



**INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
Bahia

Campus
Salvador

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA BAHIA
DEPARTAMENTO ACADÊMICO DE TECNOLOGIA MECÂNICA - DATM
CURSO TÉCNICO EM MECÂNICA INTEGRADO

SAMUEL PINHEIRO COUTINHO LOPES

TEMA: IMPLEMENTAÇÃO E ARTICULAÇÃO DA INFORMÁTICA NOS PROCESSOS
DE USINAGEM

SALVADOR - BA

2023

SAMUEL PINHEIRO COUTINHO LOPES

**TEMA: IMPLEMENTAÇÃO E ARTICULAÇÃO DA INFORMÁTICA NOS PROCESSOS
DE USINAGEM**

Atividade de conclusão de curso realizada pelo discente do curso técnico integrado ao ensino médio de mecânica Samuel Pinheiro Coutinho Lopes, sob matrícula no Instituto Federal de Educação, Ciência e Tecnologia da Bahia, contemplada como requisito de inferência para matriz técnica e conclusão da modalidade.

Orientador: Profº Raimundo Jorge Santos Paranhos, Doutor

SALVADOR - BA

2023

AGRADECIMENTOS

A princípio, devoto a minha resoluta gratidão a Deus em virtude das oportunidades que me foram oferecidas - seja no processo de efetuar esse trabalho ou nos acontecimentos que o possibilitaram -, iluminação - dado o esclarecimento e reflexão - e amparo - que revigora e sustenta. Exprimo também minha gratidão à instituição que concedeu estrutura, embasamento e maturação necessária para a realização do trabalho, além dos característicos profissionais que atuaram - e atuam - diligentemente na instituição.

Agradeço aos companheiros que estiveram presentes na instituição compartilhando dos laboratórios e salas por, em essência, terem forjado vínculos de amizade e, em função da postura, responsabilidade e competência, servirem - tal como ainda servem - de referência e norte - em especial para meus companheiros mais próximos, antes discentes do ensino integrado ao médio da instituição e agora técnicos, Jorge de Araujo, Pedro do Vale, José Gabriel e Luis Henrique.

Presto solitudes agradecimentos ao Profº Doutor Raimundo Jorge Santos Paranhos pelo auxílio, paciência, orientação e acolhimento para a realização deste trabalho além de, sobretudo, prover o conhecimento necessário para a sua realização no decorrer das suas aulas, em virtude de sua solicitude, presteza, conhecimento e comprometimento comigo e com os inúmeros alunos que já tiveram a ímpar oportunidade de o ter como docente. A menção da Profª Mestra Mariana Santos Gonçalves é imprescindível nos agradecimentos, tal como fora para a realização do trabalho, dada a matéria de Prática Profissional Articuladora ministrada e planejada por ela que tornou possível as primeiras ideias, esboços e, após devido esforço, versão inicial da aplicação IUsinagem.

Agradeço à minha mãe, Elane França Pinheiro, pelo amor e cuidados incondicionais que tornaram não só este trabalho, como também toda a minha jornada até o presente momento, possível; seu apoio que prestou alicerce fundamental e companheirismo integral.

Agradeço ao meu pai, Antônio Manoel Coutinho Lopes, pelo amor, conselhos, suporte e orientação que, não só possibilitaram toda a minha jornada e o ingresso e continuidade na instituição, como também foram condições indispensáveis para a realização do trabalho.

E, por fim, aos meus companheiros e amigos próximos que prestaram ajuda e amizade de singular importância, em especial para Gabriel de Novaes Vasconcelos e sua família.

RESUMO

A atividade de conclusão do ensino médio possui a proposta de realizar estudos e considerações importantes na área de atuação da usinagem mecânica e possível implementação da informática em diferentes propriedades de sua aplicação nas indústrias, oficinas e outros âmbitos de atuação.

A usinagem caracteriza-se como uma operação que confere à peça dimensionamentos e, em função das necessidades e produto almejado, desempenha variados processos para a obtenção das aferições desejadas - tal como o torneamento geral, fresamento, aplainamento, e outros. Em suma, o processo de usinagem é amplamente utilizado na indústria e desempenha singular importância na obtenção de inúmeras peças presentes na hodiernidade e, consoante ao exposto, possui uma quantidade exorbitante de ferramentas, máquinas e pesquisas a fim de aprimorá-lo e acrescer a sua qualidade e produtividade, com a finalidade de acompanhar o exponencial desenvolvimento da tecnologia.

Ao mesmo passo, a contemporaneidade é marcada por inovações incessantes na área da informática, como a ciência de dados e interconexão digital, amplamente articulada e implementada em outras tecnologias presentes, como é o caso da robótica e automatização de processos - dessa forma configurando o contexto da indústria 4.0.

O enfoque dos estudos, pesquisas e práticas efetuadas neste trabalho caracteriza-se pela compreensão dos parâmetros e condições presentes na usinagem, e seus respectivos critérios e avaliação; obtenção de conhecimento necessário para o desenvolvimento e integração da informática e, sobretudo, as vantagens - tal como possível necessidade - da implementação contínua e acentuada no próprio processo citado a priori.

Sua constatação é realizada por intermédio do desenvolvimento de uma aplicação web voltada para dados de ferramentas e cálculos de usinagem a fim de abstrair tarefas e melhorar a produtividade.

Em suma, não só a possibilidade e vantagens da sua empregabilidade são acentuadas pelo trabalho, como também a necessidade de investir em inovações e estudos a fim de atribuir maior eficácia, eficiência, qualidade e confiabilidade para o processo como um todo.

Palavras chave: Usinagem e informática.

ABSTRACT

The course activity aims to carry out studies and important considerations in the area of mechanical machining and possible implementation of information technology in different properties of its application in industries, workshops and other areas of activity.

Machining is characterized as an mechanical operation that gives dimensions to the part and, depending on the needs and desired product, carries out various processes to obtain the desired measurements - such as general turning, milling, planing, and others. In short, the machining process is widely used in industry and plays a singular importance in obtaining countless parts present in today's world and, accordingly to this, has an exorbitant amount of tools, machines and research in order to improve it and increase its quality and productivity, with the aim of keeping up with the exponential development of technology.

At the same pace, contemporary times are marked by incessant innovations in the area of information technology, such as data science and digital interconnection, widely articulated and implemented in other present technologies, such as robotics and process automation - thus configuring the context of industry 4.0.

The focus of the studies, research and practices performed in this work is characterized by the understanding of the parameters and conditions present in machining, and their respective criteria and evaluation; obtaining the necessary knowledge for the development and integration of information technology and, above all, the advantages - as well as possible necessity - of continuous and accentuated implementation in the process mentioned before.

This is achieved through the development of a web application focused on tool data and machining calculations in order to abstract tasks and improve productivity.

In short, not only are the possibilities and advantages of your employability are proven by this work, but also the need to invest in innovations and studies in order to provide greater effectiveness, efficiency, quality and reliability for the process as a whole.

Keywords: Machining and IT.

LISTA DE FIGURAS

Figura 2.1 - Listagem dos movimentos e tipos de operações em tornos.....	13
Figura 2.2 - Listagem dos movimentos e tipos de operações em furadeiras.....	14
Figura 2.3 - Listagem dos movimentos e tipos de operações em fresadoras.....	15
Figura 2.4 - Processos de furação.....	20
Figura 2.5 - Operação de rebaixamento.....	21
Figura 2.6 - Alargamento cilíndrico de desbaste e de acabamento, respectivamente..	21
Figura 2.7 - Processo de brochamento.....	22
Figura 2.8 - Roscamento externos com fresa de perfil múltiplo e único, respectivamente.....	23
Figura 2.9 - Direção e movimentos no torneamento.....	26
Figura 2.10 - Direção e movimentos na furação.....	27
Figura 2.11 Direção e movimento no fresamento.....	27
Figura 2.11 - Fórmula de cálculo da velocidade de corte.....	28
Figura 2.12 - Fórmula de cálculo da velocidade de corte.....	28
Figura 2.6 - Fórmula para a obtenção do percurso de avanço.....	29
Figura 2.7 - Fórmula para a obtenção do tempo de corte.....	29
Figura 2.8 - Indicação das variáveis de percurso de avanço e tempo de corte.....	30
Figura 2.9 - Representação das diferentes parcelas do tempo de fabricação por peça dada uma velocidade de corte adotada.....	31
Figura 2.10 - Cunha de corte da ferramenta.....	32
Figura 2.11 - Variação do ângulo da cunha, em função da dureza do material.....	34
Figura 2.12 - Sistema ciber físico da informação.....	37
Figura 3.1 - Template do componente toolbar.....	47
Figura 3.2 - Interface do objeto pastilha.....	48
Figura 3.3 - Inicialização dos dados pertencentes ao catálogo e atribuição do paginador à variável.....	50
Figura 3.4 - Interfaces geometry, process e parameters declaradas.....	52
Figura 3.5 - Atribuição do valor de largura da tela à propriedade de alinhamento da estrutura.....	53
Figura 3.6 - Declaração e inicialização do objeto process_config.....	54
Figura 3.7 - Função addNewGeometry() declarada.....	56
Figura 3.8 - Função resetGeometry() declarada.....	57
Figura 3.9 - Função removeGeometry() declarada.....	57
Figura 3.10 - Função checkGeometries() declarada.....	58
Figura 3.11 - Verificação da falha na validação e atribuição do erro às mensagens....	59
Figura 3.12 - Função checkDiameters() declarada.....	60

Figura 3.13 - Função insert Filter() declarada.....	62
Figura 3.14 - Função insert Calculator() declarada.....	63
Figura 3.15 - Função insertFiltrateThenCalculate() declarada.....	63
Figura 3.16 - Função fillGeometriesInsertsAndParameters() declarada.....	64
Figura 3.17 - Função unfillGeomtriesInsertsAndParameters() declarada.....	65
Figura 3.18 - Função selectionChange() declarada.....	65
Figura 3.19 - Variável tabData declarada e inicializada ao mesmo passo que a função tabDataCreator() é declarada.....	66
Figura 3.20 - Atributos e eventos designados à estrutura mat-stepper.....	67
Figura 3.21 - Template do componente app-tabs.....	69
Figura 3.22 - Início do Template do componente app-panel com sua primeira iteração	71
Figura 4.1 - Disposição das pastilhas no componente de catálogo para o usuário.....	73
Figura 4.2 - Parâmetros de usinagem e especificações das pastilha.....	73

LISTA DE SIGLAS E ABREVIATURAS

Ap - Profundidade de corte

CNC - Comando numérico computadorizado

CSS - Cascading Style Sheets

DOM - Document Object Model

Fn - Avanço por rotação

HTML - Hyper Text Markup Language

JS - JavaScript

La - Percurso

Px - Pixels

Rpm - rotações por minuto

Tc - Tempo de corte

TS - TypeScript

URL - Uniform Resource Locator

Vc - Velocidade de corte

SUMÁRIO

1. INTRODUÇÃO.....	10
1.1 Exposição do Tema.....	10
1.2 Justificativa.....	11
2. REVISÃO BIBLIOGRÁFICA.....	12
2.1 Operações de Usinagem.....	12
2.2 Grandezas físicas na usinagem.....	23
2.3 Geometria das ferramentas de corte.....	32
2.4 Desenvolvimento web.....	34
2.4.1 Definição.....	34
2.4.2 Importância.....	36
2.4.3 JavaScript.....	38
2.5 Angular.....	39
2.5.1 Origem.....	39
2.5.2 Estrutura.....	40
3. TECNOLOGIAS E MÉTODOS.....	43
3.1 Projeto.....	43
3.1.2 Objetivo.....	43
3.2 Componentes.....	45
3.2.1 Toolbar.....	45
3.2.2 Main.....	48
3.2.3 Catalog.....	49
3.2.3.1 Estrutura.....	49
3.2.4 Stepper.....	51
3.2.4.1 Responsividade.....	52
3.2.4.2 Dados Relevantes.....	53
3.2.4.3 Funções de Geometria.....	55
3.2.4.4 Funções de parâmetros e pastilhas.....	61
3.2.4.5 Convergir dados.....	66
3.2.4.6 Estrutura.....	66
3.2.5 Tabs:.....	68
3.2.5.1 Estrutura.....	69
3.2.6 Panel:.....	70
3.2.6.1 Estrutura.....	70
4. RESULTADOS.....	72
4.1 Interface.....	72

4.2 Aplicabilidade.....	73
4.3 Aplicação.....	74
5. DISCUSSÃO DOS RESULTADOS.....	74
6. CONCLUSÃO.....	74
6.1 Sugestões.....	75
7. REFERÊNCIAS BIBLIOGRÁFICAS.....	77

1. INTRODUÇÃO

1.1 Exposição do Tema

A usinagem, em sua essência, configura-se como um processo de remoção de material, caracterizado pela redução da massa com o intuito de modificar a geometria em conformidade com um produto preconcebido e anelado. Amplamente adotada na esfera industrial, essa prática ostenta uma importância primordial na origem e aplicação de inúmeras peças e componentes no domínio da técnica e tecnologia mecânica.

Ao longo de qualquer procedimento de processamento de material, tornam-se prementes as características intrínsecas do produto, as quais devem ser meticulosamente estudadas e abordadas de acordo com os critérios pertinentes à função por ele desempenhada. Estas características abrangem a forma, dimensão e propriedades do material, sendo a configuração da forma e dimensões obtidas por meio de procedimentos como a própria usinagem e a conformação mecânica.

De acordo com o supracitado, em virtude das particularidades do material, bem como de seu estado, emergem critérios adicionais no que concerne à seleção das ferramentas que submeterão a matéria-prima ao processo. O desbaste, uma fase crucial, demanda ferramentas cujas características e formas sejam distintas, a fim de alcançar não apenas o êxito, mas também uma eficiência ampliada durante o curso do processo. Materiais dotados de propriedades diversas exigirão agentes de corte com atributos e conformações diferenciados para assegurar o sucesso e a eficácia do procedimento.

Alinhado a este contexto, ressalta-se a imperatividade de ajustar as ferramentas à fase específica do processo, considerando as dimensões desejadas, a forma almejada, as capacidades operacionais da máquina, a produtividade desejada, entre outros fatores relevantes.

Por conseguinte, torna-se imprescindível um preparo e organização metódicos, juntamente com um controle constante do procedimento, além de avaliações regulares no que concerne à sua aplicabilidade atual, futura e constante, dadas as demandas dinâmicas do contexto industrial.

1.2 Justificativa

A presente atividade de conclusão de curso (ACC) se propõe a investigar a aplicabilidade dos conhecimentos adquiridos ao longo do curso técnico em mecânica industrial, com ênfase nas disciplinas de informática, lógica de programação, processos de fabricação, elementos de máquina e CNC, no contexto da usinagem e processos de fabricação mecânica. Esta justificativa fundamenta-se na relevância intrínseca desses conhecimentos para a otimização e eficiência dos processos industriais, considerando a natureza dinâmica e tecnológica do ambiente de produção atual.

A incorporação de conceitos de informática e lógica de programação se apresenta como um diferencial crucial, uma vez que possibilita a automação e controle precisos dos sistemas de usinagem, contribuindo para a melhoria da qualidade e a redução de desperdícios. A compreensão dos processos de fabricação, por sua vez, revela-se essencial para a seleção adequada de ferramentas e estratégias, como explicitado na abordagem anterior, onde a usinagem é destacada como um processo estratégico na alteração de geometrias de materiais.

Os conhecimentos em elementos de máquina, intrínsecos à formação em mecânica industrial, convergem diretamente para a compreensão da interação entre as ferramentas de corte e a matéria-prima, desempenhando um papel crucial na escolha apropriada de instrumentos para diferentes materiais e aplicações. A familiaridade com CNC (Controle Numérico Computadorizado), por sua vez, representa uma competência tecnológica essencial na programação e operação de máquinas, alinhando-se com a demanda por processos produtivos mais ágeis e precisos.

Diante desse panorama, o presente estudo visa não apenas sintetizar e aplicar os conhecimentos adquiridos em disciplinas específicas, mas também compreender de maneira integrada como tais saberes podem potencializar a eficiência e a inovação nos processos de usinagem. A justificativa centraliza-se na premência de atualização constante e na necessidade de capacitação técnica que transcenda os limites tradicionais do curso, abraçando a interdisciplinaridade como um alicerce para a excelência no contexto da mecânica industrial.

2. REVISÃO BIBLIOGRÁFICA

2.1 Operações de Usinagem

Há inúmeros métodos de processamento de materiais e um número equivalentemente avantajado de vertentes no que tange suas respectivas aplicações, de tal forma que ela se dê em virtude da conveniência e necessidades oriundas do projeto estabelecido.

As ferramentas de geometria são utilizadas de acordo com os elementos dispostos, de tal forma que há a divisão das operações em ramos caracterizados como desbaste e acabamento.

Nas operações de desbaste, a finalidade principal é garantir que haja a remoção de material - havendo, de tal forma, a irretorquível precedência da potência da máquina-ferramenta - contudo, na operação de acabamento, o sobremetal remanescente - oriundo da operação anterior - é removido tendo como parâmetro principal de controle e finalidade a obtenção da qualidade em virtude das tolerâncias esperadas a priori. MACHADO, Alisson (2015).

Entre os fatores de influência do processo, há também a ordem das tensões às quais as etapas da produção submeterão os componentes envolvidos, sendo esse um fator de essencial importância aquém à escolha de ferramentas, seu respectivo material e parâmetros adotados na máquina ferramenta - uma vez que o estado da superfície do produto origina-se de todas as composições presentes na usinagem.

Referentemente às condições da superfície final, ressalta-se que suas características presentes no produto decorrem do processo dadas tensões impostas e deformações oriundas da etapa de fabricação vigente - de tal forma que haja discernimento no resultado do processo dada a interferência das deformações, gerações de calor, vibrações e reações químicas - dessa forma, todos os elementos dispostos alteram as condições da superfície finalizada. MACHADO, Alisson (2015).

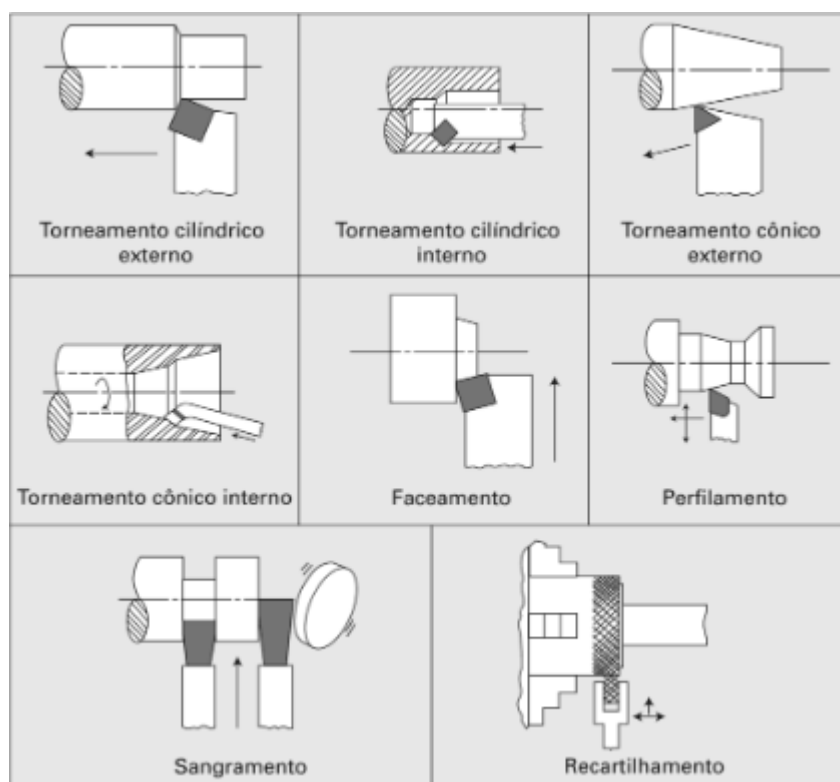
Dessa forma, salienta-se que a preocupação no que tange a potência da máquina ferramenta, ferramentas de geometria adotadas e tipo de operação existe e faz-se necessária ao implementar módulos de providência desses elementos - tal como emuladores de desbaste ou bancos de dados, como é o caso das tecnologias CNC -, a fim de controlar todas as variáveis que resultarão em mudanças nas propriedades e qualidade do produto almejado.

Evidenciam-se também a definição da própria máquina-ferramenta, uma vez que distinguem-se entre si em seu funcionamento e utilidade. São exemplos de máquinas-ferramenta utilizadas no âmbito de ofício da usinagem: Torno - horizontal, vertical ou universal -, fresadora - como a de marca Fellows, por exemplo - e furadeiras. As operações de desbaste também possuem configurações de disposição das ferramentas e geometrias distinguíveis, de tal forma que expressam-se diferentemente.

A disposição configurativa citada baseia-se no movimento de avanço adotado tal como a denominação dada à operação.

A figura abaixo em conjunto com uma listagem, adaptados do acervo constatado, exprimem tais nomenclaturas e expressões no caso da máquina-ferramenta torno mecânico:

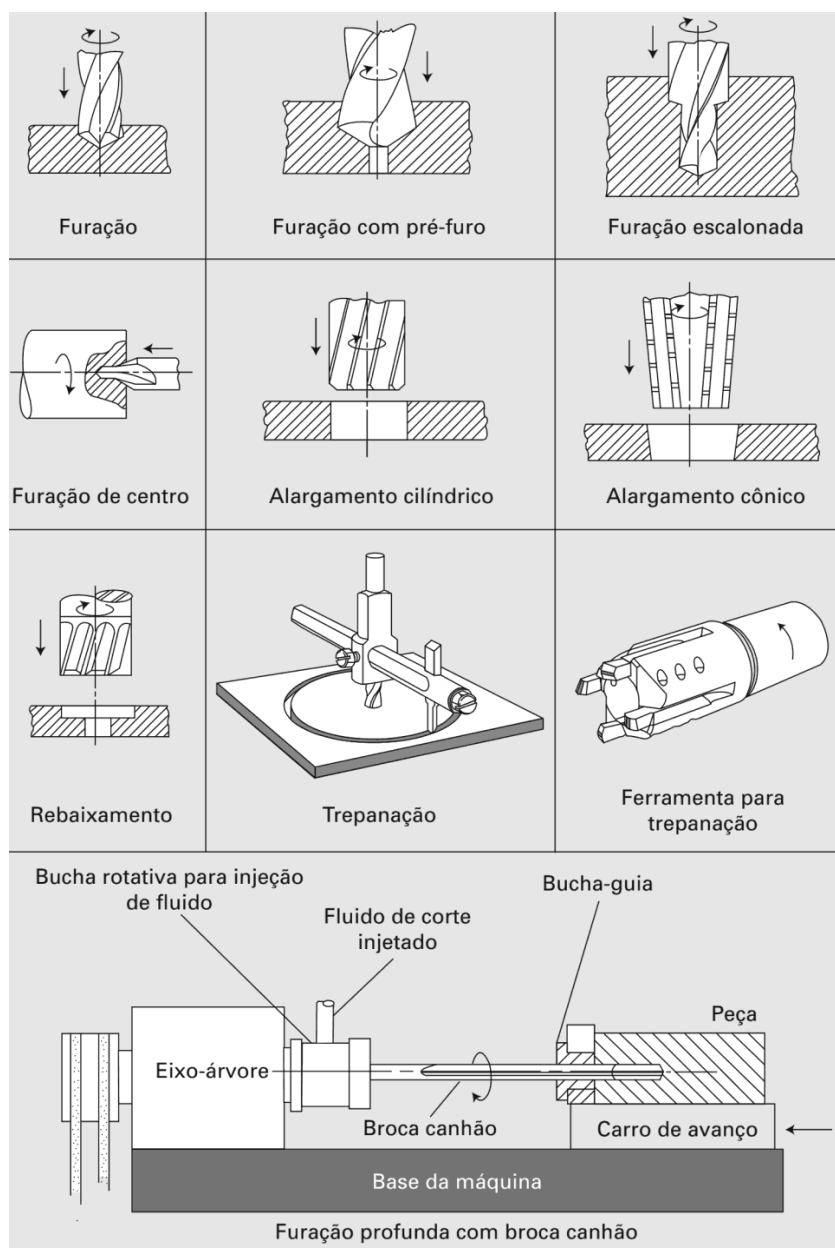
Figura 2.1 - Listagem dos movimentos e tipos de operações em tornos



Fonte: MACHADO, Alisson (2015).

O mesmo vale para as disponíveis na atuação de outras máquinas, tal como furadeiras e fresadoras, cujas operações estão predispostas em:

Figura 2.2 - Listagem dos movimentos e tipos de operações em furadeiras

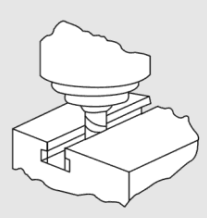
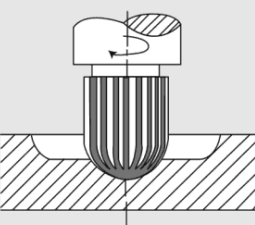


Fonte: MACHADO, Alisson (2015).

Para o caso das operações realizadas nas furadeiras - seja com movimento de avanço manual ou automático - na condição de alargamento para qualidade superior no desbaste ou furação inicial.

No caso das operações de fresamento expressam-se dada a seguinte relação de figura, nomenclatura e movimento característico da operação:

Figura 2.3 - Listagem dos movimentos e tipos de operações em fresadoras

 <p>Fresamento tangencial (discordante)</p>	 <p>Fresamento tangencial (concordante)</p>	 <p>Fresamento tangencial de canais ou de perfis</p>
 <p>Fresamento tangencial de canais ou de perfis</p>	 <p>Fresamento de topo</p>	 <p>Fresamento frontal</p>
 <p>Fresamento frontal</p>	 <p>Fresamento com fresa de topo esférica</p>	 <p>Fresamento de cavidades</p>

Fonte: MACHADO, Alisson (2015).

Subentende-se como adequação ao tipo de operação a escolha da ferramenta tal como técnica de operação, ao mesmo passo que entende-se como fatores delimitadores do tipo de

operação a geometria do produto almejado e suas propriedades dimensionais - expressão de furos, rasgos, formato, e outros.

Tocante às máquinas-ferramenta empregadas no âmbito de atuação, evidenciam-se certas características comuns evidenciadas em sua natureza por intermédio de definições inatas, sendo tais definições distinguíveis no que abrange cada uma delas.

No processo de torneamento, por exemplo, há o deslocamento da ferramenta no decorrer de uma trajetória em concorrência com a rotação da máquina, assim ocasionando desbastes em virtude dos movimentos e direções adotados.

A obtenção de superfícies de revolução por intermédio de uma ou mais ferramentas monocortantes é obtido por intermédio de processos mecânicos de usinagem, sendo o torneamento um deles. No caso da máquina-ferramenta intrínseca ao processo citado, há o giro da peça envolta do eixo de rotação da máquina e o deslocamento simultâneo da ferramenta segundo uma trajetória coplanar com o dito eixo - a fim de obter as superfícies almejadas. COSTA, Éder; SANTOS, Denis (2006).

Também é possível observar a disposição dos tipos de torneamento em virtude da trajetória do deslocamento da ferramenta e sua relação dado o eixo principal de rotação da máquina, uma vez que cada categoria de operação resultará em métodos distintos de obter formas e perfis almejados - sendo, de tal forma, imprescindível determinar qual será o torneamento utilizado na obtenção do produto.

No que é denotado sobre o processo torneamento retilíneo, evidenciam-se as seguintes definições presentes no acervo da autoria de Prof. Éder Silva Costa e Denis Júnio Santos: Processos de Usinagem; do Centro Federal de Educação Tecnológica de Minas Gerais:

Torneamento cilíndrico - Processo de torneamento no qual a ferramenta se desloca segundo uma trajetória paralela ao eixo principal de rotação da máquina.

Torneamento cônico - Processo de torneamento no qual a ferramenta se desloca segundo uma trajetória retilínea, inclinada em relação ao eixo principal de rotação da máquina.

Torneamento radial - Processo de torneamento no qual a ferramenta se desloca segundo uma trajetória retilínea, perpendicular ao eixo principal de rotação da máquina.

Perfilamento - processo de torneamento no qual a ferramenta se desloca segundo uma trajetória retilínea radial ou axial.

As categorias citadas anteriormente referem-se aos processos de torneamento com deslocamento de ferramenta submetido à trajetória retilínea, contudo, há a aplicação da técnica oriunda da trajetória de natureza curvilínea com finalidade específica.

Tais propósitos são expressos na forma com que o desbaste ocorre - em função de parâmetros definidos e finalidade do processo -, descritos dada a propriedade do processo, que pode ser de acabamento ou desbaste.

As operações de torneamento podem ser classificadas em função de sua finalidade de tal forma que dispõem-se em desbaste e acabamento - dadas suas denominações habituais, torneamento de desbaste e torneamento de acabamento. As operações de desbaste compreendem-se como operações predecessoras das de acabamento, de tal forma que o acabamento é destinado a obtenção das dimensões finais ao mesmo passo em que o desbaste visa obter as dimensões aproximadas. Categoricamente, as operações de desbaste tornam-se mais presentes dada a discrepância presente entre as dimensões iniciais do material e as almejadas pelo produto final. FERRARESI, Dino (2018).

Dito isso, separa-se na etapa do processo de fabricação a seção da produção direcionada ao propósito de remoção com maior ordem de especificidade em função das cotas adotadas no projeto - tal como as suas devidas tolerâncias - e a seção que a antecede, sendo essa a principal responsável por remover porções majoritárias do material presente na matéria antecessora ao produto - dessa forma, a etapa de acabamento torna-se de maior rigor e critério na mesma proporção em que as tolerâncias se tornam mais específicas.

Ainda no que tange às máquinas-ferramentas, o processo de usinagem necessitará da composição discernível presente nas fresadoras quando a orientação da remoção do material se provar inconcebível ou ineficiente na atuação do torno - como são muitos os casos, dada a distinção presente na configuração dos movimentos e direções adotadas pelas duas máquinas -, sendo assim, sua caracterização e definição também dispõem-se diferentemente.

Fresamento: Caracteriza-se como processo que destina-se à obtenção de superfícies quaisquer dada a atuação de ferramentas multicortantes. No decorrer do processo citado, há o giro da ferramenta e a peça - ou ferramenta - desloca-se dada uma trajetória determinada. FERRARESI, Dino (2018).

Não obstante, apesar de sua importância, os produtos originados do processo de fresamento não são oriundos somente de suas capacidades, podendo também ser obtida por intermédio de outros - como a plaina, por exemplo.

Assim como no torneamento, também há a aparição de desbaste e acabamento dentro dos processos de aplainamento que, conforme citado a priori, é capaz de obter usinagens na forma de perfis, rasgos, guias e outros - tal como são obteníveis através da fresadora -, contudo, com certa distinção no que tange alguns critérios.

O processo de aplainamento destina-se à obtenção de superfícies regradadas cujas dimensões são obtidas dado o movimento retilíneo alternativa da ferramenta ou peça, sendo esse processo disposto em forma horizontal ou vertical. Suas operações intrínsecas possuem distinções de finalidade similares às do processo de torneamento, de tal forma que discernem-se em desbaste e acabamento - aplainamento de desbaste e aplainamento de acabamento. COSTA, Éder; SANTOS, Denis (2006).

No que abrange as distinções mencionadas, é possível contemplar seu discernimento no que se refere a sua definição e utilidades:

As ferramentas multicortantes - fresas - utilizadas no fresamento produzem superfícies planas, de tal forma que - em critério de função e produtos obteníveis - possa competir com as operações de aplainamento, não obstante, não são todas as ocasiões em que um método de fabricação exerce vantagem coincidente sobre o uso. É possível executar operações de alargamento, mandrilamento e furação dada a máquina-ferramenta fresadora adequada, sendo algumas capazes de reproduzir furos com altíssima qualidade no que se refere a sua ordem de tolerâncias geométricas e disposição espacial - na forma de tolerâncias na casta de 0,0025 mm a 0,0050 mm. COELHO, Reginaldo; SILVA, Eraldo (2018).

E, a posteriori, no que toca sua relação de custo benefício e aplicabilidade:

Em função de determinada superfícies planas, é possível que plainas limadoras empenhem com maior eficiência no que se refere à remoção e velocidade de desbaste quando equiparadas às operações de fresamento convencionais, contudo, no que abrange os centros de usinagem atuadoras com tecnologias CNC, a eficiência e produtividade das fresadoras implementadas excede em categoria e aplicação - apesar do acréscimo no custo inicial e de manutenção. COELHO, Reginaldo; SILVA, Eraldo (2018).

A tecnologia CNC citada fundamenta-se no comando numérico computadorizado, de alta proeminência na contemporaneidade como uma das implementações da informática no processo - que se torna mais produtivo e seguro dado um manejo apropriado das informações e nível acrescido em virtude da ordem de especificidade dos programas, que além de atuarem mais rápido podem originar produtos com menos discernimentos no acabamento.

A ausência da implementação de tornos computadorizados em linhagens de usinagem no âmbito industrial pode acarretar em perdas no que se refere aos parâmetros de qualidade e competitividade vigentes no cenário, de tal forma que haverá perda de tempo, desgaste físico e mental dos funcionários, diminuição acentuada do rendimento dos processos presentes e, por conseguinte, menores chances de competitividade frente às oposições que implementaram as linhas com a tecnologia mencionada. Evidenciam-se também problemas habituais da indústria que influenciam diretamente nos parâmetros adotados, assim comprometendo o rendimento final resultante das operações. SILVA, Sandro; FILHO, Ramílio (2022).

Frisa-se a sua utilidade e possibilidade de implementação em inúmeras modalidades de atuação das máquinas-ferramenta, assim acentuando não tão somente quesitos de vantagens no que se refere à otimização do processo, quanto também abrangência na aplicação.

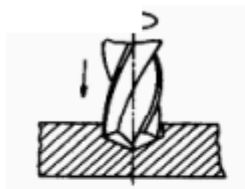
Consoante às máquinas-ferramenta e obtenção de furos em corpos submetidos aos processos de usinagem, explicita-se a possibilidade de realizar o processo por intermédio do torno ou, sistematicamente, por intermédio de máquinas destinadas prioritariamente a essa função, tal como é o caso das furadeiras - como as de bancada, portáteis e radiais - que exercem o propósito dados critérios de avanço e diâmetro relativos às dimensões desejadas.

Nesse sentido, o processo de usinagem tem pretensão fundamental em originar um furo - habitualmente cilíndrico - e que, dada uma determinada superfície, utiliza de ferramentas multicortantes com giro simultâneo e deslocamento coincidente, retilíneo ou paralelo ao eixo principal da máquina em função da peça. O processo de furação secciona-se em subdivisões discerníveis, como o processo de trepanação - volume do furo final reduzido à cavaco com parte do volume material compreendido -, furação de centros - destinada a obtenção de furos de centro -, furação escalonada - obtenção de múltiplos furos de modo simultâneo - e escareamento - abertura em uma peça pré-furada. FERRARESI, Dino (2018).

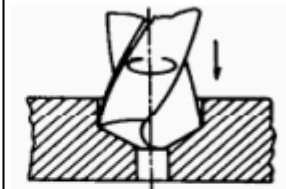
Suas respectivas denotações ilustrativas podem ser dispostas graficamente dado o seguinte acervo, subentendendo furação com pré-furação como escareamento:

Figura 2.4 - Processos de furação

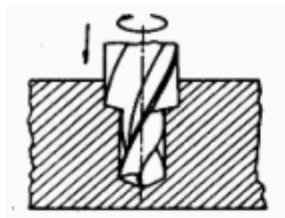
a) Furação em cheio



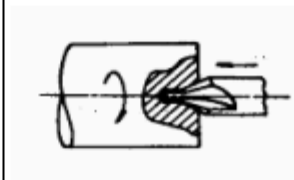
b) Furação com pré-furação



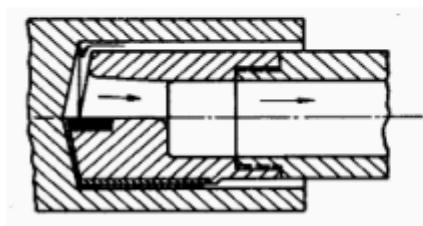
c) Furação escalonada



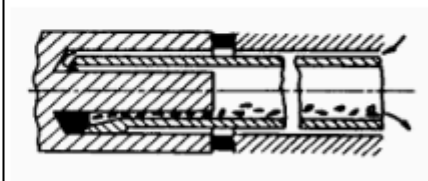
d) Furação de centros



e) Furação profunda em cheio



f) Trepanação



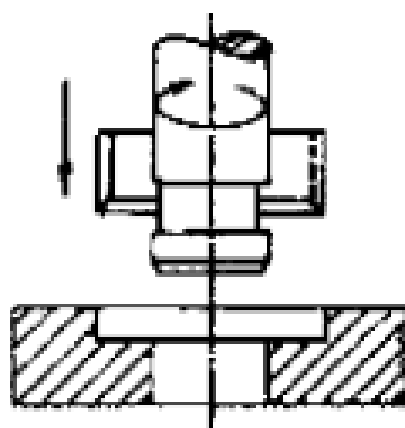
Fonte: COSTA, Éder; SANTOS, Denis (2006).

Em concordância com os expostos, compreende-se que as funções obtidas por intermédio dos dispositivos criados originam-se das necessidades contempladas pelas ramificações dos modelos de produção vigente - de tal forma que em um período de forte desenvolvimento industrial implique no acréscimo e evolução das tecnologias pertencentes ao setor, ao mesmo passo em que há a implementação de tecnologias e inovações nas técnicas aplicadas.

Dito isso, os processos de usinagem acompanham suas necessidades, assim dispondo de diferentes processos acompanhados de técnicas e empregabilidades distintas a fim de cumprir determinadas funções, Processos de acabamento, revolução com auxílio, alterações em superfícies trabalhadas anteriormente, e et cetera, são exemplos considerados.

A fim de obter a modificação da forma em extremidades de furos se dá o processo de rebaixamento, onde ocorre o deslocamento da ferramenta ou peça com giro coincidente segundo uma trajetória retilínea. FERRARESI, Dino (2018).

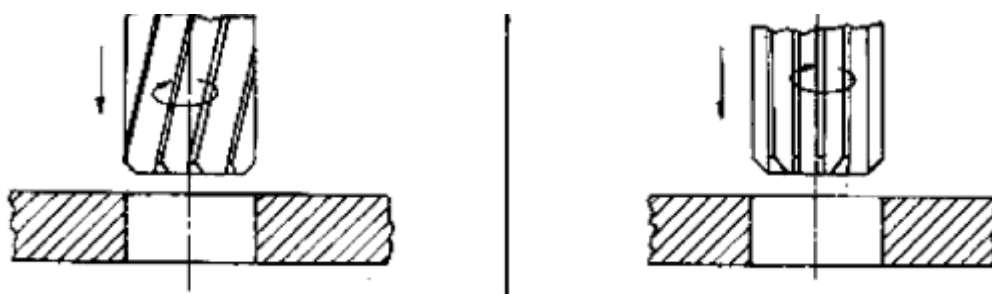
Figura 2.5 - Operação de rebaixamento



Fonte: FERRARESI, Dino (2018)

Com o propósito de realizar acabamento ou desbaste de furos cilíndricos ou cônicos através de ferramentas multicortantes - na abrangente maioria das ocasiões - faz-se uso do processo de alargamento onde ocorre deslocamento e giro da ferramenta ou peça em função de uma trajetória retilínea. COSTA, Éder; SANTOS, Denis (2006).

Figura 2.6 - Alargamento cilíndrico de desbaste e de acabamento, respectivamente



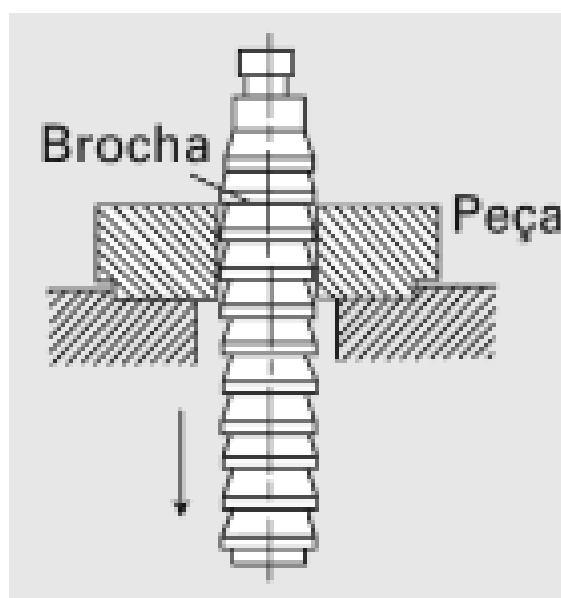
Fonte: FERRARESI, Dino (2018).

Dada a finalidade de adquirir superfícies de revolução em função do auxílio de uma ou múltiplas ferramentas de barra, é possível utilizar o processo mecânico de mandrilamento que

fundamenta-se no giro da ferramenta segundo uma trajetória determinada a priori. Sua disposição é compreendida por intermédio das subdivisões: mandrilamento cônico - revolução de superfície cônica de revolução com eixo coincidente da ferramenta -; mandrilamento de superfícies especiais - superfície de revolução com eixo periférico à ferramenta -; mandrilamento cilíndrico - superfície cilíndrica de revolução com eixo periférico da ferramenta - e mandrilamento radial - ferramenta de configuração geométrica plana com perpendicularidade em função do eixo. Ressalta-se também a divisão dados os critérios de remoção, subentendida em mandrilamento de desbaste e de acabamento. COSTA, Éder; SILVA, Denis (2006).

Em virtude da necessidade de se produzir furos com formas discernentes das cilíndricas habituais é possível utilizar uma ferramenta denominada de brocha - cujo processo no qual está inserida nomeia-se brochamento - onde, em virtude de passagens sucessivas de dentes, mudanças de forma na direção do perfil almejado são realizadas. O processo dispõe de elevada qualidade dimensional e geométrica e, dada a sua projeção configurativa inerente, possibilita a atuação na obtenção de formas externas. MACHADO, Álisson (2004).

Figura 2.7 - Processo de brochamento



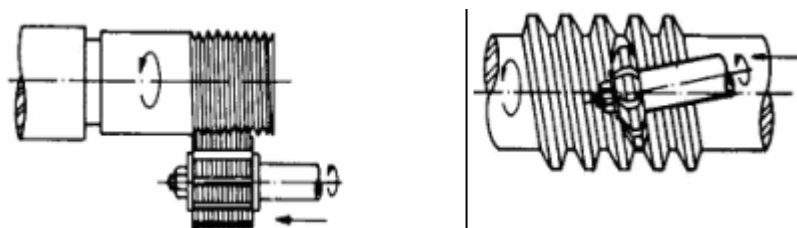
Fonte: MACHADO, Álisson (2004).

Também evidenciam-se processos cuja finalidade é fomentar ou originar características configurativas distintas, como é o caso das rosas e os processos de rosqueamento de uma

determinada superfície. A operação de abertura é extremamente diversificada e pode ser obtida através de atuação operária manual ou através das próprias máquinas-ferramentas, variando com base nas dimensões da rosca e planejamento no que se refere à produtividade e rendimento do processo. Roscas externas podem ser obtidas por cossinetes, ao mesmo tempo que também são obtíveis através de ferramentas de perfil único ou múltiplo, não obstante, roscas internas são, habitualmente, produzidas a partir de machos e ferramentas de perfil múltiplo ou único. MACHADO, Álisson (2004).

O processo de roscamento destina-se à finalidade de conseguir filetes através da abertura de sulcos helicoidais com passo uniforme. Ocorre o giro da ferramenta ou peça com deslocamento simultâneo dada uma trajetória retilínea, paralela ou inclinada e, em virtude do plano de trabalho, pode ser interno ou externo. FERRARESI, Dino (2018).

Figura 2.8 - Roscamento externos com fresa de perfil múltiplo e único, respectivamente



Fonte: COSTA, Éder; SANTOS, Júnio (2006).

2.2 Grandezas físicas na usinagem

A usinagem possui distintas grandezas intrínsecas que devem ser reguladas e estudadas a fim de não só acrescer o grau de precisão e segurança no processo de fabricação como também assegurar maior eficiência no decorrer da produção.

Consoante ao exposto, um bom controle dessas dimensões é imprescindível quando se quer utilizar da tecnologia de informática para aprimorar suas propriedades relativas à qualidade e confiabilidade - assim fazendo-se necessário prover de boa gestão dessas informações e sua respectiva interpretação.

Nos processos de fabricação, geralmente, haverá mais de um método que poderá ser empregado para fabricar um componente. A seleção de um método particular sobre outros vai depender de um grande número de fatores. Além disso, o produto final, geralmente, é o

resultado de muitos processos diferentes. Na seleção do processo, os seguintes fatores são evidenciados no acervo Teoria da usinagem dos materiais de Álisson Rocha Machado:

- Tipo do material e suas propriedades;
- Propriedades finais desejadas;
- Tamanho, forma e complexidade do componente;
- Tolerâncias e acabamento superficial requeridos;
- Processo subsequente envolvido;
- Projeto e custo de ferramental; efeito do material na vida da ferramenta ou matriz;
- Sucata gerada e seu valor;
- Disponibilidade do equipamento e experiências operacionais;
- “Lead time” necessário para iniciar a produção;
- Número de partes requeridas e taxa de produção desejada;
- Custo total de processamento.

As propriedades do material disposto irão inferir diretamente no tipo de ferramenta de desbaste escolhida - N, K, - as propriedades finais em função do estado inicial do material alteraram múltiplas tomadas de decisão referentes ao processo - podendo esse estar sob condições de usinagem distintas -, grau de complexidade do material, tal como a disposição de gasto do projeto e as tolerâncias requeridas - sendo esses elementos primordiais para avaliar o custo do ferramental e sua aplicação no decorrer do processo.

Inerente à composição e atuação das máquinas-ferramentas, evidencia-se um princípio comum para a obtenção da superfície almejada: movimento relativo apropriado entre a ferramenta e a peça. Nesse quesito, torna-se imprescindível para o estudo do ofício a definição de suas grandezas físicas preponderantes e reincidentes no processo, tais como o movimento, direção e velocidade. MACHADO, Álisson (2015).

Dessa forma, a partir dos conhecimentos dispostos mencionados, conclui-se que os movimentos presentes nas operações influenciam diretamente no resultado do processo de usinagem, sendo um de inúmeros fatores de importância significativa uma vez que influencia diretamente na direção dos movimentos, a formação do cavaco e percursos da ferramenta na peça.

No que tange a saída do cavaco e influência direta do movimento em função de sua retirada, movimentos de corte, avanço e movimentos efetivos dispõem-se entre os quais a causam ao mesmo passo que movimentos de aproximação, ajuste e correção não exercem

papel em sua formação - sendo assim caracterizados em dois grupos distintos, que categorizam movimentos com a distinção no que diz respeito a ausência de desbaste e a finalidade de sua atuação.

Saber a função do movimento requerido tal como as implicações de sua atuação e natureza inata provam-se importantes no processo de planejamento da operação, de tal forma que evidenciá-los e distingui-los manifesta-se como preceito básico nos conceitos que situam a usinagem, sendo tais aferições, no que engloba os movimentos que causam saída direta, dispostas no material Teoria da usinagem dos Materiais de Álisson Rocha Machado:

Movimento de corte: realizado entre a peça e a aresta de corte, o qual, na ausência de movimento de avanço, produz somente uma única retirada de cavaco.

Movimento de avanço: realizado entre a peça e a aresta de corte, o qual, com o movimento de corte, provoca a retirada contínua de cavaco.

Movimento efetivo: resultante dos movimentos de corte e avanço, realizados ao mesmo tempo.

Ao mesmo passo que, para as que não acarretam em saída direta, há a seguinte disposição de mesma autoria e obra:

Movimento de aproximação: realizado entre a peça e a aresta de corte, por meio do qual ambas se aproximam antes da usinagem.

Movimento de ajuste: realizado entre a peça e a aresta de corte para determinar a espessura do material a ser retirado.

Movimento de recuo: realizado entre a peça e a aresta de corte com o qual a ferramenta, após a usinagem, é afastada da peça.

Consoante ao exposto, consolida-se em sua caracterização uma questão inerente de dependência no que refere-se à origem dos movimentos - do que se resulta, e origina.

Distinguem-se duas definições distintas de movimento discerníveis por intermédio dos seguintes critérios: causa direta da saída de cavaco e ausência de influência direta na sua formação. Consoante ao exposto, o movimento efetivo de corte - responsável direto pela saída de cavaco - origina-se do movimento de corte e avanço que, categoricamente, distinguem-se em definição. FERRARESI, Dino (2018).

Entre as classificações estabelecidas, há a distinção na natureza dos movimentos de avanço e dos de corte, de tal forma que o movimento de avanço ocorre em função da posição

da peça e da ferramenta que origina, quando operado em conjunto com o movimento de corte, a aparição ininterrupta ou cíclica do cavaco no decorrer de suas revoluções.

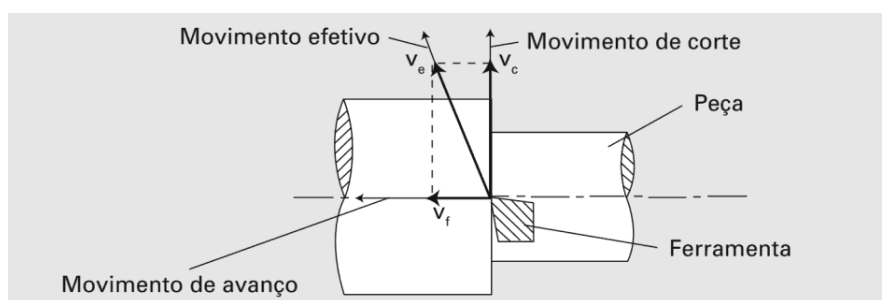
Movimento de avanço ocorre entre a peça e a ferramenta de tal forma que, em conjunto com o movimento de corte, resulte no levantamento cíclico e produção repetida do cavaco. FERRARESI, Dino (2018).

Em consonância com o aferido, dispõem-se os tipos de movimento em maior grau de especificidade em função das direções de corte como embasamento intrínseco ao ofício como composição inteligível básica na operação.

Aferem-se como direções do movimento a direção de corte, de avanço e direção efetiva de corte, dispostas dado o movimento relacionado - de tal forma que: a direção de avanço se dá pela direção imediata do movimento de avanço; a direção de corte se dá pela direção imediata do movimento de corte e a direção efetiva de corte se dá pela direção imediata do movimento efetivo de corte.

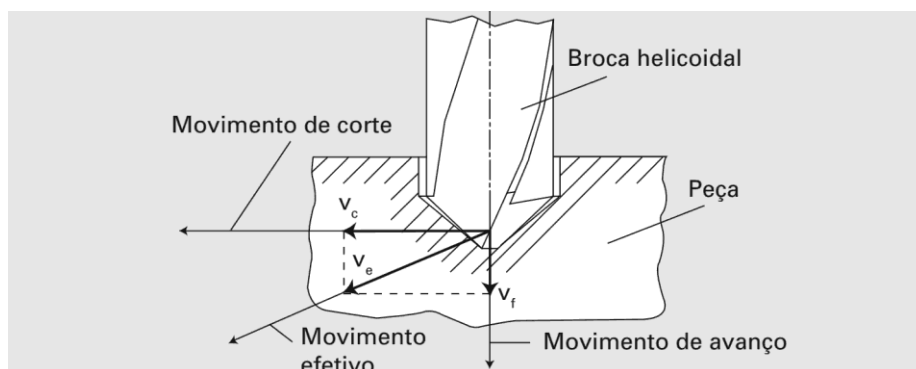
Suas caracterizações coincidem com a máquina-ferramenta adotada, sendo esse um mérito crucial a fim de dispor as definições com critério de identidade - aquém à aplicabilidade do preceito e noção teórica -, de tal forma que configure-se, ilustradamente, a relação nas seguintes expressões:

Figura 2.9 - Direção e movimentos no torneamento



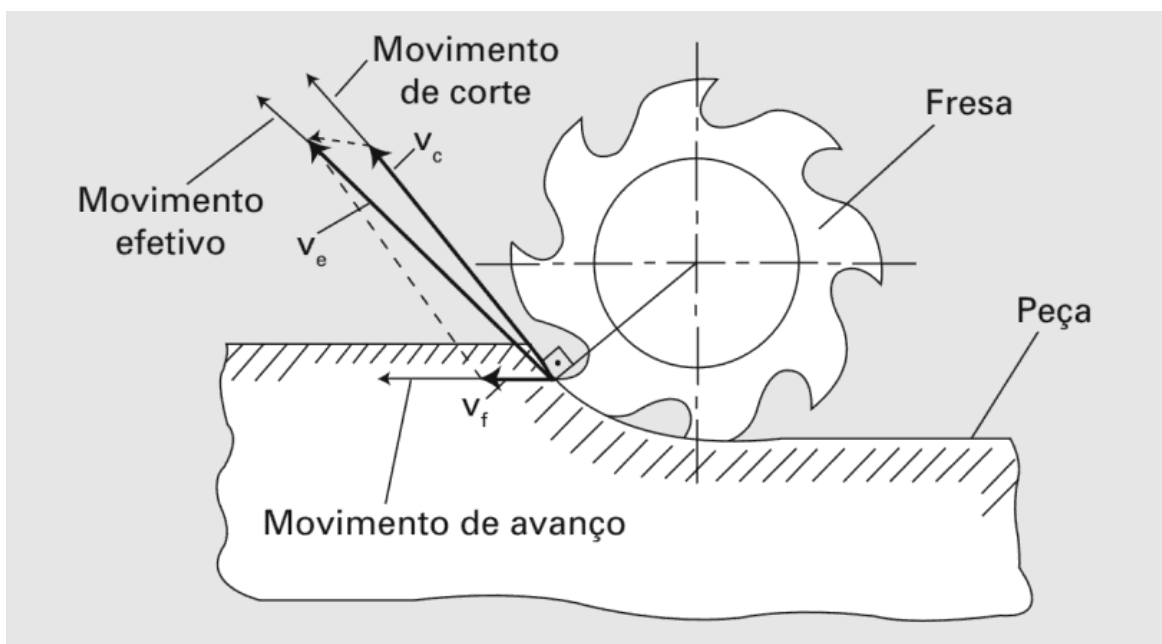
Fonte: MACHADO, Álisson (2015).

Figura 2.10 - Direção e movimentos na furação



Fonte: MACHADO, Álisson (2015).

Figura 2.11 Direção e movimento no fresamento



Fonte: MACHADO, Álisson (2015).

Entre as grandezas físicas presentes no processo torna-se imprescindível a avaliação criteriosa da velocidade de corte adotada dada a finalidade de se obter maior ordem de conformidade com o planejamento e melhor controle da operação.

O parâmetro de usinagem velocidade de corte - adotado como a variável V_c em fórmulas habituais - classifica-se como a velocidade instantânea dado um ponto de referência da ferramenta em função de uma direção e sentido do corte. MACHADO, Álisson (2004).

Figura 2.11 - Fórmula de cálculo da velocidade de corte

$$v_c = \frac{\pi \cdot d \cdot n}{1.000} \quad [\text{m/min}]$$

Fonte: MACHADO, Álisson (2015).

Onde evidenciam-se “d” como diâmetro da peça ou ferramenta - disposta em milímetros - e “n” como o número de rotações por minutos - disposta na medida de RPM -, sendo a unidade resultante expressa em metros/minuto.

Não obstante, diferentes processos podem requerer cálculos distintos para a finalidade, uma vez que seu método de atuação distingue-se e, por conseguinte, suas variáveis envolvidas na usinagem, como é o caso do processo de movimento retilíneo de aplainamento.

Figura 2.12 - Fórmula de cálculo da velocidade de corte

$$v_c = 2 \cdot c \cdot \text{gpm} / 1000 \quad [\text{m} / \text{min}]$$

Fonte: COSTA, Éder; SANTOS, Denis (2006).

A composição dessas informações habitualmente estão relatadas dada a designação referenciada pelos fabricantes - sendo então fornecida em função do produto - e, sobretudo, empenham papel crucial não só na identificação da adequação ao projeto como também na vida útil da ferramenta - originando assim outro critério de avaliação relevante na tomada de decisão.

A velocidade de corte é o parâmetro de corte mais influente na vida da ferramenta, tal como também influencia intimamente na duração presente nas etapas de produção. O tempo mínimo de confecção por peça pode ser disposto dadas as equações de percurso de avanço e

tempo de corte - T_c para tempo de corte e L_a para percurso - dispostos nas fórmulas seguintes:

Figura 2.6 - Fórmula para a obtenção do percurso de avanço

$$l_a = v_a \cdot t_c = a \cdot n \cdot t_c.$$

Fonte: FERRARESI, Dino (2018).

Figura 2.7 - Fórmula para a obtenção do tempo de corte

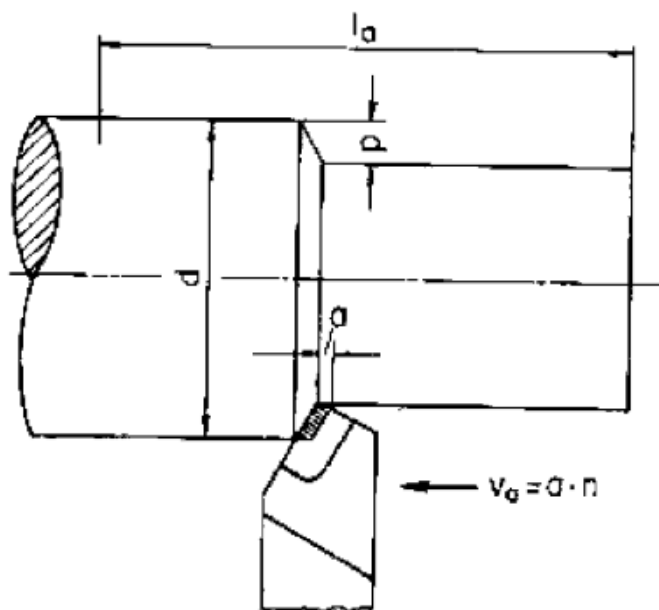
$$t_c = \frac{l_a \cdot \pi \cdot d}{1000 \cdot a \cdot v},$$

Fonte: FERRARESI, Dino (2018).

Onde “d” equivale ao diâmetro da peça, em mm; “a” denota o avanço, em mm/volta e “v” indica a velocidade de corte, em m/min. Subentende-se então, em virtude das implicações presentes na disposição das fórmulas obtidas, direta função exercida pela velocidade de corte no tempo de produção.

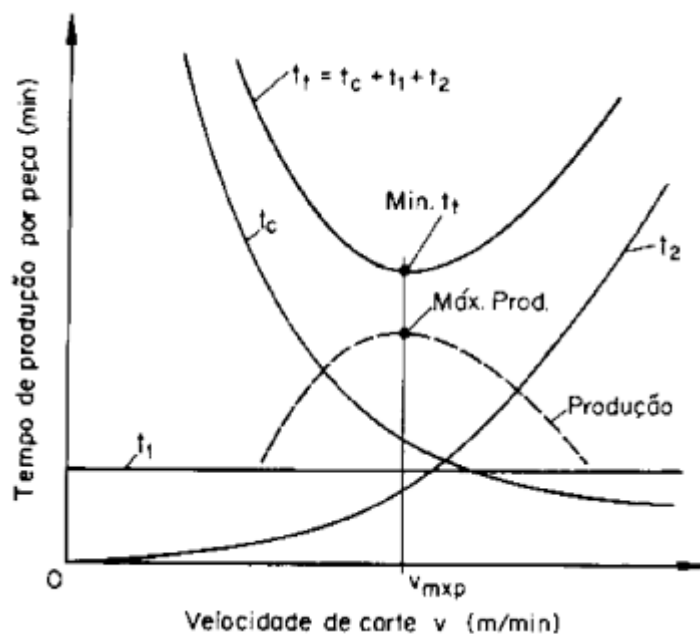
Dispostas no processo de usinagem através da seguinte ilustração:

Figura 2.8 - Indicação das variáveis de percurso de avanço e tempo de corte



Fonte: FERRARESI, Dino (2018).

Figura 2.9 - Representação das diferentes parcelas do tempo de fabricação por peça dada uma velocidade de corte adotada



Fonte: FERRARESI, Dino (2018).

Dessa forma, não tão somente a velocidade de corte influencia na vida útil da ferramenta, suas propriedades admitidas de corte, condição da superfície após o torneamento quanto, essencialmente, enlaça-se a fatores essenciais na determinação do tempo de produção.

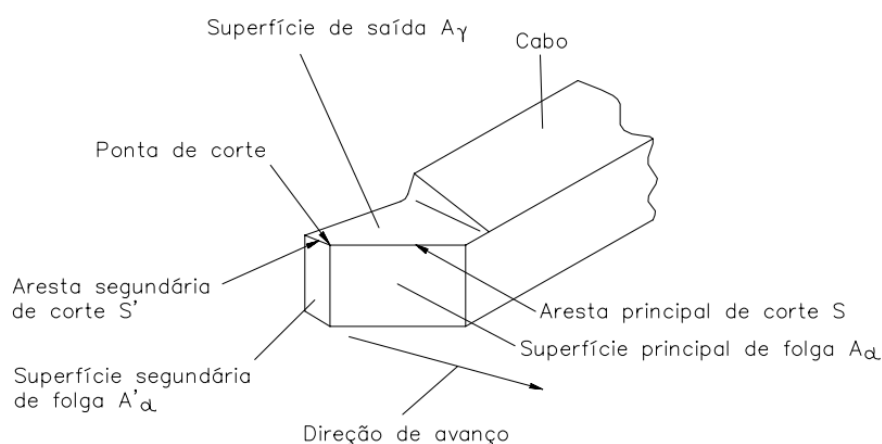
2.3 Geometria das ferramentas de corte

Dada a lei fundamental da cunha - que implica na distinção no que se refere ao trabalho exercido na máquina simples e o cisalhamento resultante - e as implicações na correspondência da ferramenta adotada em função das tensões exercidas - em conformidade com suas propriedades intrínsecas -, constata-se a importância de realizar aferições dimensionais corretas e atribuir as características físicas e químicas apropriadas para a aplicação da ferramenta na operação desejada.

As ferramentas de corte sofrem influência de suas respectivas geometrias tal como outros fatores, desse modo, torna-se necessário definir com certo rigor suas propriedades intrínsecas a fim de obter conformidade com a sua aplicação. No caso das características físicas e disposições dimensionais, evidencia-se o ângulo de “cunho” - dado o seu efeito no cisalhamento do material submetido à usinagem. COSTA, Éder; SANTOS, Denis (2006).

Entre as distintas características dimensionais que possuem influência direta na usinagem resultante, é possível observar, a critério de caracterização, as seguintes definições: cunha de corte, superfície de saída, superfície de folga, aresta de corte - discernida entre aresta principal e secundária -, ponta de corte e ponto de corte escolhido.

Figura 2.10 - Cunha de corte da ferramenta



Fonte: MACHADO, Álisson (2015).

A princípio, reside na cunha de corte grande importância no que se refere a formação de cavaco e determinação do movimento relativo.

Cunha cortante é a denominação designada à porção da ferramenta onde ocorre a originação do cavaco por intermédio do movimento relativo entre a peça e a ferramenta, sendo as suas arestas - superfícies de delimitação - denominadas de arestas de cortes. Tais arestas estão dispostas em diferentes categorias, tal como angulares ou curvilíneas. FERRARESI, Dino (2018).

Em função de qualquer material de ferramenta e de peça há uma ou mais geometrias reincidentes como adequadas ou ótimas em virtude dos parâmetros influenciados pela sua composição geométrica, tal como é evidenciado no acervo Fundamentos dos processos de usinagem da USP da autoria do professor de eng. Rodrigo Lima Stoeterau:

Formação do cavaco;

Saída do cavaco;

Forças de corte;

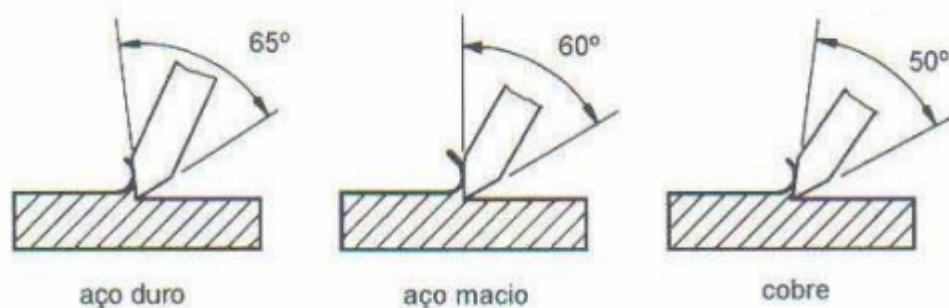
Desgaste da ferramenta;

Qualidade do trabalho.

Em convergência com o atestado, explicita-se no ângulo de cunha oferecida pela geometria da ferramenta uma relação direta quanto à resistência do material que será usinado - especificamente aquela que é oferecida ao corte -, de tal maneira que essa variação no ângulo acentue-se conforme há discernimento entre as resistências oferecidas por diferentes materiais.

O ângulo de cunha é dimensionado de acordo com a resistência que o material usinado oferece ao corte. Essa resistência será tanto maior quanto maior for a dureza e a tenacidade do material. COSTA, E. S.; SANTOS, Denis Júnio (2006).

Figura 2.11 - Variação do ângulo da cunha, em função da dureza do material



Fonte: COSTA, E. S.; SANTOS, Denis Júnio. Disciplina: Processo de Usinagem. Divinópolis, MG, p. 6, 2006.

2.4 Desenvolvimento web

2.4.1 Definição

O desenvolvimento web pode ser definido como o processo de estudo, análise, confecção e corroboração - além da origem e construção - de tecnologias voltadas para a internet - como sites de grandes empresas e aplicações online voltadas para algum serviço, como catálogos de compra, por exemplo. Em virtude da acessibilidade e facilidade de um usuário convencional de smartphones ou computadores para acessar esse meio, a área de atuação possui tendência a desenvolver-se progressivamente a fim de possibilitar maior aproveitamento dos interesses das corporações com melhor qualidade de prestação para os clientes e usuários das respectivas redes.

Tal como no âmbito dos processos de fabricação há inúmeras técnicas e tecnologias - tais quais os processos de conformação mecânica conhecidos e as máquinas-ferramenta -, no contexto do desenvolvimento de aplicações online estão inseridas inúmeras práticas, recursos, plataformas e linguagens disponíveis a fim de exercer uma grande variedade de tarefas - hodierna e comumente separadas em função de sua posição em função da interface do usuário e a natureza dos dados manejados - como ocorre na distinção das tecnologias e aplicações frontend e backend.

No que toca o frontend, seu desenvolvimento está intimamente ligado à interface de usuário, dispondo de recursos voltados para a acessibilidade, estética, conteúdo, estrutura e lógica implementada diretamente ao usuário no decorrer de sua contemplação e interação, de tal forma que disponha comumente da integração dos elementos de estruturação, estética e lógica a fim de compor e suprir um ou mais serviços - distinguindo-se do desenvolvimento backend que, majoritariamente, elabora e desenvolve elementos que não estão inseridos no contexto de percepção e interação direta com o usuário - assim sendo, não compõe diretamente os elementos intrínsecos à interface do usuário -, contudo volta-se com maior preocupação e instrumentos aquém à obtenção, disposição, transferência e envio de dados - de tal forma que, ao contrário da interface, possa os alterar e obter permanentemente por intermédio de um ou mais serviços almejados pelo cliente.

A responsabilidade fundamental de um desenvolvedor frontend refere-se à interface de usuário, de tal forma que esse planeje, construa e desenvolva o conteúdo que o usuário verá diretamente. Sua atividade laboral compreende a criação de aplicações e sites da internet por intermédio de um ou mais recursos como as linguagens de programação, sendo as mais comuns e basais: HTML, CSS e JavaScript (W3SCHOOLS, Frontend, 2024).

Dessa forma, pode se definir frontend como desenvolvimento voltado para como a aplicação se parecerá em contrapartida com como ela funcionará - apesar dessa definição não levar em consideração a lógica intrínseca ao lado cliente da aplicação que, apesar de preponderantemente menor em grande parte das aplicações de grandes empresas quando comparada com o do lado servidor, ainda se faz presente. Dessa forma, uma aplicação completa comporta um relacionamento de cliente-servidor em uma rede de computadores a fim de prestar um ou mais serviços.

Em suma, caracteriza-se como o desenvolvimento de aplicações pertencentes à internet - possuindo categorias, recursos e práticas inerentes que sistematizam o processo de criação e o tornam possível. Dessa forma, o desenvolvimento web é um produto dos vastos recursos presentes, necessidade do contexto inserido e prolongadas pesquisas no âmbito da informática, tecnologia e, de perspectiva gerencial, possível implementação e benefícios.

2.4.2 Importância

As aplicações web possuem rápido e crescente desenvolvimento no que toca a sua abrangência e aplicabilidade na hodiernidade, de tal forma que a sua extensão de uso e escopo tem afetado continuamente todos os aspectos da vida. CONTE, Tayana (2005).

Dessa forma, a sua importância se encontra compreendida nos âmbitos intrínsecos à sua aplicação - extensa e em crescimento - que incentiva ainda mais investimentos nos métodos e tecnologias voltadas ao desenvolvimento citado, de tal forma que possa ser aproveitado para diferentes setores de serviços e utilidades.

Consoante ao constatado, a sua aplicabilidade por ser orientada para inúmeros propósitos e não se limita à algum parâmetro imediato de construção e uso, mas sim aos critérios estabelecidos em função da necessidade estabelecida - em suma, sua forma se adapta às necessidades do contexto inserido, assim como toda a tecnologia.

Uma das necessidades mais discutidas na contemporaneidade cujo enfoque nas grandes empresas cresceu drasticamente a competitividade e eficiência na indústria e meios corporativos se dá na forma de um importante parâmetro: a produtividade.

O conceito de produtividade pode ser caracterizada na forma da capacidade da empresa de gerar o seu produto no decorrer de seu processo produtivos, de tal forma abrangente na qual seu conceito não limite-se ao processo de produção em si, uma vez que a geração de valor também depende de distintas etapas adjacentes de fundamental importância, tais como a compra dos bens e serviços intermediários, a estratégia de compra, logística, venda dos bens e serviços, e entre outros. Configurando uma articulação dinâmica de várias competências que devem ser exploradas e exercidas com propriedade e embasamento a fim de obter maior resultado nos critérios pertencentes a esse parâmetro. MACEDO, Mariano (2018).

Uma das alternativas mais comuns encontradas na indústria e empresas em ascensão presentes nos tempos contemporâneos tem sido investir em inovação a fim de obter vantagem no que se refere aos parâmetros de produção. A implementação da automatização dos processos, novos paradigmas de produção, constante financiamento no âmbito da pesquisa e contínua análise e desenvolvimento na administração e logística da gerência das empresas são exemplos sólidos dos esforços comuns e, atualmente, necessários feitos pelas empresas a fim de não só de destacar como também sobreviver no mercado.

O contexto atual é marcado por traços de extrema competitividade onde a inovação se tornou um desafio contínuo e permanente para todas as organizações - no que independe de

seu setor de atividade e área de negócio -, de tal forma que a gestão da inovação é fundamental para que haja melhoria perene e progressiva do valor criado e, não tão somente, sendo também o motor para levar organizações ao sucesso e competitividade forte - não sendo exceção para as organizações de natureza industrial. CARVALHO, João (2018).

Coincidente ao evidenciado, a incorporação da informática não só se faz possível em inúmeros setores quanto, em razão do contexto hodierno e competitividade, se faz necessário: assim exprimindo com substancialidade e categoria o contexto da indústria 4.0 - na qual a implementação da tecnologia digital e informática se torna proeminente e extremamente vinculada à fração majoritária dos desenvolvimentos recentes e bem sucedidos.

No caso da Indústria, a natureza basal da automatização informatizada e perspectiva gerencial de negócios voltada para a transformação digital originam o conceito de Indústria 4.0. Integração de tecnologias de informação e comunicação que permitem alcançar patamares em ascensão de produtividade, flexibilidade, gerenciamento e qualidade tornam-se novas estratégias em evidência no contexto atual, configurando a denominada Quarta Revolução Industrial. SACOMANO, José (2018).

A indústria 4.0 promete aumentos na eficácia operacional, ganhos significativos na melhoria da competitividade e produtividade, além do desenvolvimento de novos modelos de serviços, negócios e produtos. Conforme computadores e máquinas em rede conseguem comunicar facilmente entre si e com seus respectivos utilizadores é possível tornar o processo de produção mais visível e monitorável tal como acrescer a confiabilidade dos resultados obtidos, assim decrescendo a chance de ocorrerem falhas. SANTOS, Beatrice (2018).

Figura 2.12 - Sistema ciber físico da informação



Fonte: SANTOS, Beatrice (2018).

Portanto, tornar-se-á imprescindível a implementação de melhores métodos e estruturas com a finalidade de dar suporte às inovações implementadas e, crucialmente, investir planejamento em setores com potencial de inovação - tal como é o caso da usinagem e seus processos intrínsecos -, sendo a precisão, produtividade e confiabilidade dos resultados parâmetros exploráveis de proeminente evidência e rigor, no que resultam na qualidade do processo como um todo.

2.4.3 JavaScript

JavaScript possui maior praticidade no que se refere ao processo de desenvolvimento e criação de aplicações web - em função da maior facilidade de utilização de seus componentes front-end. Entre os seus benefícios inerentes é possível citar: maior responsividade, interação de usuários, ausência da necessidade de instalação de pacotes por parte do usuário - em contraste com os recursos oferecidos -, universalidade e acesso direto às entradas presentes na aplicação. GOH, Hock-Ann (2023).

Alguns parâmetros definidos a priori sustentam grande parte dos critérios de escolha voltados aos desenvolvedores e grandes empresas no momento de optar por uma determinada linguagem no que toca à criação de alguma aplicação, de tal forma que a sua disposição seja muito almejada e zelada pelos desenvolvedores de plataformas no momento de integrar novas funções ou inovar em algum aspecto - como nas frameworks, por exemplo.

JavaScript se tornou a linguagem de programação de fonte aberta mais utilizada nos últimos nove anos - habitualmente reconhecida como uma linguagem convencional e universal para navegadores - o que por sua vez infere em maior suporte por parte dos navegadores para integrar seus recursos, crescente criação e implementação de melhorias e acesso direto a diferentes dispositivos. GOH, Hock-Ann (2023).

Dessa forma, a escolha da linguagem se dá pela sua precedência, confiabilidade e fácil manutenibilidade, visto que possui constantes investimentos por parte dos desenvolvedores pertencentes à comunidades e uma variedade significativa de ferramentas disponíveis a fim de implementar utilidades ou otimizar a aplicação.

2.5 Angular

2.5.1 Origem

Um framework é construído e idealizado como uma solução para problemas e possibilitador de acréscimos no âmbito da produção empregada à tecnologia - de tal forma que seu objetivo é aumentar a produtividade e oferecer essas determinadas soluções. FONSECA, Luiz (2018).

Outra vantagem da grande parte dos frameworks é que tarefas repetitivas podem ser automatizadas, é o conceito conhecido como dry - don't repeat yourself, que pode ser traduzido para "não se repita". Em uma aplicação que tenha de manipular dados vindos de uma tabela na base de dados, as operações de inclusão, exclusão e alteração são praticamente iguais para todas as tabelas envolvidas. Não teria sentido repetir o esforço para desenvolver esse código fonte várias vezes, e a geração dessas funções poderia ser automatizada por alguma ferramenta contida no framework. MINETTO (2007).

Na contemporaneidade, inclusive no Brasil, uma boa gerência na produtividade se torna gradativamente mais crucial em um ambiente com propósitos globalizadores e de abertura externa - tal como seu efetivo crescimento. Dessa forma, a ausência da produtividade tende a diminuir a probabilidade de sobrevivência das empresas no mercado dados fatores como o acirramento da concorrência e contínua formulação de estratégias de competitividade das empresas. MACEDO, Mariano (2018).

Consoante aos expostos, a criação de um framework visa acrescer um forte e importante parâmetro de avaliação, sendo esse a produtividade - que também encontra-se intimamente enlaçado aos interesses das indústrias.

O framework Angular, desenvolvido pela Google, é capaz de suprir inúmeros requisitos no quesito escolha e prover recursos importantes para o desenvolvimento de aplicações web de página única e componentes dinâmicos e esteticamente equivalentes, sendo uma tecnologia robusta, clara e com alta reatividade. GEETHA, G, (2022).

Angular, categoricamente, é uma plataforma de desenvolvimento construída em TypeScript, que inclui cobertura de abrangente variedade de bibliotecas com distintas incrementações, recursos de utilidade, ferramentas que ajudam o desenvolvedor a construir, testar e atualizar seu código, além de uma estrutura adequada para a formulação e corroboração de aplicações web escaláveis. (GOOGLE. Angular, ago. 2023).

Sobretudo, por possuir contínuo e prevalente suporte da Google, a estrutura analisada dispõe de recorrentes atualizações, boa atividade da comunidade por parte da implementação de ideias e desenvolvimento de novos interesses e necessidades para o aprimoramento do framework e melhor manutenção dados os investimentos recentes da empresa na tecnologia - sendo assim confiável e, sobretudo, facilmente implementável e sustentável ao longo do tempo.

2.5.2 Estrutura

O framework possui uma disposição configurativa embasada por intermédio de preceitos, estruturas e de seus elementos de atividade preponderante no decorrer da construção e atuação da aplicação. Dessa forma, define-se fração majoritária de sua propriedade por intermédio de suas composições presentes - sendo exemplos consistentes, com o propósito de embasar de forma sucinta e coesa, os: módulos, componentes, diretivas, DataBinding e serviços.

Módulos são categorizados como blocos de código com um determinado propósito, tal como se comporta a definição para uma classe, função e constantes - de tal forma que um conjunto desses compõe parte importante da aplicação. Cada módulo responde a uma determinada pasta com arquivos relacionados e, dada a sua natureza de articulação em pequenas partes - se comportando como porções desmembradas da aplicação - torna-se mais preciso e eficiente realizar testes com o propósito de avaliar parâmetros importantes ou inspecionar alguma avaria na integridade do aplicativo. FONSECA, Luis (2018).

Devido a natureza configurativa da plataforma, é possível definir Angular como uma aplicação modular - de tal forma que seu sistema de módulos é denotado como: NgModules. Aquém ao seu critério de definição, os NgModules são containers dedicados a blocos coesivos de códigos em um domínio de aplicação. Entre suas utilidades cita-se a importação e exportação de funcionalidades para uso determinado (GOOGLE. Angular, ago. 2023).

Parte essencial de sua construção denota-se como os componentes, esses os quais possuem articulação conjunta implementável entre outros componentes e podem ter comportamentos expressos de maneira isolada. A caracterização de partes do componente manifesta-se no formato de hipertexto, estilização em cascata ou linguagem interpretada - HTML, CSS e TypeScript, respectivamente -, assim sendo capazes de implementar no

restante da aplicação uma só funcionalidade que poderá vir a ser submetida a testes, aprimorada ou rearranjada com maior facilidade a posteriori.

Componentes são definidos como o bloco de construção fundamental para a criação de aplicações em Angular e, por intermédio do aproveitamento desses elementos, a plataforma almeja prover estruturas de organização com responsabilidades próprias que são altamente manuteníveis e escaláveis, acrescentando a eficiência no que toca o gerenciamento do projeto (GOOGLE, Angular, ago. 2023).

Dessa forma, é possível utilizar um mesmo componente várias vezes ao longo da construção da aplicação, da mesma forma que é possível os retirar ou alterar - o que, nesse caso, implicaria na edição do componente como unidade, sendo essas mudanças replicadas em cada uma das manifestações desses blocos de construção - com muita precisão e simplicidade.

Os módulos da plataforma provem aos componentes um contexto de compilação, de tal forma que todo componente que pertença a um mesmo módulo compartilhe de um contexto de compilação próprio - ou seja, compartilhe de arquivos e elementos imprescindíveis para sua compilação, processo no qual torna-se-á possível executar determinado bloco de código. Vale também ressaltar que todo módulo raiz possui, necessariamente, um componente raiz enlaçado, contudo a adição de novos componentes a esse módulo pode ser realizada qualquer número de vezes - assim formando o contexto citado a priori.

Os componentes constroem os recursos apresentáveis presentes na aplicação, de tal forma que possa performar com alta dinâmica e articulação, sendo essa conjectura uma grande vantagem presente na estrutura, uma vez que o desmembramento dessas funcionalidades provê maior eficiência ao gerenciá-las sem comprometer seu rendimento.

É por intermédio da interação dos componentes integrados da forma mencionada acima que uma Template é formada, elemento importante a ser considerado no que abrange critério e estudo de uma plataforma de desenvolvimento, uma vez que desempenha papel crucial nos serviços prestados pela aplicação.

A interface do usuário - elementos estéticos, funcionais e gráficos acessíveis a esse - é passiva de importantes investimentos com o propósito de acrescer a acessibilidade, responsividade e praticidade das aplicações intrínsecas ao web development - de tal forma que frontend seja uma área de proeminente importância voltada, entre outros, a esse propósito -

não sendo diferente para a plataforma Angular. Denota-se como modelo esquemas configurados e estabelecidos com o objetivo de configurar o conteúdo - no caso do desenvolvimento web, em uma página de site - de tal forma que estabeleça-se com velocidade e qualidade.

Conforme o constatado, uma Template é constituída por um agrupamento de HTML - acrônimo para HyperText Markup Language, linguagem de marcação de hipertexto - que é processado e convertido no DOM - Document Object Model - com o propósito de mostrar a interface de usuário - a estrutura visível e passiva de interação pelo cliente. WUTTHICHAJ, Chansuwath; TWITTIE, Senivongse (2016).

Angular, como já mencionado a priori, disponibiliza recursos voltados à qualidade de trabalho e eficiência de construção, manutenção e atualização da aplicação, o que não é diferente para as Templates presentes.

Por intermédio de interpolação de dados é possível declarar um conteúdo dinâmico, diferentemente do estático habitual: conteúdo fluído em função dos dados enlaçados ao elemento HTML. Também é possível realizar uma atribuição dinâmica entre as propriedades e atributos de um elemento de hipertexto presente através do entrelaçamento do valor da determinada propriedade e um dado presente na aplicação, que pode ser alterado e, em razão dessa alteração, propagar mudanças inteligentes com lógica atribuída a inúmeros componentes - não tão somente limitando-se ao componente do qual se origina (GOOGLE, Angular, ago. 2023).

Os recursos descritos definem o conceito de Data Binding citado anteriormente, que fundamenta-se no vínculo não estático realizado entre porção de interação e construção lógica do programa com a Template presente - o que torna mais fácil e implementável possíveis recursos dinâmicos preferíveis ou necessários para a funcionalidade da aplicação.

Ainda no que toca a interface, os componentes exercem o papel da view - DOM virtual presente na estrutura - no que trata-se do encapsulamento dos módulos - dado o inferido, trata em primeira mão de DataBinding, MetaData e Template. Componentes localizam-se em pastas próprias com endereçamento específico para os recursos necessários mencionados, dispondo de três atributos distintos: seletor, Template Url e Style Urls, que exercem papel de endereçamento das implementações intrínsecas ao próprio componente. FONSECA, Luis (2018).

Em conformidade com o constatado, a melhor implementação dos elementos dispostos na interface às estruturas lógicas e dados intrínsecos a aplicação fundamenta-se como prioridade relevante, possibilitando integração dinâmica das funcionalidades dos diversos componentes que compõem uma determinada aplicação.

3. TECNOLOGIAS E MÉTODOS

3.1 Projeto

Um aplicativo, denominado de IUsinagem, foi idealizado com o propósito de testar as possíveis utilidades de uma aplicação web no que trata do ofício de atuação da usinagem, provocar novas reflexões aquém ao tópico e desenvolver as ideias centradas no documento: implementação da informática na usinagem.

A priori, é necessário realizar um planejamento com rigor e critério equivalente à natureza do trabalho, de tal forma que seja possível precipitar seu esboço, idealizar possíveis problemas e fomentar o seu desenvolvimento em virtude dos seus objetivos.

A princípio, como uma aplicação voltada a auxiliar nos processos de usinagem, compreende-se primeiro as suas respectivas necessidades e parâmetros de conformidade: uso das máquinas; sua configuração; dimensões, condições e material processado, além das dificuldades operacionais e vantagens latentes oferecidas pela sua possível implementação.

Fração preponderante da qualidade oferecida pelo seu processamento, quando não enlaçada pela aptidão operacional, cabe aos parâmetros estabelecidos para a sua confecção: tais como a ferramenta utilizada e suas especificações explicitadas pelo respectivo fabricante.

Tais parâmetros também norteiam e orientam a sua operação, além de, por muitas vezes, requerer adaptação dessa para atender às necessidades dos recursos dispostos.

3.1.2 Objetivo

As profundidade e velocidade de corte e avanço intrínsecas a uma pastilha são critérios de essencial integração e aperfeiçoamento caso seja almejada maior qualidade e produtividade em processos que envolvem torneamento geral - de tal forma que seu tempo de usinagem, velocidade de fuso e taxa de remoção do material sejam estabelecidos em sua função.

Não obstante, cada uma dessas designações também são obtidas em razão das condições de usinagem, material e dimensões adotadas no processo - além do tipo de operação realizada, variando caso seja de acabamento ou desbaste, por exemplo.

Dadas as variáveis apresentadas, cálculos que as envolvem e variedade de ferramentas disponíveis - cada qual com especificações distintas que irão alterar por completo as variáveis mencionadas - e a sua importância no que toca as referências de qualidade adotadas e sua constante utilização no decorrer dos processos envolvidos, julga-se de preponderante interferência e imprescindível aperfeiçoamento a melhor e mais precisa obtenção de todas essas variáveis para a atuação em oficinas de usinagem - de tal forma que a negligenciar implique em falhas e ausência de investimentos nela acarrete em perdas dadas as exigências crescente de produtividade no mercado hodierno.

Conforme o constatado, torna-se possível acrescer a confiabilidade das informações obtidas, diminuir o tempo necessário investido para as obter e, sobretudo, melhorar a qualidade de serviço e operação dos envolvidos ao implementar uma aplicação capaz de automatizar parte das tarefas com critério de qualidade equivalente.

Congruente ao exposto, o projeto implementado nasce da proposta de prover esses serviços e, uma vez que tenha êxito, ser capaz de suprir as necessidades que o seguem.

Entre os requisitos para alcançar esse objetivo, cita-se:

- Possuir os dados de interesse;
- Total habilidade de manusear esses dados a fim de atender as expectativas intrínsecas à sua atuação;
- Capacidade de interpretar as necessidades obtidas pelo usuário, tais quais o estado pré-usinagem, condições e dimensões;
- Apresentação dessas informações e incorporações de seus elementos que concedam acessibilidade e fácil manejo por parte do usuário;
- Integração de uma interface capaz de prover os critérios estabelecidos anteriormente;
- Disponibilidade e disposição adequadas para seu uso intermitente;
- Confiabilidade no que toca os dados introduzidos e valores obtidos.

Tais exigências são tratadas como condições de existência para a aplicação, de tal forma que cada uma seja integrada com tecnologias de equivalente confiabilidade à tarefa exercida.

A aplicação será criada no framework Angular - HTML, CSS e TypeScript - com integração de suas funcionalidade e recursos - tal como os módulos do Angular Material e diretivas presentes -, dispondo de componentes que interagem e são capazes de cumprir com cada uma das condições estabelecidas.

3.2 Componentes

A fim de implementar a informática por intermédio dos viés abordados anteriormente, fez-se uso de distintas tecnologias que articularam funcionalidade e utilidades a fim de trazer maior eficiência no âmbito da acessibilidade e aplicação para o projeto.

Esse terá as suas funções divididas em componentes com alta capacidade de integração e fácil incorporação nos diferentes trechos da aplicação - dessa forma, cada parte terá um propósito e trabalho, facilitando a manutenção, melhora e futuras mudanças do aplicativo.

Os componentes geralmente possuem a importação de módulos, diretivas ou outros oriundos da pasta `node modules` presente na aplicação - de onde são obtidos recursos presentes na `framework` gerenciados pelo gerenciador de recursos do `node` - `npm` - os quais variam a depender das necessidades dos componentes.

3.2.1 Toolbar

A maioria das aplicações web possuem uma aba localizada na porção superior da interface com o objetivo de mostrar informações básicas - como o título e algumas notações referentes ao seu conteúdo - e, sobretudo, fontes de utilidade - como referências a partes do aplicativo, terceiros que estejam envolvidos ou utilidades como acesso e cadastro.

Habitualmente sua utilização é proveitosa, uma vez que atende necessidades estéticas e provê informações e acessos úteis que, comumente, serão de interesse imediato do usuário ao se deparar com o aplicativo.

O componente criado para a aplicação `IUsinagem` possui a importação de dois módulos referentes à infraestrutura Angular Material: `MatToolbarModule` e `MatIconModule` - ambos responsáveis por traços comportamentais do componente e estética apresentada ao usuário.

Os módulos dispõem da estrutura `mat-toolbar` e `mat-icon`, que são incorporados no esqueleto do componente localizado no arquivo `toolbar.component.html` - cada estrutura possui certas funcionalidades exploradas a fim configurar o componente.

Duas estruturas estão localizadas, cada qual com uma classe de estilização com ramificações que almejam maior especificidade em seus estilos: `toolbar`, `t1` e `t2` - sendo `t1` e `t2`, respectivamente, responsáveis por denotar diferentes aspectos no arquivo `toolbar.component.css`

Seus elementos filhos são o título da aplicação, hiperlinks com o site oficial do IFBA e código fonte da aplicação, além dos ícones referentes a cada hiperlink utilizado - expressos pelas estruturas mat-icon.

A folha de estilo foi utilizado para prover o gradiente de cor das estruturas, especificar parâmetros de altura, posicionamento do conteúdo, sua disposição, condições do texto expresso, espaçamento do conteúdo e reconfigurar a estilização intrínseca aos hiperlinks - que naturalmente possuem decoração de texto e cores determinadas que não condizem com os traços estéticos almejados.

Figura 3.1 - Template do componente toolbar

```

1  <p>
2      <!-- First toolbar element: title -->
3      <mat-toolbar
4          class="toolbar t1"
5      >
6          <!-- Title of the page -->
7          <h1>IUSINAGEM</h1>
8      </mat-toolbar>
9
10     <!-- Second toolbar element: links -->
11     <mat-toolbar
12         class="toolbar t2"
13     >
14         <!-- First link - IFBA home page -->
15         <a
16             class="link"
17             href="https://portal.ifba.edu.br/"
18             matTooltip="Página oficial da instituição"
19             target="_blank"
20         >
21             <mat-icon>
22                 home
23             </mat-icon>
24             <span
25                 class="text"
26             >
27                 Instituto Federal de Educação, Ciência e Tecnologia da Bahia
28             </span>
29         </a>
30
31         <!-- Spacer -->
32         <span
33             class="spacer"
34         >
35         </span>
36
37         <!-- Second link - github source code page -->
38         <a
39             class="link"
40             href="https://github.com/Samuel-Pinheiro-C-Lopes/Iusinagem-v1"
41             matTooltip="Acesso ao código fonte da aplicação"
42             target="_blank"
43         >
44             <mat-icon>
45                 code
46             </mat-icon>
47             <span
48                 class="text"
49             >
50                 Código
51             </span>
52         </a>
53     </mat-toolbar>
54 </p>

```

Fonte: Autor.

A classe do componente não exporta nenhum elemento lógico contido além dos módulos importados e endereçamentos, servindo como elemento de apresentação, estética e disposição de links de interesse para o usuário.

3.2.2 Main

Componente responsável por comportar a maior parte dos componentes pertencentes às funcionalidades inerentes ao aplicativo - como os catálogos, entrada e saída de dados e estruturas lógicas - que ocupará o conteúdo total da aplicação.

Em sua estrutura lógica ocorre a importação do módulo: Mat Divider - provê a estrutura mat-divider útil para a separação dos componentes filhos presentes -, e importação dos componentes: Stepper Component, Catalog Component e Carousel Component, que serão abordados em suas categorias respectivas a posteriori no documento - cada qual exerce funções particulares.

Nesse componente há o armazenamento da interface insert - cujo tipo denota parâmetros pertencentes a uma pastilha, divididos em: iso, geometry, class, vc, fn, ap, material, condition, operation e machine - cada qual sendo o seu código, geometria, classe, velocidade de corte, avanço por rotação, profundidade de corte, material, condição, operação e porta-ferramenta.

Vetores de pastilhas - external e internal - são declaradas e inicializadas em sua classe de exportação em função dos dados obtidos do catálogo da Sandvik de ferramentas de corte, cujos conteúdos são transferidos por intermédio de diretivas aos componentes filhos que se responsabilizarão por manejar esses dados com diferentes propósitos.

Figura 3.2 - Interface do objeto pastilha

```
16 export interface insert {  
17   iso: string,  
18   geometry: string,  
19   class: number,  
20   vc: number,  
21   fn: number,  
22   ap: number,  
23   material: string,  
24   condition: string,  
25   operation: string,  
26   machine: string,  
27 }
```

Fonte: Autor.

Em sua estrutura no arquivo main.component.html há a disposição dos componentes separados pela estrutura mat-divider - denota uma linha que os separa visualmente - com a designação dos atributos de cada component: como é o caso de [data] dos componentes app-catalog e [externalInsertData] e [internalInsertData] do component app-stepper.

O conteúdo de texto da aplicação também é denotado em seu interior, dispondo de classes específicas para cada bloco de texto dada a sua serventia, tamanho e organização - a fim de serem distribuídos ao longo da aplicação.

Sua folha de estilo atribui espaçamento para o component, e alinhamento de texto e margem para o título dos respectivos catálogos.

3.2.3 Catalog

A disposição dos dados referentes às pastilhas se dá por intermédio deste componente, com páginas que exprimem as pastilhas utilizadas e processadas na aplicação - tanto para as modalidades de torneamento interno quanto externo.

3.2.3.1 Estrutura

Em sua estrutura lógica há a importação dos módulos: Mat Paginator e Mat Paginator Module - cuja função exercida no componente se dá pela implementação da paginação dos catálogos -; MatTableModule e MatTable DataSource - responsáveis pela estrutura mat-table e atribuição de interface, respectivamente - e importação da interface insert - utilizada para configurar as variáveis cuja atribuição se dará pelo atributo obtido por intermédio da diretiva @Input presente.

As diretivas Input, método enlaçado ao ciclo de vida do componente ngOnInit() advindo do OnInit, método de chamada dada a interação com componente ngAfterViewInit() da instância de componente AfterViewInit e o decorador de propriedade View Child são importadas nesse componente com o propósito de obter atualização das interações com o componente - em função das interações do usuário -, inicializar a fonte de dados da estrutura mat-table após a atribuição dos dados obtidos de seu componente pai e configuração da disposição dos itens dada a configuração de paginação atual.

Figura 3.3 - Inicialização dos dados pertencentes ao catálogo e atribuição do paginador à variável

```

53  dataSource!: MatTableDataSource<insertImg, MatPaginator>;
54  ngOnInit() {
55      const data1:any = [
56
57      ]
58
59      this.data.forEach(insert => {
60          data1.push({...insert, ...{forma: `./assets/insertsImgs/${insert.iso.charAt(0)}.png`}});
61      });
62
63      this.dataSource = new MatTableDataSource<insertImg>(data1);
64  }
65
66
67  /*
68   The paginator property is updated based on the callback method focused in
69   the MatPaginator setted, making it changes whenever the user interacts with
70   the structure
71  */
72  ngAfterViewInit(): void {
73      this.dataSource.paginator = this.paginator;
74  }
75

```

Fonte: Autor.

Com o propósito de atribuir imagens correspondentes a cada pastilha no catálogo o vetor de pastilhas é iterada após a inicialização do componente, originando um objeto com os parâmetros da pastilha e fonte da imagem correspondente a forma da pastilha - para então ser atribuída aos dados do catálogo.

A estrutura mat-table irá dispor de colunas referentes a cada objeto disposto no vetor de pastilhas, possuindo containers com a declaração da variável presente em cada iteração e coluna correspondente - tal padrão se repete para cada elemento pertencente ao objeto pastilha alterado adaptado para o catálogo.

No final de sua estrutura ocorre a disposição das linhas dadas as colunas correspondentes, assim preenchendo o catálogo em função das informações obtidas - assim não necessitando de alterações em virtude de novas pastilhas obtidas na fonte dos dados, podendo se comportar corretamente dado um número não determinado de elementos dispostos no vetor obtido.

A estrutura mat-paginator também está localizada no final da estrutura, logo após a delimitação da estrutura mat-table, com a determinação do número de páginas, rótulos e configuração dos botões adicionais.

O elemento table e classe table-insert são denotadas na folha de estilo com o objetivo de especificar as dimensões da estrutura e tamanho das imagens, respectivamente.

3.2.4 Stepper

A aplicação possui um componente com etapas de entrada de dados e respostas do usuário por intermédio da interface disposta através de passos delimitados exportado e denominado de Stepper Component.

Em sua estrutura lógica são importados os módulos: Mat Button Module - referente a traços comportamentais e estéticos de estruturas do tipo botão -; Mat Input Module - referente a estética das estruturas de entrada presentes -; Mat Icon Module - dispondo de estruturas de ícones utilizadas por suas dependências -; Mat Select Module - responsável por prover estruturas de seleção com configuração de estilização própria -; Mat Divide Module - adiciona estrutura de divisão para organizar a disposição dos elementos -; Form Module e Mat Form Field Module - utilizados na atribuição de entrada de dados e etapas do componente no que toca a sua obtenção e inserção -; Common Module - dispondo de recursos de dependência de outros módulos e articulação desses - e Mat Expansion Module - adiciona estrutura de expansão dividida em painéis dinâmicos com conteúdo oculto acessível.

Ocorre também a importação do operador map, a interface Observables e o método Breakpoint Observer - necessários para a implementação da adaptação da configuração da estrutura mat-stepper dada a tela visualizada - se tornando vertical uma vez que o tamanho seja reduzido o bastante.

Os componentes Panel Component e Tabs Component são importados e utilizados como componentes filhos com o propósito de iterar e exprimir os dados oriundos das especificações dadas pelo usuário.

A interface inserts é importada com o propósito de ser utilizada nas demais interfaces que requerem o tipo designado às pastilhas e variáveis manejadas no decorrer da estrutura lógica do componente.

A diretiva Input é importada para que haja a atribuição dos dados referentes às pastilhas utilizadas.

Há maior concentração de recursos voltados a esse componente em razão da função delegada: se comportar como funcionalidade principal da aplicação - uma vez que será nela onde ocorrerá a entrada de dados, manejo desses e expressão deles na interface.

O componente também declara e exporta as interfaces: geometry - referente aos trechos usinados que o usuário poderá inserir, em razão de um início e fim em comprimento, medida de diâmetro, tipo de usinagem e vetor com pastilhas e parâmetros referentes à

geometria determinada -; process - referente às especificações do projeto obtidas pelo usuário na interface, conforme esse entra com as dispostas seleções e dados de entrada, com os elementos element, preLength, preDiameter, condition, material, productsExternalGeometry e productsInternalGeometry, sendo esses, a peça requerida, comprimento pré-usinagem, diâmetro pré-usinagem, condição, material, geometria externa do produto e geometria interna do produto, respectivamente - e parameters - com os parâmetros de usinagem que virão a ser obtidos a posteriori da entrada de dados do usuário, contendo rpm, machining Time e metal motion Rate, sendo esses a velocidade de fuso, tempo de usinagem e taxa de remoção de metal, respectivamente.

Figura 3.4 - Interfaces geometry, process e parameters declaradas

```

23  export interface geometry {
24      start: number;
25      end: number;
26      diameter: number;
27      insertsAndParameters?: [insert, parameters][];
28      type: string;
29  }
30
31  export interface process {
32      element: string,
33      prelength: number,
34      preDiameter: number,
35      condition: string,
36      material: string,
37      productsExternalGeometry: geometry[],
38      productsInternalGeometry: geometry[],
39  }
40
41  export interface parameters {
42      rpm: number,
43      machiningTime: number,
44      metalRemotionRate: number,
45  }

```

Fonte: Autor

Em sua classe exportada há uma série de funcionalidades que são obtidas por intermédio de uma ou mais estruturas intrínsecas - como o construtor e funções - de tal forma que a sua configuração seja arranjada por categorias diferentes.

3.2.4.1 Responsividade

Stepper Orientation é declarada como um Observable do tipo Stepper Orientation - de tal forma que utilize da interface para realizar atribuições à Stepper Orientation que é

designada no construtor do componente - trecho de lógica implementada em sua construção, a priori da inicialização.

Figura 3.5 - Atribuição do valor de largura da tela à propriedade de alinhamento da estrutura

```

81     stepperOrientation: Observable<StepperOrientation>;
82
83     constructor(
84         breakpointObserver: BreakpointObserver,
85     ) {
86         this.stepperOrientation = breakpointObserver
87             .observe('(min-width: 700px)')
88             .pipe(map(({matches}) => (matches ? 'horizontal' : 'vertical')));
89     }

```

Fonte: Autor

Em seu construtor o Stepper Orientation é designado com base na largura atual da tela que, em função do valor lógico obtido da expressão - nesse caso, implementado como 'min-width:700px' - define o seu valor como horizontal ou vertical - o que será passado para as propriedades do componente, definindo a sua orientação como vertical para telas menores que seriam prejudicadas pela sua disposição horizontal.

A orientação é um atributo da estrutura mat-stepper que é manipulada na estrutura lógica do componente dada a observação de um parâmetro de tal - assim respondendo a uma visualização diferente e adequando a sua disposição a tal característica.

3.2.4.2 Dados Relevantes

Em sua estrutura há a declaração, e por muitas vezes inicialização, de múltiplas variáveis relevantes para o componente, como é o caso do objeto de tipo process denominado de process_config, contendo cada um dos seus elementos como 0, strings vazias ou vetores vazios, visto que dessa forma poderá manter os dados limpos a fim de esperar pela atribuição dada pelo usuário durante a sua entrada de dados realizada ao longo dos passos do componente.

Figura 3.6 - Declaração e inicialização do objeto process_config

```
100 public process_config:process = {  
101     element: '',  
102     preLength: 0,  
103     preDiameter: 0,  
104     condition: '',  
105     material: '',  
106     productsExternalGeometry: [  
107  
108     ],  
109     productsInternalGeometry: [  
110  
111     ],  
112 };  
113
```

Fonte: Autor

Também ocorre a declaração do external Insert Data e Internal Insert Data - e posterior inicialização com o valor atribuído pelo componente pai - como um vetor de variáveis do tipo insert, ou seja, um vetor de variáveis - cada qual representa uma categoria distinta, sendo a primeira responsável pelas pastilhas de torneamento externo e a segunda por pastilhas de torneamento interno.

Variáveis como os vetores de nome elements, conditions e materials são utilizadas para exprimir as opções de seleção dispostas nas iterações - realizadas dados os próprios vetores citados - para os elementos de seleção de cada estrutura mat-step do componente - assim, manifestando-as e designando-as à estrutura lógica dada a uma base de dados provida pela aplicação, tornando mais fácil a implementação de outras opções e continuidade da lógica da aplicação dadas essas implementações futuras.

Figura 3.7 - Declaração e inicialização de variáveis relevantes para a aplicação

```

114  /*
115   The inserts for external machining
116   inputed by the main component
117   */
118   @Input() externalInsertData!:insert[];
119
120  /*
121   The inserts for internal machining
122   inputed by the main component
123   */
124   @Input() internalInsertData!:insert[];
125
126   //available elements
127   elements:string[] = [
128     'Eixo',
129     'Flange',
130   ];
131
132   //available conditions for the element before it treatment
133   conditions:string[] = [
134     'Boa',
135     'Média',
136     'Difícil',
137   ];
138
139   //available materials for the element
140   materials:string[] = [
141     'Aço',
142     'Aço Inoxidável',
143     'Ferro Fundido',
144     'Superliga',
145   ];
146

```

Fonte: Autor

3.2.4.3 Funções de Geometria

As geometrias externas e internas dispostas nos passos do aplicativo aceitam uma quantidade não determinada de geometrias - variando de acordo com a vontade do usuário - e, dada a sua interação, a remoção delas em um determinada escopo - seja eliminando apenas uma das geometrias expressas ou o grupo como um todo, reiniciando os dados armazenados de uma categoria de geometria - além da validação das geometrias definidas, uma vez que essa se faz necessária para impedir que o usuário dê continuidade à aplicação com medidas inválidas.

Essas funções estão dispostas como:

Add New Geometry (): função com um parâmetro do tipo string que retorna um número, sendo responsável por adicionar as geometrias que o usuário preenche na interface referenciado ao objeto process_config contido na mesma classe exportada, assim adicionando

aos respectivos vetores geometrias vazias com o tipo correspondente a cada categoria - seja interna ou externa -, permitindo que o usuário às preencha na entrada de dados disponível. Como, na interface, as opções de edição das geometrias são iteradas com base nos elementos contidos nos seus respectivos vetores, adicionar elementos sem valores atribuídos permite que o usuário os edite e, uma vez que a iteração também determina como a interpolação de dados será efetuada, cada mudança será refletida na estrutura lógica da aplicação de imediato. O parâmetro serve para definir qual será a categoria acrescentada - interna ou externa - ao mesmo passo que o número retornado expressa um resultado lógico obtido pela aplicação - 0 no caso do argumento passado não ser válido, e 1 caso tenha sido válido e a mudança obtenha êxito.

Figura 3.7 - Função addNewGeometry() declarada

```

154  addNewGeometry(type: string):number {
155      if (type === "external")
156      {
157          this.process_config.productsExternalGeometry?.push({start: 0, end: 0, diameter: 0, type: "externo"});
158          return 1;
159      }
160      else if (type === "internal")
161      {
162          this.process_config.productsInternalGeometry?.push({start: 0, end: 0, diameter: 0, type: "interno"});
163          return 1;
164      }
165      else
166      {
167          return 0
168      }
169  }
170

```

Fonte: Autor

Reset Geometry (): função com um parâmetro do tipo string que retorna um número, possui o propósito de reiniciar as geometrias - assim removendo todas e limpando seus respectivos vetores - dado o tipo passado como argumento, de tal forma que a chamada resetGeometry('external') irá igualar ao vetor de geometrias internas do objeto process_config contido na classe a um vetor vazio - também removendo os elementos da interface em virtude da interpolação de dados. Da mesma forma que a função anterior, há o valor 1 retornado caso algum vetor tenha sido reiniciado ou 0 caso não haja um argumento válido - diferente de 'internal' e 'external'.

Figura 3.8 - Função resetGeometry() declarada

```

176 resetGeometry(type:string):number {
177
178     if (type === "external")
179     {
180         this.process_config.productsExternalGeometry?.splice(0, this.process_config.productsExternalGeometry.length);
181         return 1;
182     }
183     else if (type === "internal")
184     {
185         this.process_config.productsInternalGeometry?.splice(0, this.process_config.productsInternalGeometry.length);
186         return 1;
187     }
188     else
189     {
190         return 0;
191     }
192 }
193

```

Fonte: Autor

Remove Geometry (): função com dois parâmetros, um do tipo string e outro do tipo number, cada qual utilizado para obter a categoria desejada - seja interna ou externa - e índice desejado, respectivamente. Nesse caso há o intuito de remover uma geometria específica, ou seja, um elemento do vetor de geometrias contida no objeto process_config, o que por sua vez requer um índice específico - uma vez que o método splice() utilizado para remover um determinado trecho de um vetor receberá o índice obtido como um dos argumentos -, e, de praxe, um tipo para delegar qual elemento do objeto deverá ter uma de suas determinadas geometrias removida.

Figura 3.9 - Função removeGeometry() declarada

```

200 removeGeometry(type:string, idx: number):number {
201
202     if (type === "external")
203     {
204         this.process_config.productsExternalGeometry?.splice(idx, 1);
205         return 1;
206     }
207     else if (type === "internal")
208     {
209         this.process_config.productsInternalGeometry?.splice(idx, 1);
210         return 1;
211     }
212     else
213     {
214         return 0;
215     }
216 }
217

```

Fonte: Autor

Check Geometries (): função com um parâmetro de notação ‘...geometries:geometry[][]’ (denominada de spread syntax), recebendo um número não determinado a priori de vetores na forma de um vetor, cujo conteúdo expressa-se em vetores de geometrias - de tal forma que possa receber como argumento todas as categorias - tais como interno e externo - e as manejar dadas determinadas condições. Seu propósito é o de analisar os valores entrados pelo usuário e determinar a sua validade considerando algumas possibilidades dispostas nas condições presentes em cada iteração - ocorrendo uma para cada vetor inserido na matriz, e uma subsequente para os vetores de geometrias inseridas. Cada iteração responsabiliza-se por interpolar os valores dadas as referentes condições, de tal forma que valores lógicos sejam atribuídos aos vetores booleanArr e booleanArr0 em virtude dos vetores de geometrias e geometrias que correspondem às especificações, respectivamente.

Figura 3.10 - Função checkGeometries() declarada

```

223 geometriesValid:[string[], boolean] = [[], false];
224
225 checkGeometries(...geometries:geometry[ ][ ]):number {
226     const booleanArr:boolean[] = [ ];
227     const errorMsgs:string[] = [ ];
228
229     geometries.forEach((geArr) => {
230         const booleanArr0:boolean[] = [ ];
231
232         geArr.forEach((ge, idx) => {
233             if
234             (
235                 ge.start >= 0 && ge.end > 0 &&
236                 ge.start < ge.end && ge.diameter >= 0 &&
237                 ge.diameter <= this.process_config.preDiameter &&
238                 ge.end <= this.process_config.preLength &&
239                 (!geArr[idx - 1] || ge.start >= geArr[idx-1].end) &&
240                 (ge.start != 0 || ge.end != 0 || ge.diameter != 0)
241             )
242             {
243                 booleanArr0.push(true);
244             }
245         }

```

Fonte: Autor

O comprimento inicial não deve ser maior que o final e, esse, nunca poderá exceder o comprimento pré-usinagem do material - da mesma forma seu diâmetro almejado não pode ser maior do que o presente a priori, visto que é ilógico ao se referir a um processo de remoção de material -, o início de um trecho não pode maior do que o final de outro - ambos

pertencendo a uma mesma categoria - e entre outros. Algumas particularidades são consideradas na estrutura lógica composta na função, como o caso de um valor não inserido pelo usuário, alguma categoria estar vazia e outra possuir elementos - visto que não é ilógico realizar torneamento externo contudo não efetuar uma furação em uma peça -, ou todas as categorias estarem vazias, e et cetera.

Uma variável denominada de geometriesValid do tipo lógico possui um valor diferente atribuído sempre que ocorre uma chamada dessa função e, como sua chamada ocorre a cada seleção de passo na estrutura mat-stepper, atualiza constantemente a variável quanto a validade dos valores inseridos pelo usuário. Retorna um valor número para condizer com equivalências lógicas - denotando o resultado da aferição, como verdadeiro ou falso, por exemplo, e a mensagem de erro correspondente à condição que tornou a validação falsa.

Figura 3.11 - Verificação da falha na validação e atribuição do erro às mensagens

```

else
{
  if (ge.start == 0 && ge.end == 0 && ge.diameter == 0)
  {
    errorMsgs.push(`Em ${ge.type} o trecho ${idx} está vazio.`)
  }
  else if (geArr.length != 0)
  {
    if (ge.start <= 0) {
      errorMsgs.push(`Em ${ge.type} o início ${ge.start}mm é menor ou igual a 0.`);
    }
    if (ge.end <= 0) {
      errorMsgs.push(`Em ${ge.type} o final ${ge.end}mm é menor ou igual a 0.`);
    }
    if (ge.start > ge.end) {
      errorMsgs.push(`Em ${ge.type} o início ${ge.start}mm é maior que o final ${ge.end}mm.`);
    }
    if (ge.diameter > this.process_config.preDiameter) {
      errorMsgs.push(`Em ${ge.type} o diâmetro ${ge.diameter}mm é maior que o diâmetro pré-usinagem ${this.process_config.preDiameter}mm.`);
    }
    if (ge.end > this.process_config.preLength) {
      errorMsgs.push(`Em ${ge.type} o final ${ge?.end}mm é maior que o comprimento pré-usinagem ${this.process_config.preLength}mm.`);
    }
    if (geArr[idx-1]) {
      if (ge.start < geArr[idx-1].end) {
        errorMsgs.push(`Em ${ge.type} o início ${ge.start}mm é maior que o final ${ge.end}mm de seu trecho antecessor.`);
      }
    }
  }
  booleanArr0.push(false);
}

```

Fonte: Autor

A função também é responsável por chamar `checkDiameters()` com o propósito de verificar os diâmetros inseridos e, dados os erros encontrados, os armazenar e atribuir à variável de verificação implementada no componente.

`Check Diameters ()`: função com dois parâmetros de tipo referenciado às interfaces de geometrias exportadas no componente, sendo cada qual um vetor de geometrias condizente a cada categoria - dispostos como os parâmetros `external` e `internal`, respectivamente. Dada a possibilidade de haver dados incongruentes referentes ao diâmetro do material - como um diâmetro externo que seja inferior a um interno, por exemplo - a função retorna um valor lógico com o propósito de verificar a sua validade.

Figura 3.12 - Função `checkDiameters()` declarada

```

356 checkDiameters(external:geometry[], internal:geometry[]):boolean {
357   let valid:boolean = true;
358   const errMsgs:string[] = [
359
360   ]
361
362   internal.forEach((geoInt) => {
363
364     external.filter((geoExt) => {
365
366       return (geoExt.start <= geoInt.start && geoExt.end > geoInt.start)
367       || (geoExt.end > geoInt.start && geoExt.start < geoInt.end);
368
369     }).forEach((geo) => {
370
371       if (geo.diameter <= geoInt.diameter) {
372         errMsgs.push(
373           `O diâmetro ${geo.diameter}mm do trecho ${geo.start}mm - ${geo.end}mm
374           do ${geo.type} é menor ou igual ao diâmetro ${geoInt.diameter}
375           do trecho ${geoInt.start}mm - ${geoInt.end}mm do ${geoInt.type}`);
376         valid = false;
377       }
378
379     })
380   })
381   errMsgs.forEach((err) => {
382     this.geometriesValid[0].push(err);
383   });
384   return valid;
385 }
386

```

Fonte: Autor

3.2.4.4 Funções de parâmetros e pastilhas

As configurações das pastilhas armazenadas na aplicação precisam ser filtradas e, com base nas pastilhas necessárias dadas as condições estabelecidas, serem devidamente enlaçadas aos parâmetros relativos às dimensões inseridas pelo usuário. Entre as funcionalidades presentes constata-se filtrar os vetores, atribuir parâmetros às configurações do processo, preencher os parâmetros e os remover das geometrias respectivas.

Insert Filter (): possui dois parâmetros distintos, o primeiro referente a um vetor de pastilhas e o segundo um objeto de processo do tipo process. Inicialmente variáveis são criadas em seu contexto de execução com o propósito de atribuir o valor do material e condição presente nas configurações do processo a fim de adequar as strings contidas no processo às das configurações de pastilha - de tal forma que possa interpretar os valores atribuídos às seleções presentes na estrutura mat-stepper dos materiais, delegados como aço, aço inoxidável, ferro fundido e superliga, por exemplo, como p, m, k e s, respectivamente. Após tratar desses dados, a função utiliza o método filter() para obter um vetor de pastilhas dadas as condições e materiais especificados pelo usuário - retornando-o na chamada da função.

Figura 3.13 - Função insert Filter() declarada

```

392 insertFilter(insertArr:insert[], processObj: process):insert[] {
393
394     /*
395     Converts the material and condition obtained in the process obj to
396     ones that matches the inserts data
397     */
398     const material = processObj.material;
399     const condition = processObj.condition;
400     let materialN = '';
401     let conditionN = '';
402
403     materialN = material === 'aço' ? 'p' :
404     material === 'aço inoxidável' ? 'm' :
405     material === 'ferro fundido' ? 'k' :
406     material === 'superliga' ? 's' : '';
407
408     conditionN = condition === 'boa' ? 'boa' :
409     condition === 'média' ? 'media' :
410     condition === 'difícil' ? 'difícil' : '';
411
412     /*
413     Filters the insertArr received as argument accordingly to the
414     Material and Condition, declaring and initializing the
415     insertFiltered const with it's insert[]
416     */
417     const insertFiltered:insert[] = insertArr.filter((el) => {
418         return (
419             el.material.toLowerCase() === materialN.toLowerCase()
420             &&
421             el.condition.toLowerCase() === conditionN.toLowerCase()
422         );
423     });
424
425     return insertFiltered;
426 }
427

```

Fonte: Autor

Insert Calculator (): em sua estrutura há 3 parâmetros, sendo estes vetor de pastilhas, objeto de projeto e geometria, sendo a função responsável por calcular os parâmetros de usinagem pertencentes a cada pastilha e razão das dimensões designadas na interface - obtida na classe exportada por intermédio da interpolação de dados das entradas e seleções de dado - e o retorna no final de sua função na forma de um vetor de parâmetros.

Figura 3.14 - Função insert Calculator() declarada

```

434 insertCalculator(insertArr:insert[], processObj: process, geometry: geometry):parameters[] {
435
436     //parameters[] that will have all the parameters
437     const parameters:parameters[] = [
438
439     ];
440
441     /*
442     For each insert in the insert arr the parameters will receive a object of type parameter
443     with rpm, machiningTime, metalRemotionRate matched with each insert of the respective
444     insert index
445     */
446     insertArr.forEach((el) => {
447         const tempRpm:number = (1000 * el.vc)/(processObj.preDiameter * Math.PI);
448         const tempMachiningTime:number = (geometry.end - geometry.start)/(el.fn * tempRpm);
449         const tempMetalRemotionRate:number = el.fn * el.vc * el.ap;
450
451         parameters.push({rpm: tempRpm, machiningTime: tempMachiningTime, metalRemotionRate: tempMetalRemotionRate});
452     });
453
454     return parameters;
455 }
456

```

Fonte: Autor

Insert Filtrate Then Calculate (): utiliza a chamada das funções insertFilter() e insertCalculator() em conjunto com os parâmetros obtidos, sendo esses um vetor de pastilhas, objeto contendo configurações de processo e uma geometria. A princípio atribui o valor retornado pelas respectivas funções em seu contexto de execução e itera ambas utilizando de seus respectivos índices para, a posteriori, retornar uma matriz cujos vetores contém pastilhas e parâmetros no índice 0 e 1, respectivamente.

Figura 3.15 - Função insertFiltrateThenCalculate() declarada.

```

461 insertFiltrateThenCalculate(insert:insert[], processObj: process, geometry: geometry):[insert, parameters][] {
462
463     //One receives the filtered inserts and the other a array of parameters
464     const filtratedInserts:insert[] = this.insertFilter(insert, processObj);
465     const parameters:parameters[] = this.insertCalculator(filtratedInserts, processObj, geometry);
466
467     //variable that will contain the returned data
468     const insertsAndParameters:[insert, parameters][] = [
469
470     ];
471
472     //pushing the elements based on the actual index
473     filtratedInserts.forEach((el, idx) => {
474         insertsAndParameters.push([el, parameters[idx]]);
475     });
476
477     return insertsAndParameters;
478 }
479

```

Fonte: Autor

Fill Geometries Inserts And Parameters (): função sem retorno que utiliza da função insertFiltrateThenCalculate() em conjunto com o método map() a fim de iterar cada vetor de geometrias pertencentes às configurações de processo atribuindo o valor obtido através do cálculo dos parâmetros e filtro das pastilhas para cada geometria condizente. A função realiza uma chamada de função para cada vetor de geometrias a fim de preencher tanto as categorias internas e externas do processo com o passar do processamento das informações e passar de etapas por parte do usuário.

Figura 3.16 - Função fillGeometriesInsertsAndParameters() declarada

```

485     fillGeometriesInsertsAndParameters():void {
486
487         /*
488         The external geometry will receive it's data after
489         the externalInsertData is filtrated, and the filtrated
490         data is used altogether with the geometry to determine
491         the paramaters of machining in each case
492         */
493         this.process_config.productsExternalGeometry.map((geometry) => {
494             geometry.insertsAndParameters = this.insertFiltrateThenCalculate(
495                 this.externalInsertData, this.process_config, geometry);
496         });
497
498         /*
499         The internal geometry will receive it's data after
500         the internalInsertData is filtrated, and the filtrated
501         data is used altogether with the geometry to determine
502         the paramaters of machining in each case
503         */
504         this.process_config.productsInternalGeometry.map((geometry) => {
505             geometry.insertsAndParameters = this.insertFiltrateThenCalculate(
506                 this.internalInsertData, this.process_config, geometry);
507         });
508     }

```

Fonte: Autor

Unfill Geometries Inserts And Parameters (): função sem retorno cujo propósito encontra-se diametralmente oposto em relação à função fillGeometriesInsertsAndParamaters(), no que itera cada vetor de geometrias a fim de atribuir o elemento condizente às pastilhas e parâmetros como indefinido. Particularmente útil para alterar os dados sempre que ocorre a alteração dos dados por parte do usuário.

Figura 3.17 - Função unfillGeomtriesInsertsAndParameters() declarada

```

517 unfillGeomtriesInsertsAndParameters():void {
518     this.process_config.productsExternalGeometry.map((geometry) => {
519         geometry.insertsAndParameters = undefined;
520     });
521
522     this.process_config.productsInternalGeometry.map((geometry) => {
523         geometry.insertsAndParameters = undefined;
524     });
525 }
526

```

Fonte: Autor

Selection Change (): função chamada por intermédio do evento de mesmo nome presente na estrutura mat-stepper, sendo convocada a todo momento que ocorre a seleção de um passo distinto do atual. Em sua estrutura lógica ocorre o esvaziamento das mensagens de erro armazenadas pela função checkGeometries() e checkDiameters() no interior da variável de validade das geometrias, ambas as funções unfillGeomtriesInsertsAndParameters() e fillGeomtriesInsertsAndParameters() em sequência com a finalidade de atualizar as informações e, por fim, checkGeometris() com o propósito de alterar o valor da variável geometriesValid com base nas dimensões inseridas pelo usuário - assim verificando se alguma medida não possui congruência verificada uma vez alterada pelo usuário.

Figura 3.18 - Função selectionChange() declarada

```

532 selectionChange() {
533     this.geometriesValid[0].splice(0, this.geometriesValid[0].length);
534     this.tabDataCreator(this.process_config.productsExternalGeometry, this.process_config.productsInternalGeometry);
535     this.unfillGeomtriesInsertsAndParameters();
536     this.fillGeomtriesInsertsAndParameters();
537     this.checkGeometries(this.process_config.productsExternalGeometry, this.process_config.productsInternalGeometry);
538 }
539

```

Fonte: Autor

3.2.4.5 Convergir dados

Um dos componentes filhos requer uma quantidade válida de categorias a fim de criar abas de informações contidas no último passo da aplicação, dessa forma, parte da estrutura lógica do componente foi designado para validar as possíveis categorias e as filtrar com base em critério para identificar a sua validade - tal como a quantidade de elementos em cada vetor de geometrias, caso o elemento de pastilhas e parâmetros esteja definido e a aparição de aferições equivalentes a 0 no final dos trechos designados para cada geometria.

Tab Data Creator(): possui a denominada “spread syntax” para aceitar um número não determinado de argumentos a fim de os filtrar e receber uma matriz com vetores de geometrias condizentes com as condições estabelecidas, retornando a matriz equivalente. Particularmente útil para atribuir à variável tabData as categorias válidas presentes na aplicação dadas as decisões tomadas pelo usuário ao longo dos passos contidos no componente.

Figura 3.19 - Variável tabData declarada e inicializada ao mesmo passo que a função tabDataCreator() é declarada

```

547     tabData:geometry[][] = [
548
549     ];
550
551     tabDataCreator(...geometriesArr: geometry[][]):geometry[][] {
552         const filteredArr = geometriesArr.filter((el) => {
553             return el.filter((el) => {
554                 el.end !== 0 && el.insertsAndParameters
555             }) && el.length;
556         })
557         this.tabData = filteredArr;
558         return filteredArr;
559     };
560
561 }
562

```

Fonte: Autor

3.2.4.6 Estrutura

No que toca a estrutura do componente, sua linguagem de marcação é configurada e organizada quase que integralmente no interior de uma estrutura mat-stepper com propriedades delegadas a fim de estabelecer a orientação por intermédio de um observador

síncrono, sua linearidade - necessidade de alcançar passos antecessores antes de progredir -, identificação e chamada de função condizente ao evento de alteração na seleção atual - que, por sua vez, volta-se para a função `selectionChange()` citada a priori.

Figura 3.20 - Atributos e eventos designados à estrutura `mat-stepper`

```

12     <mat-stepper
13         #stepper
14         [linear] = 'true'
15         (selectionChange)="selectionChange()"
16         [orientation]="(stepperOrientation | async)!"
17     >

```

Fonte: Autor

Cada estrutura `mat-step` organiza determinadas entradas de dado, dispondo desse propósito nos quatro primeiros passos da aplicação - cada qual catalogado dados rótulos determinados em uma estrutura intrínseca -, dispondo das estruturas `mat-select` e entradas de dado com a designação `matInput` - a fim de determinar propriedades comportamentais e estilização - em conjunto com interpolação de dados entre as suas propriedades e variáveis contidas na classe exportada da estrutura lógica - de tal forma que a interface se comunique em tempo real com a porção operacional do aplicativo, tratando das informações obtidas e as exprimindo para o usuário logo em seguida.

Nos passos condizentes com a entrada dos trechos de geometrias há o uso recorrente do bloco `@for` para iterar vetores presentes na aplicação a fim de, com dinâmica e articulação, manifestar os trechos e seus respectivos valores ao usuário. Também nestes estão contidos botões com a finalidade de remover os trechos, adicionar e reiniciá-los por completo, cujos eventos de clique estão enlaçados às funções exportadas pela estrutura lógica - cuja classe exporta para outros componentes tanto o template presente, quanto a sua estrutura e folha de estilo intrínseca.

A utilização do bloco `@if` está presente com o propósito de selecionar as condições relacionadas a aparição de cada elemento contido no componente, de tal forma que os passos denotam blocos diferentes dados os dados obtidos e, no caso do último passo, articulação de mensagens de erro e painéis de categoria em virtude da validade dos dados obtidos.

Na última etapa, denominada de “Obtenção de parâmetros”. há o componente filho `app-panel` cuja propriedade denotada como `tabData` recebe as categorias e intrínsecas

informações manejadas pela estrutura lógica do componente pai, integrando as suas funcionalidades continuamente em função das mudanças realizadas pelo usuário - não limitando, dessa forma, a implementação de uma determinada inserção de dados em razão de alterações menores ou totais das intenções de seu utilizador.

Caso haja qualquer dado com validade incoerente um vetor de mensagens com erros será iterado baseado nas informações e suas incongruências respectivas - assim apontado para o usuário quais valores não são aceitos pelo aplicativo.

Sua folha de estilo designa quais especificações os campos de formulário - locais nos quais será inseridas as entradas de dado -, disposição e espaçamento da estrutura dos botões presentes no passos, disposição das entradas de geometrias, comportamento do texto dos títulos e conteúdos, estética atribuída às mensagens de erro e comportamento estético dos botões de remover geometrias.

Em razão do tamanho pertencente ao componente ocorre também maior empreendimento de especificações presentes no arquivo de formato CSS a fim de designar a quantidade mais acrescida de componentes - uma vez que os tornar atrativos para o usuário também é um fato de determinação para o uso da aplicação. Consoante ao constatado, animações de acréscimo no tamanho das mensagens de erro e botões em formato “x” são implementadas, tal como o alinhamento de texto e tamanho de fonte do conteúdo de alguns elementos - como é o caso dos títulos de cada geometria.

3.2.5 Tabs:

Componente responsável por separar dados e elementos de interface correspondentes e uma ou mais abas selecionáveis pelo usuário a fim de separar informações e permitir trânsito dinâmico e acessível das informações. Implementado no componente Stepper da aplicação com seu atributo `tabData` - utilizado no interior de seu template para dispor e organizar as informações - orientado aos vetores de geometrias uma vez processados pela estrutura lógica do componente pai, respondendo diretamente às informações entregues pelo utilizador. No interior de cada aba serão iteradas as geometrias e dispostas em componentes filhos `app-panel` - recebendo os dados de pastilha e parâmetros de cada categoria, a exprimindo a posteriori.

3.2.5.1 Estrutura

No interior de sua estrutura lógica há a importação do módulo Mat Tabs Module - encarregado de prover a estrutura mat-tab-group e seus elementos filhos mat-tab que, no caso deste componente, se encarrega de cada categoria - tal como torneamento externo e interno.

Também ocorre a importação do Painel Component, utilizado em seu template; interface de geometrias e a diretiva Input que, em sua estrutura lógica, permite a importação da variável tabData declarada como uma matriz de geometrias, inicializada dada a propriedade atribuída ao componente.

Em seu template ocorre a disposição de dois blocos @for, sendo o primeiro empregado para iterar cada categoria da variável tabData e dispor as abas com número e rotulação condizente, onde ocorrerá a iteração das geometrias pertencentes a cada geometria, exprimindo uma instância do componente app-panel com atribuição correspondente à geometria respectiva.

Figura 3.21 - Template do componente app-tabs

```

1  <p>
2      <!--
3          Each category will be displayed as a tab that, within, will have a app-panel component
4          for each geometry, displaying info like the parameters of the machining with each insert
5          and it's configurations - like vc, fn and ap, for example
6      -->
7      <mat-tab-group>
8          <!-- iterates each category inputed in the data -->
9          @for (category of tabData; track tabData; let idx = $index) {
10              <mat-tab>
11                  <!--
12                      uses the type of the first geometry of the category - that will be the same as the others - to
13                      label the tab - as external or internal, for example
14                  -->
15                  <ng-template mat-tab-label>
16                      {{category[0].type}}
17                  </ng-template>
18
19                  <!--
20                      iterates each geometry, passing it to a children app-panel component
21                      empowered by the mat expansion module of Angular Material
22                  -->
23                  @for (geometry of category;
24                      track tabData) {
25                      <app-panel [geometry] = "geometry"></app-panel>
26                  }
27              </mat-tab>
28          }
29      </mat-tab-group>
30  </p>

```

Fonte: Autor

3.2.6 Panel:

Caracteriza-se como o componente que comporta os parâmetros de usinagem e configuração das pastilhas para cada geometria preenchida pelo usuário, possuindo integração e interpolação dos dados obtidos pelos componentes de hierarquia antecessora - tais como o componente app-tabs e app-stepper, contidos na porção principal da aplicação.

3.2.6.1 Estrutura

Em sua estrutura lógica é importado o módulo Mat Expansion Module - provendo as estruturas mat-expansion-panel, que servirão para denotar as informações no formato de trecho e informações expansíveis para dado o interesse atual do utilizador - implementando um método minimalista, contudo ainda sim completo - dado o foco atual do usuário -, de dispor as informações.

Também há a importação da interface geometria e o componente Mat Divider com as finalidades de declarar a variável imputada pelo componente que o contém e separar os seus conteúdos intrínsecos, respectivamente.

Em seu template a estrutura mat-expansion-panel é denotado com título equivalente às dimensões pertencentes ao trecho da geometria recebida e, em seu conteúdo - expansível dado o clique do usuário - iteração de cada parâmetro e pastilhas referentes a geometria adotada, assim exprimindo uma lista com imagem e informações condizentes a cada ferramenta em razão da operação adotada atualmente.

Figura 3.22 - Início do Template do componente app-panel com sua primeira iteração

```

1  <p>
7  <mat-expansion-panel hideToggle>
8      <!-- Header -->
9      <mat-expansion-panel-header>
10         <!--
11             The title displays the length and diameter
12             inputed in the desired geometry
13         -->
14         <mat-panel-title
15             class="panel-title"
16         >
17             {{geometry.start}}mm - {{geometry.end}}mm ({{geometry.diameter}}mm)
18         </mat-panel-title>
19     </mat-expansion-panel-header>
20
21     <!--
22         bellow the header is the main content, displayed when the user clicks it
23         the div of class iteration will have each insert and parameter and the styles
24         for the main container of them
25     -->
26
27     <div
28         class="iteration"
29     >
30         <!--
31             iterates each insert and parameter from the geometry
32             the values displayed receive are
33             outputed with the .toFixed() method
34             to define the decimal places
35         -->
36         @for (ins_pam of geometry.insertsAndParameters;
37             track geometry.insertsAndParameters) {
38
39             <!-- It will display the number of the iteration -->
40
41             <mat-divider></mat-divider>
42
43             <h3>{{$index + 1}}.</h3>
44
45             <div
46                 class="out-tracer"
47             >

```

Fonte: Autor

A configuração de seus elementos se dá por intermédio de classes distribuídas ao longo do conteúdo com o propósito de agrupar os dados e os organizar, alinhando-os em colunas com o padrão de parâmetro e valor correspondente.

Na folha de estilo são definidas todas as especificações que configuram a disposição e posicionamento adotado atualmente pelos conteúdos e respectivas estruturas de organização dispostas na interface de usuário, de tal forma que o título do painel atual tenha sua largura e disposição de conteúdo configurada em razão das classes atribuídas. Toda descrição dos dados recebe a classe data-description de tal forma que suas respectivas colunas são designadas pela

classe data-container, cada qual permitindo ou não a sua disposição horizontal - como não é o caso de container, que precede as classes data-container, mantendo os elementos designados em linha, por exemplo.

A classe img-container é atribuída aos espaços nas quais as imagens das pastilhas estarão posicionadas, centralizando-as e determinando a sua altura; em contrapartida a classe sucessora insertImg é responsável por definir a estética e dimensões das próprias imagens - como o seu alinhamento vertical, dimensões e tempo de transição para as mudanças dimensionais atribuídas com o passar dos mouses.

4. RESULTADOS

4.1 Interface

A aplicação possui instruções no formato de texto dispostas para que o usuário entenda a natureza do projeto, sua justificativa e aplicação, permitindo que com uma leitura o usuário possa interpretar e entender suas funcionalidades, tal como a utilidade de cada componente.

Além disso, a aplicação organiza o seu conteúdo com disposta linearidade configurada através dos catálogos de pastilhas e o componente principal localizado no seu final - disposto das etapas visíveis e navegáveis pelo usuário.

Apesar da responsividade da aplicação não permitir que os catálogos e componente de passos mantenham-se totalmente acessíveis sem a utilização de uma barra de rolamento em virtude da largura que excede a possível tela do usuário, nenhuma funcionalidade da aplicação falha em função disso ao realizar os testes na aba de inspeção do navegador - visto que ainda é possível obter as informações, entrar com os dados necessários e visualizar as informações exprimidas pela aplicação.

Figura 4.1 - Disposição das pastilhas no componente de catálogo para o usuário

Pastilhas para Torneamento Interno

Forma	Iso	Geometria	Classe	Vc	F _n	A _p	Material	Condição	Operação	Uso
	CCMT 06 02 02	PF	4325	495.00 m/min	0.06 mm/r	0.25 mm	p	Boa	Acabamento	Coroturn 1078
	CCMT 09 T3 02	PF	4325	485.00 m/min	0.08 mm/r	0.25 mm	p	Boa	Acabamento	Coroturn 1078
	CCMT 09 T3 08	PF	4315	530.00 m/min	0.15 mm/r	0.40 mm	p	Boa	Acabamento	Coroturn 1078
	DCMT 07 02 02	PF	4325	495.00 m/min	0.06 mm/r	0.25 mm	p	Boa	Acabamento	Coroturn 1078
	DCMT 11 T3 02	PF	4325	485.00 m/min	0.08 mm/r	0.25 mm	p	Boa	Acabamento	Coroturn 1078
	DCMT 11 T3 08	PF	4315	530.00 m/min	0.15 mm/r	0.40 mm	p	Boa	Acabamento	Coroturn 1078
	TCMT 09 02 02	PF	4325	495.00 m/min	0.06 mm/r	0.25 mm	p	Boa	Acabamento	Coroturn 1078
	TCMT 11 03 02	PF	4325	490.00 m/min	0.07 mm/r	0.25 mm	p	Boa	Acabamento	Coroturn 1078
	TCMT 11 03 08	PF	4315	550.00 m/min	0.13 mm/r	0.40 mm	p	Boa	Acabamento	Coroturn 1078
	CCMT 06 02 08	PM	4315	560.00 m/min	0.11 mm/r	1.20 mm	p	Boa	Acabamento	Coroturn 1078

Items per page: 10 1 - 10 of 89 < > >>

Fonte: Autor

4.2 Aplicabilidade

Ao entrar com especificações de material e condição é evidente que as pastilhas expressas na etapa de obtenção possuem configurações condizentes, da mesma forma que seus valores de velocidade de fuso, taxa de remoção de metal e tempo de usinagem são congruentes com os valores recebidos - variando a depender das dimensões que antecedem a usinagem e pastilhas adequadas dadas as especificações estabelecidas.

As informações mostradas na última etapa são alteradas em tempo real, permitindo que o usuário realize alterações livremente sem ter que reiniciar a aplicação ou dispor das geometrias e condições estabelecidas anteriormente.

Figura 4.2 - Parâmetros de usinagem e especificações das pastilha

Aplicação

Por conseguinte, a obtenção dos parâmetros de usinagem correspondentes ao contexto se dará por intermédio de um componente de passo a passo localizado abaixo, cujo propósito fundamenta-se na obtenção das condições iniciais e finais do produto antecedendo e sucedendo o seu processo de usinagem.

1 Inserir a peça
2 Estado pré-usinagem
3 Geometria Externa do produto
4 Geometria Interna do produto
5 Obtenção de parâmetros

externo

0mm - 50mm (50mm)

50mm - 100mm (70mm)

Iso: CNMG 12 04 04 **Velocidade de corte:** 440.00 m/min **Avanço:** 0.20 mm/r **Condição:** Média **Material:** p **RPM:** 1400.56 r/min **Tempo de Usinagem:** 0.18 min

Geometria: PF **Classe:** 4325 **Profundidade de corte:** 0.40 mm **Uso:** T-Max® **Taxa de Remoção:** 35.20 cm³/min **Operação:** Acabamento

2.

Iso: DNMG 15 06 04 **Velocidade de corte:** 440.00 m/min **Avanço:** 0.15 mm/r **Condição:** Média **Material:** p **RPM:** 1400.56 r/min **Tempo de Usinagem:** 0.24 min

Fonte: Autor

4.3 Aplicação

O código fonte completo da aplicação e a própria em funcionamento podem ser encontrados nos endereços: <https://github.com/samuel-pinho-c-lobes/iusinagem-v1> e <https://iusinagem-v1.web.app/>, respectivamente.

5. DISCUSSÃO DOS RESULTADOS

Em suma, a aplicação cumpriu com os critérios de existência - no que foi capaz de prover dados de utilidade para a área de atuação e o manejar com base nas especificações concedidas pelo usuário de maneira intuitiva, rápida e com confiabilidade.

A aplicação atualmente se limita às pastilhas a priori armazenadas em seus componentes, de tal forma que não disponha da capacidade de adicionar novas ou editar as existentes - o que por sua vez configura uma fragilidade - além da ausência de certas operações presentes no ofício, como as ferramentas e geometrias para abertura de canais, opções relacionadas à quebra de cantos vivos, e outros. A aplicação não oferece total cobertura frente a completude das ocasiões e processos pertencentes a usinagem mecânica, o que delimita a sua utilidade aos dados já inseridos e operações pensadas a priori.

Não obstante, são desvantagens no que abrange a versão atual da aplicação, não sendo delimitadora de possíveis implementações futuras ou atualizações da mesma, ou seja, não configurando vulnerabilidades na implementação da informática no ofício, porém sim, sobretudo, modos de implementar mais funcionalidades e utilidades - assim fomentado a sua abrangência e capacidade de prestar seus serviços, auxiliando nos processos e automatizando etapas.

6. CONCLUSÃO

A implementação da informática na usinagem, tal como a sua posterior articulação, pode ser realizada por intermédio de tecnologias de rede que permitam a obtenção de dados estratégicos e cálculos com segurança, a fim de abstrair tarefas presentes no processo - assim crescendo a sua produtividade.

No caso de aplicações estáticas, é possível criar uma condição isolada de utilidade que conceda a um usuário a capacidade de se aproveitar dos seus recursos, ainda permitindo que implementações do tipo sejam realizadas e possuam êxito aquém ao favorecimento do processo e resolução de situações problemas - apesar de limitadas ao contexto estabelecido.

A indústria possui constantes inovações no âmbito da tecnologia e segue adicionando métodos de as articular, seja por meio da inserção da robótica, internet das coisas ou digitalização e importação de dados em função de uma lógica programada.

Portanto, a implementação e articulação da informática na usinagem é manifestada dada uma condição estabelecida pelo contexto hodierno e, toda tecnologia desenvolvida para tal, produto das necessidades contemporâneas em função dos desafios estabelecidos - como a competitividade e limitações operacional, por exemplo -, assim abrangendo e, gradativamente beneficiando - um número crescente de interesses pertencentes à usinagem mecânica.

6.1 Sugestões

Conforme constatado, a aplicação possui limitações no que toca a sua atuação - uma vez que depende de um único catálogo a priori armazenado nela e não admite alterações, exclusões ou adições -, e, sobretudo, possui muitas possibilidades aquém ao seu desenvolvimento posterior - em razão da quantidade exorbitante de utilidades possíveis e implementáveis.

Entre elas, constata-se algumas contempladas na forma de sugestões que, caso atendidas, irão corroborar substancialmente as suas funcionalidades e amplificar o nível de abrangência:

- Acrescentar desenhos equivalentes às etapas do processo de fabricação com as suas devidas geometrias cotadas. (Orientador).
- Aceitar edições, adições e remoções no que se refere ao catálogo de ferramentas utilizado pela aplicação. (Autor).
- Uma vez implementadas alterações no catálogo, possuir acesso e cadastro a fim de armazenar os catálogos agora personalizados para cada usuário - com a imprescindível criação de um respectivo banco de dados para os armazenar. (Autor).
- Manter os processos de fabricação realizados em um histórico ou repositório, com a finalidade dos usuários serem capazes de os acessar a posteriori sem ter que repetir todas as etapas de seleção e edição de geometrias. (Autor).

Em razão da quantidade incomensurável de possibilidades, torna-se possível realizar infinitas incrementações - dados os devidos recursos e capacidades em função das respectivas implementações - e, consoante ao exposto, tais atualizações não se limitam à resolução de

fragilidades ou limitações presentes, abrangendo também adaptações a fim de se adequar aos cenários inseridos e condições estabelecidas - ou seja, como toda tecnologia, caracteriza-se como passiva de quaisquer desenvolvimentos a fim de suprir necessidades humanas, condicionando-se aos requisitos contextuais.

7. REFERÊNCIAS BIBLIOGRÁFICAS

MACHADO, Álisson Rocha et al. Teoria da usinagem dos materiais. Editora Blucher, 2015.
Data de acesso: 15/01/2024

FERRARESI, Dino. Fundamentos da usinagem dos metais. Editora blucher, 2018. Data de acesso: 16/01/2024

DINIZ, Anselmo Eduardo; MARCONDES, Francisco Carlos; COPPINI, Nivaldo Lemos. Tecnologia da usinagem dos materiais. Artliber, 2014. Data de acesso: 16/01/2024

MACHADO, Álisson Rocha; SILVA, MB da. Usinagem dos metais. (Metal Cutting), Technical Notes, 9th version,[In Portuguese], 2004. Data de acesso: 17/01/2024

STOETERAU, R. L. Fundamentos dos processos de usinagem. São Paulo: USP, 2003. Data de acesso: 17/01/2024

FITZPATRICK, Michael. Introdução aos processos de usinagem: Série Tekne. Bookman Editora, 2013. Data de acesso: 21/01/2024

COSTA, E. S.; SANTOS, Denis Júnio. Disciplina: Processo de Usinagem. Divinópolis, MG, 2006. Data de acesso: 22/01/2024

COELHO, REGINALDO T.; DA SILVA, ERALDO JANONNE. INTRODUÇÃO AO PROCESSO DE FRESAMENTO. 2018. Data de acesso: 06/02/2024

DA SILVA, Sandro Oliveira; REIS FILHO, Ramílio Ramalho. TORNO CNC: evolução, características e importância para a usinagem. Revista Interface Tecnológica, v. 19, n. 1, p. 313-326, 2022. Data de acesso: 07/02/2024

FONSECA JR, Luiz Carlos; FONSECA, Marcio Roberto; DE LIMA RANGEL, Hélio Augusto. An applied study on Angular framework 2. Unisanta Science and Technology, v. 7, n. 1, 2018. Data de acesso: 21/02/2024

DE MATOS MACEDO, Mariano. Gestão da produtividade nas empresas. Revista Organização Sistêmica, v. 1, n. 1, 2012. Data de acesso: 12/03/2024

JAIN, Nilesh; MANGAL, Priyanka; MEHTA, Deepak. AngularJS: A modern MVC framework in JavaScript. Journal of Global Research in Computer Science, v. 5, n. 12, 2014. Data de acesso: 20/03/2024

GEETHA, G. et al. Interpretation and Analysis of Angular Framework. In: 2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS). IEEE, 2022. Data de acesso: 20/03/2024

GOOGLE. Angular, 2010. Site oficial da Google sobre a plataforma de desenvolvimento Angular e suas funcionalidades. Disponível em: <https://angular.io/guide/what-is-angular>. Data de acesso: 20/03/2024

MICROSOFT. TypeScript, 2012. Site oficial da Microsoft sobre a linguagem de programação TypeScript. Disponível em: <https://www.typescriptlang.org>. Data de acesso: 20/03/2024

CHANSUWATH, Wutthichai; SENIVONGSE, Twittie. A model-driven development of web applications using AngularJS framework. In: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). IEEE, 2016. Data de acesso: 21/03/2024

MILETTO, Evandro Manara; DE CASTRO BERTAGNOLLI, Silvia. Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP-Eixo: Informação e Comunicação-Série Tekne. Bookman Editora, 2014. Data de acesso: 21/03/2024

CONTE, Tayana; MENDES, Emília; TRAVASSOS, Guilherme Horta. Processos de desenvolvimento para aplicações web: Uma revisão sistemática. In: Proceedings of the 11th

Brazilian Symposium on Multimedia and Web (WebMedia 2005). 2005. Data de acesso: 21/03/2024

GOH, Hock-Ann; HO, Chin-Kuan; ABAS, Fazly Salleh. Front-end deep learning web apps development and deployment: a review. *Applied intelligence*, v. 53, n. 12, 2023. Data de acesso: 26/03/2024

W3SCHOOLS. Frontend, 2024. Site educacional voltado ao aprendizado de tecnologias web. Disponível em: <https://www.w3schools.com>. Data de acesso: 26/03/2024

CARVALHO, João Vidal; ABREU, António; ROCHA, Álvaro. *Informática Industrial. RISTI-Revista Ibérica de Sistemas e Tecnologias de Informação*, v. 27, p, 2018. Data de acesso: 27/03/2024

SACOMANO, José Benedito et al. *Indústria 4.0*. Editora Blucher, 2018. Data de acesso: 27/03/2024

SANTOS, Beatrice Paiva et al. *Indústria 4.0: desafios e oportunidades*. *Revista Produção e Desenvolvimento*, v. 4, n. 1, 2018. Data de acesso: 28/03/2024

