estiverem as percentagens do valor percentual máximo, melhor foi a performance da atividade de produção. A codificação utilizada pretende refletir isso mesmo, através das cores vermelho, amarelo e verde. As tabelas 5.1, 5.2 e 5.3 apresentam os intervalos percentuais associados a cada cor, dentro de cada indicador.

Disponibilidade	
Índice Percentual	Código de cor
Menor do que 70%	Vermelho
Entre 71% e 85%	Amarelo
Maior do que 86%	Verde

Tabela 5.1: Codificação colorida do indicador "Disponibilidade"

Eficiência	
Índice Percentual	Código de cor
Menor do que 80%	Vermelho
Entre 81% e 90%	Amarelo
Maior do que 91%	Verde

Tabela 5.2: Codificação colorida do indicador "Eficiência"

Qualidade	
Índice Percentual	Código de cor
Menor do que 90%	Vermelho
Entre 91% e 97%	Amarelo
Maior do que 98%	Verde

Tabela 5.3: Codificação colorida do indicador "Qualidade"

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Conclusões

Foi desenvolvido com sucesso um sistema automático de monitorização da atividade de produção, adaptado às necessidades das máquinas de injeção presentes na fábrica. Este sistema assenta numa interface gráfica com o *font-end* elaborado no *Qt-Designer* e o *back-end* codificado em cerca de 1750 linhas de *Python*. O *back-end* da interface traduz os sinais de injeção no molde em quantidade de peças produzidas, valor que é também indicado na componente gráfica do programa. O sinal de rejeição da injeção também é detetado e traduzido em peças defeituosas que devem ser rejeitadas. No entanto, como este sinal nem sempre é emitido pela máquina quando ocorre algum defeito na injeção, o programa está preparado para que o operador indique os defeitos que encontra nas peças, distinguindo as peças produzidas em peças aprovadas e rejeitadas. Os sinais de avaria emitidos pela máquina são traduzidos em indicadores visuais de alarme para serem rapidamente identificados. Estes sinais são emitidos pela máquina a um valor de tensão superior ao valor máximo de tensão dos pinos do *Raspberry Pi* e, por esse motivo, o sistema também inclui uma placa impressa com circuitos capazes de adaptar os níveis de tensão da máquina. Todos estes processos cumprem os objetivos de recolha e processamento dos sinais emitidos pela máquina.

Os eventos que não são indicados pela máquina através da emissão de sinais, como mudança de molde, manutenção e falta de recursos, podem ser ativados na interface gráfica e desativados assim que concluídos. Estes eventos são ligados e desligados premindo os botões respetivos.

O objetivo de integrar o tempo de atividade, o tempo de inatividade e o estado da máquina também é cumprido. Toda esta informação está integrada na interface gráfica e é controlada pelos sinais e ações descritas anteriormente.

Para cumprir a necessidade de identificar o operador responsável pela produção da máquina, foram implementados dois métodos de autenticação: um permite a autenticação do operador através da introdução do número de identificação e o outro, preferível e mais seguro, incorpora a tecnologia RFID com um leitor e cartões de identificação associados a cada operador. O processo de autenticação também gere trocas de turno e substituições de operadores.

Todos os dados mencionados até agora, quer tenham sido recolhidos da máquina, quer tenham sido acionados pelo operador, são registados numa base de dados alojada localmente no *Raspberry Pi*. Esta base de dados está estruturada tendo em conta as necessidades da máquina e do programa desenvolvido. Para além disso, o *Raspberry Pi* foi configurado para aceder ao servidor da base de dados central da fábrica, com o objetivo de aceder a informação relativa às ordens de fabrico e aos operadores da fabrica, dados essenciais para o funcionamento da interface. O programa integra as duas bases de dados sendo a base de dados local acessível à base de dados central para que esta atualize os dados de produção das ordens de fabrico.

A este sistema foi adicionada uma plataforma de demonstração e análise de resultados que, apesar de não fazer parte dos objetivos iniciais, foi desenvolvida por ter sido considerada útil e complementar ao sistema de monitorização da produção. Esta plataforma consiste num *site* com o *back-end* codificado em PHP e o *front-end* codificado em HTML e CSS, num total de 1470 linhas de código, e permite obter informações acerca das ordens de fabrico em curso e das ordens de fabrico concluídas. Esta plataforma está alojada no servidor do *Raspberry Pi* e pode ser acedida por qualquer computador ligado à rede da fábrica. Desta forma é possível consultar informação relativa ao volume de produção e aos eventos ocorridos nas ordens de fabrico ativas, sem ser necessária a deslocação à máquina onde está instalado o módulo do sistema de monitorização. Adicionalmente, o *site* inclui uma página que lista as ordens de fabrico concluídas e permite consultar quais os operadores que trabalharam durante a mesma, que eventos foram acionados e a sua duração, a que horas foi iniciada e encerrada, quais os dados estatísticos do volume de produção e, finalmente, qual o nível de performance da ordem de fabrico (indicado através do OEE).

Por último, a solução apresentada foi testada e validada na máquina de injeção para a qual foi projetada. Este documento não apresenta um capítulo dedicado exclusivamente aos resultados obtidos porque estes são expostos ao longo dos capítulos 4 e 5 como exemplos ilustrativos dos processos que representam. No anexo D é feito um levantamento dos custos de material da solução proposta, com vista a demonstrar a natureza de baixo custo do sistema.

6.2 Trabalho Futuro

Este projeto pode ser expandido de várias formas. Neste trabalho apenas foi testado e implementado um módulo, no entanto, seria interessante instalar mais módulos em várias máquinas de injeção e adaptar o sistema a outro tipo de máquinas com requisitos diferentes. A plataforma de análise de resultados também pode sofrer um incremento de funcionalidades, principalmente no tratamento estatístico de dados. Para além de ser feita uma avaliação da performance de produção da máquina, pode-se avaliar a performance de produção semanal, diária e ,até, fazer uma comparação da performance de produção entre turnos.

Anexo A

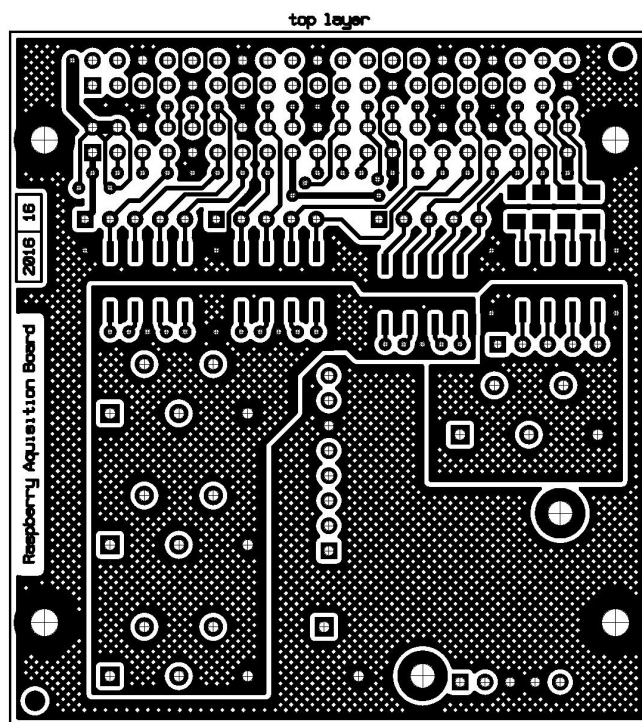
Tipos de Defeitos das Peças Rejeitadas

Código	Descrição
A1	Chupos
A2	Rebarbas
A3	Falta de Matéria
A4	Queimados
A5	Contaminação/Pintas pretas
A6	Bolhas de ar
A7	Peça danificada
A8	Dimensões incorretas
A9	Excesso de matéria
A10	Manchas/Sombras/Raiados
A11	Marcados na peça NOK
A12	Diferença de cor
A13	Deformação
B1	Falta de tinta
B2	Borbotos
B3	Picadas
B4	Casca de laranja
B5	Riscos/Brilhos
B6	Pintura/metalização descascada
B7	Excesso de tinta/metalização
B8	Rebarbas de tinta/metalização
B9	Cor/Brilho NOK
C1	Falta de componentes
C2	Componentes danificados
C3	Troca de componentes / misturados
C4	Ruído
C5	Componentes mal montado
C6	Peça danificada
C7	Teclas presas/Mau clickpoint/Bascular
C8	Grease em falta/excesso/Mal colado
C9	Iluminação NOK
C10	Consumo elétrico NOK
C11	Estanquidade NOK
C12	Teses/Dimensões NOK
C13	Má cravação / Soldadura
C14	Mau retrabalho
C15	Cola em falta/Excesso/mal colada
D1	Gordura
D2	Peça partida/danificada
D3	Resíduos (lixo)
E1	Impressão/Laser descentrado
E2	Má impressão/laser
E4	Impressão/laser com cor incorreta
E5	Impressão/laser errada
E6	Dupla impressão
F4	Testes

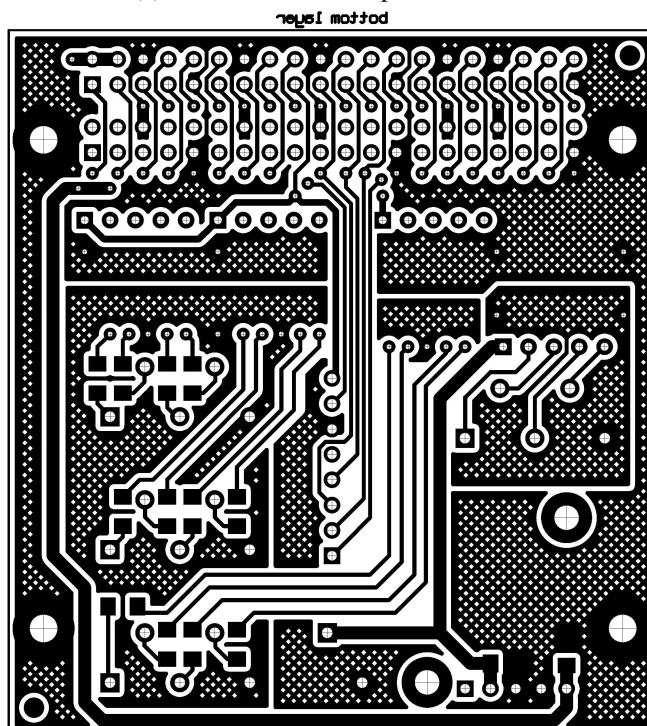
Tabela A.1: Lista de defeitos das peças rejeitadas

Anexo B

Documentos do Circuito Impresso

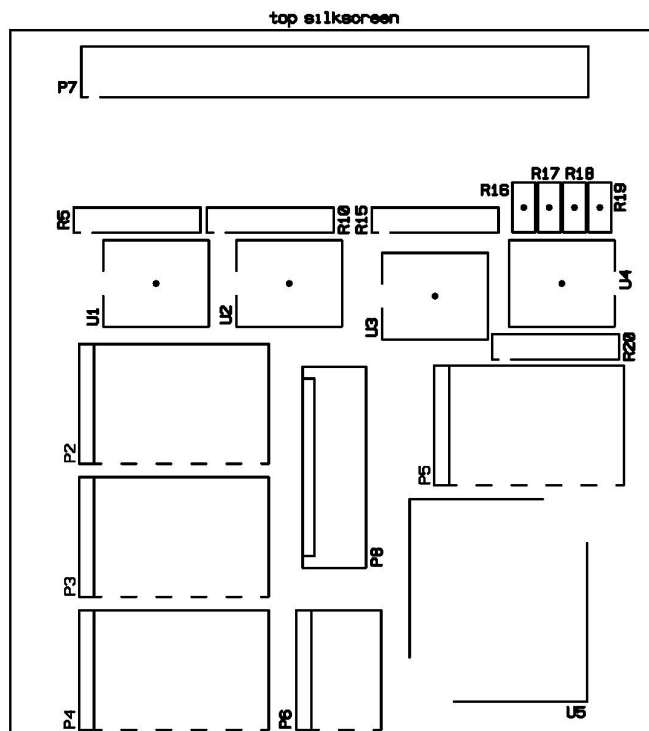


(a) Pistas da camada superior do PCB

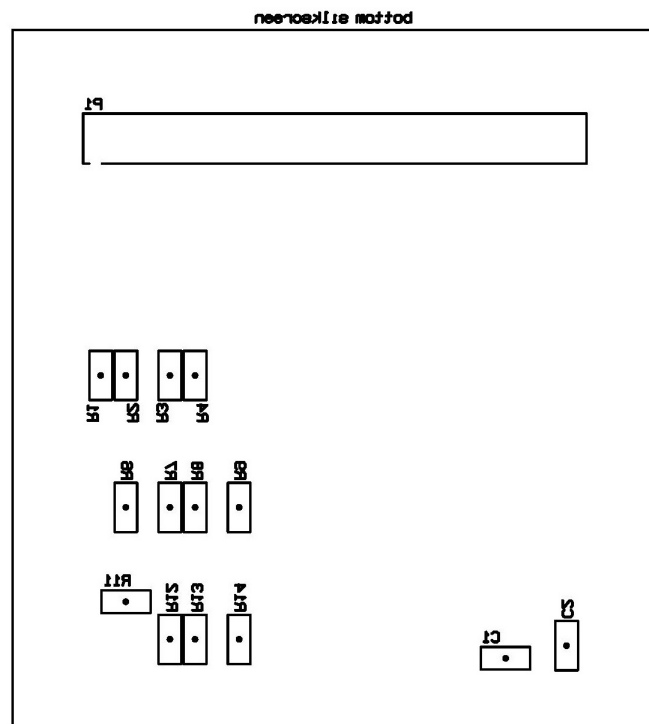


(b) Pistas da camada inferior do PCB

Figura B.2: Pistas das camadas do PCB



(a) Disposição dos componentes na camada superior do PCB



(b) Disposição dos componentes na camada inferior do PCB

Figura B.3: Disposição dos componentes nas camadas do PCB

Anexo C

Documentos de Registo da Atividade de Produção

PRETTTL
automotive

Registo diário de produção - Montagem 87 ASSY016

Referência: RE05180100 - Esquerdo RE05180200 - Direito

OF Nº:

*Selecione a OF de acordo com a ordem numérica a ser trabalhada

Data: / /

Linha / Máquina: /

Turno: 1 2 3 4 5 6 7 8

*Selecione a OF de acordo com a ordem numérica a ser trabalhada

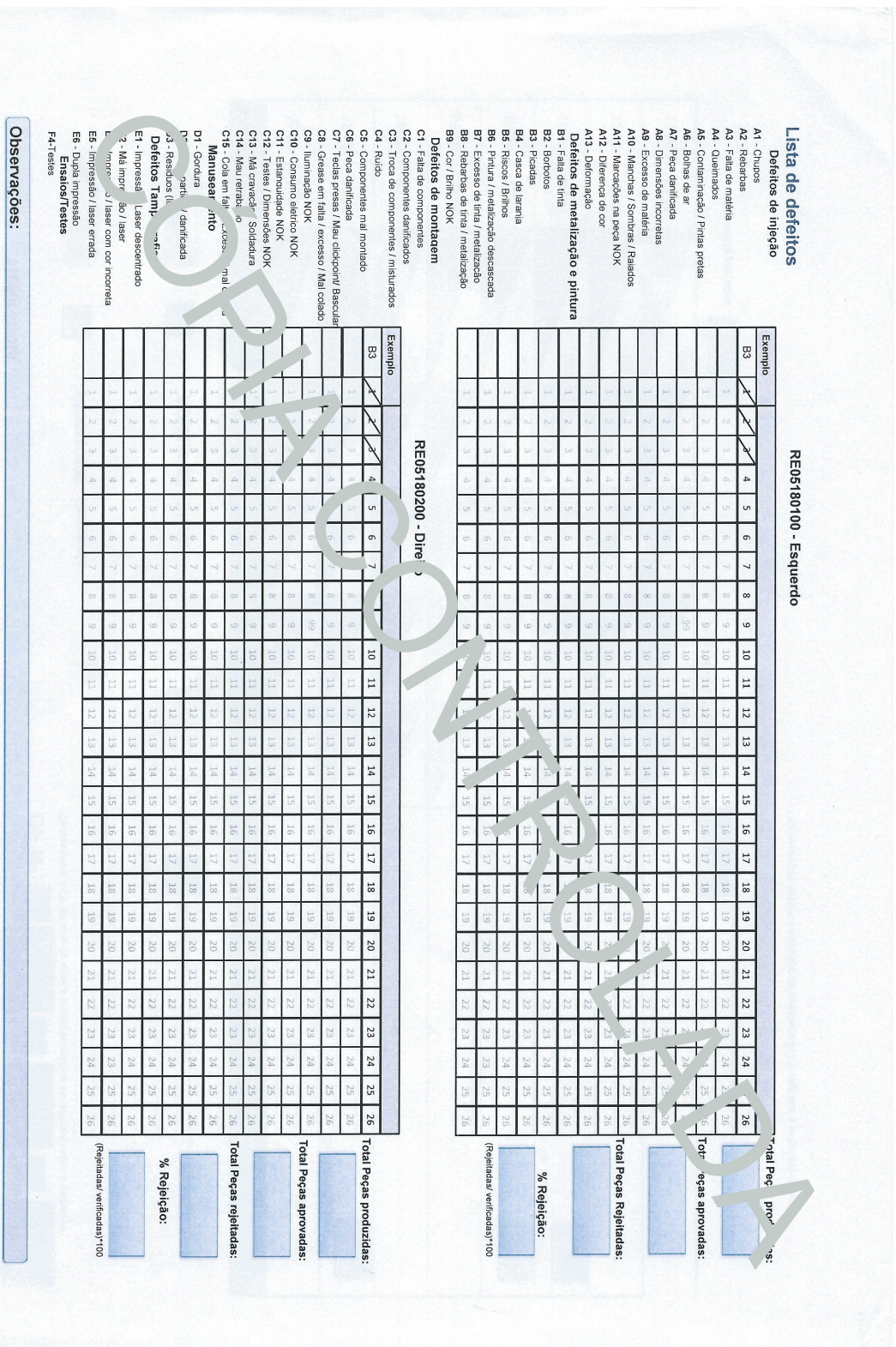
Horário	*Selecione a OF de acordo com a ordem numérica a ser trabalhada		Real		Relação		Registrar causas do desvio ao planeado
	Planeado	Esquerda	Direita	Esquerda	Direita		
06:30 - 07:30	69						
07:40 - 08:30	115	184				Op. Substituição Nº <input type="text"/>	
08:30 - 09:30	138	322					
09:30 - 10:30	138	460					
10:30 - 11:20	115	115				Op. Substituição Nº <input type="text"/>	
11:40 - 12:30	115	690				Op. Substituição Nº <input type="text"/>	
12:30 - 13:30	138	828					
13:30 - 14:00	138	966					
*Objetivo por hora		<input type="text"/> 738 Pares		Total Horas Trabalhadas:		<input type="text"/>	

* se produção total for superior ou inferior a 8:00h registrar a principal causa

Fig. 1/2

PA 0286/

Figura C.1: Folha de identificação e registo dos dados da atividade de produção



Anexo D

Levantamento de Custos do Material

A opção de criar um sistema baseado num *Raspberry Pi* acoplado a um ecrã tátil de 7 polegadas, permite oferecer uma solução de monitorização da produção de baixo custo. A informação presente neste anexo pretende validar essa afirmação fazendo um levantamento dos custos de todo o material constituinte do sistema.

Na tabela D.1 estão listados os componentes integrantes do sistema, a quantidade respetiva de cada um para compor um módulo e o preço respetivo. O preço de impressão do PCB foi alcançado usando o simulador de orçamentação da *Quick-Teck*[43]. Os preços da maior parte dos itens da lista foram retirados da loja *RS Components*[44], com a exceção dos valores do sistema RFID que foram consultados na loja *AliExpress* [45].

Material	Quantidade	Preço por unidade (€)	Preço total (€)
Sistema de Monitorização			
Raspberry Pi 2 Model B	1	33,81	33,81
Raspberry Pi Official Touchscreen	1	62,99	62,99
Circuito Impresso de Conversão			
Conector macho KK de 8 pinos	2	0,31	0,62
Conector fêmea KK de 8 pinos	2	0,51	1,02
Conector de 40 pinos	2	5,70	11,40
Conector PTSA 1.5/2-3.5-Z	1	0,30	0,30
Conector PTSA 1.5/5-3.5-Z	4	0,68	2,72
Optoacoplador TLP281-4	4	1,67	6,67
Step-down voltage converter D24V50F5	1	16,54	16,54
Barramento de resistências de 10k Ω	4	0,10	0,39
Resistência SMD 680 Ω	4	0,02	0,09
Resistência SMD 6,8k Ω	12	0,04	0,50
Impressão do PCB	1	169,50	169,50
Sistema RFID			
Leitor RFID	1	3,17	3,17
Cartões RFID	100	16,14	16,14
Preço Total (€):		287,21	

Tabela D.1: Custos discriminados de todos os materiais que compõem um módulo do sistema

Assim, estes dados levam à conclusão de que o sistema automático de monitorização da produção desenvolvido pode ser implementado numa máquina de injeção por apenas €287,21. No entanto, como existem mais máquinas de injeção na fábrica que poderiam ser favorecidas por esta solução, foi feito um estudo adicional de custos de implementação do sistema em 10 máquinas. Com este volume de encomenda, o custo de impressão por PCB decresce consideravelmente, tal como alguns dos seus componentes, como se pode ver na tabela D.1. Para além disso, nesta quantidade, o preço unitário do leitor RFID também sofre uma redução. No total, a fábrica consegue equipar 10 máquinas de injeção com este sistema por apenas €1550,03. Portanto, é possível concluir que a solução apresentada cumpre o critério de baixo custo.

Material	Quantidade	Preço por unidade (€)	Preço total (€)
Sistema de Monitorização			
Raspberry Pi 2 Model B	10	33,81	330,81
Raspberry Pi Official Touchscreen	10	62,99	620,99
Circuito Impresso de Conversão			
Conector macho KK de 8 pinos	20	0,31	6,22
Conector fêmea KK de 8 pinos	20	0,51	10,16
Conector de 40 pinos	20	5,70	114,00
Conector PTSA 1.5/2-3.5-Z	10	0,30	3,00
Conector PTSA 1.5/5-3.5-Z	40	0,63	25,20
Optoacoplador TLP281-4	40	1,39	55,60
Step-down voltage converter D24V50F5	10	16,54	165,40
Barramento de resistências de 10kΩ	40	0,10	3,92
Resistência SMD 680Ω	40	0,52	20,80
Resistência SMD 6,8kΩ	120	0,04	5,04
Impressão do PCB	10	13,57	135,73
Sistema RFID			
Leitor RFID	1	2,08	20,82
Cartões RFID	100	16,14	16,14
Preço Total (€):	1.550,03		

Tabela D.2: Custos discriminados de todos os materiais que compõem dez módulos do sistema

Referências

- [1] Pete Wood. Raspberry pi model a, 2012. Acedido a 26 de Janeiro de 2016. URL: <http://www.rs-online.com/designspark/electronics/blog/raspberry-pi-model-a>.
- [2] Pete Wood. Raspberry pi model a, 2012. Acedido a 28 de Janeiro de 2016. URL: <https://www.raspberrypi.org/documentation/usage/gpio-plus-and-raspi2/>.
- [3] Matt. Simple guide to the rpi gpio header and pins, 2012. Acedido a 28 de Janeiro de 2016. URL: <http://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/>.
- [4] Arentia. Software produção de plásticos. Acedido a 9 de Fevereiro de 2016. URL: <http://www.arentia.pt/pt/Software-producao-plasticos>.
- [5] Critical Software. Soluções à medida. Acedido a 9 de Fevereiro de 2016. URL: <http://www.criticalsoftware.com/pt/what-we-do/bespoke-solutions>.
- [6] Raspberry Pi Foundation. What is a raspberry pi? Acedido a 26 de Janeiro de 2016. URL: <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>.
- [7] Raspberry Pi Foundation. The making of pi. Acedido a 27 de Janeiro de 2016. URL: <https://www.raspberrypi.org/about/>.
- [8] Vários. *The Official Raspberry Pi Projects Book*. Liz Upton, primeira edição, 2015.
- [9] Richard Lawler. Raspberry pi credit-card sized linux pcs are on sale now, \$25 model a gets a ram bump, Fevereiro 2012. URL: <http://www.engadget.com/2012/02/29/raspberry-pi-credit-card-sized-linux-pcs-are-on-sale-now-25-mo/>.
- [10] Eben Upton. Model b now ships with 512mb of ram, 2012. Acedido a 26 de Janeiro de 2016. URL: <https://www.raspberrypi.org/blog/model-b-now-ships-with-512mb-of-ram/>.
- [11] Eben Upton. Raspberry pi model a+ on sale now at \$20, 2014. Acedido a 26 de Janeiro de 2016. URL: <https://www.raspberrypi.org/blog/raspberry-pi-model-a-plus-on-sale/>.
- [12] Eben Upton. New product launch! introducing raspberry pi model b+, 2014. Acedido a 26 de Janeiro de 2016. URL: <https://www.raspberrypi.org/blog/introducing-raspberry-pi-model-b-plus/>.
- [13] Eben Upton. Raspberry pi 2 on sale now at \$35, Fevereiro 2015. Acedido a 26 de Janeiro de 2016. URL: <https://www.raspberrypi.org/blog/raspberry-pi-2-on-sale>.

- [14] Eben Upton. Raspberry pi 3 on sale now at \$35, Fevereiro 2016.
- [15] Eben Upton. Raspberry pi 2 on sale now at \$35, 2015. Acedido a 26 de Janeiro de 2016. URL: <https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/>.
- [16] Gordon Hollingworth. The eagerly awaited raspberry pi display, 2015. Acedido a 30 de Janeiro de 2016. URL: <https://www.raspberrypi.org/blog/the-eagerly-awaited-raspberry-pi-display/>.
- [17] MIPI Alliance. Display interface specifications. Acedido a 27 de Janeiro de 2016. URL: <http://mipi.org/specifications/display-interface#DSI>.
- [18] Gareth Upton, Eben Halfacree. *Raspberry Pi User Guide*. Wiley, segunda edição, 2014.
- [19] Raspberry Pi Foundation. Frequently asked questions. Acedido a 01 de Fevereiro de 2016. URL: <https://www.raspberrypi.org/help/faqs/>.
- [20] Mark Summerfield. *Rapid GUI Programming with Python and Qt*. Prentice Hall, 2007.
- [21] Dave Kulhman. *A Python Book: Beginning Python, Advanced Python, and Python Exercises*. Dave Kulhman, primeira edição, 2013.
- [22] Fredrik Lundh. *An Introduction to Tkinter*. Fredrik Lundh, 1999.
- [23] Thomas Hansen Oleksandr Lobunets Mathieu Virbel. Kivy – a framework for rapid creation of innovative user interfaces. *Workshopband Mensch Computer 2011*, páginas 69–73, 2001.
- [24] Jorge Rei. *RFID Versus Código de Barras da Produção à Grande Distribuição*. Tese de doutoramento, Faculdade de Engenharia da Universidade do Porto, 2010.
- [25] MIFARE. Mfrc522, 2014. Acedido a 02 de Fevereiro de 2016. URL: http://www.nxp.com/documents/data_sheet/MFRC522.pdf.
- [26] A. Silberschatz, H. Korth, e S. Sudarshan. *Database System Concepts*. McGraw-Hill Education, 2010.
- [27] Mike Owens e Grant Allen. *SQLite*. Springer, 2010.
- [28] Carlos Eduardo Santos Pires, Rilson O Nascimento, e Ana Carolina Salgado. Comparativo de desempenho entre bancos de dados de código aberto. *Escola Regional de Banco de Dados, Anais da ERBD06, Porto Alegre*, 2006.
- [29] Rodrigo Evangelista Delgado, Pedro Lucas Lopes Zimmermann, e Ivanildo Barbosa. Comparação do desempenho de extensões espaciais de sgbd: Postgis e spatialite.
- [30] Esen Sagynov. Cubrid vs. mysql vs. postgresql release period comparison. Acedido a 03 de Fevereiro de 2016. URL: <http://www.cubrid.org/blog/cubrid-comparison/cubrid-vs-mysql-vs-postgresql-release-period-comparison/>.
- [31] Ben Croston. Rpi.gpio python module. Acedido a 06 de Junho de 2016. URL: <https://sourceforge.net/p/raspberry-gpio-python/wiki/Home/>.
- [32] Vishay Semiconductors. *Optocoupler, Phototransistor Output, with Base Connection*.
- [33] Toshiba. *TOSHIBA PHOTOCOUPLER GaAs IRED PHOTO-TRANSISTOR*, Setembro 2014.

- [34] Texas Instruments. *A7800 Series - Positive-Voltage Regulators*.
- [35] Pololu Robots Electronics. Pololu 5v, 5a step-down voltage regulator d24v50f5. Acedido a 30 de Maio de 2016. URL: <https://www.pololu.com/product/2851/specs>.
- [36] Phoenix Contact. *PTSA 1,5/5-3,5-Z*, Julho 2011.
- [37] Molex. *KK® 254 Solid Header, Vertical, with Friction Lock, 8 Circuits, Tin (Sn) Plating*, Setembro 2015.
- [38] Samtec. *Elevated Socket*.
- [39] Daniele Varrazzo. Psycopg – postgresql database adapter for python. Acedido a 30 de Maio de 2016. URL: <http://initd.org/psycopg/docs/>.
- [40] Randy Abramowitz Mark Morales, Ramiro Syring. Psycopg – postgresql database adapter for python. Acedido a 30 de Maio de 2016. URL: <http://www.pymssql.org/en/latest/>.
- [41] James Cameron. Pptp client, Maio 2008.
- [42] Mario Gómez. Mfrc522-python. Acedido a 05 de Junho de 2016. URL: <https://github.com/mxgxw/MFRC522-python>.
- [43] Quick-teck. Acedido a 20 de Junho de 2016. URL: <http://www.quick-teck.co.uk/>.
- [44] Rs components. Acedido a 20 de Junho de 2016. URL: <http://pt.rs-online.com/web/>.
- [45] aliexpress. Acedido a 20 de Junho de 2016. URL: <http://pt.aliexpress.com/>.