

SISTEMA INTEGRADO DE MONITORIZAÇÃO DA MAQUINAÇÃO DE MOLDES DO GRUPO SIMOLDES

JOSÉ EDUARDO FAUSTINO BARROS
DISSERTAÇÃO REALIZADA NO ÂMBITO DO
MESTRADO INTEGRADO EM ENGENHARIA METALÚRGICA E DE MATERIAIS

ORIENTADOR

PROFESSOR VITOR MANUEL BRANCO MARTINS AUGUSTO

<i>CANDIDATO</i>	José Eduardo Faustino Barros	<i>Código</i>	201405236
<i>TÍTULO</i>	SISTEMA INTEGRADO DE MONITORIZAÇÃO DA MAQUINAÇÃO DE MOLDES DO GRUPO SIMOLDES		
<i>DATA</i>	24 de Julho de 2020		
<i>LOCAL</i>	Faculdade de Engenharia da Universidade do Porto		

<i>Presidente</i>	Professor Doutor Fernando Jorge Mendes Monteiro	DEMM/FEUP
<i>Arguente</i>	Professor Doutor Abílio Manuel Pinho de Jesus	DEM/FEUP
<i>Orientador FEUP</i>	Professor Vitor Manuel Branco Martins Augusto	DEMM/FEUP
<i>Orientadora Simoldes</i>	Engenheira Jessica Ann da Silva	DEM/FEUP

Resumo

A presente dissertação foi realizada em contexto industrial na equipa de métodos e processos da *Tool Division* (divisão de aços) do Grupo Simoldes, em Oliveira de Azeméis. O desafio proposto passou por melhorar o sistema de monitorização de equipamentos CNC. O trabalho desenvolvido ao longo do semestre consistiu no estudo e análise do sistema de monitorização ao mesmo tempo que é garantido o seu correto funcionamento e manutenção. Serão identificados problemas, limitações e falhas que foram sendo progressivamente melhorados. Este sistema é indispensável à produção eficiente de moldes para injeção de plástico. O trabalho pode ser dividido em várias etapas, primeiro é feita uma contextualização introdutória da indústria de moldes, do Grupo Simoldes, do conceito de molde para injeção de plástico entre outras questões pertinentes para a dissertação, depois é escrutinada a gestão envolvida na produção de moldes, as inúmeras variáveis associadas à produção e as várias etapas e instâncias em que pode ser dividida, é então feita a transição para o sistema de monitorização das máquinas CNC do grupo, o seu estudo, análise, funcionamento, manutenção, problemas/limitações e consequentes desenvolvimentos. No final é feita uma última análise ao sistema, sendo identificadas as principais necessidades ou limitações que possam ainda existir, este levantamento deve ser tido em conta em soluções futuras que possam vir a ser desenvolvidas, interna ou externamente ao Grupo Simoldes.

Palavras Chave

Sistema de Monitorização; Controlo Numérico; Simoldes; Molde; Plástico

Índice

1. Introdução.....	1
1.1. Objetivos	1
1.2. Contextualização	1
1.2.1. Indústria de Moldes e Grupo Simoldes.....	1
1.2.2. Conceito de molde de injeção de plástico.....	4
2. Gestão da Produção de Moldes para Injeção de Plástico.....	7
3. Método de Monitorização e Gestão da Fresagem CNC	14
3.1. Contextualização	14
3.2. Análise Técnica	16
3.2.1. Recolha de Informação - Equipamentos Fidia	17
3.2.2. Recolha de Informação - Equipamentos Heidenhain iTNC530.....	21
3.2.3. Recolha de Informação - Equipamentos Heidenhain TNC640	23
3.2.4. Automatismo de Cálculo de Percentagem Diária	25
3.2.5. Sistema de Monitorização - Bases de Dados	25
3.3. Análise SWOT do Sistema de Monitorização	31
4. Melhorias do Sistema Integrado de Monitorização	34
4.1. Introdução Equipamentos CNC no Sistema	34
4.1.1. Equipamentos Selca	34
4.1.2. Outros Controlos Numéricos.....	36
4.2. Análise e Melhoria da Qualidade dos Dados Recolhidos.....	37
4.2.1. Exclusão de Equipamentos do Sistema de Monitorização	37
4.2.2. Melhoria na Recolha de Dados	38
4.3. Melhorias na Interface com o Utilizador	39
4.3.1. Programas NC e Mensagens de Erro	39
4.4. Balanço Final e Futuro do Sistema de Monitorização	41
5. Considerações Finais.....	42
Bibliografia.....	43
Anexos	45
Anexo A - Automatismo de recolha de informação em comandos Fidia	45
Anexo B - Automatismo de recolha de informação de comandos CNC iTNC 530 Heidenhain	70
Anexo C - Automatismo de recolha de informação em comandos CNC TNC 640 Heidenhain	90

Anexo D - Automatismo de compilação diária de informação	99
Anexo E - Código do menu inicial do <i>Front End</i>	101
Anexo F - Código correspondente ao <i>Dashboard</i> Empresa.....	106
Anexo G - Código correspondente ao <i>Dashboard</i> Máquina.....	110
Anexo H - Automatismo de recolha de informação desenvolvido para comandos Selca (instancia para máquinas de fresagem normal).....	127
Anexo I - Desenvolvimentos para comandos TNC 640 para recolha de programas NC	130
Anexo J - Desenvolvimentos do <i>Front End</i> para programas NC e mensagens de erro	130

1. Introdução

1.1. Objetivos

A presente dissertação foi realizada em contexto industrial na equipa de métodos e processos da *Tool Division* (divisão de aços) do Grupo Simoldes, em Oliveira de Azeméis. Os principais objetivos do trabalho desenvolvido ao longo do semestre foram:

- Em primeiro lugar estudar e realizar uma análise SWOT ao atual sistema de monitorização de equipamentos CNC do Grupo Simoldes. Desta forma identificar-se-ão os pontos fortes e as limitações do sistema, assim como as oportunidades de melhoria e ameaças para o bom funcionamento do sistema;
- Em segundo, implementar todas as alterações identificadas como necessárias para melhorar o sistema de monitorização e perceber se foram ou não proveitosas;
- Por último avaliar de novo o sistema de monitorização, no que toca aos novos desenvolvimentos implementados, concluindo que necessidades/limitações o sistema continua a ter. Estas necessidades ou limitações terão de ser tidas em conta em futuros desenvolvimentos, como um caderno de encargos geral.

1.2. Contextualização

1.2.1. Indústria de Moldes e Grupo Simoldes

A indústria de moldes tem crescido bastante em Portugal desde os anos 90, nos dias que correm o nosso país tem bastante importância no panorama internacional deste mercado, a procura externa é elevada sendo assegurada pelo conjunto de competências e capacidades das empresas portuguesas. Portugal é o 8º maior fabricante de moldes no mundo e o 3º na Europa. O setor distribui-se principalmente pelas zonas de Oliveira de Azeméis e Marinha Grande, das 536 empresas existentes a maioria são de pequena e média dimensão, empregando cerca de 11 000 pessoas [1], [2].

No ano de 2019 o valor das exportações do setor atingiu um valor de aproximadamente 614 milhões de euros (valores provisórios), cerca de 90% da produção total com valores a rondar os 682 milhões de euros. A indústria de moldes é dos maiores setores exportadores do país [1].

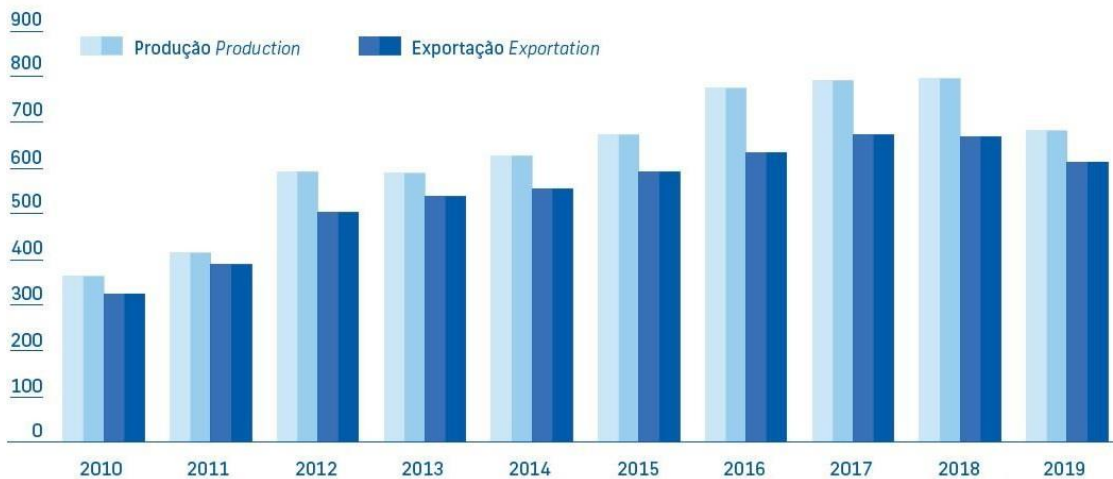


Figura 1 - Comparação em milhões de euros da produção com a exportação do setor [1]

Analisando a figura 1 percebe-se que o ano de 2018 foi o mais produtivo desde 2010, depois de um crescimento ao longo da década, o pequeno decréscimo de 2019 pode ser explicado por uma indefinição geral na indústria automóvel no rumo a tomar no setor [1]. Os principais países de destino da indústria de moldes portuguesa são a Espanha (23%), Alemanha (19%), França (12%), República Checa (7%), Polónia (5%) e Estados Unidos da América (3%), entre outros como Reino Unido, Eslováquia, Rússia e Itália. Em termos das áreas de destino dos moldes portugueses o principal é a indústria automóvel (82%) seguindo-se o das embalagens (8%) entre outros como dispositivos médicos, eletrónica e mobiliário [1].

A Simoldes é um grupo português que opera no setor dos moldes, maioritariamente para o setor automóvel, quer na injeção de peças de plástico quer na produção de moldes para este tipo de peças. Foi fundado em 1959 e hoje em dia é dos mais importantes grupos a nível mundial no setor, é composto por duas divisões, a *Plastic Division* (ou divisão de plásticos) e pela *Tool Division* ou (divisão de aços), a primeira dedica-se como o nome indica à produção de peças de plástico obtidas por injeção, a segunda dedica-se à produção de moldes para injeção de plástico maioritariamente para a indústria automóvel, sendo considerada a maior produtora a nível europeu[3], [4], [5].



Figura 2 - Vista aérea de parte do Grupo Simoldes em Oliveira de Azeméis (6 empresas)

O Grupo Simoldes começou com a Simoldes Aços, fundada em 1959 pelo comendador António Silva Rodrigues em Oliveira de Azeméis, em 1980 foi fundada a Simoldes Plásticos e com ela nasce a divisão de plásticos do grupo, em 1991 foi fundada a MDA (Moldes de Azeméis) de modo a dar mais capacidade de produção à divisão de aços, na década de 1990 foram ainda fundadas mais 6 empresas (contabilizando ambos os grupos), hoje em dia a *Tool Division* é composta por 8 empresas (6 delas em Portugal) e a *Plastic Division* por 9 empresas (3 delas em Portugal). Na figura 3 está ilustrada a estrutura do grupo, apresentando as divisões e empresas que existem. Existem ainda escritórios (que não estão apresentados na figura) situados em locais estratégicos do globo de modo a apoiar novos e atuais clientes do grupo Simoldes [4], [5].

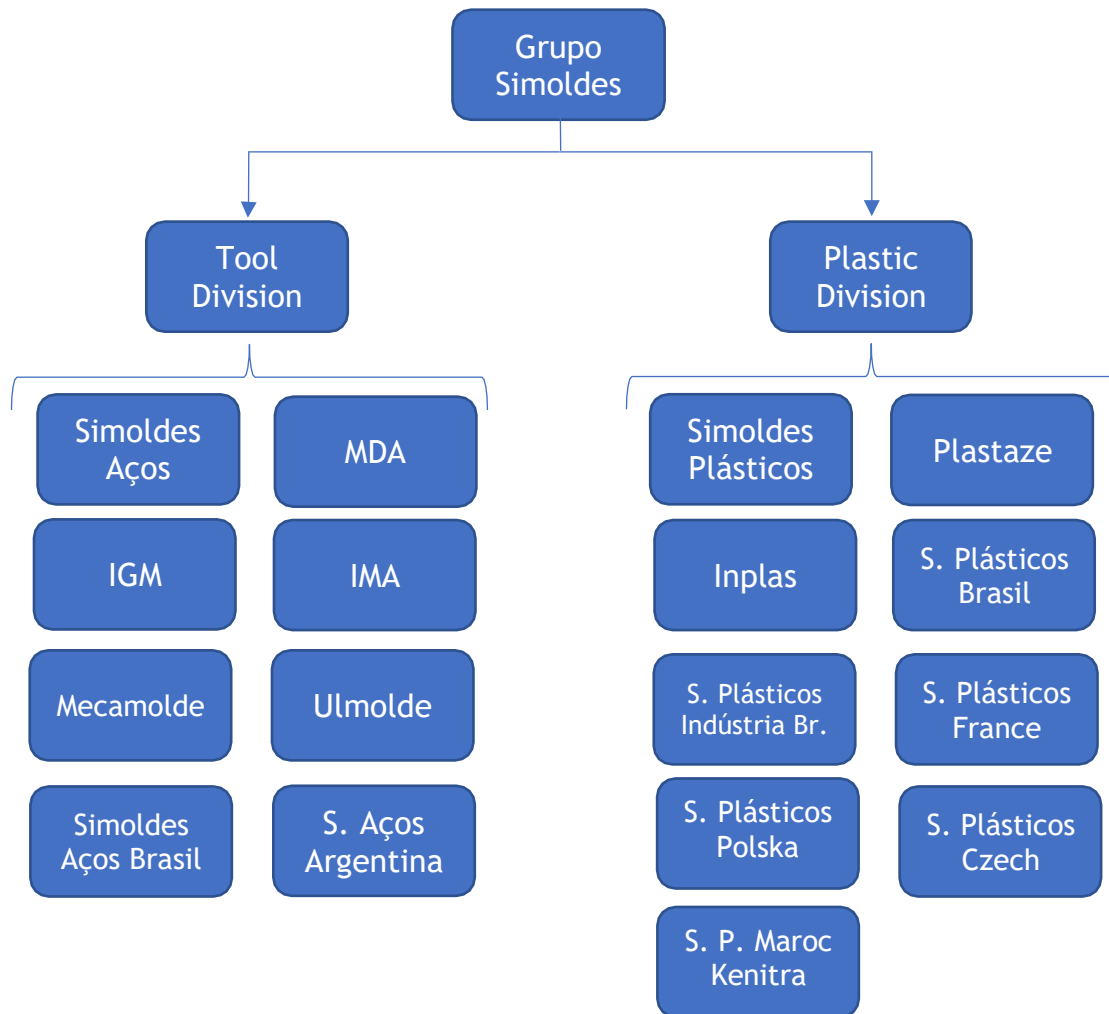


Figura 3 - Apresentação das diferentes divisões e empresas do Grupo Simoldes

1.2.2. Conceito de molde de injeção de plástico

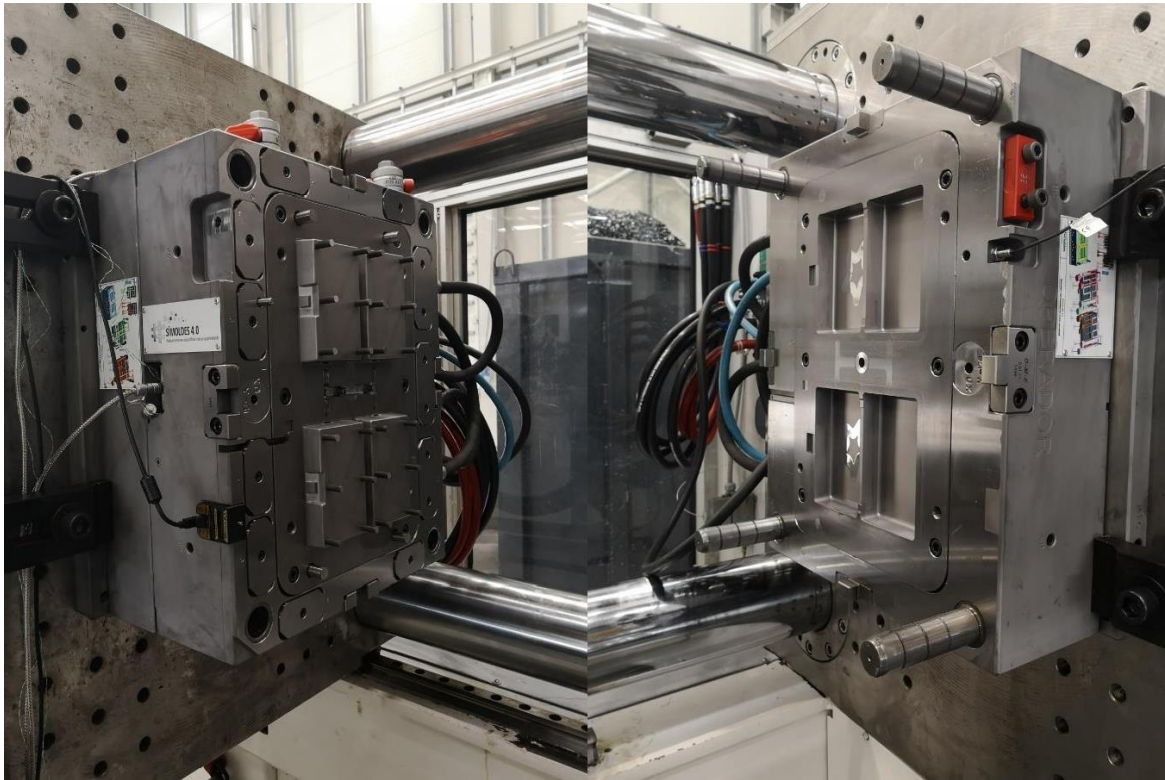
Um molde é uma ferramenta destinada a dar forma a um determinado material. No caso da injeção de plástico (exemplificado na figura 4) um molde trata-se de um modelo oco formado por vários componentes que em conjunto formam uma ou várias peças, neste caso de plástico. No entanto, um molde pode também ser um modelo sólido pelo qual algo se forma, por compressão [2], [6].



Figura 4 - Exemplo de um molde (fechado)

Os termoplásticos são materiais ideais para serem moldados, tornam-se maleáveis a temperaturas por norma entre os 200 e os 300 °C, temperaturas estas alcançadas com relativa facilidade em comparação, por exemplo, com as temperaturas necessárias para fins semelhantes nos metais. A moldação por injeção de plástico consiste, como o nome indica na injeção de material plástico, aquecido a uma certa temperatura, para o interior de um molde havendo um preenchimento correto do seu interior, de maneira a obter uma peça conforme o pretendido [2], [7].

Relativamente à estrutura de um molde para injeção de plástico, no caso de peças para indústria automóvel, trata-se de uma estrutura bastante complexa, com muitos componentes ligados entre si e com movimentos relativos entre eles. Cada molde é único e dependendo do pretendido pelo cliente a sua estrutura pode variar, no entanto por norma um molde é constituído por duas placas (uma fixa e uma móvel) que suportam os elementos moldantes (macho e cavidade), o sistema de injeção, sistema de extração, sistema de transferência de calor, assim como calços e outros acessórios necessários ao molde. Nas figuras 5 e 6 temos um exemplo de um molde aberto [2], [6], [8].



Figuras 5 e 6 - Exemplo de um molde (aberto), macho e cavidade respetivamente

- Os elementos moldantes são os componentes do molde que estão em contacto com o plástico introduzido. Os principais elementos moldantes são o macho e a cavidade do molde, podendo haver outros elementos em contacto com a peça, como postigos ou extratores [2], [6], [8];
- O sistema de injeção (ou alimentação) é constituído por componentes metálicos, como o injetor principal e canais de distribuição, que têm a função de levar o material plástico (viscoso devido à temperatura) para o interior do molde [2], [6], [8];
- O sistema de extração é o conjunto de componentes do molde que, após a injeção, extraem ou “descolam” a peça de plástico do molde. Os extratores são por norma aros compridos de aço que “empurram” a peça para fora do molde de modo a poder ser retirada, por exemplo por um sistema de ventosas [2], [6], [8];
- O sistema de transferência de calor tem a função de manipular a temperatura do plástico injetado de acordo com o planeado. Consiste num sistema de canais no interior do molde, por onde circula um fluído (por norma água) com temperatura controlada, quer à entrada quer à saída do molde [2], [6], [8];
- Os calços permitem o espaço necessário ao funcionamento do sistema de extração, podendo também ter a função de conferir ao molde a dimensão necessária para o seu funcionamento na máquina de injeção em que será instalado [2], [6], [8].

Na figura 7 temos um exemplo da cavidade de um molde, é possível identificar componentes como guias e postigos (neste caso de cor dourada).

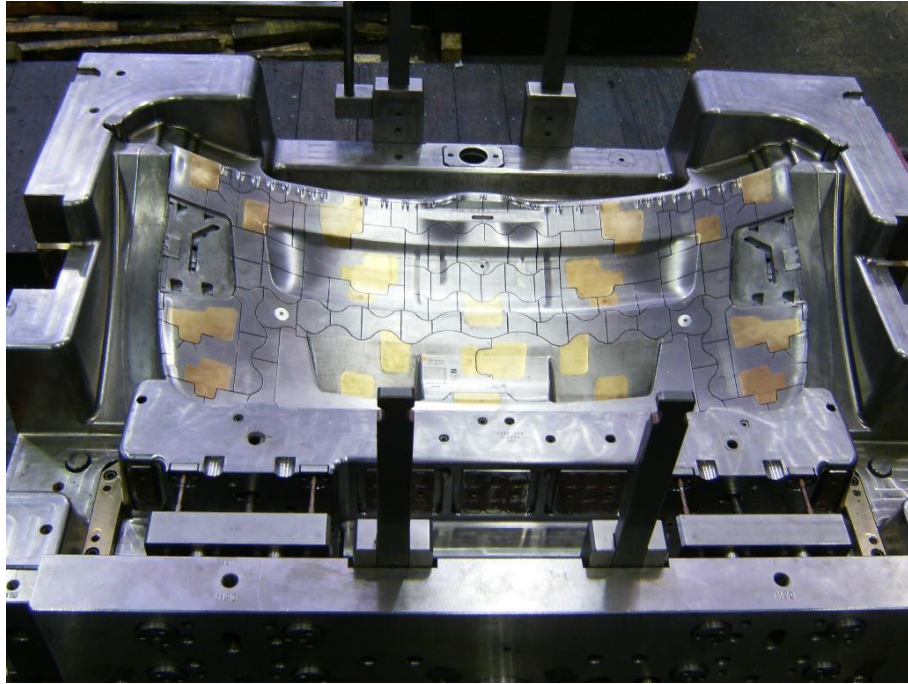


Figura 7 - Exemplo de um molde (cavidade)

2. Gestão da Produção de Moldes para Injeção de Plástico

O processo de desenvolvimento do molde inicia-se na análise dos ficheiros 2D/3D da peça de plástico e especificações técnicas da mesma. Esta análise consiste na verificação da factibilidade e viabilidade de fabrico do molde com base no polímero a usar, na espessura geral da peça, na espessura de zonas técnicas, na geometria da peça, potenciais problemas de desmoldagem (problemas na recolha da peça durante o processo de injeção), identificação de zonas de difícil acesso de refrigeração ou mesmo pontos quentes (*hotspots*) que impactam muito negativamente os tempos de ciclo e taxas de rejeição. Nesta fase verificam-se ainda aspetos específicos e técnicos da conceção do molde, tais como a análise de linhas de junta (conjunto de superfícies em torno do design da peça plástica que vão definir a separação entre as duas metades do molde), a desmoldagem de alguns componentes face a geometrias complexas da peça de plástico, as fragilidades do molde, entre outras questões [7], [9].

Tendo em conta todos os aspetos acima mencionados o objetivo é encontrar soluções para que a peça pretendida seja um produto exequível com os menores tempos de ciclo, com a menor taxa de rejeição e com a garantia que o molde que a produz é suficientemente robusto para suportar o ciclo de vida esperado. O método de produção de um molde é, portanto, um processo complexo com diversas variáveis a ter em conta, e com várias possibilidades de chegar a um molde conforme os requisitos pretendidos pelo cliente no final. A figura 8 é um VSM (*Value Stream Mapping*) onde está mapeada toda a cadeia de valor associada à produção, neste caso, de um percurso de um macho de um molde de painel de porta, na Simoldes Aços [10], [11].

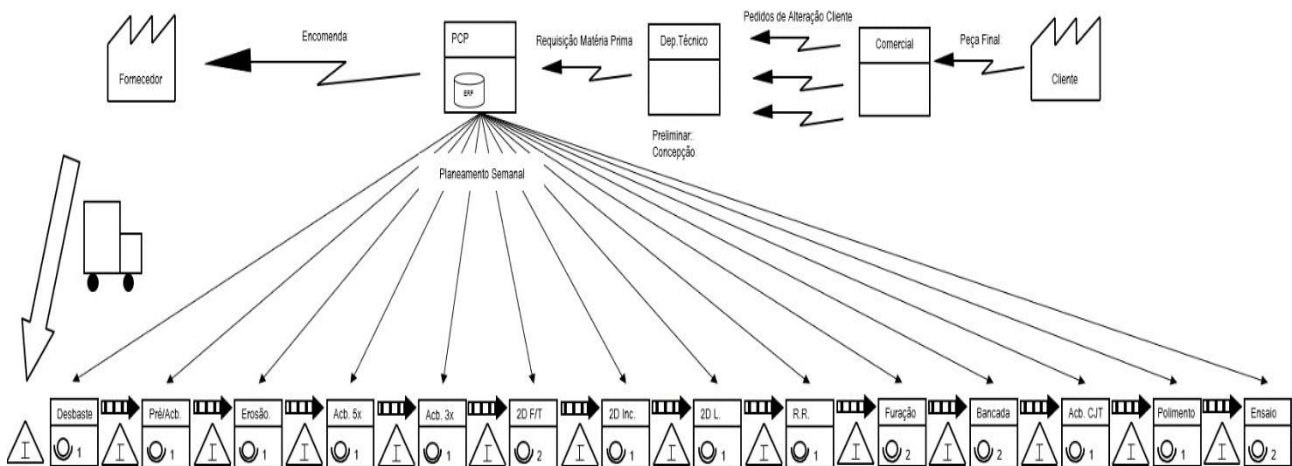


Figura 8 - Processo de fabrico de um molde de injeção de plástico (Simoldes painel porta)

Após orçamentação e todo o estudo/desenvolvimento e projeção por parte do departamento técnico ser aprovado dá-se início ao processo de produção do molde no chão de fábrica começando com a chegada do bloco de aço, a sua composição química e tratamentos térmicos aplicados variam dependendo do pretendido para o molde e do processo de fabrico planeado para o mesmo. Com um bloco de aço conforme o especificado no planeamento/projeção do molde, quer a nível de dimensional quer a nível de composição química e microestrutura, é possível iniciar a remoção de material do mesmo. A primeira etapa é denominada de desbaste e é a primeira estratégia de maquinação aplicada ao bloco, é por isso um trabalho mais grosseiro, realizado com ferramentas grandes e pouco precisas, sendo a tolerância grande o suficiente para manter a zona útil do aço intacta, ao mesmo tempo que se pretende retirar o máximo de material possível ao bloco de modo a encurtar futuras estratégias de maquinação [6], [8], [10], [12].

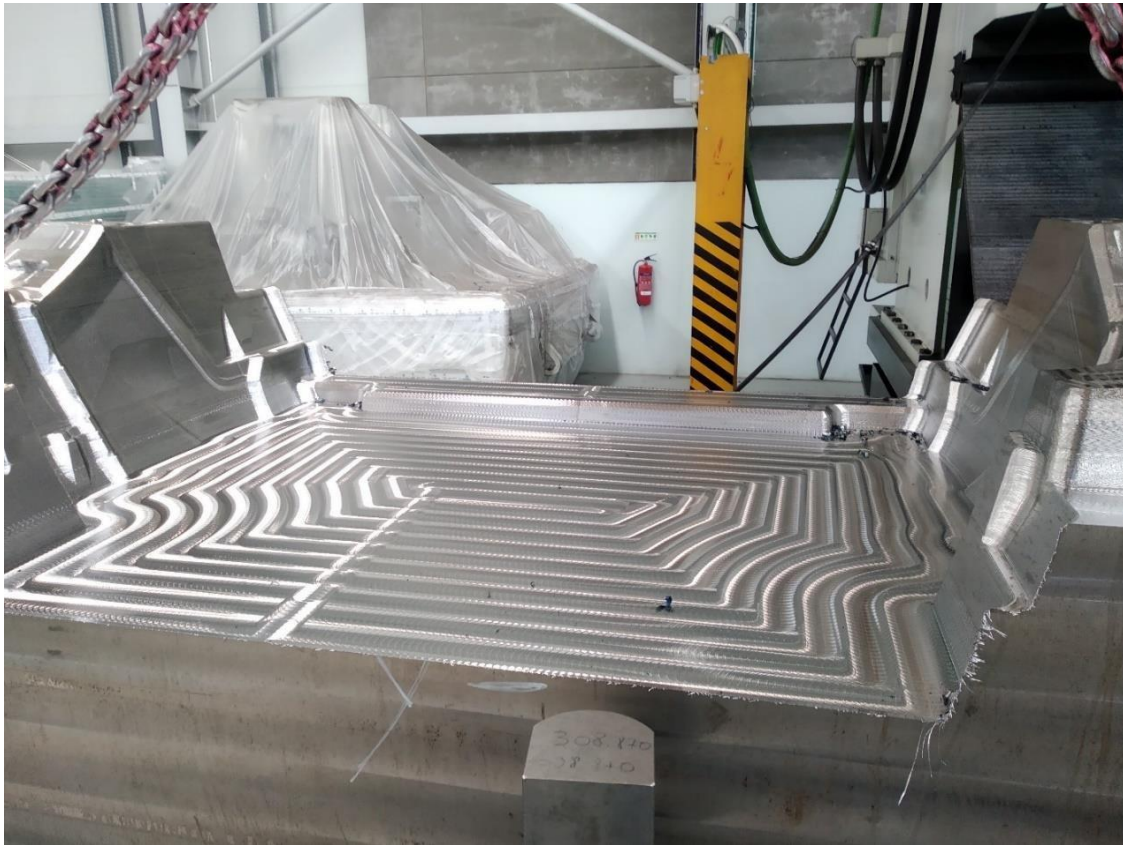


Figura 9 - Exemplo de uma cavidade desbastada

O semiacabamento (ou redesbaste) é a etapa seguinte, sendo um meio termo entre desbaste e acabamento, a taxa de remoção de material é inferior à do desbaste mas a ferramenta aproxima-se mais da zona útil do bloco, permitindo assim encurtar estratégias de acabamento futuras começando o trabalho mais próximo da zona útil do aço. O redesbaste, assim como as restantes etapas, tem de ser bem planeado à priori, visto que, um bom trabalho nesta fase pode permitir a poupanças de tempo em futuras etapas [6], [8], [12].

De seguida inicia-se o acabamento, está dividido em diferentes estratégias ou técnicas de trabalho, definidas anteriormente no departamento CAM. Dependendo do molde e da maquinação anteriormente aplicada no bloco escolhem-se as estratégias mais vantajosas de modo ao molde ficar dentro das tolerâncias e requisitos do cliente no menor período de tempo. Além de macho e cavidade um projeto de um molde envolve vários sistemas internos ao seu funcionamento, como o sistema hidráulico do molde e o sistema de transferência de calor, assim como componentes como: placas, calços, guias de extração, extratores, postigos; por vezes podem chegar à ordem dos milhares de componentes que, quando assemblados com os blocos principais formam o molde. Consequentemente, para a produção de um molde existem ainda diferentes etapas/tarefas de maquinação além das já apresentadas, como aplicação de percursos de geometria mais linear como caixas, furos, chanfros, entre outros (conhecidos na gíria por '2D'), maquinação por torneamento e furação para por exemplo o sistema de transferência de calor do molde. Na figura 10 é possível identificar vários tipos de maquinação referidos, acabamento na estrutura geral do bloco, mas também muitas

geometrias mais simples como caixas ou furações, importantes e fundamentais para o bom funcionamento do molde [6], [8].



Figura 10 - Bloco principal de um molde, exemplo de maquinações de acabamento e '2D'

Por último, em zonas específicas pode ser necessária a realização de polimento, texturização ou outro tipo de procedimento especial. Por exemplo se um molde for exportado por via marítima é importante proporcionar-lhe maior proteção contra a corrosão (tratamentos térmicos e/ou revestimentos no final da maquinação) para que no seu destino mantenha todas as propriedades e características que tinha à saída de fábrica. No caso de serem exigidas alterações ao molde pelo cliente durante a sua produção o processo de solda pode ser necessário [6], [8], [12].

O trabalho de montagem de todos os componentes é a última linha de trabalho na produção de moldes, o setor denomina-se 'Bancada' podendo envolver afinações de carácter manual além da montagem. Os técnicos de bancada são elementos com conhecimento e *know-how* especializado na área de moldes para injeção de plástico, além da montagem acompanham os ensaios e muitas vezes a instalação do molde no cliente de modo a garantir a conformidade da ferramenta de moldação produzida. O molde obtido é ensaiado em máquinas de injeção antes de ser expedido para o cliente onde é novamente testado antes de entrar em produção, de forma a garantir que as peças de plástico produzidas estão dentro dos requisitos pretendidos [6], [8], [10], [12].



Figura 11 - Setor de bancada da Simoldes Aços

Com toda a envolvimento do molde, qualquer etapa do processo tem diversas variáveis associadas, podendo-se optar por uma ou outra opção dependendo da situação, mas sempre com o objetivo de um molde conforme os requisitos do cliente no final da produção. Destas opções (ou variáveis) associadas ao processo as que podem ser consideradas mais importantes são:

- Equipamentos de maquinação - diferem entre fresadoras, mandriladoras, tornos, retificadoras entre outros equipamentos mais específicos de maquinação, estes equipamentos podem ser automáticos com controlo numérico computadorizado (CNC's) ou manuais (para trabalhos mais lineares). A gestão associada a estes meios verifica-se bastante importante, sabendo que um molde é constituído por centenas/milhares de componentes, a sua distribuição pelas máquinas disponíveis deve ser a melhor possível, a adjudicação a uma ou outra máquina tem efeito não só no tempo necessário para a sua realização mas também na qualidade do componente maquinado; Os principais fabricantes de centros de maquinação presentes no Grupo Simoldes são FPT, RAMBAUDI, CHETO, DMG Mori, entre outras [6], [12];



Figura 12 - Centro de maquinação de alta precisão FPT *Dinox*

- Controlos Numéricos - Dispositivos que automatizam equipamentos de maquinação (controlo numérico computadorizado - CNC's) existem vários fabricantes, mas todos trabalham com lógica semelhante, por exemplo: Heidenhain, Fidia, Selca, Fagor, entre outros [12];
- Ferramenta - existem vários tipos de ferramenta, de diferente tipos, tamanhos e materiais, existem ferramentas mais apropriadas para certos tipos de trabalhos e outras que acabam por ser mais gerais. Os principais grupos de ferramenta presentes no Grupo Simoldes são as fresas e as brocas, podendo ser uma ferramenta inteira ou ter associadas pastilhas que são trocadas à medida que se desgastam. Existem ainda outros componentes associados às ferramentas como cones ou alongas que dependendo do trabalho pretendido são assemblados com as ferramentas. Está associada uma gestão muito importante nestes consumíveis, a escolha de uma certa ferramenta em detrimento de outra pode significar um trabalho mais eficiente. O mercado das ferramentas para maquinação de metais é bastante denso, existindo muitas soluções para todo o tipo de necessidades [13];



Figura 13 - Exemplos de uma fresa de pastilhas e uma broca inteira [13]

- Aço - o aço utilizado para a produção de moldes é mais uma variável (macro) do processo, dependendo da liga o comportamento do molde em trabalho pode diferir assim como o trabalho de remoção de material do mesmo na produção. As ligas com mais preponderância na produção de moldes são (segundo a nomenclatura AISI) o aço P20 e o H11 [2], [6];
- Softwares CAD/CAM - Os softwares CAD/CAM (*Computer Aided Design* e *Computer Aided Manufacturing*) são ferramentas muito importantes na produção de moldes para injeção de plástico nos dias de hoje. Os softwares CAD permitem a modelação 3D do molde e todos os seus componentes, as ferramentas CAM permitem a programação e previsão dos percursos de maquinação a realizar no aço. Existe uma grande variedade de softwares CAD/CAM no mercado atual, no Grupo Simoldes estão disponíveis vários: Catia, NX, Fusion 360, PowerMILL e RTM [2], [12];
- As pessoas - os recursos humanos são a peça basilar para a produção de moldes, são o elo de ligação de todas as ferramentas citadas anteriormente, o *know-how* é algo que não se guarda, mas que se transmite. A gestão de todas as pessoas envolvidas é algo ainda mais delicado que nos pontos anteriores, visto estar ligado com quase tudo o que contribui para um molde conforme.

Como é perceptível existe muita gestão associada ao fabrico de um molde, então quando se produzem dezenas/centenas de moldes em simultâneo, a sua gestão é também exponencialmente mais complexa. Por exemplo, dependendo da máquina que será usada, quer por requisitos de máquina quer por disponibilidade da mesma (tendo em conta que há vários blocos a serem trabalhados em simultâneo) haverá maior ou menor precisão no trabalho realizado e também maior ou menor eficiência. O mesmo se passa com o tipo e tamanho de ferramenta utilizada, resultando em maior ou menor precisão e também em mais ou menos material removido por passagem. Outros fatores como o tipo de

refrigeração utilizada, o tipo de controlo numérico da máquina, os softwares CAD e CAM utilizados no processo de desenvolvimento do molde entre outros, influenciam também o tempo e a dificuldade em maquinar o molde de forma eficiente [2], [6], [12].

3. Método de Monitorização e Gestão da Fresagem CNC

3.1. Contextualização

Tendo em conta todas as variáveis associadas à produção de moldes para injeção de plástico é normal haver a necessidade de controlar ao máximo o processo. Neste capítulo será abordado o sistema integrado de monitorização da maquinação CNC de moldes do Grupo Simoldes, começando por uma contextualização introdutória do ponto de situação em termos de dados estatísticos de tempos envolvidos no fabrico de moldes, de tempos de paragem de máquina e das limitações envolvidas na recolha destes dados [6].

Em primeiro lugar é preciso enquadrar o número de equipamentos existentes no grupo, o número atual é de 228 equipamentos de maquinação em solo português, divididos entre fresadoras com controlo numérico (CNC's), fresadoras convencionais, mandriladoras, tornos, retificadoras entre outros equipamentos mais específicos de maquinação. Há uma constante procura pela melhoria, existe movimentação de máquinas ao longo do tempo em todo o grupo havendo também, uma constante renovação destes equipamentos com o passar dos anos, de modo a dar resposta às necessidades do mercado e de forma procurar inovação e eficiência na indústria. De seguida é preciso perceber outro tipo de variáveis como o tempo médio de maquinação por molde, tempo de maquinação médio anual (por empresa e por grupo) e também o tempo de paragem anual por máquina de modo a perceber verdadeiramente a necessidade de monitorização dos equipamentos maquinação [14].

Os tempos de maquinação por molde variam bastante, o principal fator é a geometria da peça de plástico projetada, por exemplo moldes de painéis de interiores de portas automóveis serão sempre maiores e mais complexos que moldes projetados para produzir peças para viseiras de proteção (usuais na conjuntura atual de pandemia). Por outro lado, há uma panóplia de variáveis associadas como as máquinas utilizadas na maquinação, os recursos humanos associados, o tipo de aço utilizado entre outros [2], [8], [14], [15].

De seguida está apresentada uma tabela com os tempos médios de maquinação por molde anualmente, por empresa e de modo global ao Grupo Simoldes (empresas em Portugal):

Tabela 1 - Tempos de maquinação por molde, em horas, para todas as empresas em Portugal do Grupo Simoldes, de 2017 a 2019

	IGM	IMA	MDA	MECA	SA	UL	GS
2017	2641	3347	6248	1062	3359	1640	3031
2018	2780	2740	5706	1465	2960	1787	2992
2019	3125	3133	6543	1566	3568	1627	3206
Média	2840	3067	6119	1330	3275	1682	3070

Como é perceptível o Grupo Simoldes é bastante diferenciado ao longo de todas as suas empresas, neste caso olhando apenas para as que têm sede em Portugal (Oliveira de Azeméis) percebe-se que, por exemplo a MDA que é especializada em moldes de grandes dimensões (cerca de 40 toneladas) principalmente para peças de plástico de exterior automóvel (como para-choques por exemplo) tem tempos médios de maquinação por molde na ordem das 6000 horas. Se estas 6000 horas fossem efetuadas ininterruptamente seriam precisos 250 dias completos para realizar toda a maquinação necessária a este tipo de moldes, ora como esta maquinação não é contínua há a necessidade de perceber quando a máquina está parada ou em curso, no caso de estar parada há também a necessidade de perceber se era previsível esta paragem, para no caso de não ser se concluir se é uma paragem justificada ou não. O caso da MDA é o mais exemplificativo desta questão por ser a empresa que produz moldes maiores e por isso ter tempos de maquinação superiores, tendo consequentemente também tempos de paragem de máquina superiores, mas o mesmo problema se passa em empresas que produzem moldes mais pequenos como é o caso da Mecamolde e da Ulmolde, há sempre a necessidade de maximizar a gestão de trabalho nos equipamentos de maquinação do grupo.

De seguida é apresentada uma tabela onde são apresentados os tempos de paragem médios anuais por máquina ao longo do Grupo Simoldes:

Tabela 2 - Horas de máquina parada, média anual entre os anos de 2017 e 2019

	IGM	IMA	MDA	MECA	SA	UL
Tempo paragem (anual)	832	889	3305	1052	992	1135

Em primeiro lugar é preciso analisar o que é mais preponderante para a variação dos tempos de paragem de equipamentos de maquinação, neste caso um molde maior pode envolver paragens de máquina superiores devido a toda a logística associada ao movimento e correta alocação de blocos de aço na ordem das 40 toneladas, como é de fácil compreensão e pode ser verificado pelos dados de paragem de máquina da MDA. Por outro lado, é também compreensível que empresas que produzem moldes para peças de plástico com tamanho mais reduzido (caso da Mecamolde e da Ulmolde) estejam associadas a tempos de paragem de máquina superiores visto que moldes mais pequenos terão, teoricamente, percursos de maquinação mais curtos e por isso menos tempo efetivo de maquinação quando comparando a moldes com tamanhos médios na realidade do Grupo Simoldes (como o caso de IGM, IMA e Simoldes Aços), ora se o tempo que o molde está no equipamento de maquinação é menor, maiores serão os números de movimentações de componentes dos moldes no chão de fábrica, levando a valores anuais de paragem de máquina superiores. É muito importante ainda, frisar que estes dados são recolhidos através de um sistema que envolve registos humanos, o que pode gerar alguma discrepância com a realidade apesar de não ser previsto que aconteça.

Chegando a este ponto de situação é compreensível a existência da necessidade de monitorização dos equipamentos de maquinação de modo a identificar com maior precisão os tempos produtivos e improdutivos ao longo de todo o processo produtivo, uma melhor

rastreabilidade de erros, entre outras questões e assim melhorar a eficiência praticada em cada equipamento de maquinação [14], [15].

3.2. Análise Técnica

O sistema de monitorização é denominado *Tracker*, o primeiro passo na sua conceção foi a de perceber de que forma realizar a recolha de informação das máquinas. Apesar de não haver uma reposta direta, tendo em conta que o desenvolvimento desta ferramenta de monitorização é interno ao Grupo Simoldes, a base do processo tenha de ser, pelo menos no início, construída de um ponto de partida mais simples para um mais amplo e complexo. Numa primeira fase de desenvolvimento não se englobou neste sistema de monitorização equipamentos convencionais (manuais), partindo-se de um pressuposto de monitorização de equipamentos CNC. Esta lógica assentar-se-á no facto de os comandos dos controlos numéricos registarem tudo o que realizam, havendo uma solução para a leitura destes registos há a possibilidade de monitorização do equipamento [12], [16].

Existem vários fabricantes de controlos numéricos computadorizados, o primeiro passo do sistema de monitorização atual do Grupo Simoldes foi dado nos comandos de origem italiana da Fidia. Após alguma investigação percebeu-se que, apesar de existirem versões deste tipo de controlos mais recentes que outras, todos eles registam todas as suas operações em *LogFiles* (um excerto de um destes ficheiros de texto apresentado na figura 14). Nestes comandos a informação está contida em dois ficheiros que serão mais importantes para a monitorização pretendida: LOGFILE.CNC e LOGFILE.OLD; ambos correspondem a registos do comando do controlo numérico relativamente à fresadora, a diferença reside ao intervalo de tempo destes registos, o ficheiro .CNC diz respeito ao registo da semana atual, enquanto que o ficheiro .OLD corresponde aos registos da semana anterior à que está agora a ser registada [17].

```
43 43-0x26750>60308 07:49:23 DPS_043 SELECTED THE M03 MODE
44 44-0x2bdb8>60309 07:49:24 DPS_018 RELEASED THE RELEASE KEY
45 45-0x3dc1a>60310 07:49:25 DPS_100 OVERRIDE FEED CHANGED FROM 36 % TO 0 %
46 46-0x2a48b>60311 07:55:36 DPS_015 PRESSED THE START KEY
47 47-0x41c10>60312 07:55:38 DPS_100 OVERRIDE FEED CHANGED FROM 0 % TO 116 %
48 48-0x43137>60313 07:55:42 DPS_100 OVERRIDE FEED CHANGED FROM 116 % TO 0 %
49 49-0x41f14>60314 08:05:40 IWS_154 NAO EXISTEM OPERACOES A INTERROMPER
50 50-0x3dd1f>60315 08:05:43 DPS_006 PRESSED THE Z AXIS SELECTION KEY
51 51-0x2b67c>60316 08:05:43 DPS_034 SELECTED THE Z AXIS
52 52-0x58ff6>60317 08:05:44 DPS_201 PRESSED JOG + KEY ON ZM AXIS START MXA = -304.630
53 53-0x3e736>60318 08:05:44 DPS_003 PRESSED THE RAPID MOVEMENT KEY
54 54-0x40fa1>60319 08:05:45 DPS_003 RELEASED THE RAPID MOVEMENT KEY
55 55-0x70af0>60320 08:05:45 DPS_201 RELEASED JOG + KEY ON ZM AXIS END MXA = -226.128 DIFF 78.502
56 56-0x4510b>60321 08:05:46 DPS_005 PRESSED THE Y AXIS SELECTION KEY
57 57-0x3062a>60322 08:05:46 DPS_033 SELECTED THE Y AXIS
58 58-0x3475e>60323 08:05:46 DPS_034 DESELECTED THE Z AXIS
59 59-0x64f14>60324 08:05:46 DPS_201 PRESSED JOG + KEY ON YM AXIS START MXA = -427.485
60 60-0x468bd>60325 08:05:46 DPS_003 PRESSED THE RAPID MOVEMENT KEY
61 61-0x55020>60326 08:05:47 DPS_100 OVERRIDE FEED CHANGED FROM 0 % TO 156 %
62 62-0x4a796>60327 08:05:49 DPS_003 RELEASED THE RAPID MOVEMENT KEY
63 63-0x7cf40>60328 08:05:49 DPS_201 RELEASED JOG + KEY ON YM AXIS END MXA = 0.452 DIFF 427.937
64 64-0x5cc97>60329 08:05:51 DPS_100 OVERRIDE FEED CHANGED FROM 156 % TO 172 %
```

Figura 14 - Excerto de um ficheiro *LOGFILE*, comando Fidia.

Outro fabricante de controlos numéricos contemplado neste sistema de monitorização é a Heidenhain, de origem alemã e bastante comum na indústria de produção de moldes para plástico. A grande maioria destes comandos divide-se em duas

versões iTNC530 e TNC640, no entanto ambas as versões registam informações sob a mesma lógica, em ficheiros do tipo `.log`. No caso da Heidenhain é possível manipular os intervalos de tempo a que o utilizador quer ter acesso, na figura 15 encontra-se um exemplo de um LOGFILE de um comando Heidenhain iTNC530 [18].

```

244 01: 3803560 30000 0x 0
245
246 Info: MAIN PLCPOS 10:42:52 Mon May 11 2020
247
248 OK: Positioning
249 AXISMASK: 0x0003
250 AX: POS: FEED: FLAG:
251 00: -9000000 30000 0x 0
252 01: 3803560 30000 0x 0
253
254 Info: REMO A_LG 10:42:55 Mon May 11 2020
255 Addr:0xAC1C1E23 Priv:0x03 No:2
256 Info: REMO A_LG 10:42:55 Mon May 11 2020
257 Addr:0xAC1C1E23 Priv:0x0B No:2
258 Info: REMO A_LG 10:42:55 Mon May 11 2020
259 Addr:0xAC1C1E23 Priv:0x0F No:2
260 Info: REMO A_LO 10:42:55 Mon May 11 2020
261 Addr:0xAC1C1E23 Priv:0x0B No:2
262 Info: REMO A_LO 10:42:55 Mon May 11 2020
263 Addr:0xAC1C1E23 Priv:0x01 No:2
264 Info: MAIN PLCPOS 10:42:56 Mon May 11 2020
265
266 OK: Positioning
267 AXISMASK: 0x0004
268 AX: POS: FEED: FLAG:
269 02: 7808000 30000 0x 0
270
271 Info: MAIN PLCPOS 10:43:00 Mon May 11 2020
272
273 OK: Positioning

```

Figura 15 - Excerto de um ficheiro `.log`, controlo iTNC 530 da Heidenhain.

Existem 130 fresadoras CNC do Grupo Simoldes conectados ao sistema de monitorização, todas com comando CNC de um dos dois fabricantes apresentados anteriormente. Existem outros comandos como Selca e Fagor que pelo seu menor peso na totalidade de equipamentos ainda não estão incluídos neste sistema de monitorização.

No âmbito geral, o sistema de monitorização recolhe informação do comando dos controlos numéricos das máquinas, através de várias instâncias em VisualBasic .NET, processa estas informações e realiza a sua introdução em bases de dados, através do *software* da Microsoft *SQL Server Management Studio*, a partir destas bases de dados outra instância de VB .NET lê os dados máquina a máquina e apresenta os mesmos na interface do sistema com os utilizadores (*Front End*). A linguagem VisualBasic .NET foi utilizada por ser bastante conhecida e difundida na comunidade informática, permite fácil interligação a bases de dados e a outras aplicações necessárias ao sistema, respondendo a todas as necessidades identificadas. O software de bases de dados *SQL Server Management Studio* foi selecionado para o sistema de monitorização por ser considerado um dos melhores no mercado, e por ser de um fornecedor com historial no Grupo Simoldes (Microsoft) [19]-[21].

3.2.1. Recolha de Informação - Equipamentos Fidia

O código do programa correspondente ao automatismo de recolha de dados de comandos CNC Fidia encontra-se em anexo (anexo A), devidamente comentado. O automatismo de recolha de informação dos equipamentos Fidia inicia-se com a recolha do ficheiro de

registo das atividades pretendido. Neste caso existem 2 ficheiros com estas informações: LOGFILE.CNC e LOGFILE.OLD; o primeiro é o ficheiro onde o comando ‘escreve’ as atividades do equipamento, ao final de uma semana de registo o ficheiro .CNC é renomeado para LOGFILE.OLD e começa um novo registo no ficheiro LOGFILE.CNC.

Nas variáveis do automatismo são declarados os códigos de todas as máquinas a serem analisadas, o primeiro bloco do código corresponde a um ciclo em que cada um destes equipamentos é analisado. É feita uma ligação à base de dados em que estão guardadas as principais informações da máquina, como por exemplo se deve ser lido o ficheiro .CNC ou .OLD e a partir de que linha deve ser feita esta leitura. Primeiro os últimos dados da máquina anterior são registados e apagados para darem lugar ao equipamento seguinte, depois é identificada a informação que deve ser processada e esta é convertida numa lista, em que cada linha da lista corresponde a uma linha do ficheiro (LOGFILE.CNC e/ou .OLD), que corresponde por sua vez a uma ação do controlo numérico.

De seguida o automatismo dá seguimento à lista criada anteriormente na rotina ‘Linhas’, esta rotina consiste num ciclo em que todas as linhas da lista são analisadas, no caso da linha conter uma ação de interesse a data/hora e os dados correspondentes são escritos na respetiva base de dados. Esta análise estende-se no máximo durante 5 minutos, este limite existe para garantir que o automatismo não fica bloqueado numa máquina, no caso de não ser suficiente, é guardado o número da última linha lida e na próxima passagem da máquina pelo automatismo a análise recomeça na linha em que terminou. As

ações de interesse do comando da máquina são identificadas por *strings* (conjunto de caracteres alfanuméricos), o automatismo identifica as seguintes:

- Data - Nos comandos Fidia o dia é identificado apenas às 00:00:01, por exemplo da seguinte forma “<< 07-Jun-2020 >>”, portanto se a linha contiver os caracteres “<<” e “>>” a data no seu interior é lida e registada na respetiva base de dados;
- Programa NC - Se a linha contiver as *strings* “IUI_009” ou “IUI_052” ao mesmo tempo que contém uma das 4 - “.FID”, “.fid”, “.tap” ou “.cnc” - significa que foi dada instrução ao equipamento para realizar um programa NC, neste caso o nome do programa é identificado e guardado na respetiva base de dados. Há o problema de muitos programas terem nomes não conformes o que retira alguma da utilidade desta análise;
- Início de trabalho - Se a linha contiver a *string* “IEX_438” significa que o equipamento está a iniciar um trabalho. Nesta situação o estado anterior é terminado: máquina parada, em trabalho (um início de trabalho anterior sem um fim declarado) ou máquina em setup; e o equipamento é considerado a partir deste instante como ligado (ou em trabalho);
- Início de trabalho manual - Se a linha contiver a *string* “DPS_043” e o informativo “SELECTED THE M03 MODE” significa que foi iniciado um trabalho manual. Nesta situação o estado anterior da máquina é terminado, esta é considerada como ligada e a hora de início de trabalho manual é registado na respetiva base de dados;
- Fim de trabalho manual - Se a linha contiver a *string* “DPS_043”, o informativo “DESELECTED THE M03 MODE” e ao mesmo tempo existe registo de início de

trabalho manual então o automatismo regista o final do trabalho manual e a duração do mesmo;

- Tempo de execução de tarefa - Se a linha contiver a *string* “IEX_452” significa que a tarefa que estava a ser realizada terminou, nesta situação o tempo de execução é registado na devida base de dados e dá-se o trabalho (manual ou automático) por terminado, passando o equipamento para o estado de parado;
- Pausas na execução da tarefa - No caso de avisos de emergência ou quando o botão de pausa é premido o tempo de execução da tarefa também para, ora para o sistema de monitorização o tempo de pausa deve também ser contabilizado na execução da tarefa. Portanto sempre que uma linha contém “IWS_999 REMOVED LINKSTATUS SESSION” (aviso de emergência) ou “DPS_016” (botão *STOP*) o momento é registado, para mais tarde ser contabilizado;
- Início de *Setup* - O equipamento é considerado em *setup* quando se está a proceder a uma troca manual de ferramenta ou a realizar outro trabalho manual necessário. O início de *setup* é considerado quando a linha contém a *string* “DPS_017” (botão *Hold*), sendo que nesta situação o estado anterior do equipamento é terminado e este passa a ser considerado em *Setup*;
- Fim de *Setup* - Se a linha contiver a *string* “DPS_018” significa que o botão *Hold* já não está premido e o equipamento já não está em *setup*. Neste caso é calculado e registado na devida base de dados o tempo de *setup*, a máquina passa para o estado de ligada;
- Início da execução de um bloco - Se a linha contiver a *string* “IWS_190” significa que está a ser definido o bloco (dimensões) da geometria a maquinar. Neste caso é terminado o estado anterior e o equipamento passa ao estado de ligada;
- Fim da definição do bloco - Se a linha a ser analisada contiver a *string* “IWS_191” significa que a definição do bloco terminou. Nesta situação é registado o tempo da ação e a máquina passa ao estado de parada;
- Início de movimento manual dos eixos da máquina - No caso da linha a analisar conter “DPS_201 PRESSED” ou “DPS_202 PRESSED” significa que um dos eixos da máquina está manualmente a deslocar-se. Nesta situação o estado anterior do equipamento é terminado e registado, passando agora para o estado de ligada;
- Fim de movimento manual dos eixos da máquina - No caso da linha a analisar conter “DPS_201 RELEASED” ou “DPS_202 RELEASED” significa que o movimento manual do eixo em questão terminou. Neste caso o automatismo calcula o tempo da tarefa realizada e passa o equipamento para o estado de parado;
- Procedimento Fidia - O procedimento Fidia está relacionado a transmissão de trabalho, dos postos CAM para a *string* da máquina, se houver a necessidade de executar vários programas NC em bloco (por exemplo ao fim de semana) recorre-se ao procedimento Fidia que permite esta ação. Se a linha contiver a *string* “IWS_130” significa que este procedimento foi iniciado, neste caso o estado anterior é registado e terminado e a máquina passa ao estado de ligada;

- Fim de procedimento Fidia - Se a linha contiver a *string* “IWS_132” significa que o procedimento Fidia terminou. Neste caso o automatismo regista o tempo da ação na respetiva base de dados e o equipamento passa ao estado de parado;
- Recolha de mensagens de erro e erros fatais - Se a linha em análise contiver o caracter “E” ou “F” numa posição específica significa que o equipamento está em erro (regular ou fatal, respetivamente). Neste caso, são registadas a empresa, a máquina, o tipo e código de erro, o descritivo do mesmo e a respetiva data.

Cada linha da lista criada a partir dos *LOGFILES* da máquina é sujeita a análise de todas estas condições, no final da análise de cada linha o número da mesma é registado na devida base de dados e o ciclo das linhas recomeça com a seguinte linha e até ao final da lista de linhas criada, ou dos 5 minutos de análise. Neste automatismo de recolha de informação de comandos de controlos numéricos Fidia o equipamento pode estar em 3 estados: Ligada (ou em corte), Parada e em *Setup*.

O código do automatismo fica completo com algumas sub rotinas, chamadas ao longo do ciclo de análise das linhas. Estas sub rotinas têm vários fins mas principalmente visam processar dados, analisar dados e registá-los nas respetivas bases de dados, as sub-rotinas presentes no automatismo são:

- Sub ‘Divisao_turnos’ - Esta rotina divide o intervalo de tempo em análise pelos 4 turnos de trabalho, 3 semanais e o outro de fim de semana;
- Sub ‘divisao_dias’ - Esta rotina está diretamente ligada à anterior, divide o intervalo de tempo em análise pelos respetivos dias, depois através da sub rotina ‘Divisao_turnos’ divide o tempo pelos turnos respetivos e introduz os dados na respetiva base de dados;
- Sub ‘Parada’ - No caso de a máquina estar parada esta rotina introduz nas respetivas bases de dados os registos de estado do equipamento;
- Sub ‘Setup_false’ - No caso de a máquina estar em *setup* esta rotina introduz nas respetivas bases de dados os registos de estado do equipamento;
- Sub ‘Ligada_false’ ou Sub ‘Corte_false’ - No caso de a máquina estar em trabalho estas rotinas introduzem nas respetivas bases de dados os registos de estado do equipamento;
- Sub ‘Tracker_programa_nc’ - Esta rotina tem o intuito de apagar dados de tabelas temporárias relativas aos dados de programas NC, para que estas possam ser preenchidas com dados novos;
- Sub ‘Tracker_escrever’ - Esta rotina realiza a ação pretendida de registo dos dados pretendidos no *software* de bases de dados SQL, é das sub rotinas mais utilizadas ao longo do automatismo;
- Sub ‘Tracker_var’ - Esta rotina tem o intuito de apagar dados de tabelas temporárias relativas aos dados da máquina a analisar no ciclo de análise de linhas, para que estas possam ser preenchidas com dados novos de outra máquina;
- Sub ‘Erro’ - Esta rotina é chamada quando na rotina ‘divisao_dias’ a diferença entre os tempos calculados é negativa. Esta rotina tem o propósito de registar na respetiva base de dados os dados deste erro de diferença temporal negativa;

- Sub ‘Corte_alt’ - Rotina chamada quando é premido o botão Stop ou quando há uma paragem de emergência no comando da máquina, tem o propósito de registar os dados do trabalho realizado até ao momento, passando a máquina para o estado de parada;
- Sub ‘Ligada_alt’ - Rotina chamada no final de operações como procedimento Fidia e movimentos manual de um eixo da máquina, tem o objetivo de registar os dados da atividade e passa o equipamento para o estado de parado.

O código do automatismo de recolha de dados de comandos controlos numéricos Fidia termina com estas rotinas, como é perceptível é complexo e há uma tentativa de recolha de uma grande quantidade de dados.

3.2.2. Recolha de Informação - Equipamentos Heidenhain iTNC530

O código do programa correspondente ao automatismo de recolha de dados de comandos iTNC 530 da Heidenhain encontra-se em anexo (anexo B), devidamente comentado. À semelhança do automatismo anterior, a recolha de informação inicia-se com a obtenção do ficheiro pretendido com estes registos, neste caso um ficheiro do tipo *.log* recolhido remotamente através do software TNCRemo, também da Heidenhain.

Nas variáveis do programa são declarados os códigos das máquinas a serem analisadas. À semelhança do automatismo anterior, a primeira etapa é apagar as informações da máquina anterior, fazendo de seguida ligação à base de dados correspondente à máquina respetiva acedendo às principais informações do equipamento. De seguida o automatismo realiza uma ligação ao TNCRemo e cria uma cópia do ficheiro *.log* desde a última data e hora de leitura (informações presentes na base de dados da máquina). Este ficheiro é convertido numa lista em que cada linha corresponde a uma ação do equipamento.

De seguida o automatismo dá seguimento à lista criada anteriormente na rotina ‘Linhas’, esta rotina consiste num ciclo em que todas as linhas da lista são analisadas, no caso da linha conter uma ação de interesse as horas e os dados correspondentes são registados na respetiva base de dados. No caso deste automatismo a análise estende-se no máximo durante 17 minutos para garantir que a análise não bloqueia, no caso de não ser suficiente na próxima passagem da máquina no automatismo a leitura recomeça no instante em que terminou. As ações/atividades identificadas pelo automatismo são:

- Data - Nestes comandos a data é identificada na mudança de dia, se a linha contiver a string “ Date:” é identificada a data do dia em questão, estes dados são registados na devida base de dados;
- Início de corte - Se a linha contiver a *string* “Stib: ON” significa que a máquina iniciou uma ação de corte. Nesta situação a máquina é considerada como parada até agora sendo estes dados registados na base de dados, o momento de início de corte é também registado e o equipamento passa a estar no estado de corte;
- Programa NC - Se uma linha contiver a *string* “TNC:” significa que foi dada instrução ao comando para realizar um programa NC, neste caso o nome do programa é recolhido e registado na respetiva base de dados. Se a linha contiver as *strings*

“TNC:” e “Info: MAIN PGM” e duas linhas antes foi registada a *string* “Stib: ON” então é feita a análise ao tipo de programa que é chamado, se fizer parte de um conjunto de programas considerados de *setup* é feito o registo dos dados de estado na respetiva base dados, no caso de ser um programa NC normal é também registado;

- Fim de corte - Se a linha em análise contiver a *string* “Stib: OFF” significa que a ação a decorrer terminou. Nesta situação o estado anterior da máquina é terminado (em corte, *setup* ou ligada) os dados são recolhidos e devidamente registados
- Máquina ligada - No caso da linha conter a *string* “Stib: BLINK” a máquina encontra-se ligada mas não em corte (em *standby*). Nesta situação a duração do estado anterior é calculado e registado;
- Troca automática de ferramenta - Existem várias *strings* que podem indicar a troca automática de ferramenta: “Info: Geo PLACEWRITE”, “ Tool” juntamente com “: place “ e “ written”, “Info: CTRL Kinematic”, “ tool nr “ e “TNC:\Laser\Medir_fresa.h”. No caso de alguma destas *strings* se verificar o *setup* é identificado, calculado e devidamente registado na respetiva base de dados;
- Troca manual de ferramenta - No caso da linha a analisar conter as *strings* ‘P39’, “Tool unclamping” ou “Desaperte a ferramenta” deu-se um início de troca manual de ferramenta, sendo o instante desta ação registado. No caso de a linha conter as *strings* ‘P40’, ‘Tool clamping’ ou “Aperte a ferramenta” deu-se o final de uma troca manual de ferramenta, nesta situação é calculado o tempo de *setup* sendo os dados registados na devida base de dados;
- Mensagem de erro - No caso de a linha conter a *string* “Error:” o equipamento encontra-se em erro. Nesta situação o automatismo recolhe a data e a hora do erro assim como o seu código e descritivo, os dados são colocados nas respetivas bases de dados;
- Verificação de sequência correta do ficheiro *.log* - Se a linha a analisar contiver uma das *strings* “Info:”, “Key:” ou “Error:”, é feita a recolha da hora desta linha e é feita a comparação com a linha anterior. No caso de a linha anterior ter um tempo posterior é feita uma cópia do ficheiro *.log* sendo gravado com o código da máquina em questão e o número da linha a analisar. Esta recolha é feita porque este tipo de fenómenos pode levar a erros na leitura de atividades do equipamento.

Todas as linhas da lista criada são sujeitas a estas condições, no caso do automatismo para comandos iTNC 530 da Heidenhain a máquina pode estar em 4 estados: em corte, parada, ligada (ou em *standby*) e em *setup*. O código do automatismo fica completo com algumas sub rotinas, chamadas ao longo do mesmo de forma a apoiar a recolha e processamento de informação:

- Sub ‘divisao_dias’ - Rotina com o objetivo de dividir os momentos iniciais e finais de uma certa ação em blocos que se encontram no mesmo dia. Após esta decomposição em dias é chamada a sub rotina ‘Divisao_turnos’. No final os dados são registados na devida base de dados;

- Sub 'Divisao_turnos' - Rotina com o objetivo de alocar a cada turno os respectivos tempos de estado do equipamento. É identificado o momento em que foi iniciada e em que foi finalizada uma certa ação, procedendo-se à divisão pelos respectivos turnos;
- Sub 'Compilacao_diaria' - Esta rotina ocorre quando na rotina 'divisao_dias' é identificada uma mudança de dia. Tem o objetivo de compilar os dados de um dia completo de modo a ser mais prático para tratamentos de dados futuros. Nesta análise apenas entram em conta as informações do dia, da máquina e dos tempos associados aos estados do equipamento;
- Sub 'Tracker_variavel' - Rotina usada para apagar dados de tabelas temporárias, de maneira a poderem ser preenchidas com dados de um outro equipamento. É utilizada quando a linha a analisar é referente a um erro;
- Sub 'Parada' - Rotina chamada quando a máquina inicia uma ação de corte, tem o propósito de calcular e registar o tempo que a máquina esteve parada;
- Sub 'Erro' - Esta rotina é chamada quando na rotina 'divisao_dias' a diferença entre os tempos calculados é negativa. Esta rotina tem o propósito de registar na respetiva base de dados os dados deste erro de diferença temporal negativa;
- Sub 'Tracker_escrever' - Esta rotina realiza a ação pretendida de registo dos dados pretendidos no *software* de bases de dados SQL, é das sub rotinas mais utilizadas ao longo do automatismo;
- Sub 'Tracker_var' - Rotina utilizada bastantes vezes ao longo de todo o código, tem o propósito de apagar dados de tabelas temporárias para poderem ser preenchidas com dados de outra máquina;
- Sub 'Tracker_programa_nc' - Esta rotina tem o intuito de apagar dados de tabelas temporárias relativas aos dados de programas NC, para que estas possam ser preenchidas com dados novos.

O automatismo de recolha de dados de controlos numéricos iTNC 530 da Heidenhain termina com estas sub rotinas, tal como o automatismo para comandos Fidia é complexo havendo uma grande recolha de dados.

3.2.3. Recolha de Informação - Equipamentos Heidenhain TNC640

O código do programa correspondente ao automatismo de recolha de dados de comandos CNC TNC640 da Heidenhain encontra-se em anexo, devidamente comentado. Tal como os automatismos anteriores, a recolha de informação inicia-se com a obtenção do ficheiro pretendido destes registos, neste caso um ficheiro do tipo *.log* recolhido através do software TNCRemo, também da Heidenhain. Estes equipamentos não podem ser incluídos no automatismo anterior porque, apesar de serem do mesmo fabricante as diferentes versões dos comandos envolvem diferentes metodologias no seu funcionamento, portanto a recolha de informação irá também ter de diferir.

Nas variáveis do programa são declarados os códigos das máquinas a serem analisadas. À semelhança dos automatismos anteriores, a primeira etapa é apagar as informações da máquina anterior, fazendo de seguida ligação à base de dados

correspondente à máquina respetiva acedendo às principais informações do equipamento. De seguida o automatismo realiza uma ligação ao TNCRemo e cria uma cópia do ficheiro *.log* desde a última data e hora de leitura (informações presentes na base de dados da máquina). Este ficheiro é convertido numa lista em que cada linha corresponde a uma ação do equipamento.

Mais uma vez em semelhança à lógica dos automatismos anteriores, de seguida é dado seguimento à lista criada anteriormente na rotina 'Linhas', esta rotina consiste num ciclo em que todas as linhas da lista são analisadas, no caso da linha conter uma ação de interesse as horas e os dados correspondentes são registados na respetiva base de dados. Neste automatismo a análise estende-se no máximo durante 60 minutos para garantir que a análise não bloqueia, no caso de não ser suficiente na próxima passagem da máquina no automatismo a leitura recomeça no instante em que terminou. As ações identificadas pelo automatismo são:

- Início de corte - Se a linha a analisar contiver a *string* "state:=NC_START" a máquina está a iniciar uma ação de corte. Nesta situação é calculado o tempo que a máquina esteve parada sendo devidamente registado, passando agora ao estado de corte;
- Fim de corte - Se a linha a analisar contiver uma das *strings* "state:=PROGRAM_STOP" ou "state:=PROGRAM_END" significa que uma ação de corte terminou, a primeira *string* é relativa à pausa de um programa a segunda ao fim de um programa. Nesta situação os dados de corte são calculados e registados, a máquina altera de estado para parada;
- Mensagens de erro - Se a linha contiver a *string* "Error: " significa que o equipamento está em erro, nesta situação as informações do erro são recolhidas, assim como a data/hora em que ocorreu sendo de seguida registados na devida base de dados.

O presente automatismo termina, tal como os anteriores, com algumas sub rotinas de apoio ao processamento, cálculo e registo dos dados recolhidos:

- Subs 'Divisao_turnos' e 'divisao_dias' - Tal como nos automatismos anteriores estas rotinas tem o propósito de dividir o intervalo de tempo de uma determinada ação, pelos dias e turnos devidos;
- Sub 'Tracker _escrever' - Esta rotina realiza a ação pretendida de registo dos dados pretendidos no *software* de bases de dados SQL, é das sub rotinas mais utilizadas ao longo do automatismo;
- Sub 'Tracker_var' - Rotina utilizada bastantes vezes ao longo de todo o código, tem o propósito de apagar dados de tabelas temporárias para poderem ser preenchidas com dados de outra máquina;
- Sub 'Tracker_variavel' - Rotina usada para apagar dados de tabelas temporárias, de maneira a poderem ser preenchidas com dados de um outro equipamento. É utilizada quando a linha a analisar é referente a um erro;
- Sub 'Erro' - Esta rotina é chamada quando na rotina 'divisao_dias' a diferença entre os tempos calculados é negativa. Esta rotina tem o propósito de registar na respetiva base de dados os dados deste erro de diferença temporal negativa.

É perceptível que este automatismo não é tão complexo nem recolhe tanta informação como os anteriores, por exemplo as máquinas só estão em um de dois estados: corte ou parada. Apesar da informação não ser tanta há a vantagem de não haver tanto erro associado às leituras, existe maior clareza nos dados recolhidos.

3.2.4. Automatismo de Cálculo de Percentagem Diária

O último automatismo não é de recolha de informações dos equipamentos, mas de processamento das mesmas, o código em VisualBasic .NET está em anexo (devidamente comentado).

Comparando com os automatismos anteriores este é mais simples, primeiro é criada uma lista com todas as máquinas integradas no sistema de monitorização. Depois máquina a máquina é feita a análise e cálculo da percentagem de trabalho diário até à última mudança de estado do equipamento, sendo estas informações devidamente registadas. O código fica completo com algumas sub rotinas de apoio ao automatismo, à semelhança dos anteriormente apresentados, estas rotinas tem o propósito de apagar dados de tabelas temporárias ou fazer a ligação à base de dados pretendida.

3.2.5. Sistema de Monitorização - Bases de Dados

Todos os automatismos descritos recolhem e geram uma grande quantidade de informação dos equipamentos CNC integrados no sistema de monitorização do Grupo Simoldes. Estes automatismos estão ativos 24 horas por dia e 365 dias por ano, a recolha de informação é constante e a sua quantidade está sempre a aumentar. Para armazenar toda a informação recolhida e interligar as várias instâncias do sistema de monitorização, é utilizado o *software SQL Server Management Studio* da Microsoft.

Existem várias bases de dados, de vários tipos e com diferentes propósitos, podem ser classificadas relativamente aos automatismos de recolha de informação, como apenas de registo ou de registo e leitura de dados.

- BD 'Var' - Única base de dados de registo e leitura de informações com 3 instâncias, uma para cada automatismo de recolha de informação. Esta tabela contém as principais informações de cada equipamento: o código da máquina, a descrição da mesma, a empresa em que se encontra, o diretório (ou IP) do ficheiro de registo, diretório da imagem da máquina, o número da linha atual, a hora de início de corte, setup, ligada ou parada, o estado da mesma, a última data de registo entre outros parâmetros necessários ao sistema de monitorização. Pode ser considerada a base de dados mais importante para a recolha de informação;
- BD 'Log_File' - Base de dados de registo de informações com 3 instâncias, uma para cada automatismo de recolha. Esta tabela contém o início e o fim de cada estado das máquinas em análise no automatismo;
- BD 'Tab_comp' - Base de dados de registo de informações comum aos 3 automatismos de recolha. Esta tabela contém as compilações diárias de tempos de cada máquina nos estados possíveis da mesma (corte, ligada, parada e *setup*), assim como estes dados de estado distribuídos pelos vários turnos;

- BD Máquina - Base de dados de registo de informação individual para cada máquina integrada no sistema de monitorização. Regista o início e o fim do estado específico da máquina e faz a sua distribuição pelos respetivos turnos;
- BD 'Mensagem_erro' - Base de dados de registo de informação comum aos 3 automatismos de recolha. É identificada a máquina, a hora e data, o tipo de erro e o descritivo do mesmo;
- BD 'Erro' - Base de dados de registo de informação comum aos 3 automatismos de recolha. Regista dados no caso de se verificarem leituras de linhas posteriores na ordem do ficheiro log mas anteriores no momento temporal, sendo registado a máquina, o momento do erro e o desfasamento temporal do erro;
- BD 'Programas_nc' e 'Programa_nc_heid' - Bases de dados de registo de informação relativa aos programas NC dos equipamentos monitorizados. É identificado o nome do programa NC, a máquina, o momento inicial e final assim como o tempo decorrido;
- BD 'Percentagem_diaria' - Base de dados de registo de informação proveniente do automatismo de cálculo de percentagem diária. É identificada a máquina e a empresa da mesma, assim como a percentagem diária de trabalho e o momento do último registo.

3.2.6. Sistema de Monitorização - *Front End*

O *Front End* corresponde à interface de todo o sistema de monitorização com o utilizador, é, portanto, a plataforma em que todo o trabalho anterior é demonstrado. A informação recolhida deve ser apresentada de forma clara e objetiva e o utilizador deve ter a possibilidade de poder escolher que dados quer consultar relativos a um certo equipamento ou empresa e durante um intervalo de tempo pretendido. O *Front End* está dividido em várias instâncias ou menus que serão de seguida apresentados:

- Menu Inicial - O menu inicial corresponde a um quadro resumo da percentagem de trabalho das várias empresas da *Tool Division*, correspondentes ao dia anterior e à semana corrente. Para tal, são recolhidos registos de atividades dos equipamentos da base de dados 'Tab_comp', com estas informações são calculadas as respetivas percentagens de trabalho apresentadas (anexo E);

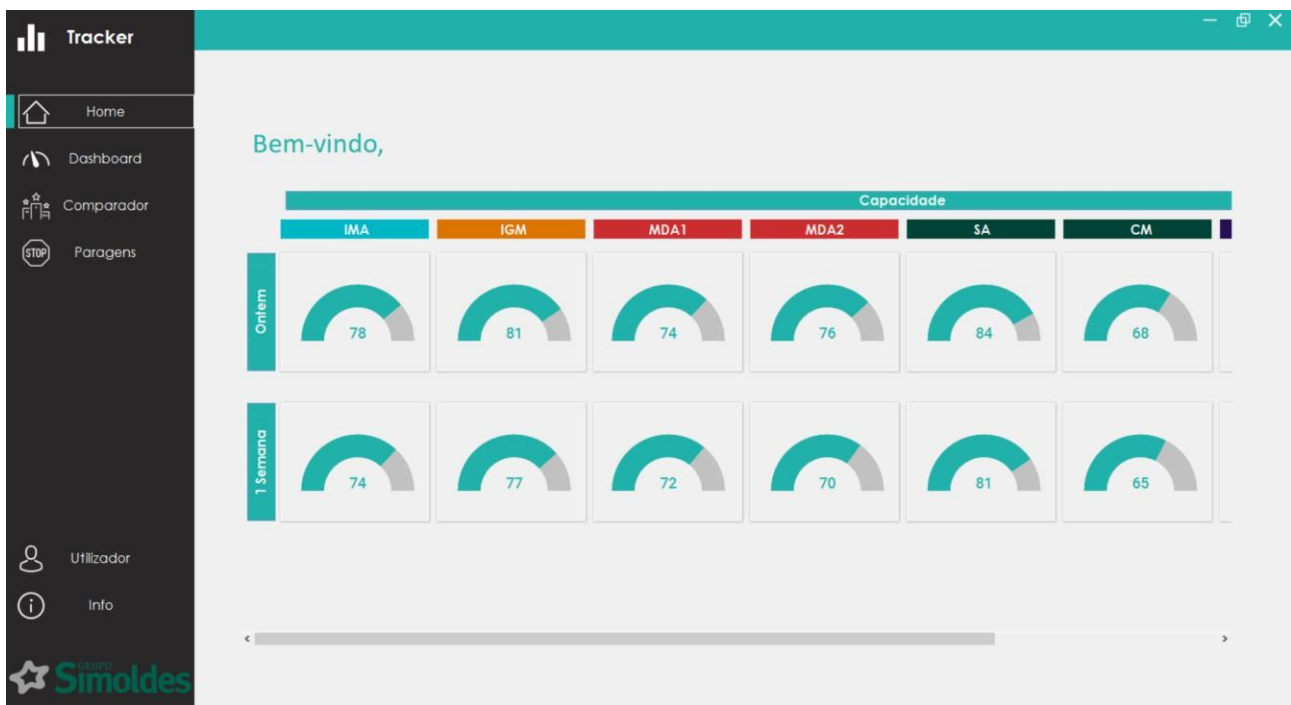


Figura 16 - Menu inicial da interface do sistema de monitorização

- *Dashboard* - Esta é a instância mais importante do *Front End*, está ela própria dividida em várias sub instâncias, em primeiro lugar o utilizador deve seleccionar a que empresa quer aceder:
 - *Dashboard* Empresa - Temos nesta sub instância (anexo F) todos os equipamentos da empresa integrados no sistema de monitorização. Cada equipamento tem uma imagem associada, o descritivo da máquina, o código interno da mesma, um semáforo identificativo do seu estado (em corte, ligada ou parada) e a percentagem diária de trabalho da mesma. Todos estes dados são retirados das respetivas bases de dados. Cada equipamento tem ainda um botão 'estado da máquina' que permite aceder a informações mais específicas dessa máquina. Os equipamentos podem ser apresentados em mosaico (por defeito) ou em lista através da vista diária, nesta vista os equipamentos são listados e além das informações da vista em mosaico são também apresentados os dados relativos à maquinação diária do equipamento;

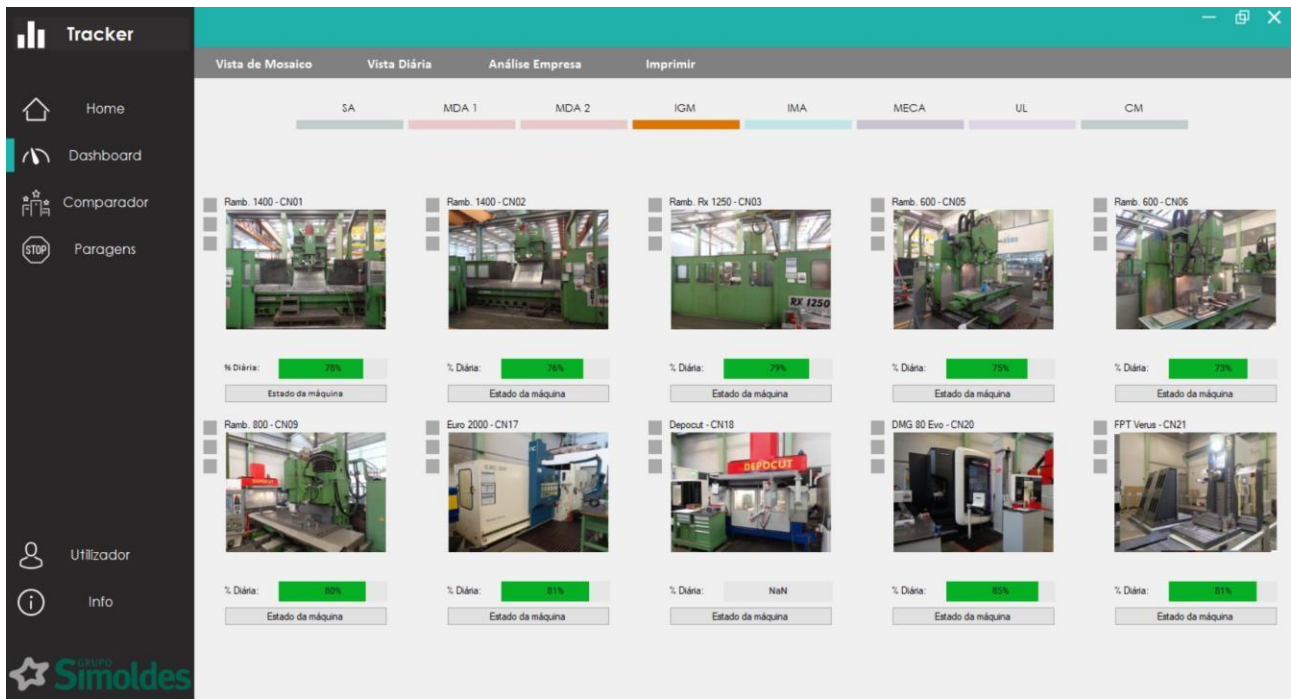


Figura 17 - Menu inicial da interface do sistema de monitorização

- *Dashboard* Máquina - Nesta janela são apresentados os dados de maquinação do equipamento em questão. É possível ao utilizador escolher o intervalo de tempo que quer analisar, se quer aceder a dados diários ou semanais entre outros pormenores, como a possibilidade de impressão (anexo G);

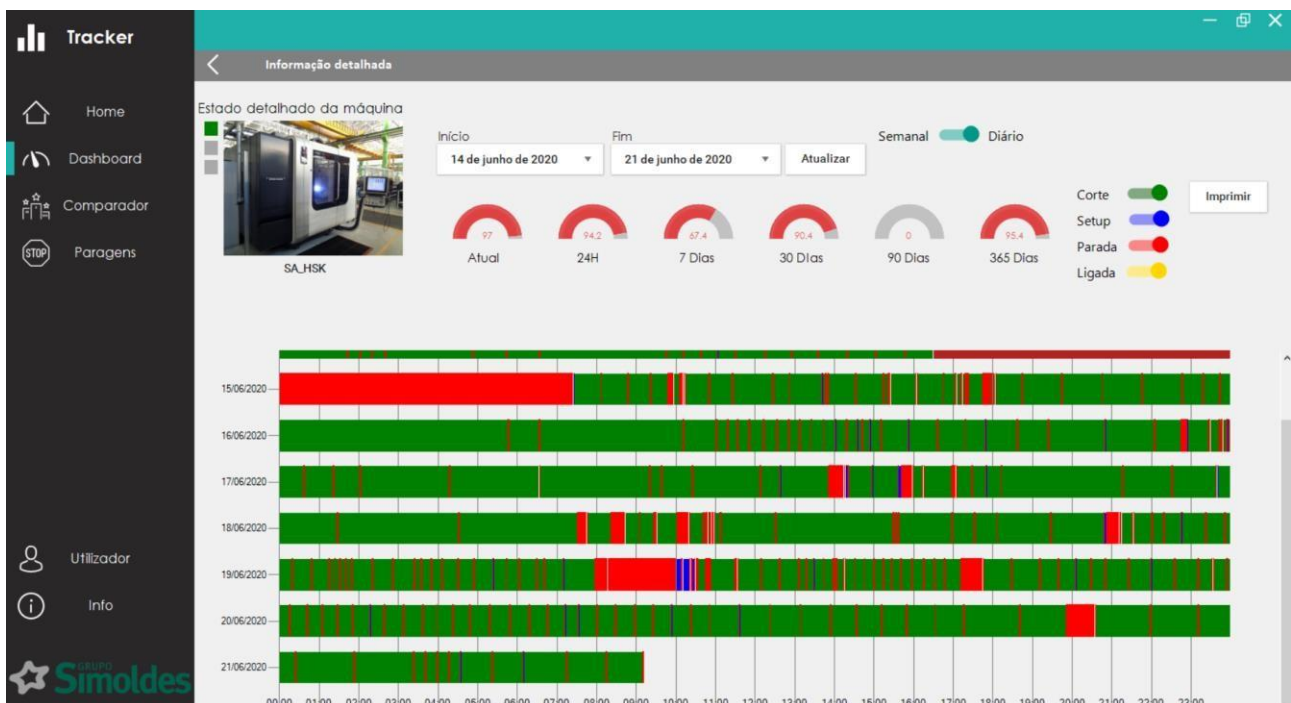


Figura 18 - *Dashboard* Máquina (diário)

- Análise Empresa - Nesta janela é apresentada a compilação dos dados de maquinação relativos às máquinas integradas no sistema de monitorização da empresa pretendida, podendo ser feita também uma análise semanal ou diária;

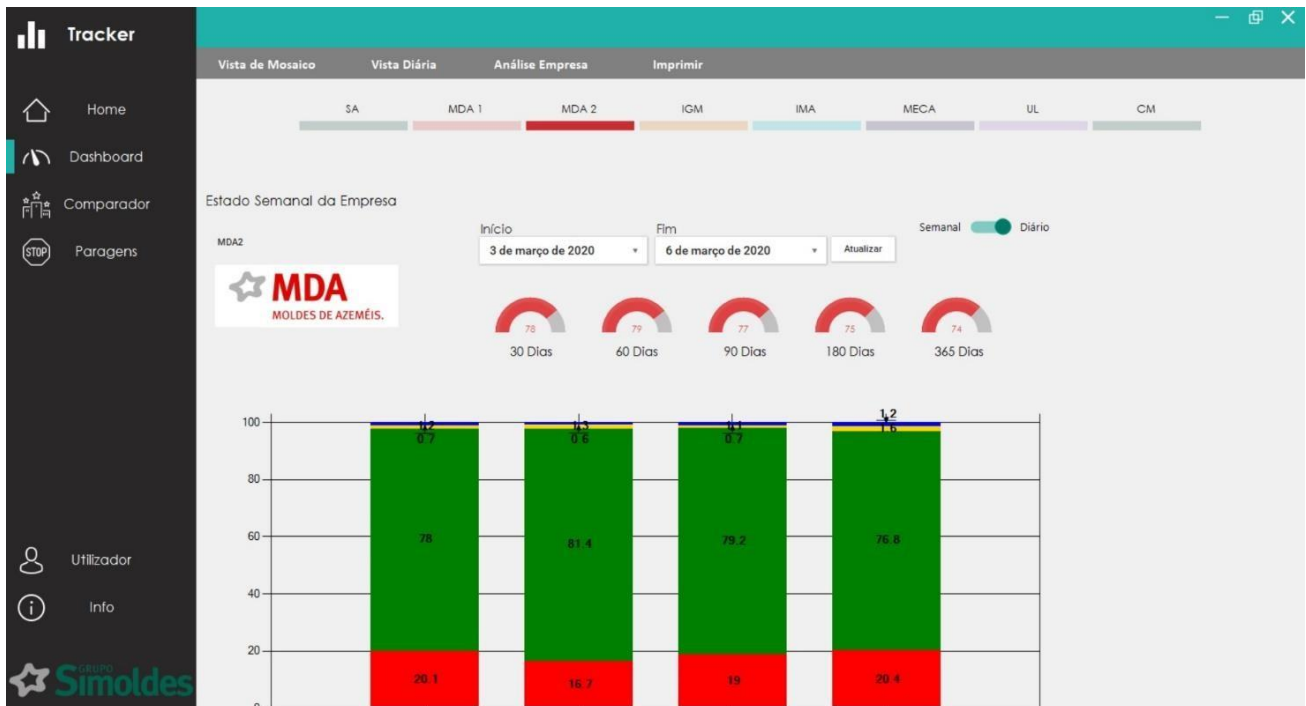


Figura 19 - Menu 'Análise Empresa'

- Comparador - Esta instância ainda não está desenvolvida, o objetivo seria haver a possibilidade de comparar equipamentos e/ou grupos de equipamentos;



Figura 20 - Instância de comparação (por desenvolver)

- Paragens - O propósito do menu 'Paragens' é justificar as paragens que o equipamento teve num certo dia. Para tal é necessário haver um sistema de *logins* em que cada utilizador pode ter acesso e efetuar estas justificações;

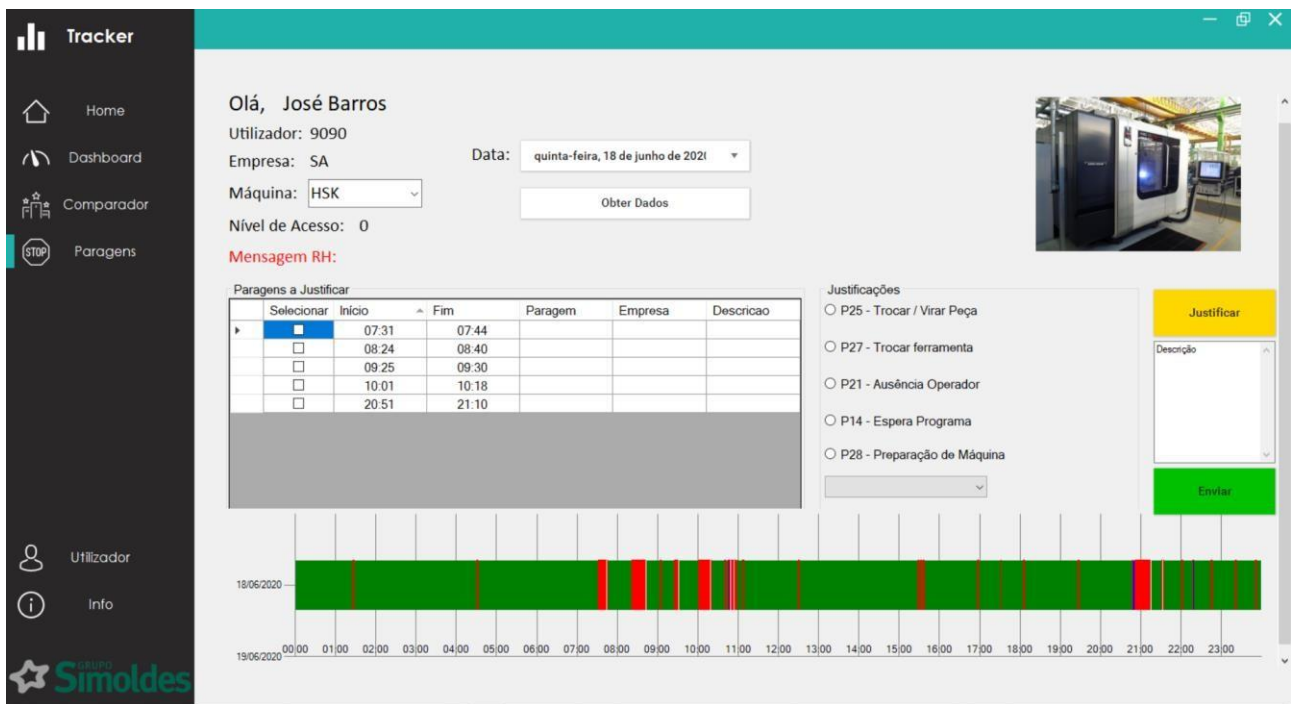


Figura 21 - Instância de justificação de paragens

- Utilizador - Este menu tem o objetivo de permitir aceder às informações do utilizador assim como a criação de um novo utilizador, no entanto ainda não está desenvolvido;



Figura 22 - Menu 'Utilizador' (por desenvolver)

- Menu Info - Neste menu estão apresentadas as principais informações do *Front End*, é meramente informativo.

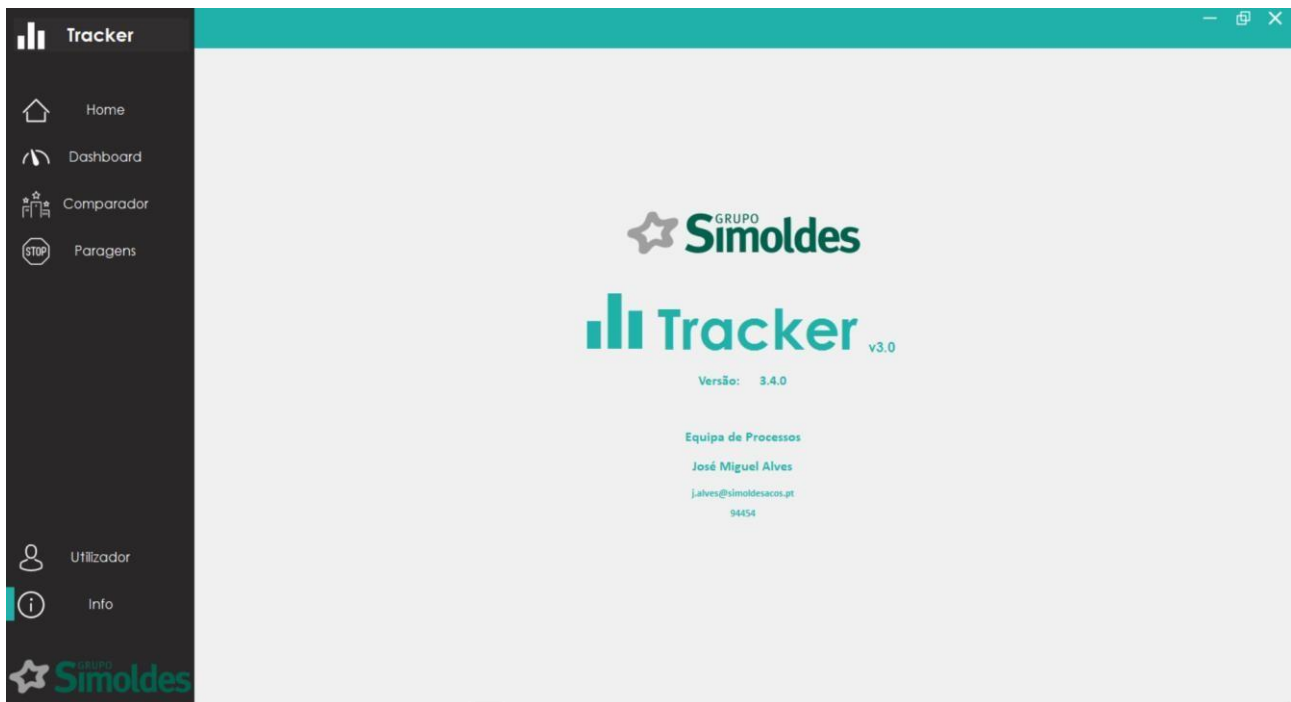


Figura 23 - Menu 'Info'

3.3. Análise SWOT do Sistema de Monitorização

A análise SWOT é uma ferramenta de análise de processos, métodos ou até de organizações/empresas, permite planificar o caminho a tomar e quais os maiores problemas que poderão comprometer este caminho. SWOT é uma sigla cujo significado é: *Strengths* (pontos fortes), *Weaknesses* (fraquezas), *Oportunities* (oportunidades de melhoria) e *Threats* (ameaças); com a análise destes parâmetros será possível perceber o estado atual do sistema de monitorização e qual o melhor caminho a tomar no seu desenvolvimento [22].

3.3.1. *Strengths* - Pontos Fortes

O sistema de monitorização do Grupo Simoldes foi desenvolvido internamente, no entanto tem capacidades muito importantes.

- O primeiro fator importante do sistema é a ligação e recolha de informação a mais de uma centena de comandos CNC, ao longo de 6 empresas em solo português. Uma necessidade muito importante para a realidade do Grupo Simoldes;
- A ligação a 3 tipos de comando CNC (de 2 fabricantes) é uma capacidade muito importante, o grupo tem vários tipos de comando CNC dentro de portas e quanto mais comandos o sistema suportar maior credibilidade terão os dados apresentados;
- A possibilidade de armazenamento de informação a longo prazo é outro ponto importante no sistema de monitorização atual, permitirá analisar ao longo do

tempo o trabalho realizado, com o objetivo da melhoria contínua das metodologias aplicadas no grupo;

- O *Front End* é também um ponto forte do sistema em funcionamento, tem um aspeto agradável e permite um rápido acesso a várias informações anteriormente recolhidas, relativas a cada empresa e a cada máquina integrada na monitorização.

Em suma o sistema de monitorização implementado tem já capacidades importantes para o seu funcionamento e sustentabilidade ao longo do tempo.

3.3.2. *Weaknesses* - Limitações

A análise às várias instâncias do sistema de monitorização já permite identificar algumas limitações ou necessidades às quais o sistema não responde. Para completar esta análise foi feito um levantamento junto das chefias das várias empresas, de modo a identificar as principais falhas ou limitações junto dos seus principais utilizadores.

A principal oportunidade de melhoria relacionada com o sistema de monitorização é o facto de as leituras efetuadas serem relativas ao que o equipamento CNC fez e não ao que está a fazer no presente momento. Os *LogFiles* correspondem ao registo de tudo o que a máquina está a fazer, muitas vezes existem até mais que uma linha de informação por segundo, o problema reside no facto de o sistema se ligar a mais de uma centena de máquinas impossibilitando a atualização a cada instante de cada equipamento da maneira que o automatismo está estruturado. Esta limitação torna o sistema numa ferramenta de recolha de dados e de monitorização do trabalho passado, algo que limita um pouco o sistema e a lógica inicial por detrás da implementação do mesmo. Além destas questões, o registo de atividades do comando da máquina é falível, muitos equipamentos são antigos e não foram concebidos com o intuito de criar um ficheiro de registo de atividades 100% fiável, é apenas uma tarefa interna ao seu sistema.

Foi reportado a falta de recolha de dados num tipo de máquina específico, presente em duas empresas do grupo, IGM e IMA. Este problema acontece num tipo de máquina específica que realiza trabalhos lineares (internamente apelidados de '2D'), neste grupo de máquinas as leituras são muito inconsistentes. Foi ainda reportado por chefias de MDA e Simoldes Aços que alguns controlos numéricos deixaram de ter o mesmo desempenho após implementação do sistema de monitorização (por exemplo a transferência de programas para o equipamento passaram a ser mais demoradas), e também que alguns equipamentos foram vendidos ou enviados para empresas do grupo no estrangeiro e que por isso deveriam ser retiradas do sistema.

O facto de haver apenas ligação a comandos Heidenhain e Fidia é uma limitação, o objetivo é haver ligação e recolha de dados a todos os equipamentos CNC assim como a possibilidade de ligação a equipamentos de maquinaria convencionais (manuais). Seria também importante a interligação a outros departamentos do Grupo Simoldes, como o planeamento e a orçamentação de moldes, havendo rastreabilidade de todos os componentes de cada molde será possível um planeamento mais eficiente e uma orçamentação mais precisa.

Outra necessidade importante reportada foi a de envio de alertas quando um equipamento tiver uma paragem de trabalho inesperada, por exemplo à noite ou ao fim

de semana em máquinas automáticas. Por outro lado, há também a necessidade de desenvolvimentos no *Front End*, apesar da aparência da interface ser agradável o conteúdo exposto é pouco, a instância de comparação de equipamentos é um exemplo de passo a dar no sistema atual. Além do mais, os automatismos de recolha de informação existentes podem precisar de algumas afinações ou desenvolvimentos, como a instância para equipamentos TNC640 no que toca às sub rotinas de recolha de erros do comando ou programas NC realizados.

Para terminar, existem ainda duas melhorias identificadas como necessárias ao processo, o sistema de monitorização deveria ter a possibilidade de ligação, recolha e processamento de dados de equipamentos de maquinaria de empresas no estrangeiro assim como a possibilidade de compatibilidade com dispositivos móveis (*smartphones* e *tablets*) e televisores. Ambas as melhorias não são possíveis por limitações de programação em linguagem VB .NET assim como o armazenamento de informação ser realizado em bases de dados *offline*. Este tipo de funcionalidades permitiria uma melhor divulgação das informações de maquinaria de cada equipamento, sendo importante para uma melhor eficiência interna, mas também muito benéfico para criar confiança em cliente, demonstrando o método e rigor no trabalho realizado.

3.3.3. *Oportunities* - Melhorias

A primeira oportunidade é a de manter em funcionamento todos os pontos fortes que o sistema já tem, apresentados anteriormente, o sistema já tem várias valências positivas. Em termos de oportunidades de melhoria identificam-se algumas principais como:

- A oportunidade de integração de diferentes tipos de comando CNC presentes no Grupo Simoldes é importante, quanto maior o número de equipamentos no sistema maior a credibilidade dos dados apresentados/recolhidos;
- A melhoria da fiabilidade e qualidade dos dados recolhidos é também um desenvolvimento importante para melhoria. A recolha e apresentação de informação relativa ao ‘presente’ dos equipamentos é uma premissa muito importante, para criar maior confiança e utilidade nas chefias das várias empresas do Grupo;
- Informações como o molde a que corresponde o componente que está a ser maquinado são muito importantes para vários departamentos do Grupo, para rastrear o trabalho realizado. Por outro lado, a apresentação das mensagens de erro dos comandos CNC podem ser informações também muito úteis para o departamento de manutenção;
- Por último, outra melhoria a implementar será a de retirar do sistema equipamentos dos quais já não seja possível a recolha de informação, por exemplo por já terem sido vendidos e substituídos. Por outro lado, do *Front End* também devem ser retiradas instâncias que não têm conteúdo de forma a criar uma interface mais ‘limpa’ com o utilizador.

3.3.4. *Threats* - Ameaças

Os riscos ou ameaças que o sistema de monitorização possui estão relacionados com as leituras realizadas e a sua veracidade, sem garantir confiança no sistema é impossível retirar conclusões importantes dos dados recolhidos. A veracidade dos dados pode obviamente estar relacionada com a programação dos automatismos desenvolvidos, mas pode também estar relacionado com problemas na própria máquina por deficiência de alguma funcionalidade. Um risco grave é o facto de se monitorizar o ficheiro de texto que o equipamento produz, não havendo monitorização direta dos movimentos mecânicos do equipamento. Muitos comandos CNC são da década de 1990, podendo obviamente responder às necessidades de maquinação, mas a sua monitorização pelo sistema implementado pode ser mais difícil. Por outro lado por ser um processo informático existe sempre o risco de haver ataques informáticos ao sistema.

4. Melhorias do Sistema Integrado de Monitorização

No último capítulo foram identificados vários problemas e limitações relacionados com o sistema de monitorização em utilização no Grupo Simoldes. A prioridade dos desenvolvimentos e alterações a realizar centrou-se em duas principais premissas: integrar o maior número possível de equipamentos CNC e melhorar ao máximo a veracidade e confiança nas leituras dos equipamentos no sistema. Por outro lado, questões como a interface do sistema com o utilizador e a análise dos dados recolhidos foram também abordadas. Possíveis soluções ou metodologias alternativas para realizar a monitorização pretendida foram também expostas.

4.1. Introdução Equipamentos CNC no Sistema

A primeira melhoria a ser trabalhada foi a de maximizar o número de equipamentos CNC no sistema de monitorização, para tal foram analisados comandos de controlos numéricos de diferentes fabricantes além dos já integrados: Selca, Fagor e versões mais antigas de equipamentos Heidenhain. Antes de qualquer alteração ou desenvolvimento no sistema o número de equipamentos presentes era de 130, dos quais 24 não apresentavam leituras (81,5% de equipamentos com leitura), todos estes equipamentos eram comandos de um de dois fabricantes: Fidia e Heidenhain.

Os desenvolvimentos de introdução de novos equipamentos no sistema de monitorização tiveram o foque de culminar em leituras simples e claras de modo a evitar possíveis erros ou fragilidades nestas introduções. Este foco visa criar confiança no trabalho realizado, portanto o principal objetivo durante estes desenvolvimentos é identificar se o equipamento está efetivamente a maquinar ou não, guardando e processando devidamente estes dados.

4.1.1. Equipamentos Selca

Os comandos Selca (de origem italiana) são equipamentos baseados no sistema operativo *Windows* da Microsoft, tal como os controlos numéricos Fidia. No Grupo Simoldes existem 8 máquinas equipadas com estes comandos, distribuídos por 3 empresas (Simoldes

Aços, MDA e IMA).

A primeira tarefa foi a de encontrar no comando o ficheiro de escrita do equipamento (equivalente ao *logfile* na Fidia). A investigação destes ficheiros foi feita ligando remotamente o PC de trabalho ao controlo numérico, concluiu-se que estes registos eram guardados numa pasta ao longo de vários ficheiros do tipo SLG (.slg). O mais recente destes ficheiros denomina-se de *LogEventi*, este ficheiro vai recebendo informações da atividade do equipamento até atingir um tamanho entre 63 e 66 KB (kilobyte), neste momento o ficheiro *LogEventi* é renomeado com a data e hora desta criação (última linha no ficheiro), por exemplo um ficheiro criado no dia 4 de Maio de 2020 às 14:03:13 é denominado de 20200504_14_03_13. Os equipamentos guardam registo de atividade do passado, podendo esta informação atingir meses, portanto o objetivo não é apenas ler a partir de agora o que equipamento está a fazer, mas também ler desde o registo mais antigo até ao momento atual de modo a permitir às chefias e ao departamento da qualidade a inclusão destes equipamentos já nos estudos, análises e ações do ano de 2020.

Com os ficheiros de registo do trabalho dos equipamentos Selca identificados, o passo seguinte passou por analisar a estrutura da informação registada, focando em recolher dados confiáveis de início e fim de arranque de material ao bloco ou componente em trabalho. Na figura 24 apresenta-se um excerto de um ficheiro de registo de atividade deste fabricante.

```
85 09/06/2020 14:02:08;SYSTEM;ESCAPE;Programmazione1.dll;
86 09/06/2020 14:02:09;SOFT_KEY_SYS;Entra Ambiente;Machine tool;
87 09/06/2020 14:02:10;SOFT_KEY_SYS;Premuto;BLOCO SIMPLES;
88 09/06/2020 14:02:10;SOFT_KEY_SYS;Rilasciato;BLOCO SIMPLES;
89 09/06/2020 14:02:10;STATO_CN;CAMBIO_STATO_PCPC;STATO_BLOCCO_SINGOLO;
90 09/06/2020 14:02:10;STATO_CN;CAMBIO_STATO_PCPC_OK;STATO_BLOCCO_SINGOLO;
91 09/06/2020 14:02:10;STATO_CN;CAMBIO_STATO_MCPC;STATO_BLOCCO_SINGOLO;
92 09/06/2020 14:02:10;STATO_CN;CAMBIO_STATO_MCPC_OK;STATO_BLOCCO_SINGOLO;
93 09/06/2020 14:02:26;ANALISI_BLOCCO;BLOCCO_SINGOLO;Blk src: X-663.55Y-296.9R --- Blk dest:
94 09/06/2020 14:02:26;STATO_CN;CAMBIO_STATO_MCPC;STATO_BLSING_IN_CORSO;
95 09/06/2020 14:02:26;STATO_CN;CAMBIO_STATO_MCPC_OK;STATO_BLSING_IN_CORSO;
96 09/06/2020 14:02:26;MSG_SYS;Scrive;Esecucao bloco simples em curso;
97 09/06/2020 14:02:40;STATO_CN;CAMBIO_STATO_MCPC;STATO_BLOCCO_SINGOLO;
98 09/06/2020 14:02:40;STATO_CN;CAMBIO_STATO_MCPC_OK;STATO_BLOCCO_SINGOLO;
99 09/06/2020 14:02:40;MSG_SYS;Scrive;Esecucao bloco simples terminada;
100 09/06/2020 14:02:41;SOFT_KEY_SYS;Premuto;MANUAL ;
101 09/06/2020 14:02:41;SOFT_KEY_SYS;Rilasciato;MANUAL ;
102 09/06/2020 14:02:41;STATO_CN;CAMBIO_STATO_PCPC;STATO_MANUALE;
103 09/06/2020 14:02:41;STATO_CN;CAMBIO_STATO_PCPC_OK;STATO_MANUALE;
104 09/06/2020 14:02:41;STATO_CN;CAMBIO_STATO_MCPC;STATO_MANUALE;
105 09/06/2020 14:02:41;STATO_CN;CAMBIO_STATO_MCPC_OK;STATO_MANUALE;
```

Figura 24 - Excerto de um ficheiro .slg, controlo iTNC 530 da Heidenhain.

É perceptível, à primeira vista, que a recolha da hora e data da ação do equipamento não será um problema visto que todas as linhas começam com estas informações. Por outro lado, é necessário perceber que *strings* (conjunto de caracteres) correspondem às ações de início e fim de corte, a lógica aplicada foi a de que estas instruções seriam repetidas sempre que a ação de corte começasse ou terminasse. Foram analisados vários ficheiros de registo onde, por via de tentativa e erro se concluiu que terão de existir dois automatismos distintos de recolha de informação, isto acontece porque 3 das máquinas realizam fresagem normal e as outras 5 realizam apenas trabalho de furação (para

sistemas de transferência de calor por exemplo), envolvendo *strings* diferentes para início e fim de maquinação.

Para os 3 equipamentos que realizam fresagem normal as *strings* que identificam o início de corte são: “MSG_SYS;Scrive;Execu” em simultâneo com “ao em curso” (o carácter “ç” induz o sistema em erro); as *strings* que indicam o final de uma ação de arranque de material são: “MSG_SYS;Scrive;Execu” em simultâneo com “ao terminata” ou “MSG_SYS;Scrive;Execu” em simultâneo com “ao interrompida” (o automatismo encontra-se em anexo). Relativamente às 5 máquinas com comando Selca que realizam furação, a *string* que indica o início de trabalho é “CAMBIO_STATO_MCPC_OK;STATO_EXEC_IN_CORSO”; a que indica fim de ação de corte é “CAMBIO_STATO_MCPC_OK;STATO_MANUALE”.

O próximo passo na integração de equipamentos Selca no sistema de monitorização passou por criar um automatismo à semelhança dos já existentes para Fidia e Heidenhain, que analise estes registos de cada máquina e os introduza em bases de dados SQL de modo a poderem ser processados pelo *Front End* do sistema. Os automatismos desenvolvidos encontram-se em anexo, devidamente comentados, foram desenvolvidos com base na lógica dos já existentes, as máquinas são analisadas uma de cada vez, o ficheiro de registo é convertido numa lista onde são procuradas as *strings* identificadas, quando uma linha contém uma das *strings* pretendidas (ou início de corte ou fim de corte) o automatismo escreve as informações recolhidas nas respetivas bases de dados. A particularidade do automatismo para os comandos Selca tem a ver com o ciclo aplicado para leitura dos registos, cada data/hora de registo (nome do ficheiro) é convertido num número, por exemplo o ficheiro 20200504_14_03_13.slg é convertido no número 20200504140313, este número é guardado numa base de dados. Ambos os automatismos analisam máquina a máquina todos os ficheiros presentes no diretório onde estes registos são guardados no equipamento, se a conversão da data/hora de algum dos ficheiros corresponder a um número superior ao guardado na respetiva base de dados significa que esse ficheiro é mais recente que o último a ser processado devendo por isso ser analisado e os seus dados recolhidos. Para o funcionamento deste automatismo de recolha de informação tiveram ainda ser criadas bases de dados para registo de informações e alterações ao *FrontEnd* para inclusão destes equipamentos.

No final deste desenvolvimento foram adicionados 8 novos equipamentos ao sistema de monitorização de equipamentos CNC do Grupo Simoldes, com leituras simples, mas fiáveis relativamente ao estado da máquina (em trabalho ou não), podendo assim contribuir para um melhor e maior controlo e análise da produção.

4.1.2. Outros Controlos Numéricos

Existem ainda outros comandos que não estão presentes no sistema de monitorização: Fagor, Mitsubishi e equipamentos Heidenhain de versões anteriores às já presentes no sistema.

Dos equipamentos em falta no sistema os controlos numéricos Heidenhain mais antigos (versões TNC 426 e anteriores) são os que apresentam maior número, 20 equipamentos. Apesar de alguns serem mais antigos que outros todos apresentam o mesmo

motivo para não estarem presentes no sistema de monitorização, não é possível a conexão remota ao comando da máquina para acesso aos *logfile*s como se fazem em versões mais recentes de comandos Heidenhain, através de TNCRemo. Foram recolhidos IP's de alguns destes equipamentos na tentativa de ligação remota aos ficheiros de registo de atividade, no entanto nenhuma das tentativas foi bem sucedida. Sem possibilidade de mais desenvolvimentos nestes equipamentos foi necessário contactar a distribuidora e representante da Heidenhain na península Ibérica, questionando a possibilidade de ligação destes equipamentos ao software TNCRemo. Para esta ligação ser possível seria necessário atualizar os comandos e equipá-los com uma placa de rede (no caso de ainda não possuírem) [23].

Existem controlos numéricos de outros fabricantes, como o comando Fagor presente em 4 máquinas do Grupo Simoldes. A primeira tarefa é a mesma que nos equipamentos anteriormente abordados, perceber como fazer uma ligação remota ao comando de modo a encontrar e analisar o ficheiro de registo de atividade. A ligação ao comando Fagor foi efetuada, à semelhança de controlos numéricos Fidia e Selca, com conexão remota ao equipamento. O ficheiro de registo das atividades do equipamento Fagor foi identificado na análise aos comandos, no entanto o seu conteúdo encontra-se codificado. Nesta fase de desenvolvimento, não foi possível a integração de equipamentos Fagor no sistema de monitorização não foi possível. Existe ainda uma máquina com comando Mitsubishi, neste caso não foi estudada a hipótese de integração porque se considerou não proveitoso o suficiente o investimento de tempo necessário a este desenvolvimento.

Equipamentos manuais foram também afastados da prioridade do trabalho realizado, a metodologia para a sua integração no sistema de monitorização teria de ser totalmente diferente dos restantes e a tendência é para a extinção ao longo do tempo destes equipamentos convencionais.

4.2. Análise e Melhoria da Qualidade dos Dados Recolhidos

Como referido anteriormente, antes de qualquer alteração ou desenvolvimento o sistema o *Front End* do Tracker tinha 130 máquinas, das quais 24 não apresentavam leituras, equivalendo a 81,5% de equipamentos com leituras. Após integração de novos equipamentos, a percentagem de equipamentos com leituras aumentou, no entanto é ainda necessário perceber porque existem equipamentos sem leituras realizadas, para se necessário excluir máquinas do sistema ou desenvolver e aplicar metodologias de recolha de informação mais eficazes para o pretendido.

4.2.1. Exclusão de Equipamentos do Sistema de Monitorização

No levantamento inicial realizado foi reportado por várias chefias que existiam equipamentos que já não estavam na empresa ou que estavam desconectados da rede *intranet* da Simoldes, mas que continuavam a ser apresentados no *Front End* do sistema. Estes equipamentos foram identificados, ao todo 14 divididos por 4 empresas do grupo, de seguida foram retirados da interface com o utilizador.

Pode aparentemente ser uma medida simples e pouco importante, mas é preciso ter em conta que o *Front End* é a única instância a que o utilizador tem acesso e conhece de todo o sistema de monitorização. É, portanto, muito importante criar coerência nos dados apresentados para criar o máximo de confiança possível no sistema.

4.2.2. Melhoria na Recolha de Dados

Foi também reportado que uma tipologia de máquina não apresentava leituras no *Front End*, existem duas Depocut em duas empresas do grupo, IGM e IMA, são centros de maquinação verticais de alto desempenho produzidas pela empresa DEPO. Estas máquinas estão equipadas com comandos CNC iTNC530 da Heidenhain, estão incluídas no automatismo de recolha de informações destes comandos, no entanto não está a ser efetuada a recolha e tratamento de dados, sendo necessária uma solução [24].



Figura 25 - Uma das máquinas com problemas de leitura de registos de atividade

Após análise da situação foi tomada a decisão de criar um automatismo para estas duas máquinas com base nos já existentes para iTNC530, com a diferença a residir no facto de apenas se fazer recolha de informações de início e fim ou pausa de corte (este automatismo não se encontra em anexo devido ao limite imposto). Não é retirada informação relativa a programas NC, mensagens de erro ou relativos a outras atividades do comando CNC, no entanto agora apresentam leituras, fiáveis, criando maior credibilidade ao sistema de monitorização implementado.

Para as máquinas que estavam sem leituras, sem razão aparente, as informações guardadas nas suas bases de dados foram apagadas. Foram introduzidas no sistema de novo e os seus dados foram registados novamente. A explicação encontrada foi a de que em algum momento da passagem da máquina no automatismo tenha ocorrido algum problema fora da norma, que não tenha gerado nenhum erro no programa, e que a partir daí tenha impossibilitado leituras.

Após as alterações apresentadas nos últimos subcapítulos terem sido executadas, quase todos os equipamentos apresentados no *Front End* passaram a ter leituras recolhidas e registadas diariamente, sendo de um rácio de 98,4%. Apenas duas máquinas estão apresentadas na interface com o utilizador sem leituras realizadas diariamente: uma tem um problema de software em que a data e hora do comando está constantemente com valores incorretos, sendo que foi contactada a empresa fabricante para resolver o problema; outra tem problemas relacionados com o acesso ao ficheiro de registo *.log* da mesma não havendo a possibilidade de acesso a este ficheiro não há monitorização, o fabricante terá também de ser contactado. Os dados recolhidos foram melhorados, é importante para a credibilidade do sistema, mas também influenciam de forma positiva a veracidade dos indicadores gerais de cada fábrica.

Tabela 3 - Quadro comparativo dos equipamentos presentes no sistema e os dados recolhidos

	NºMáqs (Antes)	%Leituras (Antes)
TOTAL	124 (130)	98,4% (81,5%)

4.3. Melhorias na Interface com o Utilizador

O *Front End* ou interface com o utilizador tinha já várias qualidades importantes antes de novos desenvolvimentos, o aspeto visual da aplicação é um ponto positivo, a navegação pelas máquinas das várias empresas é também intuitiva, foi por isso a instância de todo o sistema que menos desenvolvimentos sofreu. A maior limitação identificada na interface com o utilizador reside no facto de ser apresentada apenas informação relativa a estados da máquina como em corte, parada ou em *setup* (e em alguns casos ainda existe o estado de ligada), este estado só é registado aquando da mudança de estado, portanto, o equipamento pode estar por exemplo a maquinar horas seguidas e só no final da operação é apresentada a informação, isto é uma limitação. Os focos dos novos desenvolvimentos passaram pela apresentação de mais informação diretamente relacionada com o comando CNC, como o programa NC e/ou alguma mensagem de erro que possa ter ocorrido.

4.3.1. Programas NC e Mensagens de Erro

Há a necessidade de apresentar mais informações, fiáveis, que permitam perceber ainda melhor o estado do equipamento de maquinação, tanto relativamente ao trabalho que está a realizar, mas também relativamente a algum erro ou problema que possa ter ocorrido.

As mensagens de erro já eram identificadas e guardadas na respetiva base de dados pelos automatismos de recolha de informação dos equipamentos, no entanto não eram utilizadas. As informações relativas aos programas NC realizados pelo comando da máquina já era recolhida nos automatismos para comandos Fidia e Heidenhain iTNC530, foi feito um desenvolvimento no automatismo para comandos Heidenhain TNC640 para passar também a haver recolha nestes comandos (desenvolvimento apresentado no anexo I).

Com dados corretamente recolhidos e catalogados na respetiva base de dados é possível desenvolver a instância na interface com o utilizador em que os mesmos serão apresentados. Os desenvolvimentos realizados no código do *Front End* encontram-se no anexo J, devidamente comentados. Primeiro é feita uma ligação às bases de dados de programas NC depois estes dados são convertidos numa tabela temporária pela aplicação e são apresentadas as informações relativas ao intervalo de tempo pretendido pelo utilizador, para as mensagens de erro é aplicada a mesma lógica.

Os dados são apresentados na janela de cada máquina através de um novo botão introduzido (figura 26). O feedback das produções foi de alto valor acrescentado e de grande utilidade, neste momento é possível saber que trabalho está a ser realizado num equipamento específico sendo de interesse variado para vários departamentos do grupo. Por outro lado, as mensagens de erro são também importantes por exemplo para o departamento de manutenção, nomeadamente para perceber o panorama de cada equipamento ao longo do tempo, e também para o *reporting* e diagnóstico de problemas/avarias nas máquinas seja mais transparente, mais rápido e eficaz.

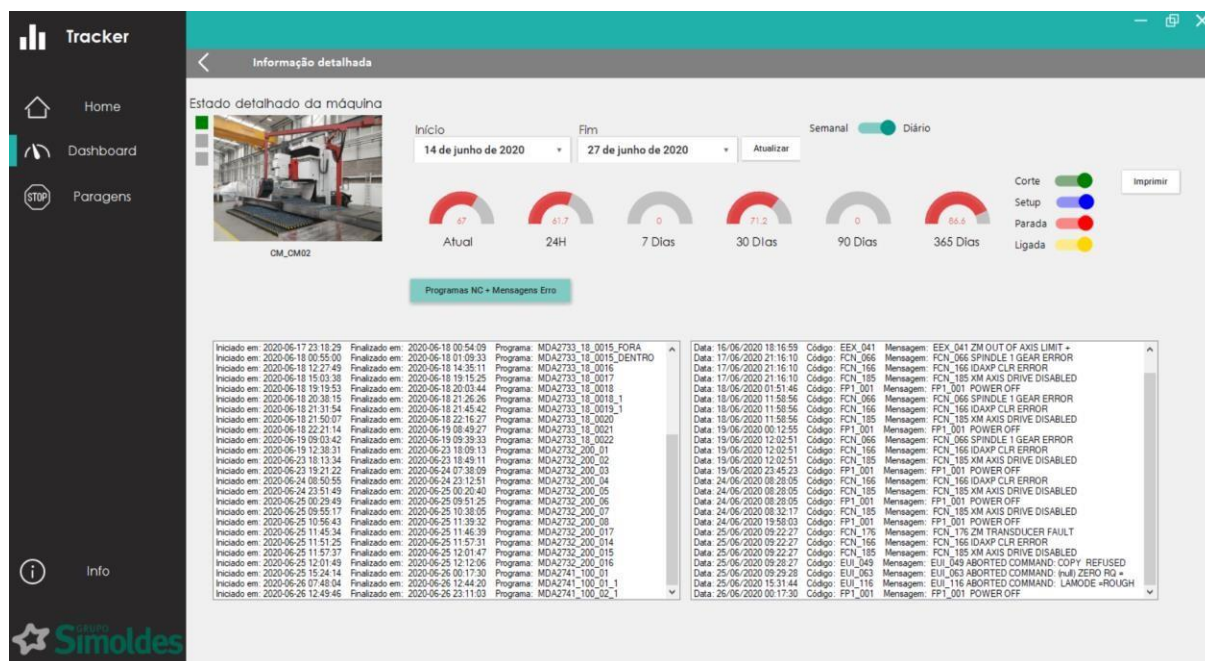


Figura 26 - Apresentação de informação relativa a programas NC e mensagens de erro

Os controlos numéricos com comando Selca não foram considerados porque a sua implementação é demasiado recente e a prioridade é que as suas leituras sejam simples e fiáveis, pelo que no futuro é um desenvolvimento a ter em conta e possível de realizar.

Há ainda necessidades de desenvolvimento na interface com o utilizador, como abordado anteriormente as instâncias de comparação de equipamentos assim como a funcionalidade de gestão de utilizadores/*logins* ainda está em falta. Estes desenvolvimentos não foram realizados porque não foram considerados tão prioritários como outros desenvolvimentos, foi dada mais importância a desenvolvimentos relativos à introdução de novos equipamentos e melhoria de leituras gerais das máquinas integradas no sistema. No que toca ao *Front End* a prioridade passou por apresentar mais informação relativa ao equipamento, algo cumprido com informações de Programas NC executados e mensagens de erro ocorridas. No futuro as instâncias de comparação e gestão de *logins* são desenvolvimentos a ter em conta.

4.4. Balanço Final e Futuro do Sistema de Monitorização

O trabalho realizado ao longo do semestre assemelha-se a um de consultoria, o sistema de monitorização implementado foi analisado e estudado, foram identificadas as suas principais qualidades que foram mantidas ou otimizadas, foram também identificados vários problemas/limitações, de seguida foram realizados vários desenvolvimentos que aumentaram a eficiência e credibilidade no sistema de monitorização da maquinaria do Grupo Simoldes. No entanto, continuam a existir limitações e necessidades que não são abrangidas pelo sistema atual:

- Em primeiro lugar seria importante ter registo do movimento efetivamente mecânico da máquina para registar o seu trabalho com 100% de fiabilidade e em '*Real Time*'. Algo que com o sistema atual é impossível de executar;
- Outra necessidade importante seria a possibilidade de reprodução em diferentes tipos de dispositivos eletrónicos como *smartphones/tablets* e televisores. Para responder a estas necessidades seria necessária programação nouro tipo de linguagem;
- Há a necessidade de integrar equipamentos do estrangeiro no sistema de monitorização, o Grupo Simoldes é internacional e está presente em países como Brasil, Argentina e Alemanha. Para responder a esta limitação o mais importante é trabalhar com um sistema de bases de dados *online* com dados acessíveis ao *Front End* em qualquer ponto do mundo;
- Por parte das produções é levantada mais uma questão com grande importância, a capacidade de o sistema enviar alertas/notificações preferencialmente por via SMS no caso de haver uma paragem imprevista em certo equipamento. Esta limitação prende-se no facto de necessitar de leituras 100% fiáveis e em cada instante, além do facto de se necessitar de interligar o sistema atual de monitorização com um dispositivo eletrónico externo que realize o envio de alertas pretendido;
- Continua a existir a limitação de não monitorizar todos os equipamentos de maquinaria presentes no grupo, máquinas manuais e alguns equipamentos com comando CNC continuam excluídos do sistema de monitorização implementado;
- Por último, continua a existir a necessidade de interligar o sistema de monitorização com o planeamento da produção de moldes, criando a possibilidade

de delegar trabalhos a certas máquinas através do *Front End*, ao mesmo tempo que se tem acesso ao desenvolvimento de cada molde e seus componentes com informações importantes como: tempo previsto para conclusão do projeto e também o custo real da produção do mesmo, apoiando também a orçamentação de moldes, entre outros. Para implementar este tipo de funcionalidades seria necessário ter já um sistema de monitorização muito sólido e quase irrepreensível, algo ainda distante da realidade atual.

Estas limitações/necessidades são as principais premissas para desenvolvimentos ou reestruturações futuras desta ferramenta de monitorização, acaba por ser um caderno de encargos muito resumido por parte do Grupo Simoldes. Existem duas abordagens ao futuro desta ferramenta: a de seguir com uma solução externa (subcontratada) ou a de retomar o desenvolvimento interno que responda a estas necessidades; ambas devem tomar como base e pilar de suporte, este trabalho

5. Considerações Finais

Os desenvolvimentos realizados e aplicados ao sistema de monitorização CNC do Grupo Simoldes tiveram resultados positivos, em várias etapas do seu processo.

O sistema de monitorização passou a ter mais comandos CNC nos seus automatismos de recolha de informação, a fiabilidade dos dados recolhidos aumentou, máquinas sem leituras foram analisadas e os seus problemas foram resolvidos, necessitam de manutenção externa ou foram excluídas se justificado. A diversidade de máquinas presentes no sistema aumentou e o rácio de equipamentos com leitura também, para 98,4%. A interface com o utilizador e a diversidade das informações apresentadas também melhoraram, passa agora a haver uma instância no menu de cada máquina com as informações relativas aos programas NC realizados pelo comando e às mensagens de erro que possam ocorrer no equipamento.

O sistema foi melhorado e ainda não foi atingido todo o potencial da estrutura atual de monitorização, no entanto, já foram identificadas várias limitações à metodologia atual como a deficiência na monitorização de equipamentos de maquinação manuais e a incompatibilidade da interface com dispositivos móveis e televisores. Foram identificadas as principais necessidades de monitorização da maquinação do Grupo Simoldes, funcionando como um caderno de encargos geral para desenvolvimentos ou reestruturações futuras.

Em suma foram alcançados os objetivos traçados no início do semestre, foram adquiridas e melhoradas várias competências em diferentes áreas de interesse, que terão importância no futuro a curto e longo prazo.

Bibliografia

- [1] Cefamol, “Indústria Portuguesa de Moldes Relatório de 2020,” 2020.
- [2] P. Martins, *Desenho e Projeto de Moldes*. FCA, 2019.
- [3] A. Larguesa, “António da Silva Rodrigues: O humilde de Azeméis que ‘molda’ automóveis,” 2018.
- [4] “Simoldes,” 2020. [Online]. Available: <http://www.simoldes.com>. [Accessed: 30-Mar-2020].
- [5] R. Freitas, “Fabrico Internacional - Ep. 6 Simoldes,” RTP, 2020.
- [6] G. Mennig and K. Stoeckhert, “Mold-Making Handbook,” G. Mennig and K. B. T.-M.-M. H. (Third E. Stoeckhert, Eds. Hanser, 2013.
- [7] A. Agazzi, V. Sobotka, R. LeGoff, and Y. Jarny, “Optimal cooling design in injection moulding process - A new approach based on morphological surfaces,” *Appl. Therm. Eng.*, vol. 52, no. 1, pp. 170-178, 2013.
- [8] P. Jones, *Mould design guide*. 2015.
- [9] N. C. Fei, N. M. Mehat, and S. Kamaruddin, “Practical Applications of Taguchi Method for Optimization of Processing Parameters for Plastic Injection Moulding: A Retrospective Review,” *ISRN Ind. Eng.*, vol. 2013, p. 462174, 2013.
- [10] F. G. Galizia, W. ElMaraghy, H. ElMaraghy, M. Bortolini, and C. Mora, “The evolution of molds in manufacturing: from rigid to flexible,” *Procedia Manuf.*, vol. 33, pp. 319-326, 2019.
- [11] O. Ogorodnyk and K. Martinsen, “Monitoring and Control for Thermoplastics Injection Molding A Review,” *Procedia CIRP*, vol. 67, pp. 380-385, 2018.
- [12] C. Relvas, *Controlo Numérico Computorizado: Conceitos Fundamentais - 4ª Edição*. 2018.
- [13] “Sandvik Coromant.” [Online]. Available: <https://www.sandvik.coromant.com/en-gb/pages/default.aspx>. [Accessed: 27-May-2020].
- [14] D. Mourtzis, E. Vlachou, V. Zogopoulos, and X. Fotini, “Integrated Production and Maintenance Scheduling Through Machine Monitoring and Augmented Reality: An Industry 4.0 Approach BT - Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing,” 2017, pp. 354-362.
- [15] P. Zheng *et al.*, “Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives,” *Front. Mech. Eng.*, vol. 13, no. 2, pp. 137-150, 2018.
- [16] X. W. Xu and S. T. Newman, “Making CNC machine tools more open, interoperable and intelligent—a review of the technologies,” *Comput. Ind.*, vol. 57, no. 2, pp. 141-152, 2006.
- [17] “Fidia S.p.A.” [Online]. Available: <http://www.fidia.it/en/>. [Accessed: 01-May-2020].
- [18] “Heidenhain.” [Online]. Available: <https://www.heidenhain.com/>. [Accessed: 01-May-2020].
- [19] *Visual Basic .Net Notes for Professionals*. GoalKicker.com, 2018.
- [20] Microsoft, “Visual Basic documentation.” [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/visual-basic/>. [Accessed: 10-May-2020].
- [21] Microsoft, “SQL Server technical documentation.” [Online]. Available: <https://docs.microsoft.com/en-us/sql/sql-server>. [Accessed: 10-May-2020].
- [22] Mind Tools, “SWOT Analysis - How to Develop a Strategy For Success.” [Online]. Available: https://www.mindtools.com/pages/article/newTMC_05.htm. [Accessed: 20-May-2020].

- [23] Heidenhain, “FARRESA.” [Online]. Available: https://www.heidenhain.pt/pt_PT/quem-somos/farresa/. [Accessed: 11-Jun-2020].
- [24] “DEPO - GMCD.” [Online]. Available: <http://depo-gmcd.com/pt-pt/>. [Accessed: 11-Jun-2020].

Anexos

Anexo A - Automatismo de recolha de informação em comandos Fidia

```
Public Class Tracker_valores_V22
'declaracao das variaveis necessarias ao automatismo
'retiradas para poupar espaco
'variaveis da base de dados
dim Cmd As SqlCommand
dim SQL As String
dim command As New SqlCommand
dim estado
dim connectionString As String = ""
dim connection As SqlConnection = New SqlConnection(connectionString)
dim da As New SqlDataAdapter
dim dt As New DataSet
'maquinas integradas nesta instancia
dim input_maq() As String = {
"IMA_CN01", "IMA_CN02", "IMA_CN03", "IMA_CN04", "IMA_CN07", "IMA_CN10",
"IMA_CN11", "IMA_CN13", "IMA_CN14", "IMA_CN15", "IMA_CN16", "IMA_CN21",
"IMA_CN22",
"UL_CN07", "UL_CN14", "UL_CN15", "UL_CN16", "UL_CN17",
"CM_CM01", "CM_CM02", "CM_CM06", "CM_CM07", "CM_CM08", "CM_CM09", "CM_CM10",
"CM_CM11", "CM_CM12", "CM_CM13", "CM_CM15", "UL_CN13"}
dim list_maquina As List(Of String) = New List(Of String)(input_maq)
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
Try
'Iniciar o timer
Timer1.Start()
Visible = True
segundos_corte_new = 0
connection.Close()
Me.Hide()
Catch ex As Exception
MsgBox("Erro 10.10: " & ex.Message)
End Try
Try
'Cada maquina na lista de maquinas 頰ujeita a este ciclo
Reiniciar:
For Each maq_log In list_maquina
Thread.Sleep(5000)
Try
If IsNothing(List_logfile) = False Then
List_logfile.clear()
End If
Catch ex As Exception
MsgBox("Erro 10.20: " & ex.Message)
End Try
'Aceder as principais informacoes da maquina
SQL = "Select * from Var where Maquina = '" & maq_log & "'"
Tracker_var()
'guardar principais informacoes da maquina em variaveis do automatismo
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
Var_logfile_old = dt.Tables("Tracker_variaveis").Rows(0).Item("Logfile_old")
diretorio_logfile = dt.Tables("Tracker_variaveis").Rows(0).Item("diretorio_CNC")
diretorio_old = dt.Tables("Tracker_variaveis").Rows(0).Item("diretorio_OLd")
linha_atual = dt.Tables("Tracker_variaveis").Rows(0).Item("Linha_atual")
data = dt.Tables("Tracker_variaveis").Rows(0).Item("data")
data_final = dt.Tables("Tracker_variaveis").Rows(0).Item("data_final")
ult_data = dt.Tables("Tracker_variaveis").Rows(0).Item("Ult_data")
empresa = dt.Tables("Tracker_variaveis").Rows(0).Item("Empresa")
End If
```

```

ReadAgain: Try 'Se o ficheiro de registo de atividades do comando CNC existe
If File.Exists(diretorio_logfile) = True Then

'registro do que o equipamento tem ligacao
SQL = "update Var set Ligacao = 1 where Maquina = '" & maq_log & "'";
update Var_backup set Ligacao = 1 where Maquina = '" & maq_log & "'"
Tracker_escrever()
List_logfile = File.ReadLines(diretorio_logfile).ToList
Else
'Registrado que o equipamento não tem ligacao
SQL = "update Var Set Ligacao = 0 where Maquina = '" & maq_log & "'"; update
Var_backup Set Ligacao = 0 where Maquina = '" & maq_log & "'"
Tracker_escrever()
GoTo Nextline
End If
Catch ex As Exception
GoTo ReadAgain
End Try
linhaCount = List_logfile.Count
'Atualizar os dados no form
maq_label.Text = maq_log.ToString
Ultdata_Label.Text = Cdate(ult_data).ToString("yyyy/MM/dd HH:mm:ss")
LinhaTotal_Label.Text = linhaCount.ToString
' Ler primeiro o Logfile old e depois o atual
If Var_logfile_old = 1 Or linhaCount < linha_atual Then
' Criar uma list com as linhas que ainda nao foram lidas
Try
ReadAgain2:          List_logfile = File.ReadLines(diretorio_old).ToList
Catch ex As Exception
MsgBox("Erro 10.30: " & ex.Message)
GoTo ReadAgain2
End Try
linhaCount = List_logfile.Count
'Apagar da list o que ja foi lido antes
Try
' Caso tenha de trocar para o ficheiro OLd
If linhaCount >= linha_atual Then
List_logfile.RemoveRange(0, linha_atual)
linhaCount = List_logfile.Count
'Atualizar o numero de linhas do ficheiro
linha_logfile = 0
'Garantir que o ficheiro atual e lido no momento seguinte
Var_logfile_old = 0
linha_atual = 0
'fazer update na respetiva Bd
SQL = "update Var Set Logfile_old = '" & Var_logfile_old & "', Linha = '" &
linhaCount & "', Linha_atual = '" & linha_atual & "' where Maquina = '" &
maq_log & "'";update Var_backup Set Logfile_old = '" & Var_logfile_old & "',
Linha = '" & linhaCount & "', Linha_atual = '" & linha_atual & "' where Maquina
= '" & maq_log & "' "
Tracker_escrever()
Linhas()
End If
' Caso o Logfile tenha sido apagado
If linhaCount < linha_atual Then
'Fazer reset as variaveis e continuar a ler o ficheiro .CNC
linha_atual = 0
List_logfile = File.ReadLines(diretorio_logfile).ToList
linhaCount = List_logfile.Count
Var_logfile_old = 0

```

```

'Reportar que o log file foi apagado
SQL = "INSERT INTO Falha_LogFile([Maquina], [data]) values('" & maq_log & "', '"
& Cdate(ult_data).ToString("yyyy/MM/dd HH:mm:ss") & "');"
Tracker_escrever()
'Ler as linhas escritas no logfile
Linhas()
End If
Catch ex As Exception
MsgBox("Erro 10.40: " & ex.Message)
End Try
linha_atual = 0
'fazer update
SQL = "update Var set Logfile_old = '" & Var_logfile_old & "', Linha = '" &
linhaCount & "', Linha_atual = '" & linha_atual & "' where Maquina = '" &
maq_log & "'; update Var_backup set Logfile_old = '" & Var_logfile_old & "',
Linha = '" & linhaCount & "', Linha_atual = '" & linha_atual & "' where Maquina
= '" & maq_log & "' "
Tracker_escrever()
GoTo ReadAgain
GoTo Logfile_CNC
ElseIf Var_logfile_old = 0 Or linhaCount > linha_atual Then ' Ler o logfile
atual
Logfile_CNC:      If linha_atual >= 0 And linhaCount > linha_atual Then
'Cutar a list do logfile para apenas conter o que ainda nao foi lido
List_logfile.RemoveRange(0, linha_atual)
'fazer update
SQL = "update Var set Linha = '" & linhaCount & "' where Maquina = '" & maq_log
& "'; update Var_backup set Linha = '" & linhaCount & "' where Maquina = '" &
maq_log & "' "
Tracker_escrever()
Linhas()
End If
End If
Nextline:  Next
GoTo Reiniciar
Catch ex As Exception
MsgBox("Erro 10: " & ex.Message)
End Try
End Sub
'Rotina de analise de Logfiles
Sub Linhas()
tempo_inic = Now
Try
'Procurar a informacao importante em cada linha e escrever nas Bds
For Each linha In List_logfile
tempo_atual = Now
'Escrever dados para o form
LinhaAtual_Label.Text = linha_atual.ToString
TextoLinha_Label.Text = linha.ToString
If datediff(dateInterval.Minute, tempo_inic, tempo_atual) < 5 Then
'data
If linha.contains("<<") And linha.contains(">>") Then
Try
separar_horas = 1
posicao_inic_data = InStr(linha, "<<") + 3
posicao_fim_data = InStr(linha, ">>") - 1
data = Mid(linha, posicao_inic_data, posicao_fim_data - posicao_inic_data)
data_final = Convert.ToDateTime(data).date
'Fazer reset as variaveis das maquinas
segundos_corte_maq = 0
segundos_parada_maq = 0

```

```

segundos_setup_maq = 0
'Fazer Update
SQL = "update Var set data_final = '" & Cdate(data_final).ToString("yyyy/MM/dd")
& "' where Maquina = '" & maq_log & "'; update Var_backup set data_final = '" &
Cdate(data_final).ToString("yyyy/MM/dd") & "' where Maquina = '" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 20.10.010: " & ex.Message)
End Try
'Escrever para a tabela de compilacao diaria
'Ler a tabela de cada maquina
Try
SQL = "SELECT * FROM " & maq_log & " where (DATEPART(yy, Inicio) = '" &
Cdate(dateAdd(dateInterval.day, -1, data_final)).ToString("yyyy") & "') AND
(dATEPART(MM, Inicio)= '" & Cdate(dateAdd(dateInterval.day, -1,
data_final)).ToString("MM") & "') AND (dATEPART(dd, Inicio)= '" &
Cdate(dateAdd(dateInterval.day, -1, data_final)).ToString("dd") & "') "
Tracker_var()
comp_corte = 0
comp_parada = 0
comp_setup = 0
comp_ligada = 0
dim t1_corte = 0
dim t2_corte = 0
dim t3_corte = 0
dim tfds_corte = 0
dim t1_parada = 0
dim t2_parada = 0
dim t3_parada = 0
dim tfds_parada = 0
dim t1_setup = 0
dim t2_setup = 0
dim t3_setup = 0
dim tfds_setup = 0
dim t1_ligada = 0
dim t2_ligada = 0
dim t3_ligada = 0
dim tfds_ligada = 0
For i = 0 To dt.Tables("Tracker_Variaveis").Rows.Count - 1
If IsDBNull(dt.Tables("Tracker_Variaveis").Rows(i).Item("Turno_1")) = True Then
t1_add = 0
Else
t1_add = dt.Tables("Tracker_Variaveis").Rows(i).Item("Turno_1")
End If
If IsDBNull(dt.Tables("Tracker_Variaveis").Rows(i).Item("Turno_2")) = True Then
t2_add = 0
Else
t2_add = dt.Tables("Tracker_Variaveis").Rows(i).Item("Turno_2")
End If
If IsDBNull(dt.Tables("Tracker_Variaveis").Rows(i).Item("Turno_3")) = True Then
t3_add = 0
Else
t3_add = dt.Tables("Tracker_Variaveis").Rows(i).Item("Turno_3")
End If
If IsDBNull(dt.Tables("Tracker_Variaveis").Rows(i).Item("Turno_fds")) = True
Then
tfds_add = 0
Else
tfds_add = dt.Tables("Tracker_Variaveis").Rows(i).Item("Turno_fds")
End If
If dt.Tables("Tracker_Variaveis").Rows(i).Item("Estado") = "Corte" Then

```

```

comp_corte = comp_corte + dt.Tables("Tracker_Variaveis").Rows(i).Item("Tempo")
t1_corte = t1_corte + t1_add
t2_corte = t2_corte + t2_add
t3_corte = t3_corte + t3_add
tfds_corte = tfds_corte + tfds_add
ElseIf dt.Tables("Tracker_Variaveis").Rows(i).Item("Estado") = "Ligada" Then
comp_ligada = comp_ligada + dt.Tables("Tracker_Variaveis").Rows(i).Item("Tempo")
t1_ligada = t1_ligada + t1_add
t2_ligada = t2_ligada + t2_add
t3_ligada = t3_ligada + t3_add
tfds_ligada = tfds_ligada + tfds_add
ElseIf dt.Tables("Tracker_Variaveis").Rows(i).Item("Estado") = "Setup" Then
comp_setup = comp_setup + dt.Tables("Tracker_Variaveis").Rows(i).Item("Tempo")
t1_setup = t1_setup + t1_add
t2_setup = t2_setup + t2_add
t3_setup = t3_setup + t3_add
tfds_setup = tfds_setup + tfds_add
ElseIf dt.Tables("Tracker_Variaveis").Rows(i).Item("Estado") = "Parada" Then
comp_parada = comp_parada + dt.Tables("Tracker_Variaveis").Rows(i).Item("Tempo")
t1_parada = t1_parada + t1_add
t2_parada = t2_parada + t2_add
t3_parada = t3_parada + t3_add
tfds_parada = tfds_parada + tfds_add
End If
Next
If comp_corte = 0 And comp_setup = 0 And comp_ligada = 0 Then
comp_parada = 86400
End If
'Escrever na tabela de compilacao
SQL = "INSERT INTO Tab_comp([data], [Maquina], [Corte_segundo], [Setup_segundo],
[Ligada_segundo], [Parada_segundo], [t1_corte], [t2_corte], [t3_corte],
[tfds_corte], [t1_parada], [t2_parada], [t3_parada], [tfds_parada], [t1_setup],
[t2_setup], [t3_setup], [tfds_setup], [t1_ligada], [t2_ligada], [t3_ligada],
[tfds_ligada]) values('" & Cdate(dateAdd(dateInterval.day, -1,
data_final)).ToString("yyyy/MM/dd") & "', '" & maq_log & "', '" & comp_corte &
"', '" & comp_setup & "', '" & comp_ligada & "', '" & comp_parada & "', '" &
t1_corte & "', '" & t2_corte & "', '" & t3_corte & "', '" & tfds_corte & "', '"
& t1_parada & "', '" & t2_parada & "', '" & t3_parada & "', '" & tfds_parada &
"', '" & t1_setup & "', '" & t2_setup & "', '" & t3_setup & "', '" & tfds_setup
& "', '" & t1_ligada & "', '" & t2_ligada & "', '" & t3_ligada & "', '" &
tfds_ligada & "')"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 20.10.020: " & ex.Message)
End Try
End If
Try
'Ler a tabela Var
'Ler as variaveis todas da tabela de variaveis
SQL = "SELECT * from Var where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_Variaveis").Rows.Count <> 0 Then

inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
nome_programa_nc = dt.Tables("Tracker_Variaveis").Rows(0).Item("Programa_nc")
hora_m03 = dt.Tables("Tracker_Variaveis").Rows(0).Item("hora_M03")
End If
If IsDBNull(nome_programa_nc) = True Then

```

```

ElseIf nome_programa_nc.contains(" & vbTab & ") Then
Replace(nome_programa_nc, " " & vbTab & " ", " ")
End If
'Apagar o apóstrofo
If IsDBNull(nome_programa_nc) = True Then
ElseIf nome_programa_nc.contains("'") Then
Replace(nome_programa_nc, "'", "")
End If
If IsDBNull(ult_data) = True And IsDBNull(data_final) = True Then
ElseIf IsDBNull(ult_data) = True Then
ult_data = data_final
End If
Catch ex As Exception
MsgBox("Erro 20.10.030: " & ex.Message)
End Try
'Programa NC
If (linha.Contains("IUI_009") Or linha.Contains("IUI_052")) And
(linha.Contains(".FId") Or linha.Contains(".fid") Or linha.Contains(".tap") Or
linha.Contains(".cnc")) Then
Try
nome_programa_nc = Mid(linha, (InStr(linha, "CNC:") + 4), InStr((InStr(linha,
"CNC:") + 4), linha, " ")) - (InStr(linha, "CNC:") + 4) - 4)
If nome_programa_nc.contains(" & vbTab & ") Then
Replace(nome_programa_nc, " " & vbTab & " ", " ")
End If
If nome_programa_nc.contains("'") Then
Replace(nome_programa_nc, "'", "")
End If
'Hora do logfile
tempo_maq = Mid(linha, (InStr(linha, ">") + 7), (InStr(linha, "IUI") - 1) -
(InStr(linha, ">") + 7))
'Transformar as horas de string para formato de hora
tempo_maq = Convert.ToDateTime(tempo_maq).TimeOfDay
'Juntar as horas do logfile com a data
tempo_nome = data_final + tempo_maq
'FAZER UPDATE MAQUINA
SQL = "update Var set Programa_nc = '" & nome_programa_nc & "' where Maquina =
'" & maq_log & "'; update Var_backup set Programa_nc = '" & nome_programa_nc &
"' where Maquina = '" & maq_log & "'"
'Sub Rotina
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 20.20.010: " & ex.Message)
End Try
End If
'CORTE
If linha.Contains("IEX_438") Then
Try
'Hora do logfile
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "IEX") - 1
tempo_maq_start = Mid(linha, posicao_inic_hora, posicao_fim_hora -
posicao_inic_hora)
'Transformar as horas de string para formato de hora
tempo_maq_start = Convert.ToDateTime(tempo_maq_start).TimeOfDay
'Juntar as horas do logfile com a data
Tempo = data_final + tempo_maq_start
tempo_maq = Tempo
'Caso não se tenha iniciado nada log a maquina e declarada como parada
Parada()

```

```

'Caso exista um inicio de Setup a maquina e declarada como em setup ate este
momento
Setup_false()
'Caso exista um inicio de ligada a maquina r declarada como ligada ate este
momento
Ligada_false()
'Caso apareca outro start a maquina e declarada como ligada ate agora
Corte_false()
estado = "Ligada"
'fazer update
SQL = "update Var set Inicio_corte = '" & Cdate(Tempo).ToString("yyyy/MM/dd
HH:mm:ss") & "', Estado = '" & estado & "', Ult_data = '" &
Cdate(Tempo).ToString("yyyy/MM/dd HH:mm:ss") & "' where Maquina = '" & maq_log &
"'; update Var_backup set Inicio_corte = '" & Cdate(Tempo).ToString("yyyy/MM/dd
HH:mm:ss") & "', Estado = '" & estado & "', Ult_data = '" &
Cdate(Tempo).ToString("yyyy/MM/dd HH:mm:ss") & "' where Maquina = '" & maq_log &
"' "

Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 20.30.010: " & ex.Message)
End Try
End If

'Inicio de trabalho manual
If linha.contains("dPS_043") And linha.CONTAINS("SELECTED THE M03 MODE") Then
Try
'Ler as horas
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "dPS") - 1
tempo_maq_start = Mid(linha, posicao_inic_hora, posicao_fim_hora -
posicao_inic_hora)
'Transformar as horas de string para formato de hora
tempo_maq_start = Convert.ToDateTime(tempo_maq_start).TimeOfDay
'Juntar as horas do logfile com a data
Tempo = data_final + tempo_maq_start
tempo_maq = Tempo

'Caso nãose tenha iniciado nada log a maquina ẽ declarada como parada
Parada()
'Caso exista um inicio de Setup a maquina ẽ declarada como em setup atẽste
momento
Setup_false()
'Caso exista um inicio de ligada a maquina ẽ declarada como ligada atẽste
momento
Ligada_false()
'escrever as horas na tabela de variaveis
SQL = "UPdATE Var set hora_M03 = '" & Cdate(Tempo).ToString("yyyy-MM-dd
HH:mm:ss") & "' where Maquina = '" & maq_log & "'; UPdATE Var_backup set
hora_M03 = '" & Cdate(Tempo).ToString("yyyy-MM-dd HH:mm:ss") & "' where Maquina
= '" & maq_log & ""
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 20.40.010: " & ex.Message)
End Try
End If

'Fim de trabalho manual
If linha.contains("dPS_043") And linha.CONTAINS("dESELECTED THE M03 MODE") And
IsDBNull(hora_m03) = False Then
Try
Try
'Ler as horas
posicao_inic_hora = InStr(linha, ">") + 7

```



```

posicao_fim_hora = InStr(linha, "dPS") - 1
tempo_maq_start = Mid(linha, posicao_inic_hora, posicao_fim_hora -
posicao_inic_hora)
'Transformar as horas de string para formato de hora
tempo_maq_start = Convert.ToDateTime(tempo_maq_start).TimeOfDay
'Juntar as horas do logfile com a data
Tempo = data_final + tempo_maq_start
tempo_maq = Tempo
tempo_corte = datediff(dateInterval.Second, Cdate(hora_m03), Cdate(Tempo))
Catch ex As Exception
MsgBox("Erro 20.40.020: " & ex.Message)
End Try
If IsDBNull(inicio_corte) = True Then
Try
estado = "Corte"
'Escrever as horas de corte no logfile
SQL = "INSERT INTO Log_File([Programa_nc], [Maquina], [data_entrada],
[data_saida], [Tempo_corte], [Estado]) values('Trabalho_Manual', '" & maq_log &
"', '" & Cdate(hora_m03).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(Tempo).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & tempo_corte & "', '" &
estado & "' );"
Tracker_escrever()
'Escrever as horas de corte na tabela da maquina
inicio = hora_m03
fim = Tempo
divisao_dias()
'Escrever na tabela do programa NC
'Atualizar a tabela de variaveis
SQL = "update Var set Ult_data= '" & Cdate(Tempo).ToString("yyyy/MM/dd
HH:mm:ss") & "', Inicio_corte = NULL, Inicio_setup = NULL, Programa_nc = NULL,
Inicio_ligada = NULL, hora_M03 = NULL, Estado = '" & estado & "' where Maquina =
'" & maq_log & "'; update Var_backup set Ult_data= '" &
Cdate(Tempo).ToString("yyyy/MM/dd HH:mm:ss") & "', Inicio_corte = NULL,
Inicio_setup = NULL, Programa_nc = NULL, Inicio_ligada = NULL, hora_M03 = NULL,
Estado = '" & estado & "' where Maquina = '" & maq_log & "' "
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 20.40.030: " & ex.Message)
End Try
Else
Try
'Atualizar a tabela de variaveis
SQL = "update Var set Ult_data= '" & Cdate(Tempo).ToString("yyyy/MM/dd
HH:mm:ss") & "', hora_M03 = NULL where Maquina = '" & maq_log & "'; update
Var_backup set Ult_data= '" & Cdate(Tempo).ToString("yyyy/MM/dd HH:mm:ss") & "',
hora_M03 = NULL where Maquina = '" & maq_log & "' "
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 20.40.040: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 20.40.050: " & ex.Message)
End Try
End If

'Tempo de execucao de tarefa
If linha.Contains("IEX_452") Then 'File execution time
Try
'Hora do logfile
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "IEX") - 1

```

```

tempo_maq_fileex = Mid(linha, posicao_inic_hora, posicao_fim_hora -
posicao_inic_hora)
'Transformar as horas de string para formato de hora
tempo_maq_fileex = Convert.ToDateTime(tempo_maq_fileex).TimeOfDay
'Juntar as horas do logfile com a data
tempo_fileex = data_final + tempo_maq_fileex
tempo_maq = tempo_fileex
'Caso haja uma linha de inicio de M03 (caso seja conjuntos de programas)
If IsDBNull(inicio_corte) = True Then
If IsDBNull(hora_m03) = False Then
inicio_corte = hora_m03
End If
End If
If IsDBNull(inicio_corte) = False Then
Try
estado = "Corte"
'Analisar a diferenca entre as datas do ultimo start e o file execution
posicao_inic_file = InStr(linha, "TIME") + 5
posicao_fim_file = InStr(linha, "(")
tempo_corte = Mid(linha, posicao_inic_file, posicao_fim_file - posicao_inic_file
- 2)
'Transformar as horas de string para formato de hora
tempo_maq_hora = Convert.ToDateTime(tempo_corte).Hour
tempo_maq_min = Convert.ToDateTime(tempo_corte).Minute
tempo_maq_seg = Convert.ToDateTime(tempo_corte).Second
tempo_corte = Convert.ToDateTime(tempo_corte).TimeOfDay
tempo_maq_seg = (tempo_maq_hora * 60 * 60) + (tempo_maq_min * 60) +
tempo_maq_seg
'PROGRAMA_NC
'escrever para uma base de dados o nome do Programa_nc
'Verificar se existe o Programa_nc na Bd
SQL = "Select * from Programas_nc where Programa_nc = '" & nome_programa_nc &
""
Tracker_programa_nc()
'Caso nª exista o item na tabela
If dt.Tables("Tracker_Programa_nc").Rows.Count = 0 Then
SQL = "INSERT INTO Programas_nc([Programa_nc], [Iniciado], [Finalizado],
[Segundo_total], [Maquina]) values('" & nome_programa_nc & "', '" &
Cdate(inicio_corte).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & tempo_maq_seg & "',
'" & maq_log & "');"
Tracker_escrever()
Else
'Adicionar o tempo de corte ao tempo
ja existente
tempo_total = dt.Tables("Tracker_Programa_nc").Rows(0).Item("Segundo_total")
tempo_total = tempo_total + tempo_maq_seg
SQL = "Update Programas_nc set Maquina = '" & maq_log & "', Segundo_total = '" &
tempo_total & "', Finalizado = '" & Cdate(tempo_maq).ToString("yyyy/MM/dd
HH:mm:ss") & "' where Programa_nc = '" & nome_programa_nc & "' "
Tracker_escrever()
End If
'LOGFILE
SQL = "INSERT INTO Log_File([Programa_nc], [Maquina], [data_entrada],
[data_saida], [Tempo_corte], [Estado]) values('" & nome_programa_nc & "', '" &
maq_log & "', '" & Cdate(inicio_corte).ToString("yyyy/MM/dd HH:mm:ss") & "', '"
& Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & tempo_maq_seg &
"', '" & estado & "');"
Tracker_escrever()
'Caso se tenha retraido a ferramenta antes de terminar o programa
Setup_false()
estado = "Corte"

```

```

'fazer update
'Verificar se a diferenca 賴superior a 0
data_dif = datediff(dateInterval.Second, Cdate(inicio_corte).date,
Cdate(tempo_fileex).date)
inicio = inicio_corte
fim = tempo_fileex
divisao_dias()
estado = "Parada"
If IsDBNull(hora_m03) = True Then
Try
'fazer update
SQL = "update Var set Ult_data= '" & Cdate(tempo_fileex).ToString("yyyy/MM/dd
HH:mm:ss") & "', Inicio_corte = NULL, Inicio_setup = NULL, Inicio_ligada = NULL,
Estado = '" & estado & "' where Maquina = '" & maq_log & "';update Var_backup
set Ult_data= '" & Cdate(tempo_fileex).ToString("yyyy/MM/dd HH:mm:ss") & "',
Inicio_corte = NULL, Inicio_setup = NULL, Inicio_ligada = NULL, Estado = '" &
estado & "' where Maquina = '" & maq_log & "' "
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 20.50.010: " & ex.Message)
End Try
Else
Try
'fazer update
SQL = "update Var set Ult_data= '" & Cdate(tempo_fileex).ToString("yyyy/MM/dd
HH:mm:ss") & "', Hora_m03= '" & Cdate(tempo_fileex).ToString("yyyy/MM/dd
HH:mm:ss") & "', Inicio_corte = NULL, Inicio_setup = NULL, Inicio_ligada = NULL,
Estado = '" & estado & "' where Maquina = '" & maq_log & "';update Var_backup
set Ult_data= '" & Cdate(tempo_fileex).ToString("yyyy/MM/dd HH:mm:ss") & "',
Hora_m03 = '" & Cdate(tempo_fileex).ToString("yyyy/MM/dd HH:mm:ss") & "',
Inicio_corte = NULL, Inicio_setup = NULL, Inicio_ligada = NULL, Estado = '" &
estado & "' where Maquina = '" & maq_log & "' "
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 20.50.020: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 20.50.030: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 20.50.040: " & ex.Message)
End Try
End If

'Aviso de emergencia que para o tempo de execucao do ficheiro
If linha.contains("IWS_999 REMOVED LINKSTATUS SESSION") Then
Try
'Hora do logfile
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "IWS") - 1
hora_maq = Mid(linha, posicao_inic_hora, posicao_fim_hora - posicao_inic_hora)
'Transformar as horas de string para formato de hora
hora_maq = Convert.ToDateTime(hora_maq).TimeOfDay
'Juntar as horas do logfile com a data
tempo_maq = data_final + hora_maq
corte_alt()
Catch ex As Exception
MsgBox("Erro 20.60.010: " & ex.Message)
End Try

```

```

End If
'Stop key
If linha.contains("dPS_016") Then
Try
'Hora do logfile
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "dPS") - 1
hora_maq = Mid(linha, posicao_inic_hora, posicao_fim_hora - posicao_inic_hora)
'Transformar as horas de string para formato de hora
hora_maq = Convert.ToDateTime(hora_maq).TimeOfDay
'Juntar as horas do logfile com a data
tempo_maq = data_final + hora_maq
corte_alt()
Catch ex As Exception
MsgBox("Erro 20.70.010: " & ex.Message)
End Try
End If
'SETUP
'Hold Key ou Pausa
If linha.contains("dPS_017") Then 'HOLD KEY (Para o calculo do tempo de setup de
ferramenta)
Try
'Obter a hora do log file
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "dPS") - 1
hora_hold_key = Mid(linha, posicao_inic_hora, posicao_fim_hora -
posicao_inic_hora)
'Transformar as horas de string para formato de hora
hora_hold_key = Convert.ToDateTime(hora_hold_key).TimeOfDay
'Juntar as horas do logfile com a data
tempo_hold_key = data_final + hora_hold_key
tempo_maq = tempo_hold_key
' Caso nãose tenha iniciado nada log a maquina ãeclarada como parada
Parada()
' Caso exista um inicio de Setup a maquina ãeclada como em setup atãeste momento
Setup_false()
' Caso exista um inicio de ligada a maquina ãeclarada como ligada ate este
momento
Ligada_false()
estado = "Setup"
'fazer update
SQL = "update Var set Inicio_setup = '" &
Cdate(tempo_hold_key).ToString("yyyy/MM/dd HH:mm:ss") & "', Ult_data= '" &
Cdate(tempo_hold_key).ToString("yyyy/MM/dd HH:mm:ss") & "', Estado = '" & estado
& "' where Maquina ='" & maq_log & "'; update Var_backup set Inicio_setup = '" &
Cdate(tempo_hold_key).ToString("yyyy/MM/dd HH:mm:ss") & "', Ult_data= '" &
Cdate(tempo_hold_key).ToString("yyyy/MM/dd HH:mm:ss") & "', Estado = '" & estado
& "' where Maquina ='" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 20.80.010: " & ex.Message)
End Try
End If
'Release Key
If linha.contains("dPS_018") Then 'RELEASED THE RELEASE KEY (Para o calculo do
tempo de setup de ferramenta)
Try
estado = "Setup"
'Obter a hora do log file
posicao_inic_hora = InStr(linha, ">") + 7

```

```

posicao_fim_hora = InStr(linha, "dPS") - 1
hora_release_key = Mid(linha, posicao_inic_hora, posicao_fim_hora -
posicao_inic_hora)
'Transformar as horas de string para formato de hora
hora_release_key = Convert.ToDateTime(hora_release_key).TimeOfDay
'Juntar as horas do logfile com a data
tempo_release_key = data_final + hora_release_key
tempo_maq = tempo_release_key
If IsDBNull(inicio_setup) = False Then
Try
'Fazer a datadiff da hora do hold Key
data_dif = datediff(dateInterval.Second, inicio_setup, tempo_release_key)
'LOGFILE
SQL = "INSERT INTO Log_File([Maquina], [data_entrada], [data_saida],
[Tempo_corte], [Estado]) values('" & maq_log & "', '" &
Cdate(inicio_setup).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & diff_hold_release &
"', '" & estado & "')";
Tracker_escrever()
inicio = inicio_setup
fim = tempo_release_key
divisao_dias()
Catch ex As Exception
MsgBox("Erro 20.80.020: " & ex.Message)
End Try
estado = "Ligada"
'fazer update
SQL = "update Var set Ult_data= '" &
Cdate(tempo_release_key).ToString("yyyy/MM/dd HH:mm:ss") & "', Inicio_setup =
NULL, Estado = '" & estado & "' where Maquina = '" & maq_log & "'; update
Var_backup set Ult_data= '" & Cdate(tempo_release_key).ToString("yyyy/MM/dd
HH:mm:ss") & "', Inicio_setup = NULL, Estado = '" & estado & "' where Maquina =
'" & maq_log & "' "
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 20.80.030: " & ex.Message)
End Try
End If
'LIGAdA

'Inicio de execucao de um bloco
If linha.contains("IWS_190") Then
Try
estado = "Ligada"
'Obter a hora do log file
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "IWS") - 1
hora_maq = Mid(linha, posicao_inic_hora, posicao_fim_hora - posicao_inic_hora)
'Transformar as horas de string para formato de hora
hora_maq = Convert.ToDateTime(hora_maq).TimeOfDay
'Juntar as horas do logfile com a data
tempo_maq = data_final + hora_maq
'Caso nãose tenha iniciado nada log a maquina ãeclarada como parada
Parada()
'Caso exista um inicio de Setup a maquina ãeclada como em setup atãeste momento
Setup_false()
'Caso exista um inicio de ligada a maquina ãeclarada como ligada ate este
momento
Ligada_false()
'fazer update

```

```

SQL = "update Var set inicio_ligada = '" & Cdate(tempo_maq).ToString("yyyy/MM/dd
HH:mm:ss") & "', Ult_data = '" & Cdate(tempo_maq).ToString("yyyy/MM/dd
HH:mm:ss") & "', Estado = '" & estado & "' where Maquina ='" & maq_log & "';
update Var_backup set inicio_ligada = '" & Cdate(tempo_maq).ToString("yyyy/MM/dd
HH:mm:ss") & "', Ult_data = '" & Cdate(tempo_maq).ToString("yyyy/MM/dd
HH:mm:ss") & "', Estado = '" & estado & "' where Maquina ='" & maq_log & ""
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 20.90.010: " & ex.Message)
End Try
End If

'Fim de execucao de um bloco
If linha.contains("IWS_191") Then
Try
estado = "Ligada"
'Obter a hora do log file
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "IWS") - 1
hora_maq = Mid(linha, posicao_inic_hora, posicao_fim_hora - posicao_inic_hora)
'Transformar as horas de string para formato de hora
hora_maq = Convert.ToDateTime(hora_maq).TimeOfDay
'Juntar as horas do logfile com a data
tempo_maq = data_final + hora_maq
ligada_alt()
Catch ex As Exception
MsgBox("Erro 20.90.020: " & ex.Message)
End Try
End If

'Movimentacao dos eixos pelo Jog
'Start Eixo positivo
If linha.contains("dPS_201 PRESSED") Then
Try
If IsDBNull(hora_m03) = True Then
estado = "Ligada"
'Obter a hora do log file
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "dPS") - 1
hora_maq = Mid(linha, posicao_inic_hora, posicao_fim_hora - posicao_inic_hora)
'Transformar as horas de string para formato de hora
hora_maq = Convert.ToDateTime(hora_maq).TimeOfDay
'Juntar as horas do logfile com a data
tempo_maq = data_final + hora_maq
'Caso nãose tenha iniciado nada log a maquina ãeclarada como parada
Parada()
'Caso exista um inicio de Setup a maquina ãeclarada como em setup atãeste
momento
Setup_false()
'Caso exista um inicio de ligada a maquina ãeclarada como ligada ate este
momento
Ligada_false()
'fazer update
SQL = "update Var set inicio_ligada = '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', Ult_data = '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', Estado = '" & estado & "'
where Maquina ='" & maq_log & "'; update Var_backup set inicio_ligada = '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', Ult_data = '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', Estado = '" & estado & "'
where Maquina ='" & maq_log & ""
Tracker_escrever()

```

```

End If
Catch ex As Exception
MsgBox("Erro 20.100.010: " & ex.Message)
End Try
End If
'End Eixo positivo
If linha.contains("dPS_201 RELEASEd") Then
Try
If IsDBNull(hora_m03) = True Then
estado = "Ligada"
'Obter a hora do log file
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "dPS") - 1
hora_maq = Mid(linha, posicao_inic_hora, posicao_fim_hora - posicao_inic_hora)
'Transformar as horas de string para formato de hora
hora_maq = Convert.ToDateTime(hora_maq).TimeOfDay
'Juntar as horas do logfile com a data
tempo_maq = data_final + hora_maq
ligada_alt()
End If
Catch ex As Exception
MsgBox("Erro 20.100.020: " & ex.Message)
End Try
End If
'Start Eixo NEGATIVO
If linha.contains("dPS_202 PRESSEd") Then
Try
If IsDBNull(hora_m03) = True Then
estado = "Ligada"
'Obter a hora do log file
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "dPS") - 1
hora_maq = Mid(linha, posicao_inic_hora, posicao_fim_hora - posicao_inic_hora)
'Transformar as horas de string para formato de hora
hora_maq = Convert.ToDateTime(hora_maq).TimeOfDay
'Juntar as horas do logfile com a data
tempo_maq = data_final + hora_maq
'Caso nãose tenha iniciado nada log a maquina ãreclarada como parada
Parada()
'Caso exista um inicio de Setup a maquina ãreclada como em setup atãste momento
Setup_false()
'Caso exista um inicio de ligada a maquina ãreclarada como ligada ate este
momento
Ligada_false()
'fazer update
SQL = "update Var set inicio_ligada = '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', Ult_data = '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', Estado = '" & estado & "'
where Maquina ='" & maq_log & "'; update Var_backup set inicio_ligada = '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', Ult_data = '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', Estado = '" & estado & "'
where Maquina ='" & maq_log & ""
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 20.100.030: " & ex.Message)
End Try
End If
'End Eixo negativo
If linha.contains("dPS_202 RELEASEd") Then

```

```

Try
If IsDBNull(hora_m03) = True Then
estado = "Ligada"
'Obter a hora do log file
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "dPS") - 1
hora_maq = Mid(linha, posicao_inic_hora, posicao_fim_hora - posicao_inic_hora)
'Transformar as horas de string para formato de hora
hora_maq = Convert.ToDateTime(hora_maq).TimeOfDay
'Juntar as horas do logfile com a data
tempo_maq = data_final + hora_maq
ligada_alt()
End If
Catch ex As Exception
MsgBox("Erro 20.100.040: " & ex.Message)
End Try
End If
'PROCEDIMENTO FIDIA
'inicio de procedimento da fidia
If linha.Contains("IWS_130") Then
Try
estado = "Ligada"
'Obter a hora do log file
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "IWS") - 1
hora_maq = Mid(linha, posicao_inic_hora, posicao_fim_hora - posicao_inic_hora)
'Transformar as horas de string para formato de hora
hora_maq = Convert.ToDateTime(hora_maq).TimeOfDay
'Juntar as horas do logfile com a data
tempo_maq = data_final + hora_maq
' Caso nãose tenha iniciado nada log a maquina ẽ declarada como parada
Parada()
' Caso exista um inicio de Setup a maquina ẽ eclada como em setup atẽste momento
Setup_false()
' Caso exista um inicio de ligada a maquina ẽ declarada como ligada ate este
momento
Ligada_false()
'fazer update
SQL = "update Var set inicio_ligada = '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', Ult_data = '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', Estado = '" & estado & "'
where Maquina ='" & maq_log & "'; update Var_backup set inicio_ligada = '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', Ult_data = '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', Estado = '" & estado & "'
where Maquina ='" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 20.100.050: " & ex.Message)
End Try
End If
'fim do procedimento da fidia
If linha.CONTAINS("IWS_132") Then
Try
estado = "Ligada"
'Obter a hora do log file
posicao_inic_hora = InStr(linha, ">") + 7
posicao_fim_hora = InStr(linha, "IWS") - 1
hora_maq = Mid(linha, posicao_inic_hora, posicao_fim_hora - posicao_inic_hora)
'Transformar as horas de string para formato de hora
hora_maq = Convert.ToDateTime(hora_maq).TimeOfDay

```



```

'Juntar as horas do logfile com a data
tempo_maq = data_final + hora_maq
ligada_alt()
Catch ex As Exception
MsgBox("Erro 20.100.060: " & ex.Message)
End Try
End If

'Recolha de mensagens de erro
If Mid(linha, InStr(linha, ">") + 16, 1) = "E" And Mid(linha, InStr(linha, ">")
+ 19, 1) = "_" Then
Try
'Verificar qual última linha escrita para a base de dados
codigo_erro = Mid(linha, InStr(linha, ">") + 16, 7)
tipo_mensagem = "Erro"
mensagem_erro = Mid(linha, InStr(linha, ">") + 16)
tempo_erro = Cdate(Mid(linha, InStr(linha, ">") + 7, 8)).TimeOfDay
If mensagem_erro.contains("") Then
mensagem_erro = Replace(mensagem_erro, "", "")
End If
'Obter última linha escrita da base de dados
SQL = "SELECT TOP(10) Empresa, Maquina, Tipo, Cod_erro, Mensagem, data, Id FROM
Mensagem_erro where Maquina = '" & maq_log & "' and Cod_erro = '" & codigo_erro
& "' ORdER BY Id dESC"
Tracker_variavel()
If dt.Tables("Tracker_Variavel").Rows.Count <> 0 Then
tempo_erro_old = dt.Tables("Tracker_Variavel").Rows(0).Item("data")
End If
'Hora do logfile
Tempo = data_final + tempo_erro
If datediff(dateInterval.Second, tempo_erro_old, Tempo) > 10 Then
'Escrever o erro na tabela da maquina
SQL = "INSERT INTO Mensagem_erro([Empresa], [Maquina], [Tipo], [Cod_erro],
[Mensagem], [data]) values('" & empresa & "', '" & maq_log & "', '" &
tipo_mensagem & "', '" & codigo_erro & "', '" & mensagem_erro & "', '" &
Cdate(Tempo).ToString("yyyy-MM-dd HH:mm:ss") & " ')"
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 20.100.070: " & ex.Message)
End Try
End If
'Recolha de mensagens de erros fatais
If Mid(linha, InStr(linha, ">") + 16, 1) = "F" And Mid(linha, InStr(linha, ">")
+ 19, 1) = "_" Then
Try
codigo_erro = Mid(linha, InStr(linha, ">") + 16, 7)
tipo_mensagem = "Fatal"
mensagem_erro = Mid(linha, InStr(linha, ">") + 16)
If mensagem_erro.contains("") Then
mensagem_erro = Replace(mensagem_erro, "", "")
End If
'Obter última linha escrita da base de dados
SQL = "SELECT TOP(10) Empresa, Maquina, Tipo, Cod_erro, Mensagem, data, Id FROM
Mensagem_erro where maquina = '" & maq_log & "' and Cod_erro = '" & codigo_erro
& "' ORdER BY Id dESC"
Tracker_variavel()
If dt.Tables("Tracker_Variavel").Rows.Count <> 0 Then
tempo_erro_old = dt.Tables("Tracker_Variavel").Rows(0).Item("data")
End If
'Hora do logfile

```

```

tempo_erro = Cdate(Mid(linha, InStr(linha, ">") + 7, 8)).TimeOfDay
Tempo = data_final + tempo_erro
If datediff(dateInterval.Second, tempo_erro_old, Tempo) > 60 Then
'Escrever o erro na tabela da maquina
SQL = "INSERT INTO Mensagem_erro([Empresa], [Maquina], [Tipo], [Cod_erro],
[Mensagem], [data]) values('" & empresa & "', '" & maq_log & "', '" &
tipo_mensagem & "', '" & codigo_erro & "', '" & mensagem_erro & "', '" &
Cdate(Tempo).ToString("yyyy-MM-dd HH:mm:ss") & "' )"
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 20.100.080: " & ex.Message)
End Try
End If
'Escrever para a base de dados a linha atual
'Adicionar (+1) ao numero de linhas
Try
linha_atual = linha_atual + 1
'fazer update
SQL = "update Var set Linha_atual = '" & linha_atual & "' where Maquina = '" &
maq_log & "'; update Var_backup set Linha_atual = '" & linha_atual & "' where
Maquina = '" & maq_log & "' "
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 20.110.010: " & ex.Message)
End Try
Else
GoTo tempo_ciclo
End If
'Proxima maquina
Next
tempo_ciclo:
'Limpar a list do logfile
List_logfile.Clear()
Catch ex As Exception
MsgBox("Erro 20: " & ex.Message)
End Try
End Sub
Public Sub divisao_turnos()
tempo1 = Nothing
tempo2 = Nothing
tempo3 = Nothing
tempo_fds = Nothing
t_inicio = Nothing
t_fim = Nothing
Try
'descobrir em que turno se iniciou
If inicio_turno.hour >= 0 And inicio_turno.hour < 7 Then
t_inicio = 3
ElseIf inicio_turno.hour >= 7 And inicio_turno.timeofday <=
Cdate("15:30:00").TimeOfDay Then
t_inicio = 1
ElseIf (inicio_turno.hour >= 15 And inicio_turno.minute >= 30) And
inicio_turno.hour < 24 Then
t_inicio = 2
End If
'descobrir em que turno de acabou
If fim_turno.Hour >= 0 And fim_turno.Hour < 7 Then
t_fim = 3
ElseIf fim_turno.hour >= 7 And fim_turno.timeofday <=
Cdate("15:30:00").TimeOfDay Then

```

```

t_fim = 1
ElseIf fim_turno.Hour >= 15 And fim_turno.Minute >= 30 And fim_turno.Hour < 24
Then
t_fim = 2
End If
'O tempo ja vem separado por data
'Verificar se ia da semana
If Cdate(inicio_turno).dayOfWeek = dayOfWeek.Saturday Or
Cdate(inicio_turno).dayOfWeek = dayOfWeek.Sunday Then
Try
'Apresentar tudo junto numa coluna diferente
tempo_fds = datediff(dateInterval.Second, inicio_turno, fim_turno)
Catch ex As Exception
MsgBox("Erro 30.10.010: " & ex.Message)
End Try
Else
'Perceber ser ecessario dividir o tempo por cada um dos turnos ou se esta tudo
no mesmo turno
If t_inicio = t_fim Then
Try
'Escrever direto na base de dados, INDICANDO O TURNO
If t_inicio = 1 Then
tempo1 = datediff(dateInterval.Second, inicio_turno, fim_turno)
ElseIf t_inicio = 2 Then
tempo2 = datediff(dateInterval.Second, inicio_turno, fim_turno)
ElseIf t_inicio = 3 Then
tempo3 = datediff(dateInterval.Second, inicio_turno, fim_turno)
End If
Catch ex As Exception
MsgBox("Erro 30.10.020: " & ex.Message)
End Try
Else
If t_inicio = 3 And t_fim = 2 Then 'Caso sejam turnos n seguidos
Try
'dividir o tempo pelos 3 turnos
'Fazer a diferenca do inicio_turno para as 7
tempo3 = datediff(dateInterval.Second, inicio_turno,
Convert.ToDateTime(Cdate(inicio_turno).date + Cdate("07:00:00").TimeOfDay))
'Fazer a diferenca do fim_turno para as 3:30
tempo2 = datediff(dateInterval.Second, Convert.ToDateTime(Cdate(fim_turno).date
+ Cdate("15:30:00").TimeOfDay), fim_turno)
tempo1 = datediff(dateInterval.Second, Convert.ToDateTime(Cdate("07:00:00")),
Convert.ToDateTime(Cdate("15:30:00")))
Catch ex As Exception
MsgBox("Erro 30.10.030: " & ex.Message)
End Try
Else 'Sao turnos seguidos
'Fazer a diferenca de tempo da hora inicial ao limite maximo do turno em questao
Try
If t_inicio = 1 Then
Try
'Fazer a diferenca do inicio_turno para as 3:30
tempo1 = datediff(dateInterval.Second, inicio_turno,
Convert.ToDateTime(Cdate(inicio_turno).date + Cdate("15:30:00").TimeOfDay))
'Fazer a diferenca do fim_turno para as 15:30
tempo2 = datediff(dateInterval.Second, Convert.ToDateTime(Cdate(fim_turno).date
+ Cdate("15:30:00").TimeOfDay), fim_turno)
tempo3 = Nothing
Catch ex As Exception
MsgBox("Erro 30.10.040: " & ex.Message)

```

```

End Try
ElseIf t_inicio = 3 Then
Try
'Fazer a diferenca do inicio_turno para as 7
tempo3 = datediff(dateInterval.Second, inicio_turno,
Convert.ToDateTime(Cdate(inicio_turno).date + Cdate("07:00:00").TimeOfDay))
'Fazer a diferenca do fim_turno para as 7
tempo1 = datediff(dateInterval.Second, Convert.ToDateTime(Cdate(fim_turno).date
+ Cdate("07:00:00").TimeOfDay), fim_turno)
tempo2 = Nothing
Catch ex As Exception
MsgBox("Erro 30.10.050: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 30.10.060: " & ex.Message)
End Try
End If
End If
End If
Catch ex As Exception
MsgBox("Erro 30: " & ex.Message)
End Try
End Sub
Sub divisao_dias()
If data_dif < 0 Then
Erro()
ElseIf Cdate(fim).day - Cdate(inicio).day > 0 Or Cdate(fim).Month -
Cdate(inicio).Month > 0 Or Cdate(fim).Year - Cdate(inicio).Year > 0 Then
dias = datediff(dateInterval.day, inicio.date, fim.date)
' dias = Cdate(fim).day - Cdate(inicio).day
If dias < 0 Then
dias = Int(((datediff(dateInterval.Second, inicio, fim)) / 86400) + 1)
End If
'Antes da meia noite
data_dif = datediff(dateInterval.Second, inicio, Cdate(inicio).date +
Cdate("23:59:59").TimeOfDay)
If data_dif < 0 Then
Erro()
Else
inicio_turno = inicio
fim_turno = Cdate(Cdate(inicio).date + Cdate("23:59:59").TimeOfDay)
divisao_turnos()
data_dif = datediff(dateInterval.Second, inicio, Cdate(inicio).date +
Cdate("23:59:59").TimeOfDay)
'Fazer Update
SQL = "INSERT INTO " & maq_log & "([Inicio], [Fim], [Estado], [Tempo],
[Turno_1], [Turno_2], [Turno_3], [Turno_fds]) values('" &
Cdate(inicio).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(Cdate(inicio).date + Cdate("23:59:59").TimeOfDay).ToString("yyyy/MM/dd
HH:mm:ss") & "', '" & estado & "', '" & data_dif & "', '" & tempo1 & "', '" &
tempo2 & "', '" & tempo3 & "', '" & tempo_fds & "');"
Tracker_escrever()
tempo1 = Nothing
tempo2 = Nothing
tempo3 = Nothing
tempo_fds = Nothing
End If
While dias > 0
If dias > 1 Then

```

```

data_dif = datediff(dateInterval.Second, Cdate(dateAdd(dateInterval.day, (1),
inicio)).date + Cdate("00:00:01").TimeOfDay, Cdate(dateAdd(dateInterval.day,
(1), inicio)).date + Cdate("23:59:59").TimeOfDay)
inicio_turno = Cdate(dateAdd(dateInterval.day, (1), inicio)).date +
Cdate("00:00:01").TimeOfDay
fim_turno = Cdate(dateAdd(dateInterval.day, (1), inicio)).date +
Cdate("23:59:59").TimeOfDay
divisao_turnos()
data_dif = datediff(dateInterval.Second, Cdate(dateAdd(dateInterval.day, (1),
inicio)).date + Cdate("00:00:01").TimeOfDay, Cdate(dateAdd(dateInterval.day,
(1), inicio)).date + Cdate("23:59:59").TimeOfDay)
'Fazer Update
SQL = "INSERT INTO " & maq_log & "([Inicio], [Fim], [Estado], [Tempo],
[Turno_1], [Turno_2], [Turno_3], [Turno_fds]) values('" &
Cdate(Cdate(dateAdd(dateInterval.day, (1), inicio)).date +
Cdate("00:00:01").TimeOfDay).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(Cdate(dateAdd(dateInterval.day, (1), inicio)).date +
Cdate("23:59:59").TimeOfDay).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & estado &
"', '" & data_dif & "', '" & tempo1 & "', '" & tempo2 & "', '" & tempo3 & "', '"
& tempo_fds & "');"
Tracker_escrever()
inicio = Cdate(dateAdd(dateInterval.day, (1), inicio)).date
tempo1 = Nothing
tempo2 = Nothing
tempo3 = Nothing
tempo_fds = Nothing
End If
If dias = 1 Then
'depois da meia noite
data_dif = datediff(dateInterval.Second, Cdate(fim).date +
Cdate("00:00:01").TimeOfDay, fim)
If data_dif < 0 Then
Erro()
Else
inicio_turno = Cdate(Cdate(fim).date + Cdate("00:00:01").TimeOfDay)
fim_turno = Cdate(fim)
divisao_turnos()
data_dif = datediff(dateInterval.Second, Cdate(fim).date +
Cdate("00:00:01").TimeOfDay, fim)
'Fazer Update
SQL = "INSERT INTO " & maq_log & "([Inicio], [Fim], [Estado], [Tempo],
[Turno_1], [Turno_2], [Turno_3], [Turno_fds]) values('" & Cdate(Cdate(fim).date
+ Cdate("00:00:01").TimeOfDay).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(fim).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & estado & "', '" & data_dif
& "', '" & tempo1 & "', '" & tempo2 & "', '" & tempo3 & "', '" & tempo_fds &
"');"
Tracker_escrever()
tempo1 = Nothing
tempo2 = Nothing
tempo3 = Nothing
tempo_fds = Nothing
End If
End If
dias = dias - 1
End While
Else
inicio_turno = inicio
fim_turno = fim
divisao_turnos()
data_dif = datediff(dateInterval.Second, inicio, fim)
'Fazer Update

```

```

SQL = "INSERT INTO " & maq_log & "([Inicio], [Fim], [Estado], [Tempo],
[Turno_1], [Turno_2], [Turno_3], [Turno_fds]) values('" &
Cdate(inicio).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(fim).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & estado & "', '" & data_dif
& "', '" & tempo1 & "', '" & tempo2 & "', '" & tempo3 & "', '" & tempo_fds &
"');"
Tracker_escrever()
tempo1 = Nothing
tempo2 = Nothing
tempo3 = Nothing
tempo_fds = Nothing
End If
End Sub
Public Sub Parada()
Try
If IsDBNull(inicio_corte) = True And IsDBNull(inicio_ligada) = True And
IsDBNull(inicio_setup) = True Then
estado = "Parada"
data_dif = datediff(dateInterval.Second, dateAdd(dateInterval.Second, (1),
ult_data), dateAdd(dateInterval.Second, (-1), tempo_maq))
'Caso o valor da data seja menor que 0
If data_dif > 0 Then
'LOGFILE
SQL = "INSERT INTO Log_File([Maquina], [data_entrada], [data_saida],
[Tempo_corte], [Estado]) values('" & maq_log & "', '" &
Cdate(ult_data).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & data_dif & "', '" &
estado & "');"
Tracker_escrever()
inicio = ult_data
fim = tempo_maq
divisao_dias()
ElseIf data_dif = 0 Then
End If
'Escrever a data de ultima data
SQL = "update Var set Ult_data = '" & Cdate(tempo_maq).ToString("yyyy/MM/dd
HH:mm:ss") & "', Estado = '" & estado & "' where Maquina ='" & maq_log & "';
update Var_backup set Ult_data = '" & Cdate(tempo_maq).ToString("yyyy/MM/dd
HH:mm:ss") & "', Estado = '" & estado & "' where Maquina ='" & maq_log & "'"
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 40: " & ex.Message)
End Try
End Sub
Public Sub Setup_false()
Try
If IsDBNull(inicio_setup) = False Then
estado = "Setup"
data_dif = datediff(dateInterval.Second, inicio_setup, tempo_maq)
inicio = inicio_setup
fim = tempo_maq
divisao_dias()
'fazer update
SQL = "update Var set Inicio_setup = NULL where Maquina = '" & maq_log & "';
update Var_backup set Inicio_setup = NULL where Maquina = '" & maq_log & "'"
Try
Tracker_escrever()
Catch ex As Exception
Tracker_escrever()
End Try

```

```

End If
Catch ex As Exception
MsgBox("Erro 40: " & ex.Message)
End Try
End Sub
Public Sub Ligada_false()
Try
If IsDBNull(inicio_ligada) = False Then
estado = "Ligada"
data_dif = datediff(dateInterval.Second, inicio_ligada, tempo_maq)
inicio = inicio_ligada
fim = tempo_maq
divisao_dias()
'fazer update
SQL = "update Var set Inicio_ligada = NULL where Maquina = '" & maq_log & "';
update Var_backup set Inicio_ligada = NULL where Maquina = '" & maq_log & "' "
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 50: " & ex.Message)
End Try
End Sub
Public Sub Corte_false()
Try
If IsDBNull(inicio_corte) = False Then
estado = "Ligada"
data_dif = datediff(dateInterval.Second, inicio_corte, tempo_maq)
'Caso a diferenca entre datas passe um dia
inicio = inicio_corte
fim = tempo_maq
divisao_dias()
'fazer update
SQL = "update Var set Inicio_corte = NULL where Maquina = '" & maq_log & "';
update Var_backup set Inicio_corte = NULL where Maquina = '" & maq_log & "' "
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 60: " & ex.Message)
End Try
End Sub
Public Sub Tracker_variavel()
Try
Tracker_variavel: connection.Close()
Try
connection.Open()
If dt.Tables.Contains("Tracker_Variavel") Then
dt.Tables("Tracker_Variavel").Clear()
End If
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dt, "Tracker_Variavel")
connection.Close()
Catch ex As Exception
If ex.ToString <> "Execution Timeout Expired. The timeout period elapsed prior
to completion of the operation or the server is not responding." Then
End If
GoTo Tracker_variavel
End Try
Catch ex As Exception
MsgBox("Erro 80: " & ex.Message)
End Try
End Sub

```

```

Public Sub Tracker_programa_nc()
Try
Tracker_programa_nc: connection.Close()
Try
connection.Open()
If dt.Tables.Contains("Tracker_Programa_nc") Then
dt.Tables("Tracker_Programa_nc").Clear()
End If
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dt, "Tracker_Programa_nc")
connection.Close()
Catch ex As Exception
GoTo Tracker_programa_nc
End Try
Catch ex As Exception
MsgBox("Erro 90: " & ex.Message)
End Try
End Sub
Sub Tracker_escrever()
tempo_inic_escrever = Now
Try
SQL_Label.Text = SQL
Tracker_escrever: connection.Close()
tempo_atual_escrever = Now
Try
Cmd = New SqlCommand(SQL, connection)
connection.Open()
command = New SqlCommand(SQL, connection)
command.ExecuteNonQuery()
connection.Close()
Catch ex As Exception
If datediff(dateInterval.Minute, tempo_inic_escrever, tempo_atual_escrever) > 2
Then
MsgBox("Erro 1.10: " & ex.Message)
End If
GoTo Tracker_escrever
End Try
Catch ex As Exception
MsgBox("Erro 1: " & ex.Message)
End Try
End Sub
Public Sub Tracker_var()
Try
Tracker_var: connection.Close()
Try
connection.Open()
If dt.Tables.Contains("Tracker_Variaveis") Then
dt.Tables("Tracker_Variaveis").Clear()
End If
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dt, "Tracker_Variaveis")
connection.Close()
Catch ex As Exception
GoTo Tracker_var
End Try
Catch ex As Exception
MsgBox("Erro 3: " & ex.Message)
End Try
End Sub
Sub Erro()
Erro: Try

```



```

SQL = "INSERT INTO Erros([Maquina], [Inicio], [Fim], [Estado], [Tempo])
values('" & maq_log & "', '" & Cdate(ult_data).ToString("yyyy/MM/dd HH:mm:ss") &
"', '" & Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & estado &
"', '" & data_dif & "');"
Tracker_escrever()
Catch ex As Exception
GoTo Erro
End Try
End Sub
Sub corte_alt()
Try
If IsDBNull(inicio_corte) = False Then
estado = "Corte"
'PROGRAMA_NC
'escrever para uma base de dados o nome do Programa_nc
'Verificar se existe o Programa_nc na Bd
SQL = "Select * from Programas_nc where Programa_nc = '" & nome_programa_nc &
""
Tracker_programa_nc()
'Caso haja uma linha de inicio de M03 (caso seja conjuntos de programas)
data_dif = datediff(dateInterval.Second, inicio_corte, tempo_maq)
'Caso nã exista o item na tabela
Try
If dt.Tables("Tracker_Programa_nc").Rows.Count = 0 Then
SQL = "INSERT INTO Programas_nc([Programa_nc], [Iniciado], [Finalizado],
[Segundo_total], [Maquina]) values('" & nome_programa_nc & "', '" &
Cdate(inicio_corte).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & data_dif & "', '" &
maq_log & "');"
Tracker_escrever()
Else
'Adicionar o tempo de corte ao tempo ja existente
tempo_total = dt.Tables("Tracker_Programa_nc").Rows(0).Item("Segundo_total")
tempo_total = tempo_total + data_dif
SQL = "Update Programas_nc set Maquina = '" & maq_log & "', Segundo_total = '" &
tempo_total & "', Finalizado = '" & Cdate(tempo_maq).ToString("yyyy/MM/dd
HH:mm:ss") & "' where Programa_nc = '" & nome_programa_nc & "' "
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 4.10: " & ex.Message)
End Try
Try
'Escrever essa hora para o tempo de corte da maquina
Setup_false()
data_dif = datediff(dateInterval.Second, inicio_corte, tempo_maq)
'LOGFILE
SQL = "INSERT INTO Log_File([Programa_nc], [Maquina], [data_entrada],
[data_saida], [Tempo_corte], [Estado]) values('" & nome_programa_nc & "', '" &
maq_log & "', '" & Cdate(inicio_corte).ToString("yyyy/MM/dd HH:mm:ss") & "', '"
& Cdate(tempo_fileex).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & data_dif & "',
'" & estado & "');"
Tracker_escrever()
estado = "Corte"
inicio = inicio_corte
fim = tempo_maq
divisao_dias()
estado = "Parada"
Catch ex As Exception
MsgBox("Erro 4.20: " & ex.Message)
End Try

```

```

Try
If IsDBNull(hora_m03) = True Then
'fazer update
SQL = "update Var set Ult_data= '" & Cdate(tempo_maq).ToString("yyyy/MM/dd
HH:mm:ss") & "', Inicio_corte = NULL, Inicio_setup = NULL, Inicio_ligada = NULL,
Estado = '" & estado & "' where Maquina = '" & maq_log & "'; update Var_backup
set Ult_data= '" & Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "',
Inicio_corte = NULL, Inicio_setup = NULL, Inicio_ligada = NULL, Estado = '" &
estado & "' where Maquina = '" & maq_log & "' "
Tracker_escrever()
Else
'fazer update
SQL = "update Var set Ult_data= '" & Cdate(tempo_maq).ToString("yyyy/MM/dd
HH:mm:ss") & "', Hora_m03 = '" & Cdate(tempo_maq).ToString("yyyy/MM/dd
HH:mm:ss") & "', Inicio_corte = NULL, Inicio_setup = NULL, Inicio_ligada = NULL,
Estado = '" & estado & "' where Maquina = '" & maq_log & "'; update Var_backup
set Ult_data= '" & Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "',
Hora_m03 = '" & Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "',
Inicio_corte = NULL, Inicio_setup = NULL, Inicio_ligada = NULL, Estado = '" &
estado & "' where Maquina = '" & maq_log & "' "
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 4.30: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 4: " & ex.Message)
End Try
End Sub
Sub ligada_alt()
Try
If IsDBNull(inicio_ligada) = False Then
Try
'Fazer a datadiff da hora do hold Key
data_dif = datediff(dateInterval.Second, inicio_ligada, tempo_maq)
'LOGFILE
SQL = "INSERT INTO Log_File([Maquina], [data_entrada], [data_saida],
[Tempo_corte], [Estado]) values('" & maq_log & "', '" &
Cdate(inicio_ligada).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & data_dif & "', '" &
estado & "' );"
Tracker_escrever()
inicio = inicio_ligada
fim = tempo_maq
divisao_dias()
estado = "Parada"
'fazer update
SQL = "update Var set Ult_data= '" & Cdate(tempo_maq).ToString("yyyy/MM/dd
HH:mm:ss") & "', inicio_ligada = NULL, inicio_setup = NULL, Estado = '" & estado
& "' where Maquina = '" & maq_log & "'; update Var_backup set Ult_data= '" &
Cdate(tempo_maq).ToString("yyyy/MM/dd HH:mm:ss") & "', inicio_ligada = NULL,
inicio_setup = NULL, Estado = '" & estado & "' where Maquina = '" & maq_log & "'
"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 2.10: " & ex.Message)
End Try
Else
Try
Corte_false()

```

```

'fazer update
SQL = "update Var set Ult_data= '" & Cdate(dateAdd(dateInterval.Second, (-1),
tempo_maq)).ToString("yyyy/MM/dd HH:mm:ss") & "', inicio_ligada = NULL,
inicio_setup = NULL, Estado = '" & estado & "'" where Maquina = '" & maq_log &
"'; update Var_backup set Ult_data= '" & Cdate(dateAdd(dateInterval.Second, (-
1), tempo_maq)).ToString("yyyy/MM/dd HH:mm:ss") & "', inicio_ligada = NULL,
inicio_setup = NULL, Estado = '" & estado & "'" where Maquina = '" & maq_log &
"' "

Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 2.20: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 2: " & ex.Message)
End Try
End Sub
End Class

```

Anexo B - Automatismo de recolha de informação de comandos CNC iTNC 530 Heidenhain

```

Public Class Tracker_heidenhain_V15
'declaracao de variaveis
'maquinas sao declaradas da mesma forma que o automatismo fidia
Private Sub Tracker_heidenhain_V14_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
Reiniciar:
Try
time_inic = Nothing
'ciclo para cada maquina na lista de maquinas do automatismo
For Each maq_log In list_maquina
nr_ficheiro = 0
Try
If IsNothing(list_logfile) = False Then
list_logfile.clear()
End If
Catch ex As Exception
MsgBox("Erro 1.10.10: " & ex.Message)
End Try
Try
If File.Exists(diretorio_logfile) Then
File.delete(diretorio_logfile)
Thread.Sleep(2000)
End If
Catch ex As Exception
GoTo Apagar
End Try
'Ler todas as variaveis dessa maquina
Try
SQL = "Select * from Var_heid where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
IP = dt.Tables("Tracker_Variaveis").Rows(0).Item("IP")
ult_edicao = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_edicao")

```

```

UltimaLinha = dt.Tables("Tracker_Variaveis").Rows(0).Item("UltimaLinha")
End If
Catch ex As Exception
MsgBox("Erro 1.10.20: " & ex.Message)
End Try
If IsDBNull(UltimaLinha) = True Then
UltimaLinha = ult_data
End If
Try
If IsDBNull(ult_data) = True Then
SQL = "SELECT TOP (5) Inicio, Fim, Estado, Tempo, Id FROM " & maq_log & "
order by id desc"
Tracker_var()
If dt.Tables("Tracker_Variaveis").Rows.Count <> 0 Then
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Fim")
End If
End If
Catch ex As Exception
MsgBox("Erro 1.10.30: " & ex.Message)
End Try
'Copiar o LogFile para o pc
Try
If IsDBNull(UltimaLinha) = True Then
procId = Shell("cmd /k cd H:\ & d: & cd d:\Tracker\Log_file\Heidenhain &
""C:\Program Files (x86)\HEIDENHAIN\TNCremo\tnccmd.exe"" " & IP & " log
LogFile_heidenhain 20.11.2017 ", AppWinStyle.Hide, True, 60000)
pr = Process.GetProcessById(procId)
pr.Kill()
Thread.Sleep(4000)
Else
procId = Shell("cmd /k cd H:\ & d: & cd d:\Tracker\Log_file\Heidenhain &
""C:\Program Files (x86)\HEIDENHAIN\TNCremo\tnccmd.exe"" " & IP & " log
LogFile_heidenhain " & Cdate(UltimaLinha).ToString("dd.MM.yyyy;HH:mm:ss") & " ",
AppWinStyle.Hide, True, 5000)
pr = Process.GetProcessById(procId)
pr.Kill()
Thread.Sleep(4000)
End If
Catch ex As Exception
MsgBox("Erro 1.10.40: " & ex.Message)
End Try
'Case o ficheiro esteja a ser copiado
Try
While datediff(dateInterval.Second, File.GetLastWriteTime(diretorio_logfile),
Now) < 3
If IsDBNull(ult_edicao) = False Then
If File.GetLastWriteTime(diretorio_logfile) = Cdate(ult_edicao) Then
GoTo nextline
End If
End If
End While
Catch ex As Exception
MsgBox("Erro 1.10.70: " & ex.Message)
End Try
'Case nao exista ficheiro
Try
If File.Exists(diretorio_logfile) = False Then
SQL = "update Var_heid set Ligacao = 0 where Maquina = '" & maq_log & "'; update
Var_backup_heid set Ligacao = 0 where Maquina = '" & maq_log & "'"
Tracker_escrever()
GoTo nextline

```

```

Else
SQL = "update Var_heid set Ligacao = 1 where Maquina = '" & maq_log & "'; update
Var_backup_heid set Ligacao = 1 where Maquina = '" & maq_log & "'"
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 1.10.80: " & ex.Message)
End Try
Readagain:
Try
'passar o ficheiro para list
list_logfile = File.ReadLines(diretorio_logfile).ToList
Catch ex As Exception
GoTo Readagain
End Try
linha_count = list_logfile.Count
'Inserir dados no Form
maq_label.Text = maq_log.ToString
Ultdata_Label.Text = Cdate(ult_data).ToString("yyyy/MM/dd HH:mm:ss")
LinhaTotal_Label.Text = linha_count.ToString
'apagar da List o que ja foi lido antes
Try
If linha_atual >= 0 Then
linha_atual = 0
'Fazer update
SQL = "update Var_heid set Linha = '" & linha_count & "', Linha_atual = '" &
linha_atual & "' where Maquina = '" & maq_log & "'; update Var_backup_heid set
Linha = '" & linha_count & "', Linha_atual = '" & linha_atual & "' where Maquina
= '" & maq_log & "'"
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 1.10.90: " & ex.Message)
End Try
linhas()
Apagar:
nextline:
Try
If File.Exists(diretorio_logfile) Then
File.delete(diretorio_logfile)
Thread.Sleep(2000)
End If
Catch ex As Exception
GoTo Apagar
End Try
Next
GoTo Reiniciar
Catch ex As Exception
MsgBox("Erro 1.10: " & ex.Message)
End Try
End Sub
'Rotina de analise as linhas do ficheiro .log
Sub linhas()
tempo_inic = Now
Try
For i = 0 To list_logfile.count - 1
tempo_atual = Now
'enquanto o tempo de analise for menor que 17mins o ciclo prosegue
If datediff(dateInterval.Minute, tempo_inic, tempo_atual) < 17 Then
linha = list_logfile(i)
'Escrever dados para o form

```

```

LinhaAtual_Label.Text = linha_atual.ToString
TextoLinha_Label.Text = linha.ToString
linha_atual = linha_atual + 1
'Escrever a nova linha na base de dados Var
SQL = "update Var_heid set Linha_atual = '" & linha_atual & "' where Maquina =
'" & maq_log & "'; update Var_backup_heid set Linha_atual = '" & linha_atual &
"' where Maquina = '" & maq_log & "'"
Tracker_escrever()
'data
If linha.contains("__date:") Then
Try
'Verificar se ja existe a linha
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
data_final = dt.Tables("Tracker_Variaveis").Rows(0).Item("data_final")
estado = dt.Tables("Tracker_Variaveis").Rows(0).Item("Estado")
End If
data = Mid(linha, (InStr(linha, ":") + 6), InStr((InStr(linha, ":") + 6), linha,
"_) - (InStr(linha, ":") + 6))
If IsDBNull(data_final) = True Then
data_final = data
End If
If Cdate(data).date > Cdate(data_final).date Then
Try
'Escrever para a base de dados de variaveis
SQL = "update Var_heid set data_final = '" & Cdate(data).ToString("yyyy/MM/dd")
& "', linha_atual = 0 where Maquina = '" & maq_log & "'; update Var_backup_heid
set data_final = '" & Cdate(data).ToString("yyyy/MM/dd") & "', linha_atual = 0
where Maquina = '" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.10: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 1.20.20: " & ex.Message)
End Try
End If
'Inicio Corte
If linha.contains("Stib: ON") Then
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid where Maquina = '" & maq_log & "'"
Tracker_var()

If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
End If
Try

```

```

hora_inic = Convert.ToDateTime(Mid(linha, (InStr(linha, "ON") + 38),
(InStr(linha, "ON") + 46) - (InStr(linha, "ON") + 38))).TimeOfDay
data_inic = Convert.ToDateTime(Mid(linha, (InStr(linha, "ON") + 51),
(InStr(linha, "ON") + 62) - (InStr(linha, "ON") + 51))).date
time_inic = data_inic + hora_inic
If IsDBNull(ult_data) = False Then
'Tempo que a maquina esteve parada. (diferenca entre a hora de paragem de corte
ate ao momento)
estado = "Parada"
time_stop = datediff(dateInterval.Second, ult_data, time_inic)
'Escrever na base de dados de log file
SQL = "INSERT INTO Log_File_heid ([Maquina], [data_entrada], [data_saida],
[Tempo_corte], [Estado]) values( '" & maq_log & "', '" &
Cdate(ult_data).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(time_inic).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & time_stop & "', '" &
estado & "') "
Tracker_escrever()
'Escrever na base de dados da maquina
data_dif = datediff(dateInterval.Second, Cdate(ult_data).date,
Cdate(time_inic).date)
inicio = ult_data
fim = time_inic
divisao_dias()
End If
estado = "Corte"
'Escrever na base de dados os valores
SQL = "update Var_heid set Inicio_corte= '" &
Cdate(time_inic).ToString("yyyy/MM/dd HH:mm:ss") & "', Estado = '" & estado &
"', Ult_data = '" & Cdate(time_inic).ToString("yyyy/MM/dd HH:mm:ss") & "' where
Maquina = '" & maq_log & "'; update Var_backup_heid set Inicio_corte= '" &
Cdate(time_inic).ToString("yyyy/MM/dd HH:mm:ss") & "', Estado = '" & estado &
"', Ult_data = '" & Cdate(time_inic).ToString("yyyy/MM/dd HH:mm:ss") & "' where
Maquina = '" & maq_log & "' "
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.30: " & ex.Message)
End Try
End If
'Programa NC
'Ler sempre as linhas de TNC
Try
If linha.Contains("TNC:") Then
Try
prog_nc = Mid(linha, (InStrRev(linha, "\") + 1), (InStrRev(linha, ".h") + 2))
If prog_nc = "" Then
prog_nc = Mid(linha, (InStrRev(linha, "\") + 1), (InStrRev(linha, ".H") + 2))
End If
Catch ex As Exception
MsgBox("Erro 1.20.40: " & ex.Message)
End Try
Try
If Not linha.IndexOf("Renishaw", 0, StringComparison.CurrentCultureIgnoreCase) >
-1 Or prog_nc.IndexOf("MdI.H", 0, StringComparison.CurrentCultureIgnoreCase) > -
1 Or prog_nc.IndexOf("NC.H", 0, StringComparison.CurrentCultureIgnoreCase) > -1
Or prog_nc.IndexOf("nc118.H", 0, StringComparison.CurrentCultureIgnoreCase) > -1
Or prog_nc.IndexOf("Troca-50%.H", 0, StringComparison.CurrentCultureIgnoreCase)
> -1 Or prog_nc.IndexOf("Troca-100%.H", 0,
StringComparison.CurrentCultureIgnoreCase) > -1 Then
'Escrever nome do programa Nc temporario

```

```

SQL = "UPDATE Var_heid set Programa_nc_temp = '" & prog_nc & "' where Maquina =
'" & maq_log & "'; UPDATE Var_backup_heid set Programa_nc_temp = '" & prog_nc &
'" where Maquina = '" & maq_log & "'"
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 1.20.50: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 1.20.60: " & ex.Message)
End Try
'Programa NC dentro do ciclo de Stib
Try
If linha.contains(" TNC:") And (list_logfile(i - 1).ToString).Contains("Info:
MAIN PGM") And (list_logfile(i - 2).ToString).Contains("Stib: ON") Then 'Ler
o nome do programa Nc.
Try
prog_nc = Mid(linha, (InStrRev(linha, "\") + 1), (InStrRev(linha, ".h")))
If prog_nc = "" Then
prog_nc = Mid(linha, (InStrRev(linha, "\") + 1), (InStrRev(linha, ".H")))
End If
Catch ex As Exception
MsgBox("Erro 1.20.70: " & ex.Message)
End Try
'Caso seja correto escrever na Bd, se for um nome generico usar a informacao na
base de dados para perceber qual e o programa que esta a correr
'Perceber se a linha chama algum programa de medicao de ferramenta
If Not linha.indexof("NC_LASER", 0, StringComparison.CurrentCultureIgnoreCase) >
-1 And (linha.indexof("dUMMY", 0, StringComparison.CurrentCultureIgnoreCase) > -
1 Or linha.indexof("PRECHG", 0, StringComparison.CurrentCultureIgnoreCase) > -1
Or linha.indexof("Laser", 0, StringComparison.CurrentCultureIgnoreCase) > -1 Or
linha.indexof("Mesure", 0, StringComparison.CurrentCultureIgnoreCase) > -1 Or
linha.indexof("Renishaw", 0, StringComparison.CurrentCultureIgnoreCase) > -1 Or
linha.indexof("Calibracao", 0, StringComparison.CurrentCultureIgnoreCase) > -1
Or linha.indexof("Palchange", 0, StringComparison.CurrentCultureIgnoreCase) > -1
Or linha.indexof("limpeza", 0, StringComparison.CurrentCultureIgnoreCase) > -1
Or linha.indexof("Medir", 0, StringComparison.CurrentCultureIgnoreCase) > -1)
Then
Try
'Alterar o estado da maquina para Setup
estado = "Setup"
'definir a hora de inicio do estado de setup
inicio_setup_hora = Cdate(Mid((list_logfile(i - 1).ToString),
InStr((list_logfile(i - 1).ToString), "Info:") + 44, InStr((list_logfile(i -
1).ToString), "Info:") + 52)).TimeOfDay
inicio_setup_dia = Cdate(Mid((list_logfile(i - 1).ToString),
InStr((list_logfile(i - 1).ToString), "Info:") + 57, InStr((list_logfile(i -
1).ToString), "Info:") + 68)).date
inicio_setup = inicio_setup_dia + inicio_setup_hora
SQL = "update Var_heid set Estado = '" & estado & "', Inicio_setup = '" &
Cdate(inicio_setup).ToString("yyyy-MM-dd HH:mm:ss") & "', Inicio_corte = NULL
where maquina = '" & maq_log & "'; update Var_backup_heid set Estado = '" &
estado & "', Inicio_setup = '" & Cdate(inicio_setup).ToString("yyyy-MM-dd
HH:mm:ss") & "', Inicio_corte = NULL where maquina = '" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.80: " & ex.Message)
End Try
'Perceber se e um programa generico ou se e um programa individual

```



```

ElseIf prog_nc = "MdI" Or prog_nc = "NC" Or prog_nc = "nc118" Or prog_nc =
"Troca-50%" Or prog_nc = "Troca-100%" Then
Try
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
prog_nc_temp = dt.Tables("Tracker_Variaveis").Rows(0).Item("Prog_nc_temp")
End If
prog_nc = prog_nc + " - " + prog_nc_temp
SQL = "update Var_heid set Programa_nc = '" & prog_nc & "' where Maquina = '" &
maq_log & "'; update Var_backup_heid set Programa_nc = '" & prog_nc & "' where
Maquina = '" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.90: " & ex.Message)
End Try
'Caso seja um programa nc normal
Else
'Escrever para a tabela de variaveis
Try
SQL = "update Var_heid set Programa_nc = '" & prog_nc & "' where Maquina = '" &
maq_log & "'; update Var_backup_heid set Programa_nc = '" & prog_nc & "' where
Maquina = '" & maq_log & "'"
Tracker_escrever()

Catch ex As Exception
MsgBox("Erro 1.20.100: " & ex.Message)
End Try
End If
End If
Catch ex As Exception
If Not ex.Message = ("Index was out of range. Must be non-negative and less than
the size of the collection." & vbCrLf & "Parameter name: index") Then
MsgBox("Erro 1.20.110: " & ex.Message)
End If
End Try
'Fim de corte
If linha.CONTAINS("Stib: OFF") Then
Try
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
prog_nc = dt.Tables("Tracker_Variaveis").Rows(0).Item("Programa_nc")
inicio_ferramenta =
dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ferramenta")
End If

```

```

hora_fim = Convert.ToDateTime(Mid(linha, InStr(linha, "OFF") + 38, (InStr(linha,
"OFF") + 46) - (InStr(linha, "OFF") + 38))).TimeOfDay
data_fim = Convert.ToDateTime(Mid(linha, (InStr(linha, "OFF") + 51),
(InStr(linha, "OFF") + 62) - (InStr(linha, "OFF") + 51))).date
time_fim = data_fim + hora_fim
If IsDBNull(inicio_corte) = False Then
'Fazer as contas de tempo
Try
If IsDBNull(inicio_setup) = False Then
Try
'Acabar com o setup antes de acabar com o corte
estado = "Setup"
hora_fim_ferramenta = Convert.ToDateTime(Mid(linha, (InStr(linha, "OFF") + 38),
(InStr(linha, "OFF") + 46) - (InStr(linha, "OFF") + 38))).TimeOfDay
data_fim_ferramenta = Convert.ToDateTime(Mid(linha, (InStr(linha, "OFF") + 51),
(InStr(linha, "OFF") + 62) - (InStr(linha, "OFF") + 51))).date
fim_ferramenta = data_fim_ferramenta + hora_fim_ferramenta
'Escrever
SQL = "update Var_heid set Fim_ferramenta = '" &
Cdate(fim_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "' where Maquina = '" &
maq_log & "'; update Var_backup_heid set Fim_ferramenta = '" &
Cdate(fim_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "' where Maquina = '" &
maq_log & "'"
Tracker_escrever()
If IsDBNull(inicio_ferramenta) = False And IsDBNull(fim_ferramenta) = False Then
tempo_setup = datediff(dateInterval.Second, inicio_ferramenta, fim_ferramenta)
Try
'Escrever os dados para as tabelas de log_file , maquinas e ferramentas
estado = "Setup"
SQL = "INSERT INTO Log_file_heid ([Maquina], [data_entrada], [data_saida],
[Tempo_corte], [Estado]) values ('" & maq_log & "', '" &
Cdate(inicio_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "', '" &
Cdate(fim_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "', '" & tempo_setup &
"', '" & estado & "'"
Tracker_escrever()
data_dif = datediff(dateInterval.Second, Cdate(inicio_ferramenta).date,
Cdate(fim_ferramenta).date)
inicio = inicio_ferramenta
fim = fim_ferramenta
divisao_dias()
SQL = "UPDATE Var_heid set Inicio_setup = NULL, Inicio_ferramenta = NULL,
Fim_ferramenta = NULL, Troca_ferramenta = NULL, nr_ferramenta = NULL WHERE
Maquina ='" & maq_log & "'; UPDATE Var_backup_heid set Inicio_setup = NULL,
Inicio_ferramenta = NULL, Fim_ferramenta = NULL, Troca_ferramenta = NULL,
nr_ferramenta = NULL WHERE Maquina ='" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.120: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 1.20.130: " & ex.Message)
End Try
End If
If IsDBNull(inicio_corte) = False Then
Try
estado = "Corte"
tempo_corte = datediff(dateInterval.Second, inicio_corte, time_fim)
'Escrever para bd
SQL = "INSERT INTO Log_file_heid([Programa_nc], [Maquina], [data_entrada],
[data_saida], [Tempo_corte], [Estado]) values('" & prog_nc & "', '" & maq_log &

```

```

", ' " & Cdate(inicio_corte).ToString("yyyy/MM/dd HH:mm:ss") & ", ' " &
Cdate(time_fim).ToString("yyyy/MM/dd HH:mm:ss") & ", ' " & tempo_corte & ", ' "
& estado & "'")
Tracker_escrever()
'Escrever na base de dados da maquina
data_dif = datediff(dateInterval.Second, Cdate(inicio_corte).date,
Cdate(time_fim).date)
inicio = inicio_corte
fim = time_fim
divisao_dias()
'Escrever o programa nc
'Verificar se o programa nc ja exista na tabela
If Not tempo_corte = 0 Then
SQL = "SELECT * from Programa_nc_heid where Programa_nc=' " & prog_nc & "' AND
Maquina = ' " & maq_log & "' "
Tracker_programa_nc()
'Caso nao exista o item na tabela
If dt.Tables("Tracker_Programa_nc").Rows.Count = 0 Then
SQL = "INSERT INTO Programa_nc_heid ([Programa_nc], [Iniciado], [Finalizado],
[Segundo_total], [Maquina]) values (' " & prog_nc & "', ' " &
Cdate(inicio_corte).ToString("yyyy/MM/dd HH:mm:ss") & ", ' " &
Cdate(time_fim).ToString("yyyy/MM/dd HH:mm:ss") & ", ' " & tempo_corte & ", ' "
& maq_log & "' "
Tracker_escrever()
Else
'Adicionar o tempo de corte ao tempo ja existente
tempo_total = dt.Tables("Tracker_Programa_nc").Rows(0).Item("Segundo_total")
tempo_total = tempo_total + tempo_corte
SQL = "Update Programa_nc_heid set Maquina = ' " & maq_log & "', Segundo_total =
' " & tempo_total & "', Finalizado = ' " & Cdate(time_fim).ToString("yyyy/MM/dd
HH:mm:ss") & "' where Programa_nc = ' " & prog_nc & "' "
Tracker_escrever()
End If
End If
'Apagar as variaveis
SQL = "update Var_heid set Ult_data = ' " & Cdate(time_fim).ToString("yyyy-MM-dd
HH:mm:ss") & "', Inicio_corte = NULL, Inicio_setup = NULL, Inicio_ligada = NULL,
Inicio_ferramenta = NULL, Fim_ferramenta = NULL, Troca_ferramenta = NULL,
nr_ferramenta = NULL, Estado = ' " & estado & "', Programa_nc = NULL where
Maquina = ' " & maq_log & "'; update Var_backup_heid set Ult_data = ' " &
Cdate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "', Inicio_corte = NULL,
Inicio_setup = NULL, Inicio_ligada = NULL, Inicio_ferramenta = NULL,
Fim_ferramenta = NULL, Troca_ferramenta = NULL, nr_ferramenta = NULL, Estado =
' " & estado & "', Programa_nc = NULL where Maquina = ' " & maq_log & "' "
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.140: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 1.20.150: " & ex.Message)
End Try
ElseIf IsDBNull(inicio_ligada) = False Then
'Fazer os mesmos calculos mas com o estado = Ligada
estado = "Ligada"
tempo_corte = datediff(dateInterval.Second, inicio_ligada, time_fim)
'Escrever para o Logfile
SQL = "INSERT INTO Log_file_heid([Programa_nc], [Maquina], [data_entrada],
[data_saida], [Tempo_corte], [Estado]) values(' " & prog_nc & "', ' " & maq_log &
"', ' " & Cdate(inicio_ligada).ToString("yyyy/MM/dd HH:mm:ss") & ", ' " &

```

```

Cdate(time_fim).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & tempo_corte & "', '"
& estado & "')"
Tracker_escrever()
Try
'Escrever na base de dados da maquina
data_dif = datediff(dateInterval.Second, Cdate(inicio_ligada).date,
Cdate(time_fim).date)
inicio = inicio_ligada
fim = time_fim
divisao_dias()
'Verificar se ja existe o programa NC na base de dados
'Atualizar as variaveis
SQL = "update Var_heid set Inicio_erro = NULL , Ult_data = '" &
Cdate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "', Inicio_corte = NULL,
Inicio_setup = NULL, Inicio_ligada = NULL, Inicio_ferramenta = NULL,
Fim_ferramenta = NULL, Troca_ferramenta = NULL, nr_ferramenta = NULL,
Programa_nc = NULL, Estado = '" & estado & "' where Maquina = '" & maq_log & "'";
update Var_backup_heid set Inicio_erro = NULL , Ult_data = '" &
Cdate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "', Inicio_corte = NULL,
Inicio_setup = NULL, Inicio_ligada = NULL, Inicio_ferramenta = NULL,
Fim_ferramenta = NULL, Troca_ferramenta = NULL, nr_ferramenta = NULL,
Programa_nc = NULL, Estado = '" & estado & "' where Maquina = '" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.160: " & ex.Message)
End Try
ElseIf IsDBNull(inicio_setup) = False Then
'Fazer os calculos para o estado de setup visto que o programa que correu foi de
setup e nao de corte
estado = "Setup"
tempo_corte = datediff(dateInterval.Second, inicio_setup, time_fim)
'Escrever para o Logfile
SQL = "INSERT INTO Log_file_heid([Programa_nc], [Maquina], [data_entrada],
[data_saida], [Tempo_corte], [Estado]) values('" & prog_nc & "', '" & maq_log &
"', '" & Cdate(inicio_setup).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(time_fim).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & tempo_corte & "', '"
& estado & "')"
Tracker_escrever()
Try
'Escrever na base de dados da maquina
data_dif = datediff(dateInterval.Second, Cdate(inicio_setup).date,
Cdate(time_fim).date)
inicio = inicio_setup
fim = time_fim
divisao_dias()
'Verificar se ja existe o programa NC na base de dados
'Atualizar as variaveis
SQL = "update Var_heid set Inicio_erro = NULL , Ult_data = '" &
Cdate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "', Inicio_corte = NULL,
Inicio_ligada = NULL, inicio_setup = NULL, Inicio_ferramenta = NULL,
Fim_ferramenta = NULL, Troca_ferramenta = NULL, nr_ferramenta = NULL,
Programa_nc = NULL, Estado = '" & estado & "' where Maquina = '" & maq_log & "'";
update Var_backup_heid set Inicio_erro = NULL , Ult_data = '" &
Cdate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "', Inicio_corte = NULL,
Inicio_ligada = NULL, inicio_setup = NULL, Inicio_ferramenta = NULL,
Fim_ferramenta = NULL, Troca_ferramenta = NULL, nr_ferramenta = NULL,
Programa_nc = NULL, Estado = '" & estado & "' where Maquina = '" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.170: " & ex.Message)
End Try

```

```

End If
Catch ex As Exception
MsgBox("Erro 1.20.180: " & ex.Message)
End Try
End If
'Ligada
If linha.Contains("Stib: BLINK") Then
Try
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
prog_nc = dt.Tables("Tracker_Variaveis").Rows(0).Item("Programa_nc")
inicio_ferramenta =
dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ferramenta")
End If
Try
If IsDBNull(inicio_setup) = False Then
'Acabar com o setup antes de acabar com o corte
estado = "Setup"
hora_fim_ferramenta = Convert.ToDateTime(Mid(linha, InStr(linha, "BLINK") + 38,
(InStr(linha, "BLINK") + 46) - (InStr(linha, "BLINK") + 38))).TimeOfDay
data_fim_ferramenta = Convert.ToDateTime(Mid(linha, (InStr(linha, "BLINK") +
51), (InStr(linha, "BLINK") + 62) - (InStr(linha, "BLINK") + 51))).date
fim_ferramenta = data_fim_ferramenta + hora_fim_ferramenta
'Escrever
SQL = "update Var_heid set Fim_ferramenta = '" &
Cdate(fim_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "' where Maquina = '" &
maq_log & "'; update Var_backup_heid set Fim_ferramenta = '" &
Cdate(fim_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "' where Maquina = '" &
maq_log & "'"
Tracker_escrever()
If IsDBNull(inicio_ferramenta) = False And IsDBNull(fim_ferramenta) = False Then
tempo_setup = datediff(dateInterval.Second, inicio_ferramenta, fim_ferramenta)
Try
'Escrever os dados para as tabelas de log_file , maquinas e ferramentas
estado = "Setup"
SQL = "INSERT INTO Log_File_heid ([Programa_nc], [Maquina], [data_entrada],
[data_saida], [Tempo_corte], [Estado]) values ('" & prog_nc & "', '" & maq_log &
"', '" & Cdate(inicio_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "', '" &
Cdate(fim_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "', '" & tempo_setup &
"', '" & estado & "'"
Tracker_escrever()
data_dif = datediff(dateInterval.Second, Cdate(inicio_ferramenta).date,
Cdate(fim_ferramenta).date)
inicio = inicio_ferramenta
fim = fim_ferramenta
divisao_dias()
SQL = "UPdATE Var_heid set Inicio_setup = NULL, Inicio_ferramenta = NULL,
Fim_ferramenta = NULL, Troca_ferramenta = NULL, nr_ferramenta = NULL WHERE
Maquina ='" & maq_log & "'; UPdATE Var_backup_heid set Inicio_setup = NULL,
Inicio_ferramenta = NULL, Fim_ferramenta = NULL, Troca_ferramenta = NULL,
nr_ferramenta = NULL WHERE Maquina ='" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception

```

```

MsgBox("Erro 1.20.190: " & ex.Message)
End Try
End If
End If
Catch ex As Exception
MsgBox("Erro 1.20.200: " & ex.Message)
End Try
If IsDBNull(inicio_corte) = False Then
If linha.CONTAINS("Stib: BLINK") Then
hora_fim = Convert.ToDateTime(Mid(linha, InStr(linha, "BLINK") + 38,
(InStr(linha, "BLINK") + 46) - (InStr(linha, "BLINK") + 38))).TimeOfDay
data_fim = Convert.ToDateTime(Mid(linha, (InStr(linha, "BLINK") + 51),
(InStr(linha, "BLINK") + 62) - (InStr(linha, "BLINK") + 51))).date
End If
time_fim = data_fim + hora_fim
'Fazer as contas de tempo
Try
If IsDBNull(inicio_corte) = False Then
estado = "Corte"
tempo_corte = datediff(dateInterval.Second, inicio_corte, time_fim)
'Escrever para o Logfile
SQL = "INSERT INTO Log_file_heid([Programa_nc], [Maquina], [data_entrada],
[data_saida], [Tempo_corte], [Estado]) values('" & prog_nc & "', '" & maq_log &
"', '" & Cdate(inicio_corte).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(time_fim).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & tempo_corte & "', '"
& estado & "')"
Tracker_escrever()
'Escrever na base de dados da maquina
data_dif = datediff(dateInterval.Second, Cdate(inicio_corte).date,
Cdate(time_fim).date)
inicio = inicio_corte
fim = time_fim
divisao_dias()
'Escrever o programa nc
'Verificar se o programa nc ja exista na tabela
Try
If Not tempo_corte = 0 Then
SQL = "SELECT * from Programa_nc_heid where Programa_nc='" & prog_nc & "' AND
Maquina = '" & maq_log & "'"
Tracker_programa_nc()
'Caso nao exista o item na tabela
If dt.Tables("Tracker_Programa_nc").Rows.Count = 0 Then
SQL = "INSERT INTO Programa_nc_heid ([Programa_nc], [Iniciado], [Finalizado],
[Segundo_total], [Maquina]) values ('" & prog_nc & "', '" &
Cdate(inicio_corte).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(time_fim).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & tempo_corte & "', '"
& maq_log & "')"
Tracker_escrever()
Else
'Adicionar o tempo de corte ao tempo ja existente
tempo_total = dt.Tables("Tracker_Programa_nc").Rows(0).Item("Segundo_total")
tempo_total = tempo_total + tempo_corte
SQL = "Update Programa_nc_heid set Maquina = '" & maq_log & "', Segundo_total =
'" & tempo_total & "', Finalizado = '" & Cdate(time_fim).ToString("yyyy/MM/dd
HH:mm:ss") & "'" where Programa_nc = '" & prog_nc & "' "
Tracker_escrever()
End If
End If
Catch ex As Exception
MsgBox("Erro 1.20.210: " & ex.Message)
End Try

```

```

'Apagar as variaveis
SQL = "update Var_heid set Ult_data = '" & Cdate(time_fim).ToString("yyyy-MM-dd
HH:mm:ss") & "', Inicio_corte = NULL, Inicio_setup = NULL, Inicio_erro = '" &
Cdate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "', Estado = '" & estado & "',
Programa_nc = NULL where Maquina = '" & maq_log & "'; update Var_backup_heid set
Ult_data = '" & Cdate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "',
Inicio_corte = NULL, Inicio_setup = NULL, Inicio_erro = '" &
Cdate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "', Estado = '" & estado & "',
Programa_nc = NULL where Maquina = '" & maq_log & ""
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 1.20.211: " & ex.Message)
End Try
ElseIf IsDBNull(inicio_erro) = False Then
'Fazer os mesmos calculos mas com o estado = Alarme
estado = "Ligada"
tempo_corte = datediff(dateInterval.Second, inicio_erro, time_fim)
'Escrever para o Logfile
SQL = "INSERT INTO Log_file_heid([Programa_nc], [Maquina], [data_entrada],
[data_saida], [Tempo_corte], [Estado]) values('" & prog_nc & "', '" & maq_log &
"', '" & Cdate(inicio_erro).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
Cdate(time_fim).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & tempo_corte & "', '"
& estado & """)
Tracker_escrever()
Try
'Escrever na base de dados da maquina
data_dif = datediff(dateInterval.Second, Cdate(inicio_erro).date,
Cdate(time_fim).date)
inicio = inicio_erro
fim = time_fim
divisao_dias()
'Verificar se ja existe o programa NC na base de dados
'Atualizar as variaveis
SQL = "update Var_heid set Inicio_erro = NULL , Ult_data = '" &
Cdate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "', Inicio_corte = NULL,
Inicio_setup = NULL, Inicio_ligada = NULL, Programa_nc = NULL, Estado = '" &
estado & "' where Maquina = '" & maq_log & "'; update Var_backup_heid set
Inicio_erro = NULL , Ult_data = '" & Cdate(time_fim).ToString("yyyy-MM-dd
HH:mm:ss") & "', Inicio_corte = NULL, Inicio_setup = NULL, Inicio_ligada = NULL,
Programa_nc = NULL, Estado = '" & estado & "' where Maquina = '" & maq_log & ""
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.220: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 1.20.230: " & ex.Message)
End Try
End If
'SETUP de FERRAMENTA
'Troca Automatica de ferramenta (funciona bem para as dMG)
If linha.CONTAINS("Info:      GEO      PLACEWRITE") Then
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid where Maquina = '" & maq_log & ""
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")

```

```

inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
prog_nc = dt.Tables("Tracker_Variaveis").Rows(0).Item("Programa_nc")
inicio_ferramenta =
dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ferramenta")
End If
'Verificar se esta dentro do periodo de corte
Try
If IsDBNull(inicio_corte) = False Then
Try
If IsDBNull(inicio_ferramenta) = True Then
'Ler a hora a que comeca a mudanca de ferramenta
hora_inicio_ferramenta = Convert.ToDateTime(Mid(linha, InStr(linha,
"PLACEWRITE") + 23, (InStr(linha, "PLACEWRITE") + 31) - (InStr(linha,
"PLACEWRITE") + 23))).TimeOfDay
data_inicio_ferramenta = Convert.ToDateTime(Mid(linha, InStr(linha,
"PLACEWRITE") + 32, (InStr(linha, "PLACEWRITE") + 47) - (InStr(linha,
"PLACEWRITE") + 32))).date
inicio_ferramenta = data_inicio_ferramenta + hora_inicio_ferramenta
'Escrever na Bd a hora de inicio de troca de ferramenta
SQL = "update Var_heid set Inicio_ferramenta = '" &
Cdate(inicio_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "' where Maquina = '"
& maq_log & "'; update Var_backup_heid set Inicio_ferramenta = '" &
Cdate(inicio_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "' where Maquina = '"
& maq_log & "' "
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 1.20.250: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 1.20.260: " & ex.Message)
End Try
End If
If linha.contains(" Tool ") And linha.CONTAINS(": place ") And linha.CONTAINS("
written") Then
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
prog_nc = dt.Tables("Tracker_Variaveis").Rows(0).Item("Programa_nc")
inicio_ferramenta =
dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ferramenta")
End If
'Verificar se esta dentro do periodo de corte
Try
If IsDBNull(inicio_corte) = False Then
Try
'Garantir que se trata de uma troca de ferramenta
'Escrever na Bd a variavel binaria de troca de ferramenta
SQL = "update Var_heid set Troca_ferramenta = 1 where Maquina = '" & maq_log &
"'"
Tracker_escrever()

```



```

SQL = "update Var_backup_heid set Troca_ferramenta = 1 where Maquina = '" &
maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.270: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 1.20.280: " & ex.Message)
End Try
End If
If linha.contains("Info: CTRL Kinematik") Then
Try
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
prog_nc = dt.Tables("Tracker_Variaveis").Rows(0).Item("Programa_nc")
inicio_ferramenta =
dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ferramenta")
End If
If IsDBNull(inicio_corte) = False Then
Try
If IsDBNull(inicio_ferramenta) = False Then
'obter a hora da troca final da ferramenta
hora_fim_ferramenta = Convert.ToDateTime(Mid(linha, InStr(linha, "Kinematik") +
23, (InStr(linha, "Kinematik") + 31) - (InStr(linha, "Kinematik") +
23))).TimeOfDay
data_fim_ferramenta = Convert.ToDateTime(Mid(linha, InStr(linha, "Kinematik") +
32, (InStr(linha, "Kinematik") + 47) - (InStr(linha, "Kinematik") + 32))).date
fim_ferramenta = data_fim_ferramenta + hora_fim_ferramenta
'Escrever
SQL = "update Var_heid set Fim_ferramenta = '" &
Cdate(fim_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "' where Maquina = '" &
maq_log & "'; update Var_backup_heid set Fim_ferramenta = '" &
Cdate(fim_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "' where Maquina = '" &
maq_log & "'"
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 1.20.290: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 1.20.300: " & ex.Message)
End Try
End If
If linha.contains(" tool nr ") Then
Try
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")

```

```

inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
prog_nc = dt.Tables("Tracker_Variaveis").Rows(0).Item("Programa_nc")
inicio_ferramenta =
dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ferramenta")
fim_ferramenta = dt.Tables("Tracker_Variaveis").Rows(0).Item("Fim_ferramenta")
End If
If IsDBNull(inicio_corte) = False Then
Try
'Perceber se se trata de uma troca de ferramentas direta ou nao
ferramenta = (Mid(linha, InStr(linha, "tool nr") + 8, (InStr(linha, "R=") - 1) -
(InStr(linha, "tool nr") + 8)))
If ferramenta = 0 Then
'Tem de ler outra vez porque houve uma troca de ferramenta por carrocel
SQL = "update Var_heid set nr_ferramenta = '" & ferramenta & "' where Maquina =
'" & maq_log & "'; update Var_backup_heid set nr_ferramenta = '" & ferramenta &
"' where Maquina = '" & maq_log & "'"
Tracker_escrever()
Else
'Caso a ferramenta seja diferente de 0, fazer a diferenca entre a data de inicio
e a data de fim
'Ler todas as variaveis dessa maquina
If IsDBNull(inicio_ferramenta) = False And IsDBNull(fim_ferramenta) = False Then
tempo_setup = datediff(dateInterval.Second, inicio_ferramenta, fim_ferramenta)
Try
'Escrever os dados para as tabelas de log_file , maquinas e ferramentas
estado = "Setup"
SQL = "INSERT INTO Log_File_heid ([Maquina], [data_entrada], [data_saida],
[Tempo_corte], [Estado]) values ('" & maq_log & "', '" &
Cdate(inicio_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "', '" &
Cdate(fim_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "', '" & tempo_setup &
"', '" & estado & "'"
Tracker_escrever()
data_dif = datediff(dateInterval.Second, Cdate(inicio_ferramenta).date,
Cdate(fim_ferramenta).date)
inicio = inicio_ferramenta
fim = fim_ferramenta
divisao_dias()
SQL = "UPdATE Var_heid set Inicio_ferramenta = NULL, Fim_ferramenta = NULL,
Troca_ferramenta = NULL, nr_ferramenta = NULL WHERE Maquina ='" & maq_log & "';
UPdATE Var_backup_heid set Inicio_ferramenta = NULL, Fim_ferramenta = NULL,
Troca_ferramenta = NULL, nr_ferramenta = NULL WHERE Maquina ='" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.310: " & ex.Message)
End Try
End If
End If
Catch ex As Exception
MsgBox("Erro 1.20.320: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 1.20.330: " & ex.Message)
End Try
End If
If linha.contains("TNC:\Laser\Medir_fresa.h") Then
Try
'obter o tempo de inicio do corte e utilizar como tempo inicial para o setup

```

```

'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
prog_nc = dt.Tables("Tracker_Variaveis").Rows(0).Item("Programa_nc")
End If
If IsDBNull(inicio_corte) = False Then
SQL = "update Var_heid set Inicio_corte = NULL, Inicio_ferramenta = '" &
Cdate(inicio_corte).ToString("yyyy-MM-dd HH:mm:ss") & "' where Maquina = '" &
maq_log & "'; update Var_backup_heid set Inicio_corte = NULL, Inicio_ferramenta
= '" & Cdate(inicio_corte).ToString("yyyy-MM-dd HH:mm:ss") & "' where Maquina =
'" & maq_log & "'"
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 1.20.340: " & ex.Message)
End Try
End If
'Troca manual de ferramenta (P39 / P40)
If linha.Contains("P39") Or linha.Contains("Tool unclamping") Or
linha.Contains("desaperte a ferramenta") Then
'Ler a linha anterior para saber a que horas ocorreu o erro
Try
linha_anterior = list_logfile(i - 1).ToString
If linha_anterior.Contains("Error:") Then
Try
hora_inicio_ferramenta = Convert.ToDateTime(Mid(linha_anterior,
InStr(linha_anterior, "Error:") + 44, (InStr(linha_anterior, "Error:") + 52) -
(InStr(linha_anterior, "Error:") + 23))).TimeOfDay
data_inicio_ferramenta = Convert.ToDateTime(Mid(linha_anterior,
InStr(linha_anterior, "Error:") + 57, (InStr(linha_anterior, "Error:") + 68) -
(InStr(linha_anterior, "Error:") + 57))).date
inicio_ferramenta = data_inicio_ferramenta + hora_inicio_ferramenta
'Escrever na Bd a hora de início de troca de ferramenta
SQL = "update Var_heid set Inicio_ferramenta = '" &
Cdate(inicio_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "' where Maquina = '"
& maq_log & "'; update Var_backup_heid set Inicio_ferramenta = '" &
Cdate(inicio_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "' where Maquina = '"
& maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.350: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 1.20.360: " & ex.Message)
End Try
End If
If linha.Contains("P40") Or linha.Contains("Tool clamping") Or
linha.Contains("Aperte a ferramenta") Then
'Ler a linha anterior para saber a que horas foi corrigido o erro
Try
linha_anterior = list_logfile(i - 1).ToString
If linha_anterior.Contains("ERRCLEARED") Then
'Ler todas as variaveis dessa maquina

```

```

SQL = "Select * from Var_heid where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
'diretorio_logfile =
dt.Tables("Tracker_Variaveis").Rows(0).Item("diretorio_CNC")
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
prog_nc = dt.Tables("Tracker_Variaveis").Rows(0).Item("Programa_nc")
inicio_ferramenta =
dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ferramenta")
fim_ferramenta = dt.Tables("Tracker_Variaveis").Rows(0).Item("Fim_ferramenta")
End If
'Calculo de tempo de Setup
Try
hora_fim_ferramenta = Convert.ToDateTime(Mid(linha_anterior,
InStr(linha_anterior, "ERRCLEARED") + 23, (InStr(linha_anterior, "ERRCLEARED") +
31) - (InStr(linha_anterior, "ERRCLEARED") + 23))).TimeOfDay
data_fim_ferramenta = Convert.ToDateTime(Mid(linha_anterior,
InStr(linha_anterior, "ERRCLEARED") + 36, (InStr(linha_anterior, "ERRCLEARED") +
47) - (InStr(linha_anterior, "ERRCLEARED") + 36))).date
fim_ferramenta = data_fim_ferramenta + hora_fim_ferramenta
If IsDBNull(inicio_ferramenta) = False And IsDBNull(fim_ferramenta) = False Then
tempo_setup = datediff(dateInterval.Second, inicio_ferramenta, fim_ferramenta)
Try
'Escrever os dados para as tabelas de log_file , maquinas e ferramentas
estado = "Setup"
SQL = "INSERT INTO Log_File_heid ([Maquina], [data_entrada], [data_saida],
[Tempo_corte], [Estado]) values ('" & maq_log & "', '" &
Cdate(inicio_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "', '" &
Cdate(fim_ferramenta).ToString("yyyy-MM-dd HH:mm:ss") & "', '" & tempo_setup &
"', '" & estado & "'"
Tracker_escrever()
data_dif = datediff(dateInterval.Second, Cdate(inicio_ferramenta).date,
Cdate(fim_ferramenta).date)
inicio = inicio_ferramenta
fim = fim_ferramenta
divisao_dias()
SQL = "UPDATE Var_heid set Inicio_ferramenta = NULL, Fim_ferramenta = NULL,
Troca_ferramenta = NULL, nr_ferramenta = NULL WHERE Maquina = '" & maq_log & "';
UPDATE Var_backup_heid set Inicio_ferramenta = NULL, Fim_ferramenta = NULL,
Troca_ferramenta = NULL, nr_ferramenta = NULL WHERE Maquina = '" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.370: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("Erro 1.20.380: " & ex.Message)
End Try
'Apagar as variaveis da ferramenta
SQL = "update Var_heid set nr_ferramenta = NULL, Inicio_ferramenta = NULL,
Fim_ferramenta = NULL, Troca_ferramenta = NULL where Maquina = '" & maq_log &
"'; update Var_backup_heid set nr_ferramenta = NULL, Inicio_ferramenta = NULL,
Fim_ferramenta = NULL, Troca_ferramenta = NULL where Maquina = '" & maq_log &
"'"
Tracker_escrever()
End If

```

```

Catch ex As Exception
MsgBox("Erro 1.20.390: " & ex.Message)
End Try
End If
'Mensagem de erro
If linha.contains("Error:") Then
Try
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid where Maquina = '" & maq_log & "'"
Tracker_var()

If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
empresa = dt.Tables("Tracker_Variaveis").Rows(0).Item("Empresa")
End If
'Ler a data / hora do erro
Try
hora_inic = Convert.ToDateTime(Mid(linha, (InStr(linha, "Error:") + 45),
(InStr(linha, "Error:") + 52) - (InStr(linha, "Error:") + 45))).TimeOfDay
data_inic = Convert.ToDateTime(Mid(linha, (InStr(linha, "Error:") + 57),
(InStr(linha, "Error:") + 68) - (InStr(linha, "Error:") + 57))).date
time_erro = data_inic + hora_inic
Catch ex As Exception
GoTo Erro_data
End Try
'Ler a linha seguinte
linha_next = list_logfile(i + 1)
'Ler o codigo de erro
linha_next = Mid(linha_next, 3)
cod_erro = Mid(linha_next, 1, InStr(linha_next, " "))
'Selecionar a ultima linha da base de dados para verificar se e necessario
escrever a linha
SQL = "SELECT TOP(10) Empresa, Maquina, Tipo, Cod_erro, Mensagem, data, Id FROM
Mensagem_erro where Maquina = '" & maq_log & "'" ORDER BY Id dESC"
Tracker_variavel()
If dt.Tables("Tracker_Variavel").Rows.Count <> 0 Then
tempo_erro_old = dt.Tables("Tracker_Variavel").Rows(0).Item("data")
End If
'Ler a mensagem de erro
mensagem_erro = Mid(linha_next, InStr(Mid(linha_next, 1), " ") + 1).ToString
If mensagem_erro.contains("") Then
mensagem_erro = Replace(mensagem_erro, "", "")
End If
'Escrever para a base de dados
If datediff(dateInterval.Second, tempo_erro_old, time_erro) > 60 Then
'Escrever o erro na tabela da maquina
SQL = "INSERT INTO Mensagem_erro([Empresa], [Maquina], [Cod_erro], [Mensagem],
[data]) values('" & empresa & "', '" & maq_log & "', '" & cod_erro & "', '" &
mensagem_erro & "', '" & Cdate(time_erro).ToString("yyyy-MM-dd HH:mm:ss") & '
)"
Tracker_escrever()
End If
Erro_data:
Catch ex As Exception
If ex.Message = "String was not recognized as a valid dateTime." Then
Else

```

```

MsgBox("Erro 1.20.400: " & ex.Message)
End If
End Try
End If
If linha.Contains("Info:") Or linha.Contains("Key:") Or linha.Contains("Error:")
Then
Try
'Ler a data / hora do erro
Try
If linha.Contains("Info:") Then
hora_inic = Convert.ToDateTime(Mid(linha, (InStr(linha, "Info:") + 44),
(InStr(linha, "Info:") + 52) - (InStr(linha, "Info:") + 44))).TimeOfDay
data_inic = Convert.ToDateTime(Mid(linha, (InStr(linha, "Info:") + 57),
(InStr(linha, "Info:") + 68) - (InStr(linha, "Info:") + 57))).date
time_info = data_inic + hora_inic
ElseIf linha.Contains("Error:") Then
hora_inic = Convert.ToDateTime(Mid(linha, (InStr(linha, "Error:") + 44),
(InStr(linha, "Error:") + 52) - (InStr(linha, "Error:") + 44))).TimeOfDay
data_inic = Convert.ToDateTime(Mid(linha, (InStr(linha, "Error:") + 57),
(InStr(linha, "Error:") + 68) - (InStr(linha, "Error:") + 57))).date
time_info = data_inic + hora_inic
ElseIf linha.Contains("Key:") Then
hora_inic = Convert.ToDateTime(Mid(linha, (InStr(linha, "Key:") + 44),
(InStr(linha, "Key:") + 52) - (InStr(linha, "Key:") + 44))).TimeOfDay
data_inic = Convert.ToDateTime(Mid(linha, (InStr(linha, "Key:") + 57),
(InStr(linha, "Key:") + 68) - (InStr(linha, "Key:") + 57))).date
time_info = data_inic + hora_inic
End If
Catch ex As Exception
If ex.Message.ToString = "String was not recognized as a valid dateTime." Then
GoTo NextLine
Else
MsgBox("Erro 1.20.410: " & ex.Message)
End If
End Try
'Comparar data atual com a Ult_data
'Verificar se existe dados de ult_data
If IsDBNull(UltimaLinha) = True Then
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
Try
UltimaLinha = dt.Tables("Tracker_Variaveis").Rows(0).Item("UltimaLinha")
Catch ex As Exception
UltimaLinha = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
End Try
End If
End If
Try
If datediff(dateInterval.Second, UltimaLinha, time_info) >= 0 Then
'Os tempos estao a bater certo, nao fazer nada
UltimaLinha = time_info
ElseIf datediff(dateInterval.Second, UltimaLinha, time_info) < 0 Then
If nr_ficheiro <= 10 Or (Cdate(ult_registro).date.ToString("yyyy-MM-dd") <>
Cdate(UltimaLinha).date.ToString("yyyy-MM-dd") And ult_maq = maq_log) Then
nr_ficheiro = nr_ficheiro + 1
ult_registro = UltimaLinha
ult_maq = maq_log
'Guardar o logfile da maquina

```

```

dim FileName = "d:\Tracker\Log_file\Heidenhain\Save\" & maq_log & "\" &
Cdate(UltimaLinha).date.ToString("yyyy-MM-dd") & "\" & Now.ToString("HHmmss") &
"_" & i.ToString
dim FileType = ".log"
dim NewCopy = FileName & FileType
dim newfolder = "d:\Tracker\Log_file\Heidenhain\Save\" & maq_log & "\" &
Cdate(UltimaLinha).date.ToString("yyyy-MM-dd") & "\"
dim dirAux = 1
diretorio:
If System.IO.File.Exists(NewCopy) = True Then
NewCopy = FileName & "_" & dirAux & FileType
dirAux = dirAux + 1
GoTo diretorio
End If
If System.IO.File.Exists(diretorio_logfile) = True Then
If System.IO.directory.Exists(newfolder) = False Then
System.IO.directory.Createdirectory(newfolder)
End If
System.IO.File.Copy(diretorio_logfile, NewCopy)
End If
End If
End If
Catch ex As Exception
MsgBox("Erro 1.20.420" & ex.Message)
End Try
Catch ex As Exception
MsgBox("Erro 1.20.400: " & ex.Message)
End Try
End If
Else
GoTo tempo_ciclo
End If
NextLine:
Next'Proxima maquina
tempo_ciclo:
'Escrever data da linha nas variaveis
'Escrever o erro na tabela da maquina
If IsDBNull(time_info) = True Then
time_info = ult_data
End If
SQL = "UPDATE Var_heid set UltimaLinha = '" & Cdate(time_info).ToString("yyyy-
MM-dd HH:mm:ss") & "' where Maquina = '" & maq_log & "'; UPDATE Var_backup_heid
Set UltimaLinha = '" & Cdate(time_info).ToString("yyyy-MM-dd HH:mm:ss") &
"'where Maquina = '" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20: " & ex.Message)
End Try
End Sub

'Sub rotinas de apoio ao automatismo
'nao foram incluidas no anexo por poupanca de espaco
'estao explicadas no corpo do texto e tem a mesma logica que as sub rotinas
apresentadas no automatismo fidia
End Class

```

Anexo C - Automatismo de recolha de informação em comandos CNC TNC 640 Heidenhain

```

Public Class Tracker_Heidenhain_tnc640_V8
'declaracao de variaveis

```

```

'maquinas sao declaradas da mesma forma que os automatismos anteriores

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
reiniciar:
Try
'ciclo para cada maquina na lista de maquinas do automatismo
For Each maq_log In list_maquina
If File.Exists(diretorio_logfile) Then
Try
File.Delete(diretorio_logfile)
Thread.Sleep(2000)
Catch ex As Exception
GoTo Apagar
End Try
End If
Try
If IsNothing(list_logfile) = False Then
list_logfile.clear()
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid_640 where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
IP = dt.Tables("Tracker_Variaveis").Rows(0).Item("IP")
ult_edicao = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_edicao")
End If
time_inic = Nothing
'copiar ficheiro .log
If IsDBNull(ult_data) = True Then
procID = Shell("cmd /k cd H:\ & D: & cd D:\Tracker\Log_file\Heidenhain &
""C:\Program Files (x86)\HEIDENHAIN\TNCremo\tnccmd.exe"" " & IP & " log
LogFile_heidenhain_640 20.11.2017 ", AppWinStyle.Hide, True, 6000)
pr = Process.GetProcessById(procID)
Thread.Sleep(2000)
pr.Kill()
Thread.Sleep(2000)
Else
procID = Shell("cmd /k cd H:\ & D: & cd D:\Tracker\Log_file\Heidenhain &
""C:\Program Files (x86)\HEIDENHAIN\TNCremo\tnccmd.exe"" " & IP & " log
LogFile_heidenhain_640 " & CDate(ult_data).ToString("dd.MM.yyyy;HH:mm:ss") & "
", AppWinStyle.Hide, True, 15000)
pr = Process.GetProcessById(procID)
Thread.Sleep(2000)
pr.Kill()
Thread.Sleep(2000)
End If
'Caso nao exista ficheiro de log
If File.Exists(diretorio_logfile) = True Then
While CDate(File.GetLastWriteTime(diretorio_logfile)).Date.ToString = "01-01-
1601 00:00:00"
MsgBox("l data")
End While

```



```

End If
' Caso o ficheiro esteja a ser copiado
While DateDiff(DateInterval.Second, File.GetLastWriteTime(diretorio_logfile),
Now) < 3
If IsDBNull(ult_edicao) = False Then
If File.GetLastWriteTime(diretorio_logfile) = CDate(ult_edicao) Then
GoTo nextline
End If
End If
End While
' Caso nao exista ficheiro
If File.Exists(diretorio_logfile) = False Then
SQL = "update Var_heid_640 set Ligacao = 0 where Maquina = '" & maq_log & "';
update Var_heid_640_backup set Ligacao = 0 where Maquina = '" & maq_log & "'";
Tracker_escrever()
GoTo nextline
Else
SQL = "update Var_heid_640 set Ligacao = 1 where Maquina = '" & maq_log & "';
update Var_heid_640_backup set Ligacao = 1 where Maquina = '" & maq_log & "'";
Tracker_escrever()
End If
Dim temp1
temp1 = Now
Readagain:
Try
' passar o ficheiro para list
list_logfile = File.ReadLines(diretorio_logfile).ToList
Catch ex As Exception
If DateDiff(DateInterval.Minute, temp1, Now) < 5 Then
GoTo Readagain
Else
MsgBox(ex.Message)
End If
End Try
linha_count = list_logfile.Count
' apagar da List o que ja foi lido antes
Try
If linha_atual >= 0 Then
linha_atual = 0
' Fazer update
SQL = "update Var_heid_640 set Linha = '" & linha_count & "', Linha_atual = '" &
linha_atual & "' where Maquina = '" & maq_log & "'; update Var_heid_640_backup
set Linha = '" & linha_count & "', Linha_atual = '" & linha_atual & "' where
Maquina = '" & maq_log & "'";
Tracker_escrever()
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
linhas()
Apagar:
nextline:
If File.Exists(diretorio_logfile) Then
Try
File.Delete(diretorio_logfile)
Thread.Sleep(2000)
Catch ex As Exception
GoTo Apagar
End Try
End If
Next

```

```

GoTo reiniciar
Catch ex As Exception
MsgBox(ex.Message)
End Try
End Sub
'Rotina de analise linha a linha dos ficheiros .log dos equipamentos
Sub linhas()
tempo_inic = Now
Try
For i = 0 To list_logfile.count - 1
tempo_atual = Now
'a analise estende se no maximo durante 60mins
If DateDiff(DateInterval.Minute, tempo_inic, tempo_atual) < 60 Then
linha = list_logfile(i)
linha_atual = linha_atual + 1
'Escrever a nova linha na base de dados Var
SQL = "update Var_heid_640 set Linha_atual = '" & linha_atual & "' where Maquina
= '" & maq_log & "'; update Var_heid_640_backup set Linha_atual = '" &
linha_atual & "' where Maquina = '" & maq_log & "'"
Tracker_escrever()
Try
Catch ex As Exception
MsgBox(ex.Message)
End Try
'Inicio Corte - NC_START
If linha.contains("state:=NC_START") Then
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid_640 where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
'diretorio_logfile =
dt.Tables("Tracker_Variaveis").Rows(0).Item("Diretorio_CNC")
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
prog_end = dt.Tables("Tracker_Variaveis").Rows(0).Item("Prog_end")
End If
'Caso seja a primeira linha de NC_START
If IsDBNull(prog_end) = False Then
'Obter o tempo de paragem entre o fim de o programa e o inicio do proximo
Try
'Obter a data da linha de NC_Start
Try
inicio_corte_hora = CDate(Mid((list_logfile(i - 1).ToString),
InStr((list_logfile(i - 1).ToString), "Info:") + 44, InStr((list_logfile(i -
1).ToString), "Info:") + 52)).TimeOfDay
inicio_corte_data = CDate(Mid((list_logfile(i - 1).ToString),
InStr((list_logfile(i - 1).ToString), "Info:") + 54, InStr((list_logfile(i -
1).ToString), "Info:") + 64)).Date
time_inic = inicio_corte_data + inicio_corte_hora
Catch ex As Exception
MsgBox(ex.Message)
End Try
'Fazer a diferenca para a data do ultimo PROGRAM_END OU PROGRAM_STOP
time_stop = DateDiff(DateInterval.Second, prog_end, time_inic)
estado = "Parada"
'Adiconar a tabela de LOG_FILE

```

```

SQL = "INSERT INTO Log_File_heid_640 ([Maquina], [Data_entrada], [Data_saida],
[Tempo_corte], [Estado]) values( '" & maq_log & "', '" &
CDate(prog_end).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
CDate(time_inic).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & time_stop & "', '" &
estado & "') "
Tracker_escrever()
'Adicionar a tabela da maquina
Try
data_dif = DateDiff(DateInterval.Second, CDate(prog_end).Date,
CDate(time_inic).Date)
inicio = prog_end
fim = time_inic
divisao_dias()
Catch ex As Exception
MsgBox(ex.Message)
End Try
Catch ex As Exception
MsgBox(ex.Message)
End Try
'Adicionar a tabela de variaveis a hora de nc_start
estado = "Corte"
SQL = "UPDATE Var_heid_640 set Prog_end = NULL, Inicio_corte = '" &
CDate(time_inic).ToString("yyyy-MM-dd HH:mm:ss") & "', Estado = '" & estado &
"', Ult_data = '" & CDate(time_inic).ToString("yyyy-MM-dd HH:mm:ss") & "' where
Maquina = '" & maq_log & "'; UPDATE Var_heid_640_backup set Prog_end = NULL,
Inicio_corte = '" & CDate(time_inic).ToString("yyyy-MM-dd HH:mm:ss") & "',
Estado = '" & estado & "', Ult_data = '" & CDate(time_inic).ToString("yyyy-MM-dd
HH:mm:ss") & "' where Maquina = '" & maq_log & "'"
Tracker_escrever()
If CDate(time_inic).Date > CDate(ult_data).Date Then
Try
SQL = "SELECT * FROM '" & maq_log & "' where (DATEPART(yy, Inicio) = '" &
CDate(DateAdd(DateInterval.Day, -1, time_inic)).ToString("yyyy") & "') AND
(DATEPART(MM, Inicio)= '" & CDate(DateAdd(DateInterval.Day, -1,
time_inic)).ToString("MM") & "') AND (DATEPART(dd, Inicio)= '" &
CDate(DateAdd(DateInterval.Day, -1, time_inic)).ToString("dd") & "') "
Tracker_var()
Dim comp_corte = 0
Dim comp_parada = 0
Dim comp_setup = 0
Dim comp_ligada = 0
Dim t1_corte = 0
Dim t2_corte = 0
Dim t3_corte = 0
Dim tfds_corte = 0
Dim t1_parada = 0
Dim t2_parada = 0
Dim t3_parada = 0
Dim tfds_parada = 0
Dim t1_setup = 0
Dim t2_setup = 0
Dim t3_setup = 0
Dim tfds_setup = 0
Dim t1_ligada = 0
Dim t2_ligada = 0
Dim t3_ligada = 0
Dim tfds_ligada = 0
For j = 0 To dt.Tables("Tracker_Variaveis").Rows.Count - 1
If IsDBNull(dt.Tables("Tracker_Variaveis").Rows(j).Item("Turno_1")) = True Then
t1_add = 0
Else

```

```

t1_add = dt.Tables("Tracker_Variaveis").Rows(j).Item("Turno_1")
End If
If IsDBNull(dt.Tables("Tracker_Variaveis").Rows(j).Item("Turno_2")) = True Then
t2_add = 0
Else
t2_add = dt.Tables("Tracker_Variaveis").Rows(j).Item("Turno_2")
End If
If IsDBNull(dt.Tables("Tracker_Variaveis").Rows(j).Item("Turno_3")) = True Then
t3_add = 0
Else
t3_add = dt.Tables("Tracker_Variaveis").Rows(j).Item("Turno_3")
End If

If IsDBNull(dt.Tables("Tracker_Variaveis").Rows(j).Item("Turno_fds")) = True
Then
tfds_add = 0
Else
tfds_add = dt.Tables("Tracker_Variaveis").Rows(j).Item("Turno_fds")
End If
If dt.Tables("Tracker_Variaveis").Rows(j).Item("Estado") = "Corte" Then
comp_corte = comp_corte + dt.Tables("Tracker_Variaveis").Rows(j).Item("Tempo")
t1_corte = t1_corte + t1_add
t2_corte = t2_corte + t2_add
t3_corte = t3_corte + t3_add
tfds_corte = tfds_corte + tfds_add
ElseIf dt.Tables("Tracker_Variaveis").Rows(j).Item("Estado") = "Ligada" Then
comp_ligada = comp_ligada + dt.Tables("Tracker_Variaveis").Rows(j).Item("Tempo")
t1_ligada = t1_ligada + t1_add
t2_ligada = t2_ligada + t2_add
t3_ligada = t3_ligada + t3_add
tfds_ligada = tfds_ligada + tfds_add
ElseIf dt.Tables("Tracker_Variaveis").Rows(j).Item("Estado") = "Setup" Then
comp_setup = comp_setup + dt.Tables("Tracker_Variaveis").Rows(j).Item("Tempo")
t1_setup = t1_setup + t1_add
t2_setup = t2_setup + t2_add
t3_setup = t3_setup + t3_add
tfds_setup = tfds_setup + tfds_add
ElseIf dt.Tables("Tracker_Variaveis").Rows(j).Item("Estado") = "Parada" Then
comp_parada = comp_parada + dt.Tables("Tracker_Variaveis").Rows(j).Item("Tempo")
t1_parada = t1_parada + t1_add
t2_parada = t2_parada + t2_add
t3_parada = t3_parada + t3_add
tfds_parada = tfds_parada + tfds_add
End If
Next
If comp_corte = 0 And comp_setup = 0 And comp_ligada = 0 Then
comp_parada = 86400
t1_parada = 30599
t2_parada = 30598
t3_parada = 25200
End If
'Escrever na tabela de compilacao
SQL = "INSERT INTO Tab_comp([Data], [Maquina], [Corte_segundo], [Setup_segundo],
[Ligada_segundo], [Parada_segundo], [t1_corte], [t2_corte], [t3_corte],
[tfds_corte], [t1_parada], [t2_parada], [t3_parada], [tfds_parada], [t1_setup],
[t2_setup], [t3_setup], [tfds_setup], [t1_ligada], [t2_ligada], [t3_ligada],
[tfds_ligada]) values('" & CDate(DateAdd(DateInterval.Day, -1,
time_inic)).ToString("yyyy/MM/dd") & "', '" & maq_log & "', '" & comp_corte &
"', '" & comp_setup & "', '" & comp_ligada & "', '" & comp_parada & "', '" &
t1_corte & "', '" & t2_corte & "', '" & t3_corte & "', '" & tfds_corte & "', '"
& t1_parada & "', '" & t2_parada & "', '" & t3_parada & "', '" & tfds_parada &

```

```

", ' ' & t1_setup & '", ' ' & t2_setup & '", ' ' & t3_setup & '", ' ' & tfds_setup
& '", ' ' & t1_ligada & '", ' ' & t2_ligada & '", ' ' & t3_ligada & '", ' ' &
tfds_ligada & "')"
Tracker_escrever()
Catch ex As Exception
MsgBox(ex.Message)
End Try
End If
End If
End If
'PROGRAM_STOP
If linha.contains("state:=PROGRAM_STOP") Or linha.contains("state:=PROGRAM_END")
Then
Try
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid_640 where Maquina = ' ' & maq_log & '"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
prog_end = dt.Tables("Tracker_Variaveis").Rows(0).Item("Prog_end")
End If
'Verificar se existe hora para inicio de corte
If IsDBNull(inicio_corte) = False Then
Try
Try
'Obter a hora da linha
fim_corte_hora = CDate(Mid((list_logfile(i - 1).ToString), InStr((list_logfile(i
- 1).ToString), "Info:") + 44, InStr((list_logfile(i - 1).ToString), "Info:") +
52)).TimeOfDay
fim_corte_data = CDate(Mid((list_logfile(i - 1).ToString), InStr((list_logfile(i
- 1).ToString), "Info:") + 54, InStr((list_logfile(i - 1).ToString), "Info:") +
64)).Date
time_fim = fim_corte_data + fim_corte_hora
'Fazer a diferenca entre a linhas de inicio e atual
tempo_corte = DateDiff(DateInterval.Second, inicio_corte, time_fim)
Catch ex As Exception
MsgBox(ex.Message)
End Try
estado = "Corte"
'Escrever na tabela de log file
Try
SQL = "INSERT INTO Log_File_heid_640 ([Maquina], [Data_entrada], [Data_saida],
[Tempo_corte], [Estado]) values( ' ' & maq_log & '", ' ' &
CDate(inicio_corte).ToString("yyyy/MM/dd HH:mm:ss") & "', ' ' &
CDate(time_fim).ToString("yyyy/MM/dd HH:mm:ss") & "', ' ' & tempo_corte & "', ' '
& estado & '" ) "
Tracker_escrever()
Catch ex As Exception
MsgBox(ex.Message)
End Try
'Escrever na tabela da maquina
Try
data_dif = DateDiff(DateInterval.Second, CDate(inicio_corte).Date,
CDate(time_fim).Date)
inicio = inicio_corte
fim = time_fim

```

```

divisao_dias()
Catch ex As Exception
MsgBox(ex.Message)
End Try
'Escrever na tabela de variaveis
Try
estado = "Parada"
SQL = "UPDATE Var_heid_640 set inicio_corte = NULL, Prog_end = '" &
CDate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "', estado = '" & estado & "',
Ult_data = '" & CDate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "' where
Maquina = '" & maq_log & "'; UPDATE Var_heid_640_backup set inicio_corte = NULL,
Prog_end = '" & CDate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "', estado = '"
& estado & "', Ult_data = '" & CDate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") &
"' where Maquina = '" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox(ex.Message)
End Try
Catch ex As Exception
MsgBox(ex.Message)
End Try
Else
'Apenas escrever a hora na tabela de variaveis
Try
estado = "Parada"
Try
fim_corte_hora = CDate(Mid((list_logfile(i - 1).ToString), InStr((list_logfile(i
- 1).ToString), "Info:") + 44, InStr((list_logfile(i - 1).ToString), "Info:") +
52)).TimeOfDay
fim_corte_data = CDate(Mid((list_logfile(i - 1).ToString), InStr((list_logfile(i
- 1).ToString), "Info:") + 54, InStr((list_logfile(i - 1).ToString), "Info:") +
64)).Date
time_fim = fim_corte_data + fim_corte_hora
Catch ex As Exception
MsgBox(ex.Message)
End Try
SQL = "UPDATE Var_heid_640 set Prog_end = '" & CDate(time_fim).ToString("yyyy-
MM-dd HH:mm:ss") & "', Estado = '" & estado & "', Ult_data = '" &
CDate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "' WHERE Maquina = '" &
maq_log & "'; UPDATE Var_heid_640_backup set Prog_end = '" &
CDate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "', Estado = '" & estado & "',
Ult_data = '" & CDate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "' WHERE
Maquina = '" & maq_log & "'"
Tracker_escrever()
Catch ex As Exception
MsgBox(ex.Message)
End Try
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
If CDate(time_fim).Date > CDate(ult_data).Date Then
Try
SQL = "SELECT * FROM " & maq_log & " where (DATEPART(yy, Inicio) = '" &
CDate(DateAdd(DateInterval.Day, -1, time_fim)).ToString("yyyy") & "') AND
(DATEPART(MM, Inicio)= '" & CDate(DateAdd(DateInterval.Day, -1,
time_fim)).ToString("MM") & "') AND (DATEPART(dd, Inicio)= '" &
CDate(DateAdd(DateInterval.Day, -1, time_fim)).ToString("dd") & "') "
Tracker_var()
Dim comp_corte = 0
Dim comp_parada = 0

```

```

Dim comp_setup = 0
Dim comp_ligada = 0
For j = 0 To dt.Tables("Tracker_Variaveis").Rows.Count - 1
If dt.Tables("Tracker_Variaveis").Rows(j).Item("Estado") = "Corte" Then
comp_corte = comp_corte + dt.Tables("Tracker_Variaveis").Rows(j).Item("Tempo")
ElseIf dt.Tables("Tracker_Variaveis").Rows(j).Item("Estado") = "Ligada" Then
comp_ligada = comp_ligada + dt.Tables("Tracker_Variaveis").Rows(j).Item("Tempo")
ElseIf dt.Tables("Tracker_Variaveis").Rows(j).Item("Estado") = "Setup" Then
comp_setup = comp_setup + dt.Tables("Tracker_Variaveis").Rows(j).Item("Tempo")
ElseIf dt.Tables("Tracker_Variaveis").Rows(j).Item("Estado") = "Parada" Then
comp_parada = comp_parada + dt.Tables("Tracker_Variaveis").Rows(j).Item("Tempo")
End If
Next
If comp_corte = 0 And comp_setup = 0 And comp_ligada = 0 Then
comp_parada = 86400
End If
'Escrever na tabela de compilacao
SQL = "INSERT INTO Tab_comp([Data], [Maquina], [Corte_segundo], [Setup_segundo],
[Ligada_segundo], [Parada_segundo]) values('" & CDate(DateAdd(DateInterval.Day,
-1, time_fim)).ToString("yyyy/MM/dd") & "', '" & maq_log & "', '" & comp_corte &
"', '" & comp_setup & "', '" & comp_ligada & "', '" & comp_parada & "')"
Tracker_escrever()
Catch ex As Exception
MsgBox(ex.Message)
End Try
End If
End If
'Mensagem de erro
If linha.contains("Error: ") Then
Try
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_heid_640 where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
linha_atual = dt.Tables("Tracker_Variaveis").Rows(0).Item("Linha_atual")
inicio_corte = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_corte")
inicio_erro = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_erro")
inicio_ligada = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_ligada")
inicio_setup = dt.Tables("Tracker_Variaveis").Rows(0).Item("Inicio_setup")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
empresa = dt.Tables("Tracker_Variaveis").Rows(0).Item("Empresa")
End If
'Ler a data / hora do erro
Dim hora_inic = Convert.ToDateTime(Mid(linha, 45, 8)).TimeOfDay
Dim data_inic = Convert.ToDateTime(Mid(linha, 55, 10)).Date
time_erro = data_inic + hora_inic
'Ler a linha seguinte
linha_next = list_logfile(i + 1)
'Ler o codigo de erro
cod_erro = linha_next
'Selecionar a ultima linha da base de dados para verificar se e necessario
escrever a linha
SQL = "SELECT TOP(10) Empresa, Maquina, Tipo, Cod_erro, Mensagem, Data, ID FROM
Mensagem_erro where Maquina = '" & maq_log & "'" AND Mensagem = '" & cod_erro &
"' ORDER BY ID DESC"
Tracker_variavel()
If cod_erro.contains("") Then
cod_erro = Replace(cod_erro, "", "")
End If
'Escrever o erro na tabela da maquina

```

```

SQL = "INSERT INTO Mensagem_erro([Empresa], [Maquina], [Cod_erro], [Mensagem],
[Data]) values('" & empresa & "', '" & maq_log & "', ' ', '" & cod_erro & "', '"
& CDate(time_erro).ToString("yyyy-MM-dd HH:mm:ss") & " ')"
Tracker_escrever()
Catch ex As Exception
MsgBox(ex.Message)
End Try
End If
Else
GoTo tempo_ciclo
End If
Next
tempo_ciclo:
Catch ex As Exception
MsgBox(ex.Message)
End Try
End Sub
'Sub rotinas de apoio ao automatismo
'nao foram incluidas no anexo por poupanca de espaco
'estao explicadas no corpo do texto e tem a mesma logica que as sub rotinas
apresentadas no automatismo fidia
End Class

```

Anexo D - Automatismo de compilação diária de informação

```

Public Class Form1
'declaracao das variaveis necessarias
Dim Cmd As SqlCommand
Dim SQL As String
Dim command As New SqlCommand
Dim estado
Dim connectionString As String = ""
Dim connection As SqlConnection = New SqlConnection(connectionString)
Dim da As New SqlDataAdapter
Dim dt As New DataSet
Dim ds As New DataSet
Dim utilizacao_atual
Dim tempo_corte
Dim segundos_atual
Dim cod_maq
Dim ult_data
Dim empresa
Dim dsc_empresa
Dim tempo_atual_escrever
Dim tempo_inic_escrever

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
inicio:
Thread.sleep(3500)
'selecionar todas as maquinas presentes no sistema
Try
SQL = "SELECT * FROM Var WHERE Maquina NOT LIKE '%TESTE%' AND Processar = 1
UNION ALL
SELECT * FROM Var_heid WHERE Maquina NOT LIKE '%TESTE%' AND Processar = 1
UNION ALL
SELECT * FROM Var_heid_640 WHERE Maquina NOT LIKE '%TESTE%' AND Processar = 1"
Tracker_var()
Catch ex As Exception
MsgBox("Erro 10: " & ex.Message)
End Try
Try

```



```

If dt.Tables("Tabela").Rows.Count <> 0 Then
For i = 0 To dt.Tables("Tabela").Rows.Count - 1
cod_maq = dt.Tables("Tabela").Rows(i).Item("Maquina")
empresa = dt.Tables("Tabela").Rows(i).Item("Empresa")
dsc_empresa = dt.Tables("Tabela").Rows(i).Item("Dsc_empresa")
ult_data = dt.Tables("Tabela").Rows(i).Item("Ult_data")
MaqLabel.Text = cod_maq.ToString
DateLabel.Text = DateTime.Parse(ult_data).Date.ToString("yyyy-MM-dd HH:mm:ss")
If ult_data.ToString = "" Then
GoTo proximo
End If
SQL = "Select * from " & cod_maq & " where (Inicio between '" &
DateTime.Parse(ult_data).Date.ToString("yyyy-MM-dd HH:mm:ss") & "' AND '" &
DateTime.Parse(ult_data).ToString("yyyy-MM-dd HH:mm:ss") & "') AND (Estado =
'Corte')"
SqlLabel.Text = SQL
Tracker_var_maquina()
Try
'Somar o valor de tempo de todas as linhas
tempo_corte = 0
For k = 0 To dt.Tables("Tabela_maquina").Rows.Count - 1
If cod_maq.Contains("_") = True Then
tempo_corte = tempo_corte + dt.Tables("Tabela_maquina").Rows(k).Item("Tempo")
Else
tempo_corte = tempo_corte +
dt.Tables("Tabela_maquina").Rows(k).Item("Tempo_corte")
End If
Next
'Fazer a conta em relacao a ultima hora escrita
segundos_atual = ((Convert.ToDateTime(ult_data).Hour) * 3600) +
((Convert.ToDateTime(ult_data).Minute) * 60) +
(Convert.ToDateTime(ult_data).Second)
utilizacao_atual = (tempo_corte * 100) / (segundos_atual)
utilizacao_atual = Math.Round(utilizacao_atual, 1)
'escrever para a BD
SQL = "SELECT * FROM Percentagem_diaria where Maquina = '" & cod_maq & "'"
Tracker_var_percentagem()
If dt.Tables("Tabela_percentagem").Rows.Count <> 0 Then
SQL = "UPDATE Percentagem_diaria set percentagem_diaria = '" & utilizacao_atual
& "', Ult_data = '" & CDate(ult_data).ToString("yyyy-MM-dd HH:mm:ss") & "'" where
Maquina = '" & cod_maq & "' "
Tracker_escrever()
Else
SQL = "INSERT INTO Percentagem_diaria([Maquina], [Empresa], [DSC_empresa],
[percentagem_diaria], [Ult_data]) values('" & cod_maq & "', '" & empresa & "',
'" & dsc_empresa & "', '" & utilizacao_atual & "', '" &
CDate(ult_data).ToString("yyyy-MM-dd HH:mm:ss") & "'"
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 20: " & ex.Message)
End Try
proximo:
Next
End If
Catch ex As Exception
End Try
GoTo inicio
End Sub
'Sub rotinas de apoio ao automatismo
'nao foram incluidas no anexo por poupanca de espaco

```

```
'tem a mesma logica que as sub rotinas apresentadas no automatismo fidia
End Class
```

Anexo E - Código do menu inicial do *Front End*

```
Public Class Dash_inicio
'declaracao das variaveis necessarias
Dim Cmd As SqlCommand
Dim SQL As String
Dim command As New SqlCommand
Dim estado
Dim connectionString As String = ""
Dim connection As SqlConnection = New SqlConnection(connectionString)
Dim da As New SqlDataAdapter
Dim dt As New DataSet
Dim ds As New DataSet
Dim linha
Dim hora_final
'lista de empresas
Dim input_empresa() As String = {"MDA1", "MDA2", "IMA", "IGM", "SA", "MECA",
"UL", "CM"}
Dim list_empresas As List(Of String) = New List(Of String)(input_empresa)
Dim utilizacao_1
Dim utilizacao_7
Dim tempo_corte
Dim tempo_total
Dim hora_atual
Dim segundos_atual
Dim utilizacao_atual
Private Sub Dash_inicio_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
Me.Width = Screen.PrimaryScreen.WorkingArea.Width - Inicio.Panell.Width
Me.Height = Screen.PrimaryScreen.WorkingArea.Height - Inicio.Panel2.Height
Porcentagem_1D()
Porcentagem_7Dias()
End Sub
'Dados de trabalho do dia anterior
Sub Porcentagem_1D()
For Each empresa In input_empresa
SQL = "SELECT * FROM Tab_comp where Maquina like '%" & empresa & "%' AND MAQUINA
NOT LIKE '%TESTE%' AND Data = '" & DateTime.Parse(DateAdd(DateInterval.Day, (-
1), Now)).Date.ToString("yyyy-MM-dd") & "' ORDER BY Data"
'Escrever para tabela virtual
Tracker_Atualias: Try
connection.Open()
If dt.Tables.Contains("Tracker_Maquina") Then
dt.Tables("Tracker_Maquina").Clear()
End If
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dt, "Tracker_Maquina")
connection.Close()
Catch ex As Exception
GoTo Tracker_Atualias
End Try
'Garantir que as vari'is estao corretas
tempo_corte = 0
tempo_total = 0
utilizacao_1 = 0
Try
For i = 0 To dt.Tables("Tracker_Maquina").Rows.Count - 1
```

```

tempo_corte = tempo_corte +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo")
tempo_total = tempo_total +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Parada_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Setup_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Ligada_segundo")
Next
'Fazer o calculo em relacao a hora escrita
utilizacao_1 = (tempo_corte * 100) / (tempo_total)
utilizacao_1 = Math.Round(utilizacao_1, 0)
Catch ex As Exception
MsgBox(ex.Message)
End Try
'Escrever no gráfico correspondente
If empresa = "MDA1" Then
If utilizacao_1 < 60 Then
MDA1_Barra_1D.BackColor = Color.FromArgb(224, 66, 68)
MDA1_Barra_1D.BackColor2 = Color.FromArgb(224, 66, 68)
MDA1_1D.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
ElseIf utilizacao_1 >= 60 Then
MDA1_Barra_1D.BackColor = Color.LightSeaGreen
MDA1_Barra_1D.BackColor2 = Color.LightSeaGreen
MDA1_1D.Items(2).ForeColor = Color.LightSeaGreen
Else
End If
MDA1_1D.Value = utilizacao_1
ElseIf empresa = "MDA2" Then
If utilizacao_1 < 60 Then
MDA2_Barra_1D.BackColor = Color.FromArgb(224, 66, 68)
MDA2_Barra_1D.BackColor2 = Color.FromArgb(224, 66, 68)
MDA2_1D.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
MDA2_Barra_1D.BackColor = Color.LightSeaGreen
MDA2_Barra_1D.BackColor2 = Color.LightSeaGreen
MDA2_1D.Items(2).ForeColor = Color.LightSeaGreen
End If
MDA2_1D.Value = utilizacao_1
ElseIf empresa = "SA" Then
If utilizacao_1 < 60 Then
SA_Barra_1D.BackColor = Color.FromArgb(224, 66, 68)
SA_Barra_1D.BackColor2 = Color.FromArgb(224, 66, 68)
SA_1D.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
SA_Barra_1D.BackColor = Color.LightSeaGreen
SA_Barra_1D.BackColor2 = Color.LightSeaGreen
SA_1D.Items(2).ForeColor = Color.LightSeaGreen
End If
SA_1D.Value = utilizacao_1
ElseIf empresa = "IMA" Then
If utilizacao_1 < 60 Then
IMA_Barra_1D.BackColor = Color.FromArgb(224, 66, 68)
IMA_Barra_1D.BackColor2 = Color.FromArgb(224, 66, 68)
IMA_1D.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
IMA_Barra_1D.BackColor = Color.LightSeaGreen
IMA_Barra_1D.BackColor2 = Color.LightSeaGreen
IMA_1D.Items(2).ForeColor = Color.LightSeaGreen
End If
IMA_1D.Value = utilizacao_1
ElseIf empresa = "IGM" Then

```

```

If utilizacao_1 < 60 Then
IGM_Barra_1D.BackColor = Color.FromArgb(224, 66, 68)
IGM_Barra_1D.BackColor2 = Color.FromArgb(224, 66, 68)
IGM_1D.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
IGM_Barra_1D.BackColor = Color.LightSeaGreen
IGM_Barra_1D.BackColor2 = Color.LightSeaGreen
IGM_1D.Items(2).ForeColor = Color.LightSeaGreen
End If
IGM_1D.Value = utilizacao_1
ElseIf empresa = "MECA" Then
If utilizacao_1 < 60 Then
MECA_Barra_1D.BackColor = Color.FromArgb(224, 66, 68)
MECA_Barra_1D.BackColor2 = Color.FromArgb(224, 66, 68)
MECA_1D.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
MECA_Barra_1D.BackColor = Color.LightSeaGreen
MECA_Barra_1D.BackColor2 = Color.LightSeaGreen
MECA_1D.Items(2).ForeColor = Color.LightSeaGreen
End If
MECA_1D.Value = utilizacao_1
ElseIf empresa = "UL" Then
If utilizacao_1 < 60 Then
UL_Barra_1D.BackColor = Color.FromArgb(224, 66, 68)
UL_Barra_1D.BackColor2 = Color.FromArgb(224, 66, 68)
UL_1D.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
UL_Barra_1D.BackColor = Color.LightSeaGreen
UL_Barra_1D.BackColor2 = Color.LightSeaGreen
UL_1D.Items(2).ForeColor = Color.LightSeaGreen
End If
UL_1D.Value = utilizacao_1
ElseIf empresa = "CM" Then
If utilizacao_1 < 60 Then
CM_Barra_1D.BackColor = Color.FromArgb(224, 66, 68)
CM_Barra_1D.BackColor2 = Color.FromArgb(224, 66, 68)
CM_1D.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
CM_Barra_1D.BackColor = Color.LightSeaGreen
CM_Barra_1D.BackColor2 = Color.LightSeaGreen
CM_1D.Items(2).ForeColor = Color.LightSeaGreen
End If
CM_1D.Value = utilizacao_1
End If
Next
End Sub
'Dados do trabalho da semana anterior
Sub Percentagem_7Dias()
'Selecionar as linhas corretas da tabela complementar
For Each empresa In input_empresa
SQL = "SELECT * FROM Tab_comp where Maquina like '%" & empresa & "%' AND MAQUINA
NOT LIKE '%TESTE%' AND Data between '" &
DateTime.Parse(DateAdd(DateInterval.Day, (-7), Now)).Date.ToString("yyyy-MM-dd")
& "' AND '" & DateTime.Parse(DateAdd(DateInterval.Day, (-1),
Now)).Date.ToString("yyyy-MM-dd") & "' ORDER BY Data"
'Escrever para tabela virtual
Tracker_7dias: Try
connection.Open()
If dt.Tables.Contains("Tracker_Maquina") Then
dt.Tables("Tracker_Maquina").Clear()
End If

```

```

da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dt, "Tracker_Maquina")
connection.Close()
Catch ex As Exception
GoTo Tracker_7dias
End Try
'Garantir que as vari'is estao corretas
tempo_corte = 0
tempo_total = 0
Try
For i = 0 To dt.Tables("Tracker_Maquina").Rows.Count - 1

tempo_corte = tempo_corte +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo")

tempo_total = tempo_total +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Parada_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Setup_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Ligada_segundo")
Next
'Fazer o calculo em relacao a hora escrita
utilizacao_7 = (tempo_corte * 100) / (Tempo_total)
utilizacao_7 = Math.Round(utilizacao_7, 0)
Catch ex As Exception
MsgBox(ex.Message)
End Try
'Escrever no gráfico correspondente
If empresa = "MDA1" Then
If utilizacao_7 < 60 Then
MDA1_Barra_7D.BackColor = Color.FromArgb(224, 66, 68)
MDA1_Barra_7D.BackColor2 = Color.FromArgb(224, 66, 68)
MDA1_7Dias.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
MDA1_Barra_7D.BackColor = Color.LightSeaGreen
MDA1_Barra_7D.BackColor2 = Color.LightSeaGreen
MDA1_7Dias.Items(2).ForeColor = Color.LightSeaGreen
End If
MDA1_7Dias.Value = utilizacao_7
ElseIf empresa = "MDA2" Then
If utilizacao_7 < 60 Then
MDA2_Barra_7D.BackColor = Color.FromArgb(224, 66, 68)
MDA2_Barra_7D.BackColor2 = Color.FromArgb(224, 66, 68)
MDA2_7Dias.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
MDA2_Barra_7D.BackColor = Color.LightSeaGreen
MDA2_Barra_7D.BackColor2 = Color.LightSeaGreen
MDA2_7Dias.Items(2).ForeColor = Color.LightSeaGreen
End If
MDA2_7Dias.Value = utilizacao_7
ElseIf empresa = "SA" Then
If utilizacao_7 < 60 Then
SA_Barra_7D.BackColor = Color.FromArgb(224, 66, 68)
SA_Barra_7D.BackColor2 = Color.FromArgb(224, 66, 68)
SA_7Dias.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
SA_Barra_7D.BackColor = Color.LightSeaGreen
SA_Barra_7D.BackColor2 = Color.LightSeaGreen
SA_7Dias.Items(2).ForeColor = Color.LightSeaGreen
End If
SA_7Dias.Value = utilizacao_7

```

```

ElseIf empresa = "IMA" Then
If utilizacao_7 < 60 Then
IMA_Barra_7D.BackColor = Color.FromArgb(224, 66, 68)
IMA_Barra_7D.BackColor2 = Color.FromArgb(224, 66, 68)
IMA_7Dias.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
IMA_Barra_7D.BackColor = Color.LightSeaGreen
IMA_Barra_7D.BackColor2 = Color.LightSeaGreen
IMA_7Dias.Items(2).ForeColor = Color.LightSeaGreen
End If
IMA_7Dias.Value = utilizacao_7
ElseIf empresa = "IGM" Then
If utilizacao_7 < 60 Then
IGM_Barra_7D.BackColor = Color.FromArgb(224, 66, 68)
IGM_Barra_7D.BackColor2 = Color.FromArgb(224, 66, 68)
IGM_7Dias.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
IGM_Barra_7D.BackColor = Color.LightSeaGreen
IGM_Barra_7D.BackColor2 = Color.LightSeaGreen
IGM_7Dias.Items(2).ForeColor = Color.LightSeaGreen
End If
IGM_7Dias.Value = utilizacao_7
ElseIf empresa = "MECA" Then
If utilizacao_7 < 60 Then
MECA_Barra_7D.BackColor = Color.FromArgb(224, 66, 68)
MECA_Barra_7D.BackColor2 = Color.FromArgb(224, 66, 68)
MECA_7Dias.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
MECA_Barra_7D.BackColor = Color.LightSeaGreen
MECA_Barra_7D.BackColor2 = Color.LightSeaGreen
MECA_7Dias.Items(2).ForeColor = Color.LightSeaGreen
End If
MECA_7Dias.Value = utilizacao_7
ElseIf empresa = "UL" Then
If utilizacao_7 < 60 Then
UL_Barra_7D.BackColor = Color.FromArgb(224, 66, 68)
UL_Barra_7D.BackColor2 = Color.FromArgb(224, 66, 68)
UL_7Dias.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
UL_Barra_7D.BackColor = Color.LightSeaGreen
UL_Barra_7D.BackColor2 = Color.LightSeaGreen
UL_7Dias.Items(2).ForeColor = Color.LightSeaGreen
End If
UL_7Dias.Value = utilizacao_7
ElseIf empresa = "CM" Then
If utilizacao_7 < 60 Then
CM_Barra_7D.BackColor = Color.FromArgb(224, 66, 68)
CM_Barra_7D.BackColor2 = Color.FromArgb(224, 66, 68)
CM_7Dias.Items(2).ForeColor = Color.FromArgb(224, 66, 68)
Else
CM_Barra_7D.BackColor = Color.LightSeaGreen
CM_Barra_7D.BackColor2 = Color.LightSeaGreen
CM_7Dias.Items(2).ForeColor = Color.LightSeaGreen
End If
CM_7Dias.Value = utilizacao_7
End If
Next
End Sub
End Class

```

Anexo F - Código correspondente ao *Dashboard* Empresa

```
Public Class Dash_maq_igm
'declaracao de variaveis necessarias ao automatismo
Dim cod_maq
Dim empresa
Dim Cmd As SqlCommand
Dim SQL As String
Dim command As New SqlCommand
Dim estado
Dim connectionString As String = ""
Dim connection As SqlConnection = New SqlConnection(connectionString)
Dim da As New SqlDataAdapter
Dim dt As New DataSet
Dim ds As New DataSet
Dim linha
Dim hora_final
Dim tempo_corte
Dim utilizacao_atual
Dim segundos_atual

Private Sub Dash_maq_IGM_Load(sender As Object, e As EventArgs) Handles
 MyBase.Load
Dim tab_maq As New Tab_Maq
Me.Width = Screen.PrimaryScreen.WorkingArea.Width - Inicio.Pane1.Width
Me.Height = Screen.PrimaryScreen.WorkingArea.Height - 213
For Each layout As FlowLayoutPanel In Me.Controls.OfType(Of FlowLayoutPanel)
'Para cada maquina no menu 'Empresa' correspondente e realizado o ciclo de
recolha de informacap
For Each panel As Panel In layout.Controls.OfType(Of Panel)
Try
SQL = "SELECT * FROM VAR WHERE Maquina like '" & panel.Name & "'
UNION ALL
SELECT * FROM VAR_HEID WHERE Maquina like '" & panel.Name & "'
UNION ALL
SELECT * FROM VAR_HEID_640 WHERE Maquina like '" & panel.Name & "' "
Tracker_var()
If dt.Tables("Semaforo").Rows.Count <> 0 Then
If dt.Tables("Semaforo").Rows(0).Item("Estado").ToString = "Corte" Or
dt.Tables("Semaforo").Rows(0).Item("Estado").ToString = "Ligada" Then
For Each semaforo As Panel In panel.Controls.OfType(Of Panel)
If semaforo.Name.Contains("Painel_verde_") = True Then
semaforo.BackColor = Color.Green
End If
If semaforo.Name.Contains("Painel_amarelo_") = True Then
semaforo.BackColor = Color.DarkGray
End If
If semaforo.Name.Contains("Painel_vermelho_") = True Then
semaforo.BackColor = Color.DarkGray
End If
Next
ElseIf dt.Tables("Semaforo").Rows(0).Item("Estado").ToString = "Parada" Then
For Each semaforo As Panel In panel.Controls.OfType(Of Panel)
If semaforo.Name.Contains("Painel_verde_") = True Then
semaforo.BackColor = Color.DarkGray
End If
If semaforo.Name.Contains("Painel_amarelo_") = True Then
semaforo.BackColor = Color.DarkGray
End If
If semaforo.Name.Contains("Painel_vermelho_") = True Then
semaforo.BackColor = Color.Red
End If
End If
```

```

Next
ElseIf dt.Tables("Semaforo").Rows(0).Item("Estado").ToString = "Setup" Then
For Each semaforo As Panel In panel.Controls.OfType(Of Panel)
If semaforo.Name.Contains("Painel_verde_") = True Then
semaforo.BackColor = Color.DarkGray
End If
If semaforo.Name.Contains("Painel_amarelo_") = True Then
semaforo.BackColor = Color.Yellow
End If
If semaforo.Name.Contains("Painel_vermelho_") = True Then
semaforo.BackColor = Color.DarkGray
End If
Next
ElseIf dt.Tables("Semaforo").Rows(0).Item("Estado").ToString = "" Or
dt.Tables("Semaforo").Rows(0).Item("Ligacao") = False Then
For Each semaforo As Panel In panel.Controls.OfType(Of Panel)
If semaforo.Name.Contains("Painel_verde_") = True Then
semaforo.BackColor = Color.DarkGray
End If
If semaforo.Name.Contains("Painel_amarelo_") = True Then
semaforo.BackColor = Color.DarkGray
End If
If semaforo.Name.Contains("Painel_vermelho_") = True Then
semaforo.BackColor = Color.DarkGray
End If
Next
End If
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
Try
If dt.Tables("Semaforo").Rows.Count <> 0 Then
'Ver se existe ligacao
If dt.Tables("Semaforo").Rows(0).Item("Ligacao") = True Then
'Carregar as percentagens atuais
Try
SQL = "SELECT * FROM Percentagem_diaria where Maquina = '" & panel.Name & "'
order by id DESC"
Tracker_var_percentagem()
Catch ex As Exception
MsgBox(ex.Message)
End Try
Try
'Verificar se a ultima data e a data de hoje
If dt.Tables("Percentagem").Rows(0).Item("Ult_data").date = Today.Date Then
If dt.Tables("Percentagem").Rows.Count <> 0 Then
For Each barra As Telerik.WinControls.UI.RadProgressBar In
panel.Controls.OfType(Of Telerik.WinControls.UI.RadProgressBar)
Dim percentagem = dt.Tables("Percentagem").Rows(0).Item("percentagem_diaria")
If percentagem.contains(".") Then
percentagem = Mid(percentagem, 1, InStr(percentagem, ".") - 1)
End If
If percentagem.contains(",") Then
percentagem = Mid(percentagem, 1, InStr(percentagem, ",") - 1)
End If
percentagem = Convert.ToDouble(percentagem)
Try
If percentagem <= 100 Then
barra.Value1 = Convert.ToDouble(percentagem)

```



```

barra.Text =
(dt.Tables("Porcentagem").Rows(0).Item("porcentagem_diaria")).ToString & "%"
Else
barra.Value1 = Nothing
barra.Text = "NaN"
End If
Catch ex As Exception
MsgBox(ex.Message & " - Erro na conversao dos dados")
End Try
Next
Else
For Each barra As Telerik.Windows.Controls.RadProgressBar In
panel.Controls.OfType(Of Telerik.Windows.Controls.RadProgressBar)
barra.Value1 = Nothing
barra.Text = "NaN"
Next
End If
Else
For Each barra As Telerik.Windows.Controls.RadProgressBar In
panel.Controls.OfType(Of Telerik.Windows.Controls.RadProgressBar)
barra.Value1 = Nothing
barra.Text = "NaN"
Next
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
Else
Try
For Each barra As Telerik.Windows.Controls.RadProgressBar In
panel.Controls.OfType(Of Telerik.Windows.Controls.RadProgressBar)
barra.Value1 = Nothing
barra.Text = "NaN"
For Each botao As Button In panel.Controls.OfType(Of Button)
botao.Enabled = False
Next
Next
Catch ex As Exception
MsgBox(ex.Message)
End Try
End If
Else
Try
For Each barra As Telerik.Windows.Controls.RadProgressBar In
panel.Controls.OfType(Of Telerik.Windows.Controls.RadProgressBar)
barra.Value1 = Nothing
barra.Text = "NaN"
For Each botao As Button In panel.Controls.OfType(Of Button)
botao.Enabled = False
Next
Next
Catch ex As Exception
MsgBox(ex.Message)
End Try
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
Next
Next
End Sub

```

```

'Se premir o botao de mais informacda maquina
Public Sub Button11_Click(sender As Object, e As EventArgs) Handles
Button1.Click, Button2.Click, Button3.Click, Button4.Click, Button5.Click,
Button6.Click, Button7.Click, Button8.Click, Button9.Click, Button10.Click,
Button11.Click, Button12.Click, Button13.Click, Button14.Click
Dim tab_maq As New Tab_Maq
Dim dash_detail As New Dash_detalhado_maq
For Each layout As FlowLayoutPanel In Me.Controls(OfType(Of FlowLayoutPanel))
For Each panel As Panel In layout.Controls(OfType(Of Panel))
For Each btn As Button In panel.Controls(OfType(Of Button))
If btn.Name = sender.name Then
For Each text As Label In panel.Controls(OfType(Of Label))
If text.Text.Contains("-") And Not text.Text.Contains(".h") And Not
text.Text.Contains(".fid") Then
Inicio.Inicio_maquina.Text = panel.Name
Inicio.Inicio_empresa.Text = Mid(panel.Name, 1, InStr(panel.Name, "_") - 1)
Try
Try
connection.Close()
connection.Open()
If ds.Tables.Contains("Dados") = True Then
ds.Tables("Dados").Clear()
End If
SQL = "SELECT * from " & panel.Name & " where Inicio between '" &
CDate(Now.Date).ToString("yyyy-MM-dd 00:00:00") & "' AND '" &
CDate(Now.Date).ToString("yyyy-MM-dd 23:59:59") & "'"
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(ds, "Dados")
connection.Close()
Catch ex As Exception
MsgBox(ex.Message)
End Try
For i = 0 To ds.Tables("Dados").Rows.Count - 1
linha = CDate(ds.Tables("Dados").Rows(i).Item("Inicio"))
If i = ds.Tables("Dados").Rows.Count - 1 Then
hora_final = DateTime.Now
Else
hora_final = ds.Tables("Dados").Rows(i + 1).Item("Inicio")
End If
Next
Catch ex As Exception
MsgBox(ex.Message)
End Try
End If
Next
For Each imagem As PictureBox In panel.Controls(OfType(Of PictureBox))
dash_detail.PictureBox1.ImageLocation = imagem.ImageLocation
Next
End If
Next
Next
Next
For Each control As UserControl In Inicio.Controls(OfType(Of UserControl))
If control.Name = "Tab_vista" Or control.Name = "user_maquinas" Or control.Name
= "Dash_empresa" Then
Inicio.Controls.Remove(control)
End If
Next
For Each controls As Control In Inicio.Controls(OfType(Of Control))
If controls.Name = "Tab_vista" Or controls.Name = "user_maquinas" Or
controls.Name = "Dash_empresa" Then

```

```

Inicio.Controls.Remove(controls)
End If
Next
Inicio.Controls.Remove(Me)
dash_detail.Cod_maquina.Text = Inicio.Inicio_maquina.Text
With dash_detail
.Name = "Dash_Detalhado_maq"
.Location = New Point(225, 101)
.Size = New Size(1680, 910)
End With
Inicio.Controls.Add(dash_detail)
With tab_maq
.Name = "Tab_maq"
.Location = New Point(225, 45)
End With
Inicio.Controls.Add(tab_maq)
End Sub
Private Sub FlowLayoutPanel1_Paint(sender As Object, e As PaintEventArgs)
Handles FlowLayoutPanel1.Paint
End Sub
End Class

```

Anexo G - Código correspondente ao *Dashboard Máquina*

```

Public Class Dash_detalhado_maq
'declaracao das variaveis necessarias
Dim data_inicio
Dim data_fim
Dim Dash_Grafico_detalhado As New Dash_Grafico_detalhado
Dim dash_grafico_semanal As New Dash_Grafico_Semanal
Dim Cmd As SqlCommand
Dim SQL As String
Dim command As New SqlCommand
Dim estado
Dim connectionString As String = ""
Dim connection As SqlConnection = New SqlConnection(connectionString)
Dim da As New SqlDataAdapter
Dim dt As New DataSet
Dim ds As New DataSet
Dim linha
Dim hora_final
Dim utilizacao_24
Dim tempo_total
Dim tempo_corte
Dim utilizacao_7
Dim utilizacao_30
Dim utilizacao_90
Dim utilizacao_365
Dim semana
Dim Semana_Old
Dim Corte_Perc
Dim Corte_Seg
Dim Ligada_Perc
Dim Ligada_Seg
Dim Justificada_Perc
Dim Justificada_seg
Dim Setup_Perc
Dim Setup_Seg
Dim Parada_Perc
Dim Parada_seg

```

```

Private Sub Dash_Detalhado_maq_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
'Definir o tamanho do form
Dim tab_maq As New Tab_Maq
Me.Width = Screen.PrimaryScreen.WorkingArea.Width - Inicio.Panell1.Width
Me.Height = Screen.PrimaryScreen.WorkingArea.Height - tab_maq.Height -
Inicio.Panel2.Height
layout_graficos.Location = New Point((Me.Width - layout_graficos.Width) / 2,
layout_graficos.Location.Y)
Painel_datas.Location = New Point((Me.Width - layout_graficos.Width) / 2,
Painel_datas.Location.Y)
Flow_layout_graficos.Width = Screen.PrimaryScreen.WorkingArea.Width -
Inicio.Panell1.Width
Flow_layout_graficos.Height = Screen.PrimaryScreen.WorkingArea.Height * 0.6
FlowLayoutPanel2.Width = Screen.PrimaryScreen.WorkingArea.Width -
FlowLayoutPanel2.Location.X - Inicio.Panell1.Width
If FlowLayoutPanel2.Width > 1005 Then
FlowLayoutPanel2.Width = 1005
End If
If layout_graficos.Width + Flow_Switch.Width >= FlowLayoutPanel2.Width Then
Flow_Switch.Width = layout_graficos.Width - 92
If Flow_Switch.Width < 485 Then
Flow_Switch.Width = 485
End If
Flow_Switch.Height = RadLabel7.Height + 5
FlowLayoutPanel2.Height = Painel_datas.Height + layout_graficos.Height +
Flow_Switch.Height
End If
Flow_layout_graficos.Location = New Point(Flow_layout_graficos.Location.X,
layout_graficos.Location.Y + layout_graficos.Height + 100)
Flow_layout_graficos.Height = Screen.PrimaryScreen.WorkingArea.Height -
Flow_layout_graficos.Location.Y - 50
FlowLayoutPanel2.Location = New Point(FlowLayoutPanel2.Location.X +
((Screen.PrimaryScreen.WorkingArea.Width - Inicio.Panell1.Width -
FlowLayoutPanel2.Location.X - FlowLayoutPanel2.Width) / 2),
FlowLayoutPanel2.Location.Y)
'Definir as datas nas combo box
Refresh()
data_1.Value = CDate(DateAdd(DateInterval.Day, -7, Now))
data2.Value = Today
Atualizar.PerformClick()
'Indicar o semaforo
Try
SQL = "SELECT * FROM VAR WHERE Maquina like '" & Inicio.Inicio_maquina.Text & "'
UNION ALL
SELECT * FROM VAR_HEID WHERE Maquina like '" & Inicio.Inicio_maquina.Text & "'
UNION ALL
SELECT * FROM VAR_HEID_640 WHERE Maquina like '" & Inicio.Inicio_maquina.Text &
"' "
Tracker_var()
Catch ex As Exception
MsgBox(ex.Message)
End Try
Try
If dt.Tables("Semaforo").Rows.Count <> 0 Then
If dt.Tables("Semaforo").Rows(0).Item("Estado").ToString = "Corte" Or
dt.Tables("Semaforo").Rows(0).Item("Estado").ToString = "Ligada" Then
For Each semaforo As Panel In Me.Controls.OfType(Of Panel)
If semaforo.Name.Contains("Painel_verde_") = True Then
semaforo.BackColor = Color.Green
End If

```

```

If semaforo.Name.Contains("Painel_amarelo_") = True Then
semaforo.BackColor = Color.DarkGray
End If
If semaforo.Name.Contains("Painel_vermelho_") = True Then
semaforo.BackColor = Color.DarkGray
End If
Next
ElseIf dt.Tables("Semaforo").Rows(0).Item("Estado").ToString = "Parada" Then
For Each semaforo As Panel In Me.Controls.OfType(Of Panel)
If semaforo.Name.Contains("Painel_verde_") = True Then
semaforo.BackColor = Color.DarkGray
End If
If semaforo.Name.Contains("Painel_amarelo_") = True Then
semaforo.BackColor = Color.DarkGray
End If
If semaforo.Name.Contains("Painel_vermelho_") = True Then
semaforo.BackColor = Color.Red
End If
Next
ElseIf dt.Tables("Semaforo").Rows(0).Item("Estado").ToString = "Setup" Then
For Each semaforo As Panel In Me.Controls.OfType(Of Panel)
If semaforo.Name.Contains("Painel_verde_") = True Then
semaforo.BackColor = Color.DarkGray
End If
If semaforo.Name.Contains("Painel_amarelo_") = True Then
semaforo.BackColor = Color.Yellow
End If
If semaforo.Name.Contains("Painel_vermelho_") = True Then
semaforo.BackColor = Color.DarkGray
End If
Next
ElseIf dt.Tables("Semaforo").Rows(0).Item("Estado").ToString = "" Or
dt.Tables("Semaforo").Rows(0).Item("Ligacao") = False Then
For Each semaforo As Panel In Me.Controls.OfType(Of Panel)
If semaforo.Name.Contains("Painel_verde_") = True Then
semaforo.BackColor = Color.DarkGray
End If
If semaforo.Name.Contains("Painel_amarelo_") = True Then
semaforo.BackColor = Color.DarkGray
End If
If semaforo.Name.Contains("Painel_vermelho_") = True Then
semaforo.BackColor = Color.DarkGray
End If
Next
End If
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
'Carregar as percentagens atuais
Try
SQL = "SELECT * FROM Percentagem_diaria where Maquina = '" &
Inicio.Inicio_maquina.Text & "'"
Tracker_var_percentagem()
Catch ex As Exception
MsgBox(ex.Message)
End Try
Dim percentagem = dt.Tables("Percentagem").Rows(0).Item("percentagem_diaria")
Try
'percentagem = 0
If percentagem.Contains(".") Then

```

```

percentagem = Mid(percentagem, 1, InStr(percentagem, ".") - 1)
End If
If percentagem.Contains(",") Then
percentagem = Mid(percentagem, 1, InStr(percentagem, ",") - 1)
End If
percentagem = Convert.ToDouble(percentagem)
Catch ex As Exception
MsgBox(ex.Message)
End Try
Try
If percentagem <= 100 Then
percentagem = Convert.ToDouble(percentagem)
Gauge_atual.Value = percentagem
Gauge_atual.Text = percentagem.ToString & "%"
Else
Gauge_atual.Value = Nothing
Gauge_atual.Text = "NaN"
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
data_inicio = Today
'24H
Try
SQL = "SELECT * FROM Tab_comp where Maquina like '%" &
Inicio.Inicio_maquina.Text & "%' AND Data = '" &
DateTime.Parse(DateAdd(DateInterval.Day, (-1),
data_inicio)).Date.ToString("yyyy-MM-dd") & "' ORDER BY Data "
Tracker_24horas: Try
connection.Close()
connection.Open()
If dt.Tables.Contains("Tracker_Maquina") Then
dt.Tables("Tracker_Maquina").Clear()
End If
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dt, "Tracker_Maquina")
connection.Close()
Catch ex As Exception
GoTo Tracker_24horas
End Try
tempo_corte = 0
tempo_total = 0
If dt.Tables("Tracker_Maquina").Rows.Count <> 0 Then
If CDate(dt.Tables("Tracker_Maquina").Rows(0).Item("Data")).Date.ToString("yyyy-
MM-dd") = DateTime.Parse(DateAdd(DateInterval.Day, (-1),
data_inicio)).Date.ToString("yyyy-MM-dd") Then
For i = 0 To dt.Tables("Tracker_Maquina").Rows.Count - 1
tempo_corte = tempo_corte +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo")
tempo_total = tempo_total +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Parada_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Setup_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Ligada_segundo")
Next
'Fazer a conta em relacao a ultima hora escrita
utilizacao_24 = (tempo_corte * 100) / (tempo_total)
utilizacao_24 = Math.Round(utilizacao_24, 1)
gauge_24.Value = utilizacao_24
gauge_24.Text = utilizacao_24.ToString & "%"
Else

```

```

gauge_24.Value = Nothing
gauge_24.Text = "NaN"
End If
Else
gauge_24.Value = Nothing
gauge_24.Text = "NaN"
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
'7 Dias
Try
SQL = "SELECT * FROM Tab_comp where Maquina like '%" &
Inicio.Inicio_maquina.Text & "%' AND Data = '" &
DateTime.Parse(DateAdd(DateInterval.Day, (-7),
data_inicio)).Date.ToString("yyyy-MM-dd") & "' ORDER BY Data "
Tracker_7dias: Try
connection.Open()
If dt.Tables.Contains("Tracker_Maquina") Then
dt.Tables("Tracker_Maquina").Clear()
End If
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dt, "Tracker_Maquina")
connection.Close()
Catch ex As Exception
GoTo Tracker_7dias
End Try
'Somar o valor de tempo de todas as linhas
tempo_corte = 0
tempo_total = 0
Try
If dt.Tables("Tracker_Maquina").Rows.Count <> 0 Then
If CDate(dt.Tables("Tracker_Maquina").Rows(0).Item("Data")).Date.ToString("yyyy-
MM-dd") = DateTime.Parse(DateAdd(DateInterval.Day, (-7),
data_inicio)).Date.ToString("yyyy-MM-dd") Then
For i = 0 To dt.Tables("Tracker_Maquina").Rows.Count - 1
tempo_corte = tempo_corte +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo")
tempo_total = tempo_total +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Parada_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Setup_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Ligada_segundo")
Next
'Fazer a conta em relacao a ultima hora escrita
utilizacao_7 = (tempo_corte * 100) / (tempo_total)
utilizacao_7 = Math.Round(utilizacao_7, 1)
Gauge_7d.Value = utilizacao_7
Else
Gauge_7d.Value = Nothing
Gauge_7d.Text = "NaN"
End If
Else
Gauge_7d.Value = Nothing
Gauge_7d.Text = "NaN"
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
Catch ex As Exception
MsgBox(ex.Message)

```

```

End Try
'30 Dias
Try
SQL = "SELECT * FROM Tab_comp where Maquina like '%" &
Inicio.Inicio_maquina.Text & "%' AND Data = '" &
DateTime.Parse(DateAdd(DateInterval.Day, (-30),
data_inicio)).Date.ToString("yyyy-MM-dd") & "' ORDER BY Data "
Tracker_30dias: Try
connection.Open()
If dt.Tables.Contains("Tracker_Maquina") Then
dt.Tables("Tracker_Maquina").Clear()
End If
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dt, "Tracker_Maquina")
connection.Close()
Catch ex As Exception
GoTo Tracker_30dias
End Try
'Somar o valor de tempo de todas as linhas
tempo_corte = 0
tempo_total = 0
Try
If dt.Tables("Tracker_Maquina").Rows.Count <> 0 Then
If CDate(dt.Tables("Tracker_Maquina").Rows(0).Item("Data")).Date.ToString("yyyy-
MM-dd") = DateTime.Parse(DateAdd(DateInterval.Day, (-30),
data_inicio)).Date.ToString("yyyy-MM-dd") Then
For i = 0 To dt.Tables("Tracker_Maquina").Rows.Count - 1
tempo_corte = tempo_corte +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo")
tempo_total = tempo_total +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Parada_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Setup_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Ligada_segundo")
Next
'Fazer a conta em relacao a ultima hora escrita
utilizacao_30 = (tempo_corte * 100) / (tempo_total)
utilizacao_30 = Math.Round(utilizacao_30, 1)
Gauge_30d.Value = utilizacao_30
Else
Gauge_30d.Value = Nothing
Gauge_30d.Text = "NaN"
End If
Else
Gauge_30d.Value = Nothing
Gauge_30d.Text = "NaN"
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
Catch ex As Exception
MsgBox(ex.Message)
End Try
'90 Dias
Try
SQL = "SELECT * FROM Tab_comp where Maquina like '%" &
Inicio.Inicio_maquina.Text & "%' AND Data = '" &
DateTime.Parse(DateAdd(DateInterval.Day, (-90),
data_inicio)).Date.ToString("yyyy-MM-dd") & "' ORDER BY Data "
Tracker_90dias: Try
connection.Open()

```



```

If dt.Tables.Contains("Tracker_Maquina") Then
dt.Tables("Tracker_Maquina").Clear()
End If
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dt, "Tracker_Maquina")
connection.Close()
Catch ex As Exception
GoTo Tracker_90dias
End Try
'Somar o valor de tempo de todas as linhas
tempo_corte = 0
tempo_total = 0
Try
If dt.Tables("Tracker_Maquina").Rows.Count <> 0 Then
If CDate(dt.Tables("Tracker_Maquina").Rows(0).Item("Data")).Date.ToString("yyyy-MM-dd") = DateTime.Parse(DateAdd(DateInterval.Day, (-90), data_inicio)).Date.ToString("yyyy-MM-dd") Then
For i = 0 To dt.Tables("Tracker_Maquina").Rows.Count - 1
tempo_corte = tempo_corte +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo")
tempo_total = tempo_total +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Parada_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Setup_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Ligada_segundo")
Next
'Fazer a conta em relacao a ultima hora escrita
utilizacao_90 = (tempo_corte * 100) / (tempo_total)
utilizacao_90 = Math.Round(utilizacao_90, 1)
Gauge_90d.Value = utilizacao_90
Else
Gauge_90d.Value = Nothing
Gauge_90d.Text = "NaN"
End If
Else
Gauge_90d.Value = Nothing
Gauge_90d.Text = "NaN"
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
Catch ex As Exception
MsgBox(ex.Message)
End Try
'365 Dias
Try
SQL = "SELECT * FROM Tab_comp where Maquina like '%" &
Inicio.Inicio_maquina.Text & "%' AND Data = '" &
DateTime.Parse(DateAdd(DateInterval.Day, (-365), data_inicio)).Date.ToString("yyyy-MM-dd") & "' ORDER BY Data "
Tracker_365dias: Try
connection.Open()
If dt.Tables.Contains("Tracker_Maquina") Then
dt.Tables("Tracker_Maquina").Clear()
End If
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dt, "Tracker_Maquina")
connection.Close()
Catch ex As Exception
GoTo Tracker_365dias
End Try

```

```

'Somar o valor de tempo de todas as linhas
tempo_corte = 0
tempo_total = 0
Try
If dt.Tables("Tracker_Maquina").Rows.Count <> 0 Then
If CDate(dt.Tables("Tracker_Maquina").Rows(0).Item("Data")).Date.ToString("yyyy-
MM-dd") = DateTime.Parse(DateAdd(DateInterval.Day, (-365),
data_inicio)).Date.ToString("yyyy-MM-dd") Then
For i = 0 To dt.Tables("Tracker_Maquina").Rows.Count - 1
tempo_corte = tempo_corte +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo")
tempo_total = tempo_total +
dt.Tables("Tracker_Maquina").Rows(i).Item("Corte_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Parada_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Setup_segundo") +
dt.Tables("Tracker_Maquina").Rows(i).Item("Ligada_segundo")
Next
'Fazer a conta em relacao a ultima hora escrita
utilizacao_365 = (tempo_corte * 100) / (tempo_total)
utilizacao_365 = Math.Round(utilizacao_365, 1)
Gauge_365d.Value = utilizacao_365
If utilizacao_365 = 0 Then
Gauge_365d.Text = "NaN"
End If
Else
Gauge_365d.Value = Nothing
Gauge_365d.Text = "NaN"
End If
Else
Gauge_365d.Value = Nothing
Gauge_365d.Text = "NaN"
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
Catch ex As Exception
MsgBox(ex.Message)
End Try
Atualizar.PerformClick()
End Sub
Private Sub RadButton1_Click(sender As Object, e As EventArgs) Handles
Atualizar.Click
Flow_layout_graficos.Controls.Clear()
Me.Cursor = Cursors.WaitCursor
If tipo_grafico.Value = True Then
Flow_layout_graficos.Controls.Clear()
data_inicio = data_1.Value
data_fim = data2.Value
'Adicionar os grafico de estado.
'Adiciona apenas o dia selecionado
If DateDiff(DateInterval.Day, data_inicio, data_fim) = 0 Then
Try
Dash_Grafico_detalhado.Chart_capacidade.Series("Corte").Points.Clear()
Dash_Grafico_detalhado.Chart_capacidade.Series("Ligada").Points.Clear()
Dash_Grafico_detalhado.Chart_capacidade.Series("Setup").Points.Clear()
Dash_Grafico_detalhado.Chart_capacidade.Series("Parada").Points.Clear()
Dash_Grafico_detalhado.Chart_capacidade.Series("Justificada").Points.Clear()
Try
connection.Close()
connection.Open()
If ds.Tables.Contains("Dados") = True Then

```

```

ds.Tables("Dados").Clear()
End If
SQL = "SELECT * from " & Cod_maquina.Text & " where Inicio between '" &
CDate(data_inicio.date).ToString("yyyy-MM-dd 00:00:00") & "' AND '" &
CDate(data_fim.Date).ToString("yyyy-MM-dd 23:59:59") & "' "
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(ds, "Dados")
connection.Close()
Catch ex As Exception
MsgBox(ex.Message)
End Try
Dash_Grafico_detalhado.Chart_capacidade.Series("Corte").YValueType =
DataVisualization.Charting.ChartValueType.Time
Dash_Grafico_detalhado.Chart_capacidade.Series("Ligada").YValueType =
DataVisualization.Charting.ChartValueType.Time
Dash_Grafico_detalhado.Chart_capacidade.Series("Setup").YValueType =
DataVisualization.Charting.ChartValueType.Time
Dash_Grafico_detalhado.Chart_capacidade.Series("Parada").YValueType =
DataVisualization.Charting.ChartValueType.Time
Dash_Grafico_detalhado.Chart_capacidade.Series("Justificada").YValueType =
DataVisualization.Charting.ChartValueType.Time
For i = 0 To ds.Tables("Dados").Rows.Count - 1
linha = CDate(ds.Tables("Dados").Rows(i).Item("Inicio"))
If i = ds.Tables("Dados").Rows.Count - 1 Then
hora_final = DateAndTime.Now
Else
hora_final = ds.Tables("Dados").Rows(i + 1).Item("Inicio")
End If
If ds.Tables("Dados").Rows(i).Item("Estado") = "Corte" Then
If (CDate(ds.Tables("Dados").Rows(i).Item("Fim")).Day -
CDate(ds.Tables("Dados").Rows(i).Item("Inicio")).Day) > 0 Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Corte").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(i).Item("Inicio")),
(CDate(ds.Tables("Dados").Rows(i).Item("Inicio")).Date +
CDate("23:59:59").TimeOfDay))
Dash_Grafico_detalhado.Chart_capacidade.Series("Corte").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(i).Item("Fim")).Date,
CDate(ds.Tables("Dados").Rows(i).Item("Fim")))
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Corte").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(i).Item("Inicio")),
CDate(ds.Tables("Dados").Rows(i).Item("Fim")))
End If
ElseIf ds.Tables("Dados").Rows(i).Item("Estado") = "Ligada" Then
If (CDate(ds.Tables("Dados").Rows(i).Item("Fim")).Day -
CDate(ds.Tables("Dados").Rows(i).Item("Inicio")).Day) > 0 Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Ligada").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(i).Item("Inicio")),
(CDate(ds.Tables("Dados").Rows(i).Item("Inicio")).Date +
CDate("23:59:59").TimeOfDay))
Dash_Grafico_detalhado.Chart_capacidade.Series("Ligada").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(i).Item("Fim")).Date,
CDate(ds.Tables("Dados").Rows(i).Item("Fim")))
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Ligada").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(i).Item("Inicio")),
CDate(ds.Tables("Dados").Rows(i).Item("Fim")))
End If
ElseIf ds.Tables("Dados").Rows(i).Item("Estado") = "Setup" Or
ds.Tables("Dados").Rows(i).Item("Estado") = "Alarme" Then

```

```

If (CDate(ds.Tables("Dados").Rows(i).Item("Fim")).Day -
CDate(ds.Tables("Dados").Rows(i).Item("Inicio")).Day) > 0 Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Setup").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(i).Item("Inicio")),
(CDate(ds.Tables("Dados").Rows(i).Item("Inicio")).Date +
CDate("23:59:59").TimeOfDay))
Dash_Grafico_detalhado.Chart_capacidade.Series("Setup").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(i).Item("Fim")).Date,
CDate(ds.Tables("Dados").Rows(i).Item("Fim")))
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Setup").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(i).Item("Inicio")),
CDate(ds.Tables("Dados").Rows(i).Item("Fim")))
End If
ElseIf ds.Tables("Dados").Rows(i).Item("Estado") = "Parada" Then
If IsDBNull(ds.Tables("Dados").Rows(i).Item("Paragem")) = True Then
If (CDate(ds.Tables("Dados").Rows(i).Item("Fim")).Day -
CDate(ds.Tables("Dados").Rows(i).Item("Inicio")).Day) > 0 Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Parada").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(i).Item("Inicio")),
(CDate(ds.Tables("Dados").Rows(i).Item("Inicio")).Date +
CDate("23:59:59").TimeOfDay))
Dash_Grafico_detalhado.Chart_capacidade.Series("Parada").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(i).Item("Fim")).Date,
CDate(ds.Tables("Dados").Rows(i).Item("Fim")))
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Parada").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(i).Item("Inicio")),
CDate(ds.Tables("Dados").Rows(i).Item("Fim")))
End If
Else
If (CDate(ds.Tables("Dados").Rows(i).Item("Fim")).Day -
CDate(ds.Tables("Dados").Rows(i).Item("Inicio")).Day) > 0 Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Justificada").Points.AddXY(linha
, CDate(ds.Tables("Dados").Rows(i).Item("Inicio")),
(CDate(ds.Tables("Dados").Rows(i).Item("Inicio")).Date +
CDate("23:59:59").TimeOfDay))
Dash_Grafico_detalhado.Chart_capacidade.Series("Justificada").Points.AddXY(linha
, CDate(ds.Tables("Dados").Rows(i).Item("Fim")).Date,
CDate(ds.Tables("Dados").Rows(i).Item("Fim")))
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Justificada").Points.AddXY(linha
, CDate(ds.Tables("Dados").Rows(i).Item("Inicio")),
CDate(ds.Tables("Dados").Rows(i).Item("Fim")))
End If
End If
End If
Dash_Grafico_detalhado.Chart_capacidade.ChartAreas(0).AxisY.Maximum =
CDate("23:59:59").ToOADate
Dash_Grafico_detalhado.Chart_capacidade.ChartAreas(0).AxisY.Interval = 1
Next
Catch ex As Exception
MsgBox(ex.Message)
End Try
'Adicionar o grafico ao Dash detalhado da maquina
With Dash_Grafico_detalhado
.Name = "Dash_grafico_detalhado"
.Size = New Size(Flow_layout_graficos.Size.Width,
Flow_layout_graficos.Size.Height)
End With
Flow_layout_graficos.Controls.Add(Dash_Grafico_detalhado)

```

```

ElseIf DateDiff(DateInterval.Day, data_inicio, data_fim) < 0 Then
Telerik.WinForms.RadMessageBox.SetThemeName(MaterialTealTheme1.ThemeName)
Telerik.WinForms.RadMessageBox.Show("As datas que inseriu nao podem ser
aplicadas.")
Else
For i = 0 To DateDiff(DateInterval.Day, data_inicio, data_fim)
Try
Dash_Grafico_detalhado.Chart_capacidade.Series("Corte").Points.Clear()
Dash_Grafico_detalhado.Chart_capacidade.Series("Ligada").Points.Clear()
Dash_Grafico_detalhado.Chart_capacidade.Series("Setup").Points.Clear()
Dash_Grafico_detalhado.Chart_capacidade.Series("Parada").Points.Clear()
Dash_Grafico_detalhado.Chart_capacidade.Series("Justificada").Points.Clear()
Try
connection.Close()
connection.Open()
If ds.Tables.Contains("Dados") = True Then
ds.Tables("Dados").Clear()
End If
SQL = "SELECT * from " & Cod_maquina.Text & " where Inicio between '" &
CDate(data_inicio.date).ToString("yyyy-MM-dd 00:00:00") & "' AND '" &
CDate(data_fim.Date).ToString("yyyy-MM-dd 23:59:59") & "' "
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(ds, "Dados")
connection.Close()
Catch ex As Exception
MsgBox(ex.Message)
End Try
Dash_Grafico_detalhado.Chart_capacidade.Series("Corte").YValueType =
DataVisualization.Charting.ChartValueType.Time
Dash_Grafico_detalhado.Chart_capacidade.Series("Ligada").YValueType =
DataVisualization.Charting.ChartValueType.Time
Dash_Grafico_detalhado.Chart_capacidade.Series("Setup").YValueType =
DataVisualization.Charting.ChartValueType.Time
Dash_Grafico_detalhado.Chart_capacidade.Series("Parada").YValueType =
DataVisualization.Charting.ChartValueType.Time
Dash_Grafico_detalhado.Chart_capacidade.Series("Justificada").YValueType =
DataVisualization.Charting.ChartValueType.Time
For j = 0 To ds.Tables("Dados").Rows.Count - 1
linha = CDate(ds.Tables("Dados").Rows(j).Item("Inicio"))
If j = ds.Tables("Dados").Rows.Count - 1 Then
hora_final = DateAndTime.Now
Else
hora_final = ds.Tables("Dados").Rows(j + 1).Item("Inicio")
End If
If ds.Tables("Dados").Rows(j).Item("Estado") = "Corte" Then
If (CDate(ds.Tables("Dados").Rows(j).Item("Fim")).Day -
CDate(ds.Tables("Dados").Rows(j).Item("Inicio")).Day) > 0 Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Corte").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(j).Item("Inicio")),
(CDate(ds.Tables("Dados").Rows(j).Item("Inicio")).Date +
CDate("23:59:59").TimeOfDay))
Dash_Grafico_detalhado.Chart_capacidade.Series("Corte").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(j).Item("Fim")).Date,
CDate(ds.Tables("Dados").Rows(j).Item("Fim")))
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Corte").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(j).Item("Inicio")),
CDate(ds.Tables("Dados").Rows(j).Item("Fim")))
End If
ElseIf ds.Tables("Dados").Rows(j).Item("Estado") = "Ligada" Then

```

```

If (CDate(ds.Tables("Dados").Rows(j).Item("Fim")).Day -
CDate(ds.Tables("Dados").Rows(j).Item("Inicio")).Day) > 0 Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Ligada").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(j).Item("Inicio")),
(CDate(ds.Tables("Dados").Rows(j).Item("Inicio")).Date +
CDate("23:59:59").TimeOfDay))
Dash_Grafico_detalhado.Chart_capacidade.Series("Ligada").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(j).Item("Fim")).Date,
CDate(ds.Tables("Dados").Rows(j).Item("Fim")))
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Ligada").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(j).Item("Inicio")),
CDate(ds.Tables("Dados").Rows(j).Item("Fim")))
End If
ElseIf ds.Tables("Dados").Rows(j).Item("Estado") = "Setup" Or
ds.Tables("Dados").Rows(j).Item("Estado") = "Alarme" Then
If (CDate(ds.Tables("Dados").Rows(j).Item("Fim")).Day -
CDate(ds.Tables("Dados").Rows(j).Item("Inicio")).Day) > 0 Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Setup").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(j).Item("Inicio")),
(CDate(ds.Tables("Dados").Rows(j).Item("Inicio")).Date +
CDate("23:59:59").TimeOfDay))
Dash_Grafico_detalhado.Chart_capacidade.Series("Setup").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(j).Item("Fim")).Date,
CDate(ds.Tables("Dados").Rows(j).Item("Fim")))
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Setup").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(j).Item("Inicio")),
CDate(ds.Tables("Dados").Rows(j).Item("Fim")))
End If
ElseIf ds.Tables("Dados").Rows(j).Item("Estado") = "Parada" Then
If IsDBNull(ds.Tables("Dados").Rows(j).Item("Paragem")) = True Then
If (CDate(ds.Tables("Dados").Rows(j).Item("Fim")).Day -
CDate(ds.Tables("Dados").Rows(j).Item("Inicio")).Day) > 0 Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Parada").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(j).Item("Inicio")),
(CDate(ds.Tables("Dados").Rows(j).Item("Inicio")).Date +
CDate("23:59:59").TimeOfDay))
Dash_Grafico_detalhado.Chart_capacidade.Series("Parada").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(j).Item("Fim")).Date,
CDate(ds.Tables("Dados").Rows(j).Item("Fim")))
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Parada").Points.AddXY(linha,
CDate(ds.Tables("Dados").Rows(j).Item("Inicio")),
CDate(ds.Tables("Dados").Rows(j).Item("Fim")))
End If
Else
If (CDate(ds.Tables("Dados").Rows(j).Item("Fim")).Day -
CDate(ds.Tables("Dados").Rows(j).Item("Inicio")).Day) > 0 Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Justificada").Points.AddXY(linha
, CDate(ds.Tables("Dados").Rows(j).Item("Inicio")),
(CDate(ds.Tables("Dados").Rows(j).Item("Inicio")).Date +
CDate("23:59:59").TimeOfDay))
Dash_Grafico_detalhado.Chart_capacidade.Series("Justificada").Points.AddXY(linha
, CDate(ds.Tables("Dados").Rows(j).Item("Fim")).Date,
CDate(ds.Tables("Dados").Rows(j).Item("Fim")))
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Justificada").Points.AddXY(linha
, CDate(ds.Tables("Dados").Rows(j).Item("Inicio")),
CDate(ds.Tables("Dados").Rows(j).Item("Fim")))
End If

```

```

End If
End If
Dash_Grafico_detalhado.Chart_capacidade.ChartAreas(0).AxisY.Maximum =
CDate("23:59:59").ToOADate
Dash_Grafico_detalhado.Chart_capacidade.ChartAreas(0).AxisY.Interval = 1
Next
Catch ex As Exception
MsgBox(ex.Message)
End Try
Next
Update()
'Adicionar o grafico ao Dash detalhado da maquina
With Dash_Grafico_detalhado
.Name = "Dash_grafico_detalhado"
.Size = New Size(1665, 90 * DateDiff(DateInterval.Day, data_inicio, data_fim))
End With
Dash_Grafico_detalhado.Chart_capacidade.Size = New
Size(Flow_layout_graficos.Size.Width, 90 * DateDiff(DateInterval.Day,
data_inicio, data_fim))
Flow_layout_graficos.Controls.Add(Dash_Grafico_detalhado)
End If
Else
'Analisar as datas introduzidas
data_inicio = data_1.Value
data_fim = data2.Value
If DateDiff(DateInterval.Day, data_inicio, data_fim) < 0 Then
'Mostrar mensagem de erro de datas
MsgBox("As datas que selecionou nao sao compativeis!", MsgBoxStyle.Critical,
"Erro!")
ElseIf DateDiff(DateInterval.Day, data_inicio, data_fim) > 0 Then
'Verificar se as datas fazem semanas completas, a data inicial tem que se o
primeiro dia da semana e a data final tem que ser o ltimo dia da semana
'Alterar a data inicial
Dim dial = DateTime.Parse(data_1.Value)
Dim dayDiff As Integer = dial.DayOfWeek - DayOfWeek.Monday
Dim monday As Date = dial.AddDays(-dayDiff)
data_1.Value = monday
'Alterar a data Final
Dim dia2 = DateTime.Parse(data2.Value)
Dim dayindex As Integer = dia2.DayOfWeek
If dayindex < DayOfWeek.Sunday Then
dayindex += 7 'Monday is first day of week, no day of week should have similar
index
End If
dayDiff = 7 - dayindex
Dim sunday As Date = dia2.AddDays(dayDiff)
If sunday < Today Then
data2.Value = sunday
End If
'Atualizar os graficos
Try
SQL = "Select * from Tab_comp where Maquina = '" & Cod_maquina.Text & "' and
Data BETWEEN '" & DateTime.Parse(monday).ToString("yyyy-MM-dd") & "' AND '" &
DateTime.Parse(DateAdd(DateInterval.Day, (1), sunday)).ToString("yyyy-MM-dd") &
"' order by Data asc "
'correr codigo generico para a semana
Tracker_variaveis_maquina: Try
connection.Close()
connection.Open()
If dt.Tables.Contains("Tracker_Capacidade") = True Then
dt.Tables("Tracker_Capacidade").Clear()

```

```

End If
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dt, "Tracker_Capacidade")
connection.Close()
Catch ex As Exception
MsgBox(ex.Message)
GoTo Tracker_variaveis_maquina
End Try
Catch ex As Exception
MsgBox(ex.Message)
End Try
Try
dash_grafico_semanal.Chart_week.Series("Corte").Points.Clear()
dash_grafico_semanal.Chart_week.Series("Parada").Points.Clear()
dash_grafico_semanal.Chart_week.Series("Setup").Points.Clear()
dash_grafico_semanal.Chart_week.Series("Ligada").Points.Clear()
For i = 0 To dt.Tables("Tracker_Capacidade").Rows.Count - 1
'Obter dados
semana = DatePart(DateInterval.WeekOfYear,
CDate(dt.Tables("Tracker_Capacidade").Rows(i).Item("Data")).Date,
FirstDayOfWeekValue:=2, FirstWeekOfYearValue:=2)
If semana <> semana_old And IsNothing(semana_old) = False Then
Corte_perc = Corte_seg * 100 / tempo_total
Ligada_perc = Ligada_seg * 100 / tempo_total
Setup_perc = Setup_seg * 100 / tempo_total
Parada_Perc = Parada_seg * 100 / tempo_total
Justificada_Perc = Justificada_seg * 100 / tempo_total
dash_grafico_semanal.Chart_week.Series("Corte").Points.AddXY(Semana_Old,
Math.Round(Corte_Perc, 1))
dash_grafico_semanal.Chart_week.Series("Ligada").Points.AddXY(Semana_Old,
Math.Round(Ligada_Perc, 1))
dash_grafico_semanal.Chart_week.Series("Setup").Points.AddXY(Semana_Old,
Math.Round(Setup_Perc, 1))
dash_grafico_semanal.Chart_week.Series("Parada").Points.AddXY(Semana_Old,
Math.Round(Parada_Perc, 1))
Corte_Perc = 0
Ligada_perc = 0
Setup_perc = 0
Parada_Perc = 0
Justificada_Perc = 0
Corte_Seg = 0
Ligada_seg = 0
Setup_seg = 0
Parada_seg = 0
Justificada_seg = 0
tempo_total = 0
semana_old = semana
End If
'LER TODOS OS VALORES DA LINHA E ADICIONAR AO VALOR TOTAL
Corte_Seg = Corte_Seg +
dt.Tables("Tracker_Capacidade").Rows(i).Item("Corte_segundo")
Setup_Seg = Setup_Seg +
dt.Tables("Tracker_Capacidade").Rows(i).Item("Setup_segundo")
Ligada_Seg = Ligada_Seg +
dt.Tables("Tracker_Capacidade").Rows(i).Item("Ligada_segundo")
Parada_seg = Parada_seg +
dt.Tables("Tracker_Capacidade").Rows(i).Item("Parada_segundo")
tempo_total = Corte_Seg + Setup_Seg + Ligada_Seg + Parada_seg + Justificada_seg
Semana_Old = semana
Next
Corte_Perc = Corte_Seg * 100 / tempo_total

```



```

Ligada_Perc = Ligada_Seg * 100 / tempo_total
Setup_Perc = Setup_Seg * 100 / tempo_total
Parada_Perc = Parada_seg * 100 / tempo_total
Justificada_Perc = Justificada_seg * 100 / tempo_total
dash_grafico_semanal.Chart_week.Series("Corte").Points.AddXY(Semana_Old,
Math.Round(Corte_Perc, 1))
dash_grafico_semanal.Chart_week.Series("Ligada").Points.AddXY(Semana_Old,
Math.Round(Ligada_Perc, 1))
dash_grafico_semanal.Chart_week.Series("Setup").Points.AddXY(Semana_Old,
Math.Round(Setup_Perc, 1))
dash_grafico_semanal.Chart_week.Series("Parada").Points.AddXY(Semana_Old,
Math.Round(Parada_Perc, 1))
Corte_Perc = 0
Ligada_Perc = 0
Setup_Perc = 0
Parada_Perc = 0
justificada_Perc = 0
Corte_Seg = 0
Ligada_Seg = 0
Setup_Seg = 0
Parada_seg = 0
Justificada_seg = 0
tempo_total = 0
dash_grafico_semanal.Chart_week.ChartAreas(0).RecalculateAxesScale()
Catch ex As Exception
MsgBox(ex.Message)
End Try
'Retirar o grafico diario
Flow_layout_graficos.Controls.Clear()
Refresh()
'Adicionar novo grafico
With dash_grafico_semanal
.Name = "Dash_grafico_Semanal"
.Size = New Size(Flow_layout_graficos.Size.Width - 10,
Flow_layout_graficos.Size.Height - 10)
End With
' dash_grafico_semanal.Chart_week.Size = New
Size(Flow_layout_graficos.Size.Width, Flow_layout_graficos.Size.Height)
dash_grafico_semanal.Chart_week.ChartAreas(0).RecalculateAxesScale()
Flow_layout_graficos.Controls.Add(dash_grafico_semanal)
dash_grafico_semanal.Visible = True
End If
End If
Me.Cursor = Cursors.Default
Refresh()
End Sub
'sub rotina de apoio ao automatismo
Public Sub Tracker_var()
Try
Tracker_var: connection.Close()
Try
connection.Open()
If dt.Tables.Contains("Semaforo") Then
dt.Tables("Semaforo").Clear()
End If
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dt, "Semaforo")
connection.Close()
Catch ex As Exception
MsgBox(ex.Message)
GoTo Tracker_var

```

```

End Try
Catch ex As Exception
MsgBox(ex.Message)
End Try
End Sub
'sub rotina de apoio ao automatismo
Public Sub Tracker_var_percentagem()
Try
Tracker_var: connection.Close()
Try
connection.Open()
If dt.Tables.Contains("Percentagem") Then
dt.Tables("Percentagem").Clear()
End If
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dt, "Percentagem")
connection.Close()
Catch ex As Exception
MsgBox(ex.Message)
GoTo Tracker_var
End Try
Catch ex As Exception
MsgBox(ex.Message)
End Try
End Sub
Private Sub corteSwitch1_ValueChanged(sender As Object, e As EventArgs) Handles
Corte_switch.ValueChanged
If Corte_switch.Value = True Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Corte").Enabled = True
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Corte").Enabled = False
End If
Dash_Grafico_detalhado.Chart_capacidade.ChartAreas(0).RecalculateAxesScale()
End Sub
Private Sub Parada_switch_ValueChanged(sender As Object, e As EventArgs) Handles
Parada_switch.ValueChanged
If Parada_switch.Value = True Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Parada").Enabled = True
Dash_Grafico_detalhado.Chart_capacidade.Series("Justificada").Enabled = True
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Parada").Enabled = False
Dash_Grafico_detalhado.Chart_capacidade.Series("Justificada").Enabled = False
End If
Dash_Grafico_detalhado.Chart_capacidade.ChartAreas(0).RecalculateAxesScale()
End Sub
Private Sub Setup_switch_ValueChanged(sender As Object, e As EventArgs) Handles
Setup_switch.ValueChanged
If Setup_switch.Value = True Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Setup").Enabled = True
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Setup").Enabled = False
End If
Dash_Grafico_detalhado.Chart_capacidade.ChartAreas(0).RecalculateAxesScale()
End Sub
Private Sub Ligada_switch_ValueChanged(sender As Object, e As EventArgs) Handles
Ligada_switch.ValueChanged
If Ligada_switch.Value = True Then
Dash_Grafico_detalhado.Chart_capacidade.Series("Ligada").Enabled = True
Else
Dash_Grafico_detalhado.Chart_capacidade.Series("Ligada").Enabled = False
End If

```

```

Dash_Grafico_detalhado.Chart_capacidade.ChartAreas(0).RecalculateAxesScale()
End Sub
Private Sub Print_Click(sender As Object, e As EventArgs) Handles Print.Click
'Fazer Reset ao Print Form
If Print_Maquina.WindowState = FormWindowState.Maximized Or
Print_Maquina.WindowState Or FormWindowState.Normal Or Print_Maquina.WindowState
Or FormWindowState.Minimized Then
Print_Maquina.Flow_layout_graficos.Controls.Clear()
Print_Maquina.Close()
Refresh()
End If
'Passar os dados para o form de impressao
'Imagem
Print_Maquina.PictureBox1.ImageLocation = PictureBox1.ImageLocation
'Descricao da maquina
Print_Maquina.Cod_maquina.Text = Cod_maquina.Text
'Data de Inicio e Fim
Print_Maquina.data_1.Value = data_1.Value
Print_Maquina.data2.Value = data2.Value
'Percentagens de trabalho
Print_Maquina.Valor_atual.Text = Gauge_atual.Value.ToString & "%"
Print_Maquina.Valor_24h.Text = gauge_24.Value.ToString & "%"
Print_Maquina.Valor_7d.Text = Gauge_7d.Value.ToString & "%"
Print_Maquina.Valor_30d.Text = Gauge_30d.Value.ToString & "%"
Print_Maquina.Valor_90d.Text = Gauge_90d.Value.ToString & "%"
Print_Maquina.Valor_365d.Text = Gauge_365d.Value.ToString & "%"
'Estado mostrados
Print_Maquina.Corte_switch.Value = Corte_switch.Value
Print_Maquina.Parada_switch.Value = Parada_switch.Value
Print_Maquina.Setup_switch.Value = Setup_switch.Value
Print_Maquina.Ligada_switch.Value = Parada_switch.Value
'Grafico
If tipo_grafico.Value = True Then
Dash_Grafico_detalhado.Chart_capacidade.Size = New
Size(Print_Maquina.Flow_layout_graficos.Size.Width,
Print_Maquina.Flow_layout_graficos.Height)
Print_Maquina.Flow_layout_graficos.Controls.Add(Dash_Grafico_detalhado)
Else
dash_grafico_semanal.Chart_week.Size = New
Size(Print_Maquina.Flow_layout_graficos.Size.Width,
Print_Maquina.Flow_layout_graficos.Size.Height)
Print_Maquina.Flow_layout_graficos.Controls.Add(dash_grafico_semanal)
End If
'Limitar o grafico a um espaco
Print_Maquina.Data_impresso.Text = "Impresso a " & Now.Date
Print_Maquina.Show()
Refresh()
If Not Inicio.Inicio_print.Text = "Multiplo" Then
If MsgBox("Deseja Imprimir esta janela ?", MsgBoxStyle.YesNo, "Imprimir Janela")
= MsgBoxResult.Yes Then
'Definicde impressao
Dim largura
Dim altura
altura = Print_Maquina.Size.Height
largura = Print_Maquina.Size.Width
Print_Maquina.WindowState = FormWindowState.Normal
Print_Maquina.PrintForm1.PrinterSettings.PrintToFile = False
Print_Maquina.PrintDialog1.PrinterSettings.DefaultPageSettings.Margins.Left = 0
Print_Maquina.PrintDialog1.PrinterSettings.DefaultPageSettings.Margins.Right = 0
Print_Maquina.PrintDialog1.PrinterSettings.DefaultPageSettings.Margins.Bottom =
0

```

```

Print_Maquina.PrintDialog1.PrinterSettings.DefaultPageSettings.Margins.Top = 50
Print_Maquina.PrintDialog1.PrinterSettings.DefaultPageSettings.Landscape = True
Print_Maquina.PrintDialog1.ShowDialog()
Refresh()
Print_Maquina.PrintForm1.PrinterSettings =
Print_Maquina.PrintDialog1.PrinterSettings
Print_Maquina.PrintForm1.Print()
Else 'Caso nao queira imprimir a janela
'Passar o grafico novamente para a janela normal do tracker
If tipo_grafico.Value = True Then
Me.Flow_layout_graficos.Controls.Add(Dash_Grafico_detalhado)
Else
dash_grafico_semanal.Chart_week.Size = New Size(Flow_layout_graficos.Size.Width
- 10, Flow_layout_graficos.Size.Height - 10)
Me.Flow_layout_graficos.Controls.Add(dash_grafico_semanal)
End If
'Adjustar as medidas ja nova janela do tracker
Atualizar.PerformClick()
'Fechar a janela de impressao
Print_Maquina.Close()
End If
End If
End Class

```

Anexo H - Automatismo de recolha de informação desenvolvido para comandos Selca (instancia para máquinas de fresagem normal)

```

Public Class Tracker_Heidenhain_tnc640_V8
'declaracao das variaveis necessarias ao automatismo
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
reiniciar:
Try
'cada maquina integrada no automatismo e sujeita a este ciclo
For Each maq_log In list_maquina
Thread.Sleep(5000)
Try
If IsNothing(list_logfile) = False Then
list_logfile.clear()
End If
If IsNothing(list_log) = False Then
list_log.clear()
End If
Catch ex As Exception
MsgBox(ex.Message)
End Try
'Ler todas as variaveis dessa maquina
SQL = "Select * from Var_selca where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
Var_logfile_old = dt.Tables("Tracker_variaveis").Rows(0).Item("Logfile_old")
diretorio_logfile = dt.Tables("Tracker_variaveis").Rows(0).Item("Diretorio_CNC")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
empresa = dt.Tables("Tracker_variaveis").Rows(0).Item("Empresa")
dsc_controlador = dt.Tables("Tracker_variaveis").Rows(0).Item("Dsc_controlador")
End If
ReadAgain: Try
If File.Exists(diretorio_logfile) = True Then
'fazer update
SQL = "update Var_selca set Ligacao = 1 where Maquina = '" & maq_log & "';
update Var_selca_backup set Ligacao = 1 where Maquina = '" & maq_log & "'"
Tracker_escrever()

```

```

list_logfile = File.ReadLines(diretorio_logfile).ToList
Else
'fazer update
SQL = "update Var_selca Set Ligacao = 0 where Maquina = '" & maq_log & "'";
update Var_selca_backup Set Ligacao = 0 where Maquina = '" & maq_log & "'";
Tracker_escrever()
End If
Catch ex As Exception
GoTo ReadAgain
End Try
nr_old = Var_logfile_old
linhas()
Next
GoTo reiniciar
Catch ex As Exception
MsgBox(ex.Message)
End Try
End Sub
'Rotina de analise dos ficheiros de registo de atividades
Sub linhas()
Try
For i = 1 To list_logfile.count - 1
linha = list_logfile(i)
nrr = linha.substring(linha.length - 21, 17)
nrr = nrr.replace("_", "")
nr = nrr
'Se nr do ficheiro em analise e superior ao guardado na BD o ficheiro e
analizado
If nr > nr_old Then
dir = linha.replace("D:\", dsc_controlador)
list_log = File.ReadLines(dir).ToList
For n = 0 To list_log.count - 1
linhaa = list_log(n)
'Inicio de corte
If linhaa.contains("MSG_SYS;Scrive;Execu") And linhaa.contains("ao em curso")
Then
SQL = "Select * from Var_selca where Maquina = '" & maq_log & "'";
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
Var_logfile_old = dt.Tables("Tracker_variaveis").Rows(0).Item("Logfile_old")
diretorio_logfile = dt.Tables("Tracker_variaveis").Rows(0).Item("Diretorio_CNC")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
empresa = dt.Tables("Tracker_variaveis").Rows(0).Item("Empresa")
dsc_controlador = dt.Tables("Tracker_variaveis").Rows(0).Item("Dsc_controlador")
inicio_corte = dt.Tables("Tracker_variaveis").Rows(0).Item("Inicio_corte")
End If
Try
'Caso seja a primeira linha de corte
If IsDBNull(inicio_corte) = True Then
time_inic = Convert.ToDateTime(linhaa.substring(0, 19))
If IsDBNull(ult_data) = False Then
'Tempo que a maquina esteve parada. (Diferenca entre a hora de paragem de corte
ate ao momento)
estado = "Parada"
time_stop = DateDiff(DateInterval.Second, ult_data, time_inic)
'Escrever na base de dados de log file
SQL = "INSERT INTO Log_File_selca ([Maquina], [Data_entrada], [Data_saida],
[Tempo_corte], [Estado]) values( '" & maq_log & "', '" &
CDate(ult_data).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
CDate(time_inic).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & time_stop & "', '" &
estado & "') "

```

```

Tracker_escrever()
'Escrever na base de dados da maquina
data_dif = DateDiff(DateInterval.Second, CDate(ult_data).Date,
CDate(time_inic).Date)
inicio = ult_data
fim = time_inic
divisao_dias()
End If
estado = "Corte"
'Escrever na base de dados os valores
SQL = "update Var_selca set Inicio_corte = '" &
CDate(time_inic).ToString("yyyy/MM/dd HH:mm:ss") & "', Estado = '" & estado &
"', Ult_data = '" & CDate(time_inic).ToString("yyyy/MM/dd HH:mm:ss") & "' where
Maquina = '" & maq_log & "'; update Var_selca_backup set Inicio_corte= '" &
CDate(time_inic).ToString("yyyy/MM/dd HH:mm:ss") & "', Estado = '" & estado &
"', Ult_data = '" & CDate(time_inic).ToString("yyyy/MM/dd HH:mm:ss") & "' where
Maquina = '" & maq_log & "' "
Tracker_escrever()
End If
Catch ex As Exception
MsgBox("Erro 1.20.30: " & ex.Message)
End Try
End If
'Final de corte
If (linhaa.contains("MSG_SYS;Scrive;Execu") And linhaa.contains("ao terminata"))
Or (linhaa.contains("MSG_SYS;Scrive;Execu") And linhaa.contains("ao
interrompida")) Then
'Selecionar principais informacoes do equipamento
Try
SQL = "Select * from Var_selca where Maquina = '" & maq_log & "'"
Tracker_var()
If dt.Tables("Tracker_variaveis").Rows.Count <> 0 Then
Var_logfile_old = dt.Tables("Tracker_variaveis").Rows(0).Item("Logfile_old")
diretorio_logfile = dt.Tables("Tracker_variaveis").Rows(0).Item("Diretorio_CNC")
ult_data = dt.Tables("Tracker_Variaveis").Rows(0).Item("Ult_data")
empresa = dt.Tables("Tracker_variaveis").Rows(0).Item("Empresa")
dsc_controlador = dt.Tables("Tracker_variaveis").Rows(0).Item("Dsc_controlador")
inicio_corte = dt.Tables("Tracker_variaveis").Rows(0).Item("Inicio_corte")
End If
time_fim = Convert.ToDateTime(linhaa.substring(0, 19))
If IsDBNull(inicio_corte) = False Then
Try
estado = "Corte"
tempo_corte = DateDiff(DateInterval.Second, inicio_corte, time_fim)
'Escrever para o Logfile
SQL = "INSERT INTO Log_file_selca ([Maquina], [Data_entrada], [Data_saida],
[Tempo_corte], [Estado]) values('" & maq_log & "', '" &
CDate(inicio_corte).ToString("yyyy/MM/dd HH:mm:ss") & "', '" &
CDate(time_fim).ToString("yyyy/MM/dd HH:mm:ss") & "', '" & tempo_corte & "', '"
& estado & "'"")
Tracker_escrever()
'Escrever na base de dados da maquina
data_dif = DateDiff(DateInterval.Second, CDate(inicio_corte).Date,
CDate(time_fim).Date)
inicio = inicio_corte
fim = time_fim
divisao_dias()
estado = "Parada"
'Apagar as variaveis
SQL = "update Var_selca set Ult_data = '" & CDate(time_fim).ToString("yyyy-MM-dd
HH:mm:ss") & "', Inicio_corte = NULL, Estado = '" & estado & "' where Maquina =

```

```

'" & maq_log & "'; update Var_selca_backup set Ult_data = '" &
CDate(time_fim).ToString("yyyy-MM-dd HH:mm:ss") & "', Inicio_corte = NULL,
Estado = '" & estado & "' where Maquina = '" & maq_log & ""
Tracker_escrever()
Catch ex As Exception
MsgBox("Erro 1.20.140: " & ex.Message)
End Try
End If
Catch ex As Exception
MsgBox("erro 1.20.180: " & ex.Message)
End Try
End If
'Proxima linha
Next
nr_old = nr
'numero do ficheiro e guardado e registado na bd
SQL = "update Var_selca set Logfile_old = '" & nr_old & "' where Maquina = '" &
maq_log & "'; update Var_selca_backup set Logfile_old = '" & nr_old & "' where
Maquina = '" & maq_log & ""
Tracker_escrever()
End If
'proxima maquina
Next
Catch ex As Exception
MsgBox(ex.Message)
End Try
End Sub
'Sub rotinas de apoio a recolha de informacao
End Class

```

Anexo I - Desenvolvimentos para comandos TNC 640 para recolha de programas NC

```

'String que identifica inicio de prog NC
If linha.contains("OpenProgram") Then
Try
'nome do programa
nc1 = InStrRev(linha, "\")
nc2 = Microsoft.VisualBasic.Strings.Right(linha, nc1)
nc3 = InStr(nc2, ",")
programa_nc = Mid(nc2, 1, nc3)
linha_next_nc = list_logfile(i + 1)
'Hora de inicio
horai = Convert.ToDateTime(Mid(linha_next_nc, 45, 8)).TimeOfDay
datai = Convert.ToDateTime(Mid(linha_next_nc, 55, 10)).Date
time_prog = datai + horai
'Registar informaco es
SQL = "INSERT INTO Programa_nc_heid([Programa_nc], [Maquina], [Finalizado],
[Iniciado], [Segundo_total]) values('" & programa_nc & "', '" & maq_log & "', '
', '" & CDate(time_prog).ToString("yyyy-MM-dd HH:mm:ss") & "', ' ')"
Tracker_escrever()
Catch ex As Exception
'MsgBox(ex.Message)
End Try
End If

```

Anexo J - Desenvolvimentos do *Front End* para programas NC e mensagens de erro

```

'Prog NC + Mensagens Erro
Private Sub RadButton1_Click_1(sender As Object, e As EventArgs) Handles
RadButton1.Click

```

```

'Limpar a grafica da janela
Flow_layout_graficos.Controls.Clear()
Me.Cursor = Cursors.WaitCursor
'leitura das datas pretendidas pelo utilizador
data_inicio = data_1.Value
data_fim = data2.Value
'Prog NC
Try
connection.Close()
connection.Open()
dtable.Clear()
'fazer limpeza da listbox se tiver items
If ListBox1.Items.Count > 2 Then
ListBox1.Items.Clear()
ListBox1.Items.Add("Programas NC")
ListBox1.Items.Add(" ")
End If
'selecionar informacoes da BD
SQL = "SELECT Programa_nc, Iniciado, Finalizado from Programas_nc where Iniciado
between '" & CDate(data_inicio.date).ToString("yyyy-MM-dd 00:00:00") & "' AND '"
& CDate(data_fim.Date).ToString("yyyy-MM-dd 23:59:59") & "' AND Maquina like '"
& Inicio.Inicio_maquina.Text & "' UNION ALL SELECT Programa_nc, Iniciado,
Finalizado from Programa_nc_heid where Iniciado between '" &
CDate(data_inicio.date).ToString("yyyy-MM-dd 00:00:00") & "' AND '" &
CDate(data_fim.Date).ToString("yyyy-MM-dd 23:59:59") & "' AND Maquina like '" &
Inicio.Inicio_maquina.Text & "' order by Iniciado asc"
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dtable)
connection.Close()
'Escrever programas NC na listbox
nc = " "
For Each tempRow In dtable.Rows
If tempRow("Programa_nc") <> nc Then
ListBox1.Items.Add("Iniciado em: " & (tempRow("Iniciado")) & "      Finalizado em:
" & (tempRow("Finalizado")) & "      Programa: " & (tempRow("Programa_nc")))
End If
nc = tempRow("Programa_nc")
Next
Catch ex As Exception
MsgBox(ex.Message)
End Try
'Erros
Try
connection.Close()
connection.Open()
dtable2.Clear()
'fazer limpeza da listbox se tiver items
If ListBox2.Items.Count > 2 Then
ListBox2.Items.Clear()
ListBox2.Items.Add("Mensagens de Erro")
ListBox2.Items.Add(" ")
End If
'selecionar informacoes da BD
SQL = "SELECT Cod_erro, Mensagem, Data from Mensagem_erro where Data between '"
& CDate(data_inicio.date).ToString("yyyy-MM-dd 00:00:00") & "' AND '" &
CDate(data_fim.Date).ToString("yyyy-MM-dd 23:59:59") & "' AND Maquina like '" &
Inicio.Inicio_maquina.Text & "' order by data asc"
da.SelectCommand = New SqlCommand(SQL, connection)
da.Fill(dtable2)
connection.Close()
err = " "

```



```

'Escrever mensagens de erro na listbox
For Each tempRow In dtable2.Rows
If tempRow("Mensagem") <> err Then
ListBox2.Items.Add("Data: " & (tempRow("Data")) & "      Cód: " &
(tempRow("Cod_erro")) & "      Mensagem: " & (tempRow("Mensagem")))
End If
err = tempRow("Mensagem")
Next
Catch ex As Exception
MsgBox(ex.Message)
End Try
ListBox1.Visible = True
ListBox2.Visible = True
Me.Cursor = Cursors.Default
End Sub

```