Faculdade de Tecnologia de Pindamonhangaba

APLICATIVO E SOFTWARE PARA CICLO TÉRMICO:

Heat Input e Equação de Adams

Matheus Felipe Andrade Gomes

Pindamonhangaba - SP
2024

Faculdade de Tecnologia de Pindamonhangaba

APLICATIVO E SOFTWARE PARA CICLO TÉRMICO:

Heat Input e Equação de Adams

Matheus Felipe Andrade Gomes

Monografia apresentada à Faculdade de Tecnologia de Pindamonhangaba para graduação no Curso Superior de Tecnologia em Mecânica: Processos de Soldagem

Orientador: Prof Dr Edilon de Oliveira França

Pindamonhangaba - SP
2024

G633 Gomes, Matheus Felipe Andrade.
Aplicativo e Software para Ciclo Térmico: Heat Input e Equação de Adams / Matheus Felipe Andrade Gomes / FATEC Pindamonhangaba, 2024.
40f.: il.

Orientador Professor Dr. Edilon de Oliveira França
Monografia (Graduação) – FATEC – Faculdade de Tecnologia de Pindamonhangaba. 2024

1. Heat Input. 2. Fórmula de Admas. 3. Java Script. 4. ReactNative. 5. Metalurgia da Soldagem. I. Gomes, Matheus Felipe Andrade. II. França, Edilon de Oliveira. III. Título.

CDD 621.4

Sueli Camargo de Souza – Bibliotecária FATEC Pindamonhangaba – CRB/8 7788

Faculdade de Tecnologia de Pindamonhangaba

"APLICATIVO E SOFTWARE PARA CICLO TÉRMICO: HEAT INPUT E EQUAÇÃO DE ADAMS".

Matheus Felipe Andrade Gomes

Monografia apresentada à Faculdade de Tecnologia de Pindamonhangaba, para graduação no Curso Superior de Tecnologia em Mecânica: Processos de Soldagem.

Comissão Examinadora

Orientador – Prof. Dr. Edilon de Oliveira França

Membro – Prof. Me. Marcelo Bergamini de Carvalho

Membro – Tecnólogo Rener Washington Mariano dos Santos

Pindamonhangaba, 04 de dezembro de 2024.

AGRADECIMENTO

MATHEUS

Agradeço, primeiramente, à minha família: ao meu pai André Gomes, à minha mãe Maria Sirlei Andrade Gomes e à minha irmã Bruna Laís de Andrade Gomes, pelo apoio incondicional e constante ao longo desta jornada. Sua paciência e incentivo foram fundamentais para que eu pudesse seguir adiante.

Ao professor Edilon de Oliveira França, pela orientação valiosa, paciência e sabedoria, que contribuíram de forma significativa para a concretização deste trabalho.

Por fim, agradeço ao meu amigo Igor de Araújo Alvarenga, pela amizade sempre presente e pelos conselhos oportunos que ajudaram diretamente no progresso deste projeto.

“Saber muito não significa ser inteligente; a inteligência não é só informação, mas também julgamento, a maneira pela qual uma informação é coordenada e utilizada.”

Carl Sagan

Matheus Felipe Andrade Gomes. 2024. 39p. **Aplicativo e Software para Ciclo Térmico: Heat Input e Equação de Adams.** Trabalho de Graduação (no Curso Superior de Tecnologia em Mecânica: Processos de Soldagem). Faculdade de Tecnologia de Pindamonhangaba. Pindamonhangaba. 2024.

RESUMO

Para se estudar metalurgia da soldagem, faz-se necessário conhecer os ciclos de aquecimento em cada camada de solda depositada e a energia de soldagem envolvida que é chamada de *Heat Input,* para o ciclo de aquecimento existe a equação de Adams. Este trabalho tem o objetivo de criar um aplicativo de celular que realiza o cálculo do ciclo térmico baseado nesta fórmula, onde foi utilizado as ferramentas de programação: *JavaScript, ReactNative*, Expo e outras bibliotecas, para cria o gráfico do ciclo térmico que varia da temperatura de fusão até a temperatura ambiente variando a distância em milímetro do centro da poça de fusão até o metal base que não sofreu aquecimento durante o processo de soldagem. Esse aplicativo pode mostrar em forma de tabela as variações da temperatura pela distância ou gerar um gráfico das mesmas condições.

Palavras-chave: *Heat Input*. Fórmula de Admas. *JavaScript*. *ReactNative*. Metalugia da soldagem.

Matheus Felipe Andrade Gomes. 2024. 39p. Application and Software for Thermal Cycle: Heat Input and Adams Equation. Undergraduate Work (in the Higher Course of Technology in Mechanics: Welding Processes). Pindamonhangaba Faculty of Technology. Pindamonhangaba. 2024.

ABSTRACT

To study welding metallurgy, it is necessary to know the heating cycles in each layer of weld deposited and the welding energy involved, which is called Heat Input, so that the heating cycle exists, an innovation by Adams. This work aims to create a cell phone application that carries out the thermal cycle project based on this formula, where programming tools were used: JavaScript, ReactNative, Expo and other libraries, to create the graph of the thermal cycle that varies in temperature melting point to room temperature, varying the distance in millimeters from the center of the melting pool to the metal base that did not undergo heating during the melting process. This application can show temperature variations over distance in table form or generate graphs with the same conditions.

Keywords: Thermal input. Admas formula. JavaScript. ReactNative. Welding metallurgy.

LISTA DE FIGURAS

Figura 1 Ciclo Térmico em soldagem 15
Figura 2 *Ciclo térmico em soldagem transformações no resfriamento.* 16
Figura 3 *Ilustração do metal base na relação tempo/temperatura.* 17
Figura 4 *Repartição térmica em soldagem* 17
Figura 5 *Repartição térmica em soldagem* 18
Figura 6 *Relação da microestrutura e a temperatura.* 18
Figura 7 *Relação da direção de viagem em cada material.* 19
Figura 8 Relação da Energia de soldagem com velocidade de resfriamento 21
Figura 9 Partes básicas de um algoritmo. 23
Figura 10 Programação utilizando a equação de Adams 28
Figura 11 Tela para gerar o gráfico 28
Figura 12 Código de Programação 29
Figura 13 Código ao campo referente ao cálculo do ciclo térmico .**Erro! Indicador não definido.**
Figura 14 valores para o cálculo do *Heat Input* 32
Figura 15 Gráfico com valores da velocidade de resfriamento calculados 34
Figura 16 valores velocidade de resfriamento em tabela 34

SUMÁRIO

1 INTRODUÇÃO 12
1.1 PROBLEMA 13
1.2 OBJETIVOS 13
1.2.1 Objetivo Geral *13*
1.2.2 Objetivos Específicos *14*
1.3 JUSTIFICATIVA 14
2 REVISÃO BIBLIOGRÁFICA 15
2.1 CICLO TÉRMICO 15
2.1.1 Condutividade e Difusividade Térmica *19*
2.1.2 Condutividade e Difusividade Térmica *20*
2.1.3 Equação de Adams *21*
2.2.1 Algoritmo *22*
2.2.2 Linguagem de Máquina e Assembler *23*
2.2.3 Linguagens de Programação *24*
2.2.4 Linguagem de Programação Javascript *25*
2.2.5 React Native *25*
2.2.5 APK *26*
3 MATERIAIS E MÉTODOS 28
4 RESULTADOS E DISCUSSÃO 32
5 CONSIDERAÇÕES FINAIS 37
REFERÊNCIAS 39

1 INTRODUÇÃO

A prática da soldagem é frequentemente conduzida por meio da aplicação localizada de calor e/ou deformação plástica, podendo acarretar modificações indesejáveis ou inaceitáveis nas propriedades do material na zona de junção. Essas alterações estão intrinsecamente vinculadas às reações que ocorrem durante os processos de solidificação e resfriamento do cordão de solda, exercendo influência sobre a microestrutura resultante. A compreensão desses fenômenos metalúrgicos é essencial em diversas aplicações de soldagem, entender o fluxo de calor permite um estudo mais eficiente do processo (MODENESI; MARQUE; SANTOS, 2024)

O fluxo de calor em soldagem pode ser simplificado em duas etapas principais: o fornecimento de calor à junta e a dissipação deste calor pela peça. Estas etapas, bem como uma explicação detalhada, serão abordadas posteriormente. Durante o processo de soldagem, o material experimenta uma variação de temperatura que depende, entre outros fatores, das condições de soldagem, das propriedades térmicas e da distância do ponto à fonte de calor. Esta variação de temperatura é conhecida como ciclo térmico de soldagem. (PAREDES, 2024)

Durante o processo de soldagem, o material experimenta uma variação de temperatura que depende entre fatores das condições de soldagem, propriedades térmicas e distância do ponto à fonte de calor, e esta variação de temperatura denomina-se ciclo térmico de soldagem. (INFOSOLDA, 2024)

Considerando esse cenário, a concepção de um aplicativo destinado a facilitar e enriquecer a compreensão desses fenômenos foi formulada como uma proposta. O desenvolvimento desse aplicativo demanda uma compreensão em Algoritmo e Programação. Inicialmente, serão abordados conceitos superficiais, com a análise de aspectos relevantes ao longo do processo para ampliar a compreensão do código. (LINTZMAYER; MOTA, 2020)

Em qualquer aplicação, seja um programa ou rotina executada pelo computador, a clareza na sequência de procedimentos para alcançar resultados é de suma importância. Esse encadeamento lógico na programação é denominado Lógica de Programação, enquanto a descrição de procedimentos é caracterizada como algoritmo. (ALURA, 2024)

A lógica de programação é essencial para guiar o comportamento do computador através do código que escrevemos. Entre as opções de linguagens como C, Java, Python e outras, optamos por JavaScript devido à sua natureza dinâmica, e ampla adoção na web. Sua flexibilidade permite uma rápida adaptação às demandas do projeto, além de sua capacidade de execução tanto no lado do cliente quanto do servidor.

Ao escolher o JavaScript, aproveita-se de uma vasta gama de frameworks(bibliotecas). Para este projeto, iremos utilizar o React Native, uma biblioteca JavaScript reconhecida por sua eficácia no desenvolvimento de aplicativos móveis. O diferencial do React Native está na criação de aplicativos nativos para iOS e Android a partir de um único código-base, proporcionando uma experiência de desenvolvimento eficiente e consistente em várias plataformas. (ALURA, 2020).

Entre as opções de linguagens, como C, Java, Python e outras, inicialmente optou-se por JavaScript, dada sua flexibilidade e ampla adoção. Contudo, para atender à necessidade de cálculos matemáticos mais complexos e geração de gráficos com maior controle, Python foi integrado ao projeto, permitindo a criação de gráficos personalizáveis e a exportação de dados para formatos como Excel. (MATPLOTLIB,2024)

1.1 PROBLEMA

Nos cálculos da equação de Adams e Heat Input usados no ciclo térmico, o uso de um aplicativo para realizar os cálculos e gráficos se faz necessários?

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Este projeto visa o desenvolvimento de um aplicativo e um software que gere resultados e gráficos, possibilitando visualizar, aumentar a precisão e compreender melhor o funcionamento e a utilidade da Equação de Adams e Heat Input.

1.2.2 Objetivos Específicos

O projeto será desenvolvido seguindo as etapas subsequentes:

- Compreensão de Lógica de Programação e Algoritmo;
- Processo descritivo de como será elaborado a lógica das equações;
- Escolha da linguagem e das ferramentas de desenvolvimento que serão usados no projeto;
- Desenvolvimento do projeto e testes;

1.3 JUSTIFICATIVA

O desenvolvimento deste aplicativo visa auxiliar alunos a resolver a equação de Adams e a equação de Heat Input, apresentando os resultados acompanhados de gráficos. Isso potencializa a compreensão dos efeitos dessas equações no processo de soldagem, destacando sua importância e seu impacto no comportamento do material durante a solda. Junto ao software é possível criar um arquivo excel com valores correspondentes aos resultados vindos do gráfico tanto para distância quanto para tempo.

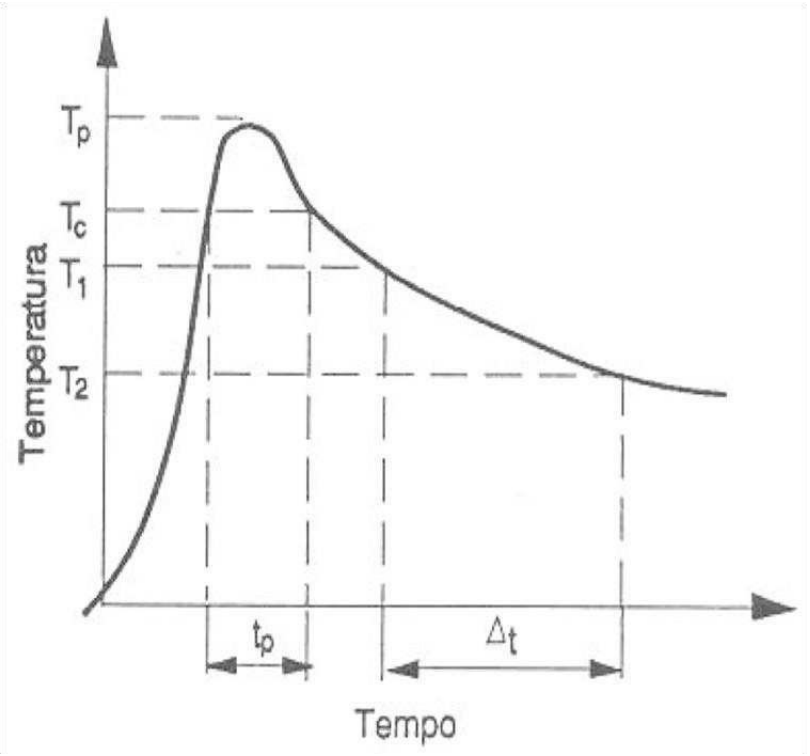
2 REVISÃO BIBLIOGRÁFICA

Para a compreensão do aplicativo e fundamental entender a finalidade que esse irá atender, com isso o assunto neste capítulo a ser analisado será o ciclo térmico, bem como tudo o que engloba o Heat input, a equação de Adams; posteriormente será dado as bases de compreensão usadas no desenvolvimento do aplicativo (MODENESI; MARQUES; SANTOS, 2024).

2.1 CICLO TÉRMICO

No processo de soldagem, o material experimenta uma variação de temperatura que depende entre fatores das condições de soldagem, propriedades térmicas e distância do ponto à fonte de calor. A esta variação de temperatura denomina-se ciclo térmico de soldagem, sendo feito um aquecimento até a temperatura de pico (Tp) e seguido por um resfriamento gradual. A Figura 01 ilustra o fenômeno (MODENESI; MARQUES; SANTOS, 2024)

Figura 1 Ciclo Térmico em soldagem

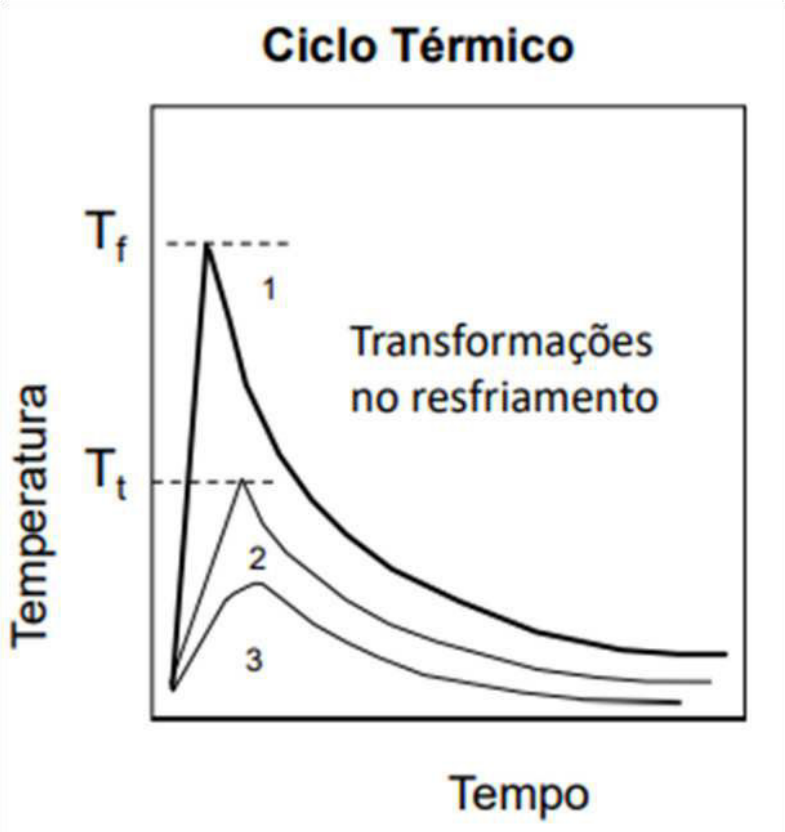


Fonte: Modenesi; Marques; Santos (2024)

O centro da poça de fusão, as temperaturas de pico e de fusão do material são iguais (Tp = Tf), e assim em cada ponto do metal base irá experimentar um ciclo térmico, e a Tp diminuirá conforme se distancia em relação à poça de fusão, e a sua velocidade de resfriamento é praticamente constante (MODENESI; MARQUES; SANTOS,2024).

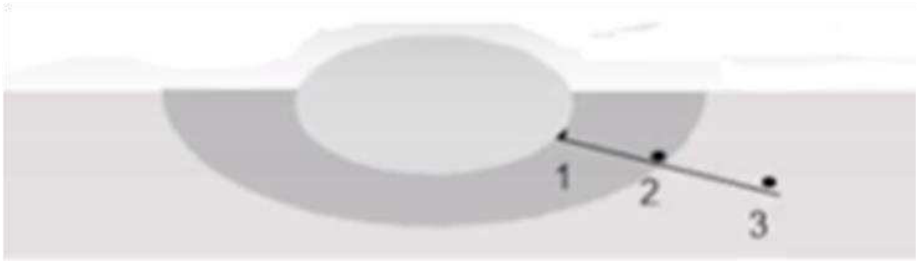
Na Figura 02 ilustra graficamente as transformações conforme o resfriamento e a Tf máximo temperatura atingida durante a soldagem zona Fundida metal liquido e 1 inicio da ZTA Zona Térmica Afetada e Tt fim da ZTA e ponto dois zona de ligação e ponto 3 metal base e na Figura 3 tem a imagem com as régios demarcadas (MODENESI; MARQUES; SANTOS,2024).

Figura 2 *Ciclo térmico em soldagem transformações no resfriamento.*



Fonte: Modenesi; Marques; Santos (2024)

Figura 3 *Ilustração do metal base na relação tempo/temperatura.*

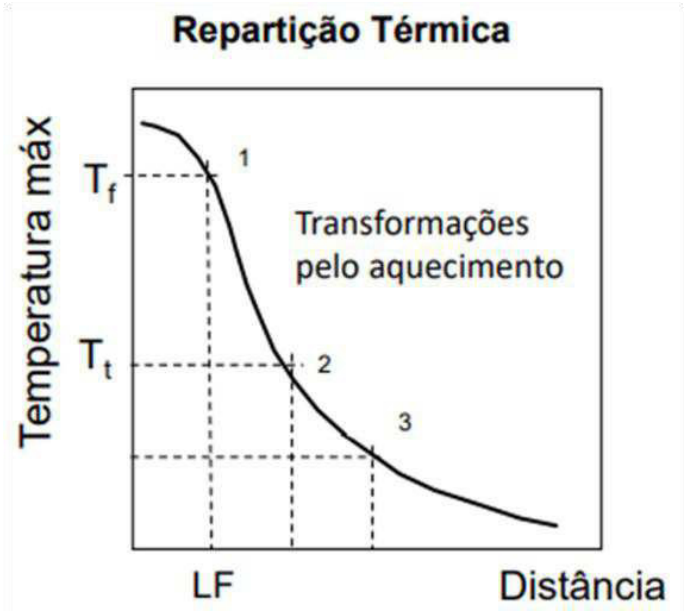


Fonte: Modenesi; Marques; Santos (2024)

E para a obtenção destes ciclos térmicos, utilizando-se de termopares, que podem ser dispostos em diversas posições numa juntada soldada é possível obter esses valores (SOLDAS BRASIL, 2024).

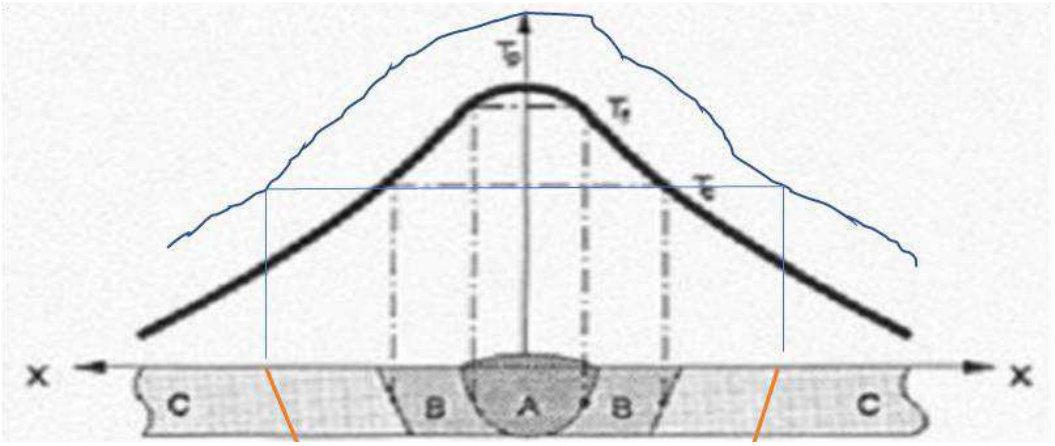
E com a curva de repartição térmica que temos a variação de temperatura máxima com a distância, em relação a linha de fusão e quando relacionamos a curva de repartição térmica com o diagrama de transformação de fases do material, temos uma ideia da microestrutura que encontraremos em determinada região do material base. A Figura 4 ilustra graficamente a relação da temperatura e a distância, e Figura 5 mostra como cada repartição será afetada SOLDAS BRASIL, 2024).

Figura 4 *Repartição térmica em soldagem*



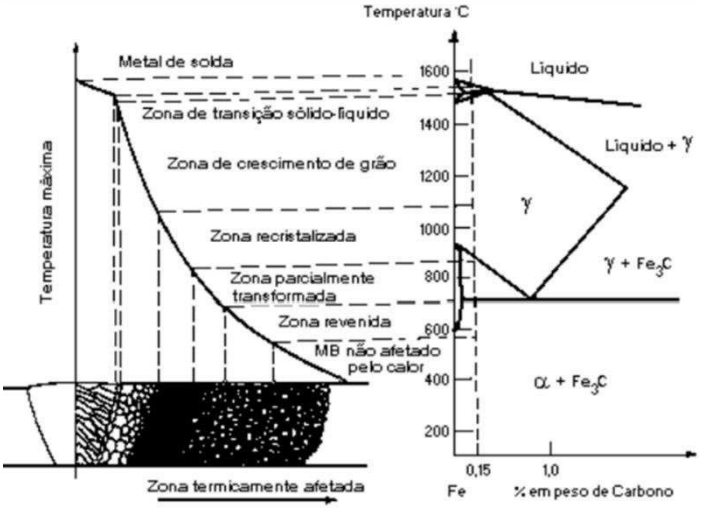
Fonte: Soldas Brasil (2024)

Figura 5 *Repartição térmica em soldagem*



Fonte: Soldas Brasil (2024)

Figura 6 *Relação da microestrutura e a temperatura.*



Fonte: Adptado Paredes (2024)

Quando relacionamos a curva de repartição térmica com o diagrama de transformação de fases do material, temos uma ideia da microestrutura que encontraremos em determinada região do material base. Conforme a Figura 6 mostra (PAREDES, 2024)

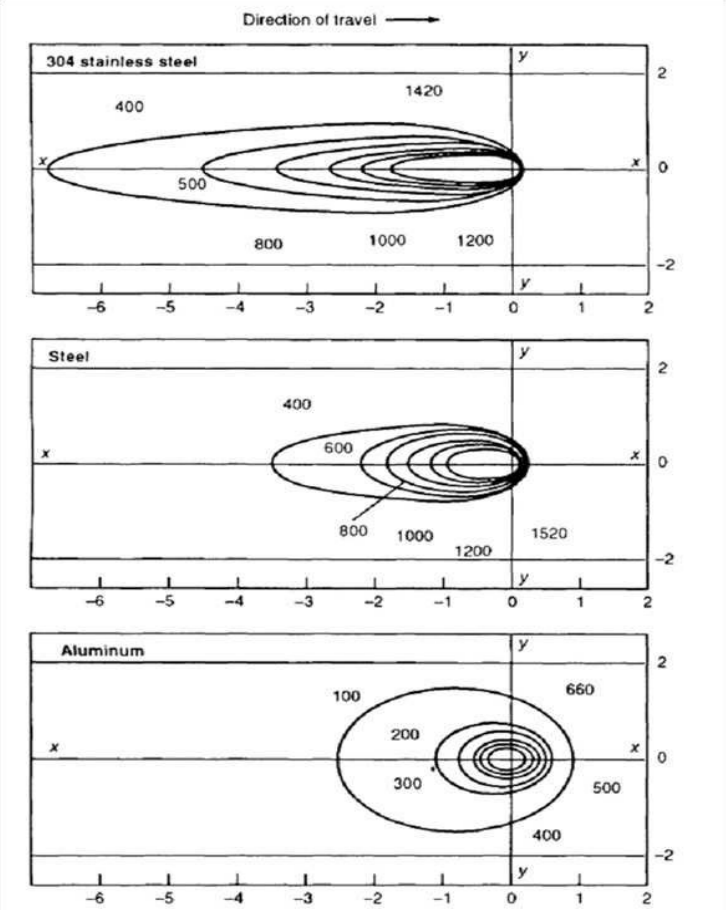
A temperatura de pico (Tp) é a temperatura máxima atingida em um dado ponto e possibilita indicar se houve a ocorrência de transformações microestruturais. Tem-se também a temperatura crítica (Tc) que é a última temperatura em que se ocorre uma transformação de fase, outro fator a se analisar é o tempo de permanência (tc),

pois quando acima de uma temperatura crítica (Tc) pode ser de interesse para materiais em que a dissolução de precipitados e/ou crescimento de grão podem ocorrer (Infosolsda, 2024).

2.1.1 Condutividade e Difusividade Térmica

Quanto mais baixo for a condutividade, maior será a poça de fusão e mais alongado, na direção oposta ao arco, será a distribuição de temperaturas. Como pode se ver na figura 7 que ilustra os efeitos desses fenômenos em um 304, um aço baixo-C e Alumínio; para um Heat Input de 4.2 KJ/s, velocidade de soldagem igual 8 mm/s (MODENESI; MARQUES; SANTOS, 2024).

Figura 7 *Relação da direção de viagem em cada material.*



Fonte: Modenesi; Marques; Santos (2024)

A difusividade térmica é uma propriedade termo física, ou seja, combina uma propriedade de transporte com uma propriedade termodinâmica, a difusividade térmica (K) relaciona a capacidade que o material tem para conduzir o calor com a capacidade para o armazenamento desse calor. Varia com a temperatura como observe-se na equação 1 (MODENESI; MARQUES; SANTOS, 2024).

Equação 1

$$k = \frac{\lambda}{pc_p}$$

Onde:

lambda (λ) corresponde a condutividade térmica (Propriedade de transporte);

(p) peso específico;

calor específico (Cp) a capacidade térmica volumétrica (Propriedade termodinâmica). Dessa forma a propriedade de transporte tem a capacidade de conduzir o calor e a propriedade termodinâmica por sua vez a de armazenar calor.

2.1.2 Condutividade e Difusividade Térmica

A relação entre a potência transmitida unidade Watts e a velocidade de soldagem em expressa em mm/seg é conhecida como aporte térmico ou energia soldagem (Heat Input) que sua unidade é dado em Joule por milímetro. Como é desmontada na equação 2 (MODENESI; MARQUES; SANTOS, 2024).

Equação 2

$$H = n \cdot \frac{V \cdot l}{Vs}$$

Onde:

V é a tensão do arco (volts);

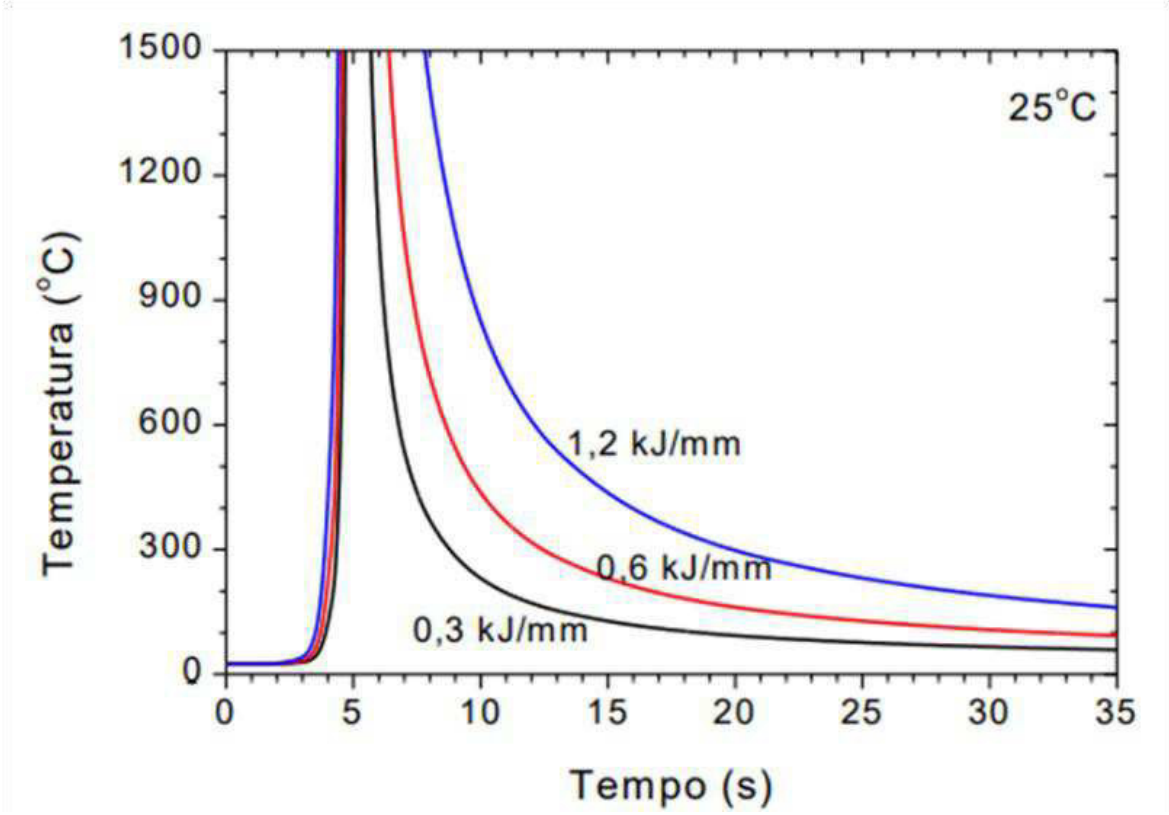
I é a corrente (ampéres);

Vs é a velocidade de soldagem (mm/s)

η é o rendimento térmico.

A escolha adequada da unidade da velocidade de soldagem ajudará nos cálculos posteriores de distribuição de temperatura, temperatura de pico e taxa de resfriamento. A Com o aumento da energia de soldagem a velocidade de resfriamento diminuir. Conforme observada na Figura 8(MODENESI; MARQUES; SANTOS, 2024).

Figura 8 Relação da Energia de soldagem com velocidade de resfriamento



Fonte: Modenesi; Marques; Santos (2024)

2.1.3 Equação de Adams

A equação 3 de Adams é uma aproximação da equação fundamental da transferência de calor, adaptada as condições de soldagem (fonte móvel de calor) Com ela é possível avaliar a influência da temperatura inicial na posição de determinada temperatura de pico. Quando Tp = Tc, esta equação nos dá a extensão

da ZTA (zona termicamente afetada) (MODENESI; MARQUES; SANTOS, 2024).

Equação 3

$$\frac{1}{T_p - T_0} = \frac{4{,}13.p.C_p.t.Y}{H_l} = \frac{1}{T_m - T_0}$$

Onde:

Tp = a temperatura de pico(°C) na distância Y;

T0 = temperatura do inicial do metal base (°C);

Tm = temperatura de fusão (°C);

t = espessura da chapa (mm);

Y = distância da temp. de pico em relação a linha de fusão (mm);

Hl = aporte térmico líquido (J/mm);

ρ = densidade do material (g/mm3);

Cp = calor específico do metal base (J/g.°C). ou, (J/mm3.K)

O 4,13 é uma constante que transforma do formato em elíptico da temperatura em linear.

2.1 CICLO TÉRMICO

Para compreender o processo e o funcionamento do aplicativo, é fundamental ter conhecimento de algoritmos, programação básica e da linguagem *JavaScript*, além de compreender bibliotecas e outros fatores (LINTZMAYER; MOTA, 2020)

2.2.1 Algoritmo

Conceito de algoritmo pode ser definido como uma sequência de instruções para resolver um determinado problema. Ao desenvolver um algoritmo está se estabelecendo um padrão de comportamento que deverá ser seguido para alcançar o resultado de um problema. E para o desenvolvimento de um algoritmo eficiente é

necessário obedecermos a algumas premissas básicas em sua construção (LINTZMAYER; MOTA, 2020):

Definir ações simples e sem ambiguidade;

Organizar as ações de forma ordenada;

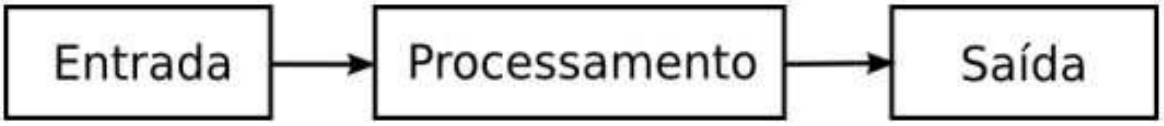
Estabelecer as ações dentro de uma sequência finita de passos;(LINTZMAYER; MOTA, 2020)

Um algoritmo quando programado em um computador é constituído pelo menos das 3 partes, sendo elas:

1. Entrada de dados;
2. Processamento de dados;
3. Saída de dados; (FERRARI; CECHINEL, 2008)

Na parte de entrada, são fornecidas as informações necessárias para que o algoritmo possa ser executado. Estas informações podem ser fornecidas quando o programa está sendo executado ou podem estar embutidas dentro do mesmo. A Figura 9 ilustra o fluxo básico de um algoritmo: (FERRARI; CHECHINEL, 2008)

Figura 9 Partes básicas de um algoritmo.



Fonte: Ferrari; Cechinel (2008)

2.2.2 Linguagem de Máquina e *Assembler*

Um computador necessita que lhe instruam ações bem específicas, estas ações devem ser passadas para o computador numa linguagem que ele possa entendê-las, chamada linguagem de máquina. Esta linguagem é composta somente por números, representados de forma binária, que, sob o ponto de vista dos computadores, representa as operações e os operandos que serão usados no processamento do programa. (FERRARI; CHECHINEL, 2008)

Para um ser humano, a linguagem de máquina é complexa de se compreender.

Assim, existe uma linguagem representada por comandos, mas que reproduzas tarefas que serão executadas dentro do computador, a linguagem de montagem (assembly). Entretanto, mesmo a linguagem de montagem é difícil de programar e os programas feitos para um determinado processador, por conterem instruções específicas deste, não funcionarão em um processador de outro tipo.(FERRARI; CHECHINEL, 2008)

2.2.3 Linguagens de Programação

Para facilitar a tarefa de programar um computador, foram criadas várias linguagens de programação. Estas linguagens são uma maneira de tentar escrever as tarefas que o computador vai realizar de maneira mais parecida com a linguagem natural. Ainda que seja muitas vezes complexo em comparação com a linguagem natural, um programa escrito em uma linguagem de programação é muito mais fácil de ser implementado, compreendido e modificado. (LINTZMAYER; MOTA, 2020).

As linguagens de programação são um meio termo entre a linguagem de máquina e a linguagem natural. Deste modo são classificadas de acordo com o nível entre a linguagem natural ou de máquina que ocupam. As linguagens muito parecidas com linguagem de máquina são chamadas de linguagens de baixo nível e suas instruções parecem-se muito com aquelas que serão executadas pelo processador. As linguagens de alto-nível são as que guardam mais semelhanças com a linguagem natural. Exemplos de linguagens de alto-nível são: *Pascal, Java, Perl, Python*, C#, PHP, entre outras. (LINTZMAYER; MOTA, 2020).

Dentre essas linguagens, optou-se por *JavaScript* para este projeto, devido à sua versatilidade e ampla adoção na web. É uma linguagem de programação de alto nível, o que significa que ela é mais próxima da linguagem natural, facilitando sua compreensão e implementação. Sua capacidade de execução tanto no lado do cliente quanto do servidor, juntamente com sua sintaxe simples e poderosa, torna-a uma escolha ideal para o desenvolvimento de aplicativos web e móveis. Além disso, o ecossistema robusto do *JavaScript* oferece uma vasta gama de bibliotecas e frameworks, como o *React Native*, que permite desenvolver aplicativos móveis de forma eficiente e consistente. (MDN Web Docs, 2024).

2.2.4 Linguagem de Programação *Javascript*

JavaScript é uma linguagem de programação de alto nível, amplamente utilizada no desenvolvimento *web*. Ela foi projetada para ser executada nos navegadores da web, permitindo a criação de interações dinâmicas e funcionais nas páginas da web. Originalmente criada por Brendan Eich em 1995, *JavaScript* rapidamente se tornou uma das linguagens mais populares devido à sua sintaxe simples e flexível. Uma das características mais poderosas do *JavaScript* é sua capacidade de manipular elementos HTML e CSS em tempo real, tornando possível criar interfaces interativas e responsivas. (MDN WEB DOCS, 2024)

Além disso, *JavaScript* é uma linguagem interpretada, o que significa que o código é executado linha por linha no navegador, sem a necessidade de compilação prévia. Com o avanço da tecnologia, *JavaScript* expandiu seu alcance além do desenvolvimento web, sendo também utilizado no desenvolvimento de aplicativos móveis, servidores e até mesmo em aplicações de machine learning. (W3SCHOOLS, 2024).

Sua versatilidade e ampla comunidade de desenvolvedores fazem do *JavaScript* uma escolha popular para uma variedade de projetos de *software*. Para este projeto, escolheu-se o *JavaScript* como a linguagem de programação principal devido à sua familiaridade e robustez, além de sua integração perfeita com o *React Native* e o Expo, que serão utilizados para desenvolver o aplicativo móvel. (MDN WEB DOCS, 2024).

2.2.5 *React Native*

React Native é um framework de desenvolvimento de aplicativos móveis criado pelo *Facebook* em 2015. Ele permite que desenvolvedores utilizem *JavaScript* e a biblioteca *React* para construir aplicativos nativos para iOS e Android. A principal vantagem do *React Native* é a capacidade de escrever código em *JavaScript* que pode ser executado em ambas as plataformas, reduzindo o tempo e o esforço necessários para desenvolver aplicativos móveis para múltiplos sistemas operacionais (REACT NATIVE, 2024).

Diferente de outras abordagens que utilizam *WebView* para renderizar componentes, o *React Native* traduz os componentes *React* diretamente em componentes nativos da plataforma, resultando em uma performance muito próxima

à de aplicativos nativos. Os componentes do React Native mapeiam para widgets nativos da plataforma, proporcionando uma experiência de usuário mais fluida e responsiva. (REACT NATIVE, 2024).

Além disso, o *React Native* permite a reutilização de grande parte do código entre diferentes plataformas, acelerando o desenvolvimento e a manutenção de aplicativos. Sua grande comunidade de desenvolvedores e a vasta quantidade de bibliotecas disponíveis facilitam a resolução de problemas e a implementação de funcionalidades adicionais. (ALURA, 2024)

2.2.5 APK

APK (*Android Package Kit*) é o formato de arquivo utilizado para distribuir e instalar aplicativos em dispositivos Android. Um arquivo APK contém todos os elementos necessários para que um aplicativo seja instalado corretamente em um dispositivo Android, incluindo o código compilado, recursos como imagens e sons, e um manifesto que descreve as permissões e características do aplicativo. (DEVELOPERS, 2024).

O processo de criação de um APK envolve a compilação do código fonte do aplicativo, a inclusão de todos os recursos necessários e a assinatura digital do arquivo para garantir sua autenticidade e segurança. Uma vez criado, o APK pode ser distribuído através de lojas de aplicativos como Google Play Store ou diretamente aos usuários por meio de download. (EXPO DOCS, 2024)

Para este projeto, o aplicativo será disponibilizado no formato APK, permitindo que os usuários de dispositivos Android possam baixar e instalar facilmente o aplicativo em seus smartphones e tablets. A escolha do formato APK se deve à sua compatibilidade universal com dispositivos Android e à facilidade de distribuição e instalação que ele oferece. (EXPO DOCS, 2024)

2.2.6 Linguagem de Programação Python

Python é uma linguagem de programação de alto nível, amplamente reconhecida por sua simplicidade e legibilidade, o que a torna uma das linguagens mais acessíveis para desenvolvedores iniciantes e experientes. Criada por Guido van Rossum em 1991, Python foi projetada para enfatizar a clareza do código, utilizando

uma sintaxe intuitiva e direta. Essa característica é especialmente útil em aplicações científicas, análise de dados e desenvolvimento de software onde a manutenção e a colaboração são essenciais (PYTHON SOFTWARE FOUNDATION, 2024).

Uma das maiores vantagens do Python é sua vasta biblioteca padrão, que inclui ferramentas para manipulação de dados, criação de interfaces gráficas, cálculo numérico e geração de gráficos. Adicionalmente, sua comunidade ativa e suporte a bibliotecas externas o tornam ideal para aplicações avançadas, como inteligência artificial, aprendizado de máquina e ciência de dados. No contexto deste projeto, a escolha pelo Python deve-se à sua capacidade de manipular dados numéricos, gerar gráficos personalizados e exportar resultados de forma eficiente (W3SCHOOLS, 2024).

O Python também é conhecido por sua portabilidade, permitindo que programas desenvolvidos sejam executados em diversos sistemas operacionais, como Windows, Linux e macOS, com alterações mínimas. Isso o torna uma escolha robusta para o desenvolvimento de aplicações desktop e sistemas científicos.

Para este projeto, foi utilizado o Python em conjunto com bibliotecas como Matplotlib, Tkinter e Pandas, permitindo a criação de gráficos detalhados e a exportação de dados para Excel. Essa solução complementa o aplicativo React Native ao oferecer uma alternativa para uso em ambientes desktop e processamento mais intensivo (PYTHON.ORG, 2024).

3 MATERIAIS E MÉTODOS

Neste trabalho foi desenvolvido um programa que funciona em celular/desktop que calcula o *Heat Input* e a Equação de Adams (Equação 3) onde foi utilizado as ferramentas de programação: *JavaScript, ReactNative*, Expo e outras bibliotecas.

Equação 3

$$\frac{1}{T_p - T_0} = \frac{4{,}13.p.C_p.t.Y}{H_l} = \frac{1}{T_m - T_0}$$

Assim determinou-se os locais que armazenaram os parâmetros que inseridos para realizar o cálculo como segue na Figura 10.

Figura 10 Programação utilizando a equação de Adams

```
<View>
  <InputField label="Temperatura de Pico (Tp)" value={inputs.tp} onChangeText={(value) => handleInputChange('tp', value)} />
  <InputField label="Temperatura Inicial (To)" value={inputs.to} onChangeText={(value) => handleInputChange('to', value)} />
  <InputField label="Tensão (V)" value={inputs.tensao} onChangeText={(value) => handleInputChange('tensao', value)} />
  <InputField label="Amperagem (I)" value={inputs.amperagem} onChangeText={(value) => handleInputChange('amperagem', value)} />
  <InputField label="Velocidade de Soldagem (mm/min)" value={inputs.velocidadeSoldagem} onChangeText={(value) => handleInputChange('velocidadeSoldagem', value)} />
  <InputField label="Valor de n" value={inputs.n} onChangeText={(value) => handleInputChange('n', value)} />
  <InputField label="Densidade do Material (p)" value={inputs.densidade} onChangeText={(value) => handleInputChange('densidade', value)} />
  <InputField label="Calor Específico (Cp)" value={inputs.calorEspecifico} onChangeText={(value) => handleInputChange('calorEspecifico', value)} />
  <InputField label="Espessura da Chapa (t)" value={inputs.espessuraChapa} onChangeText={(value) => handleInputChange('espessuraChapa', value)} />
  <InputField label="Distância em mm" value={inputs.distancia} onChangeText={(value) => handleInputChange('distancia', value)} />
  <InputField label="Temperatura de Fusão do Material (Tm)" value={inputs.temperaturaFusao} onChangeText={(value) => handleInputChange('temperaturaFusao', value)} />
</View>
```

Fonte: Elaborado pelo o Autor

Com os valores armazenados, e visando a implantação gráfica, que será entendida mais a frente foi utilizado uma função do *JavaScript o 'parseFloat'* que converte uma *string* (um dado em texto) em um número de ponto flutuante (*float*) demoatrado na Figura 11.

Figura 11 Tela para gerar o gráfico

```
const calcular = () => {
  const I = parseInputValues(inputs.amperagem);
  const V = parseInputValues(inputs.tensao);
  const n = parseInputValues(inputs.n);
  const velocidadeSoldagem = parseInputValues(inputs.velocidadeSoldagem);
  const Tp = parseInputValues(inputs.tp);
  const To = parseInputValues(inputs.to);
  const p = parseInputValues(inputs.densidade);
  const Cp = parseInputValues(inputs.calorEspecifico);
  const t = parseInputValues(inputs.espessuraChapa);
  const d = parseInputValues(inputs.distancia);
  const Tm = parseInputValues(inputs.temperaturaFusao);
```

Fonte: Elaborado pelo o Autor

Em relação a equação foi utilizado as const, onde corresponderiam aos valores inseridos pelo usuário enquanto. A primeira linha da imagem do código é para o cálculo do *Heat Input*, enquanto a segunda, terceira e quarta linha trata da Equação de Adams onde separa ela em três, resolvendo cada parte e com os resultados.

A primeira linha da imagem do código é para o cálculo do *Heat Input*, enquanto a segunda, terceira e quarta linha trata da Equação de Adams onde separa ela em três, resolvendo cada parte e com os resultados ilustradas na Figura 12.

Figura 12 Código de Programação

```
const Ht = (I * V * n * 60) / velocidadeSoldagem;
const resulUm = (1 / (Tp - To));
const resulDois = ((4.13 * (p * Cp / 1000000000) * t * 1) / Ht);
const resulTres = (1 / (Tm - To));
const resultadoAdams = ((resulUm - resulTres) / resulDois);

const ciclos = [];
for (let i = 0; i <= d; i++) {
  const Tp_mm = (1 / (((4.13 * (p * Cp / 1000000000) * t * i / Ht)) + resulTres)) + To;
  ciclos.push({ mm: i, temperatura: Tp_mm });
}

setGraficoDistancia(ciclos);

const temperaturaTempo = ciclos.map((ciclo, index) => {
  const VelSolTem = (60 / velocidadeSoldagem) * ciclo.mm;
  return {
    temperatura: Math.floor(ciclo.temperatura),
    tempo: VelSolTem.toFixed(1)
  };
});
```

Fonte: Elaborado pelo o Autor

Loop: É uma estrutura de controle que permite repetir um bloco de código várias vezes até que uma condição específica seja atendida.

For: Uma palavra-chave em JavaScript (e em outras linguagens de programação) que inicia um loop. E é frequentemente usado para iterar sobre uma sequência de elementos, como uma array, realizando uma operação para cada elemento.

Array: É uma estrutura de dados que armazena uma coleção de elementos, geralmente do mesmo tipo. Em JavaScript, uma array é uma lista ordenada de valores, que podem ser acessados por meio de um índice numérico. Arrays são muito úteis para armazenar e manipular conjuntos de dados.

Com as equações feitas iniciou o processo para a interface gráfica, partindo para a criação de input, campo onde seria inserido os valores para serem colocados pelo usuário ilustrada na Figura 13.

Figura 13 Código ao campo referente a exibição dos cálculos do ciclo térmico

```
setGraficoTempo(temperaturaTempo);

const formattedResult = `
Valor correspondente para Y em Tp = ${Tp}° é: ${resultadoAdams.toFixed(2)}
Heat Input (Ht) = ${Ht.toFixed(0)}
`;

setResultado(formattedResult);
setModalVisible(true);
```

Fonte: Elaborado pelo o Autor

Para transformar o projeto em um APK utilizável, algumas mudanças no código foram necessárias. O processo de build com Expo envolve as seguintes etapas:

1. Configurar o projeto Expo, especificando as dependências e configurações necessárias no arquivo app.json.

2. Fazer as alterações no código-fonte para garantir que o aplicativo esteja pronto para ser compilado como um APK. Isso inclui configurar corretamente as permissões e otimizar o código para a plataforma.

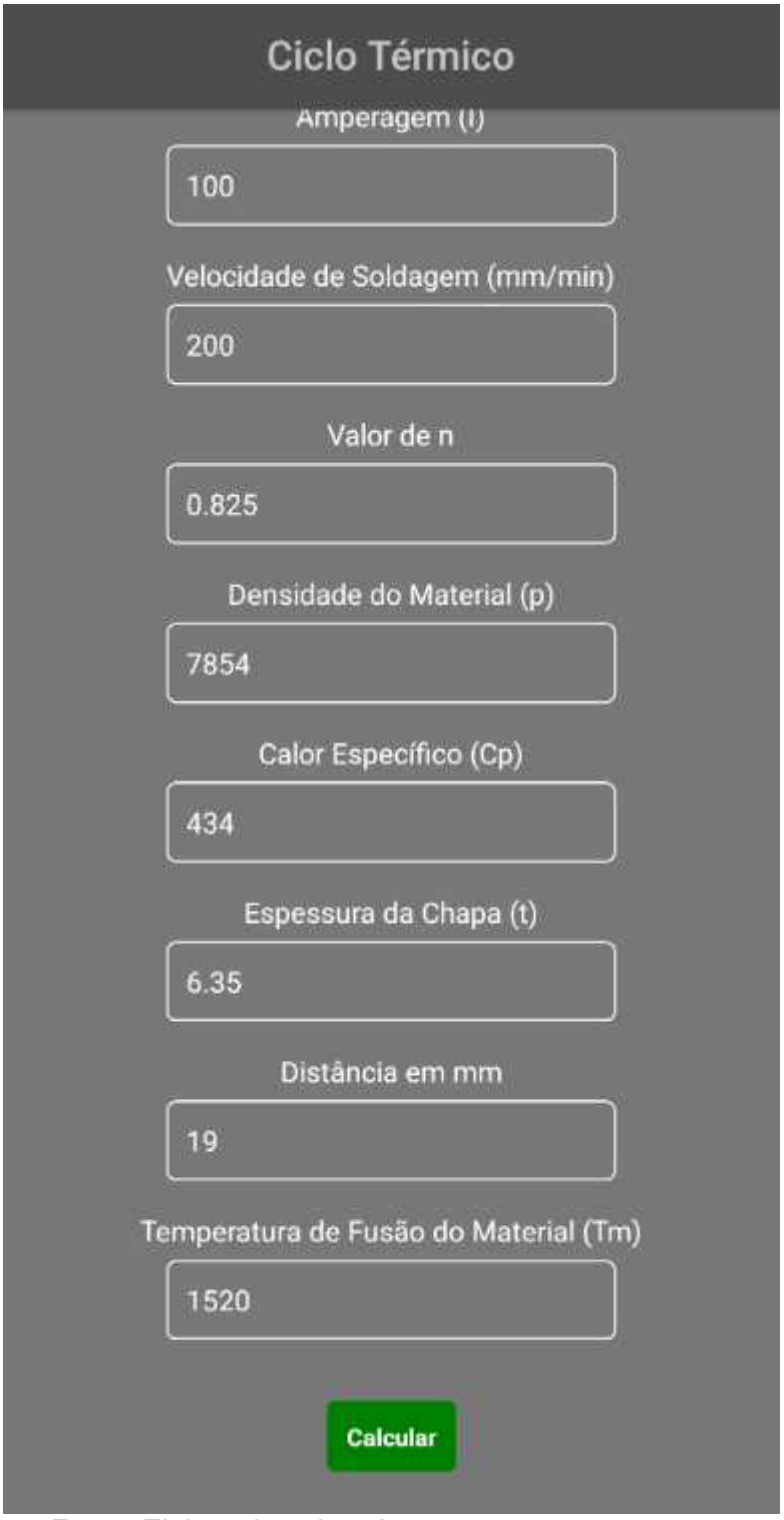
Utilizar o comando expo build:android para iniciar o processo de build. O Expo então compila o código-fonte e os recursos em um arquivo APK. Durante este processo, o Expo realiza todas as etapas necessárias, como empacotamento dos

recursos, minificação do código e assinatura digital do APK para garantir sua autenticidade e segurança.

4 RESULTADOS E DISCUSSÃO

Para fins ilustrativos foram inseridos valores nos campos da Figura 14 onde ao final dos campos tem um botão “Calcular”, onde que ao ser clicado exibe um gráfico baseado nos valores que também foram calculados.

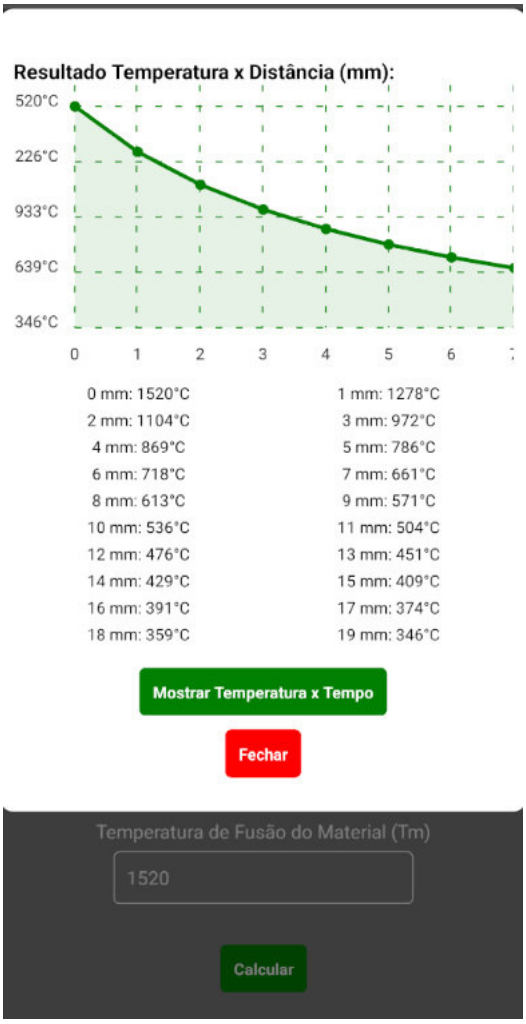
Figura 14 valores para o cálculo do *Heat Input*



Fonte: Elaborado pelo o Autor

Nesta imagem existe outro botão, com um ícone de ponto de interrogação, que ao ser clicado exibe, uma breve descrição do que é Ciclo Térmico bem como o que é Heat Input. Voltando para o ciclo de execução ao clicar em “Calcular” com os valores inseridos o gráfico será exibido conforme a Figura 15.

Figura 15 Gráfico com valores da velocidade de resfriamento calculados



Fonte: Elaborado pelo o Autor

Com isso se o usuário seguir e clicar em “Mostrar Resultado”, segue para outro modal, com os valores indicados de 1mm até 10mm, junto do valor do Heat Input como na Figura 16

Figura 16 valores correspondendetes a Y em Tp e o valor do Heat Input

Fonte: Elaborado pelo o Autor

O desenvolvimento do aplicativo focou em fornecer ao usuário uma ferramenta capaz de calcular e visualizar o ciclo térmico e o *Heat Input* durante o processo de soldagem. A proposta inicial do aplicativo foi alcançada com sucesso, permitindo que os usuários insiram parâmetros específicos e obtenham resultados precisos, bem como gráficos representativos das variações de temperatura ao longo do processo.

Os resultados obtidos através do aplicativo demonstraram coerência com os fenômenos esperados. A precisão dos cálculos depende diretamente da exatidão dos parâmetros inseridos pelo usuário, o que reforça a importância de uma coleta de dados cuidadosa. O aplicativo permite que os usuários insiram variáveis como a velocidade de soldagem, a corrente e a tensão, entre outros parâmetros, e gera gráficos que ilustram a variação da temperatura em função da distância. Esses gráficos proporcionam uma visualização clara das mudanças térmicas, facilitando a interpretação dos resultados.

Um dos principais benefícios observados foi a capacidade do aplicativo de representar graficamente o ciclo térmico. A visualização gráfica ajuda a identificar os pontos críticos onde ocorrem mudanças significativas de temperatura, o que é essencial para a análise de qualidade da solda e para a prevenção de defeitos. Os gráficos também permitem uma comparação fácil entre diferentes condições de soldagem, ajudando os usuários a otimizarem os parâmetros para alcançar melhores resultados.

Além disso, o aplicativo mostrou-se eficiente em calcular o heat input, uma métrica crucial para o controle do processo de soldagem. O heat input afeta diretamente as propriedades mecânicas e metalúrgicas da junta soldada. Ao fornecer cálculos precisos do heat input, o aplicativo auxilia os profissionais da área de soldagem a

controlar e ajustar seus processos para garantir a integridade e a qualidade das soldas.

O aplicativo desenvolvido cumpre seu objetivo de calcular e visualizar o ciclo térmico e o *Heat Input* de forma prática e precisa, proporcionando uma ferramenta valiosa para pessoas da área de soldagem. Essa abordagem possibilita um melhor entendimento e controle dos processos térmicos, contribuindo para a melhoria da qualidade das soldas e para a eficiência do processo produtivo.

Em resumo, os resultados indicam que o aplicativo é uma ferramenta eficaz e útil para a análise térmica no processo de soldagem. Ele facilita a obtenção de dados precisos e oferece uma interface intuitiva para a visualização e interpretação dos resultados.

Com base no aplicativo desenvolvido, foi criado também um programa em Python que cumpre os mesmos objetivos, adaptando-se apenas às diferenças de sintaxe, escrita e estilização características da linguagem. Apesar das alterações necessárias para adequação ao ambiente desktop, o programa preserva a lógica de cálculo e as funcionalidades principais, permitindo calcular e visualizar o ciclo térmico e o Heat Input com a mesma precisão e praticidade. Essa adaptação oferece aos usuários uma alternativa robusta e versátil para análise térmica, mantendo a eficácia e a utilidade já presentes no aplicativo original.

5 CONSIDERAÇÕES FINAIS

Com o objetivo de desenvolver um aplicativo que possibilitasse ao usuário, com conhecimento sobre os parâmetros, obter resultados relacionados ao ciclo térmico e ao heat input durante o processo de soldagem, este trabalho alcançou resultados significativos. O aplicativo calcula o heat input, fornecendo uma medida da energia térmica introduzida no material, e gera o ciclo térmico, apresentando a variação de temperatura em relação à distância. Essa funcionalidade é importante, pois permite uma análise detalhada das transformações térmicas e suas implicações na microestrutura e nas propriedades mecânicas do material.

A inclusão de gráficos que visualizam a relação entre a distância e a temperatura proporciona uma ferramenta que possibilita interpretação dos resultados e permitindo uma melhor compreensão do comportamento térmico do material durante a soldagem.

A aplicação desenvolvida atende aos requisitos propostos inicialmente e demonstra a viabilidade de utilizar tecnologias modernas para resolver problemas complexos na engenharia de soldagem. No entanto, algumas limitações foram identificadas ao longo do desenvolvimento e testes do aplicativo. Por exemplo, a precisão dos resultados depende diretamente da precisão dos parâmetros inseridos pelo usuário, o que destaca a importância de uma coleta de dados cuidadosa e precisa.

Para trabalhos futuros, sugere-se a implementação de funcionalidades adicionais que possam aumentar a robustez e a versatilidade do aplicativo. Entre essas funcionalidades, destaca-se a inclusão de uma base de dados com materiais comuns e seus respectivos parâmetros térmicos, além da integração com sensores em tempo real que permitam a coleta automática de dados durante o processo de soldagem.

Em resumo, este trabalho contribuiu de maneira significativa para o desenvolvimento de uma ferramenta prática e eficiente para a análise de ciclos térmicos e heat input na soldagem. O aplicativo desenvolvido pode ser uma valiosa

adição ao conjunto de ferramentas utilizadas por estudantes, proporcionando uma abordagem informativa do processo de soldagem.

Como consideração final, pretende-se disponibilizar o aplicativo desenvolvido na Google Play Store, tornando-o acessível a um público mais amplo de profissionais e estudantes da área de soldagem. A disponibilização na Play Store proporcionará maior visibilidade ao aplicativo e facilitará o acesso dos usuários interessados, contribuindo assim para sua adoção e utilização no contexto prático da soldagem.

Com base no desenvolvimento do aplicativo, foi também criada uma versão em Python, adaptada para execução em ambiente desktop. Este programa mantém os mesmos objetivos e funcionalidades do aplicativo original, mas se beneficia das particularidades da linguagem Python, como sua simplicidade e flexibilidade, especialmente para cálculos numéricos e visualização de dados. A transição para Python envolveu apenas adaptações na sintaxe e estilização, garantindo que a lógica de cálculo e as representações gráficas fossem preservadas. Com isso, foi possível criar uma alternativa robusta e multiplataforma para análise do ciclo térmico e do heat input.

Essa versão desktop do programa foi projetada para atender usuários que preferem realizar as análises em um ambiente de trabalho tradicional, oferecendo uma interface simples e intuitiva, além de funcionalidades como a exportação de resultados para planilhas Excel. A inclusão dessa opção demonstra a versatilidade do projeto, ampliando seu alcance e atendendo a diferentes preferências e contextos de uso.

REFERÊNCIAS

ALURA. **React Native: o que é e tudo sobre o Framework** Disponível em: < https://www.alura.com.br/artigos/react-native/> Acessado em: 01/04/2024

DEVELOPERS ANDROID**. Arquivos de expansão do APK** Disponível em: < https://developer.android.com/google/play/expansion-files?hl=pt-br/> Acessado em: 01/04/2024

EXPO DOCS**. Build APKs for Android Emulators and devices** Disponível em: < https://docs.expo.dev/build-reference/apk/> Acessado em: 01/04/2024

FERRARI, F.; CECHINELO, C. **INTRODUÇÃO A ALGORITMOS E PROGRAMAÇÃO** Disponível em: < file:///D:/TG/FFerrari-CCechinel-Introducao-a-algoritmos.pdf /> Acessado em: 01/04/2024

INFOSOLDA**. Metalurgia de Solda: Conceitos** Disponível em: <https://infosolda. com.br/metalurgia-da-soldagem-conceitos/> Acessado em: 01/04/2024

LINTZMAYER, C. N.; MOTA, G, O. **Análise de Algoritmos e Estruturas de Dados** Disponível em: <file:///D:/TG/An%C3%A1lise%20de%20Algoritmos%20e%20Estruturas%20de%20Dados.pdf/> Acessado em: 01/04/2024

MDN WEB DOCS**. JavaScript** Disponível em: < https://developer.mozilla.org/en-US/docs/Web/JavaScript/> Acessado em: 01/04/2024

MODENESI, P. J.; MARQUES, P. V.; SANTOS, D. B. **Introdução à Metalurgia da Soldagem** Disponível em : < chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/ https://demet.eng.ufmg.br/wp-content/uploads/2012/10/metalurgia.pdf/> Acessado em: 01/04/2024

PAREDES, R. S. C. **Introdução à Metalurgia da Soldagem** Disponível em: <chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/ http://ftp. demec.ufpr.br/ disciplinas/EME733/Arquivos%20da%20disciplina/Introdu__o%20_%20Metalurgia%20da%20Soldagem.pdf/> Acessado em: 01/04/2024

REACT NATIVE**. Introduction** Disponível em: < https://reactnative.dev/docs/getting-started/> Acessado em: 01/04/2024

SOLDAS BRASIL. **Metalurgia de Soldagem** Disponível em: <chrome-extension:// efaidnbmnnnibpcajpcglclefindmkaj/https://www.soldasbrasil.com.br/pdf/metalurgia-soldagem-1.pdf/> Acessado em: 01/04/2024

W3SCHOOLS**. JavaScript Tutorial** Disponível em: <https://www.w3schools.com/js/default.asp/> Acessado em: 01/04/2024

APÊNDICE

Link para aceso do código:
APLICATIVO:
https://docs.google.com/document/d/1TGzHJ-i6IIL_Zs9zETbqyYlg3HhrPjMe0k1CR4BLOdQ/edit?usp=sharing

SOFTWARE:
https://docs.google.com/document/d/1_Fc9LKQPowb4G8XsKxadA9RoMMPzMq3ev5HwaaxhSoI/edit?usp=sharing

Ou acessando o repositório no GitHub:
APLICATIVO:
https://github.com/Matheus4ndrade/C_Termico

SOFTWARE:
https://github.com/Matheus4ndrade/cTermico