

CENTRO DE TECNOLOGIA DA INDÚSTRIA QUÍMICA E TÊXTIL
FACULDADE DE ENGENHARIA QUÍMICA SENAI: CENTRO DE TECNOLOGIA DA
INDÚSTRIA QUÍMICA E TÊXTIL

RAFAEL HENRIQUE ELIAS DIAS

**DESENVOLVIMENTO DE UMA FERRAMENTA DIDÁTICA PARA O AUXÍLIO DO
PROJETO DE HIDROCICLONES**

Rio de Janeiro
2020

RAFAEL HENRIQUE ELIAS DIAS

**DESENVOLVIMENTO DE UMA FERRAMENTA DIDÁTICA PARA O AUXÍLIO DO
PROJETO DE HIDROCICLONES**

Trabalho de conclusão de curso apresentado ao curso de graduação em engenharia química, da Faculdade de Engenharia Química SENAI CETIQT, como requisito parcial para obtenção do título de bacharel em engenharia química.

Orientador(a): Tanise Mori Flores

Rio de Janeiro

2020

Ficha Catalográfica

Dias, Rafael Henrique Elias

Desenvolvimento de uma ferramenta didática para o auxílio do projeto de hidrociclones / Rafael Henrique Elias Dias. - Rio de Janeiro, 2020 entrega.

88 f. : il.

Orientadora: Tanise Mori Flores

Projeto final de curso a ser submetido à Comissão Examinadora do Curso de Bacharelado em Engenharia Química, da Faculdade SENAI CETIQT, como parte dos requisitos necessários à obtenção do grau de Bacharel em Engenharia Química

1. Simulação. 2. Ferramentas Digitais. 3. Indústria 4.0. 4. Hidrociclones. 5. Educação. I. Desenvolvimento de uma ferramenta didática para o auxílio do projeto de hidrociclones.

RAFAEL HENRIQUE ELIAS DIAS

**DESENVOLVIMENTO DE UMA FERRAMENTA DIDÁTICA PARA O AUXÍLIO DO
PROJETO DE HIDROCICLONES**

Trabalho de conclusão de curso apresentado ao curso de graduação em engenharia química, da Faculdade de Engenharia Química SENAI CETIQT, como requisito parcial para obtenção do título de bacharel em engenharia química.

Aprovada em:

Aprovado por:

Orientador: Tanise Mori Flores, M.Sc.

Banca Examinadora*:

Alex Vazzoler, D.Sc., SENAI CETIQT

Elene de Souza Freitas, M.Sc., SENAI CETIQT

Thiago Sousa de Oliveira, M.Sc., AVEVA

Rio de Janeiro

2020

*as assinaturas foram geradas eletronicamente no site <https://www.assinaturagratis.com/>, e encontram-se no final do documento.

Aos professores que buscam ajudar seus alunos
a se preparar para a quarta revolução industrial.
Aos alunos que precisam cada vez mais se
aproximar do meio digital e suas tecnologias.

AGRADECIMENTOS

Sou muito grato à minha orientadora, Tanise Flores, pelo tempo e atenção dedicados à mim, durante a execução deste trabalho.

Também me sinto inexplicavelmente grato aos professores que foram parte integral da minha formação, como pessoa e profissional. Em especial, muito obrigado: Cláudia Gimenez, Flavia Pimenta, Andrea Parente e Alberto Vieira.

Por fim, deixo meus agradecimentos à Thiago Oliveira, que me aceitou como seu mentorando por múltiplos ciclos. Foi por meio do nosso contato que me senti confiante e focado na execução deste trabalho e, por isso, sou eternamente grato.

RESUMO

O ensino de tecnologias associados à digitalização da indústria no Brasil é comumente tratado por meios extra-disciplinares e com pouca integração às disciplinas estudadas em sala de aula. Esta monografia busca contribuir para a integração de tecnologias digitais ao cotidiano do aluno, por meio de uma ferramenta que auxilia o usuário no projeto e dimensionamento de hidrociclones, a partir da solução de questões similares às estudadas no meio acadêmico. Para o desenvolvimento da ferramenta, utilizou-se os softwares Unity e Blender. O foco deste trabalho são sistemas com baixa concentração de sólidos na alimentação que usem hidrociclones Rietema ou Bradley. As soluções utilizam parâmetros calculados e propostos por Ricardo Medronho e Mônica Antunes para as configurações Rietema e Bradley. Nessas condições, é possível solucionar tais sistemas através do adimensional Stk_{50Eu} (Stokes de Corte Reduzido Euler), no qual as metodologias de soluções de Medronho se concentram.

A ferramenta, chamada HidroCSim, conta com a representação visual de um hidrociclone parcialmente fiel ao calculado, não representando precisamente as dimensões calculadas, na parte visual. A ferramenta apresenta desvios pequenos nos seus resultados, quando comparados às questões propostas por Ricardo Medronho e suas soluções.

Este trabalho tem como objetivo contribuir para o ensino de ferramentas digitais na engenharia química, buscando servir como introdução à softwares de simulação. Uma proposta de uso da ferramenta pode ser consultada nos apêndices. A ferramenta pode ser melhor compreendida por meio da monografia, caso o leitor tenha interesse em elaborar uma proposta com base na ferramenta. Tanto a ferramenta (utilizável independentemente), quanto o projeto da ferramenta (utilizável no Unity), estão disponíveis online, em um link no apêndice A.

Palavras-chave: Simulação, Ferramentas Digitais, Indústria 4.0, Hidrociclones, Educação

ABSTRACT

In Brazil, technologies associated with industry digitalization are usually studied as extra-disciplinary works, with little integration to academic and engineering disciplines. This work aims to contribute to the integration of some of said technologies to the everyday of engineering students, through a developed tool that aims to assist the user on the scaling of hydrocyclones, based on the solution of academic questions. The software Unity and Blender were used for the development of the tool. The digital tool, named HidroCSim, focus on solving systems characterized by low feed concentration (under 10% solid volume) and is limited to hydrocyclones of both Rietema and Bradley configurations. The solution uses primordially parameters and methodologies suggested by Ricardo Medronho and Mônica Antunes in their works, using also a dimensionless group called Stk_{50Eu} .

The developed tool also allows the user to consult a visual representation of a hydrocyclone, that is partially true to a real, physical hydrocyclone. The calculated results of the tool show minor deviations compared to the manual solutions of the questions presented in this work.

The main objective of this work is to assist the teaching of digital tools for chemical engineers, as an introduction to simulation software. It also presents an educational proposal that can be read in its appendix. Also, this work aims to help the reader in understanding the tool, should the reader find any interest in using its content to create any kind of teaching situation. Both the tool itself and its development project can be accessed through a link that can be found in appendix A.

Keywords: Simulation, Digital Tools, Industry 4.0, Hydrocyclones, Education

LISTA DE FIGURAS

Figura 1 -	Configuração de sistema de controle digitais em tempo real.....	19
Figura 2 -	Softwares mais úteis na engenharia química.....	21
Figura 3 -	Exemplo de código para impressão da frase " <i>Hello World!</i> " no console do Unity.....	22
Figura 4 -	Modelo de Referência <i>RAMI4.0</i>	23
Figura 5 -	Esboço da operação de um hidrociclone.....	27
Figura 6 -	Parâmetros de dimensionamento de ciclones e hidrociclones.....	28
Figura 7 -	Painel geral da ferramenta HidroCSim.....	33
Figura 8 -	Fluxograma do processo de solução da ferramenta.....	35
Figura 9 -	Painel geral da ferramenta, com a opção “Dimensionar” evidenciada..	36
Figura 10 -	Painel geral da ferramenta e vista do esboço do hidrociclone.....	38
Figura 11 -	Painel de resultados e símbolos da ferramenta.....	39
Figura 12 -	Painel de distribuição granulométrica (DTA) da ferramenta.....	40
Figura 13 -	Esboço das peças do hidrociclone separadas no Blender.....	40
Figura 14 -	Exemplo visual do hidrociclone após acoplagem das peças.....	41
Figura 15 -	Comparativo visual entre o protótipo de hidrociclone Bradley (1) e Rietema (2).....	46
Figura 16 -	Representação formada pela ferramenta para as questões 3 e 6.....	47

LISTA DE TABELAS

Tabela 1 -	Parâmetros relativos às configurações de hidrociclones.....	30
Tabela 2 -	Primeiro conjunto de parâmetros obtidos por Medronho e Antunes para hidrociclones Rietema e Bradley.....	30
Tabela 3 -	Segundo conjunto de parâmetros obtidos por Medronho e Antunes para hidrociclones Rietema e Bradley.....	31
Tabela 4 -	Comparativo de resultados obtidos entre as soluções propostas para as questões e a ferramenta.....	45
Tabela 5 -	Comparativo de resultados obtidos entre as soluções propostas para as questões e a ferramenta.....	46

LISTA DE QUADROS

Quadro 1 -	Variáveis necessárias para o projeto de hidrociclones pela ferramenta, em ordem de solução.....	37
Quadro 2 -	Variáveis necessárias para o dimensionamento de hidrociclones pela ferramenta, em ordem de solução.....	37

LISTA DE ABREVIATURAS E SIGLAS

<i>CPS</i>	<i>Cyber Physical Systems</i>
<i>I4.0</i>	<i>Indústria 4.0</i>
<i>IOS</i>	<i>Internet of Services</i>
<i>IOT</i>	<i>Internet of Things</i>
<i>M&S</i>	<i>Modelagem e Simulação</i>
<i>PSE</i>	<i>Process System Engineering</i>
<i>RAMI4.0</i>	<i>Reference Architectural Model Industrie 4.0</i>
<i>STEM</i>	<i>Science, Technology, Engineering and mathematics</i>

LISTA DE SÍMBOLOS

A	Parâmetro ‘A’ associado à configuração do hidrociclone
B	Parâmetro ‘B’ associado à configuração do hidrociclone
C	Parâmetro ‘C’ associado à configuração do hidrociclone
K	Parâmetro ‘K’ associado à configuração do hidrociclone
β	Parâmetro ‘ β ’ associado à configuração do hidrociclone
y	Variável dependente de uma função
x	Variável independente de uma função
m	Coeficiente angular da função linearizada
b	Coeficiente linear da função linearizada
k	Coeficiente associado ao coeficiente m e b da
Cv	Fração volumétrica de sólidos na corrente de alimentação do sistema
Cva	Fração volumétrica de sólidos na corrente de alimentação do hidrociclone
Cvr	Fração volumétrica de sólidos na corrente de retido do hidrociclone
d50	Diâmetro de Corte
d50’	Diâmetro de corte reduzido
Dc	Diâmetro da parte cilíndrica do hidrociclone
Di	Diâmetro do tubo de alimentação do hidrociclone
Do	Diâmetro do tubo de saída superior do hidrociclone (overflow)
Du	Diâmetro da saída inferior do hidrociclone (underflow)
e	Altura do vortex catcher interna à parte cilíndrica do hidrociclone
f	Fator de correção relativo à fração das partículas sólidas coletadas no underflow
g	Fator relativo à concentração volumétrica de sólidos na alimentação
Lh1	Altura da parte cilíndrica do hidrociclone
Q	Vazão do fluido na operação do sistema de hidrociclones
Qa	Vazão do fluido na entrada do hidrociclone
Qr	Vazão do fluido na saída do hidrociclone
Re	Número de Reynolds
RL	Razão líquida entre as correntes de alimentação e do underflow
Stk50Eu	Representa o adimensional Stokes de corte multiplicado por Euler
u	Velocidade média do fluido na seção de entrada do hidrociclone

u_c	Velocidade média do fluido na seção cilíndrica do hidrociclone
ΔP	Diferença de pressão de operação
η	Eficiência global de coleta
η'	Eficiência global de coleta reduzida
μ	Viscosidade dinâmica do fluido
ρ	Massa específica do fluido
ρ_s	Massa específica do sólido particulado

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	14
1.1. INTRODUÇÃO	14
1.2. OBJETIVOS.....	16
1.2.1. Objetivo Geral.....	16
1.2.2. Objetivos Específicos	16
1.3. JUSTIFICATIVA.....	16
1.4. DELIMITAÇÃO DO TRABALHO	17
CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA.....	19
2.1. TECNOLOGIAS DIGITAIS.....	19
2.1.1. Visão Geral	19
2.1.2. Ferramentas Digitais	20
2.1.2.1. Programação e Linguagens	21
2.1.2.2. Unity Engine e Blender	22
2.1.3. Indústria 4.0	22
2.1.3.1. Simulação	24
2.2. OPERAÇÕES DE SEPARAÇÃO SÓLIDO-LÍQUIDO	25
2.2.1. VISÃO GERAL.....	25
2.2.2. HIDROCICLONES	26
2.2.2.1. Geometrias	27
2.2.2.2. Projeto e Dimensionamento	29
CAPÍTULO 3 – METODOLOGIA.....	33
3.1 METODOLOGIA DE SOLUÇÃO DA FERRAMENTA	33
3.2 DESIGN DA FERRAMENTA	38
3.3 A PROPOSTA DE APRENDIZAGEM.....	41
CAPÍTULO 4 – RESULTADOS E DISCUSSÕES	43

4.1 CAPACIDADE DE SOLUÇÃO DA FERRAMENTA E SUA DIDÁTICA COMO FERRAMENTA DE SIMULAÇÃO	43
4.2 COMPARAÇÃO DOS RESULTADOS OBTIDOS E O COMPORTAMENTO DO PROTÓTIPO TRIDIMENSIONAL	45
CAPÍTULO 5 – CONCLUSÃO	48
5.1 CONCLUSÃO.....	48
5.2 SUGESTÕES PARA TRABALHOS FUTUROS	49
REFERÊNCIAS BIBLIOGRÁFICAS	51
APÊNDICE A – QUESTÕES PROPOSTAS MODIFICADAS.....	54
APÊNDICE B – SOLUÇÃO DA PRIMEIRA QUESTÃO	55
APÊNDICE C – SOLUÇÃO DA QUARTA QUESTÃO.....	60
APÊNDICE D – CÓDIGO DA FERRAMENTA	63
ANEXO A – QUESTÕES PROPOSTAS POR RICARDO MEDRONHO	86

CAPÍTULO 1 – INTRODUÇÃO

Este capítulo apresenta os conceitos básicos utilizados e a motivação para o desenvolvimento deste trabalho. Ainda neste capítulo, encontram-se os objetivos gerais e específicos.

1.1. INTRODUÇÃO

Há uma crescente complexidade nos processos de indústrias de transformação, onde é necessário flexibilidade, redução de custos e maior sustentabilidade (HOFMANN e RUSCH, 2016). Devido a esse contexto, é possível que o ambiente acadêmico se distancie cada vez mais da situação tecnológica da indústria, altamente automatizada e digitalizada. Espera-se que os processos industriais evoluam para Sistemas Ciber-Físicos (*Cyber-Physical Systems* - CPS) (FERREIRO *et al.*, 2016), com o auxílio da indústria 4.0.

A indústria 4.0 impacta os meios de produção de forma a estimular o desenvolvimento e aplicação de diferentes tecnologias, por conta de um mercado altamente competitivo e complexo, principalmente nos setores de inovação. (COELHO, 2016). O uso de ferramentas tecnológicas e digitais se torna indispensável neste contexto, no qual a redução de custo, o aumento da produtividade e a versatilidade de produção são fundamentais para um posicionamento competitivo no mercado globalizado.

O desenvolvimento de ferramentas de uso industrial requer conhecimentos específicos e conceitos multidisciplinares. Alguns exemplos de tópicos que necessitam de multidisciplinaridade na elaboração e implementação dessas ferramentas são o planejamento de estratégias de controle automático; a verificação de componentes e sistemas de máquinas; a engenharia, programação, projeto e teste de softwares integrados e a realização de testes e avaliações de dependência (ZHANG, 2010).

Em “*Programming skills in the industry 4.0: are chemical engineering students able to face new problems?*”, Moisés Teles dos Santos sugere que o ensino de alguma linguagem de programação, como alternativa ao ensino de Microsoft Excel nos cursos de engenharia, é interessante para o desenvolvimento do raciocínio lógico e da familiaridade com programação em geral, duas habilidades fundamentais para a solução de problemas no atual contexto industrial. Há uma preocupação com relação ao rápido desenvolvimento da indústria, que parece estar se distanciando do ensino acadêmico devido à alta competitividade e velocidade

de mudanças observadas na indústria (FERREIRO *et al*, 2016; SANTOS *et al*, 2017; KIRILLOV *et al*, 2015).

As metodologias tradicionais de ensino não são mais eficientes ou velozes o bastante para a quantidade de conteúdo necessária para a atuação do engenheiro químico na indústria (KIRILLOV *et al*, 2015), além de não promoverem o contato com equipamentos ou situações reais ou virtuais que simulem situações reais. Simulação de processos costumava ser uma ferramenta para engenheiros químicos seniores, entretanto, há uma tendência de integração de novos tópicos aos cursos de engenharia que podem ser aprimoradas com o uso da simulação de processos como um auxílio pedagógico (DAHME, HESKETH e SVELSKI, 2002).

Outra tecnologia que pode ter aplicação didática é a impressão 3D. Scalfani e Vaid (2014) criaram um conjunto de modelos de moléculas complexas impressas em 3D que podem se provar úteis no ensino de isomeria espacial. A impressão em escala laboratorial de equipamentos industriais pode contribuir para apresentação do funcionamento destes equipamentos ou como base para uma situação de aprendizagem durante etapas educativas.

Mudanças nos processos organizacionais, nos métodos de trabalho e na estrutura de pessoas em organizações ocorreram por conta da quarta revolução industrial e com isso, as empresas correm o risco de não encontrar trabalhadores capacitados para a preparação, implementação e uso das novas tecnologias (PEJIC-BACH, 2018). Há uma necessidade de engenheiros mais capacitados e tecnologicamente atualizados, com conhecimentos das mais variadas ferramentas de trabalho. O treinamento prático dessas qualidades se torna cada vez mais indispensável no ambiente acadêmico frente à rápida propagação da quarta revolução industrial.

Os hidrociclones são equipamentos que separam partículas sólidas de líquidos e misturas de sólidos suspensos em líquidos (PEÇANHA, 2014). São relativamente simples, se comparados a outros equipamentos, por não necessitarem de partes eletrônicas e não envolverem reações químicas nem trocas térmicas. Por esses motivos, o projeto e dimensionamento de um hidrociclone pode ser prático, envolvendo as relações geométricas do próprio equipamento assim como as propriedades das partículas e do fluido a serem separados. A tecnologia utilizada no projeto e dimensionamento de hidrociclones é bastante consolidada tanto academicamente quanto comercialmente. Esses equipamentos são capazes de separar partículas de 5 μm a 100 μm , sendo compostos por materiais variados e construídos em diversas dimensões e podendo variar de 10 mm a 30 m de diâmetro, o que permite que cumpram seu propósito nas mais diferentes aplicações (SINNOTT e TOWLER, 2020).

Esse trabalho busca propor novas formas de educar e ensinar, utilizando ferramentas digitais que podem ser desenvolvidas com relativa simplicidade. O objetivo das ferramentas é tornar a aprendizagem mais eficiente, enquanto estimula o uso de conhecimentos multidisciplinares. A escolha de se trabalhar com hidrociclones se dá pela ausência de reações químicas e de trocas térmicas nas suas aplicações, tornando a execução da proposta mais simples.

1.2. OBJETIVOS

1.2.1. Objetivo Geral

O presente trabalho busca desenvolver uma ferramenta aplicável ao meio acadêmico, que auxilie os usuários a integrarem o conhecimento teórico aprendido à sua aplicação prática, introduzindo-os à prática de programação de softwares de simulação simples. O foco da ferramenta é a solução de problemas relacionados ao projeto e dimensionamento do hidrociclones, enquanto o foco da parte didática envolve a apresentação do funcionamento da ferramenta e de como deve ser usada.

1.2.2. Objetivos Específicos

- Desenvolver uma ferramenta digital didática capaz de auxiliar no projeto e dimensionamento de hidrociclones;
- Elaborar questões práticas que permitam o uso didático da ferramenta, com textos de fácil compreensão e relacionáveis à parte prática da ferramenta, assim como uma interface amigável e compreensiva;
- Resolver exercícios clássicos e dimensionar hidrociclones em comparação à solução manual, como forma de validação da ferramenta, para que possa auxiliar na compreensão da escrita de códigos.

1.3. JUSTIFICATIVA

O ambiente e as técnicas educacionais modernas não suprem a necessidade de aprendizagem nas áreas de criatividade e conhecimento multidisciplinar, requeridas pelo mercado de trabalho, não capacitando suficientemente os graduandos de engenharia química

para resolver os novos problemas complexos em um ambiente de alta velocidade de desenvolvimento e mudanças (KIRILLOV *et al*, 2015) impulsionado pela indústria 4.0.

A Indústria 4.0 já está causando a reestruturação de trabalhos em diversas áreas tecnológicas, principalmente na indústria de manufatura. A publicação de Pejic-Bach, em 2019, apresenta que dentre os cinco termos mais frequentes usados em anúncios de trabalho no LinkedIn em 2018, encontram-se Machine-Learning, Big Data e Computer Science.

Geoff Maitland, em 2014, classificou habilidades computacionais como a sétima dentre as 10 habilidades que engenheiros químicos deveriam estar falando sobre. Essa profissão está profundamente ligada ao mundo digital, por conta do viés industrial, e essa dependência indica que novos engenheiros químicos podem precisar estar tecnologicamente atualizados para exercer suas profissões.

Organizações e governos buscam melhorar o ensino de ciência, tecnologia, engenharia e matemática (STEM) (RIOJAS, 2012), investindo e estimulando o desenvolvimento de novas tecnologias e metodologias de aprendizagem. Ainda se discute se programação é um tema relevante para a engenharia química, pelos conhecimentos lógicos e disciplinares envolvidos (SANTOS, 2017). A velocidade em que a indústria e suas tecnologias se atualizam e desenvolvem deixa evidente que programação é um tema relevante.

Atividades cada vez mais complexas vêm se tornando fundamentais no exercício da profissão do engenheiro químico. Este trabalho busca desenvolver uma ferramenta, como mais uma proposta de auxílio na educação de engenheiros químicos, enquanto tenta atender às necessidades apresentadas.

1.4. DELIMITAÇÃO DO TRABALHO

A ferramenta desenvolvida deve ser capaz de solucionar problemas relacionados ao dimensionamento e projeto de hidrociclones, assim como as variáveis de operação relacionadas a estes hidrociclones, através de uma simulação. O programa utiliza principalmente as equações e metodologias propostas por Medronho, em 1984 e Antunes, em 1992, no processo de solução. Essas equações possuem parâmetros aplicáveis às configurações Rietema e Bradley, nas condições de uma alimentação com baixa concentração de partículas sólidas ($< 10\%$), utilizando o grupo adimensional $Stk50Eu$ (Stokes de Corte Reduzido Euler), que é aproximadamente constante para uma mesma família de hidrociclones.

Esta ferramenta não busca solucionar problemas relacionados a ciclones gasosos ou qualquer outro separador de líquidos ou de partículas sólidas. Além disso, o programa se limita

à solução de problemas que envolvam os hidrociclones das classes Rietema e Bradley e apenas trabalha com a associação de hidrociclones em paralelo, visto que a associação em série é pouco utilizada industrialmente.

CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA

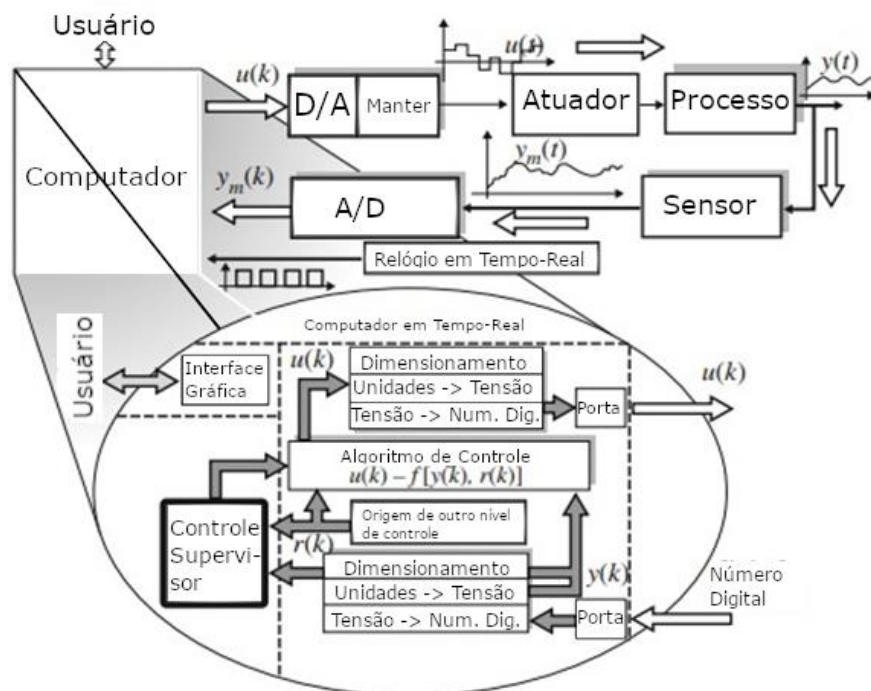
Este capítulo é dedicado à introdução dos principais fundamentos teóricos relevantes para a elaboração, compreensão e execução deste trabalho, além de apresentar ao leitor o estado da arte dos temas tratados.

2.1. TECNOLOGIAS DIGITAIS

2.1.1. Visão Geral

As tecnologias digitais podem ser compreendidas tanto a nível de *hardware* quando de *software*, compostas grandes conjuntos de portas lógicas e circuitos integrados. São compostas por sistemas e aparelhos eletrônicos que possam gerar, armazenar ou processar dados, em valores numéricos binários (1 e 0). É um campo amplo que contempla diversas áreas tecnológicas. O uso de sinais digitais para controle industrial é um exemplo de aplicação que substitui os sinais analógicos. A figura 1 apresenta um sistema digital de controle em tempo real, ilustrando sistemas digitais que possuem tanto componentes físicos e algoritmos programados em um software (ZHANG, 2010).

Figura 1 – Configuração de sistema de controle digitais em tempo real.



Fonte: Zhang (2010), adaptado

Os sistemas modernos de automação requerem melhor desempenho e precisão na interconexão de dispositivos de campo, vantagens oferecidas pela tecnologia de transmissão em redes digitais, em comparação à tecnologia analógica, mesmo que indústrias ainda utilizem transmissores de sinais elétricos de 4-20 mA principalmente por conta do seu custo reduzido (PARENTE, 2019).

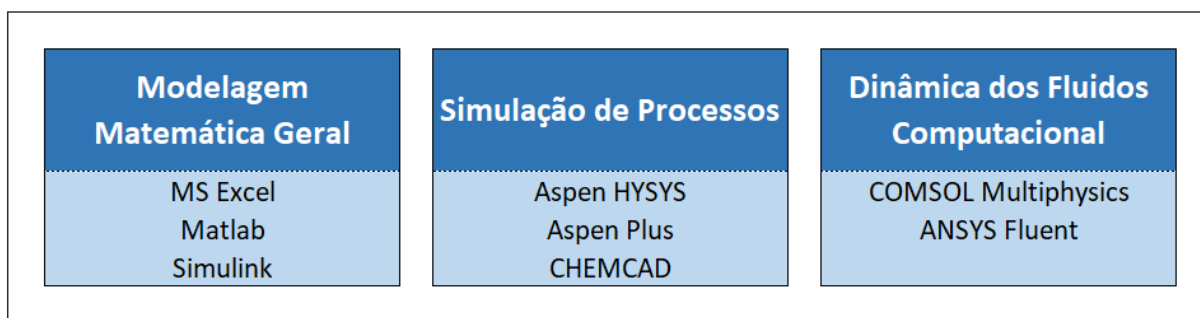
2.1.2. Ferramentas Digitais

O conceito que impulsiona a indústria 4.0 (I4.0) é o desenvolvimento de ferramentas cada vez mais capazes de trabalhar com a automatização avançada de processos, conectando o mundo físico ao digital, aumentando produtividade e eficiência. Processos complexos são controlados por sistemas que garantem confiabilidade e segurança, além de servir de base para estratégias avançadas de manutenção (SIRKKA, 2007). A I4.0, busca usar e desenvolver Sistemas Ciber-Físicos (*Cyber-Physical Systems - CPS*), que se baseiam na integração de dados e informações entre sistemas físicos e digitais. O conceito de *CPS* pode ser sumarizado como processos interoperáveis, integrados e otimizados que integram o uso de algoritmos, Big Data, Internet das Coisas (*Internet of Things - IoT*) e Serviços (*Internet of Services - IoS*), automação industrial, Cibersegurança e outras tecnologias (ALCÁCER E CRUZ-MACHADO, 2018).

As indústrias de transformação estão se tornando mais complexas e seus requisitos, mais rigorosos. Técnicas clássicas de agregação de valor não parecem mais ser suficientes, forçando indústrias a inovarem em diversos segmentos do processo, para que possam competir contra o aumento da eficiência, flexibilidade, redução de custo e sustentabilidade observada nos processos industriais recentes (HOFMANN E RUSCH, 2016).

Ivana Lukec publicou no site *Simulate Live*, em 2019, o texto “*What is the Most Useful Software in Chemical Engineering?*”, apontando alguns softwares de alta utilidade em diversas áreas de atuação de um engenheiro químico. Na figura 2 encontra-se uma lista de softwares para fins específicos.

Figura 2 – Softwares mais úteis na engenharia química.



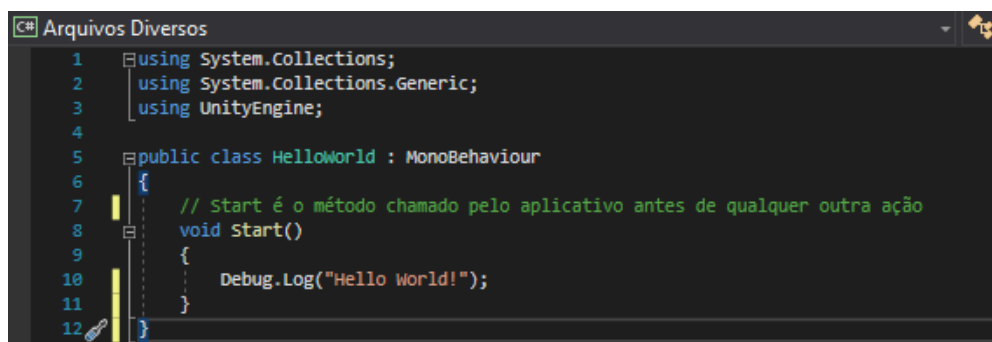
Fonte: Lukec (2019), adaptado.

2.1.2.1. Programação e Linguagens

Programar é o ato de instruir computadores a cumprirem tarefas. Uma sequência de instruções executadas por qualquer aparelho capaz de processar código (como celulares e caixas automáticos), pode ser chamada de programa. Há padrões e rotinas no cotidiano de qualquer indivíduo e, similar a receitas culinárias, programas cumprem rotinas que lhe foram instruídas (ADESANYA, 2017). Programas de computador são vistos de forma similar a receitas culinárias; indivíduos os instruem, definindo padrões e rotinas que devem ser cumpridas por um computador. Essas sequências devem ser escritas de forma que possam ser compreendidas por tal computador, a partir de uma linguagem de programação.

O C# (C-Sharp) é uma linguagem orientada a objetos, que busca ser versátil e segura para o desenvolvimento de aplicativos e softwares executados a partir da plataforma de desenvolvimento chamada .NET, que permite que qualquer código gerado para esta plataforma seja executado em qualquer dispositivo que possua alguma estrutura .NET. A linguagem C# pode ser usada na criação de aplicativos para Windows, Web XML, aplicativos de banco de dados e vários outros fins. O C# é facilmente reconhecível pelas semelhanças que possui com C, C++ e Java, enquanto busca ser simples, seguro e rápido (MICROSOFT, 2020). O C# é a linguagem usada nas versões mais recentes do software Unity® e a figura 3 apresenta como fazer o *console* do Unity® imprimir a mensagem “Hello world!”, no início do programa.

Figura 3 – Exemplo de código para impressão da frase "Hello World!" no console do Unity.



```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class HelloWorld : MonoBehaviour
6  {
7      // Start é o método chamado pelo aplicativo antes de qualquer outra ação
8      void Start()
9      {
10         Debug.Log("Hello World!");
11     }
12 }

```

Fonte: Autoria própria

2.1.2.2. Unity Engine e Blender

A plataforma de desenvolvimento 3D em tempo real do Unity® permite que artistas, designers e desenvolvedores criem experiências audiovisuais (UNITY TECHNOLOGIES, 2020). A plataforma tem como principal alvo o desenvolvimento de conteúdo audiovisual, como animações e jogos digitais. Exemplos de animações e jogos são *Cycles*, da *Walt Disney* e *Hollow Knight*, da *Team Cherry*. Um software de simulação pode ser comparado a jogos digitais, considerando que ocorre uma interação entre o usuário e a função do software programado.

O Blender é um ambiente de criação 3D de código aberto, no qual é possível realizar modelagem, animação, simulação, renderização e até edição de vídeo e desenvolvimento de jogos (THE BLENDER FOUNDATION, 2020). O seu maior diferencial é o fato de ser um software de código aberto, sendo mais conhecido pelas funções relacionadas à modelagem 3D que possui. É comum o uso do Blender na criação de modelos para animações e jogos de pequenos produtores.

2.1.3. Indústria 4.0

A Alemanha, em 2011, trouxe ao mundo um novo tópico de discussão que foi chamado de Indústria 4.0, que faz referência à quarta revolução industrial. Seu objetivo é atingir uma profunda automação dos processos, aumentando a eficiência e produtividade das indústrias, através de uma conexão do mundo físico ao virtual (ALCÁCER E CRUZ-MACHADO, 2018).

Exemplos de tecnologias impulsionadas pelo conceito da indústria 4.0 estão na solução de Controle Avançado de Processos desenvolvida pela RADIX para a Ambev, aprimorando a

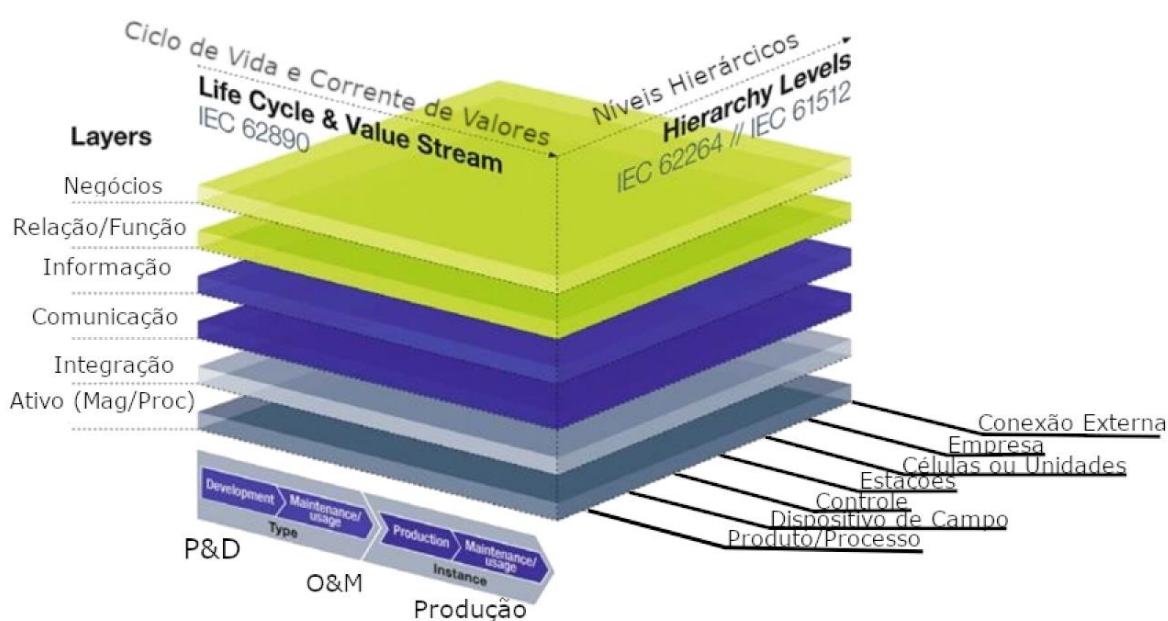
qualidade da cerveja e reduzindo seu custo de produção. O sistema desenvolvido controla o a etapa de evaporação do processo de forma precisa, mesmo sob perturbações externas (RADIX, 2019).

A automação avançada e a extensa digitalização dos processos de produção, desenvolvidas a partir de eletrônicos complexos e novas tecnologias da informação, é o que caracteriza a Indústria 4.0. O foco da I4.0 é o estabelecimento de sistemas inteligentes e comunicativos, que lidem com o fluxo de dados originados da interação e distribuição de diversos sistemas inteligentes (ALCÁCER E CRUZ-MACHADO, 2018).

O Ministério da Indústria, Comércio e Serviços atribui às três primeiras revoluções industriais características como a produção em massa, as linhas de montagem e a tecnologia da informação. Essas características tornaram a competição tecnológica o cerne do desenvolvimento econômico e, frente ao desenvolvimento tecnológico da quarta revolução industrial, um impacto ainda mais profundo é esperado (MINISTÉRIO DA INDÚSTRIA, COMÉRCIO E SERVIÇOS, 2019).

Diversas instituições e associações alemãs cooperaram no desenvolvimento do modelo de referência da arquitetura *RAMI4.0* (*Reference Architectural Model Industrie 4.0*) para a indústria 4.0. Este modelo, apresentado na figura 4, é representa uma estrutura padronizada dos fundamentos da indústria 4.0 e tem como objetivo auxiliar a implementação de tecnologias 4.0 na indústria (ALCÁCER E CRUZ-MACHADO, 2018).

Figura 4 – Modelo de Referência RAMI4.0



Fonte: Alcácer e Machado-Cruz (2018)

Dentre várias tecnologias, pode-se destacar a Manufatura Aditiva (AM), a Inteligência Artificial (AI), a Internet das Coisas (IoT), a Biologia Sintética e os Sistemas Ciber-Físicos (CPS) como as principais tecnologias que auxiliam na fusão dos mundos físico, digital e biológico. (MINISTÉRIO DA INDÚSTRIA, COMÉRCIO E SERVIÇOS, 2019).

A Manufatura Aditiva, também conhecida por Manufatura por Adição de Camada, Rápida Prototipagem e Manufatura Digital, é formada por um conjunto de tecnologias que permitem a impressão tridimensional de objetos físicos (ALCÁCER E CRUZ-MACHADO, 2018). Embora ocorram aprimoramentos no funcionamento técnico da tecnologia, existem preocupações de que a falta da especialização adequada e o baixo desenvolvimento das habilidades necessárias sejam fatores que atrasam tanto o desenvolvimento da tecnologia quanto a sua adoção no geral (FORD E MINSHALL, 2019).

A integração da Internet das Coisas e de Sistemas Ciber-Físicos às logísticas pode permitir monitoramento em tempo real dos materiais no processo e a tornar o gerenciamento de risco mais preciso (HOFMANN E RUSCH, 2016). Especialmente no contexto industrial, a Internet das Coisas Industrial é a conexão de componentes e máquinas à internet para uma melhor integração de dados e análise em tempo real (ALCÁCER E CRUZ-MACHADO, 2018).

Implementações educacionais de softwares no ensino de engenharia, incluindo ferramentas de desenvolvimento, modelagem, simulação e ambientes de programação oferecem um custo reduzido para o ensino de conceitos que, tradicionalmente, exigiriam a aquisição de materiais dispendiosos (RIOJAS *et al*, 2012).

2.1.3.1. Simulação

A simulação é o uso de técnicas matemáticas que imitam o funcionamento de uma operação, com algum nível de precisão, a partir de modelos matemáticos. A simulação computacional auxilia a definir quais produtos, processos e sistemas são mais atrativos, em diversas áreas tecnológicas. (ALCÁCER E CRUZ-MACHADO, 2018).

Ferramentas de simulação não computacionais evoluíram bastante desde sua origem e hoje são bastante utilizadas em diversas áreas e indústrias, como saúde, manufatura e entretenimento. Essas ferramentas de simulação trazem muitas vantagens para o ambiente educacional e são comumente utilizadas para ensinar e treinar indivíduos (GEORGE BROWN COLLEGE, 2018).

A simulação computacional busca prever o comportamento de um sistema real, usualmente físico, a partir da modelagem matemática desse sistema. O uso da modelagem e

simulação (*M&S*) permite que indivíduos e empresas compreendam o comportamento de sistemas hipotéticos e reais além da teoria e da experimentação, usando a matemática. As aplicações da *M&S* abrangem quase todos os aspectos da economia, da sociedade e do desenvolvimento militar, sendo encontrada no tratamento medicinal, na informação, na engenharia de trânsito, na manufatura, agricultura, educação e muitos outros (ZHANG *et al*, 2019).

Ferramentas e métodos de Engenharia de Sistemas de Processos (*PSE – Process System Engineering*) são aplicados no desenvolvimento, na análise, desenvolvimento, avaliação e otimização de processos farmacêuticos, além de indústrias petroquímicas. Verificações finais são necessárias e realizadas através de experimentos, entretanto, o auxílio computacional permite uma avaliação rápida e remoção de opções redundantes a fim de agilizar, simplificar e aprimorar planejamentos de projetos, de forma confiável e eficiente (ALCÁCER E CRUZ-MACHADO, 2018).

É comum que os tópicos cobertos por simulações sejam de alta complexidade e incerteza. Essas características fazem estes objetos de pesquisa complicados de serem analisados por métodos teóricos tradicionais (ZHANG *et al*, 2019). O nível de competitividade atual força que sejam utilizados métodos capazes de realizar estimativas confiáveis e cálculos teóricos com maior precisão e velocidade.

Ferramentas de simulação para aprendizagem já são usadas em diversas áreas, como no treinamento de pilotos e em áreas médicas. Matemáticos e economistas utilizam simulação para modelar e analisar probabilidades e estatísticas. Plantas de manufatura utilizam simulação para projetar e até treinar processos automatizados (GEORGE BROWN COLLEGE, 2018). Também permite a criação de dados que simulem um sistema de forma fiel o suficiente para que sejam fornecidos a sistemas de aprendizado de máquina.

2.2. OPERAÇÕES DE SEPARAÇÃO SÓLIDO-LÍQUIDO

2.2.1. VISÃO GERAL

A separação de sólidos de um meio líquido é comumente a última etapa em processos de correntes lamacentas, onde ocorre a separação do particulado a partir de um meio de suspensão. As partículas são postas sob a ação de campos ou gradientes de forças, para que sejam separadas do líquido com base nas características relevantes para o sistema. Normalmente, o campo do sistema é gravitacional ou centrífugo (separação pela massa da

partícula), elétrico (separação pela carga) ou magnético (separação por afinidade magnética) (PEKER, 2008). Os métodos mais utilizados industrialmente para a separação de partículas são as separações mássicas.

O dimensionamento de equipamentos industriais depende de valores financeiros, das proporções das fases e das propriedades físicas que influenciam o movimento relativo do líquido e das partículas. Em separações sólido-líquido, é comum que a principal variável de operação seja a pressão e seu efeito deve ser conhecido (COUPER *et al*, 2009). No caso dos hidrociclones, a pressão de operação influencia diretamente o diâmetro de corte do sistema.

2.2.2. HIDROCICLONES

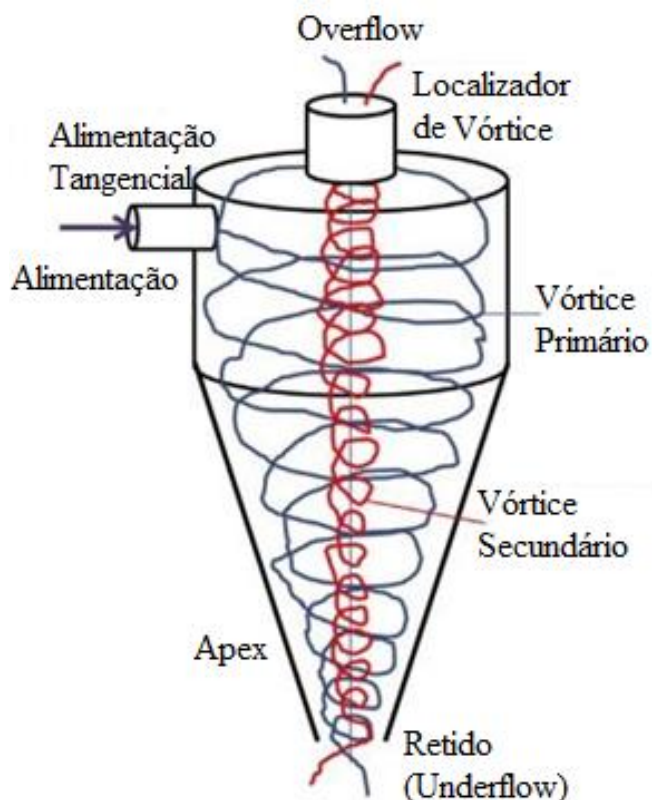
Os hidrociclones estão classificados como separadores sólido-líquido que agem através de campos de velocidade centrífuga, separando as partículas com base em suas massas e propriedades relacionadas a massa, como porosidade e massa específica. Também são chamados por ciclone hidráulico, ciclone líquido, hidroclone e ciclone (PEÇANHA, 2014).

Hidrociclones operam por inércia, similarmente a câmaras de poeira, ciclones e centrífugas: com base no fato de que a densidade das partículas sólidas suspensas é maior que a densidade do fluido no qual estão suspensas, na presença de um campo de velocidades. A alimentação do hidrociclone se dá de forma tangencial à sua câmara cilíndrica, o que estabelece sua força centrífuga. (PEÇANHA, 2014). Essa entrada é observada na figura 5, que representa a operação de um hidrociclone.

A entrada tangencial do hidrociclone possibilita a formação de um vórtice em seu interior. A separação ocorre na direção radial, por materiais mais densos tenderem a migrar para as paredes da região cônica, por conta da aceleração centrífuga da partícula e do líquido (ALDRICH, 2015).

No interior da câmara cilíndrica de um hidrociclone ocorrem dois vórtices, que fluem em direções opostas: Um para o topo, saída chamada de *overflow*, com as partículas de sólidos mais leves e outro para o fundo, saída chamada de *underflow*, com as partículas mais pesadas (ALDRICH, 2015). As saídas *overflow* e *underflow* estão representadas, respectivamente, na figura 5.

Figura 5 – Esboço da operação de um hidrociclone



Fonte: ALDRICH (2015), adaptado.

Hidrociclones possuem um design simples, baixo custo, fácil operação e pouca necessidade de manutenção. Por esses motivos, se destacaram na indústria e vêm sendo usados na indústria desde 1940 para a separação de sólidos e líquidos (ALDRICH, 2015).

2.2.2.1. Geometrias

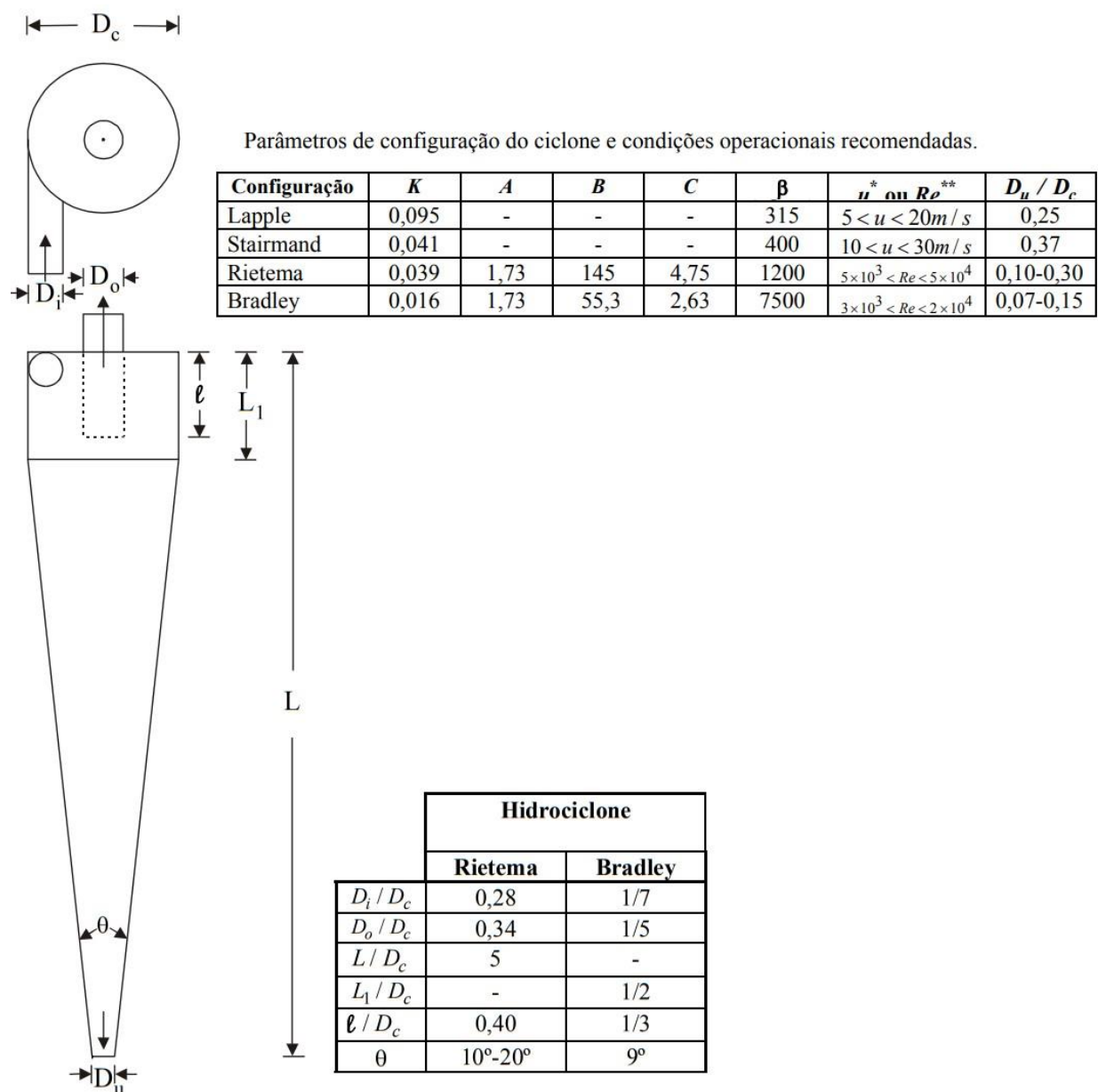
Diversos modelos de hidrociclones já foram propostas com o objetivo de explorar da forma mais eficiente possível as forças inerciais e gravitacionais, para a separação de fluidos com partículas em suspensão (ALDRICH, 2015). Ainda assim, as geometrias mais conhecidas academicamente são as geometrias de Rietema e Bradley, que serão utilizadas nesse trabalho. As geometrias de Rietema e de Bradley são entendidas como configurações de pesquisa e se diferem das utilizadas na indústria (MASSARANI, 2002).

A configuração de um ciclone é caracterizada pela relação entre suas dimensões, normalmente expressas em função do diâmetro da parte cilíndrica, chamada de D_c (MASSARANI, 2002). A figura 6 apresenta um desenho de um ciclone, em duas vistas, além

de duas tabelas que mostram diferentes parâmetros e relações usadas no dimensionamento e projeto de ciclones das respectivas classes. Nota-se que, embora apresentadas na figura, as configurações Lapple e Stairmand são características de ciclones gasosos, o que os faz diferentes de hidrociclones, inclusive a ponto de não possuírem parâmetros A, B e C.

As equações de dimensionamento de um hidrociclone serão apresentadas no próximo item deste trabalho, onde também se discutirá as relações geométricas e o dimensionamento e projeto do hidrociclone.

Figura 6 – Parâmetros de dimensionamento de ciclones e hidrociclones



Fonte: MASSARANI (2002)

2.2.2.2. Projeto e Dimensionamento

Neste tópico, serão discutidas as equações que associam parâmetros como diâmetro de corte, propriedades físicas do sistema, dimensões do equipamento e condições operacionais, a função da eficiência de coleta relativa à partícula de diâmetro D , a expressão para a eficiência global de coleta e a equação que relaciona vazão e queda de pressão no ciclone, com base no livro “Fluidodinâmica em Sistemas Particulados”, escrito por Giulio Massarani, em 2002; na tese de PhD de Ricardo de Andrade Medronho, intitulada “*Computer-aided grade efficiency evaluation and tests to verify a new theory for scale-up of hydrocyclones with a continuously variable inlet orifice*” e na publicação “*Bradley Hydrocyclones: Design and Performance Analysis*” de Mônica Antunes Pereira da Silva e de Medronho.

A equação (1) informa o diâmetro de corte (d_{50}), em função do diâmetro da parte cilíndrica e outros parâmetros. Os parâmetros D_c , μ e Q representam o diâmetro da parte cilíndrica do ciclone, a viscosidade e a vazão de fluido de alimentação respectivamente. Já f é um fator de correção relativo à coleta das partículas sólidas no *underflow*, relacionado à razão líquida (RL), que relaciona as vazões de fluido no *underflow* e na alimentação. O parâmetro g é um fator de correção relativo a concentração volumétrica de sólidos na alimentação (C_v). Por fim, K é um parâmetro dependente da configuração do hidrociclone (MASSARANI, 2002).

$$\frac{d_{50}}{D_c} = K \left[\frac{\mu D_c}{Q(\rho_s - \rho_f)} \right]^{\frac{1}{2}} \cdot f(R_L) \cdot g(C_v) \quad (1)$$

Na tabela 1 podem ser encontrados os valores dos parâmetros A , B , C , K e β para cada hidrociclone, além de às quais valores de Reynolds eles se aplicam e as proporções para D_c/D_u para os tipos de configurações Rietema e Bradley. Nota-se que há restrições para seus valores, os fazendo válidos apenas para as condições de “ u ” ou “ Re ” apresentadas. “ Re ” é referente ao Número de Reynolds e “ u ” é a velocidade média do fluido na seção de entrada do ciclone, dada pela equação (2), Re , pela equação (3) e u_c representa a velocidade média do fluido na seção cilíndrica do ciclone, obtida por (4) (MASSARANI, 2002).

$$u = \frac{Q}{B_C H_C} \quad (2)$$

$$Re = \frac{D_C u_c \rho_F}{\mu} \quad (3)$$

$$u_c = \frac{Q}{\pi D_C^2 / 4} \quad (4)$$

Tabela 1 – Parâmetros de configuração de hidrociclones

Configuração	K	A	B	C	β	Re	Du/Dc
Rietema	0,039	1,73	145	4,75	1200	$5 \times 10^3 < Re < 2 \times 10^4$	0,10-0,30
Bradley	0,016	1,73	55,3	2,63	7500	$3 \times 10^3 < Re < 2 \times 10^4$	0,07-0,15

Fonte: Massarani (2002), adaptado

A grande parte das teorias de separação sólidos em hidrociclones operados à baixas concentrações indicam que o grupo adimensional $Stk50Eu$ deve ser constante para todo hidrociclone similar à configuração que pertence (MEDRONHO, 1984). Este grupo adimensional é encontrado através da multiplicação entre o $Stk50$ (Stokes de corte) e o número de Euler do sistema. Medronho sugere que, de fato, $Stk50Eu$ não parece alterar com a diferença de pressão ou com o diâmetro de entrada, mas é afetado pela concentração da corrente de alimentação (C_v) e pela razão líquida (RL) do sistema. No mesmo trabalho, ele propõe uma correlação para $Stk50Eu$ que utilize RL e C_v , para baixas concentrações ($C_v < 10\%$) em hidrociclones da família Rietema. Essas correlações possuem um conjunto de parâmetros que são utilizados também na obtenção do número de Euler e da vazão líquida.

Tabela 2 – Primeiro conjunto de parâmetros obtidos por Medronho e Antunes para hidrociclones Rietema e Bradley

Configuração	n1	n2	n3	n4	n5	n6	n1
Rietema	0.742	8.96	0.116	-2.12	4.75	-0.3	0.742
Bradley	0.66	12.0	0.37	0.0	2.63	-1.12	0.66

Fonte: Autoria própria

Tabela 3 – Segundo conjunto de parâmetros obtidos por Medronho e Antunes para hidrociclones Rietema e Bradley

Configuração	k1	k2	k3
Rietema	0.0474	371.5	1218
Bradley	0.055	258	1.21x10 ⁶

Fonte: Autoria própria

Em 1992, Antunes e Medronho publicaram um trabalho intitulado “*Bradley Hydrocyclones: Design and Performance Analysis*”, onde aplicaram a metodologia proposta por Medronho à família de hidrociclones Bradley. Foram testadas concentrações volumétricas (Cv) de 0% a 10%, encontrando parâmetros aplicáveis às correlações de Stk50Eu, Eu e RL, similares aos parâmetros obtidos por Medronho, que podem ser consultados nas tabelas 2 e 3 e ter suas aplicações observadas nas equações (5), (6) e (7).

$$Stk_{50}Eu = k_1 \left[\ln \left(\frac{1}{RL} \right) \right]^{n1} e^{(n2 C_{va})} \quad (5)$$

$$Eu = k_2 Re^{n3} e^{(n4 C_{va})} \quad (6)$$

$$RL = k_3 \left(\frac{D_U}{D_C} \right)^{n5} Eu^{n6} \quad (7)$$

Tipicamente, distribuições de tamanhos de partículas (ou distribuições granulométricas) podem ser representadas por várias funções matemáticas. Alguns exemplos de modelos matemáticos aplicáveis são a Distribuição Log-Normal, a Distribuição Gaudin-Schuhmann e a Distribuição Rosin-Rammler (JILLAVENKATESA *et al.*, 2001). Essas distribuições foram muito estudadas e atualizadas, sofrendo inclusive alterações em suas nomenclaturas. A chamada Rosin-Rammler também é conhecida como Distribuição Weibull, com a diferença de um parâmetro a menos (MEDRONHO, 1984). A Distribuição Rosin-Rammler-Bennet ou RRB, é o modelo de distribuição de partículas utilizado pela ferramenta neste trabalho.

A distribuição granulométrica do material particulado na alimentação do sistema é uma representação da eficiência do hidrociclone para cada diâmetro de partícula encontrada na alimentação do sistema. Com o conhecimento da distribuição granulométrica, pode-se utilizar seus valores de diâmetro e eficiência (d, y) na função linearizada de RRB, em conjunto do diâmetro de corte reduzido para calcular a eficiência total do hidrociclone. O diâmetro de corte reduzido indica o diâmetro de partícula em que o hidrociclone será capaz de separar com 50%

de eficiência. A equação (8) apresenta o modelo de distribuição Rosin-Rammler, enquanto a equação (9) representa o modelo utilizado pela ferramenta, com base no modelo de eficiência de coleta reduzida proposto por Medronho, originado no modelo RRB. A equação (9) se refere a eficiência de coleta da partícula definida, sendo possível a obtenção da eficiência do hidrociclone a partir da integração dos pontos da função.

$$F_M(x) = 1 - \exp - \left(\frac{x}{x_{RR}} \right)^n \quad (8)$$

$$G'_{(x/x'_{50})} = 1 - \exp \left[- 0,693 \left(\frac{x}{x'_{50}} \right)^n \right] \quad (9)$$

CAPÍTULO 3 – METODOLOGIA

Este capítulo é dedicado à apresentação da metodologia utilizada para o desenvolvimento e conclusão do trabalho. O desenvolvimento da HidroCSim pode ser resumido em duas etapas: o desenvolvimento do processo de solução da ferramenta e o desenvolvimento do design da ferramenta e da proposta de aprendizagem. A primeira etapa é caracterizada pela escrita e documentação do código, enquanto a segunda etapa é caracterizada pela construção da apresentação da ferramenta e sua experiência de uso. O link para download da ferramenta e de seu projeto no Unity se encontra no apêndice A.

3.1 METODOLOGIA DE SOLUÇÃO DA FERRAMENTA

Todas as variáveis são consideradas nulas pela ferramenta até que o usuário preencha os devidos campos das variáveis das quais ele possui conhecimento, para solução do problema. Com base nos valores inseridos, a ferramenta irá discernir quais variáveis ela conhece e encontrará os valores das outras variáveis com base nas equações propostas por Medronho. Os valores são salvos toda vez que o usuário confirma com a tecla *Enter* ou alterna campos para a inserção de um novo valor. A interface do painel geral pode ser observada na figura 7, sendo as opções de início de solução encontradas no canto direito inferior.

Figura 7 – Painel geral da ferramenta HidroCSim

The interface is titled "Insira os valores conhecidos:". It contains the following input fields and controls:

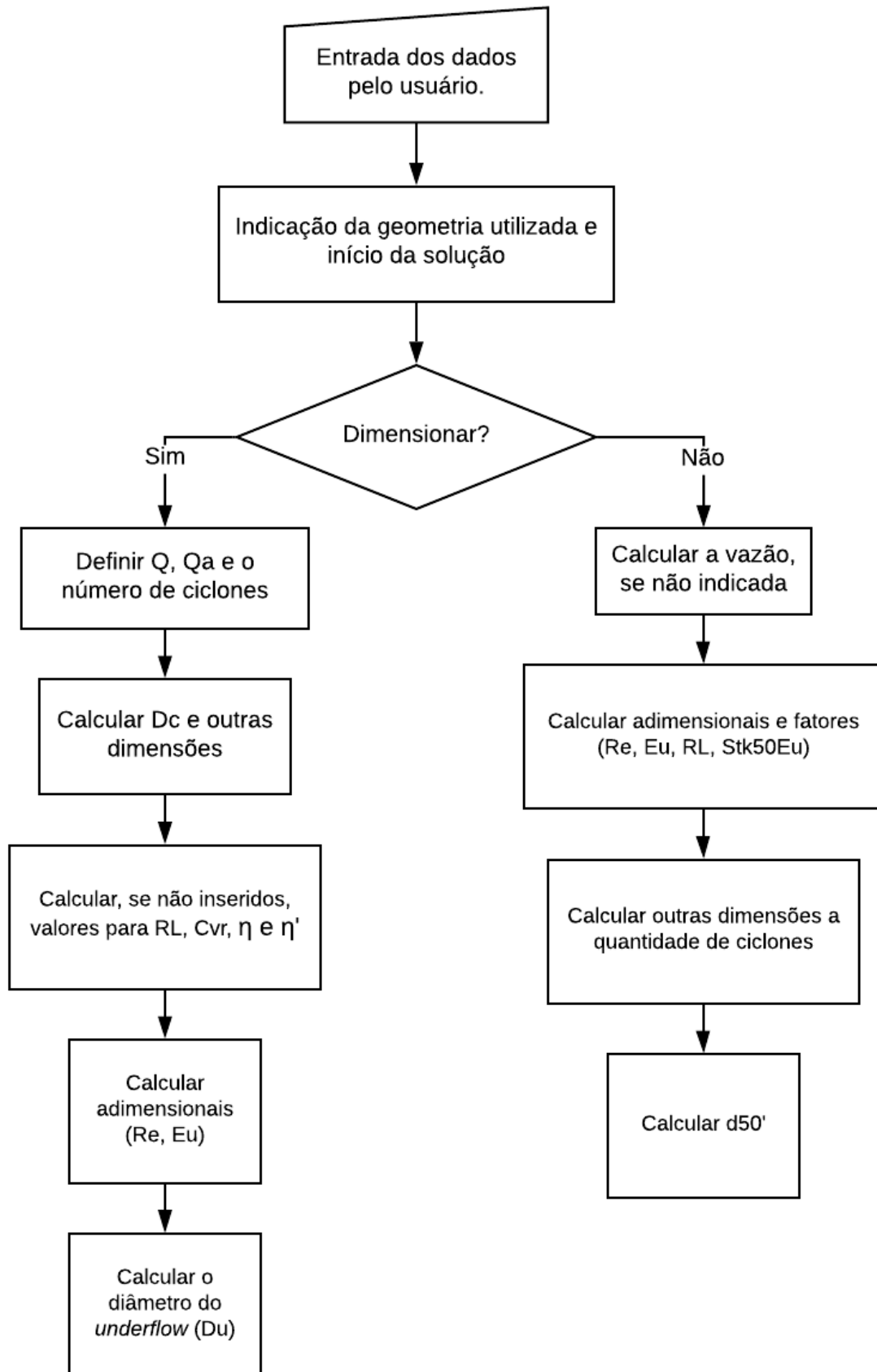
- Top Left:** "Painel DTA" button.
- Top Right:** "Resultado" button.
- Input Fields (Grid):**
 - Row 1: dP: dP (Pa), Cva: Cva, Dc: Dc (m), Lh: Lh (m)
 - Row 2: μ: μ (Pa.s), Q: Q (m³/s), Du: Du (m), Lh1: Lh1 (m)
 - Row 3: ρ: ρ (Kg/m³), Qa: Qa (m³/s), Do: Do (m), RI: RI
 - Row 4: ρS: ρS (Kg/m³), Qr: Qr (m³/s), Di: Di (m), Nº de Hidrociclones: Nº de Ciclones
 - Row 5: η: Et (<= 1), Cvr: Cvr
- Bottom Center:** "Reiniciar" button.
- Bottom Right (Soluções):**
 - "Rietema" button
 - "Bradley" button
 - "Descobrir" button
 - Checkbox "Dimensionar"
- Bottom Left:** "Visual" button.

Fonte: Autoria própria

O processo de solução se inicia quando o usuário escolhe uma das geometrias, no canto inferior da interface, definindo quais os valores relacionados à geometria são adequados para uso em várias das equações no processo de solução. Caso o usuário opte pela opção “Descobrir”, a ferramenta deve comparar as dimensões conhecidas às proporções de ambas as configurações de hidrociclones. Caso os valores sejam coerentes com as proporções das configurações Rietema e Bradley, dentro de uma margem, ela irá indicar qual geometria deve ser usada ou, se não for possível a identificação, não iniciar a solução. A figura 7 apresenta o fluxograma referente à metodologia de solução da ferramenta.

Quando iniciado o processo de solução, a ferramenta irá tentar calcular todos os resultados possíveis com as variáveis inseridas, conforme o fluxograma da figura 7. Quando não for possível calcular algum resultado, a etapa será pulada e o processo continuado. Todo resultado pode ser consultado na aba de resultados, inclusive os que não forem possíveis de calcular, os quais serão considerados equivalentes a zero. Outras soluções podem ser iniciadas a qualquer momento, após alterar ou não valores. Adicionalmente, na aba DTA, pode-se inserir valores relativos à distribuição granulométrica da composição de entrada. Desta forma, pode-se calcular a eficiência de operação com base no diâmetro de corte reduzido (d_{50}'), que também pode ser obtido através da solução inicial.

Figura 8 – Fluxograma do processo de solução da ferramenta



Fonte: Autoria própria

A solução é caracterizada por uma série de condicionais que irão discernir o que a ferramenta é capaz de encontrar com base nas variáveis conhecidas. As variáveis ΔP , μ e ρ , consideradas fundamentais, são verificadas e representam a diferença de pressão de operação do hidrociclone, a viscosidade dinâmica e a massa específica do fluido.

As etapas seguintes irão diferir caso o usuário marque a opção “Dimensionar” ou não. Na figura 8 pode-se observar o painel geral da ferramenta, com a opção dimensionar evidenciada. Se marcada, a ferramenta tentará concluir as instruções com as informações que possui. Caso não esteja marcada e não haja um valor para D_c , a solução não será iniciada.

Figura 9 – Painel geral da ferramenta, com a opção “Dimensionar” evidenciada

Fonte: Autoria própria

Além das variáveis críticas, outras variáveis são importantes na obtenção de resultados específicos. Por exemplo, para que se saiba a vazão de operação do hidrociclone (Q_a) a partir de diâmetro de centro (D_c), é necessária também a concentração volumétrica de entrada (C_v). Os quadros 1 e 2 indicam quais variáveis podem ser calculadas a partir de quais outras variáveis. O primeiro quadro considera o conhecimento de D_c e D_u e busca encontrar resultados para a operação, enquanto o segundo quadro busca encontrar as dimensões do hidrociclone. Vale ressaltar que, por ser um processo ordenado, várias variáveis são obtidas em sucessão e, por isso, não precisam ser declaradas nos campos de entrada do usuário. Por exemplo, a ferramenta prioriza a obtenção das variáveis Q_a e D_c , logo, caso seja possível a obtenção de Q_a e/ou D_c , essas variáveis serão encontradas logo no início da solução e serão utilizadas no cálculo das outras variáveis.

Quadro 1 – Variáveis necessárias para o projeto de hidrociclones pela ferramenta, em ordem de solução

Variável a ser calculada	Variáveis necessárias para cálculo
Qa	Dc, ρ , μ , ΔP , Cva
Re	ρ , Qa, μ , Dc
Eu	Re, Cva ($<0,1$)
RL	Dc, Du
Stk50Eu	RL, Cva
Di/Do/Lh/e	Dc, Configuração (Rietema/Bradley)
d50'	ρ_s , Stk50Eu, μ , ρ , Qa, ΔP , Dc
η'	Pontos da distribuição granulométrica
η	η' , RL
Cvr	η , Cva, RL

Fonte: Autoria própria

Quadro 2 – Variáveis necessárias para o dimensionamento de hidrociclones pela ferramenta, em ordem de solução

Variável a ser calculada	Variáveis necessárias para o cálculo
Qa	Q, N° de Hidrociclones
Dc	Qa, ρ , μ , ΔP , Cva
Di/Do/Lh/e	Dc, Configuração (Rietema/Bradley)
RL	η , Cva, Cvr
Cvr	η , Cva, RL
η	Cva, Cvr, RL
Re	ρ , Qa, μ , Dc
Eu	Re, Cva ($<0,1$)
Du	Dc, RL, ΔP , ρ , Qa

Fonte: Autoria própria

De forma a evitar *bugs* e garantir o funcionamento intencional da ferramenta, recomenda-se reiniciar as variáveis através do botão “Reiniciar” e adicioná-las novamente. No

caso de funcionamento inesperado da ferramenta ou, para melhor garantia do funcionamento adequado, pode-se reiniciar a ferramenta por completo.

3.2 DESIGN DA FERRAMENTA

A primeira interface da ferramenta é composta por um conjunto de campos que aceitam valores numéricos, como visto nas figuras 8 e 10. Todo campo aceita apenas valores numéricos e considera vírgulas como os devidos separadores decimais. O programa não aceita letras nos campos e será interrompido caso recebam alguma letra ou caractere especial em qualquer campo. É possível consultar duas vistas, a do painel, onde são inseridos os valores, e a do hidrociclone, composto por um conjunto de peças que compõem um esboço de um hidrociclone parcialmente fiel às dimensões encontradas na aba de resultados. Ambas as vistas estão representadas na figura 8 e só podem ser consultadas individualmente.

Figura 10 – Painel geral da ferramenta e vista do esboço do hidrociclone

The image shows a software interface titled "Painel DTA" with the instruction "Insira os valores conhecidos:". It contains several input fields for parameters:

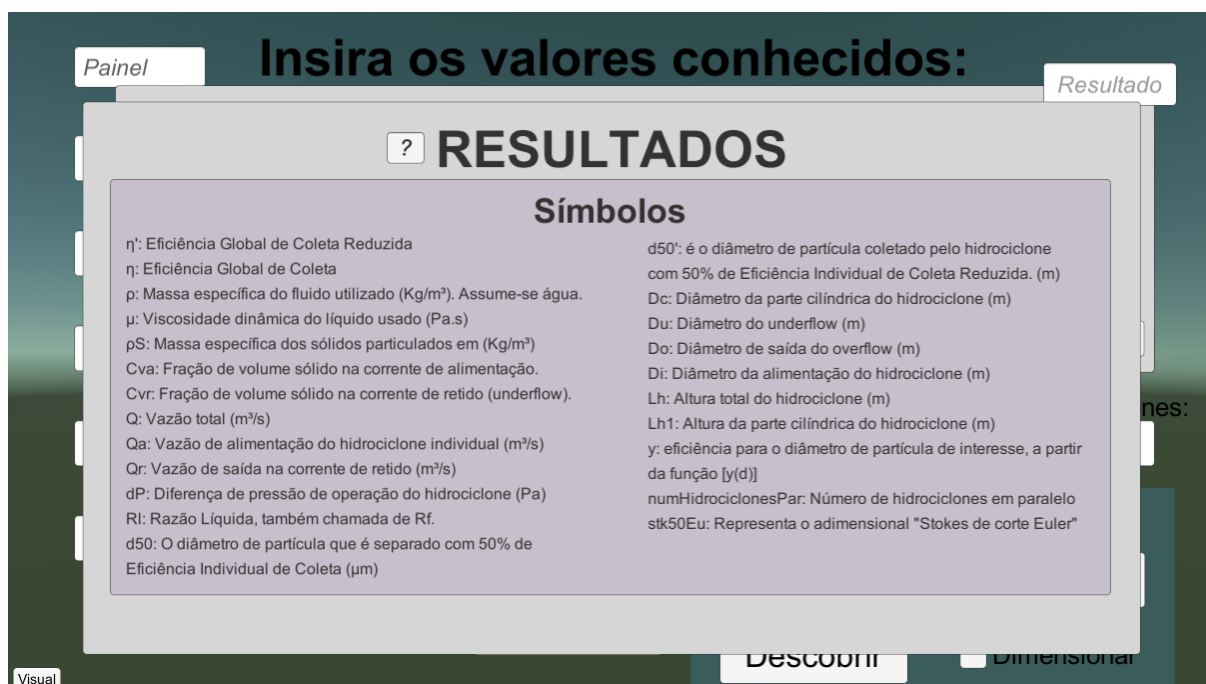
- dP:** dP (Pa)
- Cva:** Cva
- Dc:** Dc (m)
- Lh:** Lh (m)
- μ :** μ (Pa.s)
- Q:** Q (m³/s)
- Du:** Du (m)
- Lh1:** Lh1 (m)
- ρ :** ρ (Kg/m³)
- Qa:** Qa (m³/s)
- Do:** Do (m)
- RI:** RI
- ρ_s :** ρ_{Solido} (Kg/m³)
- Qr:** Qr (m³/s)
- Di:** Di (m)
- Nº de Hidrociclones:** Nº de Ciclonas
- n:** Et (<= 1)
- Cvr:** Cvr

There are buttons for "Reiniciar", "Rietema", "Bradley", "Descobrir", and "Dimensionar". A "Resultado" button is also present. Below the input fields is a 3D visualization of a hydrocyclone, with a "Visual" button and a "Rotação Y: 180" slider. The camera zoom is set to 0.8.

Fonte: Autoria própria

Ao iniciar a ferramenta, o usuário visualizará o painel de símbolos, como apresentado na figura 9. Este painel apresenta os símbolos utilizados, juntos de breves explicações, e pode ser fechado ao clicar no próprio painel ou no botão “?” (interrogação) à esquerda do título “Resultados”. Após fechado, o painel de resultados será completamente visível e, notadamente limpo. Os resultados só são visíveis após a conclusão de algum processo de solução.

Figura 11 – Painel de resultados e símbolos da ferramenta



Fonte: Autoria própria

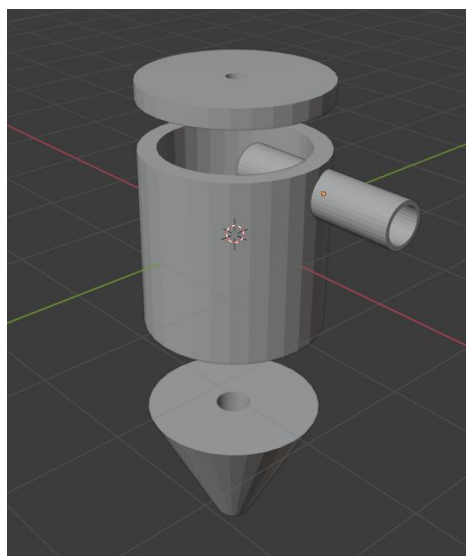
No canto esquerdo superior encontra-se um botão que permite que o usuário acesse o painel de distribuição granulométrica. Este painel está representado na figura 10 e nele deve-se inserir os pontos conhecidos da curva da distribuição granulométrica da questão, para então adicioná-los pelo botão Adicionar. Pode-se consultar os pontos já adicionados e também apagá-los. Esses pontos podem ser obtidos através da equação da curva granulométrica linearizada pelo modelo Rosin-Rammler-Bennet (RRB). Quando concluídas as adições, basta selecionar Solução e o cálculo da eficiência global de coleta reduzida e da eficiência global de coleta serão realizados. Nota-se que é necessário o conhecimento do diâmetro de corte reduzido e de um valor de razão líquida para que o resultado seja adequado. Estes valores podem ser obtidos através da solução no painel geral.

Figura 12 – Painel de distribuição granulométrica (DTA) da ferramenta

Fonte: Autoria própria

A representação tridimensional do hidrociclone foi dividida em partes, criadas usando o software Blender, sendo peças similares às encontradas na figura 11. Cada peça pode ser exportada separadamente para o Unity, que permite o agrupamento e a modificação dessas peças conforme necessário pela ferramenta. No canto esquerdo inferior se encontra o botão “Visual”, que permite que o usuário altere sua visão entre o hidrociclone e o painel geral.

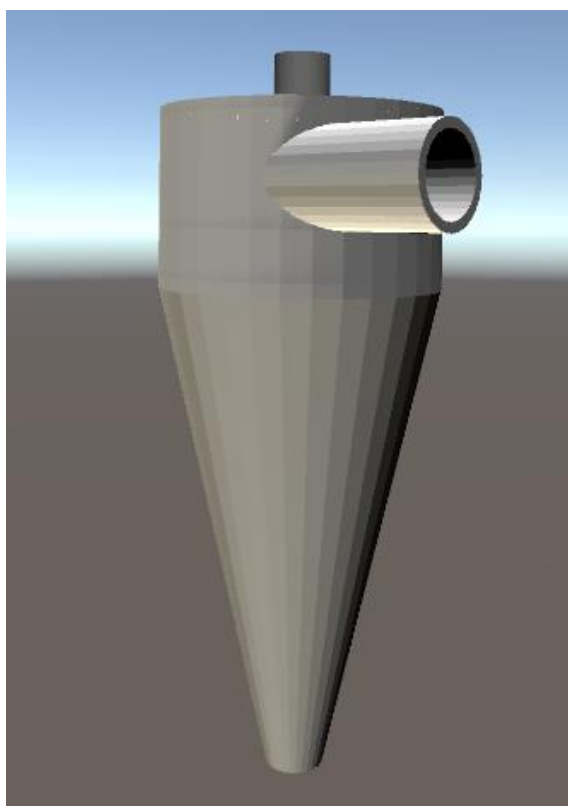
Figura 13 – Esboço das peças do hidrociclone separadas no Blender



Fonte: Autoria própria

O código limita a interação do usuário com a representação do hidrociclone para fins de simplificação do desenvolvimento da ferramenta. É possível, entretanto, rotacionar o hidrociclone no eixo Y e alterar o zoom da câmera, permitindo uma melhor visualização do hidrociclone. Após adquirir novos valores relativos às dimensões do hidrociclone através da sua metodologia de solução, a ferramenta deve automaticamente atualizar suas dimensões e apresentar a nova representação dessas dimensões. A figura 12 apresenta um esboço de como o hidrociclone deve se parecer quando as peças estão encaixadas e redimensionadas.

Figura 14 – Exemplo visual do hidrociclone após acoplamento das peças



Fonte: Autoria própria

3.3 A PROPOSTA DE APRENDIZAGEM

De forma a complementar a ferramenta e a proposta educacional do trabalho, é disponibilizado nos apêndices B e C propostas de aprendizagem que utilizem a ferramenta, com o passo a passo de como resolver as questões 1 e 4. As demais questões apresentadas são solucionadas de forma similar ao apresentado.

A metodologia de solução proposta é similar à solução manual das questões, sugerindo o uso da ferramenta em oposição à solução manual. O passo a passo deve auxiliar no entendimento de como pode ser o uso prático de uma ferramenta de simulação.

As principais questões comparadas têm suas autorias atribuídas à Ricardo Andrade Medronho e são facilmente encontradas online. A versão tratada por este trabalho foi encontrada na plataforma Poli OpenCourse, desenvolvida por Caio Costa Velasco e publicada em junho de 2012. Junto com a lista de hidrociclones, há mais outras quatro listas e suas soluções propostas, para a disciplina Operações Unitárias I lecionada por Medronho na UFRJ.

Uma das questões tratadas pelo trabalho, como exemplo de como podem ser representadas, é a sexta questão. Essa questão é uma versão invertida da terceira questão, e é escrita conforme a seguir: Uma fábrica opera uma bateria de hidrociclones Rietema a $0,0036 \text{ m}^3/\text{s}$ de suspensão aquosa a 9% v/v. Encontre o número mínimo de hidrociclones necessários e as dimensões que devem ter, sabendo que a eficiência total de operação é de 59%, a C_{vr} é 0,1154 e a vazão individual é de $4,86 \times 10^{-5}$.

CAPÍTULO 4 – RESULTADOS E DISCUSSÕES

Esse capítulo é dedicado às discussões relacionadas ao desempenho da ferramenta desenvolvida e aos resultados obtidos por essa, comparado às soluções propostas. Também serão discutidos pontos de melhoria potenciais e os desafios encontrados em algumas etapas do desenvolvimento do trabalho.

4.1 CAPACIDADE DE SOLUÇÃO DA FERRAMENTA E SUA DIDÁTICA COMO FERRAMENTA DE SIMULAÇÃO

Seis questões foram resolvidas e comparadas, sendo três conforme lista de exercícios propostos por Medronho e três questões suas respectivas soluções inversas, onde o objetivo é encontrar as dimensões do hidrociclone. Em suma, as etapas que o usuário tem de cumprir para obter a solução final das três questões não modificadas propostas por Medronho são:

1. Adicionar as variáveis conhecidas aos campos do painel geral e requisitar a solução;
2. Acessar o painel de distribuição granulométrica, adicionar todos os pontos conhecidos e requisitar a solução deste painel;
3. Requisitar, novamente, a solução no painel geral.

A primeira e a terceira etapa são caracterizadas por soluções algébricas. A primeira etapa deve calcular variáveis fundamentais como vazão, diâmetro do hidrociclone e os adimensionais. Essa etapa isolada já fornece respostas relevantes para a análise de hidrociclones, podendo revelar a necessidade de uma associação em paralelo ou informando o diâmetro de corte reduzido desse sistema. A segunda etapa é dependente da primeira, já que o diâmetro de corte reduzido (d_{50}') do sistema é fundamental na solução da eficiência global de coleta reduzida (η'), como observado na equação (10).

$$\eta' = \int_0^1 1 - \exp \left[-0,693 \left(\frac{d}{d_{50}'} \right)^3 \right] . dy \quad (10)$$

A segunda etapa da solução é, então, encarregada de ajustar os pontos da distribuição granulométrica da corrente de entrada à equação do modelo RRB linearizado (11), realizar a regressão linear e então solucionar a integral através da Regra do Trapézio. A regressão linear deve revelar tanto o coeficiente angular “m” quanto linear “b”. Há uma constante que é usada de forma a simplificar o procedimento, chamada de “k”, que é obtida em função dos

coeficientes angular e linear, conforme a equação (12) indica. A equação da reta é aqui representada pela equação (13), onde “m” é o coeficiente angular e “b” é o coeficiente linear.

$$\ln \left[\ln \left(\frac{1}{1-y} \right) \right] = m \cdot \ln(d) - m \cdot \ln(k) \quad (11)$$

$$k = \exp(b/-m) \quad (12)$$

$$y = m \cdot x + b \quad (13)$$

Após realizados os passos anteriores, basta usar a equação (11) para que se isole (d), obtendo-se a equação (14). Então, substitua “d” na equação (10) pela equação (14), e o resultado deve ser a equação (15). Esse passo é necessário para que a integral seja resolvida.

$$d = k \left[\ln \left(\frac{1}{1-y} \right) \right]^{\left(\frac{1}{m} \right)} \quad (14)$$

$$\eta' = \int_0^1 1 - \exp \left\{ - \frac{0,693}{(d_{50'})^3} \left[k \left(\ln \left(\frac{1}{1-y} \right) \right)^{\left(\frac{1}{m} \right)} \ln \right]^3 \right\} \cdot dy \quad (15)$$

Conhecendo η' , é possível calcular a eficiência global de coleta (η), usando a Razão Líquida (RL), que normalmente já foi obtida na primeira etapa da solução. A terceira etapa consiste em calcular as variáveis que antes não eram possíveis de calcular por conta da ausência de η , normalmente a concentração de volume sólido no retido (Cvr).

Já no procedimento inverso, a primeira etapa é garantir que se possui a vazão de entrada do hidrociclone. Embora os exercícios já forneçam a vazão total, de entrada e/ou a quantidade de hidrociclones associados em paralelo, essa etapa é primordial ao processo de tentativa e erro e de análise de dados envolvida numa simulação. Assumindo-se uma vazão muito alta e não estipulando um número adequado de ciclones, o usuário pode se deparar com um diâmetro de centro (Dc) muito grande para um hidrociclone real. Com isso em mente, o usuário pode buscar uma quantidade de hidrociclones mais coerente com base no processo produtivo de interesse e nos fundos disponíveis. Dito isso, nenhuma das questões tratadas nesse trabalho deixa ambíguo o número de hidrociclones do sistema, o que permite uma melhor comparação dos resultados obtidos pela ferramenta.

Diferentemente do primeiro processo de solução, o processo de dimensionamento não usa distribuição granulométrica para solução. Isso se dá porque o modelo de distribuição está

intimamente associado ao equipamento. O d_{50} varia em função da configuração do hidrociclone, viscosidade (μ) e densidade (ρ) do fluido, vazão de entrada (Q_a), da queda de pressão de operação (Δp) e do diâmetro do hidrociclone (D_c). Alguns desses valores podem ainda não ter sido decididos e, com isso em mente, é melhor estimar uma eficiência e uma concentração na corrente do retido que sejam de interesse.

Por fim, possuindo dimensões, vazões e outras variáveis que possam ser de interesse para o caso específico, é possível realizar anotações externas à ferramenta e então realizar simulações para confirmar se o comportamento esperado desses ciclones é de fato interessante. O usuário só deve se atentar ao fato de que o foco deste trabalho são concentrações de alimentação menores que 10% v/v sólido/líquido, mesmo que a ferramenta possua equações que se apliquem à outras condições também.

4.2 COMPARAÇÃO DOS RESULTADOS OBTIDOS E O COMPORTAMENTO DO PROTÓTIPO TRIDIMENSIONAL

Os enunciados das questões usadas neste trabalho podem ser consultados no anexo A e no apêndice A, no final do documento. As questões foram enumeradas de um a seis, onde as três primeiras são originais encontradas na lista de Medronho, enquanto as três últimas são versões modificadas de forma a serem solucionadas de forma inversa. Dito isso, os resultados obtidos para as três primeiras questões podem ser observados na tabela 5.

Tabela 4 – Comparativo de resultados obtidos entre as soluções propostas para as questões e a ferramenta.

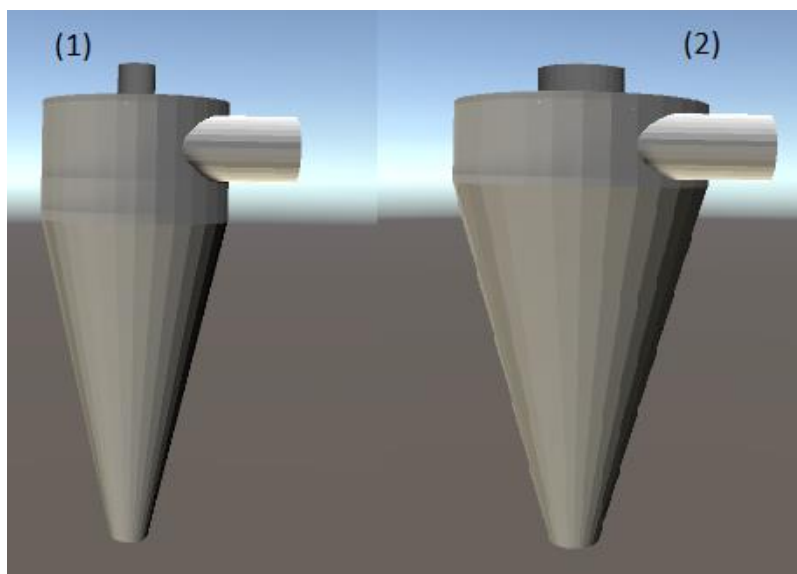
		Questão 1 (Bradley)		Questão 2 (Rietema)		Questão 3 (Rietema)	
	Nº de Cicl.	η	Cvr	Qa	η	Nº de Cicl.	η
Gabarito	18	0,92	0,043	0,0036	0,36	74	0,59
Ferramenta	17,7	0,91225	0,0412	0,00361	0,37	74,13	0,606

Fonte: Autoria própria

Além dos valores dos resultados, pode-se verificar que o protótipo de hidrociclone toma formas diferentes dependendo da configuração escolhida e nos valores dos resultados, mesmo

que não seja um formato totalmente fiel ao original. A figura 13 apresenta um comparativo entre o hidrociclone gerado pela questão 1 (Bradley) e pela questão 2 (Rietema).

Figura 15 – Comparativo visual entre o protótipo de hidrociclone Bradley (1) e Rietema (2)



Fonte: Autoria própria

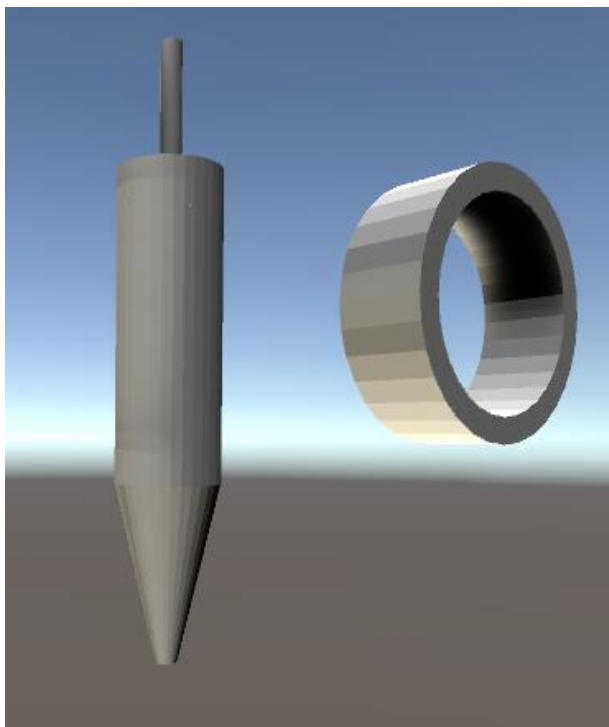
Para as questões 4, 5 e 6, as dimensões são consideravelmente próximas. Parte da diferença pode ser atribuída à aproximações observadas nas soluções propostas, inclusive, aplicadas aos parâmetros de configuração – por exemplo, (n_3) para Rietema vale 0,116, mas na solução proposta esse valor é às vezes aproximado para 0,12. A figura 14 demonstra a representação gerada pela ferramenta para os hidrociclones formados na questão 3 e 6, que possuem dimensões próximas e diminutas. Esse resultado indica a ferramenta não é capaz de gerar uma representação gráfica adequada para hidrociclones com dimensões muito pequenas. Adicionalmente, não foram realizados testes com grandes dimensões.

Tabela 5 – Comparativo de resultados obtidos entre as soluções propostas para as questões e a ferramenta.

	Questão 4 (Bradley)			Questão 5 (Rietema)		Questão 6 (Rietema)		
	Dc	Du	RL	Dc	Du	Dc	Du	N° de Cicl.
Gabarito	0,06	0,008	0,206	0,09	0,024	0,01	0,0027	74
Ferramenta	0,0595	0,00782	0,2068	0,0942	0,0255	0,0105	0,0031	74,07

Fonte: Autoria própria

Figura 16 – Representação formada pela ferramenta para as questões 3 e 6.



Fonte: Autoria própria

CAPÍTULO 5 – CONCLUSÃO

5.1 CONCLUSÃO

Foi desenvolvida uma ferramenta capaz de solucionar problemas relacionados ao projeto e dimensionamento de hidrociclones, com um desvio aceitável, utilizando tecnologias digitais comumente destinadas também a outros fins que não o de simulação. Essa ferramenta foi chamada de HidroCSim. As questões tratadas são bastante pertinentes ao meio acadêmico e, junto da ferramenta, abrem margem para o desenvolvimento de novas alternativas de ensino de conteúdos disciplinares.

Este trabalho apresenta uma proposta educacional simples para a introdução de estudantes à simulação, enquanto também apresenta um método de desenvolvimento para um meio de simulação. Nos apêndices pode-se consultar o caminho para solucionar as questões com uso da ferramenta, assim como seu código. Tanto a ferramenta (utilizável independentemente), quanto o projeto da ferramenta (utilizável no Unity), estão disponíveis online. De posse dessas informações, é possível a adaptação das questões ou da ferramenta para formulação de exercícios de simulação. Vale notar que, embora o C# seja uma linguagem independente, o código desta ferramenta é altamente dependente do Unity, por usar sistemas específicos dessa plataforma. Por isso, o código não funciona de forma independente e precisaria ser adaptado para outras plataformas.

Os softwares utilizados neste trabalho (Unity e Blender) são gratuitos para uso individual e não comercial, sendo o uso do Unity sob licença e acordo de termos que limitam seu uso por entidades comerciais ou para fins comerciais. Os resultados encontrados pela ferramenta para qualquer exercício podem ser replicados em outras plataformas e linguagens de programação, como Microsoft Excel, MatLab e Python.

Esse trabalho busca demonstrar que o ensino de engenharia química pode estar intimamente associado ao uso e desenvolvimento de ferramentas digitais que não necessariamente devem substituir a prática por meio da escrita física, mas que podem contribuir para a familiarização do estudante a práticas amplamente adotadas pela indústria.

A multidisciplinaridade envolvida no uso e desenvolvimento dessas ferramentas pode ser observada, como no uso de conhecimentos e tópicos normalmente trabalhados em diversas disciplinas, como cálculos, métodos numéricos e mecânica dos fluidos. Há várias alternativas que podem ser exploradas em sala de aula para contextos simples que podem cumprir um papel similar ao da ferramenta desenvolvida.

Em suma, este trabalho propõe meios de explorar o ensino acadêmico de disciplinas associadas à engenharia química e tem potencial para servir de auxílio para outras propostas de ensino, seja por meio da ferramenta em si, do código ou pela sua parte textual.

5.2 SUGESTÕES PARA TRABALHOS FUTUROS

- AMPLIAÇÃO DA DELIMITAÇÃO DO TRABALHO

Essa ferramenta é incapaz de solucionar adequadamente questões que tratam de configurações que não sejam Rietema ou Bradley. É possível adicionar novas configurações, como CBV-DEMCO, equações pertinentes a hidrociclones que não pertencem a uma família e até outros equipamentos que compartilhem similaridades com hidrociclones, como ciclones gasosos.

Embora seja possível calcular na ferramenta sem respeitar as dimensões da configuração escolhida, os resultados não seriam fiéis a nenhum hidrociclone real. Por exemplo, se inseridas dimensões para o hidrociclone Bradley e a solução escolhida for a de Rietema, os cálculos substituirão as dimensões de Bradley por dimensões apropriadas para Rietema e utilizar os parâmetros de Rietema. No caso da escolha certa, a substituição também será feita, mas a discrepância entre as dimensões inseridas e calculadas seriam pequenas o bastante para não interferir de forma significativa no resultado.

- NOVAS DE PROPOSTAS DE APRENDIZAGEM

Com a ferramenta em mãos, é possível utilizá-la na elaboração de situações-problema e cenários que estimulem o pensamento crítico, a tomada de decisão e o estudo e prática de técnicas associadas à simulação e ao projeto e dimensionamento de hidrociclones. Alternativamente, pode-se usar esse trabalho como base para o desenvolvimento de ferramentas ou programas similares em outros meios, como em uma planilha do Excel. As questões aqui tratadas podem servir de base e/ou como comparativos.

- APRIMORAMENTO DA FERRAMENTA

Embora a ferramenta atenda à proposta, ela peca em diversos aspectos que poderiam ser melhor tratados, seja pela experiência de usuário, pelos *bugs* existentes, escopo limitado, pela

interface não-prática ou até pelo código que pode ser otimizado para prevenção de erros e um funcionamento mais leve. Por exemplo, ela não é capaz de trabalhar adequadamente com hidrociclones de dimensões muito pequenas, ao mesmo tempo que não foi testada para maiores dimensões também. Além disso, a ferramenta não é capaz de detectar as geometrias de forma precisa, e seu método “descubra” pode falhar em situações que não deveria.

A metodologia de solução e o código de solução possuem margem para melhorias significativas. É possível que se incluam mais equações e melhores condições para a solução dos problemas, que permitam solução cenários que a ferramenta não é capaz de resolver. A ordem das instruções no código do método de solução também possui margem para melhora.

REFERÊNCIAS BIBLIOGRÁFICAS

ADESANYA, M. **A Gentler Introduction to Programming**. FreeCodeCamp, out. de 2017. Disponível em: <<https://www.freecodecamp.org/news/a-gentler-introduction-to-programming-1f57383a1b2c/>>. Último acesso: 20 de novembro de 2020.

ALCÁCER V.; CRUZ-MACHADO V. **Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems**. Elsevier, 2019.

ALDRICH, C. **Progress in Filtration and Separation**. Elsevier, 2015. Cap 1.

ANTUNES, M.; MEDRONHO, R., A. **Bradley Hydrocyclones: Design and Performance Analysis**. Springer Science+Business Media, Hydrocyclones, 1992.

VELASCO, C. **Operações Unitárias I**. Poli OpenCourse, 2012. Disponível em: <<https://sites.google.com/a/poli.ufrj.br/quimica/operacoesunitarias1>>. Último acesso: 14 de dezembro de 2020.

COELHO, P. M. N. **Rumo à Indústria 4.0**. Universidade de Coimbra, jul. de 2016.

COUPER, J. R.; PENNEY, W. R.; FAIR, J. R. **Chemical Process Equipment – Selection and Design**. Elsevier, 2ª Edição, 2009. Cap 11.

DAHM, K. D.; HESKETH, R. P.; SAVELSKI, M. J. **Is Process Simulation used Effectively in ChE Courses?** Chemical Engineering Education, v36, 2002.

FERREIRO, S.; KONDE, E.; FERNÁNDEZ, S.; PRADO, A. **Industry 4.0: Predictive Intelligent Maintenance for Production Equipment**. European Conference of the Prognostics and Health Management Society, 2016.

FORD, S.; MINSHALL, T. **Where and How 3D Printing is Used in Teaching and Education**. Additive Manufacturing, Elsevier, 2018.

MAITLAND, G. **Ten skills chemical engineers should be talking about (Day 114)**. IChemE Blog, set. de 2014. Disponível em: <<https://ichemeblog.org/2014/09/18/ten-skills-chemical-engineers-should-be-talking-about-day-114/>>. Último acesso em: 20 de novembro de 2020.

GEORGE BROWN COLLEGE. **How Simulation Tools are Transforming Education and Training**. Electronics Technician Training Online Education Program. Disponível em: <<https://www.etcourse.com/simulation-tools-transform-education-and-training.html>>. Último acesso em: 20 de novembro de 2020.

HOFMANN, E.; RÜSCH, N. **Industry 4.0 and the Current Status as well as Future Prospects on Logistics**. Elsevier, 2017.

JILLAVENKATESA, A.; DAPKUNAS, S. J.; LUM, L. S. H. **Particle Size Characterization**. National Institute of Standards and Technology, 2001.

KIRILLOV, N. P.; LEONTYEVA, E. G.; MOISEENKO, Y. A. **Creativity in engineering education**. Procedia - Social and Behavioral Sciences, Elsevier, 2015.

LUKEC, I. **What is the Most Useful Software in Chemical Engineering?** Simulate Live, 2019. Disponível em: <<http://www.simulatelive.com/product-reviews/simulation/what-is-the-most-useful-software-in-chemical-engineering>>. Último acesso: 20 de novembro de 2020.

MASSARANI, G. **Fluidodinâmica em Sistemas Particulados**. E-Papers, 2002. 2ª Edição. Cap 2.4.

MEDRONHO, R. A. **Scale-Up of Hydrocyclones at Low Feed Concentrations**. Universidade de Bradford, 1984.

MICROSOFT. **Introdução à linguagem C# e ao .NET**. Documentação do C#. Disponível em: <<https://docs.microsoft.com/pt-br/dotnet/csharp/getting-started/>>. Último acesso em: 20 de novembro de 2020.

MINISTÉRIO DA INDÚSTRIA, COMÉRCIO E SERVIÇOS. **Agenda Brasileira Para a Indústria 4.0**. ABDI. Disponível em: <<http://www.industria40.gov.br/>>. Último acesso em: 20 de novembro de 2020.

PARENTE, A. P. **Ferramentas de Automação e Machine Learning para Operação Segura e Ecoeficiente de Processos de Combustão**. Rio de Janeiro, Brasil, jul. de 2019. UFRJ.

PEÇANHA, R. P. **Sistemas Particulados**. Elsevier, 1º Edição, 2014. Cap 3.6.

PEJIC-BACH, M. *et al.* **Text mining of industry 4.0 job advertisements**. International Journal of Information Management, Elsevier, 2018.

PEKER, S.; HELVACI, S. **Solid-Liquid Two-Phase Flow**. Elsevier, 1ª Edição, 2007. Cap 8.

RADIX. **Radix expõe projetos e tecnologias de ponta em congresso na UFRJ**. Radix Engenharia, jun. de 2019. Disponível em: <<http://www.radixeng.com.br/noticias/650/radixexpoe-projetos-e-tecnologias-de-ponta-em-congresso-na-ufrj>>. Último acesso em: 20 de novembro de 2020.

RIOJAS, M. *et al.* **Educational Technologies for Precollege Engineering Education**. IEEE Transactions on Learning Technologies, IEEE Xplore, 2012.

SANTOS, M. T. DOS; JUNIOR, A. S. V.; LE ROUX, G. A. C. **Programming skills in the industry 4.0: are chemical engineering students able to face new problems?** Education for Chemical Engineers, Elsevier, 2017.

SCALFANI, V. F.; VAID, T. P. **3D Printed Molecules and Extended Solid Models for Teaching Symmetry and Point Groups**. Journal of chemical education, abr. de 2014. Disponível em: <<https://pubs.acs.org/doi/abs/10.1021/ed400887t>>. Último acesso em: 20 de novembro de 2020.

SINNOT, R.; TOWLER, G. **Chemical Engineering Design**. Elsevier, 6ª Edição, 2020. Cap 10.

THE BLENDER FOUNDATION. **About Blender.** Disponível em: <<https://www.blender.org/about/>>. Último acesso em: 20 de novembro de 2020.

UNITY TECHNOLOGIES. **Plataforma do Unity. Produtos.** Disponível em: <<https://unity.com/pt/products/unity-platform>>. Último acesso em: 20 de novembro de 2020.

ZHANG, L.; ZHOU, L.; REN, L.; LAILI, Y. **Modeling and Simulation in Intelligent Manufacturing.** Computers in Industry, Elsevier, 2019.

ZHANG, P. **Advanced Industrial Control Technology.** Elsevier, 1ª Edição, 2010. Cap. 1.

APÊNDICE A – QUESTÕES PROPOSTAS MODIFICADAS

Neste apêndice podem ser consultadas as questões 4 a 6, que são as questões 1 a 3 encontradas na lista modificadas para que fossem solucionáveis de forma inversa, após o link abaixo. O link permite o download do projeto da ferramenta para o Unity e o programa da ferramenta, em uma pasta do Dropbox:

<https://www.dropbox.com/sh/997puidakx4iycy/AACtaDeqd7JUXznGwAzw_4Ula?dl=0>

4. Uma bateria de 18 hidrociclones Bradley trata $0,013889 \text{ m}^3/\text{s}$ de suspensão aquosa de CaCO_3 1% v/v. Calcule o D_c e o D_u mais apropriados para o hidrociclone, que deve trabalhar a uma eficiência de 92% e a concentração volumétrica do underflow deve ser próxima de 0,043. A qual razão líquida cada hidrociclone deve operar?

DADOS: $\Delta P = 50 \text{ psi}$; $\rho_s = 2,8 \text{ g/cm}^3$; $\mu = 1,0 \text{ cp}$; $T = 20^\circ\text{C}$

Resp.: (a) $D_c = 0,06 \text{ m}$; (b) $D_u = 0,008 \text{ m}$; (c) 0,206

5. Um hidrociclone Rietema trata $0,0036 \text{ m}^3/\text{s}$ de suspensão aquosa a 9% v/v. Encontre as dimensões mais apropriadas para o hidrociclone, sabendo que a razão líquida usada é de 0,2776, a fração de volume sólido no retido que se quer obter é de 11,54% e a eficiência global de coleta reduzida mínima de interesse é de 0,124.

DADOS: $\rho_s = 2,5 \text{ g/cm}^3$; o fluido de operação é a água com as seguintes propriedades: $\rho = 1,0 \text{ g/cm}^3$ e $\mu = 1,0 \text{ cp}$. $\Delta P = 206843 \text{ Pa}$

Resp.: $D_c = 0,09 \text{ m}$ e $D_u = 0,024$

6. Uma fábrica opera uma bateria de hidrociclones Rietema a $0,0036 \text{ m}^3/\text{s}$ de suspensão aquosa a 9% v/v. Encontre o número mínimo de hidrociclones necessários e as dimensões que devem ter, sabendo que a eficiência total de operação é de 59%, a C_{vr} é 0,1154 e a vazão individual é de $4,86 \times 10^{-5}$.

Resp.: 75 hidrociclones em paralelo; $D_c = 0,01 \text{ m}$ e $D_u = 0,0027 \text{ m}$

APÊNDICE B – SOLUÇÃO DA PRIMEIRA QUESTÃO

É possível resolver o exercício a seguir utilizando a ferramenta de forma adequada. Esse anexo irá tratar do passo-a-passo necessário para a sua solução, com imagens da ferramenta e orientações de como deve ser feito o exercício.

1. (MEDRONHO, R.) Uma bateria de hidrociclones de Bradley, de 6 cm de diâmetro, é utilizado para tratar 50 m³/h de uma suspensão aquosa, a 1% em volume de CaCO₃. Calcular:

- a) o número de hidrociclones necessários.
- b) a eficiência total de separação;
- c) a concentração volumétrica do underflow;

DADOS: $\Delta P = 50$ psi ; $\rho_s = 2,8$ g/cm³ ; $D_u = 0,8$ cm ; $T = 20^\circ\text{C}$

Distribuição granulométrica do material particulado na alimentação:

d (μm)	5	10	15	20	30	40
y (%)	12	29	46	60	80	92

Resp.: (a) 18 hidrociclones; (b) 0,92; (c) 0,043.

Inicialmente, deve-se considerar as unidades das variáveis da questão. A ferramenta realiza seus cálculos com base no Sistema Internacional, salvo situações em que ela explicita o uso de alguma medida diferente. Desta forma, deve-se transformar as unidades para o SI:

Símbolo	Dc	Qa	Cva	ΔP	ρ_s	Du
Original	6 cm	50 m ³ /h	0,01	50 Psi	2,8 g/cm ³	0,8 cm
SI	0,06 m	0,01389 m ³ /s	0,01	344 738 Pa	2800 kg/m ³	0,008 m

Obs: A temperatura serve para indicar as condições do líquido, comumente água. Assume-se, então, a viscosidade dinâmica (μ) como 1cp, que corresponde à 0,001 Pa.s no Sistema Internacional.

Com os valores adequadamente transformados, pode-se ir à ferramenta. Quando inicializada, ela terá a aparência da figura abaixo. Deve-se fechar as janelas clicando no canto direito superior e esquerdo superior, nos botões “Resultado” e “DTA”.

Painel DTA **Insira os valores conhecidos:** **Resultado**

? RESULTADOS

Símbolos

η : Eficiência Global de Coleta Reduzida	$d50$: é o diâmetro de partícula coletado pelo hidrociclone com 50% de Eficiência Individual de Coleta Reduzida. (m)
η : Eficiência Global de Coleta	Dc: Diâmetro da parte cilíndrica do hidrociclone (m)
ρ : Massa específica do fluido utilizado (Kg/m ³). Assume-se água.	Du: Diâmetro do underflow (m)
μ : Viscosidade dinâmica do líquido usado (Pa.s)	Do: Diâmetro de saída do overflow (m)
ρ_s : Massa específica dos sólidos particulados em (Kg/m ³)	Di: Diâmetro da alimentação do hidrociclone (m)
Cva: Fração de volume sólido na corrente de alimentação.	Lh: Altura total do hidrociclone (m)
Cvr: Fração de volume sólido na corrente de retido (underflow).	Lh1: Altura da parte cilíndrica do hidrociclone (m)
Q: Vazão total (m ³ /s)	y: eficiência para o diâmetro de partícula de interesse, a partir da função [y(d)]
Qa: Vazão de alimentação do hidrociclone individual (m ³ /s)	numHidrociclonesPar: Número de hidrociclones em paralelo
Qr: Vazão de saída na corrente de retido (m ³ /s)	stk50Eu: Representa o adimensional "Stokes de corte Euler"
dP: Diferença de pressão de operação do hidrociclone (Pa)	
RI: Razão Líquida, também chamada de Rf.	
d50: O diâmetro de partícula que é separado com 50% de Eficiência Individual de Coleta (μ m)	

Visual **Descobrir** **Dimensionar**

Painel que introduz os símbolos usados e os resultados obtidos.

Os campos que aceitam os valores serão visíveis logo após o fechamento das janelas, conforme a figura a seguir. Pode-se, então, preencher os campos com seus devidos valores. Não há a necessidade de preencher todos os campos, mas a ferramenta não é capaz de resolver exercícios com informações insuficientes.

Painel DTA **Insira os valores conhecidos:** **Resultado**

dP: dP	Cva: Cva	Dc: Dc	Lh Lh
μ : mi	Q: Q	Du Du	Lh1 Lh1
ρ : Rho	Qa: Qa	Do: Do	RI: RI
ρ_s : rhoSolido	Qr: Qr	Di Di	Nº de Hidrociclones: Nº de Ciclones
η : Et	Cvr: Cvr		

Reiniciar

Soluções:

☐ Dimensionar

Visual

Painel geral, onde o usuário insere os valores de entrada.

Após preenchida, a ferramenta deve se assemelhar à próxima figura. Nesse caso, deve-se escolher a geometria do hidrociclone, que é Bradley. Embora a ferramenta seja capaz de detectar geometrias, ela não será capaz nesse exemplo, por faltar informações. Após clicar no botão Bradley, pode-se consultar os resultados obtidos na aba superior direita, chamada Resultados. Pode ser necessário fechar a aba “Símbolos”, clicando no ponto de interrogação. Quando o fizer, perceberá que já possui algumas respostas para a questão, mas ainda não possui todas. Perceba também que a representação ao fundo sofreu modificações.

Painel DTA **Insira os valores conhecidos:** **Resultado**

dP: 344738	Cva: 0,01	Dc: 0,06	Lh Lh
μ : 0,001	Q: 0,01389	Du 0,008	Lh1 Lh1
ρ : 1000	Qa: Qa	Do: Do	RI: RI
ρ_s : 2800	Qr: Qr	Di Di	Nº de Hidrociclones: Nº de Ciclones
η : Et	Cvr: Cvr		

Soluções:

Rietema Bradley

Descobrir Dimensionar

Reiniciar

Visual

Painel geral após a introdução dos valores pelo usuário.

Painel DTA

Insira os valores conhecidos:

Resultado

?

RESULTADOS

ρ : 1000Kg/m ³	Q: 0,01389m ³ /s	Dc: 0,06m
ρS : 2800Kg/m ³	Qa: 0,0007832485m ³ /s	Du: 0,008m
μ : 0,001Pa.s	RI: 0,2144466	Do: 0,012m
dP: 344738	η : 0	Di: 0,008571429m
Nº de Hiclones em Paralelo: 17,73384	Cva: 0,01	Lh: 0,4111862m
d50': 4,458226E-06	Cvr: 0	Lh1: 0,03m
Stk50Eu: 0,08244889	Geometria: Bradley	η' : 0
Eu: 9402,933		
Re: 16621,05		

Visual

Descobrir

Dimensionar

Painel de resultados após requisitada a primeira parte da solução do problema

Caso queira conferir qualquer engano, esse é um bom momento. Você pode reiniciar a ferramenta para inserir novos valores, com a garantia de que nenhum valor antigo irá interferir com os novos. Quando concluir essa etapa, feche a aba de resultados e abra a aba chamada Painel DTA. Essa aba oferece soluções com base em análises granulométricas, linearizadas pelo modelo RRB. Ela usa pontos (d,y) de um modelo de análise granulométrica. A forma adequada de solução dessa questão é através da adição dos pontos que nos foi informado no enunciado: Inclui-se o diâmetro d, em μm , e a eficiência y, adimensional, em valor decimal. Quando ambos os campos possuírem os valores corretos, deve-se clicar em Adicionar, para que os pontos sejam salvos e então, deve-se repetir o processo para o próximo ponto. Você pode consultar os pontos adicionados em “Consultar”.

APÊNDICE C – SOLUÇÃO DA QUARTA QUESTÃO

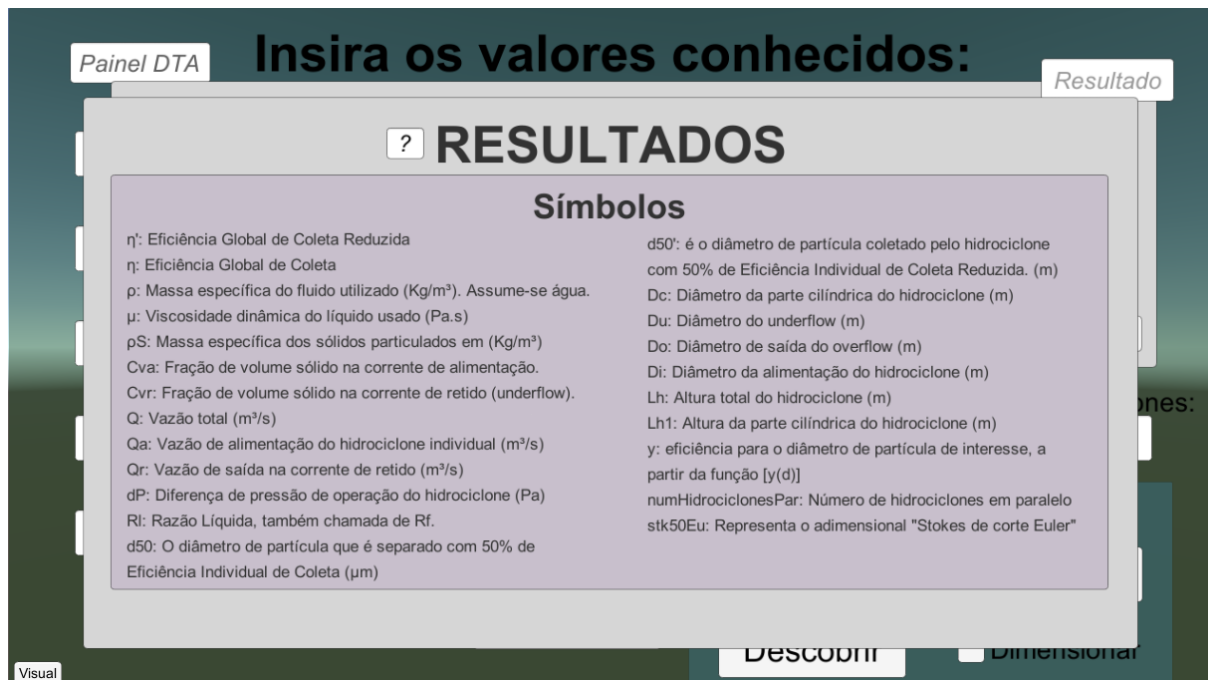
É possível resolver o exercício a seguir utilizando a ferramenta de forma adequada. Esse anexo irá tratar do passo-a-passo necessário para a sua solução, com imagens da ferramenta e orientações de como deve ser feito o exercício.

4. Uma bateria de 18 hidrociclones Bradley trata 0,013889 m³/s de suspensão aquosa de CaCO₃ 1% v/v. Calcule o Dc e o Du mais apropriados para o hidrociclone, que deve trabalhar a uma eficiência de 92% e a concentração volumétrica do underflow deve ser próxima de 0,043. A qual razão líquida cada hidrociclone deve operar?

DADOS: $\Delta P = 344\,738\text{ Pa}$; $\rho_s = 2\,800\text{ Kg/m}^3$; $\mu = 0,001\text{ Pa.s}$; $\rho_s = 1\,000\text{ Kg/m}^3$

Resp.: (a) Dc = 0,06 m; (b) Du = 0,008 m; (c) 0,206

Como os valores já se encontram em SI, não há necessidade de conversão. Quando inicializada, ela terá a aparência da figura abaixo. Deve-se fechar as janelas clicando no canto direito superior e esquerdo superior, nos botões “Resultado” e “DTA”.



Painel que introduz os símbolos usados e os resultados obtidos.

Os campos que aceitam os valores serão visíveis logo após o fechamento das janelas. Pode-se, então, preencher os campos com seus devidos valores sem a necessidade de preencher

todos os campos, mas deve-se atentar ao fato de que a ferramenta não é capaz de resolver exercícios com informações insuficientes.

Nesse caso, não há um equipamento e nem dimensões para que a ferramenta sugira uma configuração. Embora a questão tenha informado que a bateria deve ser composta por hidrociclones Bradley, pode-se usar as informações da questão para comparar as duas famílias de hidrociclone, nas mesmas condições de operação. A figura abaixo apresenta como deve ficar a ferramenta preenchida. Note que, para este caso, a opção “Dimensionar” deve estar marcada, para que a ferramenta saiba distinguir o objetivo da função. Ela irá substituir o valor de D_c pelo valor encontrado. Alternativamente, não marcar essa opção não resultará em nada, pois caso o objetivo não seja dimensionar e D_c seja zero, a ferramenta não irá iniciar a solução. Após a escolha da geometria, pode-se consultar os resultados obtidos na aba Resultado. Pode ser necessário fechar a aba “Símbolos”, clicando no ponto de interrogação.

Painel geral após a introdução dos valores pelo usuário.

Os resultados para a geometria Bradley devem ser semelhantes aos da figura abaixo. Alternativamente, é possível comparar quais são as diferenças entre as duas geometrias para este caso, observando as figuras seguintes. Nesse caso, a principal variável fixada é a vazão de entrada (Q_a), já que se possui a vazão total (Q) e o número de hidrociclones fixos. É recomendado que se reinicie as variáveis, pois algumas variáveis podem ser carregadas da simulação anterior para a nova acidentalmente. Caso o usuário queira ser ainda mais cauteloso, o programa pode ser fechado e reaberto.

Painel DTA **Insira os valores conhecidos:** **Resultado**

? RESULTADOS

p: 1000Kg/m ³	Q: 0,013889m ³ /s	Dc: 0,05951694m
pS: 2800Kg/m ³	Qa: 0,0007716111m ³ /s	Du: 0,007818613m
μ: 0,001Pa.s	RI: 0,2068217	Do: 0,01190339m
dP: 344738	η: 0,92	Di: 0,008502421m
Nº de Hiclones em Paralelo: 18	Cva: 0,01	Lh: 0,4078757m
d50': 0	Cvr: 0,043	Lh1: 0,02975847m
Stk50Eu: 0	Geometria: Bradley	η': 0,89914
Eu: 9379,007		
Re: 16506,99		

Visual Descobrir Dimensionar

Painel de resultados após requisitada a solução do problema para a configuração Bradley

Painel DTA **Insira os valores conhecidos:** **Resultado**

? RESULTADOS

p: 1000Kg/m ³	Q: 0,013889m ³ /s	Dc: 0,03798708m
pS: 2800Kg/m ³	Qa: 0,0007716111m ³ /s	Du: 0,009618316m
μ: 0,001Pa.s	RI: 0,2068217	Do: 0,01291561m
dP: 344738	η: 0,92	Di: 0,01063638m
Nº de Hiclones em Paralelo: 18	Cva: 0,01	Lh: 0,1899354m
d50': 0	Cvr: 0,043	Lh1: 0m
Stk50Eu: 0	Geometria: Rietema	η': 0,89914
Eu: 1182,012		
Re: 25862,63		

Visual Descobrir Dimensionar

Painel de resultados após requisitada a solução do problema para a configuração Rietema

APÊNDICE D – CÓDIGO DA FERRAMENTA

Nessa seção do texto ficará apresentado o código da ferramenta. Vale lembrar que o código sem modificação só tem uso dentro do Unity e precisa ser adequadamente preparado utilizando a plataforma, considerando suas ferramentas. Muito do código é voltado para a interface e o funcionamento entre objetos. Dito isso, o projeto pode ser encontrado online para consulta.

01 – EquationsScript

Esse trecho é a parte que contém o principal processo de solução e onde se encontram a maioria das equações utilizadas no trabalho.

```
using Packages.Rider.Editor.UnitTesting;
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEditorInternal;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.UIElements;

public class EquationsScript : MonoBehaviour
{
    // Objetos na hierarquia do Unity
    public GameObject Rietema;
    public GameObject toolResults;
    public GameObject DTASolucao;
    public GameObject ModeloSolucao;
    public GameObject cabecaHidrociclone;
    public GameObject cabecaCilindro;
    public GameObject corpoCilindrico;
    public GameObject entradaHidrociclone;
    public GameObject saidaOverflow;
    public GameObject saidaUnderflow;
    public GameObject mainCamera;

    public bool isRietema = false;
    public bool isBradley = false;
    public string geometriaUsada;

    public UnityEngine.UI.Toggle projToggle;
    public bool dimensionar;

    // Variáveis utilizadas pela ferramenta para solução dos problemas.

    public float nSeparacaoLinha; // Eficiência de separação ( $\eta'$ ) ou Eficiência Global de Coleta Reduzida, necessária na solução por distribuição granulométrica
    public float nSeparacaoTotal; // Eficiência de separação ( $\eta$ ) ou Eficiência Global de Coleta, após considerar a razão líquida - na solução pelo método de Medronho

    public float rho = 1000f; // Massa específica do líquido onde se encontra a suspensão de partículas em ( $\text{Kg}/\text{m}^3$ ). Assume-se água.
    public float mi = 0.01f; // Viscosidade dinâmica do líquido onde se encontra a suspensão em ( $\text{Pa}\cdot\text{s}$ )
    public float rhoSólido; // Massa específica dos sólidos particulados em ( $\text{Kg}/\text{m}^3$ )
    public float Cva; // Fração de volume sólido na corrente de alimentação do hidrociclone
    public float Cv; // Fração de volume sólido na corrente de retido do hidrociclone (underflow)
    public float Cv; // Fração de volume sólido na corrente de operação (Igual à Cva quando associados em paralelo)
    public float Q; // Vazão de operação (Vazão total de entrada em todos os hidrociclones, se em paralelo, ou a vazão de alimentação, se em série (em  $\text{m}^3/\text{s}$ )

    public float Qa; // Vazão de alimentação do hidrociclone individual. É igual à vazão de operação se a configuração dos hidrociclones for em série (em  $\text{m}^3/\text{s}$ )

    public float Qr; // Vazão de saída na corrente de retido (Vazão de saída pelo underflow) do hidrociclone individual (em  $\text{m}^3/\text{s}$ )
    public float dP; // Diferença de pressão de operação do hidrociclone individual (em Pa ou  $\text{N}/\text{m}^2$ )
    public float RI; // Razão Líquida, também chamada de Rf por Medronho. O valor, se não for estipulado ou possível de ser descoberto, é comumente atribuído como 0,1 (recomendado por Rietema)
    public float dCorte; // d50 ou  $d^*$  => O diâmetro de partícula caracterizado por ser separada com 50% de Eficiência Individual de Coleta (em micrômetros -  $\mu\text{m}$ )
    public float dCorteR; // é o diâmetro das partículas que são coletadas pelo hidrociclone, por efeito centrífugo, com Eficiência Individual de Coleta Reduzida = 50%
    public float Dc; // Diâmetro da parte cilíndrica do hidrociclone (em metros)
```

```

public float Du; // Diâmetro do underflow do hidrociclone (em metros)
public float Lh; // Altura total do hidrociclone (em metros)
public float Do; // Diâmetro da saída superior (overflow) do hidrociclone (em metros)
public float Di; // Diâmetro da alimentação do hidrociclone (em metros)
public float Da; // Diâmetro da alimentação do hidrociclone (em metros)
public float e; // Altura do vortex catcher interna à parte cilíndrica do hidrociclone (em metros)
public float Lh1; // Altura da parte cilíndrica do hidrociclone (em metros)
public float d; // Diâmetro de interesse para análise do percentual de partículas menores no underflow ou overflow
public float y; // eficiência [y(d)] para o diâmetro de partícula de interesse, quando se tem a equação para obtenção da eficiência a partir do diâmetro

[y(d)]

public float yu; // Fração de partículas menores que o diâmetro de partícula de interesse (em µm) no underflow;
public float yo; // Fração de partículas menores que o diâmetro de partícula de interesse (em µm) no overflow;
public float numHidrociclonesPar; // Quantidade de hidrociclones associados em paralelo

// Adimensionais

public float stk50Eu; // Valor usado que representa o adimensional "Stokes de corte Euler"
public float Re; // Número de Reynolds
public float Eu; // Número de Euler

// O Pi usado é obtido através da função "Mathf.PI". EXP é obtido através de Mathf.Exp e exponencias por Mathf.Pow.

// Proporções aplicáveis aos hidrociclones Rietema

float rietemaDiDc = 0.28f; // Razão do Di/Dc para hidrociclones Rietema
float rietemaDoDc = 0.34f; // Razão do Do/Dc para hidrociclones Rietema
float rietemaLDc = 5.0f; // Razão do L/Dc para hidrociclones Rietema
float rietemaEde = 0.4f; // proporção que leva em consideração a parte do vortex catcher que é interna ao hidrociclone

// Parâmetros k e n encontrados por Medronho para hidrociclones Rietema à baixas concentrações de alimentação.
float rietemak1 = 0.0474f;
float rietemak2 = 371.5f;
float rietemak3 = 1218f;

float rieteman1 = 0.742f;
float rieteman2 = 8.96f;
float rieteman3 = 0.116f;
float rieteman4 = -2.12f;
float rieteman5 = 4.75f;
float rieteman6 = -0.3f;

// Proporções aplicáveis aos hidrociclones Bradley

float bradleyDiDc = (1.0f / 7.0f);
float bradleyDoDc = (1.0f / 5.0f);
float bradleyL1Dc = (1.0f / 2.0f);
float bradleyEde = (1.0f / 3.0f);
float bradleyTheta = 9.0f;

// Parâmetros k e n encontrados por Antunes e Medronho para hidrociclones Bradley à baixas concentrações de alimentação.
float bradleyk1 = 0.0550f;
float bradleyk2 = 258f;
float bradleyk3 = 1210000f;

float bradleyln1 = 0.66f;
float bradleyln2 = 12.0f;
float bradleyln3 = 0.37f;
float bradleyln4 = 0f;
float bradleyln5 = 2.63f;
float bradleyln6 = -1.12f;

// Variáveis temporárias
private float Qtemp;

// Reset de variáveis
public void ResetAllVariables()
{
    isRietema = false;
    isBradley = false;
    geometriaUsada = "";

nSeparacaoLinha = 0f; // Eficiência de separação (ET) ou Eficiência Global de Coleta Reduzida, necessária na solução por distribuição
granulométrica
nSeparacaoTotal = 0f; // Eficiência de separação (ET) ou Eficiência Global de Coleta, após considerar a razão líquida - na solução pelo método
de Medronho

rho = 1000f; // Massa específica do líquido onde se encontra a suspensão de partículas em (Kg/m³). Assume-se água.
mi = 0.001f; // Viscosidade dinâmica do líquido onde se encontra a suspensão em (Pa.s)
rhoSólido = 0f; // Massa específica dos sólidos particulados em (Kg/m³).
Cva = 0f; // Fração de volume sólido na corrente de alimentação do hidrociclone
Cvr = 0f; // Fração de volume sólido na corrente de retido do hidrociclone (underflow)
Cv = 0f; // Fração de volume sólido na corrente de operação (Igual à Cva quando associados em paralelo)
Q = 0f; // Vazão de operação (Vazão total de entrada em todos os hidrociclones, se em paralelo, ou a vazão de alimentação, se em série (em m³/s)

```

```

m³/s)      Qa = 0f; // Vazão de alimentação do hidrociclone individual. É igual à vazão de operação se a configuração dos hidrociclones for em série (em
atribuído como 0,1 (recomendado por Rietema)
µm)        Qr = 0f; // Vazão de saída na corrente de retido (Vazão de saída pelo underflow) do hidrociclone individual (em m³/s)
           dP = 0f; // Diferença de pressão de operação do hidrociclone individual (em Pa ou N/m²)
           Rl = 0f; // Razão Líquida, também chamada de Rf por Medronho. O valor, se não for estipulado ou possível de ser descoberto, é comumente
           Dc = 0f; // Diâmetro da parte cilíndrica do hidrociclone (em metros)
           dCorte = 0f; // d50 ou d* => O diâmetro de partícula caracterizado por ser separada com 50% de eficiência pelo hidrociclone (em micrômetros -
           dCorteR = 0f; // é o diâmetro de corte reduzido ... estudar melhor isso aqui (em micrômetros - µm)
           Du = 0f; // Diâmetro do underflow do hidrociclone (em metros)
           Lh = 0f; // Altura total do hidrociclone (em metros)
           Do = 0f; // Diâmetro da saída superior (overflow) do hidrociclone (em metros)
           Di = 0f; // Diâmetro da alimentação do hidrociclone (em metros)
           Da = 0f; // Diâmetro da alimentação do hidrociclone (em metros)
           e = 0f; // Altura do vortex catcher interna à parte cilíndrica do hidrociclone (em metros)
           Lh1 = 0f; // Altura da parte cilíndrica do hidrociclone (em metros)
           stk50Eu = 0f; // Valor usado que representa "Stokes de corte Euler"
           d = 0f; // Diâmetro de interesse para análise do percentual de partículas menores no underflow ou overflow
           y = 0f; // eficiência [y(d)] para o diâmetro de partícula de interesse, quando se tem a equação para obtenção da eficiência a partir do diâmetro

[y(d)]      yu = 0f; // Fração de partículas menores que o diâmetro de partícula de interesse (em µm) no underflow;
           yo = 0f; // Fração de partículas menores que o diâmetro de partícula de interesse (em µm) no overflow;

           Re = 0f;
           Eu = 0f;

           numHidrociclonesPar = 0f;
           Qtemp = 0f;

           InputFieldScript BotaoSolucaoDTA = DTASolucao.GetComponent<InputFieldScript>();
           BotaoSolucaoDTA.kRegressao = 0f;
           BotaoSolucaoDTA.coefAngular = 0f;
           BotaoSolucaoDTA.coefLinear = 0f;

           UiTexts telaResultados = toolResults.GetComponent<UiTexts>();
           telaResultados.resultsTextUpdate();

           cabecaHidrociclone.GetComponent<PartesHidrociclone>().updateDimension();
           cabecaCilindro.GetComponent<PartesHidrociclone>().updateDimension();
           entradaHidrociclone.GetComponent<PartesHidrociclone>().updateDimension();
           corpoCilindrico.GetComponent<PartesHidrociclone>().updateDimension();
           saidaOverflow.GetComponent<PartesHidrociclone>().updateDimension();
           saidaUnderflow.GetComponent<PartesHidrociclone>().updateDimension();

           resetAllText();
       }
       public void resetAllText()
       {
           var inputFieldTexts = GameObject.FindGameObjectsWithTag("InputFieldText");

           foreach (GameObject inputField in inputFieldTexts)
           {
               InputField inputFieldText = inputField.GetComponent<InputField>();

               inputFieldText.text = "";
               //inputFieldText.text = " ";
               Debug.Log(inputField.name + " : " + inputFieldText.text);
           }
       }

       // Método de solução
       public void StartSolution()
       {

           // Conferindo se o método foi realmente chamado
           Debug.Log("Solução iniciada!");

           if (dP == 0 || mi == 0 || rho == 0) { Debug.Log("Confira se não há informações faltando, como: dP, µ ou a massa específica (rho) do líquido");

return; }

           if (isRietema == false && isBradley == false)
           {
               Debug.Log("Não há uma geometria escolhida");
               ChecagemDaGeometria();
               return;
           }

           else if (isRietema)
           {
               Debug.Log("A geometria a ser usada é Rietema!");
           }

           else if (isBradley)
           {

```

```

        Debug.Log("A geometria a ser usada é Bradley!");
    }

    if (dP != 0 && mi != 0 && rho != 0 && (isBradley == true || isRietema == true)) // Hidrociclones precisam da diferença de pressão de trabalho
    para que sejam dimensionados.
    {
        if ((projToggle.isOn != true))
        {
            // Conferindo se o hidrociclone já possui dimensões (com base em Dc e Du)
            if (Dc == 0) { Debug.Log("Não há Dc, considere adicioná-lo ou marcar a opção DIMENSIONAR."); return; }

            if (Dc != 0 && rho != 0 && mi != 0 && dP != 0)
            {
                if (Qa == 0)
                {
                    if (Cva != 0)
                    {
                        VazaoPorDc(Dc, rho, mi, dP, Cva); // Obtém-se a vazão por Dc e Cva
                    }
                    else if (numHidrociclonesPar != 0 && (Q != 0 && Qa == 0))
                    {
                        Qa = Q / numHidrociclonesPar; // no caso de se saber a vazão total e o número de hidrociclones
                    }
                    else if (numHidrociclonesPar == 0 && (Q != 0 && Qa == 0)) // Se zero hidrociclones, considera-se 1 hidrociclone
                    {
                        numHidrociclonesPar = 1f;
                        Qa = Q / numHidrociclonesPar;
                    }
                }

                if (Qa != 0)
                {
                    NumeroDeReynolds(rho, Qa, mi, Dc); // Utiliza-se a vazão
                }

                // Condiçõais para o Número de Euler
                if (Re != 0 && (Cva != 0 && Cva < 0.1f))
                {
                    NumeroDeEuler(Re, Cva); // Calcula-se Euler a partir de Reynolds, usando a correlação proposta por Medronho para um Cva < 0.1;
                }
                else if (dP != 0 && Dc != 0 && rho != 0 && Qa != 0)
                {
                    NumeroDeEuler(dP, Dc, rho, Qa);
                }
                else if (Dc != 0 && Di != 0 && Do != 0 && Du != 0 && Lh != 0 && e != 0 && Cva != 0 && Re != 0)
                {
                    float tempDc = Dc * 100;
                    float tempDi = Di * 100;
                    float tempDo = Do * 100;
                    float tempDu = Du * 100;
                    float tempLh = Lh * 100;
                    float tempe = e * 100;
                    float tempCva = Cva;

                    NumeroDeEuler(Dc, Di, Do, Du, Lh, e, Cva, Re);
                }

                RazaoLiquida(); //

                StkCorteEuler(Rl, Cva);

                if (Dc != 0)
                {
                    CalcDimensoesHidrociclone(Dc);
                }

                if (numHidrociclonesPar != 0)
                {
                    Q = Qa * numHidrociclonesPar;
                }
                else if (numHidrociclonesPar == 0)
                {
                    NumeroHidrociclonesParalelo(Q, Qa);
                }

                if (rhoSolido == 0)
                {
                    Debug.Log("É necessário a massa específica das partículas sólidas para a obtenção de d50'");
                    return;
                }
                else if (rhoSolido != 0)
                {
                    d50ReduzidoPorStk(stk50Eu, mi, rho, Qa, rhoSolido, dP, Dc); // Obtenção do d50' (diâmetro de corte reduzido) a partir da equação de
                    Strokes de Corte Euler (Stk50Eu)
                }
            }
        }
    }

```

```

InputFieldScript BotaoSolucaoDTA = DTASolucao.GetComponent<InputFieldScript>();

// Eficiência global pede a distribuição granulométrica do material na alimentação
// alternativamente, pode-se descobrir a eficiência global com Rl, Cva e Cvr.

if ((BotaoSolucaoDTA.kRegressao != 0 && BotaoSolucaoDTA.coefAngular != 0))
{
    EficienciaTotalPorDTA(); // Cálculo da eficiência total - Encontrar um jeito de só fazer isso no caso de já se ter usado DTA

    CvrPorEteRl(nSeparacaoTotal, Cva, Rl);
}
else if ((BotaoSolucaoDTA.kRegressao == 0 && BotaoSolucaoDTA.coefAngular == 0) && Cvr == 0)
{
    Debug.Log("É necessário que sejam adicionadas informações referentes à análise granulométrica da alimentação");
}
else if ((BotaoSolucaoDTA.kRegressao == 0 && BotaoSolucaoDTA.coefAngular == 0) && Cvr != 0)
{
    EficTotalPorCvr(Cva, Cvr, Rl);
    Debug.Log("Utilizou-se Cva, Cvr e Rl para a obtenção da eficiência total");
}
}
}

else if (projToggle.isOn) // Nesse caso, o objetivo da questão é dimensionar o hidrociclone
{
    if (Qa == 0 && Q != 0 && numHidrociclonesPar != 0)
    {
        Qa = (Q / numHidrociclonesPar);
    }
    if (numHidrociclonesPar == 0 && Q != 0 && Qa != 0)
    {
        NumeroHidrociclonesParalelo(Q, Qa);
    }
    if (numHidrociclonesPar != 0 && Qa != 0)
    {
        Q = (numHidrociclonesPar * Qa);
    }

    DcPorVazao(Qa, rho, mi, dP, Cva);

    // Lembrar de sugerir atenção às dimensões do hidrociclone e à eficiência!
    CalcDimensoesHidrociclone(Dc);

    // Após conhecida a vazão, o Dc e a quantidade de hidrociclones, obtém-se o Rl e/ou o Cvr se necessário e possível.

    if (Rl == 0 && ((Cvr != 0 && Cva != 0 && nSeparacaoTotal != 0) || (Qa != 0 && Qr != 0 && Cva != 0 && Cvr != 0) || (Qr != 0 && Qa != 0 && nSeparacaoTotal != 0 && Cva != 0 && Cvr == 0)))
    {
        RazaoLiquida();
    }

    if (nSeparacaoTotal != 0 && Cva != 0 && Rl != 0 && Cvr == 0)
    {
        CvrPorEteRl(nSeparacaoTotal, Cva, Rl);
    }

    if (nSeparacaoTotal == 0 && (Cva != 0 && Cvr != 0 && Rl != 0))
    {
        Debug.Log("Encontra-se a eficiência total utilizando Cva, Cvr e Rl.");
        EficTotalPorCvr(Cva, Cvr, Rl);
    }
    else if (nSeparacaoTotal == 0 && nSeparacaoLinha != 0 && Rl != 0)
    {
        Debug.Log("Encontra-se Eficiência Total pela Eficiência Reduzida");
        EficienciaTotalPorDTA();
    }

    NumeroDeReynolds(rho, Qa, mi, Dc);

    // Número de Euler
    if (Re != 0 && (Cva != 0 && Cva < 0.1f))
    {
        NumeroDeEuler(Re, Cva); // Calcula-se Euler a partir de Reynolds, usando a correlação proposta por Medronho para um Cva < 0.1;
    }
    else if (dP != 0 && Dc != 0 && rho != 0 && Qa != 0)
    {
        NumeroDeEuler(dP, Dc, rho, Qa);
    }

    DiametroUnderflow();
}

```

```

    }
}

// Atualização da aba de resultados

UiTexts telaResultados = toolResults.GetComponent<UiTexts>();
telaResultados.resultsTextUpdate();

// Atualização das partes visuais do hidrociclone

cabecaHidrociclone.GetComponent<PartesHidrociclone>().updateDimension();
cabecaCilindro.GetComponent<PartesHidrociclone>().updateDimension();
entradaHidrociclone.GetComponent<PartesHidrociclone>().updateDimension();
corpoCilindrico.GetComponent<PartesHidrociclone>().updateDimension();
saidaOverflow.GetComponent<PartesHidrociclone>().updateDimension();
saidaUnderflow.GetComponent<PartesHidrociclone>().updateDimension();

}

// Outros métodos

public void CvrPorEteRl(float nSeparacaoTotal, float Cva, float Rl)
{
    Cvr = (nSeparacaoTotal * Cva) / (Rl - (Rl * Cva) + nSeparacaoTotal * Cva);
    Debug.Log("Cvr: " + Cvr);
}

// Números adimensionais
public void NumeroDeReynolds(float rho, float Qa, float mi, float Dc)
{
    Re = (4 * rho * Qa) / (Mathf.PI * mi * Dc);
    Debug.Log("Re: " + Re);
}

public void NumeroDeEuler(float dP, float Dc, float rho, float Qa)
{
    Eu = ((Mathf.Pow(Mathf.PI, 2)) * dP * (Mathf.Pow(Dc, 4))) / (8 * rho * (Mathf.Pow(Qa, 2)));
    Debug.Log("Eu: " + Eu);
}

public void NumeroDeEuler(float Re, float Cva)
{
    if (isRietema)
    {
        if (Cva <= 0.10f)
        {
            Eu = (rietemak2 * Mathf.Pow(Re, rieteman3) * Mathf.Exp(rieteman4 * Cva));
            Debug.Log("Euler certo");
        }
        else
        {
            Eu = (Mathf.Pow(Mathf.PI, 2f) * dP * Mathf.Pow(Dc, 4f)) / (8 * rho * Mathf.Pow(Qa, 2f));
            Debug.Log("Euler errado");
        }
    }
    else if (isBradley)
    {
        if (Cva <= 0.10f)
        {
            Eu = (bradleyk2 * Mathf.Pow(Re, bradley3) * Mathf.Exp(bradley4 * Cva));
        }
        else
        {
            Eu = (Mathf.Pow(Mathf.PI, 2f) * dP * Mathf.Pow(Dc, 4f)) / (8 * rho * Mathf.Pow(Qa, 2f));
        }
    }
    Debug.Log("Eu: " + Eu);
}

public void NumeroDeEuler(float Dc, float Di, float Do, float Du, float Lh, float e, float Cva, float Re)
{
    float Do2MaisDu2 = (float) (Mathf.Pow(Do, 2.0f) + Mathf.Pow(Du, 2.0f));
    Eu = (float)(21.8f * Mathf.Pow(Dc, 0.57f) * Mathf.Pow(Dc / Di, 2.61f) * Mathf.Pow((Dc / Do2MaisDu2), 0.42f) * Mathf.Pow((Dc / (Lh - e)),
0.98f) * (Mathf.Pow(Re, 0.12f) * Math.Exp(-0.51f * Cva)));
}

public void StkCorteEuler(float Rl, float Cva)
{
    if (isRietema)
    {
        stk50Eu = rietemak1 * Mathf.Pow((Mathf.Log(1 / Rl)), rieteman1) * Mathf.Exp(rieteman2 * Cva);
    }
    else if (isBradley)
    {
        stk50Eu = bradleyk1 * Mathf.Pow((Mathf.Log(1 / Rl)), bradley3) * Mathf.Exp(bradley4 * Cva);
    }

    Debug.Log("Stk50Eu: " + stk50Eu);
}
}

```

```

// Eficiência
public void EficienciaTotalPorDTA()
{
    nSeparacaoTotal = Rl + (nSeparacaoLinha * (1 - Rl));
    Debug.Log("Eficiência total de separação: " + nSeparacaoTotal);
}
public void EficTotalPorCvr(float Cva, float Cvr, float Rl)
{
    float nSepTotalTemp = ((Cva / (1 - Cva)) * ((1 - Cvr) / Cvr)) * (1 / Rl);

    nSeparacaoTotal = (1 / nSepTotalTemp);
}
public void RazaoLiquida()
{
    if (isRietema)
    {
        if (Dc != 0 && Du != 0)
        {
            Rl = rietemak3 * Mathf.Pow((Du/Dc), rieteman5) * Mathf.Pow(Eu,rieteman6);
            Debug.Log("Rl por Dc e Du");
        }
        else if (Qa != 0 && Qr != 0 && Cva != 0 && Cvr != 0)
        {
            Rl = (Qr * (1 - Cvr)) / (Qa * (1 - Cva));
        }
        else if (Qr != 0 && Qa != 0 && nSeparacaoTotal != 0 && Cva != 0 && Cvr == 0)
        {
            Rl = (Qr - (nSeparacaoTotal * Cva * Qa)) / (Qa * (1 - Cva));
        }
        else if (Cva != 0 && Cvr != 0 && nSeparacaoTotal != 0)
        {
            Rl = nSeparacaoTotal * (Cva / (1 - Cva)) * ((1 - Cvr) / Cvr);
        }
    }
    else if (isBradley)
    {
        if (Dc != 0 && Du != 0)
        {
            Rl = bradleyk3 * Mathf.Pow((Du / Dc), bradley5) * Mathf.Pow(Eu, bradley6);
        }
        else if (Qa != 0 && Qr != 0 && Cva != 0 && Cvr != 0)
        {
            Rl = (Qr * (1 - Cvr)) / (Qa * (1 - Cva));
        }
        else if (Qr != 0 && Qa != 0 && nSeparacaoTotal != 0 && Cva != 0 && Cvr == 0)
        {
            Rl = (Qr - (nSeparacaoTotal * Cva * Qa)) / (Qa * (1 - Cva));
        }
        else if (Cva != 0 && Cvr != 0 && nSeparacaoTotal != 0)
        {
            Rl = nSeparacaoTotal * (Cva / (1 - Cva)) * ((1 - Cvr) / Cvr);
        }
    }
    Debug.Log("Rl: " + Rl);
}

public void DiametroUnderflow()
{
    float tempDu;

    if (isRietema)
    {
        // Equação consultada do Peçanha. Resolvida utilizando Usando Rl(Du, Dc, Eu) e Eu(dP, rho, Vc).
        Du = 0.227f * Mathf.Pow((Mathf.Pow(Dc, 5.95f) * Rl * Mathf.Pow((dP / (rho * Mathf.Pow(Qa, 2))), 0.3f)), 0.211f);
    }
    else if (isBradley)
    {
        // Equação encontrada a partir de Rl(Du, Dc, Eu) e número de Euler. Parâmetros por Antunes.
        tempDu = (Rl * Mathf.Pow(Dc, 2.63f)) / (bradleyk3 * Mathf.Pow(Eu, -1.12f));

        Du = Mathf.Pow(tempDu, (1/2.63f));
    }
    Debug.Log("Du: " + Du);
}

public void VazaoPorDc(float Dc, float rho, float mi, float DP, float Cva)
{
    if (isRietema)
    {
        Qtemp = (Dc * (Mathf.Pow(dP, 0.24f) * Mathf.Pow(mi, 0.028f))) / (4.0f * Mathf.Pow(rho, 0.27f) * Mathf.Exp(-0.52f * Cva));
        Qa = Mathf.Pow(Qtemp, 1 / 0.51f);
    }
    else if (isBradley)
    {
        Qtemp = (Dc * (Mathf.Pow(dP, 0.23f) * Mathf.Pow(mi, 0.085f))) / (3.5f * Mathf.Pow(rho, 0.31f));
    }
}

```

```

        Qa = Mathf.Pow(Qtemp, 1 / 0.54f);
    }
    Debug.Log("Qa: " + Qa);
}
public void DcPorVazao(float Qa, float rho, float mi, float DP, float Cva)
{
    if (isRietema)
    {
        Dc = (4f * (Mathf.Pow(rho, 0.27f)) * (Mathf.Pow(Qa, 0.51f))) / ((Mathf.Pow(mi, 0.028f)) * (Mathf.Pow(dP, 0.24f)));
    }
    else if (isBradley)
    {
        Dc = (3.5f * (Mathf.Pow(rho, 0.31f)) * (Mathf.Pow(Qa, 0.54f))) / ((Mathf.Pow(mi, 0.085f)) * (Mathf.Pow(dP, 0.23f)));
    }

    Debug.Log("Dc: " + Dc);
}
public void d50ReduzidoPorStk(float stk50Eu, float mi, float rho, float Qa, float rhoSolido, float dP, float Dc)
{
    float dCorteRTemp;
    float dRho = rhoSolido - rho;
    float dPxDc = dP * Dc;
    dCorteRTemp = (stk50Eu * (36 * mi * rho * Qa)) / ((Mathf.PI * dRho * dPxDc));
    dCorteR = (float)Mathf.Pow(dCorteRTemp, (1f / 2f));
    Debug.Log("d* Temp: " + dCorteRTemp);
    Debug.Log("d*": " + dCorteR);
}

public void NumeroHidrociclonesParalelo(float Q, float Qa)
{
    if (Q != 0.0f && (Q/Qa != 1.0f))
    {
        float tempResult = ((Q / Qa));

        numHidrociclonesPar = tempResult;

        Debug.Log("Numero de hidrociclones em paralelo: " + numHidrociclonesPar);
    }
    else if (Q == 0.0f)
    {
        //MakeQEqualToQa(Q, Qa);

        float tempResult = 1.0f;

        numHidrociclonesPar = tempResult;

        Debug.Log("Numero de hidrociclones em paralelo: " + numHidrociclonesPar);
    }
}
public void MakeQEqualToQa(float Q, float Qa)
{
    Q = Qa;
}

public void EficienciaGlobalDeColetaRed()
{
    nSeparacaoLinha = (nSeparacaoTotal - RL) / (1 - RL);
    Debug.Log("Efic. Global de Coleta Reduzida: " + nSeparacaoLinha);
}

public void ChecagemDaGeometria()
{
    if (Dc != 0)
    {
        if ((Lh / Dc) == rietemaLDc)
        {
            Debug.Log("A geometria apropriada é Rietema Lh/Dc");
            isRietema = true;
        }
        else if ((Lh1 / Dc) == bradleyL1Dc)
        {
            Debug.Log("A geometria apropriada é Bradley Lh1/Dc");
            isBradley = true;
        }
        else if ((Di / Dc) == rietemaDiDc)
        {
            Debug.Log("A geometria apropriada é Rietema Di/Dc");
            isRietema = true;
        }
        else if ((Di / Dc) == bradleyDiDc)
        {
            Debug.Log("A geometria apropriada é Bradley Di/Dc: " + (Di/Dc) + " bradleyDiDc " + bradleyDiDc);
            isBradley = true;
        }
        else if ((Do/Dc) == rietemaDoDc)
        {

```

```

        Debug.Log("A geometria apropriada é Rietema Do/Dc");
        isRietema = true;
    }
    else if ((Do / Dc) == bradleyDoDc)
    {
        Debug.Log("A geometria apropriada é Bradley Do/Dc");
        isBradley = true;
    }
    else if ((e / Dc) == rietemaeDc)
    {
        Debug.Log("A geometria apropriada é Rietema e/Dc");
        isRietema = true;
    }
    else if ((e / Dc) == bradleyeDc)
    {
        Debug.Log("A geometria apropriada é Bradley e/Dc");
        isBradley = true;
    }
    else
    {
        Debug.Log("Não há geometria adequada para esse modelo de hidrociclone. Os cálculos serão tratados de uma forma alternativa.");
        return;
    }
}

else if (Dc == 0)
{
    Debug.Log("Não há valor para Dc, logo a checagem não será realizada");
}
}

public void Eficienciasyuyo()
{
    float tempyu = yu * nSeparacaoTotal;
    float tempdenom = 1 - nSeparacaoTotal;
    yo = (y - (yu * nSeparacaoTotal)) / (1 - nSeparacaoTotal);
    Debug.Log("yo = " + yo);
}

public void CalcDimensoesHidrociclone(float Dc)
{
    if (isRietema)
    {
        //rietemaDiDc = 0.28f; // Razão do Di/Dc para hidrociclones Rietema
        //rietemaDoDc = 0.34f; // Razão do Do/Dc para hidrociclones Rietema
        //rietemaLDc = 5.0f; // Razão do L/Dc para hidrociclones Rietema
        //rietemaeDc = 0.4f; // proporção que leva em consideração a parte do vortex catcher que é interna ao hidrociclone

        Di = rietemaDiDc * Dc;
        Do = rietemaDoDc * Dc;
        Lh = rietemaLDc * Dc;
        e = rietemaeDc * Dc;
    }
    else if (isBradley)
    {
        //bradleyDiDc = (1 / 7);
        //bradleyDoDc = (1 / 5);
        //bradleyL1Dc = (1 / 2);
        //bradleyeDc = (1 / 3);
        //bradleyTheta = 9;

        //float trigonBradleyRad;
        // Debug.Log("trigonBradleyRad: " + trigonBradleyRad);
        Debug.Log("bradleyTheta: " + bradleyTheta);
        Debug.Log("bradleyTheta/2: " + bradleyTheta/2);
        Debug.Log("PI: " + Math.PI);
        Debug.Log("90 - theta/2: " + (90.0f - (bradleyTheta/2.0f)));
        Debug.Log("90 - theta/2: " + (90.0f - (bradleyTheta/2.0f)) * (float) Math.PI);
        Debug.Log("90 - theta/2: " + (90.0f - (bradleyTheta/2.0f)) * (float) Math.PI / 180.0f);

        Di = bradleyDiDc * Dc;
        Do = bradleyDoDc * Dc;
        Lh1 = bradleyL1Dc * Dc;
        e = bradleyeDc * Dc;
        Lh = Mathf.Tan((90.0f - (bradleyTheta / 2.0f)) * (float) Math.PI / 180.0f) * (Dc / 2) + Lh1;
        //Debug.Log("Tangente: " + Mathf.Tan(trigonBradleyRad));
    }
}
}
}

```

02 – InputFieldScript

Esse trecho é essencial para a solução por distribuição granulométrica. Aqui se encontra a interação entre os campos de entrada de valores e o processo de solução por linearização, adequação ao modelo RRB e solução da integral de eficiência de coleta reduzida.

```
using JetBrains.Annotations;
using Packages.Rider.Editor.UnitTesting;
using System.Collections;
using System.Collections.Generic;
using UnityEditor.UIElements;
using UnityEngine;

public class InputFieldScript : MonoBehaviour
{
    public GameObject RietemaVazio;
    public GameObject DTAOverlay;
    public GameObject gameCanvas;
    public GameObject toolResults;
    public GameObject ExplanationButton;

    public float diametroDTA = 0f;
    public float eficienciaDTA = 0f;
    public float coefAngular;
    public float coefLinear;
    public float kRegressao;

    public class DTA
    {
        public float d; // Valor do diâmetro de partícula que resulta na porcentagem de eficiência em y;
        public float y; // Eficiência relativa ao diâmetro de partícula d. => y(d)
    }

    List<DTA> listOfDTA = new List<DTA>();
    List<float> listOfX = new List<float>();
    List<float> listOfY = new List<float>();

    void Start()
    {
        //DTAOverlay.SetActive(false);
        EquationsScript scriptHidrociclone = RietemaVazio.GetComponent<EquationsScript>();
        if (this.name == "DTAaddbutton" || this.name == "DTAResetButton" || this.name == "DTAConsulta" || this.name == "DTASolucao")
        {
            GameController canvasController = gameCanvas.GetComponent<GameController>();
        }
        if (this.name == "ListSize")
        {
            GameObject ConsultPanel = transform.Find("ConsultPanel").gameObject;
            ConsultPanel.SetActive(false);
        }
    }

    // Adicionar botão "Consultar" que permite que sejam consultados os pontos já adicionados.

    public void GetInput(string value)
    {
        // RietemaScript é um nome ruim, ScriptHidrociclone seria melhor. Entretanto, ainda não atualizei!
        EquationsScript RietemaScript = RietemaVazio.GetComponent<EquationsScript>();
        GameController canvasController = gameCanvas.GetComponent<GameController>();

        if (this.name == "rho")
        {
            Debug.Log("You entered: " + value + " for " + this.name);
            Debug.Log("You changed " + this.name + " from " + RietemaScript.rho + " to " + value);
            if (value == null || value == "") { return; }
            RietemaVazio.GetComponent<EquationsScript>().rho = float.Parse(value);
        }

        else if (this.name == "mi")
        {
            Debug.Log("You entered: " + value + " for " + this.name);
            Debug.Log("You changed " + this.name + " from " + RietemaScript.mi + " to " + value);
            if (value == null || value == "") { return; }
            RietemaVazio.GetComponent<EquationsScript>().mi = float.Parse(value);
        }

        if (this.name == "rhoSolido")
        {
            Debug.Log("You entered: " + value + " for " + this.name);
        }
    }
}
```

```

        Debug.Log("You changed " + this.name + " from " + RietemaScript.rhoSolido + " to " + value);
        if (value == null || value == "") { return; }
        RietemaVazio.GetComponent<EquationsScript>().rhoSolido = float.Parse(value);
    }

    else if (this.name == "Q")
    {
        Debug.Log("You entered: " + value + " for " + this.name);
        Debug.Log("You changed " + this.name + " from " + RietemaScript.Q + " to " + value);
        if (value == null || value == "") { return; }
        RietemaVazio.GetComponent<EquationsScript>().Q = float.Parse(value);
    }

    else if (this.name == "Qa")
    {
        Debug.Log("You entered: " + value + " for " + this.name);
        Debug.Log("You changed " + this.name + " from " + RietemaScript.Qa + " to " + value);
        if (value == null || value == "") { return; }
        RietemaVazio.GetComponent<EquationsScript>().Qa = float.Parse(value);
    }

    else if (this.name == "Qr")
    {
        Debug.Log("You entered: " + value + " for " + this.name);
        Debug.Log("You changed " + this.name + " from " + RietemaScript.Qr + " to " + value);
        if (value == null || value == "") { return; }
        RietemaVazio.GetComponent<EquationsScript>().Qr = float.Parse(value);
    }

    else if (this.name == "dP")
    {
        Debug.Log("You entered: " + value + " for " + this.name);
        Debug.Log("You changed " + this.name + " from " + RietemaScript.dP + " to " + value);
        if (value == null || value == "") { return; }
        RietemaVazio.GetComponent<EquationsScript>().dP = float.Parse(value);
    }

    else if (this.name == "Rl" || this.name == "RIAlt")
    {
        Debug.Log("You entered: " + value + " for " + this.name);
        Debug.Log("You changed Rl from " + RietemaScript.Rl + " to " + value);
        if (value == null || value == "") { return; }
        RietemaVazio.GetComponent<EquationsScript>().Rl = float.Parse(value);
    }

    else if (this.name == "Et")
    {
        Debug.Log("You entered: " + value + " for " + this.name);
        Debug.Log("You changed " + this.name + " from " + RietemaScript.nSeparacaoLinha + " to " + value);
        if (value == null || value == "") { return; }
        RietemaVazio.GetComponent<EquationsScript>().nSeparacaoLinha = float.Parse(value);
    }

    else if (this.name == "Dc")
    {
        Debug.Log("You entered: " + value + " for " + this.name);
        Debug.Log("You changed " + this.name + " from " + RietemaScript.Dc + " to " + value);
        if (value == null || value == "") { return; }
        RietemaVazio.GetComponent<EquationsScript>().Dc = float.Parse(value);
    }

    else if (this.name == "Du")
    {
        Debug.Log("You entered: " + value + " for " + this.name);
        Debug.Log("You changed " + this.name + " from " + RietemaScript.Du + " to " + value);
        if (value == null || value == "") { return; }
        RietemaVazio.GetComponent<EquationsScript>().Du = float.Parse(value);
    }

    else if (this.name == "Do")
    {
        Debug.Log("You entered: " + value + " for " + this.name);
        Debug.Log("You changed " + this.name + " from " + RietemaScript.Do + " to " + value);
        if (value == null || value == "") { return; }
        RietemaVazio.GetComponent<EquationsScript>().Do = float.Parse(value);
    }

    else if (this.name == "Di")
    {
        Debug.Log("You entered: " + value + " for " + this.name);
        Debug.Log("You changed " + this.name + " from " + RietemaScript.Di + " to " + value);
        if (value == null || value == "") { return; }
        RietemaVazio.GetComponent<EquationsScript>().Di = float.Parse(value);
    }

    else if (this.name == "e")

```

```

{
    Debug.Log("You entered: " + value + " for " + this.name);
    Debug.Log("You changed " + this.name + " from " + RietemaScript.e + " to " + value);
    if (value == null || value == "") { return; }
    RietemaVazio.GetComponent<EquationsScript>().e = float.Parse(value);
}

else if (this.name == "Lh")
{
    Debug.Log("You entered: " + value + " for " + this.name);
    Debug.Log("You changed " + this.name + " from " + RietemaScript.Lh + " to " + value);
    if (value == null || value == "") { return; }
    RietemaVazio.GetComponent<EquationsScript>().Lh = float.Parse(value);
}

else if (this.name == "Lh1")
{
    Debug.Log("You entered: " + value + " for " + this.name);
    Debug.Log("You changed " + this.name + " from " + RietemaScript.Lh1 + " to " + value);
    if (value == null || value == "") { return; }
    RietemaVazio.GetComponent<EquationsScript>().Lh1 = float.Parse(value);
}

else if (this.name == "numCiclonosParalelos")
{
    Debug.Log("You entered: " + value + " for " + this.name);
    Debug.Log("You changed " + this.name + " from " + RietemaScript.numHidroiclonosPar + " to " + value);
    if (value == null || value == "") { return; }
    RietemaVazio.GetComponent<EquationsScript>().numHidroiclonosPar = float.Parse(value);
}

else if (this.name == "Cva")
{
    Debug.Log("You entered: " + value + " for " + this.name);
    Debug.Log("You changed " + this.name + " from " + RietemaScript.Cva + " to " + value);
    if (value == null || value == "") { return; }
    RietemaVazio.GetComponent<EquationsScript>().Cva = float.Parse(value);
}

else if (this.name == "Cvr")
{
    Debug.Log("You entered: " + value + " for " + this.name);
    Debug.Log("You changed " + this.name + " from " + RietemaScript.Cvr + " to " + value);
    if (value == null || value == "") { return; }
    RietemaVazio.GetComponent<EquationsScript>().Cvr = float.Parse(value);
}

else if (this.name == "Et")
{
    Debug.Log("You entered: " + value + " for " + this.name);
    Debug.Log("You changed " + this.name + " from " + RietemaScript.nSeparacaoTotal + " to " + value);
    if (value == null || value == "") { return; }
    RietemaVazio.GetComponent<EquationsScript>().nSeparacaoTotal = float.Parse(value);
}

else if (this.name == "mcoef")
{
    Debug.Log("You entered: " + value + " for " + this.name);
    Debug.Log("You changed " + this.name + " from " + coefAngular + " to " + value);
    if (value == null || value == "") { return; }
    canvasController.coefAngular = float.Parse(value);
    coefAngular = float.Parse(value);
}

else if (this.name == "kreg")
{
    Debug.Log("You entered: " + value + " for " + this.name);
    Debug.Log("You changed " + this.name + " from " + kRegressao + " to " + value);
    if (value == null || value == "") { return; }
    canvasController.kRegressao = float.Parse(value);
    kRegressao = float.Parse(value);
}

else if (this.name == "d50r")
{
    Debug.Log("You entered: " + value + " for " + this.name);
    Debug.Log("You changed " + this.name + " from " + RietemaScript.dCorteR + " to " + value);
    if (value == null || value == "") { return; }
    RietemaVazio.GetComponent<EquationsScript>().dCorteR = float.Parse(value);
}
}

public void ConfirmGeometryOnClick()
{
    EquationsScript ScriptHidroiclonos = RietemaVazio.GetComponent<EquationsScript>();

```

```

        if (this.name == "Rietema") { ScriptHidrociClone.isBradley = false; ScriptHidrociClone.isRietema = true; ScriptHidrociClone.geometriaUsada =
"Rietema"; ScriptHidrociClone.StartSolution(); }
        else if (this.name == "Bradley") { ScriptHidrociClone.isRietema = false; ScriptHidrociClone.isBradley = true; ScriptHidrociClone.geometriaUsada
= "Bradley"; ScriptHidrociClone.StartSolution(); }
        else if (this.name != "Rietema" && this.name != "Bradley") { Debug.Log("Não há uma geometria selecionada."); ScriptHidrociClone.isBradley
= false; ScriptHidrociClone.isRietema = false; ScriptHidrociClone.StartSolution(); }
    }
    public void DTAButtonOnClick()
    {
        GameObject DTAPanel = transform.Find("Panel").gameObject;

        if (DTAPanel.activeSelf == true)
        {
            DTAPanel.SetActive(false);
        }
        else if (DTAPanel.activeSelf == false)
        {
            DTAPanel.SetActive(true);
        }
    }
    public void ResultsButtonOnClick()
    {
        GameObject ResultsPanel = transform.Find("PanelResult").gameObject;

        if (ResultsPanel.activeSelf == true)
        {
            ResultsPanel.SetActive(false);
            ExplanationButton.SetActive(false);
        }
        else if (ResultsPanel.activeSelf == false)
        {
            ResultsPanel.SetActive(true);
            ExplanationButton.SetActive(true);
        }
    }
    public void ExplanationButtonOnClick()
    {
        GameObject ExplanationPanel = transform.Find("PanelExplanation").gameObject;

        if (ExplanationPanel.activeSelf == true)
        {
            ExplanationPanel.SetActive(false);
        }
        else if (ExplanationPanel.activeSelf == false)
        {
            ExplanationPanel.SetActive(true);
        }
    }
    public void GetDTAInput(string value)
    {
        GameController canvasController = gameCanvas.GetComponent<GameController>();
        if (this.name == "diametro")
        {
            Debug.Log("You entered: " + value + " for " + this.name);
            //Debug.Log("You changed " + this.name + " from " + diametroDTA + " to " + value);
            if (value == null || value == "") { return; }
            diametroDTA = float.Parse(value);
            canvasController.diametroDTA = float.Parse(value);

        }
        else if (this.name == "eficiencia")
        {
            Debug.Log("You entered: " + value + " for " + this.name);
            //Debug.Log("You changed " + this.name + " from " + eficienciaDTA + " to " + value);
            if (value == null || value == "") { return; }
            eficienciaDTA = float.Parse(value);
            canvasController.eficienciaDTA = (float.Parse(value));
        }
    }
    public void GetDTAAddButton()
    {
        GameController canvasController = gameCanvas.GetComponent<GameController>();
        DTA currentDTA = new DTA();

        if (this.name == "DTAaddbutton")
        {
            canvasController.ControllerGetDTAAddButton();

            UiTexts uiText = toolResults.GetComponent<UiTexts>();
            uiText.ListSizeResetText(canvasController);
            uiText.ListSizeUpdateText(canvasController);
        }
    }
}

```

```

public void GetDTAResetButton()
{
    GameController canvasController = gameCanvas.GetComponent<GameController>();

    if (this.name == "DTAResetButton")
    {
        canvasController.listOfDTA.Clear();

        UiTexts uiText = toolResults.GetComponent<UiTexts>();
        uiText.ListSizeResetText(canvasController);

        Debug.Log("Os pontos previamente adicionados foram removidos.");
    }
    else if (this.name == "AllResetButton")
    {
        EquationsScript hidrociclone = RietemaVazio.GetComponent<EquationsScript>();

        canvasController.listOfDTA.Clear();
        hidrociclone.ResetAllVariables();

        UiTexts uiText = toolResults.GetComponent<UiTexts>();
        uiText.ListSizeResetText(canvasController);
    }
}

public void GetDTAListSize()
{
    GameController canvasController = gameCanvas.GetComponent<GameController>();

    if (this.name == "ListSize")
    {
        GameObject ConsultPanel = transform.Find("ConsultPanel").gameObject;
        UiTexts uiText = toolResults.GetComponent<UiTexts>();

        Debug.Log("You have added " + canvasController.listOfDTA.Count + " values to the list.");

        if (ConsultPanel.activeSelf == true)
        {
            ConsultPanel.SetActive(false);
        }
        else if (ConsultPanel.activeSelf == false)
        {
            ConsultPanel.SetActive(true);

            uiText.ListSizeResetText(canvasController);

            uiText.ListSizeUpdateText(canvasController);
        }
    }
}

public void GetDTASolution()
{
    GameController canvasController = gameCanvas.GetComponent<GameController>();
    EquationsScript scriptHidrociclone = RietemaVazio.GetComponent<EquationsScript>();
    UiTexts telaResultados = toolResults.GetComponent<UiTexts>();

    if (this.name == "DTASolucao")
    {
        LinearizarPorRRB();
        RegressaoLinearRRB();
        RegraDoTrapezio();
        scriptHidrociclone.EficienciaTotalPorDTA();
        telaResultados.resultsTextUpdate();
    }
    else if (this.name == "ModeloSolucao")
    {
        if (kRegressao == 0 && coefAngular == 0)
        {
            coefAngular = canvasController.coefAngular;
            kRegressao = canvasController.kRegressao;
        }

        EficienciaYPeloModelo();
        RegraDoTrapezio();
        RegraDoTrapezioEficUnderflow();
        scriptHidrociclone.EficienciaTotalPorDTA();
        telaResultados.resultsTextUpdate();
    }
    else { Debug.Log("Faltam informações para iniciar a solução"); }
}

public void LinearizarPorRRB()
{
    GameController canvasController = gameCanvas.GetComponent<GameController>();

```

```

foreach (var item in canvasController.listOfDTA)
{
    // Linearizando todos os pontos da lista gerada pelo usuário.
    float novoY = Mathf.Log(Mathf.Log(1 / (1 - item.y))); //  $\ln(\ln(1/(1-y))) = m \cdot \ln(d) - m \cdot \ln(k)$ 
    Debug.Log("novoY: " + novoY);

    float novoX = Mathf.Log(item.d);
    Debug.Log("novoX: " + novoX);

    listOfX.Add(novoX);
    canvasController.listOfX.Add(novoX);

    listOfY.Add(novoY);
    canvasController.listOfY.Add(novoY);

    // Fazendo-se regressão linear, encontra-se a e b; a = m

    //  $\ln(\ln(1/(1-y))) = m \cdot \ln(d) - m \cdot \ln(k)$  <====>  $y = (a \cdot x) + b$ 

    //  $\ln(\ln(1/(1-y))) = y$ 
    // a = m
    // b = -m * ln(k)

    // Sabendo-se a:

    // Para achar k:

    // a = m

    // b = y - (a*x)
    // b = -a * ln(k)

    // ln(k) = (b/-a)
    // k = exp(b/-m)

    // Mathf.Log(Mathf.Log(1 / (1 - item.y)));
}
}
public void RegressaoLinearRRB()
{
    GameController canvasController = gameCanvas.GetComponent<GameController>();
    float deltaY = canvasController.listOfY[(listOfY.Count - 1)] - canvasController.listOfY[0];

    float deltaX = canvasController.listOfX[(listOfX.Count - 1)] - canvasController.listOfX[0];

    coefAngular = deltaY / deltaX;
    canvasController.coefAngular = deltaY / deltaX;
    Debug.Log("Coeficiente Angular: " + coefAngular);

    coefLinear = canvasController.listOfY[(canvasController.listOfY.Count - 1)] - coefAngular *
    canvasController.listOfX[(canvasController.listOfX.Count - 1)];
    canvasController.coefLinear = canvasController.listOfY[(canvasController.listOfY.Count - 1)] - canvasController.coefAngular *
    canvasController.listOfX[(canvasController.listOfX.Count - 1)];
    Debug.Log("Coeficiente Linear: " + coefLinear);

    kRegressao = Mathf.Exp(coefLinear / (-1*coefAngular));
    canvasController.kRegressao = Mathf.Exp(coefLinear / (-1*coefAngular));
    Debug.Log("kRegressao: " + kRegressao);
}
public float IntegralEficTotalRed(float yvalue)
{
    EquationsScript scriptHidrociclone = RietemaVazio.GetComponent<EquationsScript>();
    GameController canvasController = gameCanvas.GetComponent<GameController>();

    Debug.Log("Integral Efic Total");

    if (kRegressao == 0 || coefAngular == 0)
    {
        coefAngular = canvasController.coefAngular;
        kRegressao = canvasController.kRegressao;
    }
    else { }

    // Mesclando partes da equação em uma variável

    float d50r = scriptHidrociclone.dCorteR * 1000000;

    float ytemp = Mathf.Pow(Mathf.Log(1 / (1 - yvalue)), (1/coefAngular));

    float leftSide = -0.693f / Mathf.Pow(d50r, 3);

    float ytempxk = ytemp * kRegressao;

    float RightSide = Mathf.Pow(ytempxk, 3);

```

```

float preexp = RightSide * leftSide;

float posexp = Mathf.Exp(preexp);

// Finalizando o cálculo

float functionresult = (1 - posexp);
Debug.Log("Result for "+ yvalue + ": " + functionresult + " : " + posexp);
return functionresult;
}
public float IntegralYParaSolucaoModelo(float yvalue)
{
    EquationsScript scriptHidrociclone = RietemaVazio.GetComponent<EquationsScript>();
    GameController canvasController = gameCanvas.GetComponent<GameController>();

    Debug.Log("Integral pelo modelo!");

    if (kRegressao == 0 || coefAngular == 0)
    {
        coefAngular = canvasController.coefAngular;
        kRegressao = canvasController.kRegressao;
    }
    else { }

    float d50r = scriptHidrociclone.dCorteR * 1000000;

    float ytemp = Mathf.Pow(Mathf.Log(1 / (1 - yvalue)), (1 / coefAngular));

    float leftSide = -0.693f / Mathf.Pow(d50r, 3);

    float ytempkx = ytemp * kRegressao;

    float RightSide = Mathf.Pow(ytempkx, 3);

    float preexp = RightSide * leftSide;

    float posexp = Mathf.Exp(preexp);

    float functionresult = (1 - posexp);
    Debug.Log("Result for " + yvalue + ": " + functionresult);

    return functionresult;
}
public void RegraDoTrapezio()
{
    GameController canvasController = gameCanvas.GetComponent<GameController>();
    EquationsScript scriptHidrociclone = RietemaVazio.GetComponent<EquationsScript>();

    List<float> listOfResults = new List<float>();
    float tempDelta = 0.015f;
    float result = 0.00f;
    Debug.Log("result no começo: " + result);
    float newResult = 0.00f;
    Debug.Log("newResult no começo: " + newResult);
    int count = 0;

    //for (float i = 0.005f; i < 0.990f; i = i+tempDelta)
    //{
    //    result = result + IntegralEficTotalRed(i);
    //    listOfResults.Add(result);
    //    count++;
    //}

    for (float i = 0.005f; i < 0.990f; i = i + tempDelta)
    {
        float result1 = 0.0f;
        float result2 = 0.0f;
        float secondResult = 0.0f;

        result1 = IntegralEficTotalRed(i);
        result2 = IntegralEficTotalRed(i + tempDelta);

        secondResult = ((result1 + result2) * tempDelta) / 2;

        result = result + secondResult;

        count++;
    }

    // Após o for loop:

    newResult = result;
    //newResult = secondResult / listOfResults.Count;
    Debug.Log("O resultado da Eficiencia Total Reduzida (ET) é: " + newResult);
}

```

```

//Debug.Log("O resultado da Eficiencia Total (ET) é: " + newResult);
Debug.Log("listOfResults.count:" + count);
//Debug.Log("listOfResults.count:" + listOfResults.Count);
scriptHidroiclone.nSeparacaoLinha = newResult;

scriptHidroiclone.EficienciaTotalPorDTA();
}
public void RegraDoTrapezioEficUnderflow()
{
    GameController canvasController = gameCanvas.GetComponent<GameController>();
    EquationsScript scriptHidroiclone = RietemaVazio.GetComponent<EquationsScript>();

    List<float> listOfResults = new List<float>();
    float tempy = scriptHidroiclone.y;
    float tempDelta = 0.015f;
    float newResult = 0.00f;
    float result = 0.00f;
    Debug.Log("result no começo: " + result);
    Debug.Log("newResult no começo: " + newResult);
    int count = 0;

    for (float i = 0.005f; i < tempy; i = i + tempDelta)
    {
        float result1 = 0.0f;
        float result2 = 0.0f;
        float secondResult = 0.0f;

        result1 = IntegralYParaSolucaoModelo(i);
        result2 = IntegralYParaSolucaoModelo(i + tempDelta);

        secondResult = ((result1 + result2) * tempDelta)/2;

        result = result + secondResult;

        count++;
    }

    // Após o for loop:

    //result = result / count;
    Debug.Log("Resultado Integral de y(d) de 0 a y: " + result);
    Debug.Log("listOfResults.count:" + count);

    newResult = (1 / scriptHidroiclone.nSeparacaoTotal) * ((scriptHidroiclone.Rl * tempy) + ((1 - scriptHidroiclone.Rl) * result));

    Debug.Log("O resultado da Eficiência pelo underflow (yu) é: " + newResult);

    scriptHidroiclone.yu = newResult;

    scriptHidroiclone.Eficienciasyuyo();
}
public void EficienciaYPeloModelo()
{
    GameController canvasController = gameCanvas.GetComponent<GameController>();
    EquationsScript scriptHidroiclone = RietemaVazio.GetComponent<EquationsScript>();

    // Método para quando já se possui a análise granulométrica linearizada pelo modelo RRB, formando uma função função y(d) linear.
    // O modelo segue o formato:
    // y = 1 - EXP( - (d / k) ^ m), com d em µm. O usuário deve inserir os valores k e m.
    // float tempDKReg = scriptHidroiclone.d / kRegressao;
    scriptHidroiclone.y = 1 - (Mathf.Exp(-Mathf.Pow((scriptHidroiclone.d / kRegressao), coefAngular)));
    Debug.Log("y = " + scriptHidroiclone.y);
}
}

```

03 – PartesHidroiclone

Esse trecho se refere à montagem da representação tridimensional do hidroiclone calculado pela ferramenta.

```

using System;
using System.Collections;
using System.Collections.Generic;
using UnityEditorInternal;
using UnityEngine;

public class PartesHidroiclone : MonoBehaviour
{
    float rietemaDiDc = 0.28f;

```

```

float rietemaDoDc = 0.34f;
float rietemaLDc = 5.0f;
float rietemaEdc = 0.4f;
float bradleyDiDc = (1.0f / 7.0f);
float bradleyDoDc = (1.0f / 5.0f);
float bradleyLiDc = (1.0f / 2.0f);
float bradleyEdc = (1.0f / 3.0f);

[SerializeField]
public GameObject hidrocicloneGameObj;
public GameObject tampaHidrociclone;
public GameObject cilindroCabeca;
public GameObject entradaHidrociclone;
public GameObject corpoCilindrico;
public GameObject saidaOverflow;
public GameObject saidaUnderflow;

float escalaRepresentacao = 1000.0f;

public void updateDimension()
{
    EquationsScript hidrociclone = hidrocicloneGameObj.GetComponent<EquationsScript>();

    Debug.Log("Atualizando dimensoes para " + this.name);

    // Lh = altura total do hidrociclone || Lh1 = altura da parte cilíndrica

    float alturaConica = hidrociclone.Lh - hidrociclone.Lh1;

    float coneYPos;

    if (this.gameObject == tampaHidrociclone)
    {
        var tempX = escalaRepresentacao * (float)Math.Round(hidrociclone.Dc, 3);
        var tempY = escalaRepresentacao * (float)Math.Round(hidrociclone.Dc, 3);
        var tempZ = 5.0f;

        this.gameObject.transform.localScale = new Vector3(tempX, tempY, tempZ);
    }
    else if (this.gameObject == cilindroCabeca)
    {
        var tempX = escalaRepresentacao * (float)Math.Round(hidrociclone.Dc, 3);
        var tempY = escalaRepresentacao * (float)Math.Round(hidrociclone.Dc, 3);
        var tempZ = this.gameObject.transform.localScale.z;

        this.gameObject.transform.localScale = new Vector3(tempX, tempY, tempZ);
    }
    else if (this.gameObject == corpoCilindrico)
    {
        GameObject coneTemp = saidaUnderflow.gameObject;

        var tempX = escalaRepresentacao * (float)Math.Round(hidrociclone.Dc, 3);
        var tempY = escalaRepresentacao * (float)Math.Round(hidrociclone.Dc, 3);
        var tempZ = (escalaRepresentacao * (float)Math.Round(hidrociclone.Lh1, 3))/3; // old: escala/10

        var xPos = this.gameObject.transform.localPosition.x;
        var yPos = 2.95f - (tempZ/100); // old: tempZ/100
        var zPos = this.gameObject.transform.localPosition.z;

        // (yPos, zScale) --> (2.9, 5); (2.85, 10); (2.8, 15); (2.75, 20); (2.5, 45) -->
        // (zScalecc, yPosccu) --> (5, 2.35); (10, 2.25); (15, 2.15); (45, 1.55);

        this.gameObject.transform.localScale = new Vector3(tempX, tempY, tempZ);
        this.gameObject.transform.localPosition = new Vector3(xPos, yPos, zPos);

        var conexPos = coneTemp.transform.localPosition.x;
        coneYPos = 2.95f - (2 * (tempZ / 100)); // old: tempZ/100
        var coneZPos = coneTemp.transform.localPosition.z;

        coneTemp.GetComponent<PartesHidrociclone>().saidaUnderflowUpdate(conexPos, coneYPos, coneZPos);
    }
    else if (this.gameObject == entradaHidrociclone)
    {
        float escalaEntradaCabecaX = (2000.0f/6f);

        var tempX = 0.8f * cilindroCabeca.transform.localScale.z;
        var tempY = cilindroCabeca.transform.localScale.z - (cilindroCabeca.transform.localScale.z * 0.2f);
        var tempZ = escalaEntradaCabecaX * 2f * (float)Math.Round(hidrociclone.Dc, 3);

        this.gameObject.transform.localScale = new Vector3(tempX, tempY, tempZ);

        float DcPosx = -0.5f - (hidrociclone.Dc);

```

```

float DcPosy = 3.2f;
float DcPosz = (hidrociclone.Dc * 10) - (tempx / 75);

this.gameObject.transform.localPosition = new Vector3(DcPosx, DcPosy, DcPosz);
}
else if (this.gameObject == saidaOverflow)
{
    var tempx = escalaRepresentacao * (float) Math.Round(hidrociclone.Do, 3);
    var tempy = escalaRepresentacao * (float) Math.Round(hidrociclone.Do, 3);
    var tempz = this.gameObject.transform.localScale.z;

    this.gameObject.transform.localScale = new Vector3(tempx, tempy, tempz);
}
}

public void saidaUnderflowUpdate(float conexPos, float coneyPos, float coneZPos)
{
    EquationsScript hidrociclone = hidrocicloneGameObj.GetComponent<EquationsScript>();
    float alturaConica = hidrociclone.Lh - hidrociclone.Lh1;

    this.transform.localPosition = new Vector3(conexPos, coneyPos, coneZPos);

    float ypos = 2 + (hidrociclone.Lh) - (hidrociclone.Lh);

    var tempx = escalaRepresentacao * (float) Math.Round(hidrociclone.Dc, 3);
    var tempy = escalaRepresentacao * (float) Math.Round(hidrociclone.Dc, 3);
    var tempz = (escalaRepresentacao * (float) Math.Round(alturaConica, 3))/3;
    Debug.Log("tempz:" + tempz);

    var xpos = this.gameObject.transform.localPosition.x;
    if (hidrociclone.Lh1 == 0)
    {
        ypos = 2.95f - (hidrociclone.Lh * 3.3f);
    }
    else if (hidrociclone.Lh1 != 0)
    {
        ypos = 2.95f - (hidrociclone.Lh * 3.3f) - (hidrociclone.Lh1 * 3.3f);
    }
    var zpos = (this.gameObject.transform.localPosition.z);
    Debug.Log("ypos: " + ypos);

    this.gameObject.transform.localScale = new Vector3(tempx, tempy, tempz);
    this.gameObject.transform.localPosition = new Vector3(xpos, ypos, zpos);
}
}

```

04 – MainCameraButton, UiTexts, Sliders e GameController

Esse trecho traz scripts menores que foram utilizado para soluções menores, como armazenamento de variáveis e botões e textos na interface da ferramenta.

MainCameraButton

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MainCameraButton : MonoBehaviour
{
    public GameObject mainCamera;
    public GameObject secondCamera;
    public GameObject hidrocicloneObject;

    public bool isCameraOnHidrocyclone;

    public void Start()
    {
        mainCamera.SetActive(true);
        secondCamera.SetActive(false);
    }
}

```

```

public void updatePosicaoCamera()
{
    EquationsScript hidrociclone = hidrocicloneObject.GetComponent<EquationsScript>();
    GameObject camera = mainCamera.gameObject;

    Debug.Log("A função foi chamada");

    if (isCameraOnHidrocyclone)
    {
        isCameraOnHidrocyclone = false;

        mainCamera.SetActive(true);
        secondCamera.SetActive(false);

        Debug.Log("isCameraOnHidrocyclone == true");
    }
    else if (isCameraOnHidrocyclone == false)
    {
        isCameraOnHidrocyclone = true;

        secondCamera.SetActive(true);
        mainCamera.SetActive(false);
        this.gameObject.SetActive(true);
        Debug.Log("isCameraOnHidrocyclone == false");
    }
}
}

```

UiTexts

```

using System.Collections;
using System.Collections.Generic;
using System.Threading;
using UnityEngine;
using UnityEngine.UI;

public class UiTexts : MonoBehaviour
{
    [SerializeField]
    public GameObject firstResultsText;
    [SerializeField]
    public GameObject secondResultsText;
    [SerializeField]
    public GameObject thirdResultsText;
    [SerializeField]
    public GameObject RietemaVazio;

    public GameObject consultText;

    public string updatedText;

    public void resultsTextUpdate()
    {
        Text textA = firstResultsText.GetComponent<Text>();
        Text textB = secondResultsText.GetComponent<Text>();
        Text textC = thirdResultsText.GetComponent<Text>();
        EquationsScript scriptHidrociclone = RietemaVazio.GetComponent<EquationsScript>();

        textA.text = "ρ: " + scriptHidrociclone.rho + "Kg/m³\nnpS: " + scriptHidrociclone.rhoSolido + "Kg/m³\nnpμ: " +
            scriptHidrociclone.mi + "Pa.s\nndP: " + scriptHidrociclone.dP + "\n\nNº de Hiclones\n em Paralelo: " +
            scriptHidrociclone.numHidrociclonesPar + "\n\nnd50': " + scriptHidrociclone.dCorteR + "\n\nStk50Eu: " +
            scriptHidrociclone.stk50Eu + "\n\nEu: " + scriptHidrociclone.Eu + "\n\nRe: " + scriptHidrociclone.Re;
        textB.text = "Q: " + scriptHidrociclone.Q + "m³/s\nnQa: " + scriptHidrociclone.Qa + "m³/s\nnRl: " +
            scriptHidrociclone.Rl + "\n\nq: " + scriptHidrociclone.nSeparacaoTotal + "\n\nCva: " + scriptHidrociclone.Cva +
            "\n\nCvr: " + scriptHidrociclone.Cvr + "\n\nGeometria: " + scriptHidrociclone.geometriaUsada;
        textC.text = "Dc: " + scriptHidrociclone.Dc + "m\nnDu: " + scriptHidrociclone.Du + "m\nnDo: " +
            scriptHidrociclone.Do + "m\nnDi: " + scriptHidrociclone.Di + "m\n\nLh: " + scriptHidrociclone.Lh + "m\n\nLh1: " +
            scriptHidrociclone.Lh1 + "m\n\nq: " + scriptHidrociclone.nSeparacaoLinha;
    }

    public void ListSizeUpdateText(GameController canvasController)
    {
        Text actualConsultTexts = consultText.GetComponent<Text>();

        int count = 1;

        foreach (var item in canvasController.listOfDTA)
        {
            Debug.Log("Ponto: " + (item.d + ", " + item.y));
        }
    }
}

```

```

        actualConsultTexts.text = actualConsultTexts.text + "Ponto "+count+": (" + (item.d + ", " + item.y) + ")"
+ "\n";

        count++;
    }
}

public void ListSizeResetText(GameController canvasController)
{
    Text actualConsultTexts = consultText.GetComponent<Text>();

    actualConsultTexts.text = " ";
}
}

```

Sliders

```

using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Sliders : MonoBehaviour
{
    public GameObject hidrociclone;
    public GameObject secCamera;
    public GameObject canvasGameObj;
    public Canvas canvasSecCamera;

    public Slider YRotSlider;
    public Text YRotText;

    public Slider zoomSlider;
    public Text zoomText;

    void Start()
    {
        // Parametros para o slider de rotação em Y
        YRotSlider.maxValue = 360.0f;
        YRotSlider.minValue = 0.0f;
        YRotSlider.value = 180.0f;

        // Parametros para o slider de zoom da câmera
        zoomSlider.maxValue = 1.0f;
        zoomSlider.minValue = 0.0f;
        zoomSlider.value = 0.8f;
    }

    public void updateCameraZoom()
    {
        float tempX = secCamera.transform.position.x;
        float tempY = secCamera.transform.position.y;
        float ScaledSliderValue = 20.0f + (zoomSlider.value * 10);
        float sliderValue = zoomSlider.value;

        secCamera.transform.position = new Vector3(tempX, tempY, ScaledSliderValue);

        zoomText.text = "Camera Zoom: " + Math.Round(sliderValue, 2);
    }

    public void updateYRotation()
    {
        float tempX = hidrociclone.transform.rotation.x;
        float tempZ = hidrociclone.transform.rotation.y;
        float sliderValue = YRotSlider.value;

        hidrociclone.transform.eulerAngles = new Vector3 (tempX, sliderValue, tempZ);

        YRotText.text = "Rotação Y: " + Math.Round(sliderValue);
    }
}

```

GameController

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class GameController : MonoBehaviour
{
    [SerializeField]
    private InputField rho;
    [SerializeField]
    private InputField mi;
    [SerializeField]
    private InputField rhoSolido;
    [SerializeField]
    private InputField Cv;
    [SerializeField]
    private InputField Cva;
    [SerializeField]
    private InputField Cvr;
    [SerializeField]
    private InputField Q;
    [SerializeField]
    private InputField Qa;
    [SerializeField]
    private InputField Qr;
    [SerializeField]
    private InputField dP;
    [SerializeField]
    private InputField Rl;
    [SerializeField]
    private InputField alfa;
    [SerializeField]
    private InputField Dc;
    [SerializeField]
    private InputField d50;
    [SerializeField]
    private InputField d50R;
    [SerializeField]
    private InputField Du;
    [SerializeField]
    private InputField Lh;
    [SerializeField]
    private InputField Do;
    [SerializeField]
    private InputField Di;
    [SerializeField]
    private InputField Da;
    [SerializeField]
    private InputField e;
    [SerializeField]
    private InputField Lh1;
    [SerializeField]
    private InputField Theta;
    [SerializeField]
    private InputField Stk50Eu;
    [SerializeField]
    private InputField Re;
    [SerializeField]
    private InputField Eu;
    [SerializeField]
    private InputField Stk50R;
    [SerializeField]
    private InputField numCiclonesPar;
    [SerializeField]
    private InputField numCiclonesSerie;

    public float diametroDTA = 0f;
    public float eficienciaDTA = 0f;
    public float coefAngular;
    public float coefLinear;
    public float kRegressao;

    public class DTA
    {
        {
            public float d; // Valor do diâmetro de partícula que resulta na porcentagem de eficiência em y;
            public float y; // Eficiência relativa ao diâmetro de partícula d. => y(d)
        }
    }
}

```

```
public List<DTA> listOfDTA = new List<DTA>();
public List<float> listOfX = new List<float>();
public List<float> listOfY = new List<float>();

public void ControllerGetDTAAddButton()
{
    DTA currentDTA = new DTA();
    Debug.Log("currentDTA no começo da função: " + currentDTA.d + " e " + currentDTA.y);

    currentDTA.d = diametroDTA;
    currentDTA.y = eficienciaDTA;

    Debug.Log("O ponto (" + diametroDTA + "," + eficienciaDTA + ") foi adicionado.");

    // listOfDTA.Add(currentDTA);
    listOfDTA.Add(currentDTA);
}
}
```

ANEXO A – QUESTÕES PROPOSTAS POR RICARDO MEDRONHO

1. Uma bateria de hidrociclones de Bradley, de 6 cm de diâmetro, é utilizado para tratar $50 \text{ m}^3/\text{h}$ de uma suspensão aquosa, a 1% em volume de CaCO_3 . Calcular:

- a) o número de hidrociclones necessários.
- b) a eficiência total de separação;
- c) a concentração volumétrica do *underflow*;

DADOS: $\Delta P = 50 \text{ psi}$; $\rho_s = 2,8 \text{ g/cm}^3$; $D_u = 0,8 \text{ cm}$; $T = 20^\circ\text{C}$

Distribuição granulométrica do material particulado na alimentação:

d (μm)	5	10	15	20	30	40
y (%)	12	29	46	60	80	92

Resp.: (a) 18 hidrociclones; (b) 0,92; (c) 0,043.

2. Devido a alterações ocorridas em um processo utilizado por uma indústria, um hidrociclone Rietema de 9,0 cm de diâmetro terá suas variáveis operacionais modificadas para $\Delta P = 30 \text{ psi}$, $C_v = 9\%$, $D_u = 2,4 \text{ cm}$. Pede-se calcular qual será a vazão de operação e a eficiência total, sabendo-se que $\rho_s = 2,5 \text{ g/cm}^3$ e que o fluido de operação é a água com as seguintes propriedades: $\rho = 1,0 \text{ g/cm}^3$ e $\mu = 1,0 \text{ cp}$.

Distribuição granulométrica do material particulado na alimentação:

d (μm)	2	4	6	8	10	14	20
y (%)	35,5	59,0	72,0	80,5	86,0	92,0	96,0

Resp.: $Q = 0,0036 \text{ m}^3/\text{s}$ e $E_T = 0,36$.

3. Deseja-se substituir o hidrociclone de Rietema do problema acima por uma bateria de hidrociclones de Rietema de 1,0 cm de diâmetro com $D_u = 0,27$ cm. Calcular o número de ciclones em paralelo a ser utilizado, se deseja-se manter inalterada a vazão de operação. Calcular, também, a eficiência total obtida.

Resp.: 74 hidrociclones em paralelo e $E_T = 0,59$.



Datas e horários baseados no fuso horário (GMT -3:00) em Brasília, Brasil
Sincronizado com o NTP.br e Observatório Nacional (ON)
Certificado de assinatura gerado em 15/12/2020 às 00:01:47 (GMT -3:00)

Rafael Henrique - TCC CORRIGIDO.pdf

 ID única do documento: #eab8c9f9-9099-42b5-a83f-d684178c4774

Hash do documento original (SHA256): 5d98596d012b8e5c57681dd8c59e881057d2c27655cd4cb79b5a2faafe07f99f

Este Log é exclusivo ao documento número #eab8c9f9-9099-42b5-a83f-d684178c4774 e deve ser considerado parte do mesmo, com os efeitos prescritos nos Termos de Uso.

Assinaturas (4)

- ✓ Alex Vazzoler (Participante)
Assinou em 15/12/2020 às 10:49:03 (GMT -3:00)
- ✓ Thiago Sousa de Oliveira (Participante)
Assinou em 15/12/2020 às 08:31:14 (GMT -3:00)
- ✓ Tanise Mori Flores (Participante)
Assinou em 15/12/2020 às 10:08:13 (GMT -3:00)
- ✓ Elene de Souza Freitas (Participante)
Assinou em 15/12/2020 às 12:49:28 (GMT -3:00)

Histórico completo

Data e hora

15/12/2020 às 00:01:47
(GMT -3:00)

15/12/2020 às 08:31:14
(GMT -3:00)

Evento

Rafael Henrique Elias Dias solicitou as assinaturas.

Thiago Sousa de Oliveira (Autenticação: e-mail thiago.oliveira@aveva.com; IP: 201.17.127.216) assinou. Autenticidade deste documento poderá ser verificada em [\[\[LINK\]\]](#). Assinatura com validade jurídica conforme MP 2.200-2/01, Art. 10º, §2.



Data e hora

15/12/2020 às 10:08:13
(GMT -3:00)

Evento

Tanise Mori Flores (Autenticação: e-mail tmflores@cetiqt.senai.br; IP: 201.17.125.214) assinou. Autenticidade deste documento poderá ser verificada em [\[\[LINK\]\]](#). Assinatura com validade jurídica conforme MP 2.200-2/01, Art. 10o, §2.

15/12/2020 às 10:49:03
(GMT -3:00)

Alex Vazzoler (Autenticação: e-mail avazzoler@cetiqt.senai.br; IP: 177.54.100.109) assinou. Autenticidade deste documento poderá ser verificada em [\[\[LINK\]\]](#). Assinatura com validade jurídica conforme MP 2.200-2/01, Art. 10o, §2.

15/12/2020 às 12:49:28
(GMT -3:00)

Elene de Souza Freitas (Autenticação: e-mail esfreitas@cetiqt.senai.br; IP: 179.127.193.118) assinou. Autenticidade deste documento poderá ser verificada em [\[\[LINK\]\]](#). Assinatura com validade jurídica conforme MP 2.200-2/01, Art. 10o, §2.