

MATHEUS HEITOR SILVA FERREIRA

**DESENVOLVIMENTO DE UM PROGRAMA SUPERVISÓRIO PARA
A OPERAÇÃO AUTOMATIZADA DE DUAS UNIDADES DE
PROCESSAMENTO DE PINOS POR ATRITO**



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA

2019

MATHEUS HEITOR SILVA FERREIRA

**DESENVOLVIMENTO DE UM PROGRAMA SUPERVISÓRIO PARA
A OPERAÇÃO AUTOMATIZADA DE DUAS UNIDADES DE
PROCESSAMENTO DE PINOS POR ATRITO**

Projeto de Fim de Curso apresentado ao curso de graduação em Engenharia Mecatrônica da Universidade Federal de Uberlândia, como parte dos requisitos para a obtenção do título **BACHAREL em ENGENHARIA MECATRÔNICA**.

Orientadora: Profa. Dr.-Ing. Vera Lúcia D. S. Franco

UBERLÂNDIA - MG

2019

AGRADECIMENTOS

Agradeço ao engenheiro Dênis Soares de Freitas por compartilhar uma parte do seu vasto conhecimento técnico na área de instrumentação e controle, me auxiliando durante todo o desenvolvimento do trabalho. Graças à dedicação genuína dele para com a Engenharia Elétrica e Mecatrônica, mudei boa parte dos meus conceitos e aprendi bastante a respeito da nossa profissão.

Agradeço também à professora Dr.-Ing. Vera Lúcia Donizeti de Sousa Franco pela sua dedicação com a Faculdade de Engenharia Mecânica (FEMEC), e por me guiar e orientar durante os meus últimos períodos na academia. A ela, os meus sinceros agradecimentos por ter me dado a oportunidade de aprender no Laboratório de Tecnologia em Atrito e Desgaste (LTAD).

Ao professor Rafael Ariza Gonçalves, eu agradeço pelo auxílio teórico e prático com relação às UPPAs e, também, por providenciar os materiais necessários para realização dos ensaios. Sua ajuda foi muito importante na execução deste trabalho.

Agradeço também todos os professores, engenheiros, técnicos e demais colaboradores do LTAD pelo auxílio e apoio nas atividades do projeto e, por agregarem conhecimento na formação dos estudantes que trabalham no laboratório. Em especial ao professor Sinésio Domingues Franco; aos engenheiros Henry Fong Wang, Fernando Buiatti Rodrigues, Raphael Rezende Pires, Juliano Oséias de Moraes, Rodrigo Freitas da Silva Alvarenga e Douglas Ferreira Cabral.

À minha família, no qual buscamos o caminho da verdade, mesmo sabendo das consequências vindouras e da guerra que enfrentaremos. Sempre no caminho do amor e da honra.

À UFU e LTAD, por toda a estrutura, corpo técnico e auxílio financeiros disponibilizados, indispensáveis para o desenvolvimento de um trabalho de qualidade.

À Petróleo Brasileiro S.A (PETROBRAS), pelo apoio financeiro.

FERREIRA, M. H. S. **Desenvolvimento de um programa supervisorio para a operação automatizada de duas unidades de processamento de pinos por atrito.** 2019. 81 f. Trabalho de Conclusão de Curso, Universidade Federal de Uberlândia, Uberlândia.

Resumo

Esse trabalho apresenta o desenvolvimento de um programa supervisorio utilizando o *software* LabVIEW® para operar de forma otimizada duas unidades de Processamento de Pinos por Atrito (UPPA), com vistas a atender diversas situações de experimentos e ensaios referentes à técnica de processamento de pinos por atrito. O programa abrange desde a definição dos parâmetros de ensaio, comunicação entre os equipamentos, coleta de dados experimentais até o tratamento e apresentação dos resultados em forma de tabelas e gráficos. Adicionalmente, o programa foi desenvolvido utilizando programação modular, de tal forma que os projetos a serem desenvolvidos no Laboratório de Tecnologia em Atrito e Desgaste (LTAD) mantenham uma identidade visual padronizada. Foi desenvolvido também rotinas de segurança para a realização de ensaios de reparo por atrito, e finalmente foram realizados ensaios para verificar a funcionalidade do programa desenvolvido.

Palavras-chave: Reparo por Atrito, Programa Supervisorio, Interface de Usuário, Programação Gráfica.

FERREIRA, M. H. S. **Development of a supervisory software for automated operation of friction pillar processing machines**. 2018. 81 p. Undergraduate thesis, Universidade Federal de Uberlândia, Uberlândia.

Abstract

This work presents the development of a supervisory program using LabVIEW® software in order to optimally operate Friction Hydro Pillar Processing (FHPP) machines to attend various conditions of tests and experiment the test conditions. The program covers from definition of test parameters, communication between equipment, experimental data acquiring to data treatment and presentation of results in form of tables and graphs. In addition, the program was developed using modular programming, in a way that future projects developed by the *Laboratório de Tecnologia em Atrito e Desgaste* (LTAD) keeps a standardized visual identity. Safety routines to accomplishment of tests were also developed and, finally, tests were performed to verify the functionality of the software.

Keywords: FHPP, Supervisory Control, User Interface, Graphical Programming.

LISTA DE FIGURAS

Figura 1: Processo de reparo por costura para o reparo de trincas. (a) Cavidade passante (HWANG, 2010). (b) Cavidade não passante (DE LIMA FILHO, 2008).....	6
Figura 2: Esquema ilustrativo do processo FTPW (MARTINAZZI, CHLUDZINSKI, <i>et al.</i> , 2014)	8
Figura 3: Esquema ilustrativo do processo FPPW (HWANG, 2010)	8
Figura 4: Divisão das fases durante o processo de pinos por atrito. As imagens à esquerda representam o esquema apresentado por Meyer (2002), enquanto os gráficos à direita são dados provenientes de um ensaio realizado na UPPA3.....	11
Figura 5: Esquema ilustrativo da Unidade de Processamento de Pinos por Atrito 1 (UPPA1) (HWANG, 2010)	14
Figura 6: Esquema hidráulico de uma UPPA (FORMOSO, 2012).	15
Figura 7: Estrutura física do sistema de controle de força e rotação das UPPAs (FREITAS, 2014).....	16
Figura 8: Esquema ilustrativo da Unidade de Processamento de Pinos por Atrito 2 (UPPA2)	18
Figura 9: Sensores responsáveis pela medição de força, torque e comprimento de queima da UPPA2	19
Figura 10: Unidade de Processamento de Pinos por Atrito 3 (UPPA3).....	20
Figura 11: A) Ilustração da Unidade de Processamento de Pinos por Atrito X (UPPAX), B) Cilindro de reparo (HWANG, 2010).	21
Figura 12: Cores prediletas da população (HALLOCK, 2003).....	24
Figura 13: Transformação da identidade visual do sistema operacional <i>Windows®</i> ao longo do tempo (MUSGRAVE, 2012).....	25

Figura 14: Estrutura de comunicação CLP-Computador utilizada nas UPPAs (adaptador de FREITAS, 2014).....	26
Figura 15: Comparação de estruturas do código “Hello world!”, a) é o algoritmo escrito na linguagem C, b) é o algoritmo na linguagem gráfica LabVIEW®.....	27
Figura 16: Componentes principais do <i>software</i> LabVIEW®. A) Painel Frontal, B) Diagrama de blocos.....	28
Figura 17: Comparação entre algoritmo que gera e mostra um vetor contendo dez elementos, a) Linguagem baseada em texto, b) Linguagem gráfica	29
Figura 18: Fórmula matemática complexa representada em: a) linguagem textual b) linguagem gráfica.....	30
Figura 19: Esquema simplificado de aquecimento indutivo da UPPA3.	32
Figura 20: Interface Homem Máquina (IHM) utilizada para configuração de parâmetros de ensaio (BORGES, 2011).	35
Figura 21: Interface gráfica do primeiro programa supervisorio da UPPA3 (FORMOSO, SANTOS, <i>et al.</i> , 2009).....	35
Figura 22: Interface do segundo programa supervisorio desenvolvido para a realização de reparos por atrito na UPPA3 (FREITAS, 2014)	36
Figura 23: Estrutura básica do programa, composta pela divisão da interface gráfica em cinco painéis, onde cada painel é responsável por uma sub-rotina específica do programa supervisorio.....	39
Figura 24: Principais objetos esperados dentro de cada painel do programa.	40
Figura 25: Estrutura de transferência de dados entre as sub-rotinas do programa. O quadro representa um subVI, as setas indicam se a informação está entrando ou saindo. O índice <i>n</i> indica um notificador, o índice <i>q</i> indica uma fila	44
Figura 26: Estrutura de transferência de dados entre as rotinas mostradas de forma alternativa. A base da seta indica o emissor dos dados, enquanto a ponta da seta indica o destinatário	45

Figura 27: Fluxograma do funcionamento geral do programa supervisorio.....	49
Figura 28: Funções usadas para leitura dos dados capturados do CLP.	50
Figura 29: Exemplos de objetos desenvolvidos para a interface gráfica do programa supervisorio.	51
Figura 30: Etapas executadas no processo de reparo por atrito (FREITAS, 2014)...	55
Figura 31: Para definir qual equipamento – UPPA2 ou UPPA3 – será utilizado na realização dos ensaios, deve-se alterar a posição das válvulas destacadas.	56
Figura 32: Amostra após a realização do processo de reparo por atrito (RODRIGUES, 2016).	58
Figura 33: Interface do programa supervisorio SUPPA32.	59
Figura 34: Sub-rotina “Ensaio” do programa SUPPA32, onde o usuário informa os parâmetros de ensaio.	60
Figura 35: No menu “Análise” é possível carregar ensaios e comparar resultados ..	61
Figura 36: Janela do programa SUPPA32 durante a realização de ensaios. Os gráficos exibem os dados relativos à região de interesse do ensaio.	64
Figura 37: Resultados para o primeiro ensaio de reparo por atrito realizado na UPPA3. Os gráficos mostram todos os dados coletados durante o ensaio, sem o filtro que exclui dados coletados fora da região de interesse.....	65
Figura 38: Resultados de rotação e comprimento de queima do segundo ensaio de reparo por atrito realizado na UPPA3.....	67
Figura 39: Resultados de rotação e comprimento de queima do segundo ensaio de reparo por atrito realizado na UPPA3.....	68

LISTA DE TABELAS

Tabela 1: Comparativo da faixa de operação das versões do equipamento UPPA.	22
Tabela 2: Descrição do conteúdo e função das filas e notificadores do programa....	47
Tabela 3: Resultados obtidos na gravação de vídeos com duração de 30 segundos.	63

LISTA DE SÍMBOLOS

- η – Velocidade rotacional do pino (rpm)
- P – Pressão aplicada ao pino no ponto de contato (MPa)
- F – Força axial de soldagem aplicada ao pino (kN)
- Δl – Comprimento de queima do pino (mm)
- t – Tempo (s)
- $\Delta l/t$ – Taxa de consumo do pino (mm/s)
- N – Potência consumida durante a soldagem (kW)
- V – Velocidade linear do pino (m/s)
- n – Notificador do LabVIEW®
- q – Fila do LabVIEW®

LISTA DE ABREVIATURAS

CLP – Controlador Lógico Programável
FHPP – *Friction Hydro Pillar Processing*
FPPW – *Friction Pull Plug Welding*
FPS – Quadros Por Segundo
FTPW - *Friction Tapered Plug Welding*
IHM – Interface Homem Máquina
LTAD – Laboratório de Tecnologia em Atrito e Desgaste
LVDT – *Linear Variable Differential Transformer*
MB – Metal base
NI – *National Instruments®*
OLE – *Object Linking and Embedding*
OPC – *Open Platform Communications* (antigamente *OLE for Process Control*)
PID – Proporcional Integral Derivativo
PPA – Processamento de Pinos por Atrito
RN – Rede Neural
subVI - Um VI chamado a partir de outro VI
UFU – Universidade Federal de Uberlândia
UI – *User Interface*
UPPA – Unidade de Processamento de Pinos por Atrito
VI – *Virtual Instrument* (Instrumento Virtual)
ZTA – Zona Termicamente Afetada

SUMÁRIO

1.	INTRODUÇÃO	1
2.	REVISÃO BIBLIOGRÁFICA.....	6
2.1.	Reparo por atrito	6
2.2.	Unidade de Processamento de Pinos por Atrito (UPPA).....	14
2.3.	Histórico das versões do equipamento	17
2.4.	Projeto de programas.....	22
2.5.	Programação com base no <i>software</i> LabVIEW®	26
3.	ASPECTOS CONSTRUTIVOS DA UNIDADE DE PROCESSAMENTO DE PINOS DE ATRITO (UPPA)	31
3.1.	Unidade hidráulica	31
3.2.	Sistema de aquecimento.....	32
3.3.	Instrumentação, controle e aquisição de dados	33
3.4.	Programa supervisorio UPPA3	34
4.	DESENVOLVIMENTO DO PROGRAMA SUPERVISÓRIO	38
4.1.	Estrutura do programa desenvolvido	38
4.2.	Comunicação entre as partes do programa	43
4.3.	Fluxograma de funcionamento do programa desenvolvido.....	48
4.4.	Aquisição de dados	50
4.5.	Interface gráfica	51
4.6.	Rotinas de segurança	53
4.7.	Captura de vídeo.....	55
4.8.	Procedimento de ensaio.....	55
5.	RESULTADOS E DISCUSSÕES	59
5.1.	Programa supervisorio	59
5.2.	Testes das rotinas do programa.....	62

5.3.	Gravação de vídeos	63
5.4.	Ensaio de reparo por atrito	64
5.5.	Validação do programa pelos operadores do equipamento	69
6.	CONCLUSÃO	70
7.	TRABALHOS FUTUROS	72

CAPÍTULO I

1. INTRODUÇÃO

Os recursos energéticos são o foco dos interesses estatais, gerando disputas econômicas e geopolíticas desde a primeira revolução industrial. Apesar dos avanços tecnológicos, o principal recurso da matriz energética é o mesmo desde a segunda revolução industrial em 1850, o petróleo. O petróleo é uma das matérias-primas mais importantes da civilização moderna. É utilizado como fonte de energia e seus derivados são transformados em plástico, borracha sintética, tintas, corantes, adesivos, solventes, detergentes, explosivos, produtos farmacêuticos e de cosmética, entre outras muitas aplicações.

Em 2017 o Brasil se tornou o maior produtor de petróleo da América Latina, sendo o décimo maior produtor mundial (ANP, 2017). Por esse motivo, reduzir os custos de extração, transporte, armazenamento, processamento e/ou distribuição do petróleo resulta na economia de recursos e aumento dos lucros, lastreando a posição do país como grande produtor dentro desse mercado. No Brasil, a maior parte das reservas de petróleo está em campos marítimos, o que tem levado atividades de perfuração a atingirem profundidades cada vez maiores. A Petróleo Brasileiro S.A (PETROBRAS) espera uma produção total de óleo e gás de 3,41 milhões de boed¹ em 2021 (PETROBRAS, 2019).

A extração em área marítima é feita com a utilização de plataformas que servem como base de instalação de equipamentos de extração, conhecidas como plataformas *offshore*. O custo de operação e manutenção dessas estruturas é elevado, ainda mais quando se trata de aplicações em águas profundas. Em média, o custo de manutenção *offshore* é vinte vezes mais caro que a manutenção em estruturas terrestres (CAIXETA, 2011).

¹ Barris de óleo equivalente por dia

As estruturas *offshore* e navios estão constantemente sendo submetidos a condições extremas no ambiente que operam. Devido ao movimento das ondas, essas estruturas sofrem carregamentos e tensões significativas, que podem levar ao surgimento de trincas de fadiga, principalmente em conexões soldadas. Tempestades frequentes em alto mar também intensificam os efeitos da fadiga, produzindo trincas e reduzindo a vida útil das estruturas.

A soldagem convencional a arco elétrico é o principal método empregado no reparo de trincas em estruturas produtivas *offshore*. Porém esse método de reparo e manutenção possui desvantagens que limitam sua aplicação na indústria petroquímica. Nesses casos, como se trabalha com temperaturas elevadas, a solda apresenta problemas como uma grande Zona Termicamente Afetada (ZTA). Quando se utiliza essa técnica em local com alto risco de explosão, esta é feita em ambiente submerso, como é o caso de muitos componentes da indústria do petróleo. Trabalhando com altas temperaturas num ambiente submerso, obtêm-se um resfriamento rápido, que pode causar tensões residuais e problemas como trincas. Em equipamentos de elevada espessura e difícil soldabilidade, como os tanques de armazenamento de água e óleo em navios de produção de petróleo, é comum encontrar trincas decorrentes da flutuação das tensões ou cargas de fadiga que podem levar inclusive à ruptura de componentes (PRADO, 2018).

Além disso, existem problemas relacionados à fusão e solidificação, como a adsorção de gases da atmosfera pela poça de fusão, levando à formação de fases frágeis de baixa tenacidade à fratura, além da fragilização pelo hidrogênio e a formação de poros no interior do cordão de solda. Diante desse contexto, surge a necessidade de estudar e desenvolver novas técnicas de reparo a fim de minimizar ou eliminar os problemas encontrados nos métodos tradicionais de soldagem (BORGES, 2011). Técnicas que buscam diminuir a quantidade de calor aportado, diminuir a formação e adsorção de gases residuais e reduzir a periculosidade envolvida de aplicação em áreas classificadas.

Como tentativa de solucionar os problemas da solda convencional, foi desenvolvida uma técnica de reparo alternativa, conhecida como reparo por atrito. A solda por atrito é um processo de união no estado sólido, onde é produzida a união metalúrgica do material através da rotação ou movimento relativo entre as partes sob forças compressivas ou de tração. Como vantagens dessa técnica, citam-se: a) pouco influenciada pelo ambiente marinho, b) pode ser parcialmente ou totalmente

automatizada, eliminando-se assim a necessidade de mergulhadores, e c) baixos riscos quando da sua utilização em áreas classificadas ou em ambientes explosivos (SOUZA, 2006).

Derivada do processo de solda por atrito, surgiu a técnica *Friction Hydro Pillar Processing* (FHPP) ou Processamento de Pinos por Atrito (PPA), que consiste no preenchimento de um furo cilíndrico não passante através de um pino consumível, submetido a uma velocidade de rotação e pressionado contra o furo através da ação de forças axiais. O calor resultante do processo leva à redução do limite de escoamento do material, facilitando o fluxo plástico que se desloca ao longo da direção axial do pino, levando à união metalúrgica entre as partes e ao consequente preenchimento do furo (PIRES, 2007).

A pesquisa em soldagem por atrito na Universidade Federal de Uberlândia (UFU) iniciou no ano de 2006, com o projeto e construção do equipamento Unidade de Processamento de Pinos por Atrito (UPPA1), apoiado pela Petrobras (SOUZA, 2006). Atualmente o LTAD possui três equipamentos UPPA em operação, cada um com características diferentes, abrangendo uma ampla faixa de parâmetros de soldagem como rotação, força, torque e temperatura da amostra. Um desses equipamentos pode, adicionalmente, realizar ensaios em ambiente submerso.

Ao longo dos anos, estudos relacionados ao processo de soldagem por atrito nas unidades UPPA levaram a melhorias no sistema de instrumentação e controle desses equipamentos. Borges (2011) realizou a instrumentação da UPPA3 usando um Controlador Lógico Programável (CLP) para executar o acionamento e comandos, porém a aquisição de dados era realizada através de placas de aquisição e o processo para realização de ensaios dependia de uma Interface Homem Máquina (IHM).

Caixeta (2011) otimizou o sistema de controle da UPPA2, trabalhando com um cartão PID (Proporcional, Integral e Derivativo) analógico para controle de força e um cartão PID digital para controle de velocidade de rotação, diminuindo o tempo necessário para se atingir o *setpoint* de força e aumentando o valor de força que poderia ser aplicado no reparo por atrito, sem que o equipamento sofresse travamento do sistema de rotação. Adicionalmente implementou no sistema o reparo por atrito utilizando dois estágios, no qual os parâmetros do processo (força e comprimento e queima) são alterados durante a realização do ensaio.

Formoso (2012) trabalhou na UPPAX, estruturando os circuitos eletrônicos e instrumentação do equipamento, bem como no ajuste dos parâmetros de controle e

desenvolvimento de um programa para exibição dos gráficos e acompanhamento dos parâmetros ao longo do ensaio. A aquisição de dados usava placas de aquisição PCI em conjunto com o CLP e a interação com o operador passou a ser exclusivamente no computador, sem necessidade de IHM, através de um programa supervisorio desenvolvido em LabVIEW®.

Freitas (2014) realizou a reestruturação elétrica e de programação da UPPA2 e UPPA3, eliminando do circuito o conversor de frequência em tensão, o circuito subtrator e os cartões de controle PID, implementado o sistema de instrumentação e controle no qual este passou a ser totalmente controlado pelo CLP. Foi modificado o sistema de controle de rotação, no qual foi implementado o controlador do tipo RN (Rede Neural) para o controle de rotação, associado ao controlador PID para controle de força, além disso foi desenvolvido um sistema antitravamento do conjunto de rotação durante os ensaios. Um novo sistema de rápida despressurização da linha hidráulica contribuiu para prevenir a existência de golpes de aríete que poderiam danificar o equipamento. O computador passou a ter como função apenas a interação com o operador do equipamento, o envio dos parâmetros de ensaio ao CLP, exibição de gráficos e salvamento de dados, sendo desenvolvido um programa supervisorio utilizando LabVIEW®.

Assim, o objetivo principal do presente trabalho é produzir um novo sistema de supervisão e aquisição de dados para a UPPA2 e UPPA3, de tal forma que o mesmo possa ser adotado como interface e estrutura padrão para todos os outros processos realizados dentro do LTAD, que utilizem programação em LabVIEW®.

Para melhor clareza e compreensão do trabalho desenvolvido, este texto está organizado da seguinte forma:

O Capítulo II apresenta uma revisão bibliográfica sobre o processo de solda por atrito e do equipamento de reparo por atrito, bem como a linguagem de programação utilizando o *software* LabVIEW®, utilizada no programa supervisorio desenvolvido neste trabalho.

O Capítulo III apresenta a descrição do equipamento para o qual foi desenvolvido o programa supervisorio.

O Capítulo IV apresenta as técnicas empregadas no desenvolvimento do programa supervisorio, bem como os procedimentos experimentais para a realização dos ensaios.

O Capítulo V apresenta os resultados obtidos, juntamente com a sua discussão, em função da sequência de realização de testes e dos ensaios.

O Capítulo VI apresenta as principais conclusões.

O Capítulo VII apresenta sugestões para trabalhos futuros e potenciais aprimoramentos no programa desenvolvido.

O Capítulo VIII apresenta as referências bibliográficas citadas neste trabalho.

CAPÍTULO II

2. REVISÃO BIBLIOGRÁFICA

2.1. Reparo por atrito

O processo de soldagem por atrito utilizado nos equipamentos UPPA foi desenvolvido pelo “*The Welding Institute*” (Cambridge, Inglaterra) (THOMAS, 1997), onde a trinca pode ser reparada removendo-se a mesma através de cavidades individuais (cavidade fechada ou aberta), seguido do preenchimento sequencial de cada cavidade através do processamento termomecânico de pinos de aço pela soldagem por atrito, conforme ilustrado na Figura 1.

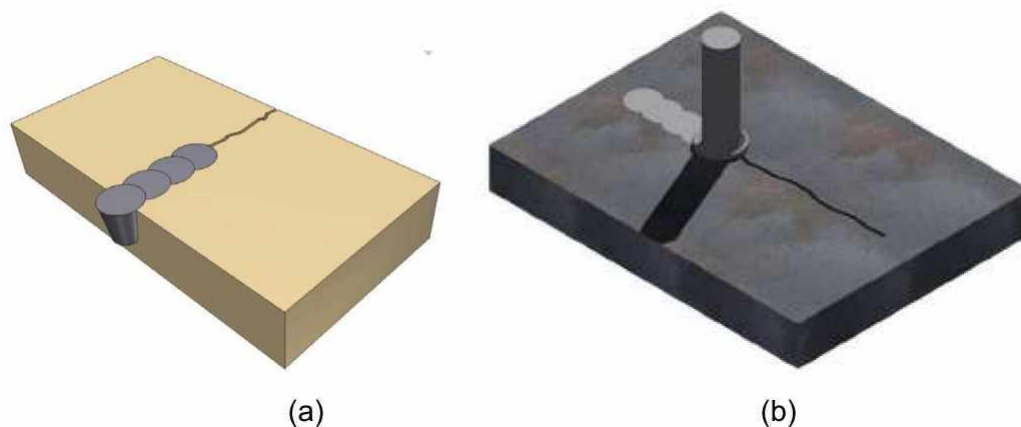


Figura 1: Processo de reparo por costura para o reparo de trincas. (a) Cavidade passante (HWANG, 2010). (b) Cavidade não passante (DE LIMA FILHO, 2008).

Nesse processo, a soldagem se dá pela introdução coaxial de um pino consumível, com formato cilíndrico ou cônico (“*tapered*”), ambos com extremidade arredondada na região de início da soldagem, onde o pino é girado em alta rotação e submetido simultaneamente a esforços de compressão contra a superfície da cavidade. Através do calor gerado entre as superfícies em contato, o material do pino

assume uma condição viscoplástica, possibilitando o desenvolvimento de um fluxo de material em elevada temperatura ao longo da interface da junta soldada e promovendo, conseqüentemente, o preenchimento da cavidade e sua união metalúrgica com o metal-base (MB). Esta técnica de preenchimento de cavidades pelo processamento de pinos por atrito é denominada por FHPP (*"Friction Hydro-Pillar Processing"*), pino cilíndrico, ou FTPW (*"Friction Tapered Plug Welding"*), pino com formato cônico. Já a soldagem denominada costura (*"stitch welding"*), se dá pela sobreposição de pinos processados (PAES, 2013).

Os processos FHPP ou FTPW consistem no preenchimento de um furo não passante por intermédio da introdução de um pino submetido a uma velocidade de rotação e esforços de compressão contra a cavidade do furo. Devido ao seu curto ciclo térmico e baixo aporte de calor, o processo pode ser empregado em linhas de transporte de petróleo, gás e metanol, sem a necessidade de paradas de produção. Tem-se, assim, um método com elevado potencial de uso na indústria *offshore* em atmosferas explosivas (BLAKEMORE, 1993).

A primeira etapa do processo de reparo por atrito é a abertura de um furo cego (não passante), para remoção do defeito no metal. Em seguida, é realizado o preenchimento desse furo por intermédio da introdução coaxial de um pino (consumível), estando este submetido a uma velocidade de rotação e a esforços de compressão contra a cavidade do furo. A geometria da cavidade deve ser bastante próxima à do pino, sendo o diâmetro da cavidade ligeiramente maior. A força de compressão necessária para a obtenção de uma junta soldada de qualidade pode variar significativamente, de 10-1000 kN, a depender do diâmetro do pino, juntamente com a velocidade rotacional selecionada, do material do pino utilizado, da potência e da rigidez do equipamento de soldagem por atrito. Na técnica FTPW (pinos cônicos), uma parcela da deformação plástica encontra-se nas paredes laterais e não somente ao longo da seção transversal do pino, como é normalmente o caso da técnica FHPP (pinos cilíndricos) (PAES, 2013).

A geometria cilíndrica é em geral preferida para as condições de reparo em estruturas de parede grossa e o perfil cônico para as estruturas de paredes mais finas. Isto porque nas estruturas com espessuras muito elevadas, ter-se-ia excessivo aumento do diâmetro com o uso do pino cônico. Isso demandaria equipamentos de maior potência, tornando o processo inconveniente para aplicações em que o equipamento de soldagem tem que se deslocar até o local de interesse (PAES, 2013).

Devido ao calor gerado pela interação entre as superfícies de contato, o material do pino é mantido em uma condição viscoplástica, permitindo fluxo de material, que se desenvolve ao longo da direção axial do consumível, promovendo-se, assim, o completo preenchimento do furo (PIRES, 2007). Outra característica interessante é que o processo pode ser considerado autolimpante, uma vez que durante o reparo as impurezas tendem a ser eliminadas juntamente com o *flash* ou rebarba. Portanto, eliminam-se algumas partículas que poderiam ser responsáveis pelo aparecimento de descontinuidades na região soldada (CAIXETA, 2011). A Figura 2 ilustra o uso da técnica FTPW.

Uma variante do processamento de pinos por atrito (FHPP ou FTPW) é a técnica *Friction Pull Plug Welding* (FPPW), que consiste em preencher um furo passante através da rotação e tração do pino. Os dois processos diferem, em que o FHPP (ou FTPW) requer um suporte de reação à carga interno, enquanto o FPPW reage externamente à carga. A Figura 3 ilustra o uso da técnica FPPW.

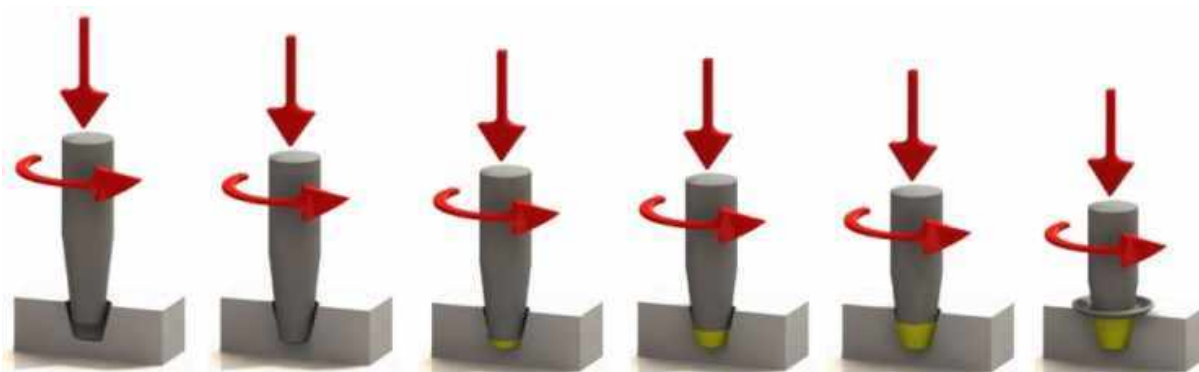


Figura 2: Esquema ilustrativo do processo FTPW (MARTINAZZI, CHLUDZINSKI, *et al.*, 2014)

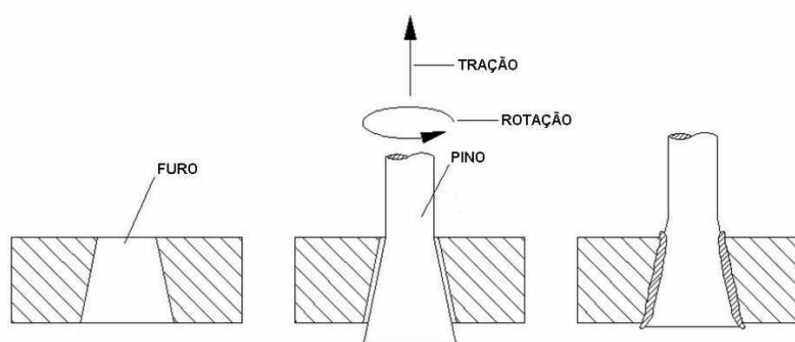


Figura 3: Esquema ilustrativo do processo FPPW (HWANG, 2010)

Por ser um processo realizado no estado sólido, com ausência de material fundido, estas técnicas não apresentam os inconvenientes metalúrgicos apresentados pela soldagem a arco elétrico, e prestando-se otimamente para realização em condições subaquáticas, visto que este processo não sofre influência da pressão ambiente (NIXON, 1986). Os principais parâmetros do processo FTPW/FHPP são:

η – Velocidade rotacional do pino (rpm)

P – Pressão aplicada ao pino no ponto de contato (MPa)

F – Força axial de soldagem aplicada ao pino (kN)

Δl – Comprimento de queima do pino (mm)

t – Tempo de soldagem (s)

$\Delta l/t$ – Taxa de consumo do pino (mm/s)

N – Potência consumida durante a soldagem (kW)

V – Velocidade linear do pino (m/s)

Para η elevado, obtém-se um maior aporte térmico, entretanto leva-se um maior tempo para atingir as condições plásticas ideais devido a ação de polimento entre as superfícies de contato, favorecendo assim a propagação de calor e a intensificação da ZTA (PIRES, 2007). As altas rotações ocasionarão, além do maior volume de material aquecido, uma menor taxa de resfriamento e queda nas propriedades mecânicas, como limite de escoamento e dureza (ELLIS, 1972).

Segundo Paes (2013) os valores de η e P , mesmo previamente fixados, variam instantaneamente ao longo do tempo de soldagem, em função das características geométricas da junta soldada (espessura/largura da chapa, diâmetro do pino/cavidade) e do tipo de material do pino (aços, ligas de níquel, entre outros). Isto porque a resistência torcional a quente do material do pino pode ser ultrapassada ao longo do seu processamento devido às altas temperaturas, elevados tempos de soldagem e alto carregamento axial aplicado ao pino, a depender dos parâmetros fixados de η e P . O valor de P deve ser compatível de modo a não haver nem falta de adesão (P baixo), nem expulsão de material plastificado em excesso pelo anular (P alto).

Segundo Pires (2007), o aumento de F aumenta a força de atrito e, conseqüentemente, o torque resistivo. Assim, em uma mesma velocidade, quanto maior a força, maior será o torque requerido pelo sistema. O parâmetro força axial controla também o gradiente de temperatura na região da solda, a potência requerida

do equipamento e a taxa de queima. No entanto, valores muito elevados de **F** podem conduzir o processo a difíceis condições de controle, conforme pode ser visto no trabalho de Freitas (2014), em vista de elevadas temperaturas locais e das altas taxas de queima. A força de forjamento consiste em aplicação de **F** após a conclusão do reparo. Nesta etapa, a rotação é interrompida e aplica-se uma carga axial de valor igual ou superior à força axial utilizada durante a soldagem, por um tempo pré-determinado. Esta etapa é usualmente aplicada para se obter uma melhoria nas propriedades do material soldado.

O comprimento de queima (Δl) é o parâmetro utilizado para definir a duração de um processo de soldagem por atrito, ou seja, o processo ocorre até que o comprimento de queima da peça seja atingido. Quanto a esse parâmetro, observam-se três comportamentos predominantes durante o processo de reparo por atrito. No início do processo a taxa de queima tende a ser elevada, sendo maior para pinos consumíveis formados por uma ponta cônica, de diâmetro relativamente pequeno. Na fase de processamento a taxa de consumo ou queima do pino pode ser dita constante, pois a área de contato torna-se constante e não há variação na força. Na etapa de forjamento a taxa de queima é baixa ou nula, pois não há deformação no pino e somente uma acomodação do material deformado (CAIXETA, 2011).

Quanto observado ao longo do tempo, o comprimento de queima é chamado taxa de queima ($\Delta l/t$), e esta define a velocidade e consumo das peças durante o processo de soldagem. A taxa de queima é fortemente dependente dos parâmetros η e **F**. A combinação de altas forças axiais e baixas rotações propiciam maiores taxas de queima (MEYER, 2003).

A potência (**N**) depende fortemente da força axial, do torque e da rotação. Para processos com elevadas forças axiais, é necessária uma máquina de soldagem com maior potência e com inércia suficiente para superar os esforços relacionados ao torque resistivo elevado, principalmente em baixas rotações. O diâmetro do pino também deve ser observado, pois pinos de maior diâmetro elevam significativamente o requisito de potência e rigidez do equipamento de soldagem.

Um valor inicial da velocidade linear do pino (**V**) muito baixo não permite uma adesão adequada na interface, principalmente no ponto inicial de torque, momento em que a superfície do metal-base ainda está fria. Portanto, é necessário um valor mínimo dessa velocidade no momento do contato, devendo ser compatível com **P**.

A divisão do processo de soldagem por atrito é uma das formas de descrever os ciclos de soldagem por atrito e os mecanismos relacionados ao processo. A Figura 4 mostra as fases de uma soldagem de atrito convencional, mostrando as quatro fases presentes no processo, comparando o comportamento teórico esperado com gráficos obtidos durante um ensaio realizado na UPPA3.

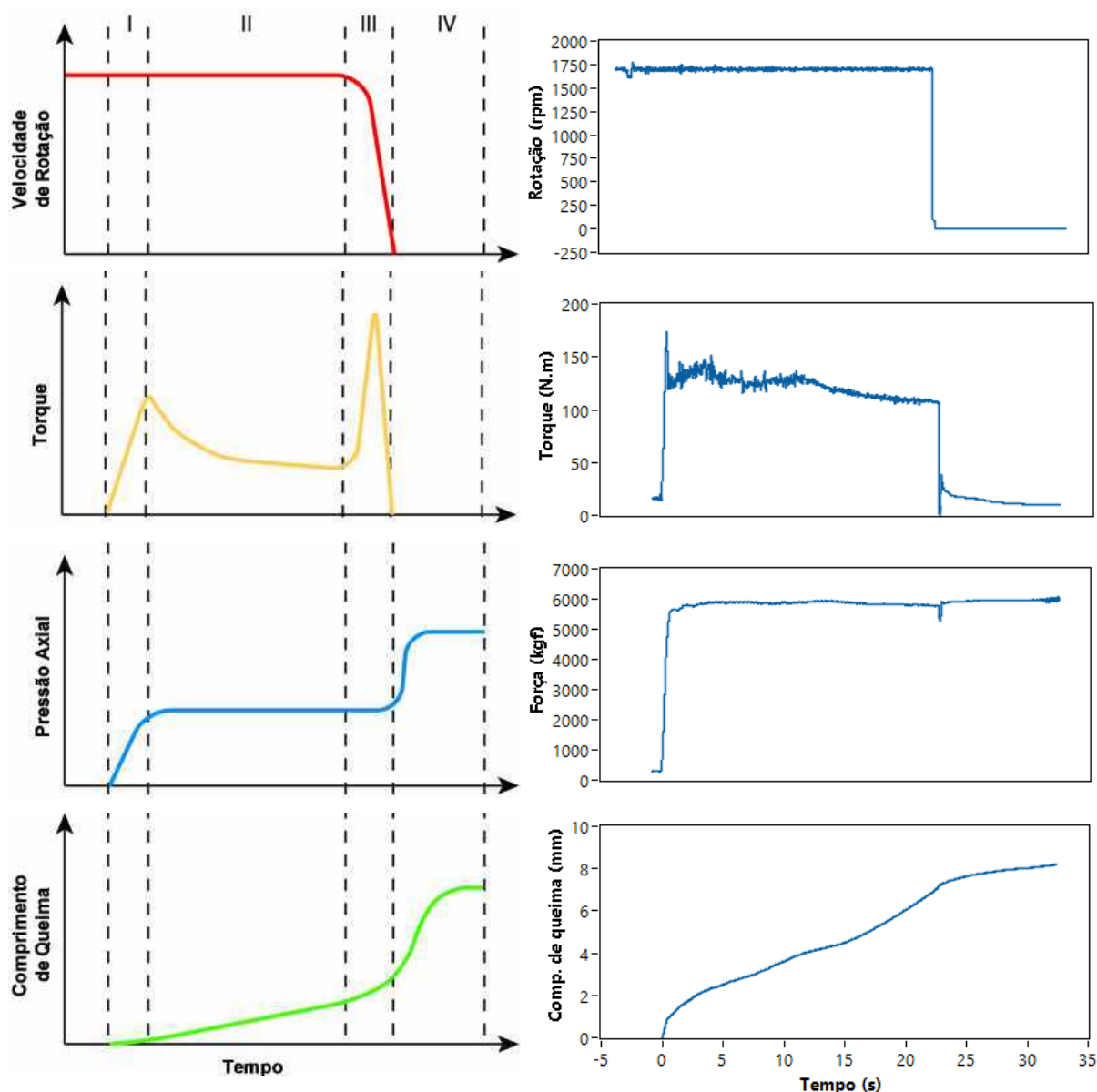


Figura 4: Divisão das fases durante o processo de pinos por atrito. As imagens à esquerda representam o esquema apresentado por Meyer (2002), enquanto os gráficos à direita são dados provenientes de um ensaio realizado na UPPA3.

Na primeira fase (Fase I), denominada de fase de atrito, ocorre a interação entre as superfícies em contato. Nesta fase a área de contato superficial é pequena, e uma severa deformação plástica é imposta devido à aplicação de uma elevada carga axial, resultando no “alisamento ou planificação” das superfícies em contato, seguida por eventos de microadesão (RODRIGUES, 2016). Como resultado desta deformação, novas superfícies são expostas, levando a uma condição de deslizamento a seco (“sem lubrificação”). O aquecimento local na interface surge devido ao cisalhamento das junções adesivas, formando filmes de cisalhamento, que liberam energia na forma de calor. A temperatura pode chegar a valores muito próximas da temperatura de fusão do material. Durante o processamento o calor é dissipado levando à plastificação do material, resultando na destruição das camadas de óxidos e remoção das superfícies contaminadas devido à dinâmica do processo e às altas forças radiais, por isso o processo é dito autolimpante. Segundo Lebedev e Chernenko (citado por RODRIGUES, 2016), este fenômeno inicia-se em uma área anular a aproximadamente dois terços do raio, avançando para o centro da peça e posteriormente para seu exterior, até que em toda área de contato é formado o filme cisalhante, que é então extrudado radialmente. A extração das camadas de óxidos e lubrificantes deixa a superfície limpa, aumentando consideravelmente o coeficiente de atrito e, conseqüentemente, o torque.

Na segunda fase (Fase II), denominada fase de aquecimento, ocorre o aumento da temperatura do material plastificado na forma de anel obtido na Fase I. Esse material altamente plastificado é, então, conduzido para zonas de menor temperatura da área de atrito e, devido ao rápido resfriamento do material, um maior atrito é novamente produzido nestas zonas em particular e a área em forma de anel é ampliada em direção ao centro rotacional (NICHOLAS, 1995). À medida que a temperatura do material e a quantidade de material comprimido aumentam, ocorre a redução da resistência mecânica, sendo o material direcionado para fora da área de atrito formando uma rebarba (*flash*). O processo se repete à medida que a espessura da camada plastificada diminui, fazendo com que as áreas frias entrem em atrito, aquecendo e formando regiões altamente plastificadas. O processo continua enquanto não houver equilíbrio de temperatura e zona plastificada, resultando em material comprimido para formação do *flash*. O equilíbrio destas ações, resulta na saturação do calor na área de atrito e zonas periféricas, fazendo com que o campo de temperatura estabilize, reduzindo as superfícies em contato e o torque resistivo devido

o atrito. Deste ponto em diante tem-se a taxa de queima praticamente constante. A Fase II termina quando o processo alcança o comprimento de queima pré-ajustado. (MEYER, 2003).

Na terceira fase (Fase III), denominada fase de frenagem ou desaceleração, ocorre a diminuição da velocidade rotacional, resultando no aumento da resistência ao cisalhamento devido à redução da taxa de deformação do material, e, conseqüentemente, o aumento considerável do torque, atingindo seu segundo pico (CROSSLAND, 1971). Observando o gráfico de torque na Figura 4, nota-se que o pico de torque obtido nos ensaios da UPPA3 nessa etapa é referente a um “torque resistivo”. Quando a rotação se aproxima de zero, a geração de calor é reduzida bruscamente e, conseqüentemente, a temperatura do material também diminui, provocando um aumento rápido da sua resistência mecânica e fazendo com que o torque gerado pelo próprio processo realize a parada do motor hidráulico da UPPA3. Adicionalmente, contribui para esse pico o fato do motor diesel, que fornece a energia necessária ao funcionamento do sistema de rotação, ser desligado e o sistema de rotação ser despressurizado no momento em que a etapa de forjamento é iniciada, com o objetivo de evitar fortes impactos (golpes de aríete) no motor hidráulico durante a sua parada (FREITAS, 2014).

Na etapa de forjamento (Fase IV) a junta soldada formada é submetida a uma deformação por compressão por uma força axial, a qual ou é mantida igual à da etapa de aquecimento ou pode mesmo assumir valores superiores. Este estágio é considerado como uma etapa simultânea de tratamento térmico e de forjamento da junta soldada, intensificando os mecanismos de adesão e conferindo melhores propriedades de resistência mecânica e tenacidade. O estágio de forjamento tem duração de alguns segundos, após o qual o processo é interrompido e a peça soldada pode ser removida dos pontos de fixação do equipamento de soldagem (PAES, 2013). No ensaio mostrado na Figura 4 (realizado na UPPA3), a força de forjamento possui o mesmo valor da força empregada na etapa de aquecimento.

2.2. Unidade de Processamento de Pinos por Atrito (UPPA)

No LTAD, a exploração da técnica de reparo por atrito iniciou-se com a construção da Unidade de Processamento de Pinos por Atrito (UPPA1), apresentada esquematicamente na Figura 5. Para explorar diferentes faixa de operação, outras versões foram desenvolvidas, UPPA2, UPPA3 e UPPAX, sendo esta última portátil, visando a flexibilidade para aplicações em campo e em ambiente submerso (FREITAS, 2014). Todos esses equipamentos foram construídos com o apoio da Petrobras.

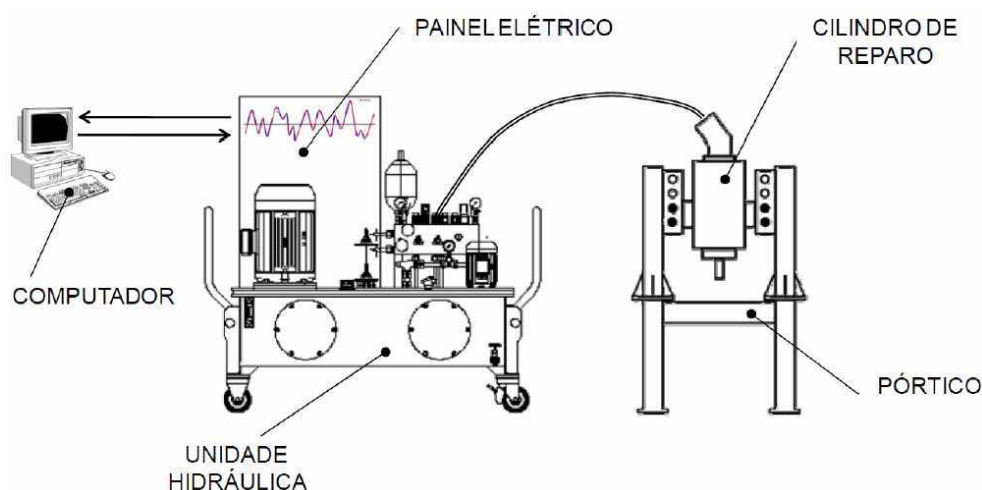


Figura 5: Esquema ilustrativo da Unidade de Processamento de Pinos por Atrito 1 (UPPA1) (HWANG, 2010)

Os componentes fundamentais de uma UPPA são: unidade hidráulica, bloco de válvulas, cilindro de reparo, pórtico de sustentação e o sistema de controle. A unidade hidráulica tem a função de prover energia tanto ao motor hidráulico, quanto à haste hidráulica do cilindro de reparo, possuindo um sistema que monitora a pressão e o controle do fluxo hidráulico demandado durante o processamento do pino consumível (FORMOSO, 2012). O bloco de válvula é constituído por um conjunto de componentes hidráulicos (válvulas servo-solenoides, válvulas direcionais de acionamento elétrico, válvulas limitadoras de pressão, válvulas de retenção e transdutores de pressão) responsáveis pelo controle do fluxo e pressão do óleo que suprem o equipamento de reparo (RODRIGUES, 2016)

A Figura 6 apresenta um esquema universal do sistema hidráulico de uma UPPA. Na figura estão representados os dois conjuntos motobomba principais, sendo um deles responsável pela alimentação do motor hidráulico e implementação da rotação e torque, e o outro pela alimentação do cilindro hidráulico e implementação de deslocamento e força axial no eixo (FORMOSO, 2012).

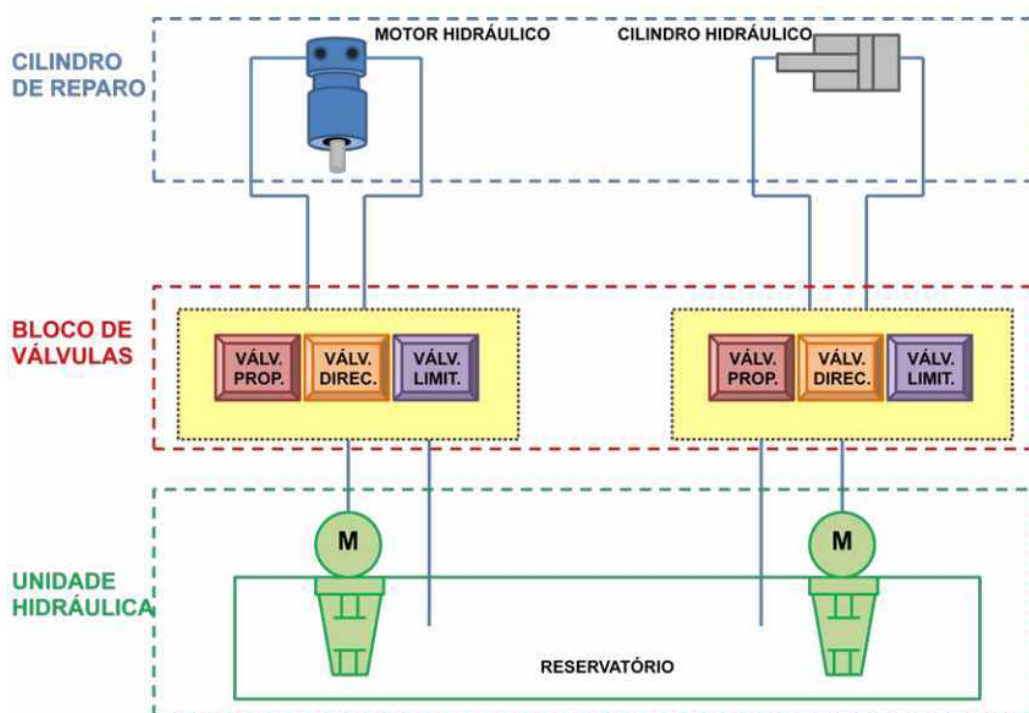


Figura 6: Esquema hidráulico de uma UPPA (FORMOSO, 2012).

Os parâmetros que devem ser controlados durante o processo de reparo por atrito são dois: força e rotação. A força aplicada no pino é controlada a partir da válvula direcional proporcional responsável pelo acionamento do cilindro hidráulico. Já a rotação é controlada pela válvula direcional proporcional responsável pelo acionamento do motor hidráulico (FREITAS, 2014).

A aquisição dos valores medidos pelos sensores é realizada a partir de um CLP, que é responsável por todas as funções de tratamento, conversão dos sinais e lógica de controle. Os dados adquiridos pelo CLP são transferidos para um computador onde está situado o programa supervisor, cujo desenvolvimento é o objetivo desse trabalho. Esse programa, desenvolvido em LabVIEW®, é responsável pela interface com o usuário e pelo salvamento dos dados. Os detalhes do programa supervisor serão discutidos no Capítulo IV. A interconexão entre CLP e computador

é realizada através da arquitetura *Ethernet*, e utiliza o padrão OPC (*Open Platform Communications*) para a interface com o programa supervisor.

A estrutura física do sistema de controle das UPPAs é apresentada esquematicamente na Figura 7, que mostra que a força aplicada pelo cilindro é obtida indiretamente através de transdutores de pressão, que produzem um sinal analógico em tensão proporcional à força axial aplicada. O CLP calcula os ajustes PID baseado na diferença entre os sinais de *setpoint* e *feedback*, enviando à válvula direcional proporcional servo-controlada o sinal de controle.

Já o controle de rotação do motor hidráulico, nada mais é do que um controle de vazão. Isso porque a vazão de óleo que passa pela válvula é a mesma que passa pelo motor e esta é diretamente proporcional à sua rotação (FREITAS, 2014). O valor nominal de vazão da válvula do sistema de rotação é obtido através dos sensores de pressão, pois a vazão nominal da válvula de controle de rotação é proporcional à diferença de pressão por aresta da mesma (BOSCH REXROTH, citado por FREITAS, 2014). A medida de rotação é obtida a partir de um sensor de efeito Hall, que obtém o valor da rotação a partir da frequência de pulsos gerados por esse sensor quando o motor hidráulico é acionado.

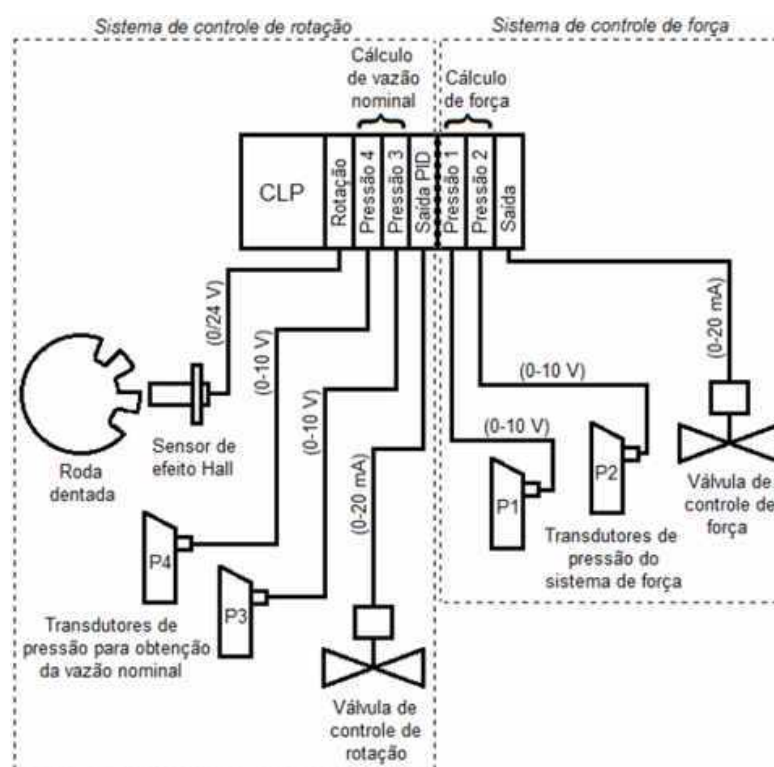


Figura 7: Estrutura física do sistema de controle de força e rotação das UPPAs (FREITAS, 2014).

2.3. Histórico das versões do equipamento

2.3.1. Unidade de Processamento de Pinos por Atrito 1 (UPPA1)

A UPPA1 (Figura 5) consiste em um equipamento de acionamento hidráulico, com cilindro de haste vazada, capaz de processar pinos com força unidirecional de compressão de até 50 kN e rotações de até 8.000 rpm (SOUZA, 2006). A unidade hidráulica é composta por um reservatório com capacidade de 400 litros e três sistemas motobomba. O cilindro de reparo é constituído de um motor hidráulico de pistão axial com torque.

Cada conjunto motobomba tem uma finalidade específica. O primeiro conjunto, com potência de 0,75 KW (1 CV), é responsável pela recirculação e filtragem do óleo. O segundo conjunto motobomba, com potência de 1,5 KW (2 CV), é responsável por fornecer fluído à haste hidráulica com pressão de até 12 MPa, e vazão de 5 litros/min, o que garante uma velocidade máxima de deslocamento da haste de aproximadamente 11 mm/s. O terceiro e último conjunto motobomba tem potência de 75 KW (100 CV), sendo responsável pelo acionamento do motor hidráulico. A vazão é de até 99 litros/min e pressão máxima da ordem de 35 MPa, o que garante um torque máximo no motor hidráulico de 57 Nm (SOUZA, 2006).

A UPPA1 foi utilizada nos trabalhos realizados por Pires (2007) com objetivo de otimizar os parâmetros de processo de reparo por atrito, correlacionados à diferentes tipos de geometrias de pinos e furos, com foco de reparos sem a presença de defeitos. Pires (2007) concluiu que a qualidade do reparo depende fortemente dos parâmetros de processo, sendo a força axial o parâmetro preponderante no tempo de processamento, ou seja, quanto maior for esse parâmetro menor é o tempo de processamento.

A estrutura hidráulica da UPPA1 foi utilizada no equipamento UPPAX, sendo reaproveitados o reservatório de óleo de 400 L, o conjunto motobomba do sistema de rotação, o motor elétrico do sistema de recirculação e o trocador de calor. Todos os outros componentes foram substituídos, inclusive as mangueiras hidráulicas (RODRIGUES, 2016).

2.3.2. Unidade de Processamento de Pinos por Atrito 2 (UPPA2)

A necessidade de se testar diferentes processos e materiais, que demandavam equipamentos de maior potência, levaram ao desenvolvimento da UPPA 2 e, posteriormente, da UPPA 3. Estas duas versões partilham de uma unidade hidráulica em comum, porém com pórticos de características diferentes.

A UPPA2, ilustrada na Figura 8, foi desenvolvida com o objetivo de explorar a influência de outros valores dos parâmetros força axial e rotação no processo de reparo por atrito. Seu projeto é similar à concepção da UPPA1, porém essa versão é mais robusta. A UPPA2 possui capacidade máxima de força axial de 245 kN e rotação de até 2500 rpm (CAIXETA, 2011). Podem-se destacar duas partes principais do cilindro de reparo: motor hidráulico e cilindro hidráulico.

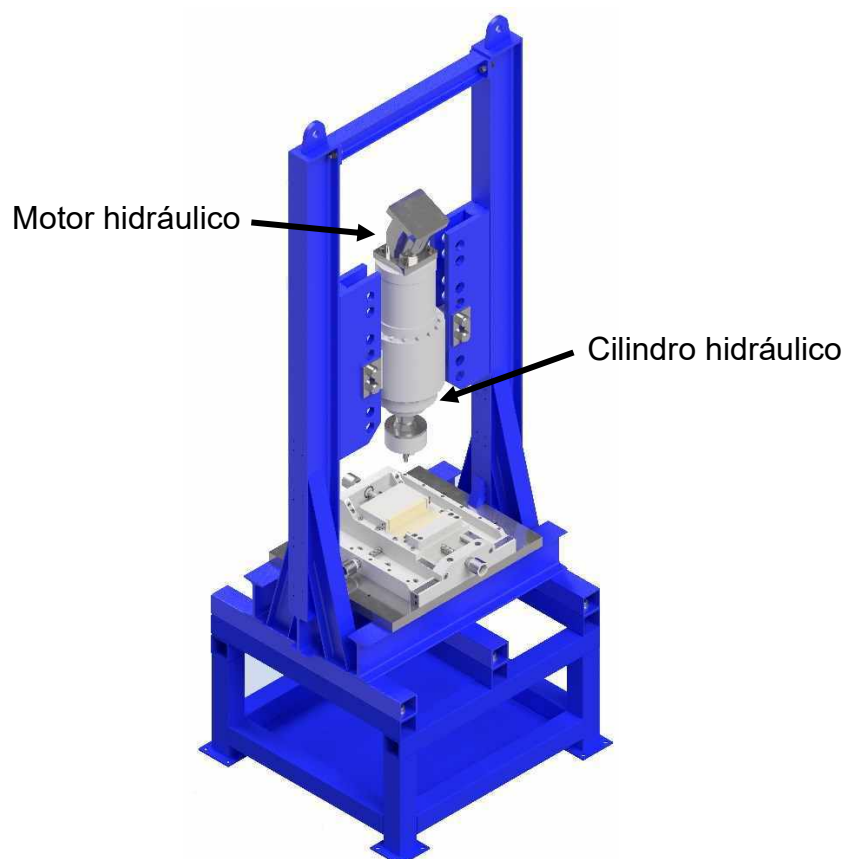


Figura 8: Esquema ilustrativo da Unidade de Processamento de Pinos por Atrito 2 (UPPA2)

O sistema de controle é composto pelos componentes mostrados na Figura 7 e possui, adicionalmente, os seguintes sensores:

- Sensor do tipo LVDT (*Linear Variable Differential Transformer*): Responsável por obter a posição da haste do cilindro. Dessa forma, através do valor de posição é medido o comprimento de queima durante o reparo;
- Sensores indutivos que detectam o fechamento da porta de segurança, garantindo a segurança do equipamento e do operador durante os ensaios;
- Uma câmera USB fixada na estrutura é responsável pela gravação de vídeos durante a realização de ensaios e monitoramento visual do processo.

O valor da força e do torque são obtidos através dos sensores de pressão instalados na cabeça de reparo, conforme mostrado na Figura 9.

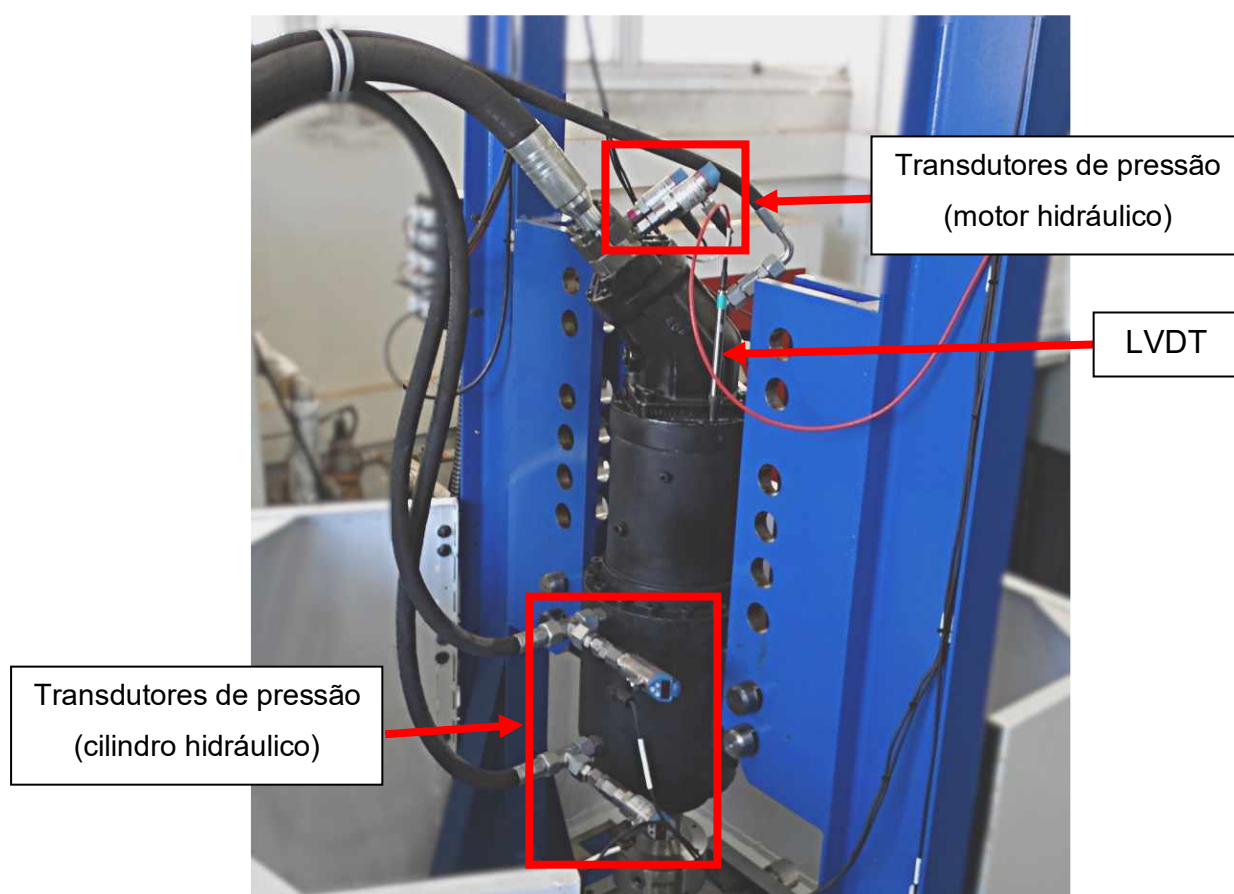


Figura 9: Sensores responsáveis pela medição de força, torque e comprimento de queima da UPPA2

2.3.3. Unidade de Processamento de Pinos por Atrito 3 (UPPA3)

O reparo por atrito com cargas da ordem de algumas dezenas de toneladas exige um sistema mecânico robusto para suportar os esforços e tensões às quais todo o equipamento que realiza o processo fica submetido. Assim, a UPPA3, apresentada na Figura 10, foi projetada e construída para atender às necessidades de aplicação de maiores valores de força axial. Esta versão foi projetada especificamente para realização de ensaios em laboratório (HWANG, 2010).

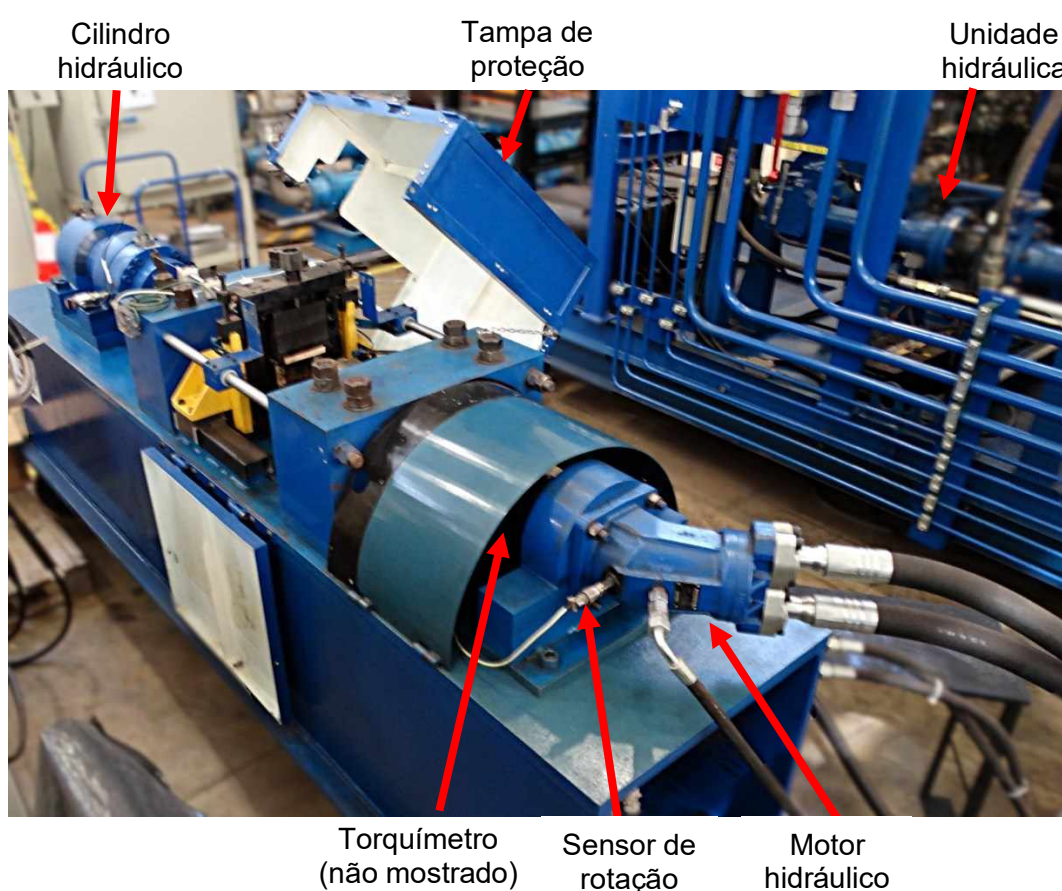


Figura 10: Unidade de Processamento de Pinos por Atrito 3 (UPPA3)

A UPPA3 foi usada nos ensaios de validação do programa supervisor desenvolvido nesse trabalho. Maiores detalhes dessa unidade serão apresentados no próximo capítulo.

2.3.4. Unidade de Processamento de Pinos por Atrito X (UPPAX)

A UPPAX, previamente chamada UPPAIV, foi projetada por Hwang (2010), com o objetivo de levar o equipamento a campo. O cilindro de reparo é portátil, possui uma massa de 15 kg, sendo capaz de atender às operações de soldagem e reparo por atrito através da compressão (FTPW) e tração (FPPW) de pinos consumíveis. O equipamento é hermeticamente selado, sendo adequado para uso em imersão contínua em água do mar a uma profundidade de até 30 metros (3 bar) (RODRIGUES, 2016). A Figura 11 mostra os componentes e o cilindro de reparo portátil do equipamento. A patente da concepção da UPPAX foi requerida em 2007 e concedida em 2014 sob o título "Equipamento para Reparo de Estruturas Metálicas e Métodos de Funcionamento", processo PI 0704190-0, no Instituto Nacional da Propriedade Industrial (INPI) (HWANG, 2010).

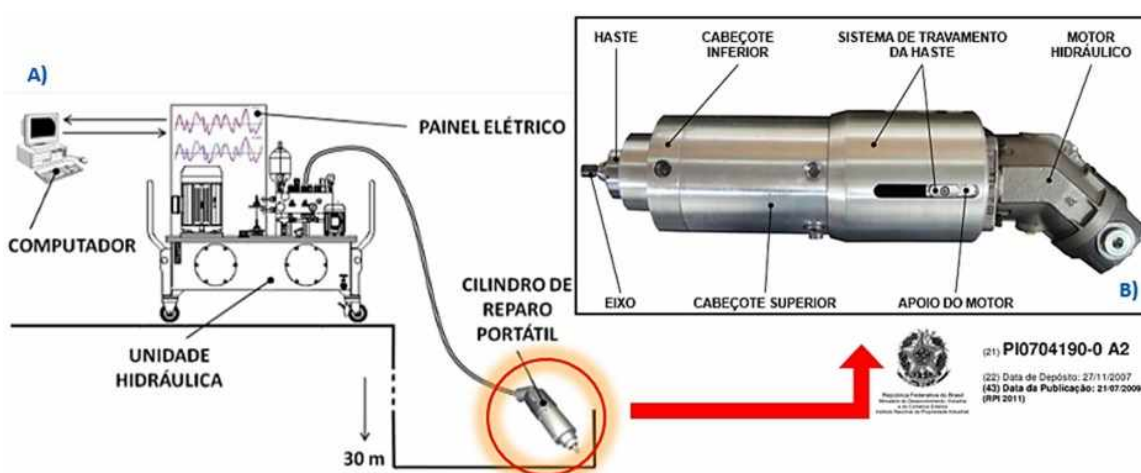


Figura 11: A) Ilustração da Unidade de Processamento de Pinos por Atrito X (UPPAX), B) Cilindro de reparo (HWANG, 2010).

Para atender completamente às operações de reparo por atrito, o equipamento deve ser capaz de realizar as três principais etapas do processo de reparo de trincas por atrito (Furação, Preenchimento e Fresamento) de forma rápida, precisa, confiável e economicamente viável. Rodrigues (2016) trabalhou na reestruturação e modernização da UPPAX, com o objetivo de viabilizar o reparo de trincas, com enfoque em trincas não passantes, que se estendem ao longo de toda, ou quase toda, espessura de chapas de aço de estruturas da indústria de óleo e gás.

O cilindro de reparo é constituído por um motor hidráulico de pistões axiais de 16 cc, capaz de transmitir uma rotação de até 5.500 rpm e torque máximo de 88 Nm. O motor está acoplado a um cilindro hidráulico de haste vazada com capacidade de força axial de 40 kN, sendo seu interior constituído por um conjunto de sistemas rotativos (eixo, rolamentos e acoplamento). O cilindro é composto por dois cabeçotes que contém a haste. Através dos cabeçotes é possível fazer a fixação do cilindro de reparo nas estruturas auxiliares de fixação (RODRIGUES, 2016).

A Tabela 1 apresenta um comparativo da faixa (limite superior) de operação de todas as versões do equipamento UPPA.

Tabela 1: Comparativo da faixa de operação das versões do equipamento UPPA.

DESCRIÇÃO		UPPA2	UPPA3	UPPAX	UPPA1
Cilindro hidráulico	Força axial (kgf)	24.500	40.000	4.000	5.000
	Velocidade desl. máxima (mm/s)	24	11	38	11
Motor hidráulico	Rotação (rpm)	2.500	1.750	5.500	8.000
	Torque máximo (Nm)	500	616	88	57

2.4. Projeto de programas

2.4.1. Interface de usuário

Design de interface é a prática responsável pelo planejamento, desenvolvimento e aplicação de uma solução com o objetivo de facilitar a experiência do usuário e estimular sua interação com um objeto físico ou digital (MAIA, 2016).

A Teoria da Cor é a ciência que lida com a influência das cores nas atividades humanas. As cores podem afetar diferentes grupos e indivíduos de maneiras diferentes (ESCOBAR, 2013). Ajustes simples no matiz ou saturação de uma cor podem contribuir para que ela transmita um sentimento ou outro. Os efeitos das cores podem representar o sucesso ou fracasso de algumas interfaces. De forma geral, as principais cores transmitem os seguintes sentimentos quando utilizadas no design de interfaces.

- **Vermelho:** provoca exaltação, estimula os batimentos cardíacos e estimula tomada de decisão. Em tonalidades mais claras é energético e nas mais escuras transmite poder e elegância. É muito usado em botões para finalização de vendas e na construção de sites relativos à comida. Essa cor é utilizada pelo aplicativo *iFood*™.
- **Laranja:** Em suas tonalidades escuras, pode ser associado com a terra e o outono. Pela sua associação com a mudança de estações, pode representar mudança e movimento em geral. Por associar-se com a fruta de mesmo nome, remete à saúde e vitalidade. É considerado mais amigável e convidativo, atraindo atenção visual sem ser tão esmagador quanto o vermelho. É muito usado em *calls-to-action*, que são links de uma página que levam os usuários a realizar ações ou palavras utilizadas para orientar o usuário, como "saiba mais" e "acesse agora". Essa cor foi utilizada no website *Groovespark*® e é utilizada pela empresa *Nickelodeon*®.
- **Amarelo:** O olho humano é mais sensível à cor amarela que a outras cores. É considerada a cor quente mais clara e enérgica. Amarelos mais suaves são usados como uma cor unissex para bebês e crianças mais novas. Em tonalidade mais claras, transpassa um sentimento calmo de alegria (otimismo, jovial), diferente dos amarelos vibrantes. Amarelos-escuros e puxados para o dourado podem ser usados em projetos que necessitam um senso de permanência, tradição e antiguidade. Essa cor agrada a maioria dos usuários. Essa cor é utilizada para destacar ofertas e liquidações. A empresa *McDonalds*® tem como cores principais o vermelho e o amarelo.
- **Verde:** É a cor mais ligada a temas naturais. Pode representar novos começos, crescimento e abundância. É apropriado para *designs* relativos à riqueza, estabilidade, renovação e natureza. Em tonalidades mais claras é mais energizante e vibrante, os verdes-oliva são mais representativos do mundo natural. Verdes escuros são os mais estáveis e remetem à fortuna. É a cor mais fácil de processamento visual. Essa cor é utilizada pelo *Spotify*™ e em sistemas operacionais Linux como o *OpenSUSE*® e *Manjaro*®.
- **Azul:** O azul pode ter vários significados dependendo de sua tonalidade. Os mais claros são relaxantes e acalmam, os mais vibrantes remetem à refrescância e os escuros são excelentes para sites corporativos ou *designs*

que pretendem transmitir força e confiança. Muito usado em sites de bancos e negócios. É a cor favorita da maioria das pessoas, portanto é uma das cores mais usadas no *design* de interfaces. Essa cor é utilizada pelo *Twitter*®, *Microsoft Windows*®, *Microsoft Word*® e muitos outros aplicativos.

- **Roxo:** Por muito tempo associado à realeza, pode ser usado no *design* para transmitir riqueza e luxúria. Em tonalidades mais claras remete à primavera e romance. É uma das cores preferidas do público feminino. Muito usado em sites da indústria de cosméticos e, especificamente, de produtos antienvelhecimento. Essa cor é utilizada no website de compras *Zoom*®.

Em geral as pessoas preferem as cores azul, verde, roxa e preta. O gráfico Figura 12 mostra o resultado de um estudo de marketing relacionado às cores favoritas das pessoas.



Figura 12: Cores prediletas da população (HALLOCK, 2003)

Adicionalmente ao desenvolvimento de interfaces e escolha de cores, ao longo dos anos tem ocorrido também a mudança na forma dos ícones presentes em logotipos, símbolos, campanhas de marketing e outros. A identidade visual de muitos programas tem mudado para uma interface limpa e visualmente agradável, usando poucos efeitos e poucas cores na sua execução. Esse fenômeno é facilmente visualizado na mudança da logomarca do sistema operacional *Windows*®, mostrado na Figura 13.



Figura 13: Transformação da identidade visual do sistema operacional *Windows®* ao longo do tempo (MUSGRAVE, 2012).

2.4.2. Programa supervisorio

Sistemas de Supervisão e Aquisição de Dados, proveniente do seu nome em inglês *Supervisory Control and Data Acquisition* (SCADA), também chamado de programa supervisorio ou *software* SCADA, são sistemas que utilizam programas para monitorar e supervisionar as variáveis e os dispositivos de sistemas de controle conectados através de servidores/*drivers* de comunicação específicos. Estes sistemas podem assumir topologia mono-posto (*standalone*), cliente-servidor ou múltiplos servidores-clientes. O protocolo de interface para estabelecer a relação cliente-servidor mais usado é o OPC – antes definido como o acrônimo *OLE for Process Control*, atualmente chamado *Open Platform Communications* – um padrão industrial para interconectividade de sistema.

As especificações deste padrão são mantidas pela *OPC Foundation®*, uma organização dedicada ao desenvolvimento de tecnologias aplicadas a interoperabilidade na automação a fim de criar e gerenciar especificações que padronizam a comunicação das arquiteturas de acesso a dados *on-line*, alarmes, registros de eventos, comandos e bancos de dados de diferentes equipamentos, de vários fabricantes que comunicam em diferentes protocolos. Seu funcionamento é baseado no OLE (*Object Linking and Embedding*) de componentes orientados a objeto, por meio das tecnologias COM e DCOM da *Microsoft®* que permitem que aplicações troquem dados que podem ser acessados por um ou mais computadores que usam uma arquitetura cliente/servidor, mesmo que essas aplicações trabalhem sobre sistemas que utilizem protocolos diferentes.

Por ser uma plataforma independente, garante o fluxo contínuo de informações entre dispositivos de vários fornecedores, como *Siemens®*, *Bosch Rexroth®*, *ABB®*, *Festo®*, *Rockwell®* e outros. Quando o padrão foi lançado pela

primeira vez em 1996, seu propósito era abstrair os protocolos específicos do CLP (como *Modbus*, *Profibus*, etc.) em uma interface padronizada, permitindo que os sistemas SCADA fizessem interface com um “intermediário” que convertesse solicitações genéricas de leitura/gravação OPC para solicitações específicas do dispositivo e vice-versa. Como resultado, surgiu uma solução que permite aos usuários finais implementarem sistemas usando os melhores produtos, interagindo perfeitamente via OPC.

Com a introdução de arquiteturas orientadas a serviços em sistemas de manufatura, surgiram novos desafios em segurança e modelagem de dados. A *OPC Foundation*® desenvolveu então as especificações do OPC UA – *Unified Architecture* – para atender a essas necessidades e, ao mesmo tempo, forneceu uma arquitetura de plataforma aberta com tecnologia abrangente em recursos, preparada para o futuro, escalável e extensível (OPC FOUNDATION, 2019).

Nos equipamentos UPPA desenvolvidos pelo LTAD, a transferência de dados entre o computador e o CLP se dá através do servidor OPC, que envia e/ou recebe informações ao CLP, conforme mostrado na Figura 14.

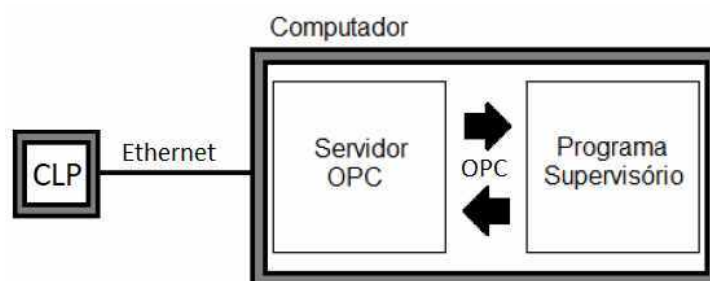


Figura 14: Estrutura de comunicação CLP-Computador utilizada nas UPPAs (adaptado de FREITAS, 2014).

2.5. Programação com base no software LabVIEW®

O LabVIEW® é um *software* projetado para o rápido desenvolvimento de aplicações de engenharia que exijam teste, medição ou controle. Segundo a *National Instruments*® (NI) há duas coisas que tornam o LabVIEW® valioso para engenheiros. A primeira é a rápida e extensa integração de *hardware* disponível para uma ampla variedade de aplicações, desde instrumentos de bancada e placas de aquisição de

dados até protocolos definidos por *software* e *hardware* de computador embarcado baseado em FPGA. A segunda é a abordagem gráfica do LabVIEW® à programação, na qual o usuário pode programar à maneira como pensa – de forma similar ao *Visual Thinking* (Pensamento Visual) – usando um método semelhante ao fluxograma, chamado fluxo de dados gráficos (*dataflow*) para mover dados de uma função para outra.

O LabVIEW® recompila seu código com cada ação, o que significa que o usuário pode detectar e corrigir erros de codificação à medida que eles acontecem, ao invés de ter que compilar e consertar as coisas no final. Além disso, o compilador processa diretamente o diagrama em código de máquina, ou seja, não gera código intermediário baseado em texto. Além disso, o LabVIEW® representa o paralelismo em seu código naturalmente, e a natureza gráfica torna isso fácil de visualizar.

A Figura 15 mostra uma comparação entre o algoritmo clássico de programação “Hello world!” utilizando a linguagem de programação C e a linguagem gráfica utilizada pelo LabVIEW® (linguagem G).

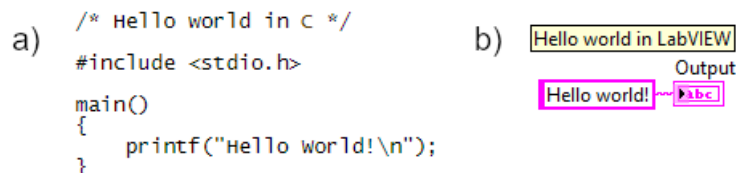


Figura 15: Comparação de estruturas do código “Hello world!”, a) é o algoritmo escrito na linguagem C, b) é o algoritmo na linguagem gráfica LabVIEW®

Programas criados em LabVIEW® são chamados *Virtual Instruments* (VIs) porque sua aparência e operação imitam instrumentos físicos, como osciloscópios e multímetros. Os principais componentes de um VI são o Painel Frontal (*Front Panel*) e o Diagrama de Blocos (*Block Diagram*), mostrados na Figura 16. O Painel Frontal é a interface de usuário na qual o usuário interage com o programa, contendo terminais responsáveis pela entrada e saídas de dados do programa – chamados de controles e indicadores, respectivamente. Já o Diagrama de Blocos é o local aonde está localizado o código que, conforme citado anteriormente, é representado de funções gráficas que realizam diversas ações e controlam os objetos do Painel Frontal (NATIONAL INSTRUMENTS, 2008).

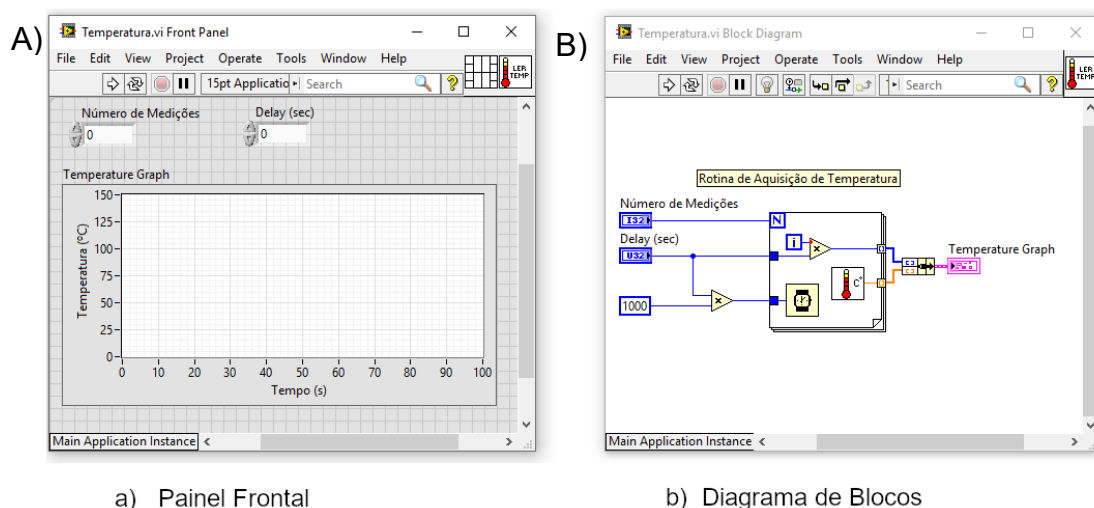


Figura 16: Componentes principais do *software* LabVIEW®.

A) Painel Frontal, B) Diagrama de blocos

A execução de VIs segue o modelo de fluxo de dados. Um nó do diagrama de blocos é executado assim que recebe todas as entradas de que necessita. Quando um nó é executado, ele produz dados, que são enviados ao próximo nó do fluxo de dados. A movimentação dos dados pelos nós determina a ordem na qual os VIs e funções são executados no diagrama de blocos. Os fios são usados para transferir dados entre objetos do diagrama de blocos, onde cada fio somente tem uma fonte de dados, mas pode ser conectado como entrada a várias funções que leem os dados. Os fios podem ser de diferentes cores, estilos e largura, dependendo dos tipos de seus dados. Além disso, a direção dos fios precisa estar correta.

Visual Basic, C++, JAVA e a maior parte das linguagens de programação textual segue um modelo de fluxo de controle da execução do programa. No fluxo de controle, a ordem sequencial dos elementos do programa determina a ordem de execução deste programa. Já no fluxo de dados, a ordem de execução depende de como os fios são ligados. Os nós do LabVIEW® (funções, estruturas e subVIs²) possuem entradas, processam dados e produzem saídas. Portanto, sempre que todas as entradas de um nó contiverem dados válidos, esse nó será agendado para execução e, assim que um processador estiver disponível (sujeito às CPUs disponíveis e à conclusão dos nós previamente agendados), o nó será executado.

² Um VI chamado a partir de um diagrama de blocos de outro VI é chamado de subVI

Em geral, a comparação entre linguagens gráficas e baseadas em texto (ou vice-versa) fornece uma comparação "injusta". Alguns programas simples baseados em linguagem de programação textual podem se tornar programas complexos do LabVIEW®, e programas simples de LabVIEW® podem se tornar extensos algoritmos em linguagem de programação textual. A escolha da ferramenta de trabalho muito dependerá do paradigma da linguagem, do uso e finalidade do programa. As Figuras 17 e 18 ilustram esse caso, comparando dois algoritmos em linguagem textual e em linguagem gráfica, onde a Figura 17 mostra o algoritmo para criar um vetor contendo como elementos os números inteiros 100 até 109 e a Figura 18 é parte de um algoritmo contendo a fórmula $T = \frac{1}{A+BM+CM^2+DM^3} - 273,15$, onde $M = \ln \frac{R}{10000}$

Um desafio de programação é passar dados através dos *loops* sem criar dependência de dados e o paradigma da programação gráfica do ambiente LabVIEW® é ideal para arquiteturas de dados paralelos. Paralelismo refere-se a executar múltiplas tarefas ao mesmo tempo. Considere o exemplo de criar e exibir duas ondas seno com diferentes períodos. Usando o paralelismo natural de uma linguagem de programação gráfica, cada forma de onda será colocada dentro de uma estrutura de repetição própria, sendo executada de forma paralela sem a necessidade de criar funções, gerenciar alocação de recursos ou configurações extras. Maiores informações a respeito do LabVIEW® podem ser encontradas no site da *National Instruments®*.

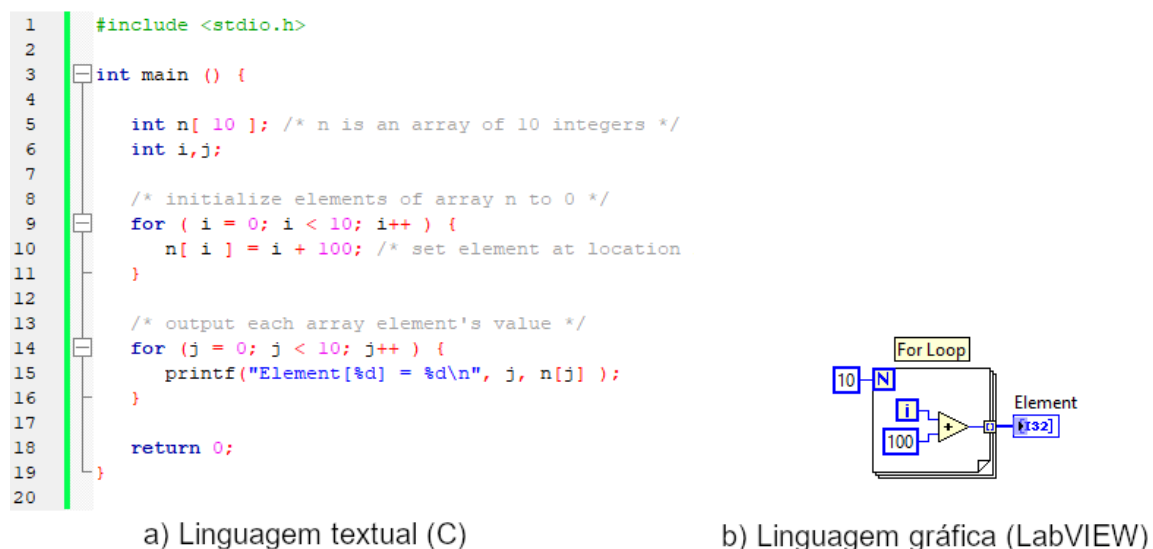


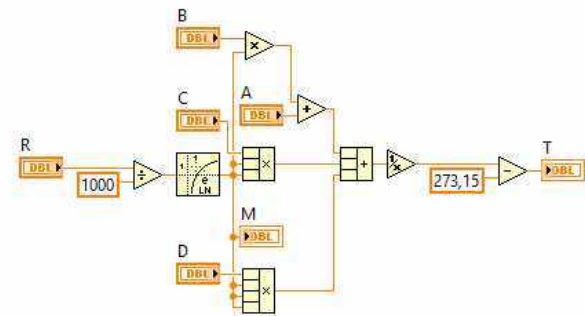
Figura 17: Comparação entre algoritmo que gera e mostra um vetor contendo dez elementos, a) Linguagem baseada em texto, b) Linguagem gráfica

```

/* Fórmula para cálculo da temperatura */
double M;
M=log(R/10000);
T=(1/(A+B*M+C*M*M+D*M*M*M))-273.15;

```

a) Linguagem textual (C)



b) Linguagem gráfica (LabVIEW)

Figura 18: Fórmula matemática complexa representada em: a) linguagem textual
b) linguagem gráfica.

2.5.1. Estruturas de programação

O padrão de *design* **máquina de estados** consiste em uma estrutura de repetição em conjunto com uma estrutura de casos. Essa arquitetura permite executar diferentes códigos enquanto o *loop* executa, dependendo de uma condição pré-definida. Essa condição é chamada de estado e corresponde a um caso próprio dentro da estrutura de casos. Em linguagem C essa arquitetura é implementada através do comando *switch case*.

O padrão de *design* **mestre-escravo** consiste de múltiplos *loops* em paralelo. Cada loop pode executar tarefas em diferentes taxas. Um *loop* age como mestre e os demais como escravos. A estrutura mestre controla todas as outras, se comunicando usando técnicas de mensagem. Essa estrutura normalmente é usada quando o programa precisa responder a comandos provenientes do usuário enquanto simultaneamente coleta dados.

O padrão de *design* **produtor-consumidor** é baseado no mestre-escravo, melhorando o compartilhamento de dados entre múltiplos *loops* executando em diferentes taxas. As estruturas de repetição são separadas em duas categorias – aquelas que produzem dados e aquelas que consomem os dados produzidos. Filas fazem a transferência de dados entre os *loops*. Portanto um *buffer* de dados se faz necessário. Um *buffer* é um dispositivo de memória que armazena dados temporários entre dois dispositivos ou, neste caso, vários *loops*.

CAPÍTULO III

3. ASPECTOS CONSTRUTIVOS DA UNIDADE DE PROCESSAMENTO DE PINOS DE ATRITO (UPPA)

Por não ter a concepção de ser enviada à campo, a estrutura da UPPA3 difere dos demais equipamentos. As principais diferenças desta unidade em relação às versões anteriores estão na forma construtiva do equipamento e na maior capacidade de força axial de 500 kN (valor de projeto). O sistema de rotação e de translação são independentes, ambos montados em uma mesma estrutura, ou seja, não existe um cilindro de reparo como apresentado nas versões anteriores. A ideia visa atender também o reparo de dutos, já que os movimentos de rotação e translação são independentes (BORGES, 2011).

Outro diferencial desta versão é a utilização de um sistema de aquisição de torque independente, o qual permite medir o sinal de torque diretamente. Nas outras versões das UPPAs, a aquisição de torque é realizada com base no monitoramento das pressões de entrada e saída do motor hidráulico.

3.1. Unidade hidráulica

A unidade hidráulica para o acionamento do cilindro de reparo contém um reservatório com capacidade de 630 L, três sistemas motobomba e um bloco de válvulas. O motor hidráulico da cabeça de reparo é acionado por um conjunto motobomba composto por um motor diesel MWM com potência de 215 CV acoplado a uma bomba de pistão axial variável de disco inclinado com capacidade máxima de vazão de 180 L/min e pressão máxima de operação de 350 bar. O cilindro da cabeça de reparo é acionado por um conjunto motobomba composto por um motor elétrico de potência de 10 CV acoplado a uma bomba de pistão axial variável de disco inclinado

com capacidade máxima de vazão de 12 L/min e pressão máxima de operação de 290 bar.

No trabalho realizado por Freitas (2014), foi constatado que a pressão da linha contendo o cilindro hidráulico está ajustada de forma que a força máxima que pode ser atingida na haste do cilindro é de 400 kN. Já a pressão máxima ajustada na linha contendo o motor hidráulico é de 310 bar, o que é capaz de gerar no eixo do motor hidráulico um torque de aproximadamente 616 N.m. A vazão máxima da bomba permite uma rotação máxima do eixo do motor hidráulico de 1.750 rpm sem que haja queda significativa de pressão na linha.

3.2. Sistema de aquecimento

A UPPA3 conta com um sistema de aquecimento indutivo, utilizado em ensaios com pré-aquecimento do metal-base do corpo de prova, cujo esquema simplificado é apresentado na Figura 19. Um sensor infravermelho é usado para a aquisição da temperatura da amostra.

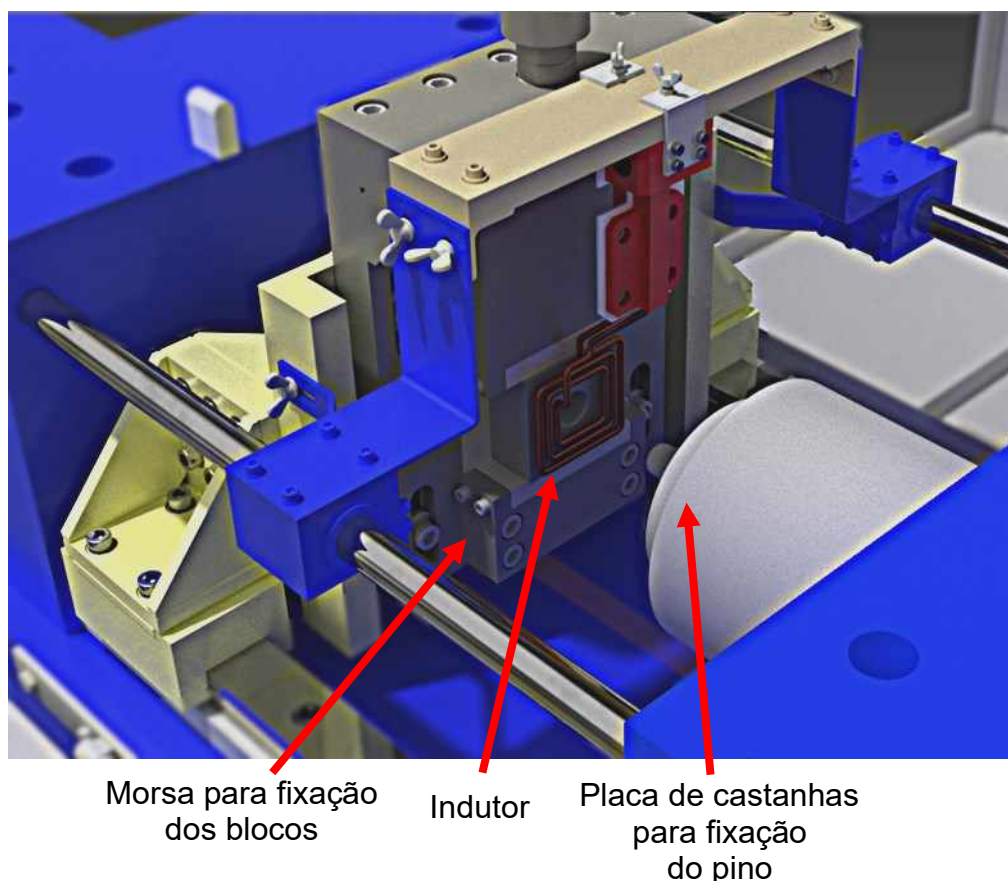


Figura 19: Esquema simplificado de aquecimento indutivo da UPPA3.

3.3. Instrumentação, controle e aquisição de dados

A aquisição de dados é realizada pelo CLP modelo *IndraControl* L25 da *Bosch Rexroth*®, sendo este também usado na UPPA2. É composto pelos seguintes módulos:

- Módulo RS485, utilizado pelo sistema de aquisição de temperatura da amostra (sensor infravermelho);
- Módulo contador de pulsos, utilizado pelo sensor de rotação;
- Módulo com 32 entradas digitais, utilizado para verificar estado de relés, fins de curso e outros;
- Módulo com 32 saídas digitais, utilizado na partida do motor diesel, para ligar lâmpadas e outros;
- Seis módulos contendo, cada um, duas entradas analógicas, que são utilizados para adquirir os sinais dos transdutores de pressão e torque, LVDTs e o sinal de *feedback* das válvulas de controle;
- Dois módulos contendo, cada um, duas saídas analógicas, utilizados no controle de força e rotação.

Os parâmetros adquiridos durante o processo de reparo por atrito são: força, rotação, comprimento de queima do pino e torque.

- A **força** é adquirida através de dois sensores de pressão localizados no cilindro hidráulico;
- O motor hidráulico da UPPA3 possui internamente uma roda dentada, constituída de material ferromagnético, e um sensor de efeito Hall. Assim, a **rotação** é obtida a partir da frequência de pulsos gerados por esse sensor quando o motor hidráulico é acionado;
- A posição da haste do cilindro é obtida a partir de um sensor do tipo LVDT que, através do valor de posição, indica o valor de medida do **comprimento de queima** durante o reparo. O cilindro hidráulico em questão tem curso de 200 mm.
- O **torque** é obtido através de um transdutor de torque acoplado ao eixo do motor hidráulico.

Em adição a esses sensores, a UPPA3 também possui dispositivos adicionais de segurança e monitoramento:

- Sensores indutivos e fins de curso, que tem a função de garantir a segurança do equipamento e do operador durante os ensaios;
- Uma câmera USB fixada na estrutura, responsável pela gravação de vídeos durante a realização de ensaios e monitoramento visual do processo.

O controle de força é realizado através de um controlador PID digital e o controle de rotação através de um controlador do tipo rede neural de um neurônio e função de ativação linear, conforme esquema apresentado na Figura 7. Um sistema antitravamento entra em ação quando a rotação medida está 100 rpm abaixo do *setpoint* de rotação. Esse sistema consiste em desativar o controle de força e fechar a válvula direcional do cilindro, impedindo seu avanço. Maiores detalhes sobre o sistema de controle estão disponíveis no trabalho desenvolvido por Freitas (2014).

3.4. Programa supervisório UPPA3

Inicialmente, além do CLP e computador, a UPPA3 necessitava de um dispositivo para a parametrização de cada ensaio realizado. Esse dispositivo era a Interface Homem Máquina (IHM), que ficava acoplada à porta do painel elétrico. A IHM era composta por um programa construído à aplicação para ensaios de reparos por atrito, possuindo em sua interface botões diversos de configuração ativados e desativados com toques na tela, como mostrado na Figura 20. Esse sistema era necessário para a configuração inicial e definição dos valores de cada parâmetro do processo. Os valores de referência, após serem inseridos na IHM, eram enviados aos cartões PID e CLP a fim de iniciar o processo.



Figura 20: Interface Homem Máquina (IHM) utilizada para configuração de parâmetros de ensaio (BORGES, 2011).

O programa supervisorio, desenvolvido no *software* LabVIEW®, responsável pela leitura das principais variáveis e monitoramento dos ensaios, possuía a interface conforme mostrado na Figura 21. Os valores de rotação, força axial, torque e comprimento de queima eram exibidos por meio de gráficos e atualizados em tempo real até o final do processo.

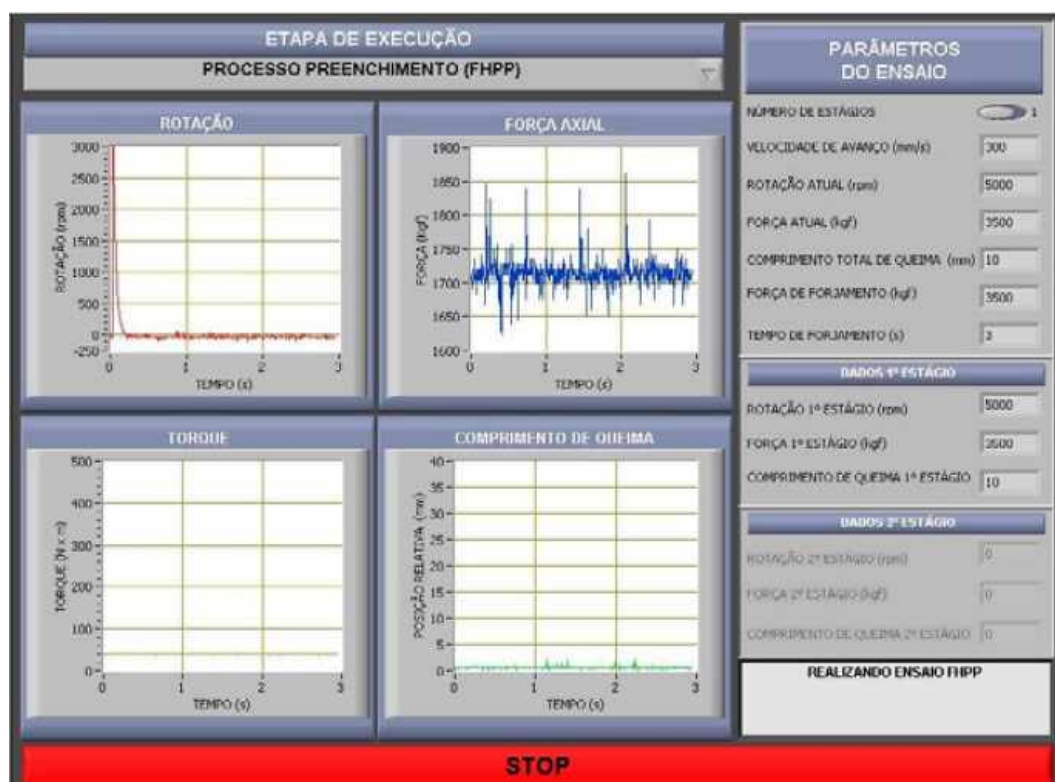


Figura 21: Interface gráfica do primeiro programa supervisorio da UPPA3 (FORMOSO, SANTOS, *et al.*, 2009)

Freitas (2014) modificou a estrutura da UPPA3, de forma que a realização de ensaios passou a utilizar apenas o programa supervisorio, sem necessidade da IHM, que realiza a interface entre o operador e a UPPA3. Na Figura 22 é apresentada a interface desse programa, também desenvolvido no *software* LabVIEW®. Os valores dos parâmetros de processo podem ser visualizados durante o ensaio a partir dos gráficos e o reparo por atrito pode ser acompanhado em tempo real a partir de uma câmera, cuja imagem é mostrada na parte superior direita da janela do programa.

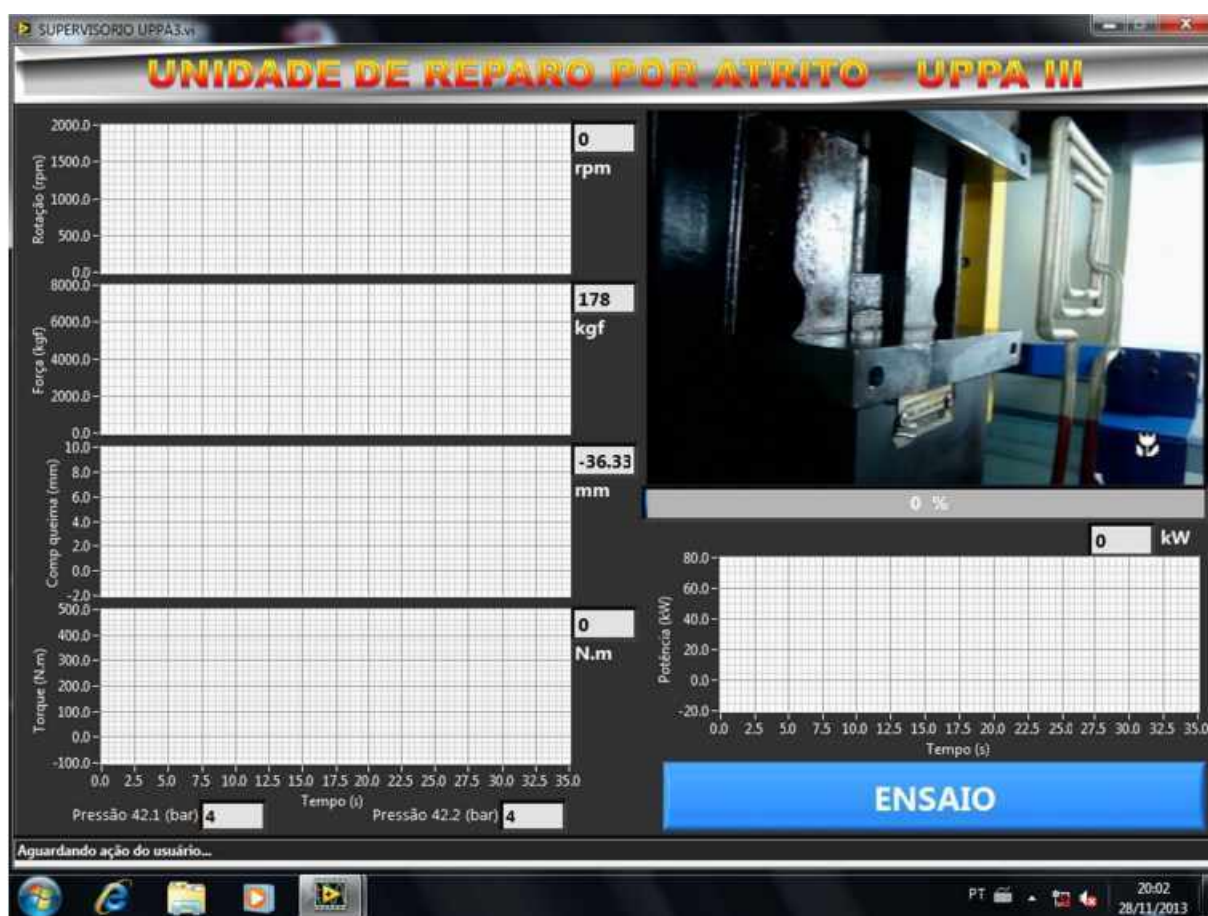


Figura 22: Interface do segundo programa supervisorio desenvolvido para a realização de reparos por atrito na UPPA3 (FREITAS, 2014)

Ao longo do desenvolvimento de projetos no LTAD, novos programas foram desenvolvidos, contando com diferentes interfaces, estruturadas de acordo com o conhecimento e gosto de cada programador. Observando os projetos desenvolvidos por grandes empresas, nota-se a adoção de uma identidade visual unificada, que tem

por objetivo padronizar documentos, fortalecer e divulgar a marca institucional. Portanto, uma interface gráfica similar entre os diferentes programas desenvolvidos pelo LTAD é desejável. Sendo este, conforme mencionado no Capítulo I, um dos objetivos do desenvolvimento desse trabalho

Adicionalmente pode-se citar a última reestruturação da UPPA, realizada em 2018, na qual houve mudanças na estrutura física e de programação da UPPA3. Nesse caso foi necessário a adequação da programação do sistema de controle, visando atender ao novo CLP *Bosch Rexroth*® instalado para o controle da UPPA2 e UPPA3. Após essa adequação, viu-se a necessidade de desenvolver um novo programa supervisor. Esse desenvolvimento está descrito no próximo capítulo desse trabalho.

CAPÍTULO IV

4. DESENVOLVIMENTO DO PROGRAMA SUPERVISÓRIO

4.1. Estrutura do programa desenvolvido

Um dos problemas mais importantes em engenharia de *software* tem sido como programar grandes e complexas partes de um programa. Muitas vezes um grande número de engenheiros precisa trabalhar em conjunto no desenvolvimento e otimização de programas, de modo que este opere de maneira ótima.

O primeiro passo do desenvolvimento de um programa é a definição das diretrizes de projeto. Com o objetivo de ser escalável e personalizável, foi definido que o programa desenvolvido deveria ser do tipo modular. Modularização significa separar e organizar o código com base nas tarefas executadas. Desta forma, o código torna-se reutilizável e mais fácil de gerenciar e depurar³.

Ao depurar programas grandes, descobrir como e quando ocorreu qualquer erro pode se tornar um processo complexo e dificultoso. Programas desorganizados podem demandar horas de retrabalho do programador, que procura descobrir onde ocorreu o erro e os problemas que este pode causar na execução do programa. No entanto, se um programa é projetado com modularidade em mente, cada tarefa tem sua própria seção separada dentro do código. Então, se há um problema em uma determinada função ou rotina, o programador sabe para onde olhar e pode assim gerenciar uma parcela menor de código.

Como estratégia de modularização, foi definido a divisão da estrutura da janela (interface gráfica) em partes, onde cada parte é responsável por uma tarefa específica. O programa é composto por um VI principal chamado “*main.vi*”, que faz o papel de interface homem-máquina, exibindo as informações e recebendo os comandos realizados pelo usuário. Esse VI principal é composto por cinco painéis.

³ Procurar, detectar e corrigir erros em um programa de computador

A Figura 23 mostra a estrutura-base da interface do programa. O conteúdo contido em cada painel será detalhado nos próximos subtópicos desse capítulo.

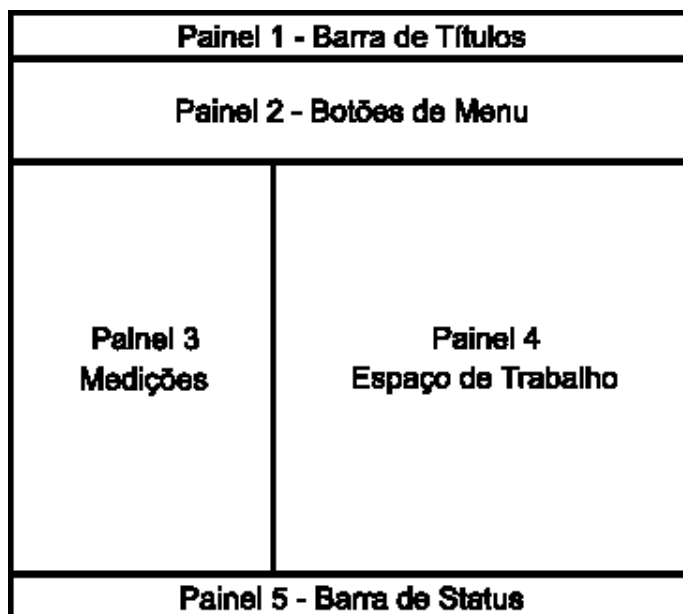


Figura 23: Estrutura básica do programa, composta pela divisão da interface gráfica em cinco painéis, onde cada painel é responsável por uma sub-rotina específica do programa supervisorio.

A Figura 24 mostra os principais objetos presentes em cada um dos painéis do programa principal. Conforme citado anteriormente, o Painel Frontal é a interface de usuário, contendo terminais responsáveis pela entrada e saídas de dados do programa – chamados de controles e indicadores, respectivamente. Sabendo antecipadamente quais objetos constituem um painel, obtém-se informações importantes sobre o código desenvolvido. Observando, por exemplo, que o “Painel 3” (Figura 24) é constituído apenas por indicadores, conclui-se que nessa parte do código não há captura de dados do operador e, portanto, as rotinas desenvolvidas não precisam incorporar interação com o usuário.

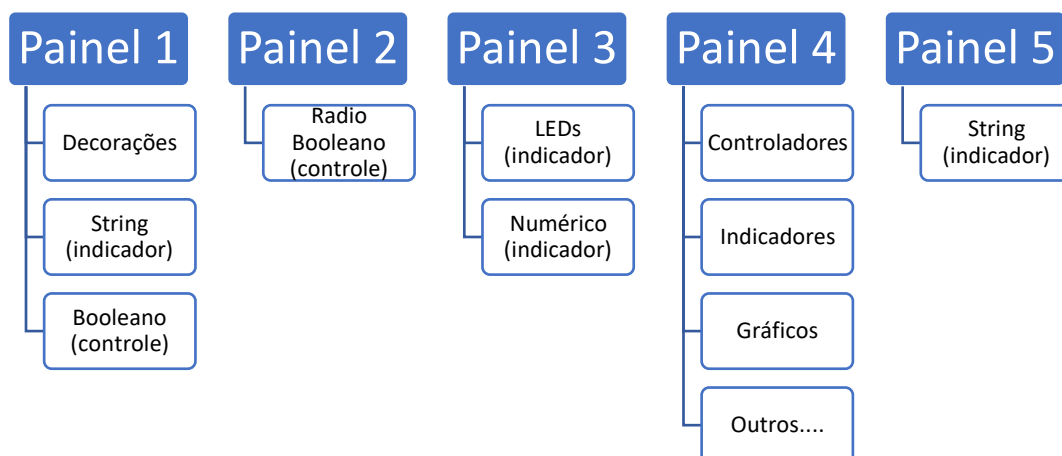


Figura 24: Principais objetos esperados dentro de cada painel do programa.

4.1.1. Painel 1 – Barra de Títulos

Mostra o nome do programa e os botões créditos, minimizar, maximizar e fechar. O botão créditos deve exibir informações a respeito da UFU, LTAD e patrocinadores do projeto. Os botões minimizar, maximizar e fechar desencadeiam eventos na “*Event Structure*” do LabVIEW®, executando rotinas que imitam a função da barra de títulos presente nos sistemas operacionais convencionais. Conforme indicado pelo nome, o botão minimizar esconde a janela do programa supervisório, o botão maximizar deixa a tela em seu tamanho máximo e o botão fechar encerra a execução do programa supervisório.

A rotina de fechamento consiste em ações que finalizam o programa de forma eficiente e segura, na qual o programa aguarda cada subVI finalizar sua execução antes de prosseguir com a finalização do LabVIEW. Dessa forma não há perda de dados ou rotinas indesejadas executando em segundo plano. Ainda há rotinas de segurança nesse painel que impedem o usuário de acionar o botão “Fechar”⁴ em momentos indesejados, como durante a execução dos ensaios.

⁴ Detalhes da rotina de segurança e fechamento estão apresentados na seção 4.6.

4.1.2. Painel 2 – Botões de Menu

Contém um objeto chamado “*Radio Buttons*”, que é um tipo especial de booleano contendo múltiplos botões. Quando o usuário pressiona um desses botões, este mudará seu estado para verdadeiro enquanto todos os outros botões desse objeto permanecerão no estado falso. Dessa forma, apenas um único botão pode estar ativo a cada instante e esse botão é usado para indicar ao programa qual sub-rotina deverá ser exibida no Painel 4, agindo como um “menu seletor”.

4.1.3. Painel 3 – Medições

Exibe um subVI chamado “Medições”, que realiza toda e qualquer comunicação com quaisquer CLPs (Controlador Lógico Programável) e sensores do sistema. Os dados coletados são enviados para processamento nos subVIs presentes no Painel 4. Aqui também são processados os comandos e *setpoints* definidos pela interação do usuário com as outras partes do programa.

Todas as medições de força, rotação, torque, comprimento de queima, pressão, bem como a obtenção das imagens da câmera são realizadas no subVI executado através desse painel. Nessa rotina não existem objetos do tipo “*Control*”, que captam informações do usuário, apenas objetos do tipo “*Indicator*”, que apenas exibem informações. Portanto, o usuário não fará modificações ou envio de comandos nesse parte do programa supervisorio.

4.1.4. Painel 4 – Espaço de Trabalho

Composto por vários sub-painéis, esse local é responsável por captar a maioria das ações do usuário, repassando os dados para processamento dentro da sub-rotina presente no Painel 3. Ações como definição de parâmetros de ensaio, escolha do momento de início do ensaio, movimentação manual de atuadores e definição das configurações de sensores são sempre realizadas nas rotinas pertencentes a esse módulo. As principais sub-rotinas desenvolvidas são: Home, Ensaio, Análise e Configurações. Novas sub-rotinas podem ser criadas de acordo com a necessidade de cada projeto.

- **Home:** Utilizada em movimentações manuais do equipamento, acionamento individual de componentes e auxílio em operações de calibração e manutenção;
- **Ensaio:** É responsável por todos os ensaios realizados. O usuário informa os parâmetros desejados e, após verificação das rotinas de segurança, o ensaio é realizado. Nessa janela é realizado a exibição, tratamento e salvamento dos dados do ensaio realizado;
- **Análise:** Essa janela é usada para carregar ensaios realizados anteriormente e comparar resultados;
- **Configurações:** Nessa janela são definidas configurações como informações acerca do ensaio como local de salvamento dos dados, informações sobre as configurações da câmera, entre outros.

Como estratégia de desenvolvimento do módulo “Espaço de Trabalho”, foi definido que todas os subVIs dessa parte do algoritmo teriam a mesma estrutura-base e as funcionalidades específicas seriam desenvolvidas a partir dessa estrutura. Isso porque pode haver casos em que uma sub-rotina precise realizar operações que, a princípio, poderiam ser classificadas como de outra. Por exemplo, pode haver casos em que um programa necessite realizar processos de calibração do equipamento, portanto esta operação manual deve ser realizada no subVI “Home”, logo, este deve ser capaz de obter, tratar, salvar e apresentar dados, funcionalidades que a princípio estão previstas para acontecer no subVI “Ensaio”.

Nesse painel estão presentes os mais variados objetos como controladores e indicadores booleanos, numéricos, *strings*, vetores, matrizes e gráficos. Rotinas de segurança estão previstas de forma a impedir o usuário de realizar ações indevidas durante certas etapas do processo. Durante a execução dos ensaios, por exemplo, o usuário é impedido de fechar o programa, mudar o menu atualmente exibido ou enviar novos comandos ao CLP sem antes interromper o ensaio em execução.

4.1.5. Painel 5 – Barra de Status

Esse painel é responsável por processar e exibir mensagens informativas para o usuário. Alguns exemplos de mensagens são: erros nos sensores, estado do ensaio em execução, se ocorreu erros no ensaio e dicas de funcionamento geral do programa.

4.2. Comunicação entre as partes do programa

Por se tratar de um programa modularizado e também pelo fato de que os subVI nos Painéis 3 e 4 (Figura 23) estão sempre em execução – coletando dados e/ou aguardando operações do usuário – transmitir dados somente através de fios conforme o sistema de fluxo de dados do LabVIEW® não é possível.

A transferência de dados entre as partes do programa é feita usando filas e notificadoros. As filas funcionam conforme a arquitetura produtor-consumidor e são ideais para transmitir dados na forma N-1 (vários transmissores e apenas um receptor). Notificadoros funcionam de acordo com a arquitetura mestre-escravo e são indicados para transmitir dados na forma 1-N (um transmissor e vários receptores)

O padrão de *design* mestre-escravo no LabVIEW® consiste em múltiplos *loops* em paralelo. Cada *loop* pode executar tarefas em diferentes taxas, onde um *loop* age como mestre e os demais como escravos. A estrutura mestre controla todas as outras, se comunicando através de mensagem. As mensagens são enviadas na forma de *broadcast*, método no qual a mensagem é transmitida para todos os receptores simultaneamente. Porém nessa arquitetura apenas último dado transmitido é preservado, podendo então ocorrer perda de dados entre transmissor e receptor.

Já o padrão de *design* produtor-consumidor, baseado no mestre-escravo, melhora o compartilhamento de dados entre múltiplos *loops* executando em diferentes taxas. As estruturas de repetição são separadas em duas categorias – aquelas que produzem dados e aquelas que consomem os dados produzidos. Filas fazem a transferência de dados entre os *loops*, portanto um *buffer*⁵ de dados se faz necessário. Os dados são armazenados de forma sequencial, formando a fila de dados. Quando

⁵ Dispositivo de armazenamento de caráter transitório, utilizado durante uma operação de transferência ou transmissão de dados entre unidades de armazenamento ou de processamento que operam com tempo de acesso, velocidades ou formatos distintos

o *loop* consumidor executa, este retira os dados da fila de forma sequencial, garantindo com que não haja perda de dados na comunicação.

A estrutura de filas foi utilizada em processos considerados mais críticos, como a captura de dados durante o ensaio, onde não pode haver perda de dados. A estrutura de notificadores foi utilizada principalmente para envio e recebimento de comandos entre os módulos do programa, pois a comunicação via *broadcast* facilita sua implementação. Para não haver erros na troca de mensagens, cada subVI possui um identificador de remetente único, facilitando o processamento da mensagem no receptor.

Para que possam ser identificadas, cada fila e notificador deve possuir um nome único no programa. A Figura 25 apresenta o esquema da estrutura de comunicação entre as sub-rotinas presentes no programa, onde o índice *n* indica um notificador e o índice *q* indica uma fila, enquanto a palavra após o índice representa o nome da fila/notificador. O quadro represente um VI, os itens localizados à esquerda de cada quadro indicam uma entrada de dados, enquanto os itens à direita dos quadros indicam saída de dados. O sentido de transferência de dados também pode ser representado pelas setas, que ilustram se há entrada ou saída durante a troca de informações.

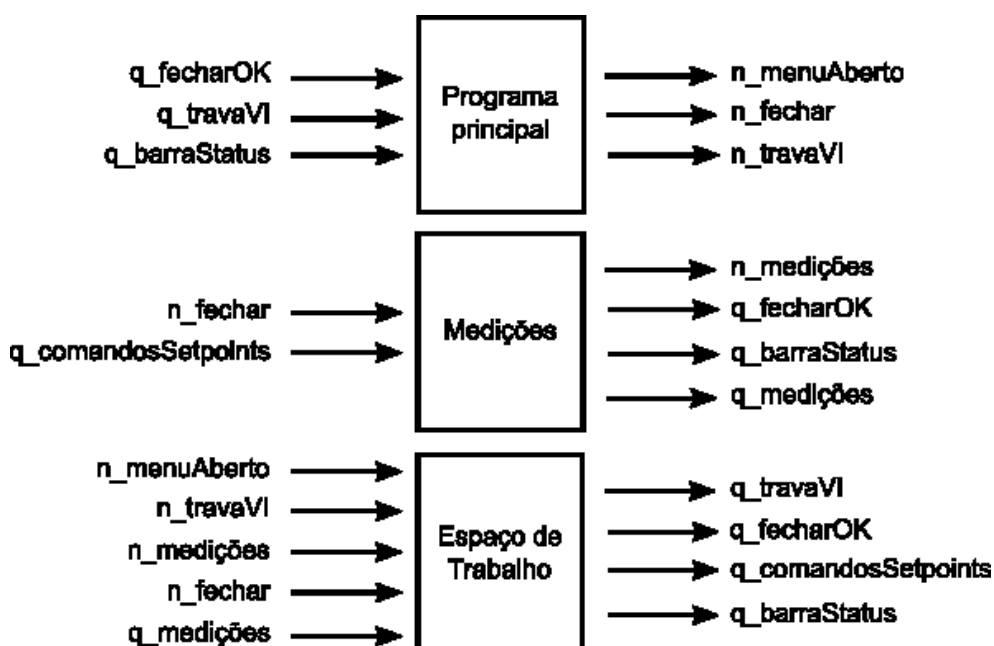


Figura 25: Estrutura de transferência de dados entre as sub-rotinas do programa. O quadro representa um subVI, as setas indicam se a informação está entrando ou saindo. O índice *n* indica um notificador, o índice *q* indica uma fila

A Figura 26 mostra a mesma estrutura descrita no parágrafo anterior, porém de forma alternativa. A base de cada seta indica o emissor, a partida dos dados, enquanto a ponta da seta indica o destinatário, que irá processar os dados.

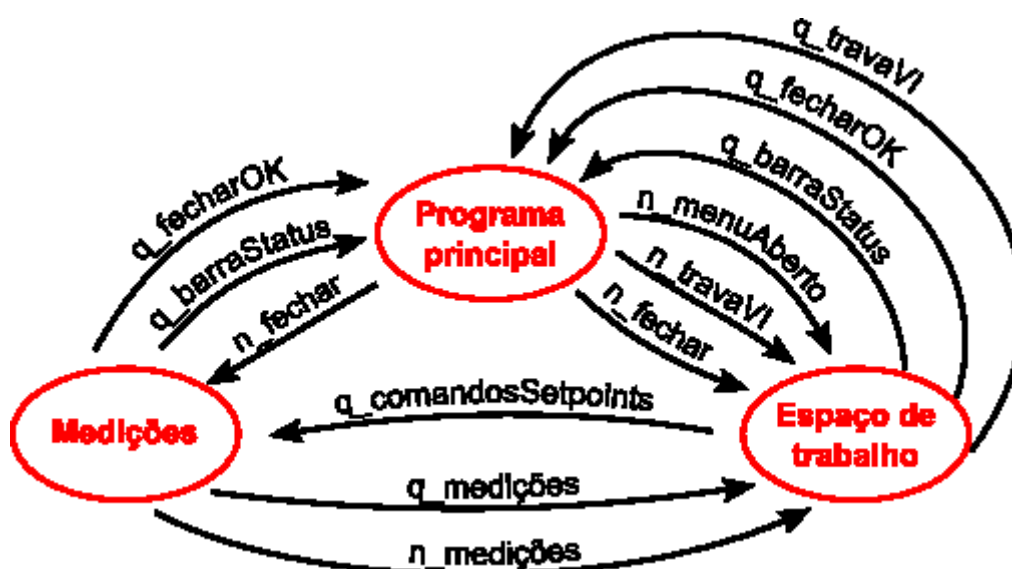


Figura 26: Estrutura de transferência de dados entre as rotinas mostradas de forma alternativa.

Os dados transmitidos nas filas e notificadores podem ser de qualquer tipo (booleano, numérico, *string*, variante, entre outros) e agrupados como escalares, vetores, *cluster* ou matrizes. Porém as informações a respeito do dado transmitido devem ser informadas no momento de criação de cada fila e notificador, permanecendo fixa durante toda a execução do programa. Isso significa, por exemplo, uma vez definido que a fila “*n_fechar*” transmita dados na forma de booleano, essa fila só poderá receber booleanos durante toda a execução do programa. Dessa forma, a definição do tipo de dado transmitido em cada fila é um requisito de projeto do programa.

A Tabela 2 mostra os detalhes dos dados transmitidos nas filas e notificadoros, bem como um resumo de sua função no programa. A primeira coluna mostra o nome da fila/notificador no algoritmo, onde o índice q indica uma fila, o índice n indica um notificador. A segunda coluna mostra a estrutura de dados das filas/notificadoros, que envolve dados do tipo numérico, booleano, *strings* e outros tipos que podem ser armazenados na forma de escalares, vetores, matrizes ou clusters. A terceira coluna mostra uma descrição resumida da função da fila/notificador dentro do programa. Novas filas e notificadoros podem ser acrescentados ao programa, caso seja necessário.

Uma observação importante a respeito da troca de dados entre as sub-rotinas do programa está relacionada aos subVIs executados no Painel 4 – Espaço de Trabalho (Figura 23). A quantidade de sub-painéis presentes nesse painel deve ser igual ao número de botões do menu presentes no Painel 2. Além disso, deve ser garantido que apenas um desses sub-painéis esteja processando dados das filas e notificadoros. Quando o menu “Home” estiver em execução, por exemplo, o sub-painel ativo deve ser o que contém as rotinas do subVI que corresponda a esse menu. Somente esse subVI deve retirar dados das filas ou processar informações dos notificadoros. Para isso foi implementado o notificador “n_menuAberto”, que impede condição de corrida nas filas e notificadoros, pois cada subVI só escutará a troca de mensagens nas filas e notificadoros quando o menu correto do Painel 2 estiver selecionado.

O procedimento de troca de informações utiliza envio de mensagens, códigos em vetores ou *clusters*, comandos e outras estruturas de codificação. Para isso foi definido padrões de nomenclatura, ordens de posições dentro de vetores e outras instruções que visam facilitar o reconhecimento do tipo de variável e indicam a estrutura da mensagem que será transmitida para os receptores. O Apêndice A lista as diretrizes definidas no sistema de troca de informações entre os módulos do programa, que devem ser seguidas ao criar subVIs, filas, notificadoros ou mensagens. Essa estrutura também pode ser usada como base de comunicação no desenvolvimento de outros programas supervisórios para equipamentos do LTAD.

Tabela 2: Descrição do conteúdo e função das filas e notificadoros do programa.

Nome	Conteúdo	Função
q_barraStatus	<i>Cluster</i> contendo: • remetente [i32] • identif. da mensagem [i32] • conteúdo [string]	Os subVI enviam mensagens para serem exibidas ao usuário na Barra de Status do programa principal.
q_comandosSetpoints	<i>Cluster</i> contendo: • comando [u32]; • vetor de setpoints [f64] • vetor de comandos [string]	Executa funções de acordo com parâmetros informados pelo usuário. Exemplos de comandos: iniciar ensaio, movimento de atuadores, gravar vídeo, entre outros.
q_fecharOK	Numérico [i32]	Os subVIs informam ao programa principal que terminaram a execução de todos os seus laços, permitindo assim o que o programa seja encerrado com segurança
q_medições	Matriz de seis colunas do tipo numérico, que contém as medições realizadas pelos sensores [f64]	Essa fila é responsável por transmitir os dados durante o ensaio. Esses dados são usados para armazenamento em arquivos e/ou plotagem de gráficos.
q_travaVI	Booleano	Os subVI informam ao programa principal para executar rotinas de segurança
n_fechar	Booleano	Variável responsável por finalizar a execução dos laços
n_medições	<i>Cluster</i> contendo: • Vetor de estado sensores [u32]; • Vetor com medições atuais [f64] • Vetor de strings	Responsável por mostrar ao usuário o estado dos sensores, as últimas medições captadas pelos sensores e eventuais mensagens de texto provenientes do sistema de medição e/ou CLP
n_menuAberto	Numérico [i32]	Informa às rotinas do Espaço de Trabalho qual dos botões do menu está atualmente selecionado.
n_travaVI	Booleano	Variável responsável por iniciar e interromper rotinas de segurança, impedindo que o usuário saia do programa, troque o menu ativo ou execute comandos indesejados.

4.3. Fluxograma de funcionamento do programa desenvolvido

O fluxograma simplificado de funcionamento do programa supervisor é apresentado na Figura 27. A imagem mostra que a interação do operador com o programa se dá basicamente dentro de uma das quatro sub-rotinas – Home, Ensaio, Análise e Configurações – constituintes do Painel 4 (ver Figura 23). A escolha de qual sub-rotina será exibida depende da escolha do menu, que é realizada através do Painel 2; a escolha é feita simplesmente clicando em um botão presente nesse painel. As quatro sub-rotinas possuem, conforme mencionado anteriormente, uma estrutura-base semelhante, de forma que todas estão aptas a executar quaisquer funções supervisórias, tendo como requisito apenas o desenvolvimento específico da utilidade desejada.

Por exemplo: a captura de imagens é feita pela câmera USB através do subVI que faz as Medições (Painel 3), que transmite o dado através de uma fila aos subVIs do Painel 4. A subVI “Home” permite o salvamento da imagem de forma manual, com início e término do processo definido de forma manual, enquanto o subVI “Ensaio” está configurado para armazenar de forma automática as imagens em disco apenas durante a execução de um ensaio. O subVI “Análise” simplesmente descarta o dado recebido e o subVI “Configurações” apenas permite modificar o arquivo de configurações responsável por informar qual a resolução da câmera usada na aquisição de imagens.

É importante destacar que cada uma dessas sub-rotinas – Home, Ensaio, Análise e Configurações – processa dados apenas quando está sendo exibida (ativa) no Painel 4. Caso contrário, poderia haver condição de corrida nas filas e também o consumo desnecessário de recursos computacionais.

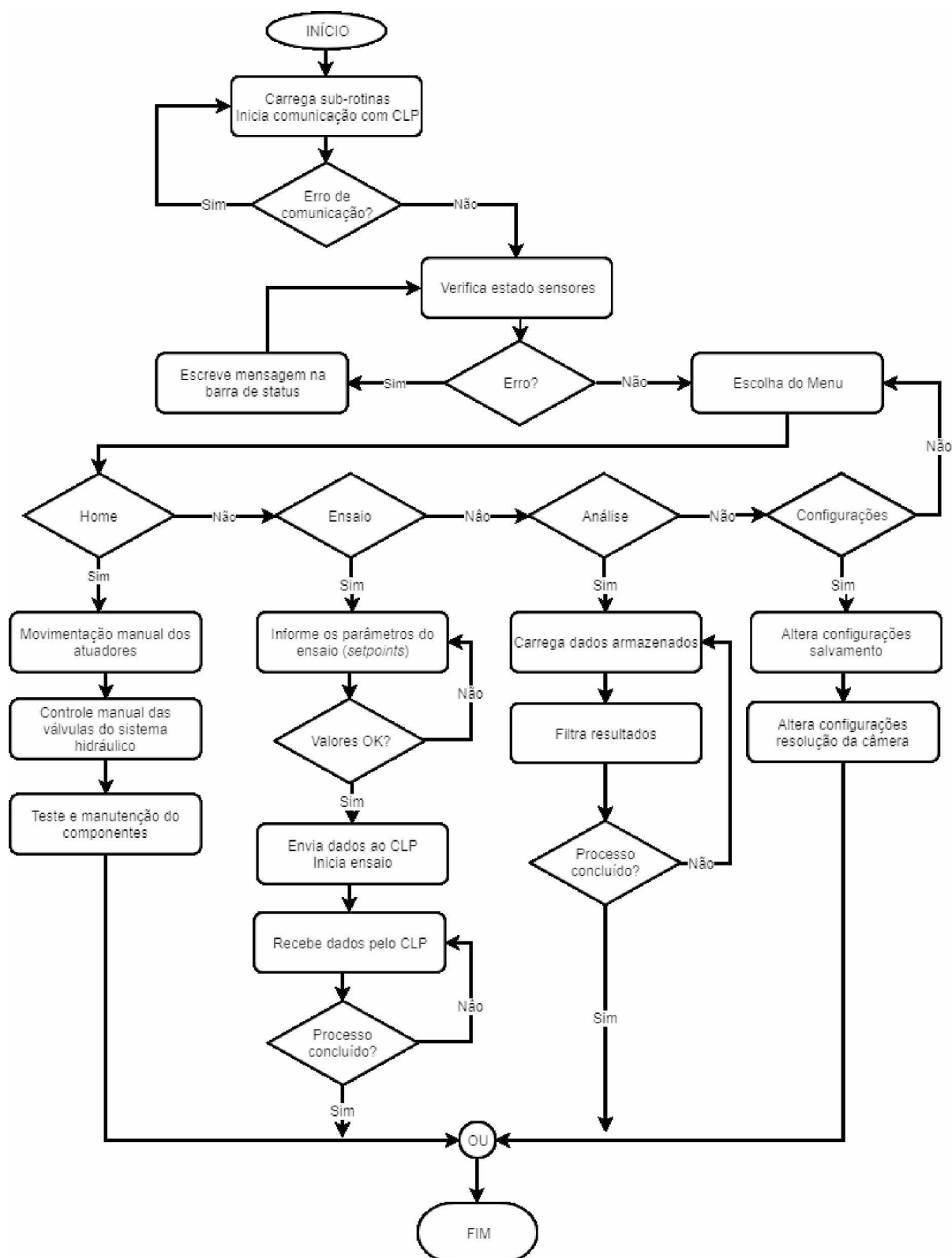


Figura 27: Fluxograma do funcionamento geral do programa supervisor.

4.4. Aquisição de dados

A aquisição de dados dos sensores é realizada pelo CLP, que transmite as informações para o servidor OPC. O LabVIEW® coleta os dados através de uma função presente na biblioteca *Datalogging and Supervisory Control* (DSC), que ajuda a desenvolver aplicativos de IHM/SCADA ou de *data logging* com grandes quantidades de canais. O DSC é um *software add-on* para o ambiente de programação LabVIEW®, que pode ser utilizado para desenvolver um sistema distribuído de controle e monitoramento com *tags* que variam de algumas dezenas a dezenas de milhares, inclui ferramentas que permitem registrar dados em um banco de dados, histórico em rede, monitorar tendências históricas e em tempo real, gerenciar alarmes e eventos, conectar o LabVIEW® e dispositivos OPC (NATIONAL INSTRUMENTS, 2019).

A Figura 28 mostra as funções utilizadas para estabelecer conexão, ler dados e enviar comandos ao CLP, respectivamente. Como o computador está conectado ao CLP através da arquitetura *Ethernet*, é necessário que cada dispositivo tenha um endereço IP. Para receber ou enviar dados ao CLP, o programa supervisorio utiliza o sistema de *tags* (ou *node path*), que é o método mais utilizado para esse tipo de comunicação.

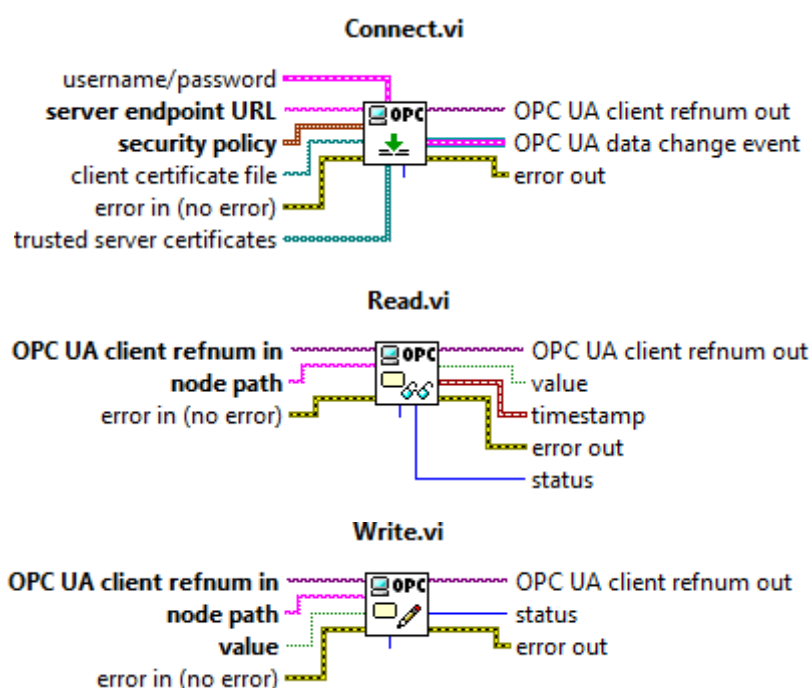


Figura 28: Funções usadas para leitura dos dados capturados do CLP.

De forma similar às funções apresentadas na Figura 28, a captura da imagem da câmera USB é realizada por funções pertencentes à biblioteca NI-IMAQdx, que pertence ao módulo *Vision Acquisition*. Este é um pacote de *software* que adquire, mostra e registra imagens fornecidas por uma grande variedade de tipos de câmera.

4.5. Interface gráfica

Um dos benefícios que o LabVIEW® oferece sobre as linguagens de programação tradicionais são as bibliotecas integradas de controles e indicadores. Como o LabVIEW® foi projetado para engenheiros e cientistas, essa biblioteca inclui controles de contexto específicos, como botões, marcadores e *switches*, além de indicadores como medidores, escalas e termômetros. O conjunto de objetos do Painel Frontal (interface gráfica) desenvolvidos nesse trabalho foram personalizados e adaptados utilizando como base as bibliotecas disponibilizadas por usuários da comunidade LabVIEW® (POWELL, 2016). A Figura 29 mostra alguns exemplos de objetos desenvolvidos para a interface de usuário do programa supervisorio.

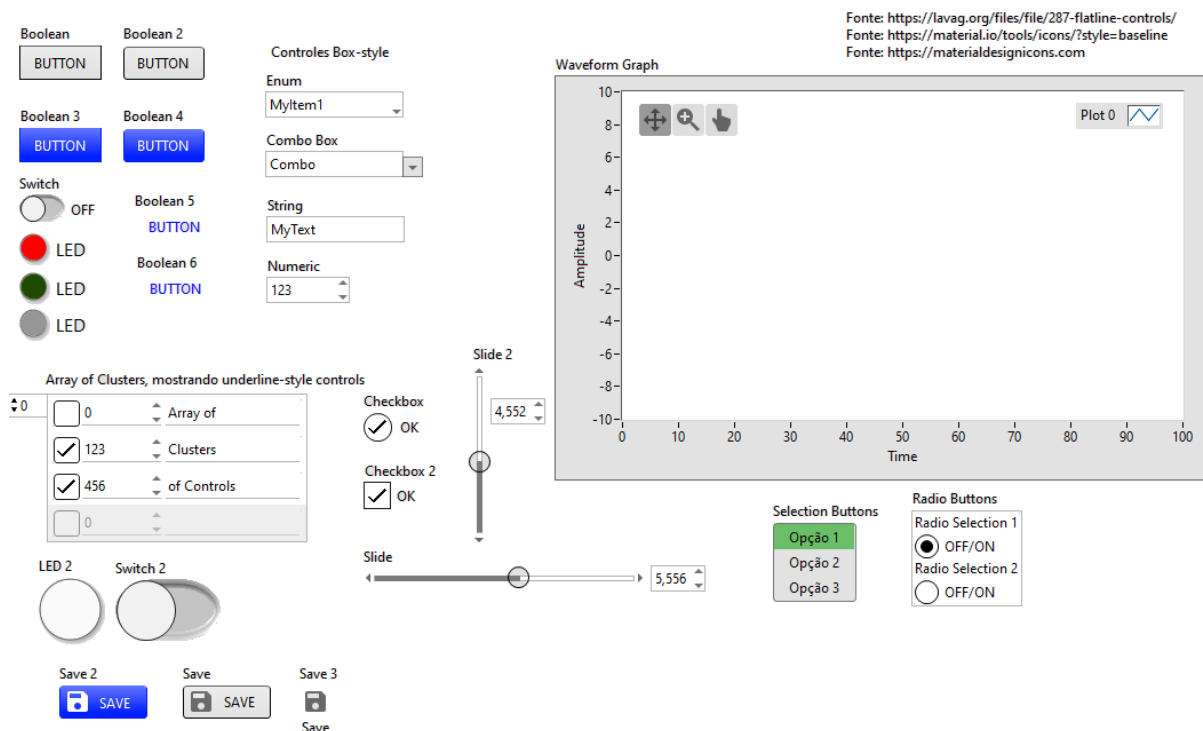


Figura 29: Exemplos de objetos desenvolvidos para a interface gráfica do programa supervisorio.

A cor escolhida para o menu do programa foi o azul, enquanto para o restante da interface foi utilizado a cor cinza claro. No entanto foi desenvolvida uma sub-rotina chamada “subvi_cores.vi” na qual o usuário pode trocar a cor de fundo dos painéis. Essa ferramenta pode ser modificada para alterar de forma fácil a cor da fonte e também dos ícones das barras presentes no programa.

A escolha do formato e estilo dos ícones dos botões seguiu a tendência minimalista utilizada no projeto de interface para dispositivos móveis. Conforme citado anteriormente, a geometria dos ícones mudou visando uma interface limpa e visualmente agradável, usando poucos efeitos e poucas cores na sua execução. Por esse motivo, foi adotado ícones baseados no estilo “ícones planos” (*flatline icons*), recomendado para interfaces limpas e para dispositivos móveis. Alguns ícones foram totalmente desenvolvidos para esse trabalho, enquanto outros foram adaptados dos ícones disponíveis no website *Material Design*, que disponibiliza de forma aberta (*open-source*) componentes, ícones, fontes e recursos (GOOGLE INC., 2019).

Como estratégia de modularização dos ícones, botões e outros objetos, foi utilizado a ferramenta Definições de Tipo (*Type Definitions*) do LabVIEW®, que consiste na criação de um único arquivo contendo o objeto personalizado. *Type Definitions* são controles personalizados (arquivos no formato *.ctl) que podem ser colocados em qualquer VI. Quando o arquivo .ctl é alterado, todos os VIs que usam esse controle também serão alterados, o que elimina a necessidade de alterar cada VI manualmente. Quando um *Type Definition* é usado, o LabVIEW® garante que o tipo de dados é o mesmo em qualquer lugar onde o controle é usado. Uma *Strict Type Definition* é ainda mais restritiva, onde LabVIEW® garante que quase todas as configurações e personalizações do objeto, como cores e tamanho, sejam as mesmas em todas as instâncias desse objeto no programa supervisorio.

A interface gráfica tem por objetivo aprimorar e facilitar a relação entre operador e o programa. Para isso foi utilizado nos ícones a ferramenta conhecida como *mouse hover* (ou *mouse over*), que é uma ação que acontece quando o usuário pára (*over*) ou “paira” (*hover*) o ponteiro do mouse em cima de um elemento específico de um objeto do Painel Frontal, causando um efeito visual que ajuda na interação homem-máquina. Essa ação pode, ainda, desencadear a ferramenta “*Tip Strip*”, que nada mais é que uma breve descrição da funcionalidade de um objeto, exibida quando o usuário deixa o cursor sobre um objeto por alguns decissegundos. Essas

informações geralmente aparecem em uma pequena caixa de texto próximas ao cursor do mouse. Adicionalmente, foi documentado informações importantes para o programador no *Context Help* do LabVIEW®, que é uma ferramenta de ajuda muito utilizada por programadores.

4.6. Rotinas de segurança

O comprimento máximo de queima é um parâmetro de segurança e interrompe o ensaio a qualquer momento em que seja atingido. Isso foi feito porque, dependendo da força utilizada no ensaio, durante a etapa de forjamento pode haver um avanço muito grande do pino, o qual está fixado por uma placa de castanhas. Com isso, há o risco de face do bloco tocar a placa de castanhas e danificá-la. Como a etapa de forjamento, diferentemente das outras etapas, é controlada pelo tempo e não pelo comprimento de queima, isso garante que não haverá um avanço exagerado do pino ao fim do reparo (FREITAS, 2014). Essa rotina de segurança está programada no CLP, porém, como redundância, não é permitido inserir valores de comprimento de queima maiores que 20 mm (curso do cilindro hidráulico da UPPA3) no programa supervisor.

A fim de se evitar que o usuário envie ao CLP parâmetros que não estejam na faixa de trabalho do equipamento, foi programado um limite de valores aceitos como *setpoints* durante a configuração dos ensaios. Além disso, o programa apresenta algumas funções de segurança, como por exemplo:

- **Verificação das faixas dos parâmetros de ensaio:** foram selecionadas faixas específicas de operação de rotação (estando os valores de rotação entre 1.000 e 3.000 rpm) e de força axial (estando os valores entre 2.000 e 40.000 kgf). Estas faixas estão restringidas no programa, porém caso o operador consiga de alguma forma entrar com valor fora das faixas pré-estabelecidas, antes de iniciar o programa retornará uma mensagem de erro e requisitará novamente a entrada dos parâmetros de ensaio.
- **Interrupção automática da coleta de dados em caso de travamento do motor hidráulico:** caso ocorra travamento do sistema de rotação, o CLP interrompe o processo e envia ao programa um comando que encerra a captura de dados. Os dados adquiridos até o momento do travamento são salvos e uma mensagem de erro é exibida ao usuário.

- **Botão “Parar Ensaio”:** durante todo o ensaio, é exibido um botão que pode ser acionado pelo operador a qualquer instante. Após acionado, esse booleano inicia a sequência de parada, enviando o comando de interrupção ao CLP e encerrando o ensaio atual. Quando a interrupção da coleta dos dados está configurada para ser realizada no modo manual, o operador deve clicar no botão “Parar Ensaio” para interromper a coleta dos dados e execução do salvamento desses dados.
- **Deteção de mudança no estado da Emergência:** caso haja alteração no estado do botão de emergência durante a execução do ensaio, o programa interrompe o salvamento de dados e exibe uma mensagem de erro ao usuário. O mesmo acontece caso haja perda de comunicação com o CLP, aumento excessivo na temperatura do óleo do sistema hidráulico, falha no cilindro ou motor hidráulico ou travamento do sistema de rotação.

Ao clicar no botão “Fechar”, é executado a rotina de fechamento do programa, que consiste nos seguintes passos:

1. Após o clique no botão “Fechar”, um Evento é enfileirado e resolvido na estrutura *Event Case* do VI principal (main.vi);
2. Uma mensagem com o comando FECHAR é enviada para a barra de status;
3. O notificador n_fechar é alimentado com o valor “verdadeiro”;
4. São encerradas as seguintes filas e notificadores: n_medições, n_travaVI, n_menuAberto, n_câmera, n_finalizarVídeo, q_medições, q_comandosSetpoints e q_travaVI;
5. Os laços de cada sub-rotina são encerrados. O VI principal aguarda uma resposta de cada subVI antes de passar para a próxima etapa do fechamento;
6. Depois de receber a confirmação de cada subVI, a fila n_fechar e o notificador q_fecharOK são encerradas;
7. O LabVIEW é encerrado.

4.7. Captura de vídeo

Para capturar os vídeos, foi necessário escolher um codificador (*codec*) que conseguisse manter uma boa qualidade de gravação e, ao mesmo tempo, comprimir o tamanho do arquivo de vídeo gerado (.avi). Optou-se pela escolha de um *codec* dentre os disponibilizados pela *National Instruments* junto ao LabVIEW® com o objetivo de facilitar a exibição dos vídeos capturados, pois, caso os vídeos necessitem ser transferidos para outros computadores, não haverá necessidade de instalar aplicativos ou *codecs* para exibição de vídeos.

Para escolha do *codec* foi realizado testes de gravação na câmera USB, onde foram gravados vídeos com duração de 30 segundos cada, com a resolução da câmera configurada como 1280 x 720 pixels (720p) e a taxa de captura em 30 quadros por segundo. Além disso, o *codec* não deveria causar mudanças no tempo de gravação, ou seja, sem incluir ou retirar quadros do *buffer* da câmera no arquivo, mantendo o padrão de 30 quadros por segundo (*fps*).

4.8. Procedimento de ensaio

Com o objetivo de validar a usabilidade do programa desenvolvido, foram realizados ensaios de validação. Na UPPA3, os ensaios de reparo por atrito podem ser realizados com um ou dois estágios. Considerando a sequência da execução do reparo apresentada na Figura 30, no caso da utilização de dois estágios, a etapa de aplicação de força e rotação pode ser dividida em duas sub-etapas, onde os valores de força aplicada são diferentes. No caso da utilização de apenas um estágio, utiliza-se apenas um valor de força nessa etapa.

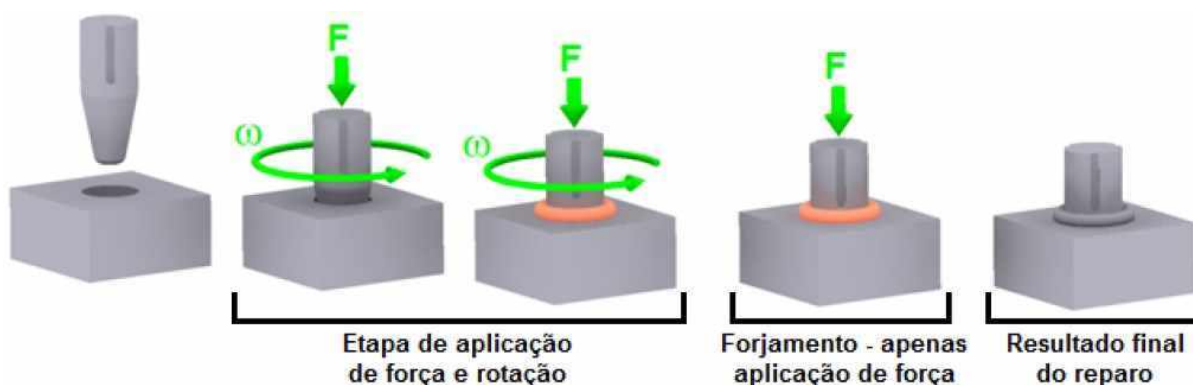


Figura 30: Etapas executadas no processo de reparo por atrito (FREITAS, 2014)

Para escolher qual equipamento – UPPA2 ou UPPA3 – utilizado no ensaio, deve-se mudar o conjunto de válvulas mostrado na Figura 31. Sensores fins-de-curso instalados nessas válvulas indicam ao CLP qual equipamento foi selecionado. O CLP então seleciona a programação adequada para que o controle dos sensores, atuadores e variáveis do processo seja executado de forma correta. Parâmetros de soldagem como rotação, força, torque, temperatura da amostra, tempo de forjamento e comprimento de queima são definidos pelo operador na sub-rotina “Ensaio” (ver tópico 4.1.4).

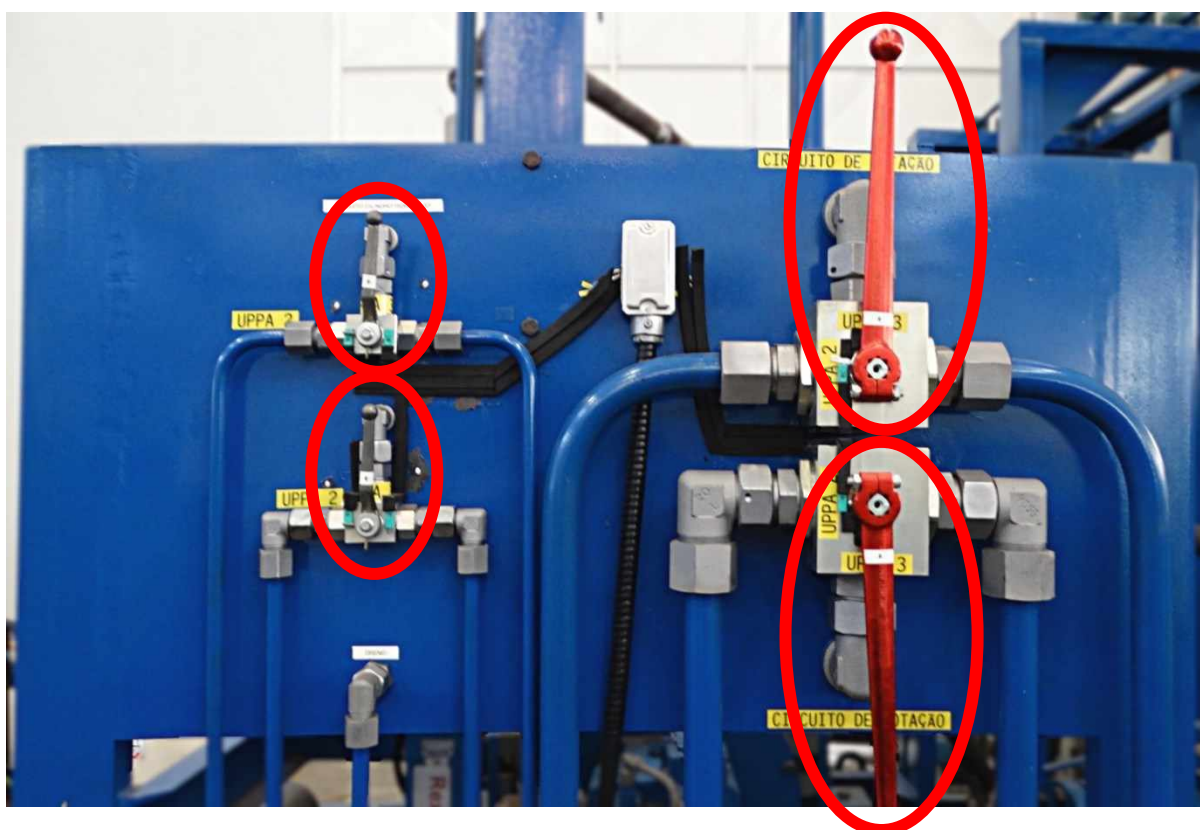


Figura 31: Para definir qual equipamento – UPPA2 ou UPPA3 – será utilizado na realização dos ensaios, deve-se alterar a posição das válvulas destacadas.

O usuário tem as opções de 1 ou 2 estágios para realização dos ensaios. Para um estágio, o processamento é realizado com valores de *setpoint* de rotação e força axial constantes até o final do processo. Para dois estágios, o processamento ocorre com um valor de *setpoint* inicial para força até que seja atingido o comprimento de queima definido no primeiro estágio, depois inicia o segundo estágio, no qual o valor da força aplicada muda, permanecendo até a etapa de forjamento. A etapa de forjamento é a última etapa do processo de reparo executado nas UPPAs 2 e 3. Nessa etapa, a rotação é zerada e aplica-se uma força de forjamento, que pode ter valor diferente da força aplicada no primeiro e/ou segundo estágios, permanecendo pelo tempo especificado no *setpoint* “Tempo de forjamento”.

Uma vez iniciado o ensaio, os dados adquiridos de Rotação, Força Axial, Torque e Comprimento de Queima são traçados nos gráficos e é realizado o acompanhamento dos valores lidos pelo sensor de posição. Enquanto a leitura de posição é menor que o “comprimento de queima” definido pelo operador, o CLP mantém os *setpoints* de rotação e força inalterados e segue enviando os dados adquiridos. Para o caso de 1 estágio, quando o valor da posição transpõe o “comprimento de queima” definido pelo operador, o programa passa à Etapa de Forjamento. Para o caso de 2 estágios, o programa atualiza as saídas dos *setpoints* de Rotação e Força Axial, e repete a etapa anterior até que o “comprimento de queima” do segundo estágio seja atingido, passando à etapa de Forjamento (FREITAS, 2014).

Conforme citado anteriormente, a etapa de Forjamento caracteriza uma etapa pós-processamento para acomodação, acabamento e melhoria na qualidade do material processado. Nesta etapa, inicialmente o CLP interrompe a rotação através do envio de um *setpoint* nulo ao controlador de rotação e, em seguida, atualiza o *setpoint* do controlador de força com o valor correspondente à “Força de Forjamento”. A duração desta etapa é definida pelo usuário através do parâmetro “Tempo de Forjamento”. Após este tempo, a pressão aplicada pelo eixo do cilindro hidráulico é aliviada através do envio de um *setpoint* nulo ao controlador da força. Na sequência, as válvulas direcionais são desabilitadas (FREITAS, 2014). O programa supervisorio interrompe automaticamente a aquisição da matriz dos dados adquiridos durante o processo e salva em arquivo de texto (.txt), porém caso o usuário queira monitorar a temperatura da amostra há a opção de interromper a aquisição de forma manual.

Para os ensaios, foi escolhida a geometria de pinos cônica. As dimensões do pino e do bloco utilizados foram definidas com base nos resultados obtidos por Pires (2007) que, em seu trabalho, avaliou o efeito da geometria na qualidade do reparo. O objetivo do ensaio é verificar a funcionalidade e integração do programa supervisorio com o CLP e o equipamento. A Figura 32 mostra o resultado esperado de ensaios de reparo por atrito realizados.

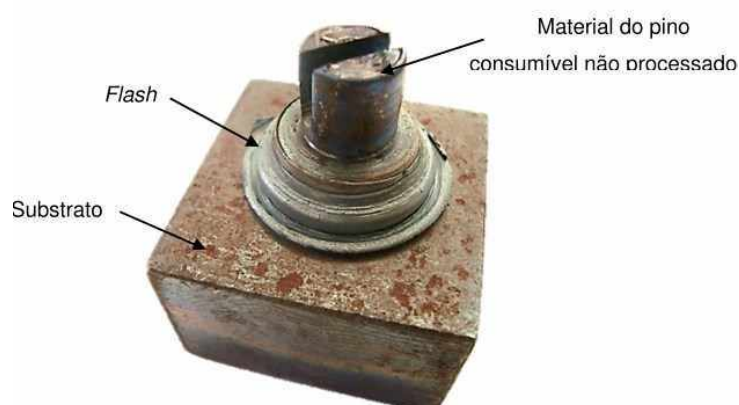


Figura 32: Amostra após a realização do processo de reparo por atrito (RODRIGUES, 2016).

CAPÍTULO V

5. RESULTADOS E DISCUSSÕES

5.1. Programa supervisorio

A Figura 33 mostra a interface do programa supervisorio desenvolvido para as UPPA2 e UPPA3, chamado **SUPPA32**. A sub-rotina mostrada atualmente no Painel 4 é a “Home”, onde o usuário faz movimentações manuais do cilindro hidráulico para posicionar corretamente o pino e o bloco da amostra. Pode, também, usar essa janela para acionar o controle manual do sistema de rotação, realizar movimentação manual do cilindro para diversas posições ou utilizar o equipamento para funções de prensa hidráulica. Operações de manutenção e calibração do equipamento podem ser executadas no botão “Acessar válvulas”, que permite alterar a vazão da bomba de pistão axial do motor hidráulico, a operação manual das válvulas limitadoras de pressão e da válvula direcional proporcional de vazão do cilindro e motor hidráulico.

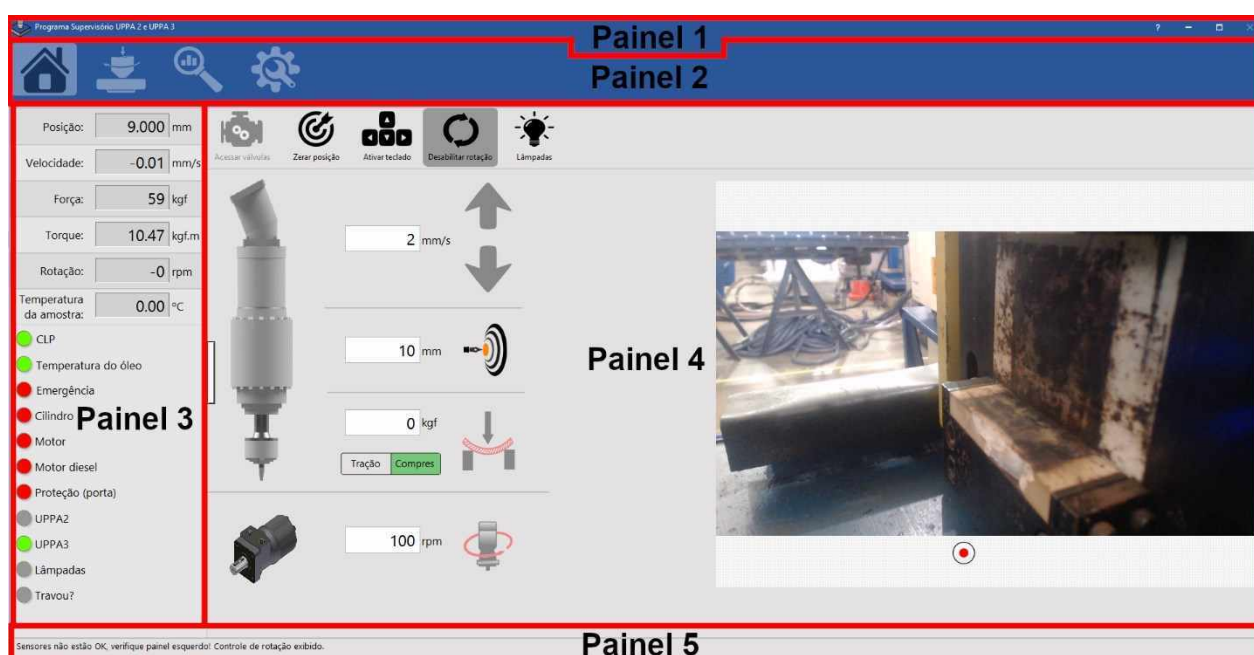


Figura 33: Interface do programa supervisorio SUPPA32.

A janela de inserção de parâmetros de ensaio é mostrada na Figura 34. Nela pode ser selecionado o número de estágios do ensaio (um ou dois estágios) e podem ser inseridos os respectivos valores de força, rotação, comprimento de queima, força de forjamento, tempo de forjamento, comprimento máximo de queima e nome da pasta de salvamento dos dados do ensaio. No local indicado para o salvamento dos dados, são armazenados os respectivos valores de força, rotação, comprimento de queima e torque obtidos ao longo do ensaio e a filmagem do ensaio obtida através da câmera. Caso a pasta de salvamento não seja informada, uma mensagem de erro é informada na barra de status. O usuário pode clicar no botão “Gerar nome” para criar automaticamente o nome da pasta de salvamento. Ao clicar no botão “Iniciar Ensaio”, o programa supervisorio verifica todos os parâmetros informados e, caso estejam corretos, inicia do ensaio. Ao fim do ensaio é informado ao usuário o caminho completo do diretório de salvamento dos dados.



Figura 34: Sub-rotina “Ensaio” do programa SUPPA32, onde o usuário informa os parâmetros de ensaio.

A Figura 35 mostra o local ao qual podem ser carregados os dados de ensaios realizados anteriormente e também comparar os resultados de dois ensaios. Os dados são filtrados para exibir somente a região de interesse⁶ do ensaio. Clicando no botão “Ver dados brutos”, são exibidos todos os dados coletados. O *range* das escalas pode ser alterado clicando nos extremos de cada eixo do gráfico e alterando o valor indicado; se houver alteração no valor da escala de tempo, todos os outros gráficos são ajustados de forma a manter uniformidade nos dados exibidos.

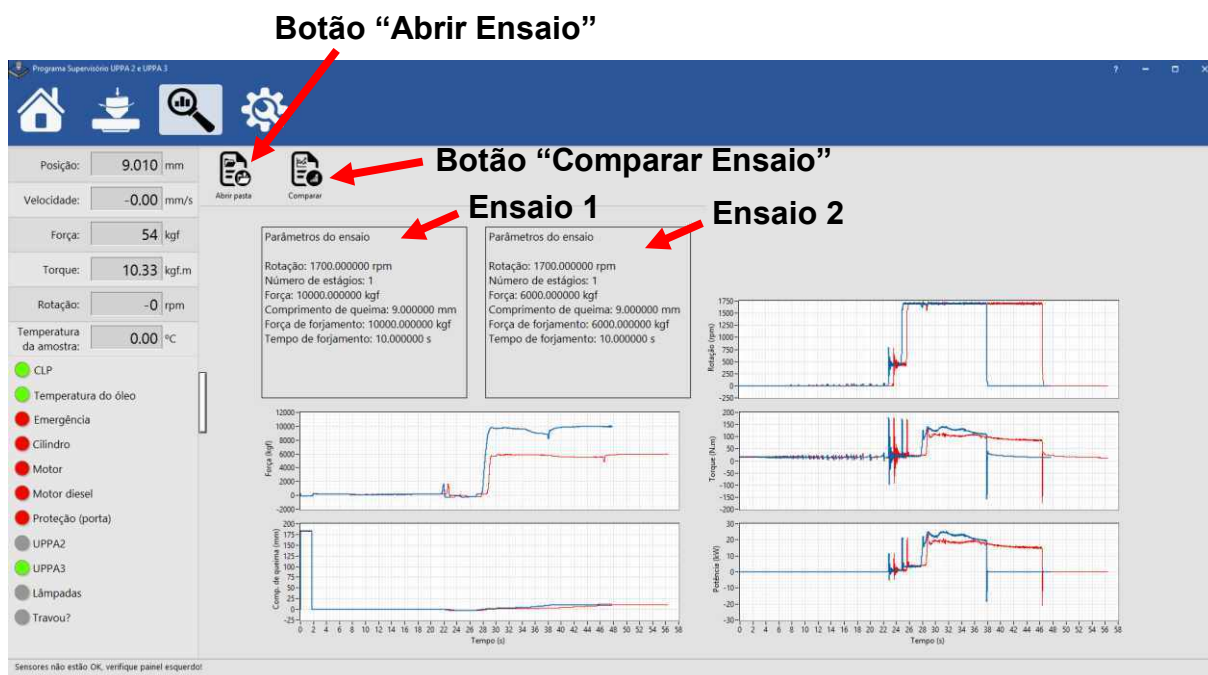


Figura 35: No menu “Análise” é possível carregar ensaios e comparar resultados

Como mencionado anteriormente, os dados são salvos em arquivos .txt, facilitando a utilização dos resultados em outros *softwares*, como *Matlab*® e *Excel*®. Em cada ensaio são gerados três arquivos de texto: o primeiro contém os dados; o segundo contém os parâmetros de ensaio inseridos pelo operador; o terceiro contém a data e hora de realização de ensaio, a descrição e unidade de medida de cada coluna da matriz de dados.

⁶ A região de interesse é definida como o local de contato entre pino e bloco. O contato permanece até que o reparo atinja o comprimento de queima desejado, que indica o fim do processo

5.2. Testes das rotinas do programa

Durante a execução do programa, foi testado a funcionalidade das rotinas de segurança, na qual foi averiguado que o programa consegue reestabelecer comunicação com o CLP após uma desconexão, ler o estado da emergência e demais dispositivos de segurança, bloqueando e/ou impedindo ações de comando, quando necessário.

Além disso, realizou-se avaliações na execução das rotinas desenvolvidas para facilitar a utilização do programa supervisorio pelo operador. Pode ser citado, por exemplo, o bloqueio do acesso a certos menus e botões do programa supervisorio quando há movimentação manual dos atuadores (cilindro e motor hidráulico); bloqueio do botão “Iniciar Ensaio” enquanto os sensores e atuadores estiverem desligados ou inativos; indicação visual (via botão que simula um LED) de qual sensor/atuador da UPPA não está ligado; exibição de mensagens de ajuda quando o usuário pára o mouse sobre um botão (ferramenta *tip strip*) e a opção de esconder temporariamente o painel que exibe os valores das medições obtidas pelo CLP em tempo real.

Após a verificação e validação dessas rotinas, foram testadas as rotinas responsáveis pelo salvamento de dados. Para isso, foi desenvolvido um subVI que simula o envio de dados do CLP para o programa supervisorio. O programa supervisorio conseguiu exibir e filtrar corretamente os dados enviados pelo simulador, salvando os resultados em arquivos de texto no diretório informado pelo usuário no menu “Configurações”.

Em seguida, foi testado o subVI “Análise”, que exibe os resultados de ensaios. O novo programa supervisorio conseguiu abrir e exibir corretamente os resultados salvos nos arquivos de texto; conseguindo, ainda, exibir resultados obtidos em dados salvos pelo antigo programa supervisorio. O filtro que exibe apenas a região de interesse do gráfico funcionou corretamente para todos os ensaios analisados, facilitando a análise dos resultados.

A rotina que reajusta o tamanho da janela do programa, tamanho dos painéis, posicionamento dos botões e textos de acordo com a resolução do monitor do computador funcionou ajustando essas características conforme a resolução do monitor muda. Porém, o tamanho dos botões e tamanho das fontes deve ser ajustado de forma manual pelo programador para manter a proporção adequada à resolução do monitor.

5.3. Gravação de vídeos

Os resultados obtidos na captura dos vídeos estão apresentados na Tabela 3, onde a primeira coluna indica o *codec* utilizado na gravação, a segunda coluna indica o tamanho final do arquivo a ser armazenado no disco e a terceira coluna indica se ocorreu algum problema na utilização do *codec* especificado.

Durante os testes, foi identificado que apenas três codificadores conseguiram executar a gravação sem falhas. E, destes três codificadores, o que teve os melhores resultados foi “Motion JPEG (MJPEG) (NI Vision)”. E o motivo disso foi porque o mesmo conseguiu gerar um arquivo que ocupa menos espaço de armazenamento, mas, com uma qualidade de vídeo igual a dos outros codificadores testados. Devido a isso, ele foi adotado como o codificador padrão para as gravações.

Tabela 3: Resultados obtidos na gravação de vídeos com duração de 30 segundos.

Nome do <i>Codec</i>	Tamanho do arquivo (MB)	Observação
Nenhum codec (gravação sem compressão)	1945,6	
Cinepak Codec by Radius	13,9	Não conseguiu realizar a gravação
Codec IYUV Intel	1013,76	Alguns quadros não foram gravados
Microsoft Video 1 (CRAM)	130,0	
FF Video Codec 1 (FFV1) (NI Vision)	135,0	Alguns quadros não foram gravados
Motion JPEG (MJPEG) (NI Vision)	28,4	
Y800 Uncompressed Grey Scale (NI Vision)	-	Não conseguiu realizar a gravação
YUV 4:2:0 Planar (NI Vision)	1177,6	

5.4. Ensaios de reparo por atrito

Após a realização dos testes, foram feitas correções e validações de todas as funções do programa. Também foram executados ensaios de reparo por atrito para validar o novo programa supervisor.

A Figura 36 mostra a janela do programa SUPPA32 durante a realização de ensaios. Na imagem é possível ver os gráficos, imagem da câmera e os botões responsáveis por iniciar um novo ensaio e parar ensaio atual. Nessa janela os gráficos também foram programados para mostrar apenas a região de interesse do ensaio. O quadro localizado no canto superior esquerdo do Painel 4 mostra os parâmetros de ensaio.

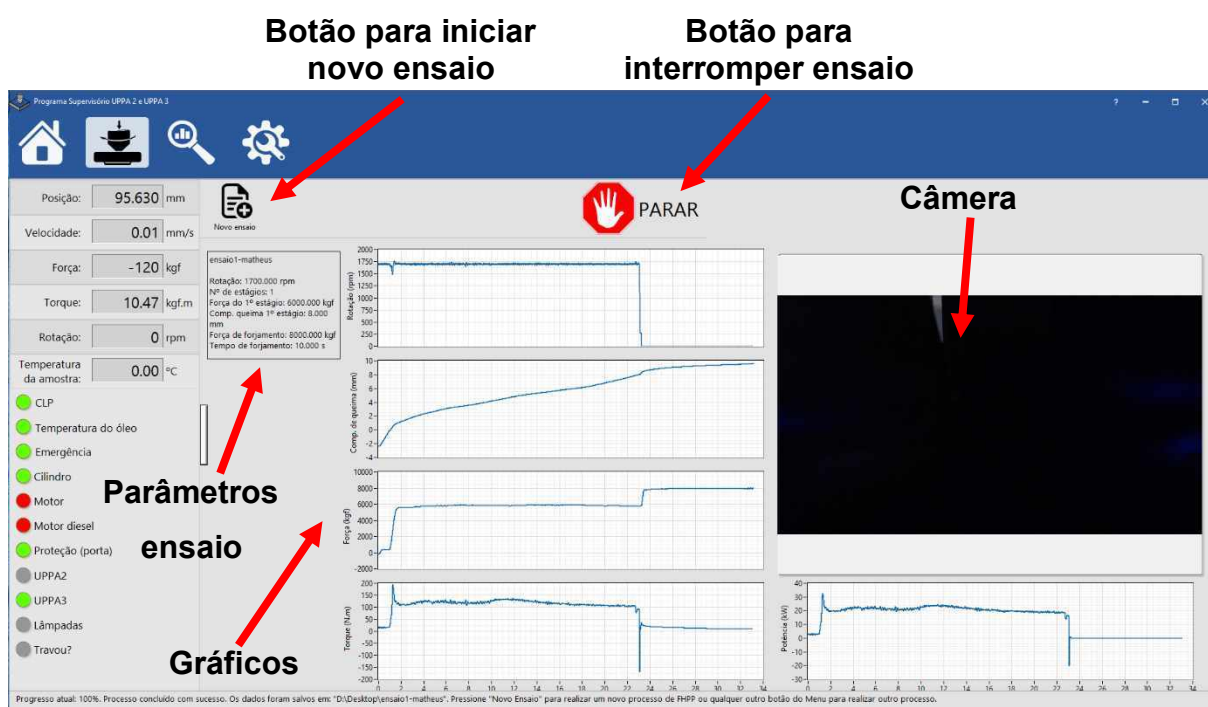


Figura 36: Janela do programa SUPPA32 durante a realização de ensaios. Os gráficos exibem os dados relativos à região de interesse do ensaio.

O primeiro ensaio foi realizado com um estágio, rotação 1700 rpm, força 6000 kgf, comprimento de queima 7 mm, força de forjamento 6000 kgf e tempo de forjamento 10 s. A Figura 37 mostra todos os dados coletados durante o ensaio, sem o filtro que exibe apenas a região de interesse. Nos gráficos, o contato entre pino e bloco (início do reparo por atrito) ocorre no momento em que o comprimento de queima atinge o valor zero, o que corresponde ao instante zero ($t = 0$) nos gráficos.

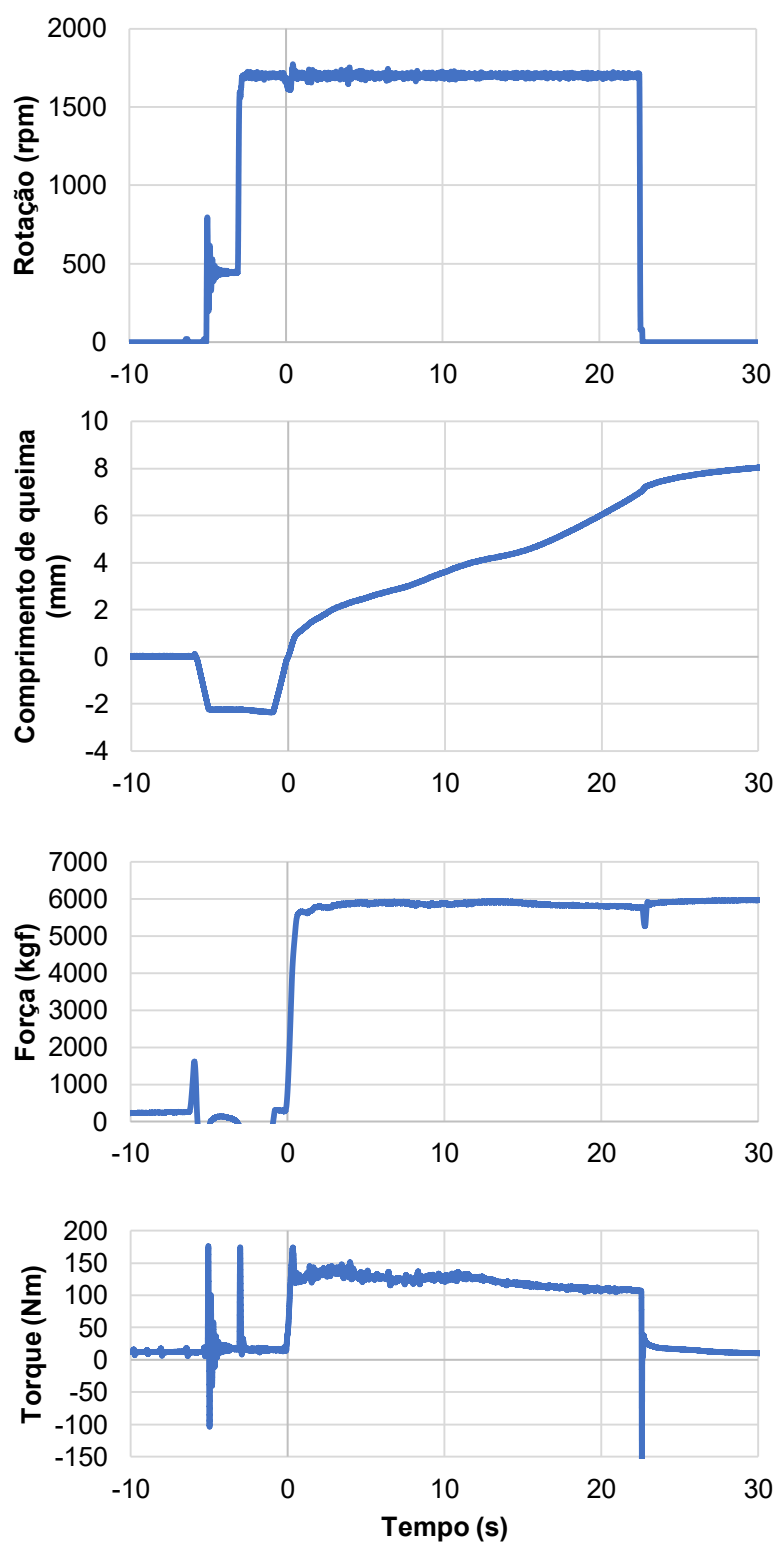


Figura 37: Resultados para o primeiro ensaio de reparo por atrito realizado na UPPA3. Os gráficos mostram todos os dados coletados durante o ensaio, sem o filtro que exclui dados coletados fora da região de interesse.

Pode ser observado que para tempos anteriores ao contato ($t < 0$), há um pico inicial de força, seguido de valores negativos, que é referente ao processo de detecção do ponto de contato entre pino e bloco. Após o contato, o CLP envia o comando para o pino recuar 2 mm, depois o sistema de rotação é ligado, onde há o pico inicial de rotação proveniente da partida do motor hidráulico.

O primeiro pico de torque ocorre quando $t = -5,05$ s, momento do contato entre pino e bloco (detecção da posição inicial), quando o sistema de rotação ainda não foi acionado; o segundo pico de torque ocorre em $t = -3,03$ s, momento de partida do motor hidráulico. O terceiro pico de torque ocorre no momento em que há o contato entre pino e bloco, e seu valor chega a 171,3 Nm. Devido ao atrito, o material começa a se aquecer, diminuindo a resistência mecânica, provocando uma redução no torque. Depois, com a remoção dos óxidos e imperfeições das superfícies dos materiais, redução do tamanho das asperidades e aumento da área de contato real entre pino e bloco, o torque vai aumentando. Devido ao crescente aquecimento em decorrência do atrito e consequente redução da resistência mecânica do material, o toque volta a diminuir ao longo do tempo.

Quando o valor do comprimento de queima desejado é atingido, o motor hidráulico cessa a rotação. Nesse instante há grande oscilação na força, que acontece devido ao sistema hidráulico de força estar interligado com o de rotação, a grande variação na linha de rotação provoca um distúrbio na pressão de alimentação do sistema de força. Após essa oscilação, o sistema volta a se estabilizar e a força aplicada assume o valor definido no parâmetro “Força de forjamento”, com erro máximo menor que 2%.

Quando a rotação é levada a zero, pode ser observado um vale de torque, gerado pelo próprio processo. A rápida redução na rotação, aliada ao desligamento do motor diesel e despressurização do sistema de rotação causam a redução brusca na geração de calor e, consequentemente, a redução na temperatura do material, provocando um aumento na rigidez do material, gerando um ciclo que causa o vale.

O comprimento de queima tem seu valor inicial aumentado rapidamente, em vista da pequena área de contato entre pino e bloco. Com o aumento na área de contato – real e aparente – entre pino e bloco, a taxa de queima vai reduzindo. Porém, com o aumento da temperatura do material e, por consequência, redução de seu limite de escoamento a taxa de queima volta a crescer. Ao fim da rotação, a taxa de queima reduz bruscamente, pois não há geração de calor. Na etapa de forjamento, o aumento

no comprimento de queima refere-se à acomodação do material deformado do pino durante a atuação da força de forjamento.

O segundo ensaio foi realizado com um estágio, rotação 1700 rpm, força 6000 kgf, comprimento de queima 8 mm, força de forjamento 8000 kgf e tempo de forjamento 10 s. Os resultados de rotação e comprimento de queima são apresentados na Figura 38, enquanto os resultados de força e torque são apresentados na Figura 39. Os resultados mostram que os *setpoints* foram atingidos e que o CLP conseguiu realizar o controle da força e rotação durante todo o ensaio. O programa supervisor exibiu e registrou corretamente os dados em arquivos .txt, que podem ser exportados para outros *softwares*.

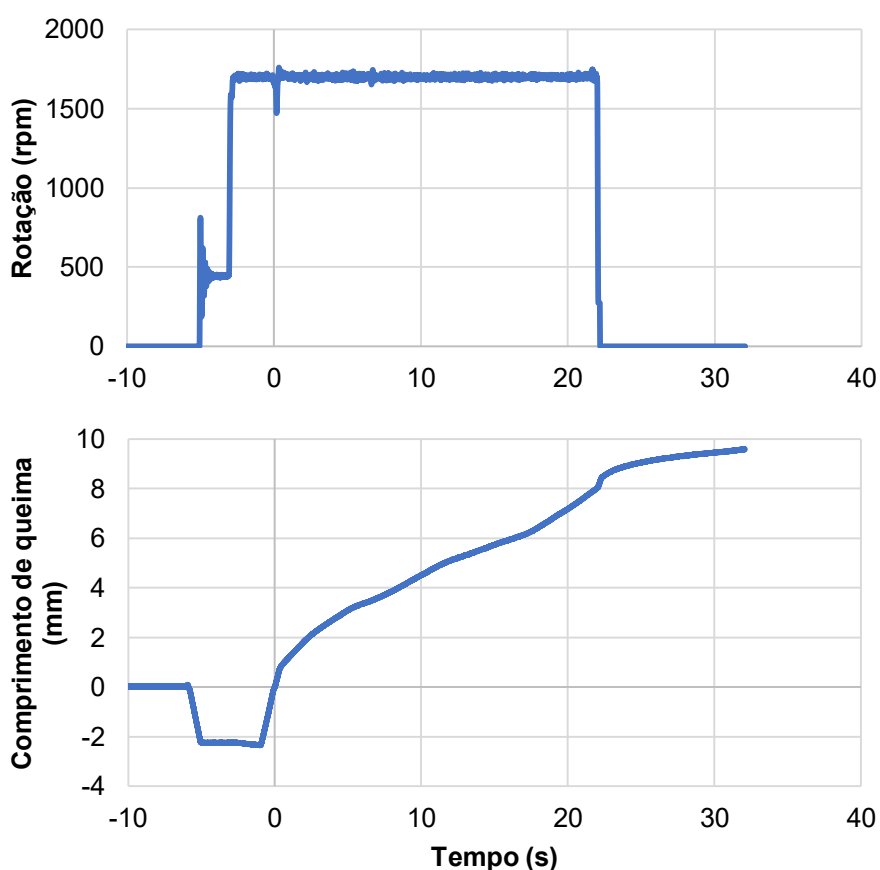


Figura 38: Resultados de rotação e comprimento de queima do segundo ensaio de reparo por atrito realizado na UPPA3.

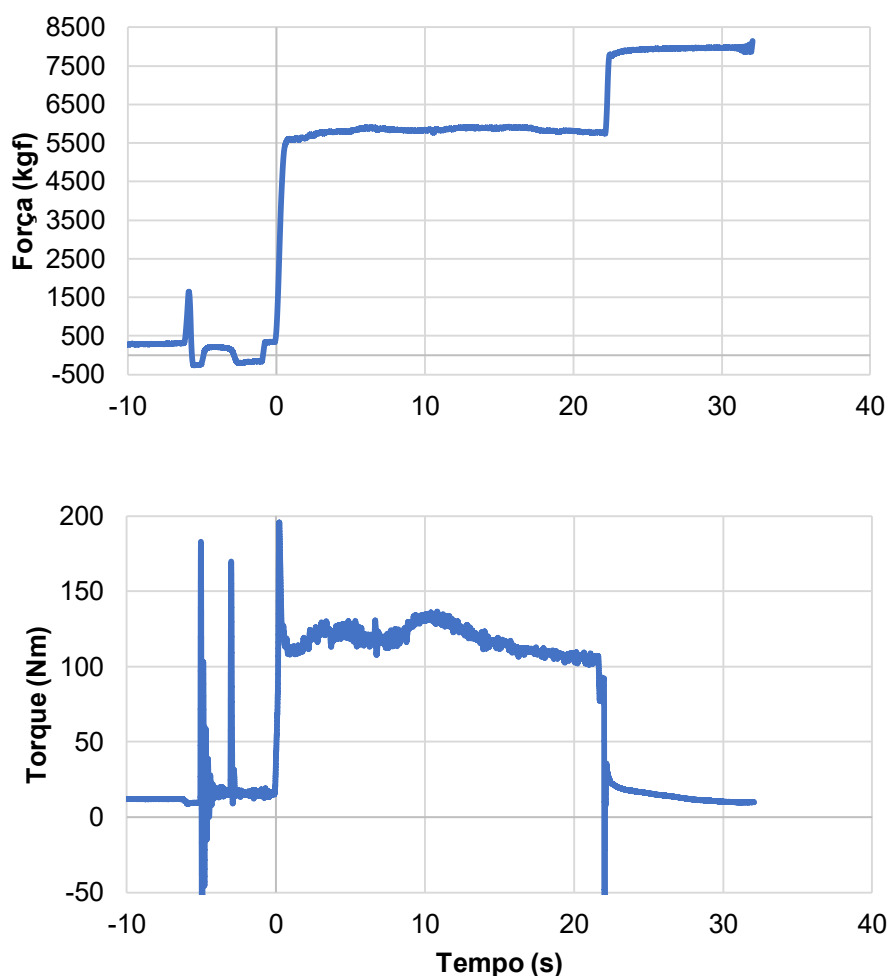


Figura 39: Resultados de rotação e comprimento de queima do segundo ensaio de reparo por atrito realizado na UPPA3.

Em ambos ensaios, o programa supervisorio obteve êxito em salvar os dados coletados pelo CLP. Além disso, o operador do equipamento pode monitorar os ensaios de reparo por atrito em tempo real através do acompanhamento visual dos gráficos e da imagem capturada pela câmera. Portanto, é possível afirmar que o programa supervisorio está apto a realizar ensaios de reparo por atrito nos equipamentos UPPA.

5.5. Validação do programa pelos operadores do equipamento

A última etapa de validação foi a apresentação do programa e sua interface gráfica aos operadores do equipamento. A função deles foi a de testarem as funcionalidades do programa de duas formas:

a) por conta própria – com objetivo de verificar se a interface gráfica é simples, amigável e de fácil aprendizado;

b) após a explicação detalhada de todos os botões, funcionalidades e funções dos objetos disponíveis na interface gráfica, bem como explicação de algumas particularidades sobre a estrutura de comunicação entre as rotinas do programa – com o objetivo de localizar possíveis falhas na lógica de estrutura do programa.

Após os testes, os operadores sugeriram mudanças em algumas partes das janelas do programa. A partir dessas observações, foi modificado o tamanho de fonte, imagens de ícones, posição de objetos, texto exibido em *tip strips* e outros detalhes visuais em alguns objetos da interface gráfica, visando aprimorar o uso do programa supervisor.

Após as implementações, o programa supervisor foi submetido novamente a análise dos operadores, que conseguiram utilizar as funções e executar a sequência necessária para realizar ensaios de forma correta, sem grandes dificuldades de uso na nova interface gráfica. Além disso, relataram que a interface simplificada e limpa é amigável, de fácil utilização, que as imagens dos botões estão coerentes com a função dos mesmos e a lógica de distribuição das janelas Home, Ensaio, Análise e Configurações facilita a operação segura do equipamento.

CAPÍTULO VI

6. CONCLUSÃO

Neste trabalho foi desenvolvido um programa supervisorio para operação de equipamentos de processamento de pinos por atrito. Foi realizado também a estruturação modular do programa supervisorio, de forma que este opere de forma otimizada equipamentos de ensaio e seja usado como base para os próximos programas supervisorios a serem desenvolvidos no LTAD. Os resultados obtidos permitem concluir que o programa desenvolvido atendeu os objetivos propostos nesse trabalho. Com o programa supervisorio desenvolvido é possível concluir que:

- a) Os resultados obtidos após a implementação do sistema supervisorio estão de acordo com os dados disponíveis na literatura;
- b) A combinação dos resultados obtidos nos ensaios, com a facilidade de operação do equipamento com o novo programa, culminou na adoção do novo sistema (SUPPA32) como programa supervisorio da UPPA2 e UPPA3;
- c) As rotinas de segurança automatizadas do programa supervisorio, em conjunto com as rotinas de segurança do CLP, permitem a operação segura do equipamento;
- d) O programa supervisorio mostra os gráficos do ensaio em tempo real, possibilitando o monitoramento e intervenção humana caso alguma falha seja detectada durante a execução dos ensaios;
- e) A interface gráfica desenvolvida de acordo com as novas tendências de interface limpa e amigável foi de fácil aprendizado e assimilação pelos operados do equipamento;

- f) A estrutura modular desenvolvida funcionou conforme os requisitos de projeto, onde as rotinas podem ser executadas individualmente para testes e depuração, bem como podem ser desenvolvidos subVIs para uso em múltiplos módulos do programa;
- g) A estrutura de programação mestre-escravo (usando notificadores) em conjunto com a estrutura produtor-consumidor (usando filas) é adequada para a comunicação entre os diversos subVIs de um programa desenvolvido no LabVIEW®;
- h) A estrutura-base, interface gráfica e de comunicação entre as partes do programa podem ser utilizadas no desenvolvimento de novos programas supervisórios em novos projetos do LTAD.

CAPÍTULO VII

7. TRABALHOS FUTUROS

Em continuidade ao trabalho apresentado, sugerem-se estudos e possíveis aprimoramentos no código desenvolvido:

1. Em relação a interface gráfica, adicionar a opção de redimensionamento automático dos ícones e objetos conforme resolução do monitor do computador. Para isso pode ser estudado a usabilidade da biblioteca *LAVA UI Tools*, desenvolvida pela comunidade do LabVIEW® para utilização em códigos com redimensionamento e reposicionamento;
2. Na modularização pode ser usado, ainda, outras ferramentas de transferências de dados como *Action Engines*, *Global Variables* ou *Data Value Reference* (DVR). Essas estruturas, em especial DVR, podem ser adequadas caso o programa desenvolvido necessite de alto desempenho computacional;
3. Com relação as modificações sugeridas pelos usuários que podem ser desenvolvidas futuramente, citam-se: alterar tamanho e quantidade de gráficos exibidos, de forma que o usuário possa escolher qual variável será exibida; alterar as imagens exibidas no programa quando a UPPA2 for usada; criar um sistema de “temas” para o programa como claro/escuro.

REFERÊNCIAS BIBLIOGRÁFICAS

ANP. **Brasil se torna maior produtor de petróleo da América Latina**. Governo do Brasil. Brasília. 2017.

BLAKEMORE, G. R. **Friction stud welding in hazardous areas**. 9. ed. USA: Welding & Metal Fabrication, v. 61, 1993.

BORGES, B. M. **Instrumentação, controle e operacionalização de uma unidade de reparo por atrito com capacidade de carga de 500 kN**. Dissertação de mestrado. Uberlândia: Universidade Federal de Uberlândia. 2011. p. 99.

CAIXETA, L. A. **Otimização de parâmetros de processamento de pinos por atrito em unidade com capacidade de 245 kN**. Dissertação de mestrado. Uberlândia: Universidade Federal de Uberlândia. 2011. p. 122.

CROSSLAND, B. **Friction Welding - Recommended Practices for Friction Welding**. USA: In Contemporary Physics, p. 559-574, v. 12, 1971.

DE LIMA FILHO, L. C. P. **Desenvolvimento de um programa para controle do processo de furação utilizando um equipamento de reparos por atrito**. Projeto de fim de curso. Uberlândia: Universidade Federal de Uberlândia. 2008. p. 52.

ELLIS, C. R. G. **Continous Drive Friction Welding of Mild Steel**. Welding journal. Miami, FL: American Welding Society. 1972. p. 197.

ESCOBAR, L. O impacto das cores na interface. **Homem Máquina Design**, 2013. Disponível em: <<https://www.homemmaquina.com.br/design/cores-na-interface/>>. Acesso em: 21 set. 2018.

FORMOSO, C. M. **Projeto e desenvolvimento de um sistema de instrumentação e controle para uma unidade de processamento de pinos por atrito portátil**. Dissertação de mestrado. Uberlândia: Universidade Federal de Uberlândia. 2012. p. 88.

FORMOSO, C. M. et al. Desenvolvimento de um programa de controle e monitoramento de uma unidade de processamento de pinos por atrito para o reparo de dutos e tubulações de transporte de derivados de petróleo. **19º POSMEC - Simpósio do Programa de Pós-graduação**, Uberlândia-MG, 2009. 2.

FREITAS, D. S. **Controle de força e rotação de uma unidade de reparo por atrito usando controlador PID e inteligência artificial**. Dissertação de mestrado. Uberlândia: Universidade Federal de Uberlândia. 2014. p. 191.

GOOGLE INC. Material Design Icons. **Material Design**, 2019. Disponível em: <<https://material.io/resources/icons/?style=baseline>>. Acesso em: 22 jan. 2019.

HALLOCK, J. Colour Assignment: Favorite color. **Joe Hallock**, 2003. Disponível em: <<http://www.joehallock.com/edu/COM498/preferences.html>>. Acesso em: 22 set. 2018.

HWANG, H. F. **Desenvolvimento, projeto, construção e teste de um cilindro de reparo por atrito portátil**. Dissertação de mestrado. Uberlândia: Universidade Federal de Uberlândia. 2010. p. 78.

MAIA, L. Design de interface: qual a importância? **Agência Duo-Me Design**, 2016. Disponível em: <<https://agenciaduo.me/blog/design-de-interface-voce-sabe-o-que-e-e-qual-a-importancia/>>. Acesso em: 21 set. 2018.

MARTINAZZI, D. et al. **Mapeamento de inclusões não metálicas do tipo sulfetos de manganês (MnS) no processo de soldagem por fricção com pino consumível em aço C-Mn**. 69º Congresso Anual da ABM. São Paulo: Laboratório de Metalúrgica Física (LAMEF). 2014. p. 14.

MEYER, A. **Friction Hydro Pillar Processing – Bonding Mechanism and Properties**. Braunschweig: Technische Universität Braunschweig, 2003. 123 p.

MUSGRAVE, D. Windows Logos through the years. **Blog Windows**, 2012. Disponível em: <<https://blogs.msdn.microsoft.com/developingfordynamicsgp/2012/09/16/windows-logos-through-the-years/>>. Acesso em: 09 jul. 2019.

NATIONAL INSTRUMENTS. **LabVIEW Basics II: Development Course Manual**. Austin, Texas: National Instruments, v. II, 2008.

NATIONAL INSTRUMENTS. Datalogging and Supervisory Control. **LabVIEW Modules**, 2019. Disponível em: <<http://www.ni.com/pt-br/shop/select/labview-datalogging-and-supervisory-control-module>>. Acesso em: 22 jan. 2019.

NICHOLAS, E. D. **Friction Hydro Pillar Processing**. 11ª Annual North American Research Conference. USA: [s.n.]. 1995. p. 9.

NIXON, J. **Effects of Water Depth on Friction Stud Welding Parameters**. School of Welding and Joining Technology. Cranfield, England: Cranfield University. 1986.

OPC FOUNDATION. What is OPC? **OPC Foundation - The Industrial Interoperability Standard**, 2019. Disponível em: <<https://opcfoundation.org/about/what-is-opc/>>. Acesso em: 05 jun. 2019.

PAES, M. T. P. **Processamento termomecânico de pinos de aço no interior de cavidades de aço C-Mn através da soldagem por atrito rotacional.** Tese de doutorado. Rio de Janeiro: Universidade Federal do Rio de Janeiro. 2013. p. 318.

PETROBRAS. Exploração e Produção de Petróleo e Gás. **Petróleo Brasileiro S.A (Petrobras)**, 2019. Disponível em: <<http://www.petrobras.com.br/pt/nossas-atividades/areas-de-atuacao/exploracao-e-producao-de-petroleo-e-gas/>>. Acesso em: 11 abr. 2019.

PIRES, R. R. **Efeitos da geometria, da força axial e da rotação no reparo por atrito.** Dissertação de mestrado. Uberlândia: Universidade Federal de Uberlândia. 2007. p. 68.

POWELL, J. Flatline Controls 1.8.0. **LabVIEW Advanced Virtual Architects (LAVA)**, 2016. Disponível em: <<https://lavag.org/files/file/287-flatline-controls/>>. Acesso em: 22 jan. 2019.

PRADO, D. F. **Influência da geometria, comprimento de queima e modo de aplicação da força no reparo por atrito com furos passante em aço inoxidável duplex.** Trabalho de conclusão de curso. Uberlândia: Universidade Federal de Uberlândia. 2018. p. 80.

RODRIGUES, F. B. **Desenvolvimento, Projeto e Construção de Sistemas e Acessórios Mecânicos Destinados à Automação de um Cilindro de Reparo por Atrito Portátil.** Dissertação de mestrado. Uberlândia: Universidade Federal de Uberlândia. 2016. p. 191.

SOUZA, R. J. D. **Desenvolvimento, projeto e construção de um equipamento de reparo de trincas por atrito.** Dissertação de mestrado. Uberlândia: Universidade Federal de Uberlândia. 2006. p. 87.

THOMAS, W. M. **Friction Plug Extrusion.** GB nº 2 306 365 A, The Welding Institute 1997.

Apêndice A

Estrutura da nomenclatura usada no programa

SUPPA32

Estrutura da nomenclatura usada no programa SUPPA32

A1. ABREVIATURAS

Tabela A1: Lista de abreviaturas usadas nos nomes das variáveis e arquivos

Abreviatura	Significado
C	Cluster
B	Boolean
E	Enum
I32	Inteiro de 32 bits
U32	Inteiro de 32 bits (positivo)
F64	Double (DBL)
S	String
A	Array (Vetor)
Q	Queue (Fila)
N	Notifier (Notificador)
M	Matriz
K	Constante
IMG	Imagem gerada pelo NI-IMAQdx
UE	User Event
XY	Gráficos

A2. REMETENTES

Tabela A2: Lista de remetentes de notificação e fila

Número	Remetente
-1	VI principal (main)
0	SubVI – Medições
1	SubVI – Espaço de trabalho 1
2	SubVI – Espaço de trabalho 2
3	SubVI – Espaço de trabalho 3
4	SubVI – Espaço de trabalho 4

A3. BARRA DE STATUS

Tabela A3: Lista de comandos para processar na barra de status

Comando	O que faz?
ATUALIZAR	Atualiza mensagens exibidas na barra de status
FECHAR	Exibe mensagem de saída na barra de status
RESET	Apaga todo o conteúdo exibido na barra de status

A4. VETORES

Vetores da fila de comandos e setpoints

Tabela A4: Vetor de comandos (u32_comandos)

Valor	Função	Variável correspondente CLP
0	Ocioso	
1	Parar ensaio	Logic.Application.UserVarGlobal.STOP
2	Iniciar ensaio	Logic.Application.Subprog_FHPP.b inicia_fhpp
3	Recuar cilindro manual	Logic.Application.Subprog_ativa_Velocidade.bativa_recuo
4	Avançar cilindro manual	Logic.Application.Subprog_ativa_Velocidade.bativa_avanco
5	Ir para posição manual	Logic.Application.Subprog_ativa_Velocidade.bativa_posicao
6	Força de tração manual	Logic.Application.Subprog_ativa_forca.bativa_forca (valores positivos)
7	Força de compressão manual	Logic.Application.Subprog_ativa_forca.bativa_forca (valores negativos)
8	Zerar posição manual	Logic.Application.UserVarGlobal.C_Tarapos
9	Rotação manual	Logic.Application.Subprog_ativa_Rot.bativa_rot
10	Ligar lâmpadas	Logic.Application.UserVarGlobal.Liga_lampadas_m
11	Ativar pressão do cilindro	Logic.Application.UserVarGlobal.Aciona_pressao_cil_m
12	Ativar vazão do cilindro	Logic.Application.UserVarGlobal.Aciona_vazao_cil_m
13	Alterar vazão bomba do motor hidráulico (pos. 29)	Logic.Application.UserVarGlobal.SetValvBombaMHPos29
14	Ativar pressão de rotação	Logic.Application.UserVarGlobal.Aciona_pressao_rot_m
15	Ativar vazão de rotação	Logic.Application.UserVarGlobal.Aciona_vazao_rot_m

16	Ajustar válvula de pressão do cilindro	(não implementado)
17	Ajustar válvula proporcional do cilindro	Logic.Application.UserVarGlobal.abValv_vazao_cil
18	Ajustar válvula pos. 29 (bomba)	Logic.Application.UserVarGlobal.SetValvBombaMHPos29
19	Ajustar válvula de pressão do motor hidráulico	(não implementado)
20	Ajustar válvula proporcional do motor hidráulico	Logic.Application.UserVarGlobal.abValv_vazao_rot

Tabela A5: Variáveis movimentação manual (para uso em a_setpoints) – F64

Função	Variável correspondente CLP
Velocidade manual	Logic.Application.UserVarGlobal.Setpoint_vel_cil
Posição manual	Logic.Application.UserVarGlobal.Setpoint_pos_cil
Força manual	Logic.Application.UserVarGlobal.Setpoint_forca_cil_E1
Rotação manual	Logic.Application.UserVarGlobal.Setpoint_rot
Setpoint válvula limitadora de pressão (pos. 23) [%]	(não implementado)
Setpoint válvula direcional proporcional (pos. 41) [%]	Logic.Application.UserVarGlobal.abValv_vazao_cil
Setpoint válvula da bomba (pos. 29) [%]	Logic.Application.UserVarGlobal.SetValvBombaMHPos29
Setpoint válvula limitadora de pressão (pos. 46) [%]	(não implementado)
Setpoint válvula direcional proporcional de vazão (pos. 40) [%]	Logic.Application.UserVarGlobal.abValv_vazao_rot

Tabela A6: Vetor de setpoints do ensaio a_setpoints – F64

Posição	Função	Variável correspondente CLP
0	Rotação	Logic.Application.UserVarGlobal.Setpoint_rot
1	Número de estágios	Logic.Application.UserVarGlobal.dois_estagios
2	Força do 1º estágio	Logic.Application.UserVarGlobal.Setpoint_forca_cil_E1
3	Comp. de queima do 1º estágio	Logic.Application.UserVarGlobal.Cqueima1
4	Força do 2º estágio	Logic.Application.UserVarGlobal.Setpoint_forca_cil_E2
5	Comp. de queima do 2º estágio	Logic.Application.UserVarGlobal.Cqueima2
6	Força de forjamento	Logic.Application.UserVarGlobal.Setpoint_forca_forj
7	Tempo de forjamento	Logic.Application.UserVarGlobal.T_forjamento

Tabela A7: Vetor de strings para comandos e setpoints

Posição	Função	Valores possíveis
No programa SUPPA32 não será necessário		

Vetores do notificador de medições

Tabela A8: Vetor de estado dos sensores (a_estadoSensores) – U32

Posição	Sensor
0	Sensores OK
1	Comunicação OK
2	Temperatura óleo OK
3	Emergência OK
4	Cilindro OK
5	Motor OK
6	Motor diesel OK
7	Proteção OK
8	UPPA2 OK
9	UPPA3 OK
10	Lâmpadas ligadas
11	Ensaio travou se TRUE

Tabela A9: Vetor de medições (a_medições) – F64

Posição	Sensor
0	Posição
1	Velocidade
2	Força
3	Torque
4	Rotação
5	Temperatura da amostra
6	Pressão da bomba do cilindro hidráulico [bar]
7	Pressão 1 (pos. 42.3) do cilindro hidráulico [bar]
8	Pressão 2 (pos. 42.4) do cilindro hidráulico [bar]
9	Pressão da bomba do motor hidráulico [bar]
10	Pressão 1 (pos. 42.1) do motor hidráulico [bar]
11	Pressão 2 (pos. 42.2) do motor hidráulico [bar]
12	12 = Vazão da bomba do motor hidráulico (pos. 29) [%]
13	13 = Etapa atual FHPP (total=8)

Tabela A10: Vetor contendo texto relacionado a medições (a_stringMedições)

Posição	Função	Valores possíveis
Na UPPA não será necessário		

A5. MENSAGENS DE ERRO

Principais mensagens de erro dentro do cluster de erros

Tabela A11: Mensagens de erro personalizadas (cluster de erro)

Código	Local	Descrição
2000	subvi_lerDados	Arquivo com os parâmetros não encontrado.
2001	subvi_lerDados	Não foi encontrado nenhum arquivo contendo dados de ensaio.