

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA QUÍMICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

JOSIVAN PEDRO DA SILVA

ESTUDO DA MINIMIZAÇÃO DE SABÃO NA PRODUÇÃO DE BIODIESEL POR VIA
ETANÓLICA E AUTOMAÇÃO DE BAIXO CUSTO EM UMA COLUNA DE
DESTILAÇÃO REATIVA

Recife
2018

JOSIVAN PEDRO DA SILVA

ESTUDO DA MINIMIZAÇÃO DE SABÃO NA PRODUÇÃO DE BIODIESEL POR VIA
ETANÓLICA E AUTOMAÇÃO DE BAIXO CUSTO EM UMA COLUNA DE
DESTILAÇÃO REATIVA

Tese submetida ao curso de Pós-Graduação em
Engenharia Química da Universidade Federal de
Pernambuco, como parte dos requisitos necessários
à obtenção do grau de Doutor em Engenharia
Química.

Área de concentração: Processos Químicos
Industriais

Orientador: Prof. Dr. José Marcos Francisco da Silva.

Coorientadores: Prof. Dr. José Geraldo Pacheco Filho

Prof. Dr. Thibério Souza Pinho Costa

Recife
2018

Catalogação na fonte
Bibliotecária Maria Luiza de Moura Ferreira, CRB-4 / 1469

S586e Silva, Josivan Pedro.

Estudo da minimização de sabão na produção de biodiesel por via etanólica e automação de baixo custo em uma coluna de destilação reativa/ Josivan Pedro Silva - 2018.

174 folhas, il., tabs.

Orientador: Prof. Dr. José Marcos Francisco da Silva.

Coorientador: Prof. Dr. José Geraldo Pacheco Filho.

Prof. Dr. Thibério Souza Pinho Costa.

Tese (Doutorado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Química, 2018.

Inclui Referências, Apêndices e Anexo.

1. Engenharia Química.
 2. Biodiesel.
 3. Coluna de destilação reativa.
 4. Saponificação.
 5. Automação.
- I. Silva, José Marcos Francisco da (Orientador).
II. Pacheco Filho, José Geraldo (Coorientador). III. Costa, Thibério Souza Pinho (Coorientador). IV. Título.

UFPE

660.2 CDD (22. ed.)

BCTG/2018-425

JOSIVAN PEDRO DA SILVA

**ESTUDO DA MINIMIZAÇÃO DE SABÃO NA PRODUÇÃO DE
BIODIESEL POR VIA ETANÓLICA E AUTOMAÇÃO DE BAIXO
CUSTO EM UMA COLUNA DE DESTILAÇÃO REATIVA**

Tese submetida ao curso de Pós-Graduação em Engenharia Química da Universidade Federal de Pernambuco, como parte dos requisitos necessários à obtenção do grau de Doutor em Engenharia Química.

Aprovada em 2 de Março de 2018

Prof. Dr. José Marcos Francisco da Silva (Orientador)
Universidade Federal de Pernambuco

Prof. Dr. Luiz Stragevitch (Examinador Interno)
Universidade Federal de Pernambuco

Prof. Dr. Mohand Benachour (Examinador Interno)
Universidade Federal de Pernambuco

Dr. James Correia de Melo (Examinador Externo)
Centro de Tecnologias Estratégicas do Nordeste

Prof. Dra. Ivoneide de Carvalho Lopes Barros (Examinador Externo)
Universidade Federal Rural de Pernambuco

Prof. Dr. Paulo André da Silva Gonçalves (Examinador Externo)
Universidade Federal de Pernambuco

AGRADECIMENTOS

Aos meus pais, Laudicéia Maria dos Santos e José Pedro da Silva, por todo incentivo, companhia, paciência, confiança em todos os momentos. E a todos em minha família.

A minha esposa Ana Carolina pelo carinho, e por se fazer presente em minha vida.

Aos meus orientadores José Marcos e José Geraldo pela orientação, amizade e dedicação na realização deste trabalho, na qual sempre se mostraram prestativos em todas as oportunidades. Agradeço especialmente ao meu co-orientador Thibério Souza por estar apoiando lado a lado nos momentos do processo de pesquisa, desenvolvimento e construção da coluna de destilação reativa.

A todos que fazem parte do Laboratório de Simulação e Dinâmica de Processos Químicos – LSDPQ, especialmente a Ronaldy Miller, por toda a ajuda prestada e companheirismo na execução dos experimentos. Também ao próprio LSDPQ pelo apoio com relação a estrutura necessária à pesquisa.

Ao LAC (Laboratório de Combustíveis) por todo auxílio nas análises e caracterizações necessárias, agradeço a Rafaela Sales pela forma prestativa que me ajudou na comunicação direta com o LAC, e pelas análises feitas por ela.

Agradeço ao professor Alexandre Ricardo Pereira Schuler por todo o auxílio, nas análises cromatográficas, e na elucidação de problemas pertinentes a essa tese.

Ao Departamento de Engenharia Química e seus professores, especialmente ao professor Luís Stragevitch, pelo auxílio prestado pelo LAC, a professora Fernanda Pimentel pela parceria na pesquisa, A professora Otidene Rocha pela participação em diversos artigos.

A CAPES pelo auxílio financeiro. E a todos que contribuíram direta ou indiretamente para realização deste trabalho.

RESUMO

O biodiesel é a principal alternativa para o óleo diesel, pois é um combustível limpo que causa menos impactos ambientais que o diesel de petróleo. O processo de produção de biodiesel mais utilizado é em batelada, entretanto os processos contínuos têm se mostrado promissores. A formação de subprodutos indesejados pode ser um fator crucial na viabilidade do processo quando o álcool utilizado no processo é o etanol. Os carboxilatos (sabões) devem ser minimizados na produção, pois dificultam os processos de separação posteriores, assim como reduz a pureza do biodiesel final. Este trabalho tem como intuito estudar as condições de operação de uma coluna de destilação reativa de pequeno porte, para minimizar a formação de sabão e maximizar a produção de biodiesel, partindo de resultados experimentais. Para isso, foram implementadas melhorias na coluna, com relação a automação e supervisão tridimensional, utilizando-se materiais de baixo custo para o desenvolvimento deste trabalho. Foi possível concluir que as condições de reação via etanólica é favorecida com teores de catalisador acima de 1% mássico com relação ao óleo. E que a condição operacional que favorece a saponificação é o aumento da carga térmica do refervedor. Ainda foi possível concluir que o uso de interfaces homem máquina mais intuitivas tornam o processo produtivo mais eficiente.

Palavras-chave: Biodiesel. Coluna de destilação reativa. Saponificação. Automação.

ABSTRACT

Biodiesel is the main alternative for diesel oil because it is a clean fuel that causes less environmental impacts than petroleum diesel. The most used biodiesel production process is in batch, however the continuous processes have shown to be promising. The formation of unwanted by-products may be a crucial factor in the viability of the process when the alcohol used in the process is ethanol. Carboxylates (soaps) should be minimized in production as they hamper subsequent separation processes, as well as reduce the purity of the final biodiesel. This work aims to study the operating conditions of a small reactive distillation column to minimize the formation of soap and to maximize biodiesel production, starting from experimental results. For this purpose, improvements were made to the column in relation to automation and three-dimensional supervision, using low-cost materials for the development of this work. It was possible to conclude that the reaction conditions via ethanol are favored with catalyst contents above 1% of the mass in relation to the oil. And that the operational condition that favors the saponification is the increase of the thermal load of the reboiler. It was still possible to conclude that the use of more intuitive man-machine interfaces makes the productive process more efficient.

Keywords: Biodiesel. Reactive distillation column. Saponification. Automation.

LISTA DE FIGURAS

Figura 1 -	O processo de produção do acetato de metila pela Eastman Chemical.....	29
Figura 2 -	Diagramas ternários de equilíbrio químico, de acordo com Venimadhavan et al. (1994).....	31
Figura 3 -	Mapa de curvas residuais de destilação para um sistema com dois azeótropos binários, de acordo com Venimadhavan et al. (1994).....	32
Figura 4 -	Configuração geral de um estágio J na coluna de destilação reativa. (L=vazão líquida, F=vazão de alimentação, V=vazão de vapor, Q=calor cedido, Z,R=são retiradas laterais).....	37
Figura 5 -	Configuração inicial da coluna de destilação reativa.....	45
Figura 6 -	Configuração final da coluna de destilação reativa, e representação 3D da coluna no supervisório.....	47
Figura 7 -	Sistema para teste de transesterificação em batelada.....	52
Figura 8 -	Cromatograma resultante da análise do biodiesel etílico.....	59
Figura 9 -	Arquitetura desenvolvida.....	63
Figura 10 -	Esquema elétrico desenvolvido para a coluna.....	64
Figura 11 -	a) Esquema simulado no LTSpice [©] b) pico de tensão de 630V ao desligar uma solenoide c) pico de tensão de 43V após a adição do Snubber.....	66
Figura 12 -	Montagem final do sistema de proteção contra surtos de corrente e tensão, snubber.....	68
Figura 13 -	Sistema para controle de bombas e motores utilizando o transistor TIP120.....	68
Figura 14 -	Arduino IDE.....	69
Figura 15 -	Interface de desenvolvimento do Unity 3D [©]	71
Figura 16 -	Interface do supervisório da coluna de destilação reativa.....	72
Figura 17 -	Funções do supervisório da coluna de destilação reativa, da esquerda para direita, de cima para baixo tem-se: Sensores, câmeras, console e rede.....	74

Figura 18 -	Visão geral das telas do aplicativo desenvolvido.....	75
Figura 19 -	Gráfico de Pareto para a conversão do óleo.....	82
Figura 20 -	Gráfico de resposta da conversão para carga térmica e refluxo.....	83
Figura 21 -	Gráfico de resposta da conversão para razão molar e refluxo.....	84
Figura 22 -	Gráfico de resposta da conversão para Razão molar e carga térmica.....	85
Figura 23 -	Valores observados em função dos valores previstos.....	86
Figura 24 -	Gráfico de Pareto para a concentração de catalisador residual.....	87
Figura 25 -	Gráfico de resposta da concentração de catalisador residual para razão molar e refluxo.....	88
Figura 26 -	Gráfico de resposta da concentração de catalisador residual para carga térmica e refluxo.....	89
Figura 27 -	Gráfico de resposta da concentração de catalisador residual para carga térmica e da razão molar.....	90
Figura 28 -	Valores observados em função dos valores previstos.....	91
Figura 29 -	Gráfico de Pareto para a concentração de sabão residual.....	92
Figura 30 -	Gráfico de resposta da concentração de sabão para carga térmica e refluxo.....	93
Figura 31 -	Gráfico de resposta da concentração de sabão para o refluxo e a razão molar.....	94
Figura 32 -	Gráfico de resposta da concentração de sabão para a razão molar e a carga térmica.....	95
Figura 33 -	Valores observados em função dos valores previstos.....	96

LISTA DE TABELAS

Tabela 1 -	Vantagens e desvantagens na aplicação de processos reativos de separação.....	22
Tabela 2 -	Influência dos valores de Da e K segundo Venimadhavan et al. (1994).....	35
Tabela 3 -	Características de refervedores e condensadores, adaptado de Machado (2009).....	39
Tabela 4 -	Composição mássica do óleo de algodão.....	48
Tabela 5 -	Características físicas do óleo de algodão segundo Ramadhas et al. (2005).....	49
Tabela 6 -	Massa molar dos ácidos que compõem o óleo de algodão.....	49
Tabela 7 -	Resultados obtidos.....	61
Tabela 8 -	Condições de operação no teste inicial da coluna de destilação reativa.....	76
Tabela 9 -	Resultados experimentais obtidos.....	77
Tabela 10 -	Condições de operação experimental da coluna de destilação reativa.....	79
Tabela 11 -	Parâmetros variados para o estudo da produção de sabão em uma coluna de destilação reativa.....	80
Tabela 12 -	Planejamento fatorial e resultados obtidos.....	80

SUMÁRIO

1	INTRODUÇÃO.....	12
2	REVISÃO BIBLIOGRÁFICA.....	14
2.1	Processos de produção do biodiesel.....	14
2.2	Saponificação.....	16
2.3	Automação.....	17
3	FUNDAMENTAÇÃO TEÓRICA.....	21
3.1	Processos reativos de separação.....	21
3.2	Destilação reativa.....	26
3.2.1	<i>Aplicações da destilação reativa na industria.....</i>	28
3.2.2	<i>Teoria sobre a destilação reativa.....</i>	30
3.2.3	<i>Modelagem e projeto.....</i>	34
3.2.4	<i>Modelagem matemática.....</i>	36
3.3	Reação de transesterificação.....	39
3.4	Reação de saponificação.....	40
3.5	Planejamento fatorial.....	42
4	MATERIAIS E MÉTODOS.....	45
4.1	Coluna de destilação e automação.....	45
4.2	Síntese do biodiesel.....	48
4.2.1	<i>Testes em batelada.....</i>	48
4.2.2	<i>Ensaio de transesterificação do óleo de algodão na coluna de DR.....</i>	53
4.2.3	<i>Planejamento fatorial para análise da saponificação na coluna de DR.....</i>	54
4.3	Métodos de análise dos produtos.....	54
5	RESULTADOS E DISCUSSÃO.....	57
5.1	Testes em batelada.....	57
5.2	Custos e automação.....	62
5.3	Desenvolvimento do supervisório.....	70
5.4	Testes na coluna de destilação reativa.....	75
5.5	Planejamento fatorial.....	79
5.5.1	<i>Resultados do planejamento com relação a conversão.....</i>	81
5.5.2	<i>Resultados do planejamento com relação ao catalisador residual.....</i>	86
5.5.3	<i>Resultados do planejamento com relação ao sabão.....</i>	91

6	CONCLUSÕES E PERSPECTIVAS.....	97
6.1	Conclusões.....	97
6.2	Sugestões para trabalhos futuros.....	99
	REFERÊNCIAS.....	101
	APÊNDICE A - SCRIPT NO ARDUINO.....	109
	APÊNDICE B - SCRIPT PRINCIPAL DO SUPERVISÓRIO.....	113
	APÊNDICE C - CÁLCULO DA ACIDEZ DE ÓLEOS VEGETAIS.....	159
	APÊNDICE D - SCRIPT EM MATLAB PARA CÁLCULO DA QUANTIDADE DE REAGENTES E CATALISADOR EM UMA TRANSESTERIFICAÇÃO.....	160
	APÊNDICE E - PROCEDIMENTO OPERACIONAL DE UTILIZAÇÃO DA COLUNA DE DESTILAÇÃO REATIVA.....	162
	APÊNDICE F - FAZENDO A ANÁLISE ESTATÍSTICA DE UM PLANEJAMENTO FATORIAL NO STATISTICA 12.0.....	164
	APÊNDICE G - DETALHAMENTO DOS CÁLCULOS ESTEQUIOMÉTRICOS.....	165
	APÊNDICE H - AVALIAÇÃO DO MODELO DE GENERALIZAÇÃO DO TEOR DE ÉSTER.....	168
	APÊNDICE I - AVALIAÇÃO DO MODELO DE GENERALIZAÇÃO DA CONCENTRAÇÃO DE CATALISADOR RESIDUAL.....	170
	APÊNDICE J - AVALIAÇÃO DO MODELO DE GENERALIZAÇÃO DA CONCENTRAÇÃO DE SABÃO.....	172
	ANEXO A TABELA DE REAGENTES UTILIZADOS E SUAS INFORMAÇÕES.....	174

1 INTRODUÇÃO

Nas últimas décadas a produção de fontes energéticas renováveis tem sido destaque mundial, pois é um ponto fundamental para o desenvolvimento sustentável, e entre as diversas fontes o biodiesel tem se destacado. O biodiesel é um combustível alternativo biodegradável e renovável com características semelhantes ao diesel de petróleo (BOWMAN et al., 2006; BALAT; BALAT, 2008). Normalmente, ele é produzido a partir de fontes biológicas, como óleos vegetais, gordura animal ou mesmo resíduos de cozinha óleos provenientes da indústria alimentar (ENCINAR et al., 2005; KULKARNI; DALAI, 2006).

Segundo Sheehan et al. (1998) e Taher e Al-Zuhair (2016) existem diversas vantagens do biodiesel sobre o diesel convencional: Reduz a dependência de petróleo. Não é limitado assim como os combustíveis fosseis. Reduz a emissão de poluentes perigosos para a saúde. Biodegradável e de baixa toxicidade. Beneficia a economia interna.

Ao contrário do diesel de petróleo, que é uma mistura de hidrocarbonetos, o biodiesel consiste de ésteres metílicos ou etílicos de ácidos graxos. Ésteres graxos são atualmente produzidos por diversos métodos, seguido de passos de purificação e neutralização do produto (HANNA et al., 2005; MEHER et al., 2006). Atualmente, as tecnologias de produção mais comuns empregam catalisadores homogêneos, em batelada, onde ambas as etapas de reação e separação podem criar gargalos. As principais matérias primas são óleos e o metanol, porém a disponibilidade de etanol é grande em nosso país, sua utilização pode ter caráter estratégico.

De acordo com Zhou et al. (2007), Souza et al. (2014) e Hernández, Hernández e Petriciolet (2015), a destilação reativa (DR) tem se mostrado mais favorável em diversos aspectos que o processo convencional de produção, em batelada, pois é um processo química que reúne em uma única unidade as reações químicas e a separação. Segundo He (2006), as vantagens são significativas da DR sobre o processo sequencial convencional, como por exemplo, a alta taxa de conversão química, alta produtividade, instalações de dimensões reduzidas e menor custo de operação. O desempenho de uma coluna de destilação reativa é influenciado por diversos parâmetros, tais como razão de refluxo, relação álcool/óleo na alimentação, temperatura, tamanho das zonas de reação e separação entre outros.

Um dos inconvenientes do processo de produção de biodiesel, por catálise alcalina, é a produção indesejada de sabões. Devido a essas reações concorrentes o rendimento de uma reação de transesterificação pode ser alterado, proveniente da redução da disponibilidade de íons OH⁻ no meio reacional segundo Eze et al. (2014). O sabão produzido leva também a dificuldades na purificação posterior, pois o biodiesel resultante apresenta características gelatinosas, e até mesmo pode solidificar devido ao excesso de sabão. Desta forma, fica clara a necessidade de minimizar essa reação indesejada na destilação reativa.

Processos complexos como a destilação reativa necessitam de uma instrumentação de automação mais precisa e robusta, pois neste caso, deve-se atender as condições de separação e de reação simultaneamente (KISS; BILDEA, 2012; PISARENKO et al., 2011). Para isso é possível utilizar o conceito de **Indústria 4.0** leva ao uso e desenvolvimento de processos mais eficientes e inteligentes, Além da utilização de sistemas ciber-físicos para monitoramento de processos reais, onde se cria uma cópia virtual do processo físico e se controla por aplicativos via Internet, empregando também o conceito de Internet das coisas (**IOT**). Onde é permitida a conexão via Internet para o acompanhamento de dados de sistemas físicos ou virtuais (MINERVA et al., 2015).

O presente trabalho tem como objetivo geral estudar experimentalmente os parâmetros operacionais da destilação reativa que minimizam a produção de sabão e maximizam a conversão do óleo em biodiesel. Desenvolvendo, como ferramenta auxiliar, sistemas de automação e supervisão de baixo custo em uma coluna de DR. Tem-se como objetivos específicos:

- Executar a automação de baixo custo de uma coluna de destilação reativa, utilizando conceitos da Industria 4.0 e Internet das coisas, além de desenvolver um supervisório 3D, para acompanhamento a distância do processo de produção de biodiesel;
- Realizar um planejamento fatorial para encontrar condições que maximizem conversão do óleo, maximizem a quantidade de catalisador no meio reacional, e minimizem a concentração de sabão no biodiesel produzido, e com isso compreender a influência dos parâmetros operacionais sobre a saponificação. Assim como estudar as características de produção do biodiesel por via etílica, métodos de análise, purificação, secagem dos reagentes e produtos e parâmetros de reação.

2 REVISÃO BIBLIOGRÁFICA

O desenvolvimento de pesquisas sobre os processos de biodiesel tem avançado significativamente nos últimos anos, uma revisão bibliográfica foi feita com a intenção de entender melhor o estado da arte.

2.1 Processos de produção do biodiesel

O biodiesel pode ser produzido por diversos processos, reações e matérias primas. A questão principal é produzi-lo com o menor custo possível, de forma a se obter este biocombustível com a pureza elevada, e dentro das especificações da legislação vigente.

Processos descontínuos de transesterificação (em batelada) com catálise alcalina são realizados utilizando o hidróxido de sódio ou de potássio, conseguem uma atividade catalítica alta, em condições amenas de reação (40-80°C), segundo Leung et al. (2010). Apresenta como desvantagens a necessidade de um óleo com baixo teor de ácidos graxos livres, necessita ocorrer em condições isentas de água, pode levar à saponificação e à formação de emulsões, e gera uma grande quantidade de água poluída durante a purificação (MEHER et al., 2006; LEUNG et al. 2010). Com relação ao processo em larga escala, a produtividade dos equipamentos é baixa e os custos de operação são elevados (HANNA et al., 2005; LOTERO et al., 2005).

Como exposto por Silva et al. (2011), que estudaram a produção de biodiesel utilizando etanol, o processo pode ser otimizado pela aplicação de um planejamento fatorial 2⁴ e metodologia de superfície de resposta. Os efeitos combinados de temperatura, catalisador, a concentração, o tempo de reação e a proporção molar de álcool em relação ao óleo foram investigados e otimizados. As condições ideais para a produção de ésteres etílicos foram as seguintes: temperatura de 56,7°C, tempo de reação em 80 minutos, razão molar álcool/óleo a 9:1 e concentração de catalisador de 1,3% mássico em relação ao óleo.

Como descrito por Reyero et al. (2015) a condição ideal para a produção de biodiesel é utilizando razão molar etanol/óleo entre 12 e 24 com concentração de catalisador de 0,3% em massa, com o maior tempo de reação possível.

Processos descontínuos de transesterificação (em batelada) com catálise ácida são realizados utilizando ácido sulfúrico (H_2SO_4), tem como pontos positivos as reações de

esterificação e a transesterificação ocorrendo simultaneamente, e impedem a formação de sabão, segundo Leung et al. (2010). Como pontos negativos, o processo gasta mais água de purificação, água que necessita de um tratamento mais difícil, a reação necessita de maiores temperaturas que o processo catalisado por bases, necessita de maior tempo de reação, e a atividade catalítica é pequena (DIZGE et al. 2009). Com relação à produção em larga escala ocorre uma maior corrosão dos equipamentos, a produtividade é baixa e os custos de operação são elevados (LEUNG et al. 2010)

Os processos contínuos combinam a purificação e transesterificação, são chamados de processos reativos de separação, são baseados em esterificação ou reações de transesterificação realizadas na presença de catalisadores líquidos ou sólidos em unidades integradas, tais como: destilação/absorção/extração ou reatores de membrana (KISS; BILDEA, 2012), permitindo maior produtividade. No entanto, a maioria destes processos continuam enfrentando as desvantagens da utilização de catalisadores homogêneos embora catalisadores sólidos tenham sido utilizados nas duas últimas décadas (DALE, 2003; KISS et al., 2006a.). Vários processos de destilação reativa-foram relatados (HE et al, 2006; KISS et al., 2006b.).

Processos supercríticos foram desenvolvidos para resolver o problema da miscibilidade óleo-álcool que dificulta a cinética da transesterificação, bem como para tomar vantagem de não utilizar um catalisador. As condições de operação ainda são bastante severas ($T > 240^\circ C$, $p > 80$ bar) e, portanto, requerem equipamento especial e custo elevado (MARCHETTI;ERRAZU, 2008).

Processos enzimáticos de produção de biodiesel têm baixos requisitos energéticos, como a reação é realizada em temperaturas de $50\text{--}55^\circ C$ e à pressão ambiente. No entanto, devido aos rendimentos mais baixos e o tempo longo de reação, os processos enzimáticos não podem competir ainda com outros processos industriais (TAN et al., 2006; TAHER; AL-ZUHAIR, 2016; GERPEN, 2005).

Processos em dois passos são muito mais simples como os triacilglicerois são hidrolisados primeiro a ácidos graxos que são então esterificados num segundo passo para produzir ésteres graxos (KUSDIANA; SAKA, 2004; MINAMI; SAKA, 2006). Estes processos são agora muito atraentes e ganham mercado devido a vantagens óbvias: glicerol de elevada pureza é obtido como subproduto da etapa de hidrólise, e o passo de esterificação

pode ser realizada utilizando catalisadores sólidos numa instalação integrada reativa-separativa (KISS et al. 2006b). O uso de catalisadores sólidos evita as etapas de neutralização e lavagem. Além disso, quando as matérias-primas consistem principalmente de AGL, apenas a etapa de esterificação é necessária.

A literatura recente é abundante em estudos sobre os processos integrados, como a destilação reativa (BOON-ANUWAT et al., 2015; PODDAR et al. 2017; KISS; BILDEA, 2012; XIAO et al., 2013; KISS et al., 2008; KISS et al., 2006b) e absorção reativa (RAY; RAY, 2016; KISS; BILDEA, 2012; GHAEMI et al., 2009; NOERES et al., 2003). No entanto, a maioria dos estudos relatados em processos de separação reativa para a produção de biodiesel são baseadas unicamente em destilação reativa convencional (HE et al., 2006; KISS et al., 2006b; KISS et al., 2008), ou alternativas como destilação reativa (RD) e dupla destilação reativa (NGUYEN; DEMIREL, 2011; DIMIAN et al., 2009).

Como exposto por Silva et al. (2010) que apresentou um processo eficiente usando colunas de destilação reativa aplicadas à produção de biodiesel a partir de óleo de soja e bioetanol. As principais variáveis de processo que afetam a produção de biodiesel são concentração de catalisador, temperatura de reação, nível de agitação, relação molar de etanol /óleo, tempo de reação, e tipo de matéria-prima. Um planejamento experimental foi usado para otimizar a concentração do catalisador (0,5-1,5% mássico em relação ao óleo) e relação molar de etanol/óleo (3:1 a 9:1) sob uma taxa de refluxo de 83 mL/min e 6 minutos de tempo de reação.

2.2 Saponificação

O estudo da formação de carboxilatos (sabões) vem sendo feito à quase um século, pois eles surgem naturalmente em reações de transesterificação, reação a qual é utilizada em diversos processos há muitas décadas (SMITH; LEVENSON, 1940). Na literatura é possível observar que o teor de sabão interfere em diversas características do biodiesel, como por exemplo, a densidade e a viscosidade (PHANKOSOL; SUDAPRASERT; KRISNANGKURA, 2015).

Segundo (LEUNG et al., 2010) os óleos vegetais e gorduras podem conter pequenas quantidades de água e ácidos graxos livres (AGL). Para uma transesterificação catalisada por uma base, o catalisador alcalino que é usado reagirá com o AGL para formar sabão, por isso o

óleo deve ser pré-tratado para que haja o mínimo possível de AGL. A Equação 11 mostra a reação de saponificação do catalisador (hidróxido de sódio) e o AGL, formando sabão e água. Assim como a água, originada quer dos óleos e gorduras ou formada durante a reação de saponificação, retarda a reação de transesterificação através da reação de hidrólise. Pode hidrolisar os triacilglicerois em diacilglicerois e formar mais AGL. A reação de hidrólise típica é mostrada na Equação 14. O que está de acordo com o exposto por Eze et al. (2015). De acordo com estes dois autores a saponificação é fortemente favorecida por altas temperaturas, o que pode ser um ponto chave para o controle da saponificação. Especificamente segundo Eze et al. (2015) a reação de saponificação é ainda mais intensa quando se produz biodiesel via etanólica, sendo de 5 a 7 vezes mais rápida. Como citado por Leung et al. (2010) deve haver cuidados também na lavagem do biodiesel formado para evitar a saponificação.

Durante toda revisão bibliográfica e até o atual momento não foram verificados artigos que estudam a formação de sabão na produção de biodiesel em processos contínuos, mais especificamente na destilação reativa. Os artigos citados anteriormente estudam apenas o processo em batelada. Em uma coluna de destilação reativa há uma série de parâmetros operacionais que podem ser estudados, os quais não foram explorados na literatura. Isto reforça a **originalidade** e o **ineditismo** deste trabalho, **vindo para suprir uma lacuna nos trabalhos científicos da atualidade**.

2.3 Automação

Os processos industriais têm evoluído significativamente desde a primeira revolução industrial no século XIX, no transcorrer do tempo diversas formas de organização do trabalho foram desenvolvidas, como o Taylorismo, Fordismo e o Toyotismo, todas estas sistemáticas buscam o aumento da eficiência e da produtividade nos processos industriais segundo (NOBLE, 2011). Com o desenvolvimento da informática nos anos de 1970, foi possível a obtenção de maiores rendimentos e um maior controle sobre o produto acabado. Os processos produtivos têm se tornado mais complexos e maiores, com o passar do tempo, e o uso de computadores tem se tornado cada vez mais comum, devido a reduções sucessivas no custo destes equipamentos, assim como também o ganho no controle e automação do processo produtivo.

Os computadores são utilizados principalmente na análise de informações da produção para assim atuar sobre qualquer desvio dos padrões pré-estabelecidos, na indústria química a sistemática é a mesma, este tipo de controle é comumente chamado de “controlador de realimentação”, apesar da ideia deste tipo de controle existir desde a Grécia antiga (250 a.c.) para controle do fluxo de água. O controle de processos moderno foi desenvolvido por James Watt, em 1788, para o desenvolvimento de geradores a vapor. Porém nesta época todos os controles eram baseados em sistemas mecânicos e não eletrônicos como são atualmente. Os tipos mais comuns de controladores, PID (proporcional, integral e derivativo), foram comercializados a partir dos anos 1930, e ganharam grande aceitação nos anos de 1940. Apenas após 1950 que os componentes eletrônicos para controle foram inseridos no mercado, como exposto por Franchi (2008) e Seborg (2016). Os primeiros computadores para controle de processos industriais apareceram entre os anos de 1950 e 1960, a partir de 1980 o uso de componentes digitais tem se tornado comum, o que levou a grande impacto positivo na automação de processos.

O monitoramento do processo industrial se dá por meio de um sistema de supervisão, que nada mais é do que um tipo de software que permite monitorar e controlar partes ou todo um processo industrial. Atualmente, além de poder visualizar na tela tudo o que está acontecendo no processo industrial, também é tarefa do sistema supervisório controlá-lo, modelo o qual se denomina SCADA (*Supervisory Control And Data Aquisition*), todas as variáveis lidas são processadas por uma unidade de processamento qualquer e através deste processamento se ter o retorno do que precisa ser alterado no processo, a partir deste ponto será utilizado um sistema de acionamento que enviará a informação para o atuador que estará na planta. A união destas três etapas caracteriza um sistema supervisório segundo Boyer (2009).

O primeiro registro da utilização de um sistema supervisório data de 1981, estes supervisórios eram dotados de uma tecnologia primitiva visto que os computadores da época não tinham muitos recursos e ocupavam enormes espaços. Alguns supervisórios utilizavam microcomputadores, mas por se tratar de nova tecnologia, seu custo era extremamente alto, por este motivo somente plantas de energia e petróleo utilizavam-se destas tecnologias, pois o retorno financeiro viabilizava o alto custo (BOYER, 2009).

Com a crescente demanda dos processos industriais e as novas práticas de gestão, que com o passar do tempo exigiam informações do processo em intervalos de tempo cada vez

mais curtos, aliado a isto a grande evolução computacional com a descoberta dos transistores, começa-se a desenvolver sistemas supervisórios cada vez mais robustos, até que em meados dos anos de 1980 surgem os sistemas SCADA, os quais não só supervisionam o processo, mas também o controlam. Surgia deste ponto um novo mercado emergente, o mercado de sistemas supervisórios. Já no início dos anos 1990 já havia mais de 120 fornecedores que disputavam tal mercado, mas com tanta concorrência, o principal problema era o sistema operacional, a plataforma, no qual estes supervisórios iriam rodar. Foram utilizados inúmeros sistemas operacionais como MS-DOS, OS/2, Qnx, Windows, Unix e Vms, até que empresas internacionais, como a US Data, lançaram seus novos sistemas supervisórios multiplataforma, ou seja, poderiam rodar em qualquer sistema operacional. Isto trouxe uma enorme vantagem para fins de portabilidade e conectividade, pois a partir deste ponto seu sistema poderia se conectar com outro em qualquer outra plataforma. Até que a Microsoft entrou de vez nesta briga e lançou seu novo sistema operacional Windows NT, que contava com um sistema multitarefa, que é essencial para o bom funcionamento de um sistema supervisório, e assim conseguiu superar de vez as ineficiências do Windows 3.11 e conquistou o mercado de supervisórios. Reconhecendo a nova demanda do mercado, as empresas desenvolvedoras de supervisórios migraram seus sistemas para a plataforma Windows, o que resultou numa convergência de padrões e módulos a qual consolidou de vez o mercado de supervisórios. Algumas empresas se destacaram, como a Wonderware, fabricante do InTouch, e duas brasileiras, a Elipse e a Indusoft segundo (BOYER, 2009).

Visando atender uma maior porção do mercado, as empresas fabricantes de hardware de controle entraram de vez na briga pelo mercado de supervisórios, pois ao invés de vender uma parte da solução, as mesmas ambicionavam vender a solução completa em termos de controle, o que gerou uma maior difusão do modelo SCADA. A maioria destas empresas, como a GE, RockWell, Foxboro e Schneider não desenvolveram seu próprio sistema, a alternativa encontrada foi comprar empresas menores e adequar os sistemas ao seu hardware. Desta forma no final da década de 1990 só restavam aproximadamente 15 empresas com mercado global para os sistemas supervisórios.

O crescente aumento da capacidade de processamento gráfico nos anos 2000, permitiu o desenvolvimento de softwares antes impensáveis, a visualização tridimensional (3D) de objetos e processos têm aumentado junto com essa evolução. Aplicativos de visualização 3D tem sido desenvolvidos e popularizados desde os anos 1980, e tiveram início com os

programas de desenho técnico (CAD) sendo o AutoCAD® um dos primeiros, posteriormente o 3d Studio Max® foi desenvolvido pela mesma empresa a Autodesk®. O qual é utilizado para produção de animações em 3d assim como visualização artística e arquitetônica, a redução no custo dos computadores a partir de 1990, proporcionou a popularização deste tipo de software, muitas pessoas que antes necessitavam de computadores que custavam 30 mil dólares em 1990 só necessitavam gastar cerca de 3 mil dólares no ano 2000 para obter resultados cada vez mais realistas segundo Sathyanarayana e Kumar (2008). No começo do século XXI programas de visualização 3D já eram utilizados em diversas áreas do conhecimento: desenvolvimento de equipamentos, modelagem 3D, filmes, entre outros, a aplicação que tem levado ao limite da tecnologia é o desenvolvimento de jogos, onde é possível obter algo próximo da realidade em muitos casos, e tudo ocorre em tempo real pela interação do usuário. Porém não é possível encontrar muitos artigos recentes na literatura utilizando sistemas de visualização avançada em processos (MACCARANA et al. 2016).

Com a crescente aplicação do conceito de indústria 4.0 o uso de softwares 3D, impressoras 3D, e da tecnologia avançada aplicada a química, houve um grande crescimento no número de publicações envolvendo estes conceitos: Kitson et al. (2016) desenvolveu uma sistemática de produção de recipientes para reação utilizando uma impressora 3D. Conseguindo eficiência e rapidez no desenvolvimento de reatores, atendendo as necessidades específicas de cada experimento. Com o uso de automação de baixo custo Nejatimoharrami et al. (2016) desenvolveu experimentos bioquímicos com menos esforço, mais precisão, menos barulho e menor ruído nos resultados. As aplicações em microfluidica também tem se beneficiado do conceito de IOT e indústria 4.0, Sharma et al. (2017) utilizou impressoras 3D para a confecção de microreatores de baixo custo, ressaltando os pontos positivos e negativos, inclusive com relação ao meio ambiente.

A partir do contexto citado anteriormente, pode-se perceber que o momento atual é propício para o desenvolvimento de softwares de controle em 3D, pois até o momento os programas de controle de processo se limitam a visualização do processo em 2D, onde se controla e atua sobre as correntes materiais e energéticas do processo. Com o aumento da capacidade de processamento dos computadores e as sucessivas reduções de custos dos mesmos, levou o desenvolvimento do controle de processos a um novo patamar, o acompanhamento interno dos equipamentos, visualizado em tempo real por câmeras, e também por visualizações 3D dos processos e equipamentos. O acompanhamento de um

processo em 3D é intuitivamente mais apurado que o acompanhamento por simples gravuras, que muitas vezes não representam o objeto real nos painéis de controle industriais. Todos estes conceitos estão inseridos nas idéias de indústria 4.0, que significa uma nova revolução em termos de acompanhamento e operação de processos industriais.

3 FUNDAMENTAÇÃO TEÓRICA

Diversos conceitos consolidados foram levados em consideração nesta pesquisa, por tanto, uma fundamentação teórica foi necessária.

3.1 Processos reativos de separação

Processos reativos de separação são os únicos que combinam reações químicas e separações físicas em um único equipamento.

Em processos convencionais se costuma ter diversos equipamentos nos quais são necessárias transferências de materiais entre os mesmos, assim como também refluxos materiais dirigidos para o mesmo equipamento, o que eleva o gasto energético, pois são necessárias bombas hidráulicas. Nos processos reativos de separação, a integração tanto energética quanto mássica ocorre em um mesmo equipamento, diminuindo o custo com acessórios como bombas e trocadores de calor. Segundo Kulprathipanja (2002) e Sundmacher e Kienle (2002) têm como características:

- Costumam ter custo reduzido em comparação com os processos convencionais;
- Muitos processos estão consolidados, pois já existem há mais de 60 anos (como a extração reativa);
- Estão sendo amplamente estudados por questões de otimização e custo reduzido;
- Existe um crescente interesse que é refletido no aumento do número de publicações anuais.

Mesmo apresentando maiores dificuldades na modelagem e desenvolvimento, os processos reativos de separação são alvo de interesse que vem aumentando nos últimos tempos. Uma forma de ponderar os pontos positivos e negativos deste tipo de processo é fazer uma comparação direta, e assim avaliar lado a lado, como citado na Tabela 1:

Tabela 1 - Vantagens e desvantagens na aplicação de processos reativos de separação.

Vantagens	Desvantagens
Melhora as taxas de reação	Tecnologia relativamente nova
Aumenta a conversão	Janela limitada de aplicações
Aumenta a seletividade	Desenvolvimento experimental extenso
Reduz a severidade das condições	Requer uma modelagem complexa
Aumenta a vida do catalisador	Aumenta a complexidade operacional
Separação simplificada	Custo de desenvolvimento significativo
Aumenta a qualidade do produto	Riscos na construção de equipamentos maiores
Benefícios na integração energética	
Reduz custos de operação	
Reduz o investimento de capital	
Novas configurações dos processos	
Desenvolvimento de novos equipamentos	

Fonte: Kulprathipanja (2002).

Segundo Kulprathipanja (2002) e Kreul et al. (1999) as vantagens são numerosas nos processos reativos de separação, em alguns casos, a melhoria nas taxas de reação pode permitir ao operador reduzir a severidade da reação (temperatura, pressão, pH, por exemplo). Isso pode levar a melhorias também na seletividade, qualidade do produto, vida do catalisador, devido a união de um processo de separação e reação é possível reduzir os custos com a produção de equipamentos mais compactos.

O acoplamento das forças de transferência de massa e dos mecanismos de reação pode aumentar a vida do catalisador em muitos processos, (KULPRATHIPANJA, 2002; DOHERTY, 1992). Uma das desativações primárias do catalisador pode ocorrer por inibição via produtos pesados, mais polares ou produtos secundários indesejáveis. O aumento da transferência de massa no separador reativo oferece uma melhor irrigação e renovação de superfície no catalisador, deixando a superfície do catalisador livre de inibidores. O aumento de seletividade pode ser proveniente pela redução na geração de produtos secundários responsável em certas situações pela inibição do catalisador. A combinação desses efeitos

oferece melhorias na atividade do catalisador, aumentando a produtividade e reduzindo custos.

A junção de reação e separação no mesmo equipamento une os efeitos de transferência de calor, o que conduz a formas alternativas de se projetar a adição e a remoção de calor nos processos reativos de separação. Como citado por Kreul et al. (1999) e também por Sundmacher e Kienle (2002), em muitos casos reatores exotérmicos (fontes de calor) podem ser acoplados diretamente a processos intensivos de separação como colunas de destilação (utilizando trocadores de calor). Na separação reativa, a troca térmica pode ocorrer por contato direto ao contrário das transferências indiretas por trocadores de calor dos processos convencionais, assim é possível oferecer opções diferentes e mais eficientes de projeto para integração energética no processo.

Reduções de incrustações e coqueamento são possíveis no projeto de separadores reativos, os quais podem afetar a vida do catalisador. A diminuição na formação de produtos secundários diminui a formação de “polímeros” ou “pesados” os quais depositam substâncias de alta massa molecular no reator convencional em equipamentos de processamento posterior, levando a incrustações nos mesmos. Incrustações e coqueamento reduzidos nos trocadores de calor levam a coeficientes de transferência de calor maiores. A redução desses efeitos indesejados no interior da coluna de destilação leva a operações sustentáveis com a performance idealizada no projeto. Redução nas incrustações levam a longos tempos de funcionamento sem interrupções de operação para limpeza do equipamento para restaurar a performance operacional. O que leva a um crescimento na eficiência de operação, leva a um maior rendimento do processo de forma geral, reduzindo inclusive tempo de manutenção e custo de limpeza na operação da unidade, como exposto por Kreul et al. (1999).

Quando o acoplamento de separação e reação leva a uma redução no número de equipamentos necessários no processo e uma redução no tamanho dos mesmos, benefícios podem ser conseguidos em termos de projeto. Além disso, o aumento da segurança da unidade, se projetos de separadores reativos reduzem a quantidade de catalisador, reagentes ou produtos químicos perigosos no equipamento, a unidade será feita “inerentemente segura”.

Os benefícios acima podem também levar à redução no investimento de capital através da consolidação de múltiplos equipamentos do processo num só lugar e/ou através da eliminação das correntes de reciclo. Impregnação de um adsorvente com catalisador para a

destruição e deposição de produtos secundários pode levar a uma forma simplificada de recuperação no qual a corrente de produto recirculado pode ser eliminada em favor de uma vazão levemente maior de tratamento do produto.

Outra vantagem a ser considerada é que por serem tecnologias recentes, os resultados podem ser patenteados (SUNDMACHER; KIENLE, 2002), diferenciando o processo da concorrência e garantindo a exclusividade dos benefícios econômicos. As patentes podem ser ainda uma porta aberta para a comercialização de licenças de uso para outros, podendo inclusive ser uma fonte de faturamento adicional.

Os benefícios acima citados por Kulprathipanja (2002) podem levar a uma redução significativa no custo de operação, aumentar a produtividade e a qualidade do produto, refletindo em um melhor uso das matérias primas e um menor custo de matéria prima por unidade de produto. Aumentar a atividade e a vida útil de um catalisador leva a uma redução do custo do mesmo. Benefícios na integração energética resultam na redução do custo com utilidades. Redução de incrustações e coqueamento levam a uma melhor operação na produção, com custos menores de manutenção. E ainda a possibilidade de serem desenvolvidas patentes sobre o processo.

As desvantagens também precisam ser consideradas. Processos reativos de separação são geralmente tratados como “nova tecnologia”. O desenvolvimento e comercialização destas novas tecnologias geralmente estão associados com um alto custo e um alto risco em sua implantação.

A janela operacional associada com o desenvolvimento de separações reativas é normalmente mais limitada que o desenvolvimento de sistemas de separação e reação em separado segundo Kiss e Bildea (2012). A junção das condições de projeto requerem desempenhos razoáveis de reação e separação, que nos processos de separação reativa são muito estreitos e às vezes inexistentes, às vezes as sobreposições de condições para reação e separação é tão pequena que suas aplicações ficam muito restritas.

O desenvolvimento de processos de separação reativa normalmente requer um enorme esforço segundo Kulprathipanja (2002) e Kreul et al. (1999), pois soluções que funcionam para uma aplicação não necessariamente se aplicam para uma situação parecida. Mapas e receitas para o desenvolvimento de processos nem sempre estão disponíveis, embora importantes descobertas tenham sido feitas nos últimos tempos, como métodos para

selecionar e desenvolver separadores reativos para aplicações mais gerais. Dados experimentais em grande número e que tenha confiabilidade também são necessários para esclarecer e validar os princípios envolvidos e estabelecer uma janela de operação para aplicação no processo.

O acoplamento de reação e separação necessariamente aumenta a complexidade do processo de modelagem com o acoplamento de forças para a reação e separação. Como citado por Sundmacher e Kienle (2002). Modelagens que resultem em sucesso requerem uma análise próxima dos fenômenos relevantes em micro e macro escala. A modelagem precisa levar em consideração a produtividade necessária e o espaço físico disponível, a precisão obtida é função do modelo utilizado, dos parâmetros e dos dados disponíveis assim como o poder computacional.

O extenso desenvolvimento de um equipamento frequentemente requer detalhes de como os fenômenos de reação e separação podem ser acoplados em um único vaso. Um maior cuidado é necessário para se evitar problemas de desempenho tanto no desenvolvimento quanto na produção de equipamentos maiores (escala industrial), como citado por Kreul et al. (1999). O projeto de separadores reativos normalmente exibe uma grande interação entre os componentes em operação e o acesso restrito a certos componentes em operação. Problemas como falhas na operação inicial de uma unidade, assim como dificuldades no acesso para uma rápida manutenção, são um grande incentivo para investir um maior esforço no desenvolvimento do equipamento.

O desenvolvimento de processos reativos de separação geralmente aumenta a complexidade de operação e controle do processo segundo Kiss e Bildea (2012), Sundmacher e Kienle (2002) e Mansouri et al. (2016). A junção de reação e separação normalmente reduz o número de variáveis de controle que podem ser usadas para manipular o processo. Variáveis tradicionais de controle do processo como temperatura, vazão e pressão, pois elas estão incorporadas na operação da unidade como “respostas” e não “entradas”, assim, se torna mais difícil relatar causa e efeito no controle do processo. A complexidade da rede de controle precisa ser aumentada para acomodar controles indiretos, novas variáveis de controle precisam ser identificadas, e em muitos casos, controles chaves de processos precisam ser estabelecidos para inferir ou estimar respostas dos processos para melhorá-los. Faixas de controle do processo são frequentemente restritas a uma pequena janela de operação. A estabilidade do sistema de controle do processo e a robustez podem ser comprometidas por

pequenas mudanças nos parâmetros de projeto e pode conduzir a resultados drasticamente diferentes do esperado se não houver a modelagem adequada. A vulnerabilidade do sistema a perturbações pode crescer, com isso a inicialização e o desligamento da unidade necessitam de cuidados especiais, inclusive para manter a integridade do catalisador durante a inicialização.

Os riscos associados à comercialização de uma nova tecnologia de separação reativa são usualmente grandes segundo Kreul et al. (1999). Estes podem ser minimizados com pesquisas sobre a demanda comercial e no desenvolvimento, porém os riscos não podem ser eliminados completamente, estes são ampliados pela falta de flexibilidade operacional para um separador reativo, e pelo fato de que provavelmente um equipamento do tipo ainda não foi projetado. Opções para resolver as deficiências de desempenho com mudanças nos parâmetros de operação ou procedimentos são restritos pela falta de flexibilidade operacional. Quedas de desempenho após o desenvolvimento do equipamento também podem elevar os custos de desenvolvimento.

3.2 Destilação reativa

Segundo (SUNDMACHER; KIENLE, 2002), o princípio da destilação reativa é simples, uma parte da coluna de destilação que contém catalisador é utilizada para realizar a reação desejada. O catalisador pode estar empacotado em fase sólida, ou pode estar na mesma fase das espécies reagentes. A alimentação deverá ocorrer próxima à secção do catalisador, acima ou abaixo dependendo da volatilidade. A reação ocorre normalmente na fase líquida na presença do catalisador, os produtos da reação são removidos continuamente, como destilado ou como produto de fundo, enquanto se mantêm os reagentes no interior da coluna.

Com a destilação e a reação ocorrendo no mesmo equipamento é possível obter diversas vantagens em comparação à sequência convencional reação-separação, como citado por (KULPRATHIPANJA, 2002).

- Deslocamento de equilíbrio – a remoção de um ou mais produtos da fase reacional causa um reestabelecimento de equilíbrio para altas conversões. Se as volatilidades relativas forem favoráveis pode ser possível manter os reagentes na coluna, retirando apenas os produtos. Mesmo se apenas um produto for continuamente retirado da fase

reacional, o crescimento na conversão ainda leva ao benefício da redução do custo com reciclos.

- Redução no custo da planta – simplificação ou eliminação do sistema de separação pode levar a uma redução significativa de custo.
- Benefícios de integração energética – se a reação for exotérmica, o calor da reação pode ser utilizado para prover calor de vaporização e reduzir a carga no refervedor. Outras formas de integração energética podem ser obtidas através do uso de condensadores, refervedores e bombas.
- Eliminação de azeótropos – A destilação reativa é particularmente vantajosa quando o reator produz uma mistura de espécies que formam azeótropos entre si. Em muitos casos uma separação convencional requer muitas colunas de destilação ou inertes para quebrar a azeotropia, porém, a escolha certa nas condições de destilação reativa pode permitir que a composição dos componentes na mistura azeotrópica seja deslocada pela reação química, evitando, assim, que a azeotropia ocorra.
- Aumento de seletividade – a remoção de um dos produtos da mistura reacional ou mantendo uma baixa concentração de um dos reagentes pode levar a redução das reações secundárias e aumentar a produtividade do produto desejado.
- Projeto de reatores multifuncionais – em alguns casos muitos catalisadores diferentes podem ser adicionados em diversas secções da coluna, permitindo mais de uma função reacional, o que pode ser interessante quando se quer processar uma alimentação impura.

Apesar dos diversos pontos positivos, há também os pontos negativos:

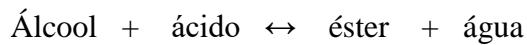
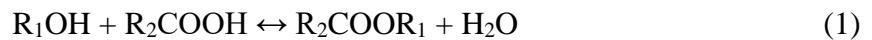
- Restrições de volatilidade – os reagentes e os produtos devem ter uma diferença significativa de volatilidade para manter altas concentrações de reagentes e baixa concentração de produtos na zona de reação na coluna.
- Tempo de residência requerido – se o tempo de residência para a reação for longo, uma grande coluna com grandes pratos será requerida para garantir tempos altos de retenção, com isso pode ser mais econômico utilizar o arranjo reator-separador.

- Condições de processamento – em alguns processos as condições ótimas de temperatura e pressão podem estar longe das condições ótimas de reação, e o contrário pode ocorrer.

Apesar destes inconvenientes, em alguns processos os benefícios da destilação reativa podem levar a uma redução significativa de custo, o que levou a um considerável interesse nessa tecnologia desde o início de sua comercialização no início dos anos de 1980. A tecnologia envolvida na destilação reativa foi enormemente beneficiada pela experiência adquirida nas primeiras aplicações industriais, do aumento do entendimento teórico e de modelagem, como também no desenvolvimento de novos catalisadores. Nas seções seguintes estes assuntos serão detalhados, porém será visto agora a primeira aplicação comercial desta tecnologia.

3.2.1 Aplicações da destilação reativa na indústria

Destilação reativa foi inicialmente considerada para processos de esterificação em uma série de patentes por (BACKHAUS, 1921, 1922, 1923) em processo de esterificação que genericamente pode ser expresso por:

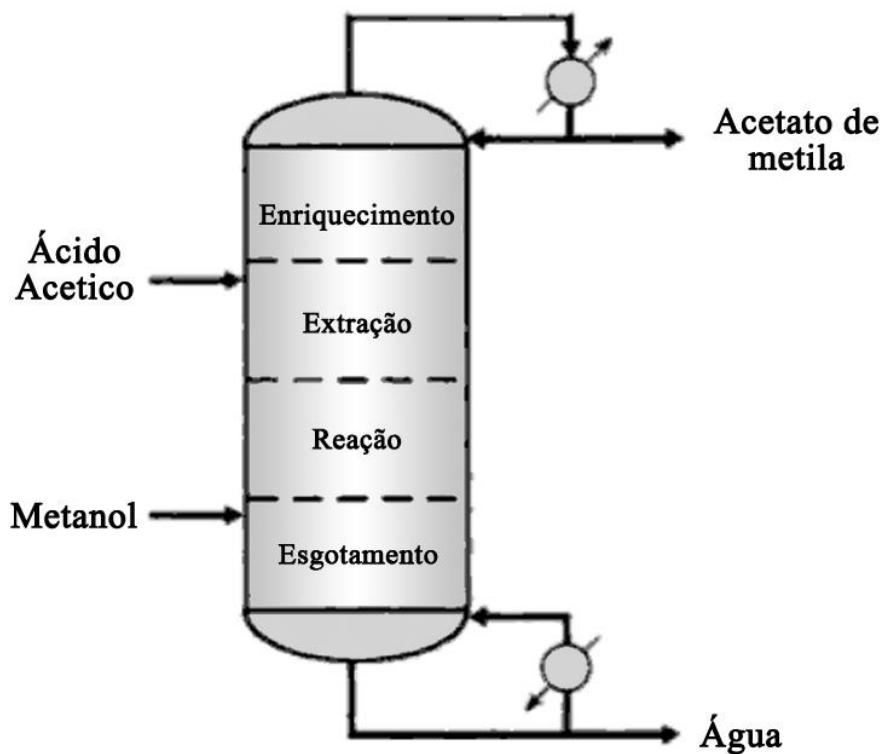


Muitos fatores fazem este sistema ideal para a destilação reativa segundo (SUNDMACHER; KIENLE, 2002). Primeiro: misturas de ácidos orgânicos, ésteres, álcoois, e água têm uma forte tendência a formar azeótropos. Em muitos processos de esterificação azeótropos binários podem ser formados entre reagentes e produtos, assim como também é possível a ocorrência de azeótropos ternários e superiores. Por isso, a separação do produto desejado da mistura reacional é normalmente cara, utilizando muitas colunas de destilação. Segundo: reações de esterificação ocorrem em temperaturas moderadas na fase líquida, sob condições que são convenientes para a destilação. Terceiro: as reações são limitadas por equilíbrio, levando a incentivar a remoção dos produtos para melhorar o desempenho da reação. Quarto: por causa da existência de azeótropos, normalmente é possível explorar diferenças de volatilidade na coluna de destilação reativa para manter altas concentrações de reagentes na zona reacional.

Tudo que foi exposto é bem representado pelo processo de produção do acetato de metila, desenvolvido pela *Eastman Chemical* nos anos de 1980 (AGREDA; PARTIN, 1984; AGREDA; PARTIN; HEISE, 1990). O acetato de metila é formado pela reação entre metanol e ácido acético. Existem azeótropos entre acetato de metila e metanol, entre acetato de metila e água, e o azeótropo seguinte formado entre água e ácido acético, o que demonstra a dificuldade na separação destes componentes por processos convencionais.

A coluna de destilação reativa desenvolvida por *Eastman Chemical*, está apresentada simplificadamente na Figura 1. A operação do projeto é melhor entendida considerando as funções de cada secção da coluna: a secção do topo serve para enriquecer o produto acetato de metila, nessa secção as únicas espécies presentes em quantidades significativas são o acetato de metila e o ácido acético. Não existem azeótropos entre essas espécies, e esta secção atua como um retificador convencional. O ácido acético é alimentado abaixo desta secção, e flui para baixo, para a secção de extração, onde o ácido acético age como “lavador” para qualquer água que tenha sido levada para cima no vapor e quebra o azeótropo acetato de metila-água por destilação extractiva.

Figura 1 - O processo de produção do acetato de metila pela *Eastman Chemical*.



Fonte: Kulprathipanja (2002)

O ácido acético entra nos estágios reativos da coluna, onde encontra o metanol e realiza a esterificação. Ácido sulfúrico é utilizado como catalisador para a reação, e os estágios reativos são designados para a conversão completa de ácido acético. O acetato de metila que é produzido nos estágios reativos é vaporizado, removendo o calor da reação e deslocando o equilíbrio da reação para altas conversões. Finalmente, no fundo da coluna qualquer metanol e ácido acético não reagido são removidos junto com a água.

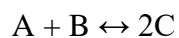
Muitos problemas práticos vieram no desenvolvimento comercial deste processo, e estes são descritos por Agreda et al. (1990). Como a necessidade de se utilizar uma coluna lateral para se remover pequenas impurezas. Após testes de bancada e testes pilotos, *Eastman Chemical* produziu 23 toneladas por hora em uma unidade comercial em 1983, esta unidade sofreu pequenas modificações para remover as limitações e otimizá-la, e em 1990 Eastman Chemical divulgou que esta unidade estava operando a 125% da capacidade de projeto. O processo Eastman de produção de acetato de metila serviu de exemplo em muitos estudos acadêmicos de destilação reativa, pois, a não utilização de uma fase sólida de catalisador faz o sistema ser relativamente simples de ser estudado.

3.2.2 Teoria sobre a destilação reativa

O forte interesse industrial na destilação reativa nas últimas décadas fez com que o entendimento teórico fosse ampliado sobre colunas de destilação reativa e os métodos de desenvolvimento de projetos. Em seguida será dada uma introdução para algumas propriedades importantes de equilíbrio na destilação reativa e introduz os fundamentos da modelagem baseada em taxas.

Embora a modelagem na destilação reativa necessite considerar taxas de reação, abordagens sobre os comportamentos limites podem ser obtidas considerando o sistema em equilíbrio.

Segundo (VENIMADHAVAN et al. 1994) o comportamento de equilíbrio nos processos de destilação reativa varia entre dois extremos, correspondendo ao controle por equilíbrio químico, e o controle por equilíbrio de fases. Isso pode ser ilustrado por um exemplo simples, considerando a reação.



Essa reação possui apenas 3 componentes, E pode permitir examinar um gráfico bidimensional do espaço de composição, para visualizar o efeito do equilíbrio de reação ou de fase.

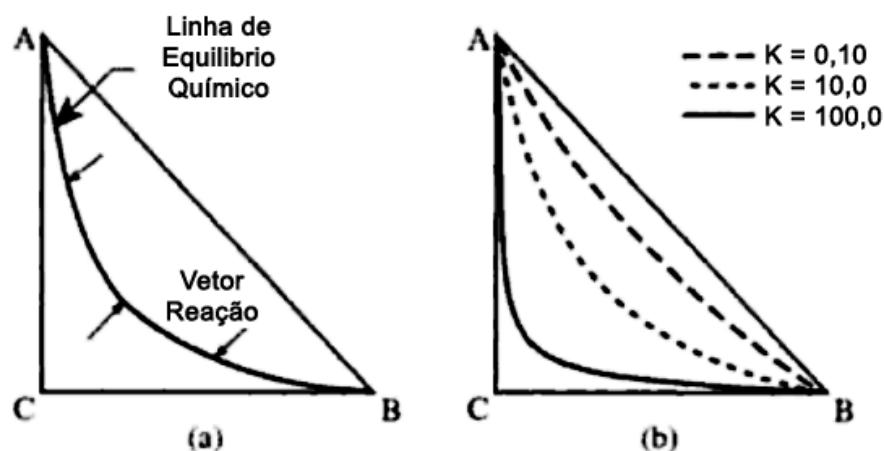
A constante de equilíbrio (K) pode ser dada por:

$$K = \frac{x_c^2}{x_a x_b} \quad (2)$$

Onde x_i é a fração em mol do componente i , e K pode ser encontrado pela mudança na energia livre de Gibbs da reação.

Se o valor da constante de equilíbrio for conhecido, então é possível plotar o espaço das possíveis composições de ABC que satisfaçam a equação anterior. A Figura 2a mostra este gráfico, onde qualquer composição inicial irá reagir de acordo com a equação citada acima, quando este ponto de equilíbrio químico é atingido, e não há mais reação. Qualquer processo de reação pode ser plotado em um diagrama triangular como um vetor que segue a equação de reação, partindo de um ponto inicial de composição e terminando na linha que satisfaz a equação de equilíbrio.

Figura 2 - Diagramas ternários de equilíbrio químico



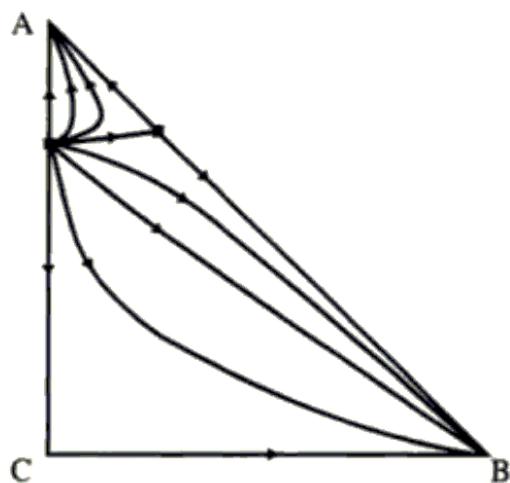
Fonte: Venimadhavan et al. (1994).

A Figura 2(b) mostra um diagrama com diversas linhas de equilíbrio químico para diversos valores de K . Para valores baixos de K a curva de equilíbrio se aproxima da aresta A-

B do diagrama. Isso é esperado, pois para baixos valores de K apenas uma pequena parte da mistura de alimentação é convertida; no limite oposto, para altos valores de K a curva de equilíbrio se aproxima das arestas A-C e B-C. Novamente isto é esperado, pois um alto valor de K favorece a formação de uma grande quantidade de C, neste caso os reagentes inicialmente abundantes são quase completamente consumidos, fazendo o resultado do equilíbrio ser essencialmente C.

O diagrama de composição mostra o equilíbrio químico e pode ser contrastado com o diagrama que mostra o equilíbrio de fases. Para processos de destilação o diagrama importante a ser mostrado é o de curvas residuais ou mapa de linhas de destilação. Estes diagramas mostram os perfis de composição em refluxo total para colunas de recheio e colunas de estágio, respectivamente. Cada curva residual (ou linha de destilação) é marcada com uma seta que indica a direção do crescimento de temperatura. A Figura 3 mostra um mapa de curvas residuais para um sistema que exibe azeótropos binários de baixo ponto de ebullição entre os componentes A-B e A-C.

Figura 3 - Mapa de curvas residuais de destilação para um sistema com dois azeótropos binários.



Fonte: Venimadhavan et al. (1994).

Uma característica dos mapas de curvas residuais é que as curvas começam e terminam em pontos estacionários das equações diferenciais das quais as curvas residuais são calculadas. Esses pontos estacionários correspondem a componentes puros ou azeótropos. Existem três tipos de pontos estacionários segundo Kulprathipanja (2002):

- Pontos instáveis – Todas as curvas residuais se dirigem para fora do ponto.
- Pontos estáveis – Todas as curvas residuais se dirigem em direção ao ponto.
- Pontos de sela – Algumas curvas se dirigem para fora e outras em direção ao ponto.

Por exemplo, na Figura 3, os componentes A e B são nós estáveis, o azeótropo A-C é um nó instável, e o componente puro C e o azeótropo A-B são pontos de sela. Doherty e Perkins (1979) desenvolveram uma relação matemática entre o número de pontos estacionários de cada tipo. Os pontos estacionários são importantes no projeto de destilação, como a natureza dos pontos estacionários pode nos dizer que tipo de azeótropo ou componente puro será mais fácil ou difícil de separar via destilação.

Sistemas de destilação reativa podem conter azeótropos que agem como pontos estacionários. A existência destes tipos de azeótropos foi sugerida por Barbosa e Doherty (1987) e provada por uma série de experimentos por Song et al. (1997). Ung e Doherty (1995) descrevem as condições necessárias e suficientes para a ocorrência de um azeótropo reativo: se um azeótropo reativo ocorrer, então uma mistura em ebulação é formada com composição constante, mas não igual, na fase líquida e na fase vapor; isto pode ocorrer se o balanço da taxa de reação e de vaporização ocorrer a composição constante. Estes artigos influenciam trabalhos até os dias atuais, como o de Rastegari et al. (2017).

O comportamento de equilíbrio da maioria dos sistemas reativos de destilação será o intermediário entre o equilíbrio de fases e o equilíbrio químico. Venimadhavan et al. (1994) sugere o número de Damköhler para modelar esta variação. O número de Damköhler é definido como,

$$Da = \frac{\frac{H_0}{V}}{\frac{1}{k_1}} = \frac{H_0 k_1}{V} \quad (3)$$

sendo H_0 é a retenção líquida (mol); k_1 é a constante cinética de pseudo primeira ordem (s^{-1}), e V é vazão molar de vapor ($mol \cdot s^{-1}$).

O número de Damköhler é a razão entre duas características; tempo de residência líquida (H_0/V) e o tempo característico de reação ($1/k_1$). Para baixos valores de Damköhler,

Da < 0,5, a taxa de reação é lenta com relação ao tempo de residência disponível, e o sistema é dominado pelo equilíbrio de fases. Para altos valores de Damköhler, Da > 10, a taxa de reação é rápida e o equilíbrio químico é praticamente atingido em cada estágio reativo. Se o número de Damköhler variar entre estes dois valores, então, não há um controle do equilíbrio de fase nem do equilíbrio químico, e um modelo baseado em taxa precisa ser utilizado, como descrito na próxima seção.

É importante entender a existência de azeótropos reativos e como a formação dos azeótropos reativos pode limitar o desempenho dos sistemas de destilação reativa. Na prática, isso requer o balanço preciso entre a taxa de reação e a taxa de vaporização. Normalmente, colunas de destilação reativa exibem seções nas quais as composições mudam muito pouco de estágio para estágio, se aproximando dos valores estacionários. Venimadhavan et al. (1994) apresentaram uma análise detalhada desta situação e mostram que o local dos pontos estacionários varia entre o equilíbrio de fase e o equilíbrio químico com o aumento do número de Damköhler.

A dificuldade de modelagem na destilação reativa como um processo de estágios em equilíbrio é devido aos seguintes fatores:

- Equilíbrio de fases
- Taxa de transferência de massa entre a fase líquida e de vapor
- Transferência de massa com reação (normalmente na fase líquida)
- Taxa de reação
- Atividade do catalisador, e possível transferência de massa com o mesmo.
- Transferência de massa dos produtos na fase líquida

3.2.3 Modelagem e projeto

O desenvolvimento de modelos realísticos de processos é o passo chave para avaliar a viabilidade de um processo de destilação reativa, como exposto por Kulprathipanja (2002). Boa parte da modelagem pode ser realizada utilizando simuladores comerciais. Entre as empresas desenvolvedoras se tem a Aspentech® e Simulation Sciences®. Uma modelagem bem sucedida requer uma compreensão tanto do equilíbrio químico quanto do equilíbrio de fases no processo.

A modelagem do processo de equilíbrio precisa começar com a determinação das constantes de equilíbrio químico e dos componentes do equilíbrio de fase no sistema não reativo. A presença de azeótropos pode ser determinada pela observação de dados para misturas binárias, ternárias ou quaternárias das correntes de alimentação e de produtos na ausência de catalisador. O equilíbrio de fases também é importante para os estágios não reacionais da coluna de destilação reativa, por exemplo, para os estágios de retificação ou purificação acima ou abaixo da zona de reação da coluna.

O número de Damköhler e a constante de equilíbrio químico podem ser utilizados para produzir um quadro de possibilidades preliminar da utilização da destilação reativa, como proposto por (VENIMADHAVAN et al., 1994). A Tabela 2, é uma adaptação deste artigo, mostrando o impacto das diferentes combinações de Da e K no processo.

Tabela 2 - Influência dos valores de Da e K segundo Venimadhavan et al. (1994).

Da	K	Impacto no processo
Baixo	Baixo	Reação direta lenta e reação reversa rápida, essencialmente, sem formação de produtos e o processo é muito difícil de processar em qualquer vaso, destilação reativa normalmente não oferece vantagem sobre o processo convencional em um reator por conta do grande volume necessário.
Baixo	Alto	Reação direta lenta, mas também reação reversa lenta. Nesse caso a formação de produto é alta. Destilação reativa pode trazer benefícios a menos que o volume de retenção líquida não seja muito grande.
Alto	Baixo	Formação rápida de produtos, mas também rápida reação reversa, baixa formação de produtos. Destilação reativa pode ser vantajosa se os produtos podem ser removidos rapidamente da zona de reação, prevenindo a reação reversa.
Alto	Alto	Reação direta rápida e reação reversa lenta. Essencialmente esta é uma reação irreversível, que pode se desenvolver num reator simples. Destilação reativa não é aconselhada a menos que seja utilizada para obter condições que minimizem reações secundárias.

Se a destilação reativa é uma opção viável, então o número de Damköhler pode ser usado para determinar se o processo será controlado pelo equilíbrio de fases ou pelo equilíbrio químico como citado na secção 2.2.2.

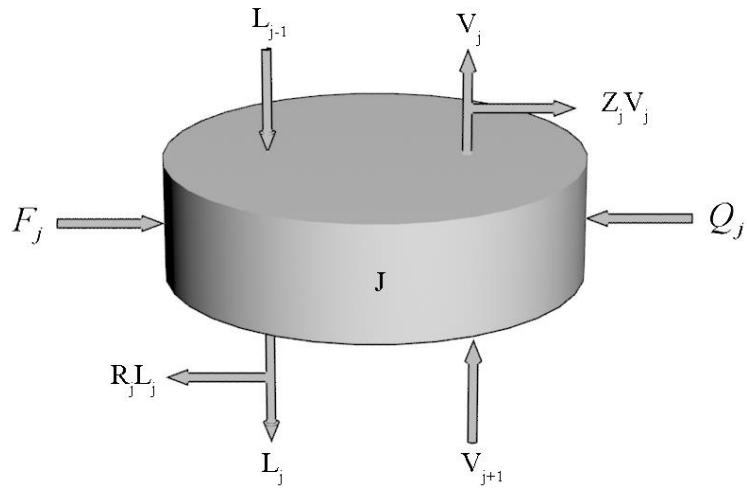
- $Da < 0,5$: controle do equilíbrio de fase. Se o equilíbrio de fase controla, então os estágios reativos serão fortemente influenciados pelos azeotropos que podem estar presentes na mistura não-reacional. Para baixos valores de Da um grande tempo de residência é necessário, modelos baseados em taxas precisam ser utilizados para projetar colunas neste regime, já que modelos de equilíbrios de fase não são capazes de descrever as características dos produtos, pois não é possível atingir o equilíbrio em cada estágio.
- $Da > 10,0$: controle do equilíbrio químico. Nesse caso os estágios podem ser assumidos como no estado de equilíbrio químico, cada estágio reativo pode ser modelado como um reator químico em equilíbrio com produtos líquidos e na forma de vapor. Muitos simuladores comerciais podem modelar estágios como estes.
- $0,5 < Da < 10,0$: limitado por taxas. Nesse caso intermediário de Da, modelos baseados em taxas são necessários, pois efeitos cinéticos e de equilíbrio de fases são importantes.

O engenheiro projetista tem certa liberdade para variar certos parâmetros do número de Damköhler, tais como retenção líquida do prato (ou retenção do estágio para colunas empacotadas) e a vazão de vapor. Ainda é possível variar o valor de Da variando a vazão de vapor, que pode ser modificada mudando a temperatura de operação do processo ou por pequenas modificações na pressão da coluna. A faixa de valores de Da que pode ser alcançada será limitada por considerações práticas como a escolha do recheio da coluna, inundação, arraste, entre outros.

3.2.4 Modelagem matemática

A destilação reativa envolve simultaneamente reação e destilação, como já foi dito. Segundo Seader e Henley (1998) a reação normalmente ocorre na fase líquida, ou na superfície sólida do catalisador em contato com a fase líquida. O esquema de prato utilizado na modelagem normalmente segue o que é exposto na Figura 4.

Figura 4 - Configuração geral de um estágio J na coluna de destilação reativa. (L=vazão líquida, F=vazão de alimentação, V=vazão de vapor, Q=calor cedido, Z,R=são retiradas laterais)



Fonte: autor

Em colunas de destilação reativa, um termo de taxa de reação necessita ser adicionado ao balanço de massa do componente i no estágio j . Assumindo que em cada estágio, a fase líquida está completamente misturada, a função resíduo $f_{i,j}^m$ é dada para o balanço material, e é expressa pela equação (4) (SEADER; HENLEY, 1998; MACHADO, 2009):

$$f_{i,j}^m = (R_j + 1)L_{i,j} + (Z_j + 1)V_{i,j} - (L_{i,j-1} + V_{i,j+1} + F_{i,j} + \sum_{k=1}^{nr} v_{i,k} \xi_{k,j}) = 0 \quad (4)$$

Na equação (4) $(R_j + 1)L_{i,j}$ é a vazão molar do composto i que sai do estágio j como líquido, da qual $R_j L_{i,j}$ é uma retirada lateral de líquido e $L_{i,j}$ é a vazão que chega ao estágio subsequente. Sendo $(Z_j + 1)V_{i,j}$ a vazão molar do composto i no vapor que sai do estágio j , da qual $Z_j V_{i,j}$ é uma retirada lateral de vapor e $V_{i,j}$ é a vazão que chega ao estágio seguinte. Nesta equação, $v_{i,k}$ é o coeficiente estequiométrico do componente i na reação k , $\xi_{k,j}$ é a taxa de reação da reação k no estágio j e nr representa o número de reações químicas no processo.

Segundo (MACHADO, 2009) diversos parâmetros devem ser conhecidos, entre eles é possível citar a temperatura e a pressão no estágio (T_j e P_j) e a vazão molar de cada um dos componentes i ($F_{i,j}$) na corrente de alimentação. Porém outros parâmetros devem ser especificados, tais como o número de estágios, razão de refluxo no condensador e no

referedor, número de reações químicas e a massa de catalisador em cada estágio da zona reacional.

Admitindo condições ideais, nas quais as correntes que deixam um estágio estejam em equilíbrio e que a coluna opere em condição de baixa pressão (aproximadamente 1 bar), as equações que definem o equilíbrio de fases têm, então, podem ser escritas como:

$$f_{i,j}^{eq} = \ln(x_{i,j}^I P_j) - \ln(x_{i,j}^{II} \gamma_{i,j}^{II} P_{i,j}^{sat}) = 0 \quad (5)$$

Na Equação (5) $x^I_{i,j}$ e $x^{II}_{i,j}$ são as frações em mol do componente i nas correntes de vapor e líquido que deixam o estágio j , P_j é a pressão de cada estágio, $P_{i,j}^{sat}$ é a pressão de saturação do componente i no estágio j e $\gamma_{i,j}^{II}$ é o coeficiente de atividade do componente i na fase líquida que deixa o estágio j . Em condições ideais desprezam-se a correção de Poynting e o coeficiente de fugacidade dos compostos puros saturados segundo (MACHADO, 2009). A expressão para a taxa de reação de cada reação k em um estágio j pode ser representada por:

$$\xi_{k,j} = \sum_{p=1}^2 k_{k,j} \prod_{i=1}^{nc} C_{i,j}^{\alpha_{i,k}} = \left[\sum_{p=1}^2 A_p \exp\left(-\frac{E_p}{RT_j}\right) \right]_k \prod_{i=1}^{nc} C_{i,j}^{\alpha_{i,k}} \quad (6)$$

Sendo $k_{k,j}$ é a constante cinética da reação k no estágio j , onde $p=1$ indica a reação no sentido direto e $p=2$ indica a reação no sentido inverso. $k_{k,j}$ pode ser expresso em termos da equação de Arrhenius, onde A_p é o fator de frequência, E_p é a energia de ativação, R é a constante universal dos gases ideais, e T_j é a temperatura no prato j . $\alpha_{i,k}$ é a ordem cinética do componente i em cada reação k e i, j C é a concentração molar do componente i no estágio j . Segundo (SEADER; HENLEY, 1998) a reação pode ser tratada como irreversível se $k_2 = 0$, reversível se k_2 for negativo e diferente de zero, ou ainda em equilíbrio. Aplicando-se logaritmo na Equação 06, é possível expressá-la na forma de resíduo:

$$f_{k,j}^r = \ln k_{k,j} + \sum_i^{nc} \alpha_{i,k} \ln \left(\frac{x_{i,j}^{II}}{V_j^{II}} \right) - \ln \xi_{k,j} = 0 \quad (7)$$

Na qual, V_j^{II} representa o volume molar da fase II (líquida) no estágio j . Admitindo-se que o volume molar seja o de uma solução ideal. As Equações 05 e 06 podem ser expressas

em função das atividades dos componentes na fase líquida ao invés das concentrações de cada componente. Assim, nesta situação, a Equação 06 torna-se:

$$f_{k,j}^r = \ln k_{k,j} + \sum_i^{nc} \alpha_{i,k} \ln(x_{i,j}^{II} \gamma_{i,j}^{II}) - \ln \xi_{k,j} = 0 \quad (8)$$

Devido ao gradiente de temperatura na coluna, a operação não é isotérmica, assim tem-se uma temperatura diferente em cada prato da coluna, representada por T_j . É necessária, portanto, a equação do balanço de energia:

$$f_j^h = (R_j + 1)H_j^{II} + (Z_j + 1)H_j^I - (H_{j-1}^{II} + H_{j+1}^I + H_{Fj} + Q_j) = 0 \quad (9)$$

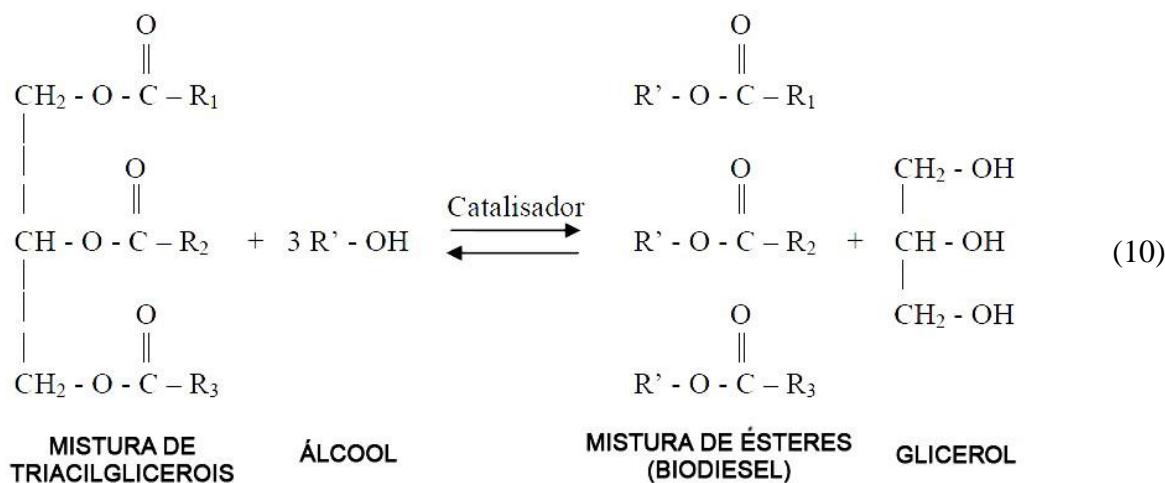
Na Equação 08, $(Z_j+1) H_j^I$ e $(R_j+1) H_j^{II}$ são as entalpias totais das correntes de vapor e líquido que deixam o estágio j e Q_j é o calor fornecido em cada estágio. Convenciona-se valor negativo ao calor que deixa a coluna e positivo para o calor que é fornecido à coluna.

Tabela 3 - Características de referedores e condensadores, adaptado de Machado (2009).

Condensador (estágio 1)		Referedor (estágio N)	
Parcial	Total	Parcial	Total
$Z_1=0$	$Z_1=0$	$Z_N=0$	$Z_N \neq 0$
$R_1=0$	$R_1 \neq 0$	$R_N=0$	$R_N=0$

3.3 Reação de transesterificação

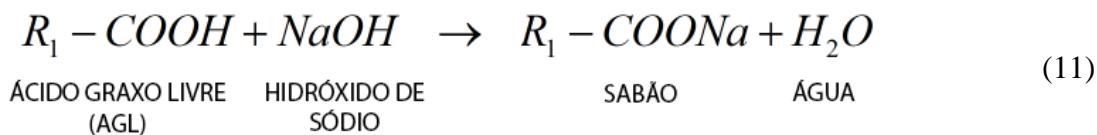
Uma das formas de se promover uma reação de transesterificação é quando um óleo de origem vegetal, ou gordura animal, entra em contato com um álcool (YEE et al., 2010; BARBOSA et al., 2010). O produto desta reação é uma mistura de éster (biodiesel) e glicerol como apresentado na equação 10.



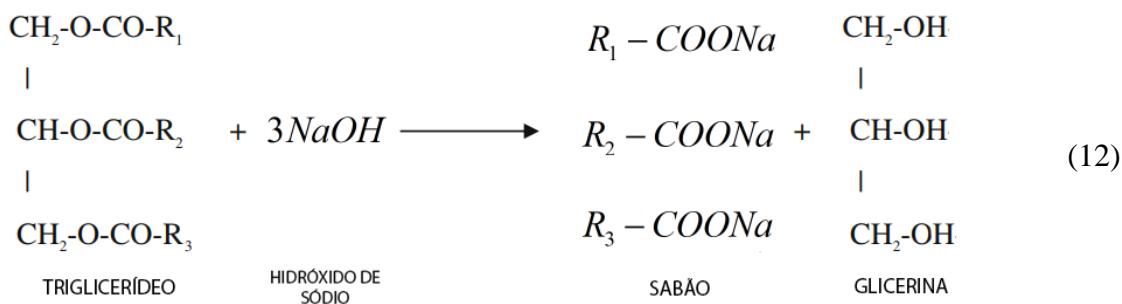
A presença de um catalisador é importante para auxiliar a reação. A operação com excesso de álcool se faz necessária quando se deseja deslocar a reação para o lado dos produtos. O álcool utilizado normalmente é o metanol ou etanol. O metanol é mais barato, porém é proveniente do petróleo, ou seja, não renovável. Já o etanol é mais caro porém é renovável, pois é produzido a partir de origem vegetal. O biodiesel produzido por via etanólica apresenta um número de cetano maior, ou seja, tem uma qualidade maior como biodiesel, porém segundo Leung et al. (2010) e Reyero et al. (2015) a reação de transesterificação utilizando etanol é mais lenta, produz mais carboxilatos (sabões) e requer uma série de cuidados.

3.4 Reação de saponificação

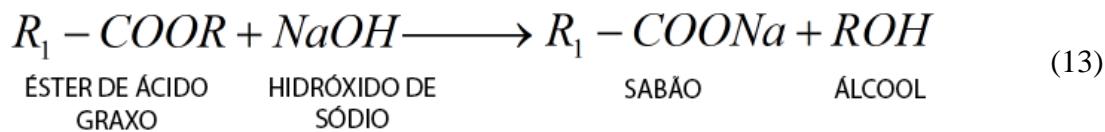
Óleos vegetais e gorduras podem conter pequenas quantidades de água e ácidos graxos livres (AGL). Para uma transesterificação catalisada por uma base, o catalisador alcalino que é usado reage com o AGL e forma sabão. A Equação 11 mostra a reação de saponificação do catalisador, no caso hidróxido de sódio, e o AGL, formando sabão e água segundo Leung et al. (2010).



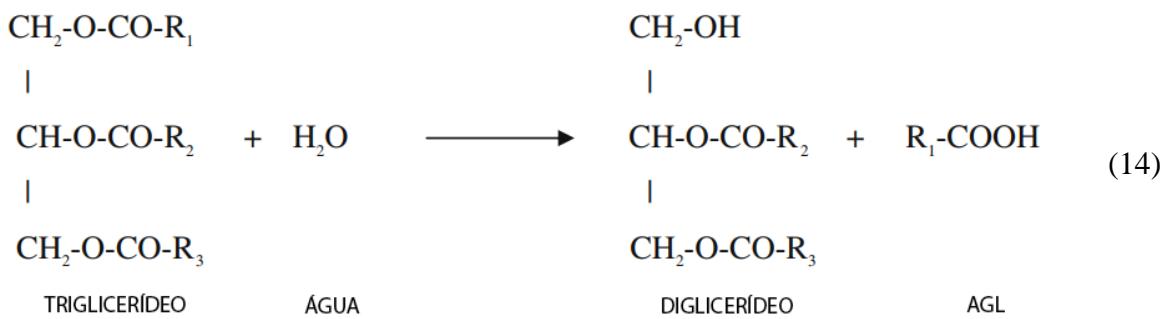
A reação de saponificação também ocorre a partir do triacilglicerol (EZE et al., 2014) reagindo com o catalisador básico, promovendo uma reação irreversível e formando sabão e glicerina, segundo a seguinte reação:



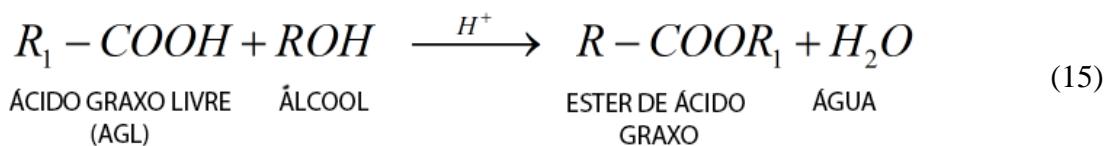
O éster produzido também está sujeito a saponificação, na presença de íons OH⁻ do catalisador, formando sabão, e regenerando o álcool utilizado, segundo Eze et al. (2014), como descrito na equação a seguir:



Como citado por Leung et al. (2010) o óleo adequado deve ter índice de acidez inferior a 2,5% mássico para não haver excesso de sabão no produto final, diminuindo a probabilidade da reação da Equação 10. Estas reações são indesejáveis porque o sabão diminui o rendimento do biodiesel e inibe a separação dos ésteres resultantes da reação. Além disso, o ácido graxo reage com o catalisador, o que significa que mais catalisador será necessário e, portanto, o processo envolverá um maior custo, segundo Gerpen et al. (2004). A água, originada pelos óleos e gorduras ou formada durante a reação de saponificação, retarda a transesterificação reação através da reação de hidrólise. Pode hidrolisar os triacilglicerois aos diglicérideos e forma mais AGL. A reação é mostrada na Equação 14:



No entanto, o AGL pode reagir com álcool para formar éster (biodiesel) por uma reação de esterificação catalisada por ácido. Essa reação é muito útil para óleos ou gorduras com um alto teor de AGL, como mostrado na Equação 15:



Normalmente, o catalisador para esta reação é o ácido sulfúrico concentrado. Devido a taxa de reação lenta e a elevada razão molar de álcool/óleo necessária, a esterificação catalisada por ácido não ganhou tanta atenção quanto a transesterificação catalisada por bases.

3.5 Planejamento fatorial

O planejamento fatorial de experimentos é constituído de um conjunto de técnicas estatísticas que proporcionam um método sistemático para planejar, executar e analisar experimentos. Partindo de um planejamento fatorial é possível, por exemplo, determinar se uma variável influencia significativamente a resposta de um experimento como exposto por Cunico et al. (2008). O planejamento fatorial de experimentos permite estimar os efeitos de diversos fatores, assim como as interações entre os mesmos sobre a variável de resposta.

Um processo pode depender de uma única variável, isso torna seu estudo simples, pois um gráfico em duas dimensões representa bem o processo. Porém quando um processo sofre

influencia de diversas variáveis o estudo se torna mais complexo. Necessitando de um estudo multivariado.

Nos mais diversos ramos do conhecimento humano, há a necessidade de se conhecer quais variáveis são importantes em um processo, para que se tenha um controle sistemático sobre o mesmo. E desta forma entender a faixa de valores possíveis destas variáveis, segundo Seborg (2016). Para se obter a otimização de um processo deve-se escolher um valor máximo e mínimo das variáveis durante a execução dos experimentos, como descrito por Cunico et al. (2008).

O planejamento fatorial é uma técnica com uso crescente em trabalhos científicos. Seu emprego possibilita conhecer as variáveis ou a combinação de variáveis que exercem uma maior influência sobre a resposta de um processo. Com este tipo de estudo é possível, por exemplo, reduzir o custo operacional, melhorando o rendimento a partir das respostas obtidas no planejamento fatorial, segundo Sayuri (2008).

Os métodos de otimização surgiram da necessidade de se aproveitar melhor o desempenho dos mais diversos sistemas, sendo a ideia de otimização correspondente a obter o melhor resultado possível de uma dada situação. Segundo Cunico et al. (2008) as instruções para se obter um ótimo são expressas através de métodos matemáticos que procuram maximizar ou minimizar alguma propriedade específica do sistema em estudo. É na escolha da propriedade a ser otimizada e das condições de controle onde que se aplica o maior esforço. Com isto é possível obter o conjunto de variáveis que leva o processo a ter um desempenho otimizado.

O planejamento fatorial tem sido aplicado em diversas pesquisas como um método onde se observa a variação simultânea das variáveis, onde as variáveis de interesse apresentam influências significativas na resposta. Para se realizar um planejamento fatorial escolhem-se as variáveis a serem estudadas e realizam-se experimentos para diferentes valores das variáveis estudadas.

De forma geral, o planejamento fatorial pode ser representado por b^a , onde “a” é o número de fatores e ”b” é o número de níveis escolhido para cada fator. Nos planejamentos experimentais as variáveis são normalmente exploradas em 2 níveis, sendo eles normalmente designados por (+) e (-). A atribuição destes sinais aos níveis superiores e inferiores respectivamente é feita de forma arbitrária e não interfere na realização dos experimentos ou

na interpretação dos resultados, além de permitir esquematizar o planejamento na forma de Tabelas.

Segundo Sayuri (2008) esta codificação de variáveis embora pareça desnecessária, é de grande utilidade quando se realizam cálculos para determinar qual a influência das variáveis estudadas e das suas interações no sistema de estudo.

Dentre os programas utilizados para se analisar os resultados do planejamento fatorial se tem o Statistica®, produzido pela STATSOFT®. O Statistica® fornece um abrangente conjunto de ferramentas para análise, gestão e visualização de bases de dados entre muitos outros. As suas técnicas incluem as mais vastas seleções de modelagem preditiva, agrupamentos (clustering) e ferramentas exploratórias. O Statistica® é uma plataforma analítica e testada, com mais de duas décadas de experiência no fornecimento de resultados academicamente comprovados, com uma comunidade mundial superior a 1 milhão de usuários, segundo dados da própria STATSOFT®.

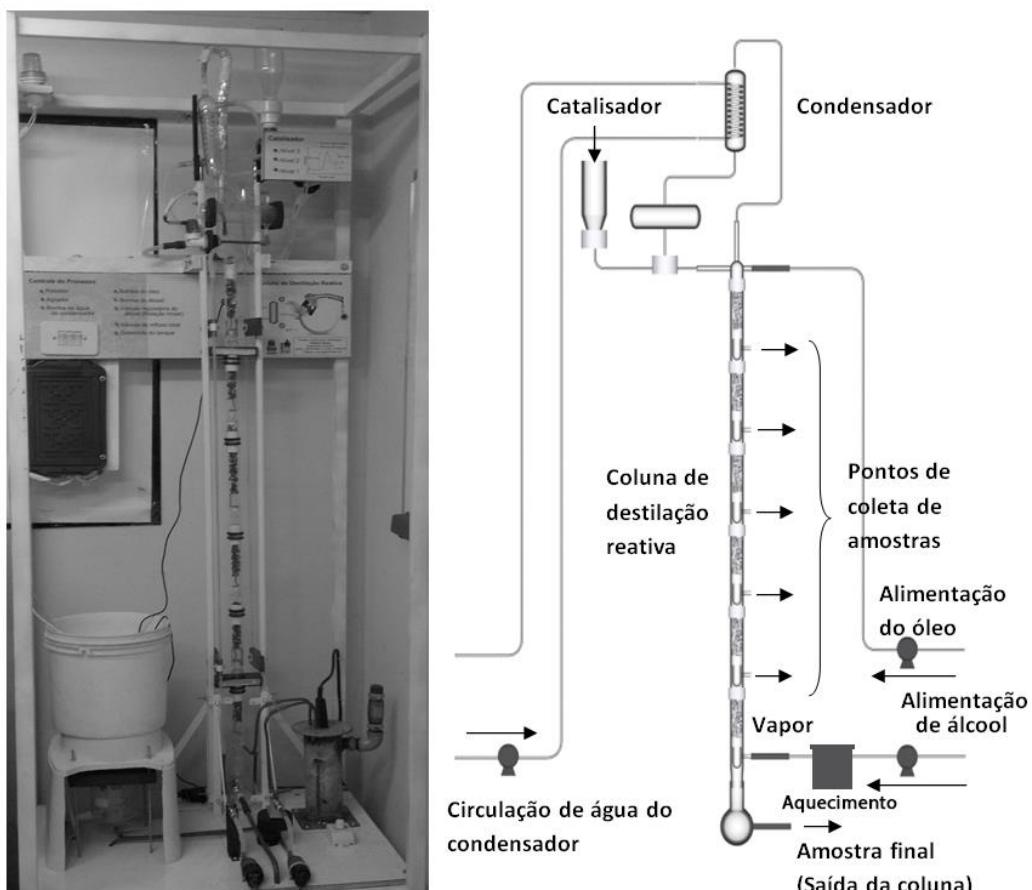
4 MATERIAIS E MÉTODOS

Para o desenvolvimento da pesquisa uma série de matérias e insumos foram necessários, para se atingir o objetivo necessário diversas metodologias foram utilizadas. Estes pontos são detalhados a seguir.

4.1 Coluna de destilação e automação

Inicialmente adaptou-se a coluna de destilação reativa presente no LSDPQ (Laboratório de simulação e dinâmica de processos), desenvolvida no próprio laboratório. Que é composta por uma coluna de vidro com 1,10m de altura e 4cm de diâmetro. Preenchida em 6 seções compostas por anéis de cobre de $\frac{1}{2}$ polegada, com 10-13cm cada seção, formando o recheio da coluna. Conforme a Figura 5.

Figura 5 - Configuração inicial da coluna de destilação reativa.

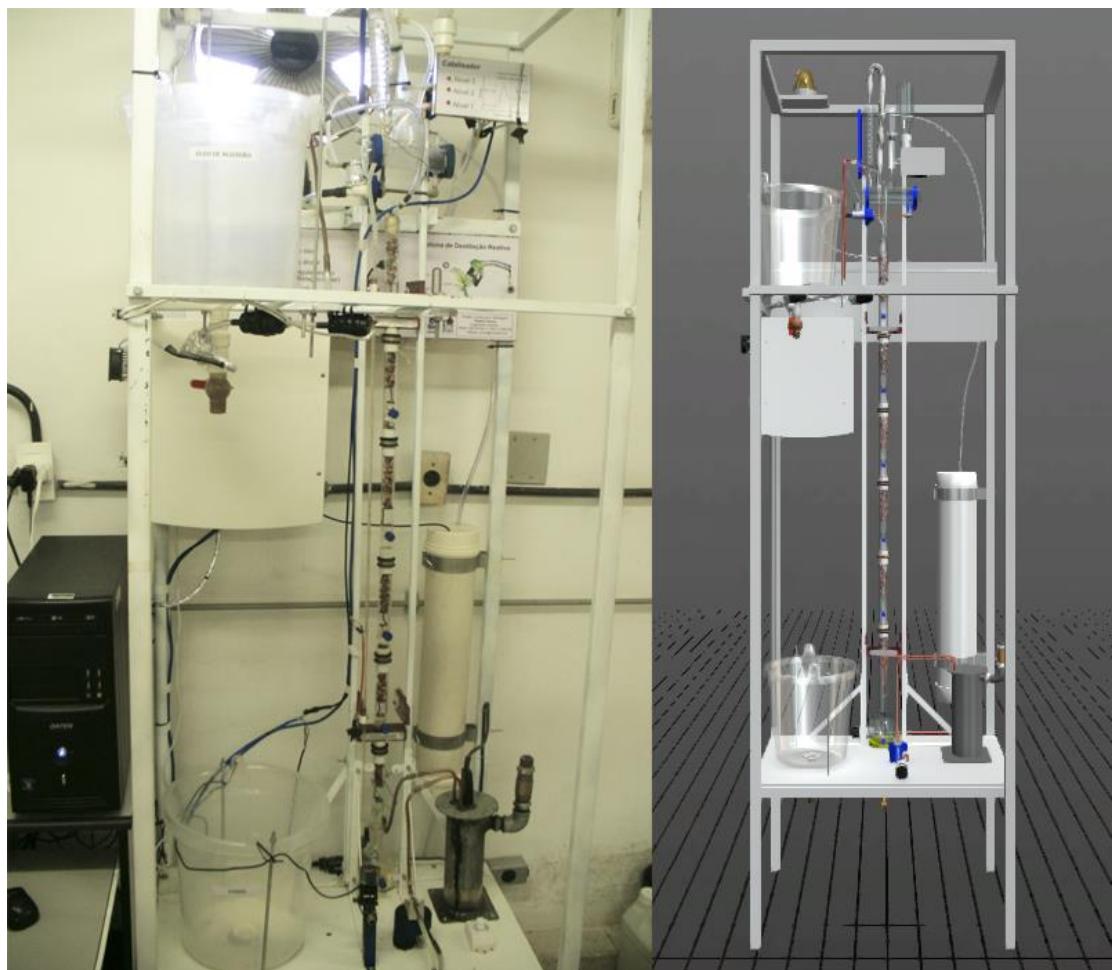


Fonte: autor.

O esquema geral e os fluxos materiais podem ser visualizados na Figura 5. A coluna contém um refervedor responsável pelo aquecimento do álcool alimentado em sua base, composto de uma resistência elétrica variável de até 1000W, operando com 220V. Além de um condensador e uma serpentina de resfriamento. Álcool é vaporizado no refervedor, que é segregado da base da coluna. Em seguida, o vapor de álcool sobe pela coluna, onde o mesmo transfere calor para a mistura reacional e é condensado no condensador. Após condensação, o mesmo é misturado no topo da coluna com o catalisador e o óleo. Essa mistura líquida desce ao longo da coluna, e entra em contato com o vapor de álcool que sobe. Por outro lado, a temperatura da mistura líquida que sai na base é recolhida da coluna para tratamento final de lavagem e análise. Inicialmente a coluna era praticamente desprovida de um sistema de automação/controle.

Após o desenvolvimento deste trabalho a coluna ganhou sensores de pressão, temperatura, vazão, assim como um controle mais eficaz dos solenoides de acionamento de catalisador e refluxo. O processo desenvolvido permitiu o controle preciso das vazões das bombas de óleo e álcool, assim como adicionou um sistema de resfriamento mais robusto. Utilizando materiais de baixo custo foi possível obter melhores condições de operação para a coluna. Os testes de saponificação foram realizados após as melhorias realizadas, A configuração final da coluna está exposta na Figura 6.

Figura 6 - Configuração final da coluna de destilação reativa, e representação 3D da coluna no supervisório.



Fonte: autor.

Um programa de computador foi desenvolvido para acionar e receber informações da coluna, com ele era possível programar a atuação de certos equipamentos, ligá-los e desligá-los, inclusive por voz, acompanhar o valor atual e o histórico dos sensores, visualizar uma representação tridimensional da coluna de destilação reativa, entre muitas outras características que tornaram a operação da coluna mais simples satisfatória e confiável. Para isso o software foi desenvolvido no Unity 3D[®] no qual é possível acessar os arquivos que compõem o projeto de desenvolvimento. O Unity 3D[®] permite o desenvolvimento de softwares em 3D para controles e visualização de processos, como os desenvolvidos pelo LSDPQ (SILVA et al., 2016, TSCHOEKE et al., 2016a, 2016b e LIMA et al., 2015). Este software permite desenvolver programas de computador que incluem visualizações 3D, ampliando o nível de realismo na representação computacional. A capacidade de

processamento atual dos computadores permite representações mais fiéis aos processos, do que as Figuras estáticas em 2D que comumente são utilizadas nos supervisórios. A utilização de supervisórios em 3D é algo pouco explorado academicamente até o momento, o que indica inovação. **Os detalhes da automação desenvolvida e os resultados obtidos disto serão expostos nos resultados e discussão.**

4.2 Síntese do biodiesel

O processo de produção de biodiesel via etanol é conhecido por ser mais difícil que utilizando metanol. Por tanto, a execução dos experimentos foi dividida em três partes.

4.2.1 Testes em batelada

Foram realizados testes em batelada para que houvesse um conhecimento básico do processo de produção de biodiesel via etílica, o qual apresenta características que o tornam mais difícil do que via metílica. Os cálculos estequiométricos envolvem uma série de passos extensos.

Os reagentes utilizados estão expostos no Apêndice A11. O óleo utilizado nos testes foi o óleo de algodão, proveniente da usina de biodiesel instalada no município de Caetés, interior do estado de Pernambuco, A composição do óleo de algodão segundo a usina e a literatura é exposta na Tabela 4 a seguir.

Tabela 4 - Composição mássica do óleo de algodão.

Ácido Hexadecanóico (Palmítico) %	Ácido Octadecanóico (Estearílico) %	Ácido Octadecenoíco (Oleico cis 9) %	Ácido Octadecadienóico (Linoleico cis-9 cis 12) %	Referência
28,2	0,9	13,1	57,6	Óleo utilizado
28,7	0,9	13,0	57,4	Goering et al. (1982)
28	1	13	58	Demirbas (2003)
28,03	0,89	13,27	57,51	Ramadhas et al. (2005)

É possível observar a semelhança entre as informações. Os dados físicos segundo (RAMADHAS et al., 2005) estão expostos na Tabela a seguir.

Tabela 5 - Características físicas do óleo de algodão segundo Ramadhas et al. (2005).

Propriedades físicas	Valor
Densidade relativa	0,912
Viscosidade (mm ² /s) a 40°C	50
Ponto de fulgor (°C)	210
Poder calorífico (MJ/kg)	39,6

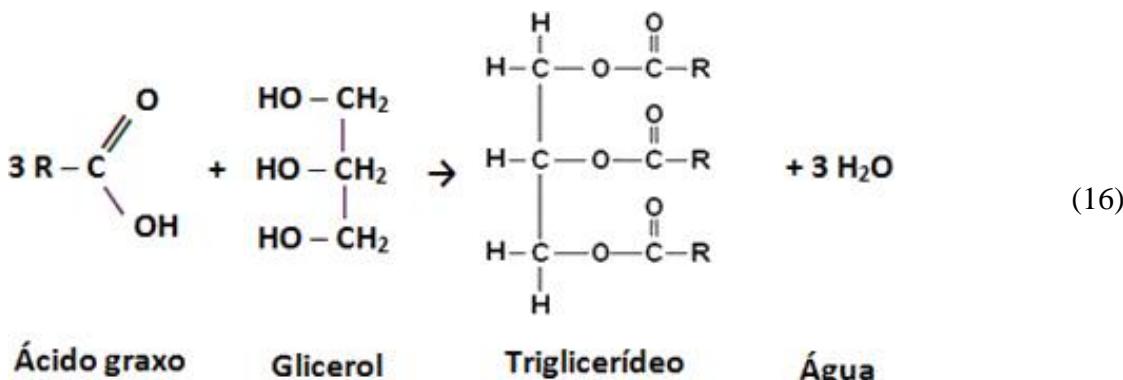
Tendo em vista essas informações e levando em consideração as massas molares dos ácidos em questão:

Tabela 6 - Massa molar dos ácidos que compõem o óleo de algodão.

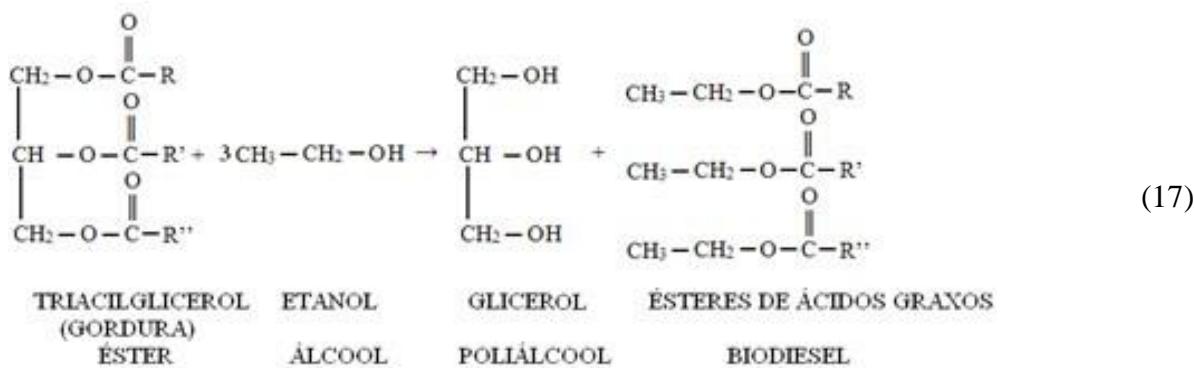
Ácido Hexadecanóico (Palmitílico) g/mol	Ácido Octadecanóico (Estearílico) g/mol	Ácido Octadecenóico (Oleico cis 9) g/mol	Ácido Octadecadienóico (Linoleico cis-9 cis 12) g/mol
256,4241	284,47724	282,4614	280,4455

A massa molar média do ácido foi calculada pela transformação em fração molar e se retirou a média ponderada, o que leva a 273,8 g/mol.

Triacilglicerol é o nome genérico de qualquer tri-éster oriundo da combinação do glicerol com ácidos, mais especialmente ácidos graxos (ácidos carboxílicos de longa cadeia), no qual as três hidroxilas (do glicerol) sofreram condensação carboxílica com os ácidos, os quais não precisam ser necessariamente iguais. De forma simplificada, um triacilglicerol é formado pela união de três ácidos graxos a uma molécula de glicerol, cujas três hidroxilas (grupos -OH) ligam-se aos radicais carboxílicos dos ácidos graxos como na equação química 16.



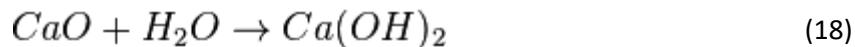
A massa molar média de um triacilglicerol proveniente de algodão pode ser dada seguindo os cálculos estequiométricos do apêndice G, e tem como valor **859,2 g/mol**. O método utilizado e o valor encontrado estão coerentes com o que é encontrado em outros trabalhos (CHAVES, 2008; FRAGA et al., 2014). Para a transesterificação se tem a Equação química 17:



Segundo os cálculos do apêndice **G** a massa molar media do éster é de 301,7588 g/mol. O método utilizado e o valor encontrado estão coerentes com o que é encontrado na literatura (DEMIRBAS, 2003).

Para definição da quantidade de reagente utilizado, em um caso específico para a conversão, uma série de fatores tem que ser levada em consideração, os mesmos estão expostos no apêndice G.

Tanto o álcool adicionado, quanto a mistura catalisador/álcool precisam ser secadas para eliminação da uma eventual fração de água, o CaO foi utilizado pois na presença de água como na Equação química 18:



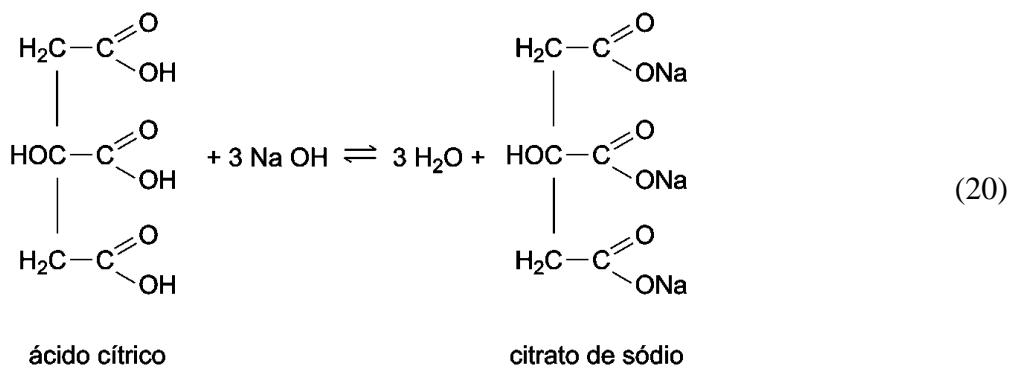
O hidróxido resultante da Equação 18 além de remover a água é insolúvel em etanol, uma simples decantação pode remover o álcool anidro do precipitado branco. Definindo f como fração em massa de água no álcool. Deve-se utilizar a seguinte quantidade de CaO, os cálculos que definem a quantia de CaO estão expostos no apêndice G, e resultam na seguinte equação:

$$X = \frac{3,1129 \cdot f \cdot M}{P} \quad (19)$$

Por exemplo, para 32g de álcool hidratado, com 10% em massa de água, e o CaO a 95% de pureza tem-se segundo a equação 19:

$$X = \frac{3,1129 \cdot 0,1 \cdot 32}{0,95} = 10,4855 \text{ g de CaO}$$

Após a reação o catalisador precisa ser neutralizado, é possível utilizar ácido cítrico monohidratado ($C_6H_8O_7 \cdot H_2O$ MM=192,123+18,015=210,138g/mol) segundo a reação da Equação 20:



A água de hidratação não participa da reação. A quantidade de ácido a ser utilizada pode ser definida seguindo a estquiometria descrita no apêndice **G**.

Ou também pode ser neutralizada utilizando ácido sulfúrico ($\text{MM}=98,078 \text{ g/mol}$) segundo a reação da Equação 21:



Atadashi et al. (2011) sugere um excesso maior de ácido para garantir a neutralização total, ao menos o dobro da quantidade estequiométrica para neutralização, ou que a concentração de ácido seja de 10% em volume caso o ácido seja fraco, como o H_3PO_4 .

Os cálculos estequiométricos são discutidos detalhadamente no apêndice **G**, devido ao fato de serem extensos, um script em MATLAB[®] foi desenvolvido para automatizar todos os cálculos referentes a estequimetria e preparo para a reação, o código está no apêndice **D**.

Foram realizadas reações de transesterificação em um recipiente sob agitação, e com temperatura medida por um arduino. Segundo os trabalhos de (REYERO et al., 2015 e SILVA et al., 2010) e utilizando também álcool etílico ‘PA’, após processo de secagem com óxido de cálcio (CaO). O catalisador utilizado foi o hidróxido de sódio (NaOH) diluído em etanol e seco com sílica gel. Diversas condições foram testadas, nas quais as temperaturas utilizadas estavam na faixa de 40 e 70°C, a quantidade de catalisador entre 0,5 e 1,3% mássico (catalisador/óleo) a razão molar entre 9 e 30 (etanol/óleo), e tempos de residência entre 1,5h e 5h. A montagem foi realizada conforme a Figura 7:

Figura 7 - Sistema para teste de transesterificação em batelada.



Fonte: autor.

A purificação do biodiesel produzido era executada em diversas etapas, seguindo trabalhos como o de (ATADASHI et al. 2011):

Neutralização/Lavagem ácida: após a reação o processo era interrompido com uma lavagem ácida seguindo as informações aqui citadas

Decantação convencional (2) ou Decantação por centrifugação (1): Uma decantação era realizada para que a fase pesada (água glicerina, álcool e sais) fossem separados do biodiesel e do óleo não reagido.

Decantação a quente: Agilizava o processo de decantação e removia resíduos de água no biodiesel produzido.

Lavagem Ácida: uma nova lavagem ácida era realizada.

Decantação (12 horas): uma decantação longa era realizada.

Decantação a quente (74°C por 20 min): uma nova decantação a quente era realizada.

Fervura (118°C por 2h): uma fervura era realizada para remover qualquer água residual.

Secagem com sílica gel: Para garantir a isenção de água, o biodiesel produzido era armazenado em recipientes contendo sílica gel.

4.2.2 Ensaio de transesterificação do óleo de algodão na coluna de DR

Para verificar se todas as modificações feitas na coluna foram efetivas, testes foram realizados. Seguindo o procedimento operacional do apêndice E, foi possível testar a eficácia das melhorias efetuadas. As condições utilizadas foram: vazão de óleo de 0,05 L/min, vazão de álcool de 0,1 L/min, razão molar resultante (álcool/óleo) de 38,14 mol/mol, concentração de catalisador de 1% mássico com relação a vazão de óleo, vazão de catalisador de 18,2 mL/min, razão de refluxo de 0, carga térmica de 500 W (em média) e pressão de 103,20 kPa. Seguindo o procedimento operacional, os sensores e os sistemas de atuação são testados, assim como o abastecimento de óleo, álcool e catalisador são realizados, uma vez que as condições de operação são satisfatórias o experimento é iniciado, após atingir o estado estacionário pela indicação do supervisório, as amostras são coletadas e analisadas para se obter o teor de sabão e catalisador na corrente de fundo, foi utilizada a norma AOCS.

Posteriormente, após purificação do produto da coluna, uma nova análise é realizada para se avaliar o teor de éster no biodiesel produzido. Para isso utilizou-se a norma EN 14103. Por fim, todos os equipamentos eram limpos e lavados com etanol após a operação. A coluna foi operada digitalmente utilizando computadores com o supervisório desenvolvido, desta forma, a mesma foi acompanhada por dois computadores ao mesmo tempo, incluindo a assistente virtual, EVA. Os resultados foram obtidos e ajustes foram feitos em função dos mesmos, para que o planejamento fatorial completo fosse executado.

4.2.3 Planejamento fatorial para análise da saponificação na coluna de DR

Um planejamento fatorial foi executado com o intuito de verificar que parâmetros minimizam a produção de sabão como sub-produto. Para isso condições e procedimentos semelhantes aos testes com a coluna foram aplicados, os detalhes serão expostos em resultados e discussão. Variou-se: O refluxo de 0 e 100%, carga térmica de 200W a 400W e razão molar entre 15/1 e 25/1 (álcool/óleo), a partir disto, diversos diagramas de Pareto bem como os gráficos de superfície foram gerados no Statistica[®], da mesma forma que diversos trabalhos na literatura (NOSHADI, AMIN, PARNAS. 2012; PISARENKO et al., 2011), com eles foi possível tirar diversas conclusões sobre a influência dos parâmetros operacionais sobre a conversão do óleo e sobre a concentração de sabão, utilizando os métodos de análise expostos a seguir.

4.3 Métodos de análise dos produtos

O biodiesel produzido e purificado como exposto anteriormente foi analisado com relação a teor de éster, catalisador residual e teor de sabão.

A EN 14103 é uma norma europeia utilizada para a quantificação do teor de éster. Esta foi desenvolvida para análise de biodiesel proveniente de oleaginosas, as quais apresentam em sua composição ácidos graxos com cadeias carbônicas predominantemente entre C14:0 e C24:1. E pode ser utilizada para quantificação de ésteres etílicos segundo Gasparini (2010).

Para encontrar o teor de éster foi utilizado um cromatógrafo gasoso modelo GC-17A da SHIMADZU com injeção automática, coluna Carbowax, utilizando como padrão interno o

C17 (heptadecanoato de metila), para atender os ensaios EN 14103, EN 14105 e EN 14110 estabelecidos na resolução ANP nº 07/08.

Para quantificação do teor de catalisador residual e o teor de sabão foi utilizada a norma AOCS métodos cc 17-79, a qual é um simples processo de titulação, o processo detalhado é exposto a seguir:

1. Coletar uma amostra cuja quantidade vai depender do nível esperado de catalisador e sabão. 0,5mL de glicerina bruta, ou 5 mL de biodiesel não lavado, ou 100 mL de biodiesel lavado. Dissolva em 100mL de solvente hidratado (2% de água), como solvente pode-se utilizar acetona ou o próprio etanol.
2. Adicione 2mL do indicador fenolftaleína a 1% (dissolvido em álcool)
3. Titule com HCl a 0,1mol/L até a fenolftaleína mudar de vermelho para incolor. Isto indica que o catalisador residual foi neutralizado na amostra, esta quantia de HCl utilizada será designada como “A”.
4. Adicione 1mL do indicador azul de bromofenol a 0,4% (dissolvido em água). Este indicador muda de cor próximo de pH 4,5.
5. Titule até o bromofenol mudar de azul para amarelo, isto indica que o pH é baixo o suficiente para que todo o sabão seja transformado em ácido graxo e sal. O ácido forte adicionado (HCl) é atraído pelo íon metálico do sabão, mais do que para a cadeia ácido graxo. Então o metal se combina com Cl para produzir NaCl. Define-se o volume de HCl utilizado na segunda titulação como “B”.

Agora a quantidade de HCl adicionada durante a primeira titulação, nos diz o quanto de catalisador ainda existe na amostra, e a segunda titulação nos diz o quanto de sabão estava presente.

Tomando a quantia “A” como o volume de HCl adicionado na primeira titulação e seguindo os cálculos: [“A” mL de HCl a 0,1mol/L] x [1 litro/ 1000mL] x [0,1 mol/L de HCl] x [1 mol de NaOH/mol de HCl] x [40,0 g/mol de NaOH]/[mL da amostra] = gramas de NaOH/mL de amostra.

$$X = \frac{A \cdot 0,1 \cdot 40,0}{1000 \cdot V_a} = \text{gramas de NaOH/mL de amostra}$$

Isto dá a quantia de catalisador livre na amostra, assumindo que o catalisador é NaOH, substituir as massas molares de outros catalisadores é possível.

Utilizando o volume “B” de HCl na segunda titulação deve-se seguir os cálculos a seguir: [“B” mL de HCl a 0,1mol/L] x [1 litro/ 1000mL] x [0,1 mol/L de HCl] x [1 mol de sabão/mol de HCl] x [304,4 g/mol de sabão]/[mL da amostra] = gramas de sabão/mL de amostra.

$$X = \frac{B \cdot 0,1 \cdot 304,4}{1000 \cdot V_a} = \text{gramas de sabão/mL de amostra}$$

Às vezes a quantia obtida pode chegar a mg/kg, então o número deve ser multiplicado por um milhão, os cálculos assumem o sabão como oleato de sódio, para oleato de potássio, utilizou-se a massa molar de 320,56 g/mol.

Quando a quantidade de catalisador e sabão é pequena, o ideal é utilizar amostras em “branco”, sem nenhuma amostra, deve-se titular o branco e descontar a quantia da quantidade obtida com amostras.

5 RESULTADOS E DISCUSSÃO

Com a metodologia descrita os resultados obtidos puderam ser discutidos e comparados com a literatura.

5.1 Testes em batelada

Diversos ensaios foram feitos para se observar as características do processo de produção de biodiesel via etílica em batelada. Diversas condições experimentais de produção e purificação foram testadas com o intuito de se preparar para as reações na coluna de destilação reativa.

Diversos artigos na literatura citam condições ótimas de produção de biodiesel utilizando etanol: Reyero et al. (2015) encontraram a condição ideal utilizando razão molar etanol/óleo entre 12 e 24 com concentração de catalisador de 0.3% em massa, com o maior tempo de reação possível, explicando também que para reduzir a quantidade de sabão formado é necessário utilizar uma razão molar alta. Silva et al. (2010) apresentaram um planejamento experimental que foi usado para otimizar a concentração do catalisador (0,5-1,5% em massa) e relação molar de etanol/óleo (3:1 a 9:1). As condições ideais encontradas por Silva et al. (2011) para a produção de ésteres etílicos foram as seguintes: temperatura de 56,7°C, tempo de reação em 80 minutos, razão molar álcool/óleo de 9:1 e concentração de catalisador de 1.3% mássico em relação ao óleo. Citando também que mais catalisador deve ser adicionado para neutralizar a acidez do óleo.

Tendo em vista os resultados obtidos em artigos foi possível definir as condições experimentais para um teste inicial. A reação ocorreu em um recipiente agitado onde a temperatura foi mantida entre 60°C e 70°C, com 0,5% de hidróxido de sódio em massa de catalisador com relação ao óleo, todos os reagentes estão detalhadamente expostos no anexo A. Com o intuito de atingir a maior conversão possível do óleo, o tempo de reação foi de uma hora e meia e razão molar álcool/óleo de 9:1. Após a reação houve uma série de etapas de purificação e lavagens como descrito no tópico 4.2.1 deste trabalho.

O resultado obtido não foi satisfatório, o teor de éster foi de 23%. Resultado obtido por cromatografia como descrito no tópico 4.3 deste trabalho, muito distante das conversões acima de 98% de Silva et al. (2011) e de Reyero et al. (2015). Diversas hipóteses foram feitas e testadas para tentar explicar o ocorrido.

Como descrito na literatura a reação de transesterificação etanólica é conhecidamente mais lenta que por via metanólica como citado por Leung et al. (2010) e Reyero et al. (2015). Conclui-se inicialmente que as condições fossem muito brandas para que a reação com etanol ocorresse consideravelmente, seguindo as recomendações do trabalho de Silva et al. (2011) o novo teste foi realizado utilizando uma temperatura de 60 ° C, tempo de reação em 120 minutos, razão molar álcool/óleo de 12:1 e concentração de catalisador de 1.0% mássico em relação ao óleo.

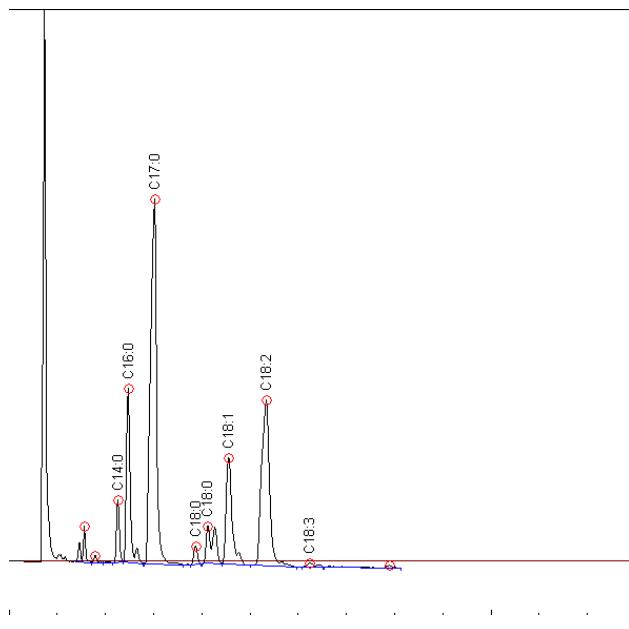
O resultado encontrado novamente não foi satisfatório com conversão do óleo de 41%, chegando a conclusão de que deveria haver um controle maior da matéria prima utilizada. O álcool etílico assim como o etóxido de sódio (catalisador) adicionado foi seco com um maior excesso de CaO e sílica gel. Pois quando isto não era feito a simples adição do catalisador misturado com o etanol, e adicionado ao óleo formava uma quantidade considerável de sabão visível pelo turvamento e a formação de “blocos” de sabão na mistura, inutilizando a reação. Isto se deve a quantidade considerável de catalisador adicionado que passou para 1.0%, ficando claro que a adição deveria ser lenta e gradativa, para que não houvesse uma área de maior concentração de íons OH⁻ que facilitasse a formação de sabão na região de adição. O biodiesel produzido desde então era conservado em sílica gel para que qualquer resíduo de água fosse removido e não alterasse consideravelmente o teor de éster. Com estas mudanças um novo experimento foi realizado nas mesmas condições anteriores, mas com a matéria prima totalmente seca.

O resultado novamente não foi satisfatório cerca de 43%, No trabalho de Silva et al. (2011) foi verificado que a quantidade de catalisador adicionado poderia ser aumentada para 1.3% e que a acidez do óleo deveria ser medida com mais rigor, pois catalisador extra deveria ser adicionado para neutralizar a acidez. A acidez encontrada foi de 1,15% seguindo o método do apêndice C valor relativamente alto, mas dentro do que permitido segundo Leung et al. (2010). Um novo teste foi realizado utilizando uma temperatura de 60 ° C, tempo de reação em 120 minutos, razão molar a 15:1 e concentração de catalisador de 1.3% mássico em relação ao óleo, mas uma quantidade suficiente para neutralizar a acidez do óleo. A adição de mais hidróxido de sódio favorece a reação de saponificação. As quantias foram calculadas pelo programa desenvolvido pelo autor e disponível no apêndice D.

O resultado obtido foi um teor de éster de 53%, mas ainda distante do obtido por Silva et al. (2011) e Reyero et al. (2015), mesmo utilizando condições mais severas não foi possível

reproduzir o resultado destes autores. Os resultados indicaram que o método de análise poderia não detectar um biodiesel de cadeia carbônica mais longa, como os que são produzidos via etanol. O laboratório responsável pelas análises o LAC (Laboratório de Combustíveis do Departamento de Engenharia Química da UFPE) tem um procedimento padrão para análise de biodiesel, normalmente metílico, porém com a adição de mais um carbono no biodiesel etílico o processo pode levar a um resultado incorreto. Certos picos mais afastados por terem uma cadeia longa não estavam sendo considerados levando a um teor de éster errôneo, o cromatograma resultante pode ser observado na Figura 8:

Figura 8 - Cromatograma resultante da análise do biodiesel etílico.



Fonte: autor

Com esta correção uma nova análise foi feita levando ao valor de 64%, resultado ainda em desacordo com a literatura. Desta forma, optou-se por fazer o processo de transesterificação em mais de uma etapa. Partindo-se do biodiesel formado, a reação foi processada levando em consideração apenas o óleo residual presente no biodiesel, como pode ser visto no apêndice D. O resultado obtido foi de 69%, a concentração teve um aumento de 5%. O biodiesel assim formado, passou pelo processo de purificação mais uma vez, reduzindo seus teores de glicerina e de impurezas. Repetindo todo o processo de reação e purificação mais 2 vezes foi possível obter biodiesel a 93%, pois ao final deste processo o mesmo já havia sido purificado 4 vezes. De acordo com a literatura foi possível chegar à seguinte conclusão: **o método de purificação estava incorreto**. Segundo os trabalhos de Kreul et al. (1999) e

Atadashi et al. (2011) são necessárias múltiplas lavagens para purificação, até que não haja uma mudança significativa no pH da água residual, apenas duas lavagens estavam sendo feitas. Desta forma tentou-se novamente reproduzir uma condição próxima a do trabalho de Silva et al. (2011) onde foi utilizada a temperatura de 60°C, tempo de reação em 90 minutos, razão molar a 10:1 e concentração de catalisador de 1.3% mássico em relação ao óleo.

O resultado obtido foi de 91,5% em apenas uma batelada partindo unicamente de óleo de algodão, etanol e NaOH. O resultado encontrado está coerente com o que foi obtido por Anastopoulos et al. (2009) que não conseguiu obter biodiesel especificado (teor de éster superior a 97,5%) em menos de duas bateladas consecutivas, o que **reforça o uso de um processo de multi estágios**, como a coluna de destilação reativa. O resultado obtido não foi idêntico ao de Silva et al. (2011), que é contraditório ao resultado de Anastopoulos et al. (2009), porém foi muito superior ao do primeiro teste, que foi de 23%.

Durante a purificação seguindo o tópico 4.2.1 deste trabalho, muitos detalhes puderam ser observados. Durante a lavagem do biodiesel a água adicionada deve ser acidificada para neutralizar o catalisador residual e eliminar álcool e glicerina do biodiesel. Porém a quantidade adicionada varia de autor para autor (KREUL et al., 1999; ATADASHI et al., 2011) durante os processos de purificação diversos teores de ácido na lavagem foram testados, desde água destilada “neutra” (pH 6,5) até água ácida (pH 2,0). Utilizando a água neutra percebeu-se a formação intensa de sabão, chegando a inutilizar a amostra de biodiesel, se mostrando uma forma inadequada de purificação em nossos testes, pois a água atua como catalisador na formação de sabão segundo as reações das Equações 11, 12 e 13, conforme os autores Leung et al. (2010) e Eze et al. (2014). A adição de água com pH 4,5 levou a uma lavagem com pouca formação de sabão, e uma decantação lenta, mas o biodiesel resultante turvava e tendia a formar uma “borra” de sedimentos após semanas. O valor de pH de 4,5 está de acordo com o pH do biodiesel puro segundo a norma AOCS. Por outro lado, a lavagem com pH 2,0 se mostrou mais favorável de todas, **pois formava menos sabão, além de necessitar de um menor tempo de decantação. Este tempo foi reduzido pela metade**, quando comparado com a lavagem num pH 4,5. **O biodiesel resultante não formava borra mesmo após meses** de armazenamento, se mostrando estável. O que indica uma condição ideal, se não fossem as especificações da ANP. A resolução ANP Nº 45 de 25/08/2014 define que o índice de acidez máximo permitido é de 0,50 mg KOH/g de amostra. O biodiesel resultante da purificação com água a pH 2,0 tinha valor 2 vezes superior ao limite da

especificação. De forma geral para atender a especificação ficou definido que a água de lavagem deveria ter pH inferior a 4,5 nos experimentos posteriores, mas não atingir o pH 3,0. Pois com isso se obtém uma rápida decantação, e um biodiesel estável dentro das especificações.

Uma das limitações no processo de purificação do biodiesel é o tempo de decantação na purificação, que é maior quando se produz biodiesel etílico. Foi observado durante a fervura do biodiesel para eliminação de água, que a baixas temperaturas (50°C) havia a formação de gotas no fundo do recipiente, referentes a excesso de glicerina e de água ainda presentes. Esse fato pode ser explicado devido a **diminuição da viscosidade da mistura durante o aquecimento, que favoreceu a decantação**. Ficou claro que uma das limitações no processo de purificação do biodiesel etílico pode ser resolvida utilizando uma decantação a quente, método utilizado neste trabalho. Dentre estas diversas características da transesterificação etílica a Tabela 7 a seguir resume o que se concluiu.

Tabela 7: Resultados obtidos.

Situação	Condições
Condições de transesterificação mais severas quando etanol é utilizado	Catalisador >1,0% em massa com relação ao óleo
Quantificação do índice de acidez	Menor que 2,0% para permitir transesterificação
Os reagentes não podem conter traços de água	Secagens com CaO ou sílica gel devem ser feitas
A mistura do catalisador deve ser gradativa	Para não criar zonas de alta basicidade, favorecendo a saponificação
O método cromatográfico deve ser adequado	Permitindo a caracterização de ésteres de cadeia longa
O método de purificação deve conter muitas lavagens	Mais que 4 lavagens devem ser feitas
O pH da água de lavagem deve ser adequado	O pH deve estar entre 4,5 e 3,0.
A decantação a quente reduz o tempo de decantação	Temperatura acima de 50°C

Os resultados da Tabela 7 foram utilizados para se prevenir de futuros problemas no processamento das amostras e dos reagentes para operação da coluna de destilação reativa.

Com essas informações obtidas foi possível se precaver e prever certos gargalos durante a pesquisa.

5.2 Custos e automação

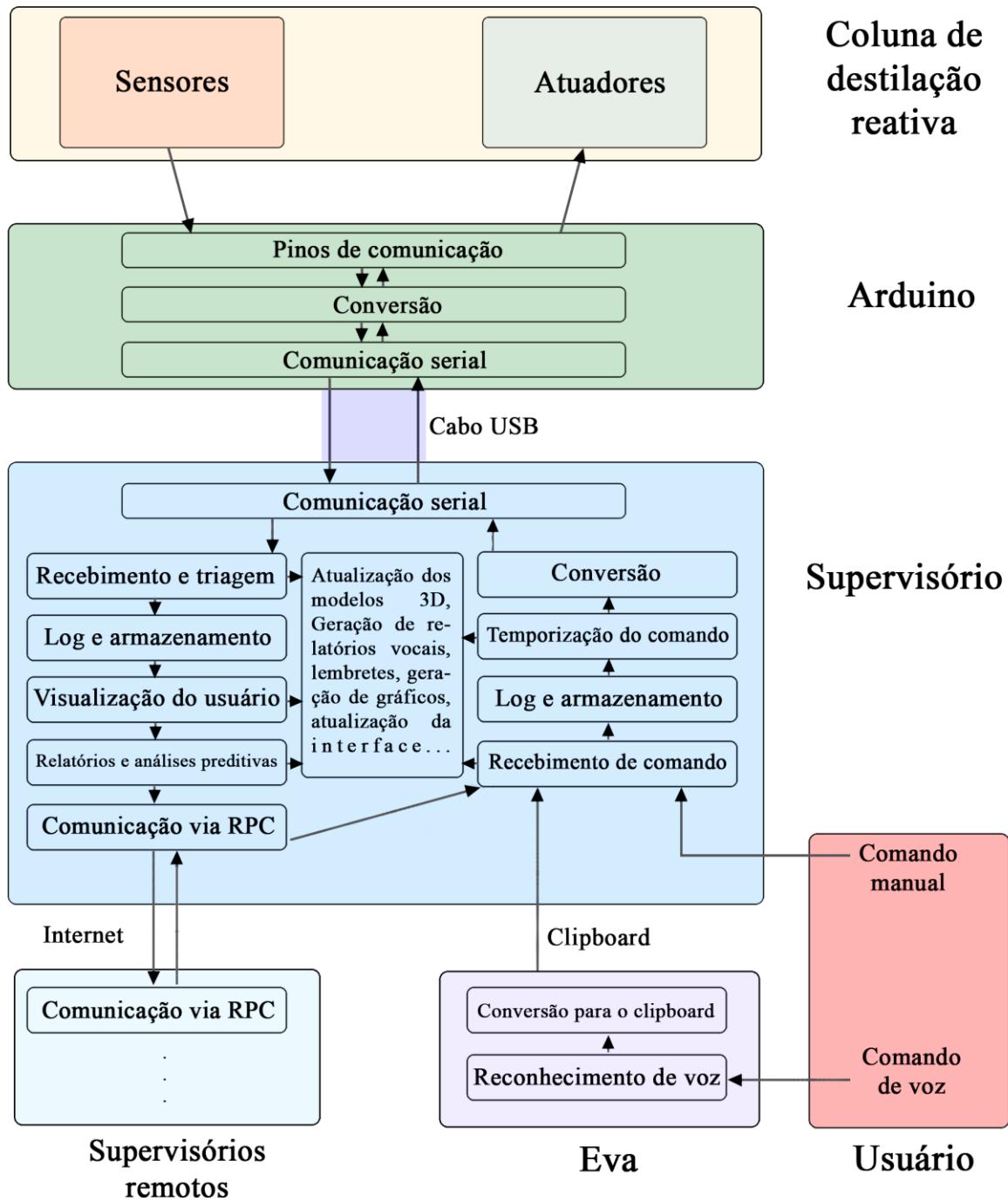
A automação foi iniciada com uma coluna de DR sem o sistema supervisório em condições limitadas de operação conforme Figura 5.

A coluna dispõe de uma alimentação de etanol vaporizado em sua base, o aquecimento é feito a partir de um refervedor. O óleo é alimentado no topo da coluna e o vapor de álcool que sobe ao topo da coluna é condensado. O catalisador é adicionado junto com o álcool condensado, na forma de etóxido de sódio diluído em álcool. A coluna é composta por um recheio de anéis metálicos de $\frac{1}{2}$ polegada de diâmetro, são seis estágios ao todo, onde há pontos laterais de amostragem, ao todo 6.

Para o processo de automação foi utilizada uma plataforma comum em projetos de automação em pequena escala, e que vem sendo utilizado em trabalhos científicos (Kubínova e Slegr 2015; Silva et al. 2016; Tschoeke et al. 2016a e 2016b), o **Arduino**. Que é uma plataforma de prototipagem eletrônica de hardware livre, projetada com um micro controlador Atmel AVR com suporte de entrada/saída embutido, e usa linguagem de programação que é essencialmente C/C++. Devido à facilidade e a grande disponibilidade de informação sobre o arduino, e possibilidade de comunicação com um computador, optou-se pelo mesmo. A arquitetura desenvolvida é representada pelo fluxograma da Figura 9.

Na Figura 9 é possível observar como todos os componentes desenvolvidos se intercomunicam. A coluna tem diversos sensores e atuadores que estão ligados ao arduino, de forma detalhada na figura 10. O arduino faz a comunicação entre os sensores e o computador, via USB, e para isso necessita de diversas lógicas de conversão, expostas no apêndice A, O programa supervisório instalado no computador, é responsável por fazer a intercomunicação e o gerenciamento de todas as informações, detalhadas no apêndice B. Informações tanto recebidas pelo usuário, quanto por outros supervisórios e pela coluna, além disso outro processo denominado EVA (*Extensible Virtual Assistant*) faz o reconhecimento da voz do usuário para recebimento de comandos. Na figura 9 a seguir é possível compreender melhor a comunicação entre o hardware e o software.

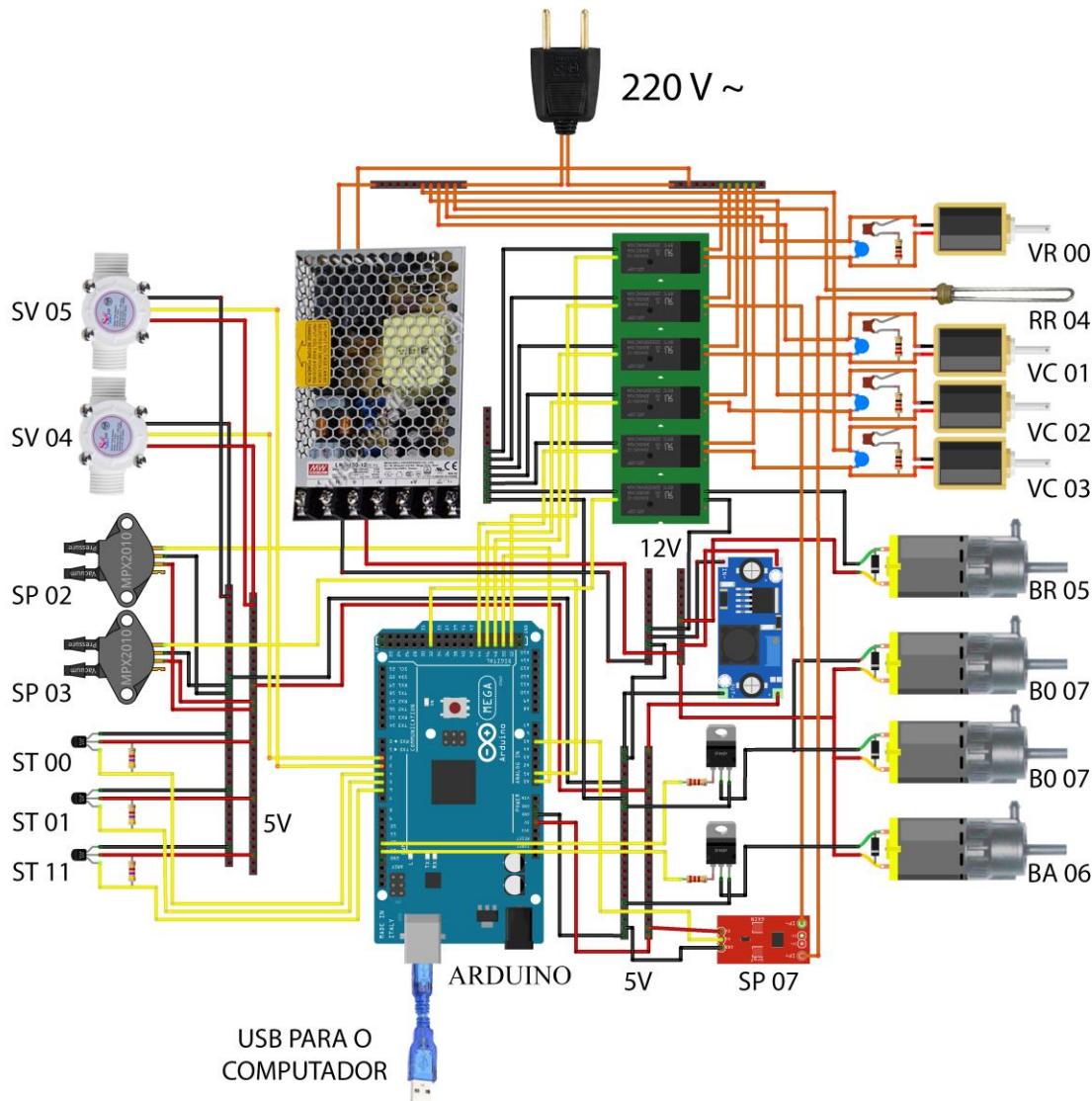
Figura 9 - Arquitetura desenvolvida.



Fonte: autor.

O Arduino permite a conexão de sensores e atuadores no mesmo, desempenhando um papel semelhante a um PLC (Controlador lógico programável), mas com apenas uma fração do custo. O esquema dos circuitos elétricos está exposto na Figura 10:

Figura 10 - Esquema elétrico desenvolvido para a coluna.



Fonte: autor.

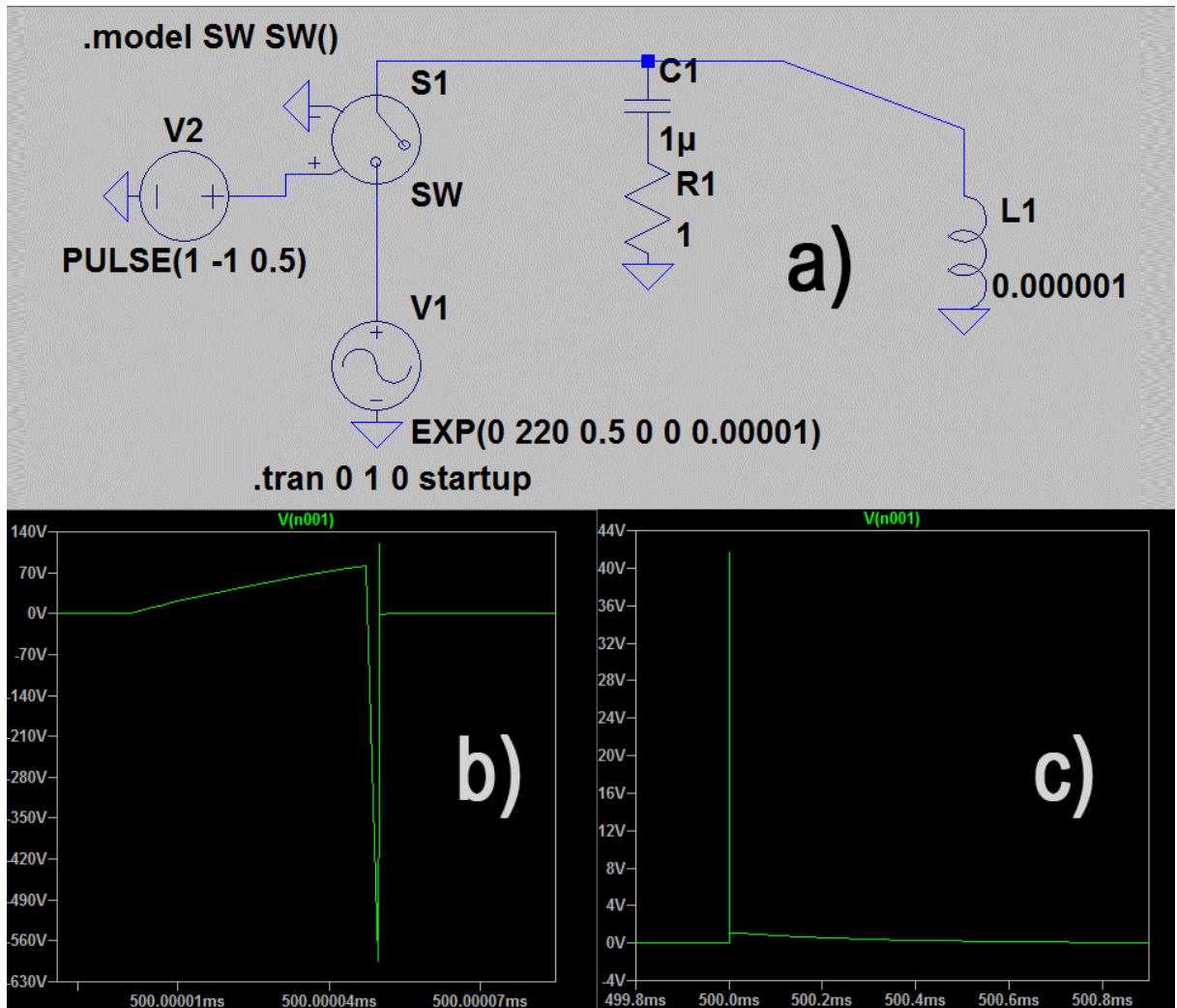
Na Figura 10 é possível observar o complexo circuito de sensores e atuadores montados. O sistema conta com três termômetros, que medem a temperatura da base e do topo da coluna, assim como a temperatura do tanque de resfriamento do condensador, representados na Figura anterior por ST00, ST01 e ST 11 respectivamente. Existem dois medidores de pressão, para o topo e para base da coluna (SP02 e SP03). Foram adicionados dois medidores de vazão para o óleo e para o álcool (SV04 e SV05), os quais foram calibrados para estes líquidos. A partir dessas vazões e das relações estequiométricas é

possível achar a razão molar álcool/óleo, fator importante no processo. A potência utilizada no refervedor pode ser medida a partir de um medidor de corrente (SP07), colocado de forma invasiva no circuito de atuação da resistência do refervedor, como mostra a Figura 10.

Com relação aos atuadores têm-se quatro válvulas solenoides, onde uma serve para permitir ou parar o refluxo de álcool da coluna, outras três servem para controlar a quantidade de catalisador adicionada (VR00, VC01, VC02 e VC03). É possível controlar também a resistência do refervedor (RR04), e alternar entre ligada e desligada, em ciclos predefinidos, como será apresentado posteriormente. O sistema conta ainda com quatro bombas (BR05, BA06, BO07 e BO07), provenientes de esguicho de para brisa de carro, onde duas delas servem para bombear óleo (BO), devido a sua viscosidade, outra serve para bombear álcool (BA), é possível controlar suas intensidades, utilizando o PWM (*pulse width modulation*), onde é possível modular ciclos de trabalho da ordem de microssegundos, e sua atuação sobre um transistor permite a amplificação dessa modulação para que ela atinja os níveis de potência exigidos pelas bombas. A bomba de resfriamento (BR05) serve para resfriar o condensador do topo da coluna.

Diversos problemas ocorreram durante a montagem do circuito da Figura 10. Um deles foi a interferência momentânea causada nos sensores quando as válvulas solenoides eram ligadas e desligadas, o problema era causado pelo colapso do campo eletromagnético quando a válvula é abruptamente desligada. Diversas alternativas foram testadas, uma delas foi a blindagem eletromagnética dos cabos e dos solenoides, que não se mostrou eficaz o suficiente devido a impossibilidade de se aterrar o equipamento. Simulações posteriores utilizando o LTSpice[©], um software de simulação para circuitos elétricos, demonstraram que o pico de tensão nos solenoides quando desligadas chegavam a 630V, como na Figura 11:

Figura 11 - a) Esquema simulado no LTSpice[®] b) pico de tensão de 630V ao desligar uma solenoide c) pico de tensão de 43V após a adição do *Snubber*.



Fonte: autor.

As simulações no LTSpice[®] confirmaram a questão de que os solenoides estavam causando um pulso eletromagnético que interferia fortemente com os sensores e cabos, a própria conexão com o arduino era interrompida devido a isso. Porém segundo o trabalho de Dubilier (2011) a solução é o uso de um “supressor de voltagem transiente”, diversos foram testados, mas o que teve o melhor desempenho foi o “*Snubber* capacitivo, reforçado com varistor”, o dimensionamento foi seguido segundo o trabalho de Dubilier (2011), onde as especificações de resistências, capacitores e varistores deveriam ser feitas. Quando a dissipação energética não é crítica, um passo a passo simples pode ser seguido:

Utiliza-se um resistor cerâmico de 2 watts, onde foi escolhido um de 5 watts por segurança. O valor da resistência tem de ser escolhido para manter o fluxo da corrente após o desligamento, e deve ser dimensionada segundo a Equação 10:

$$R \leq \frac{V_0}{I} = \frac{220}{8,0} = 27,5 \Omega \quad (22)$$

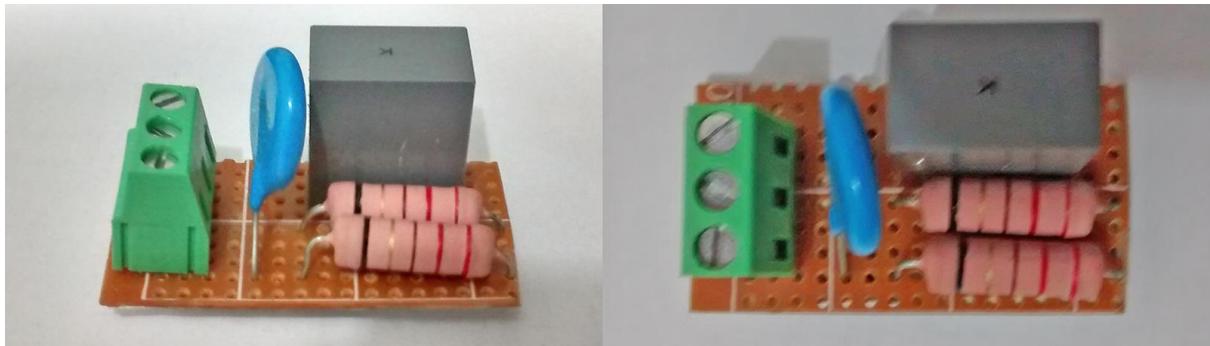
Na Equação 22 V_0 é a voltagem alternada, no caso 220V e I é a corrente nominal de operação, no caso de 8A, que é a corrente de operação de um solenoide, que resulta na utilização de uma resistência menor ou igual a $27,5 \Omega$, foi utilizada uma resistência de 25Ω . Já que 25Ω é menor que $27,5 \Omega$ pelo dimensionamento é tido como adequado. Combinações que gerassem uma resistência equivalente também são aplicáveis.

Para dimensionar o capacitor que dissipe um watt por duas vezes a transição da frequência por segundo, utilizando a equação 23:

$$C_s = \frac{1}{V_0^2 f_s} = \frac{1}{(220)^2 \cdot 60} = 344 \text{ nF} \quad (23)$$

Na equação 23 V_0 é a voltagem alternada, 220V, e f_s é a frequência de oscilação da tensão alternada, 60Hz, são vendidos capacitores de 334 nF, suficientemente próximo do desejado. Um varistor foi adicionado para aumentar a segurança foi escolhido um com voltagem de corte de 250V, acima da qual ele reduz a sua resistência permitindo a passagem de pico de corrente. Após o dimensionamento o *snubber* foi montado como na Figura 12:

Figura 12 - Montagem final do sistema de proteção contra surtos de corrente e tensão, *snubber*.

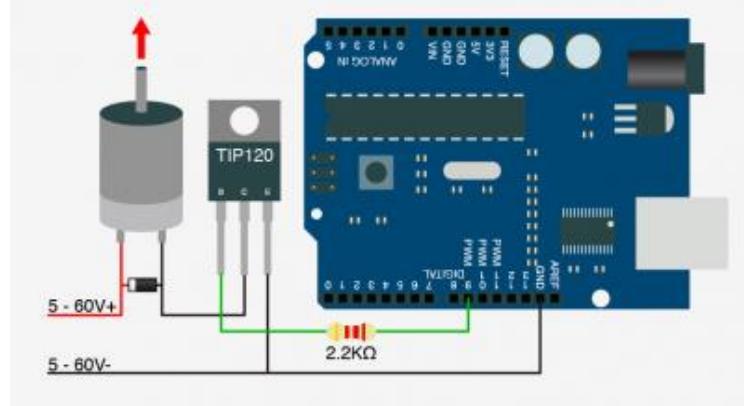


Fonte: autor

Para funcionar o *Snubber* tem de ser adicionado em paralelo com a carga indutiva a se proteger, os mesmos foram adicionados em todos os solenoides, como pode ser observado na Figura 10. O desempenho foi satisfatório como é possível observar na Figura 11, os picos de tensão foram reduzidos de 630V para 43V segundo o simulador, e na aplicação prática se verificou raras interferências no sistema elétrico da coluna, cerca de 10% dos casos ainda apresentam interferências, mas com uma intensidade reduzida em comparação ao que ocorria anteriormente.

Outro problema encontrado na montagem da automação foi de controlar as bombas utilizando um arduino, pois as bombas utilizam 12V e correntes de até 5A, uma potência que não é disponível diretamente pelo arduino. Optou-se por utilizar um transistor no qual teria a função de controlar uma corrente maior (bomba) em função de uma corrente menor (arduino) como exposto no trabalho High-power control... (2011), a Figura 13 exemplifica a aplicação:

Figura 13 - Sistema para controle de bombas e motores utilizando o transistor TIP120



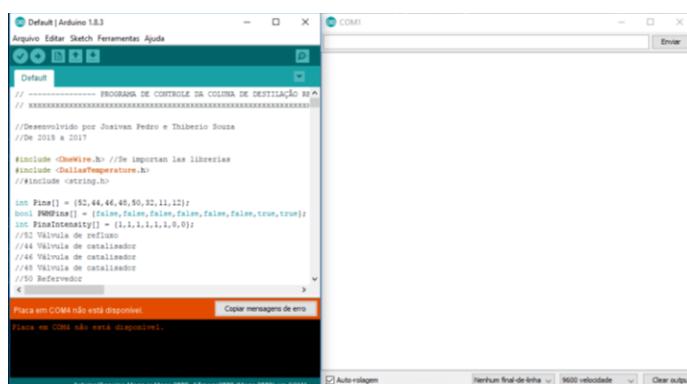
Fonte: <http://bildr.org/2011/03/high-power-control-with-arduino-and-tip120/>

Segundo o trabalho High-power control... (2011) o transistor TIP120 tem características adequadas para controlar sistemas de demanda de altas correntes e voltagens contínuas, utilizando micro controladores. Com limite de voltagem de 60V e de correntes de até 5A este transistor atende as necessidades do projeto. Um ponto negativo no mesmo foi seu aquecimento durante a operação, dissipadores de calor tiveram que ser adicionados. Principalmente para as bombas de óleo, que devido a viscosidade do fluido consumiam uma potência alta, fazendo o transistor superar os 100°C. Neste caso, além do dissipador foi adicionado um mini ventilador (*cooler*) para mantê-lo resfriado. Com isso foi possível controlar as bombas utilizando o PWM do arduino, onde ciclos de trabalho curtos (ms) são definidos para controlar a intensidade, com isso foi possível desenvolver o sistema de controle das bombas, como observado na Figura 10. A partir disto as bombas poderiam ser controladas em 256 níveis diferentes, levando a um controle de vazão entre 0% e 100%.

Estes foram alguns dos problemas encontrados na montagem da parte elétrica. Todos eles demandando pesquisas e testes, para que a coluna de destilação atingisse um nível satisfatório de operação, controle e segurança. Pois os trabalhos posteriores demonstraram que o processo produtivo se tornou mais dinâmico e fluido, do que quando a coluna ainda não havia sido automatizada. Executando em horas o que necessitava de dias anteriormente.

Outro ponto fundamental do trabalho foi a utilização do arduino como interface entre a coluna e o computador. O arduino necessita receber a programação para manipular as informações recebidas e para atuar sobre os equipamentos, a partir de uma conexão com um computador é possível utilizar o arduino IDE para programar o micro controlador, como na imagem a seguir:

Figura 14 - Arduino IDE.



Fonte: autor

A Figura 14 mostra a interface de desenvolvimento no arduino, a partir dela é possível escrever o código em C++ e definindo a porta serial de comunicação, o programa pode ser enviado ao micro controlador. O apêndice A, mostra parte do código utilizado para o desempenho das funções do arduino, nele é possível observar as bibliotecas utilizadas, as variáveis definidas, a inicialização dos pinos e dos sensores, a leitura dos sensores, a leitura das informações provenientes do supervisório, entre muitas outras funções desempenhadas.

Na coluna, foram adicionados dois recipientes (baldes) para armazenamento do etanol e do óleo, para que as bombas trabalhassem com mais facilidade, utilizando a gravidade como parte da energia e deixando as bombas afogadas para que não trabalhassem a seco. Um tanque para guardar a água de resfriamento foi construído, permitindo melhores condições para a bomba de resfriamento, e permitindo o acompanhamento de sua temperatura. A configuração final da coluna está na Figura 6.

Na Figura 6 é possível verificar a implementação da automação, e das melhorias na coluna. Os fluxos mássicos da Figura 5 foram mantidos. Na Figura 6 é possível observar a representação em 3D da coluna, desenvolvida para a construção do supervisório.

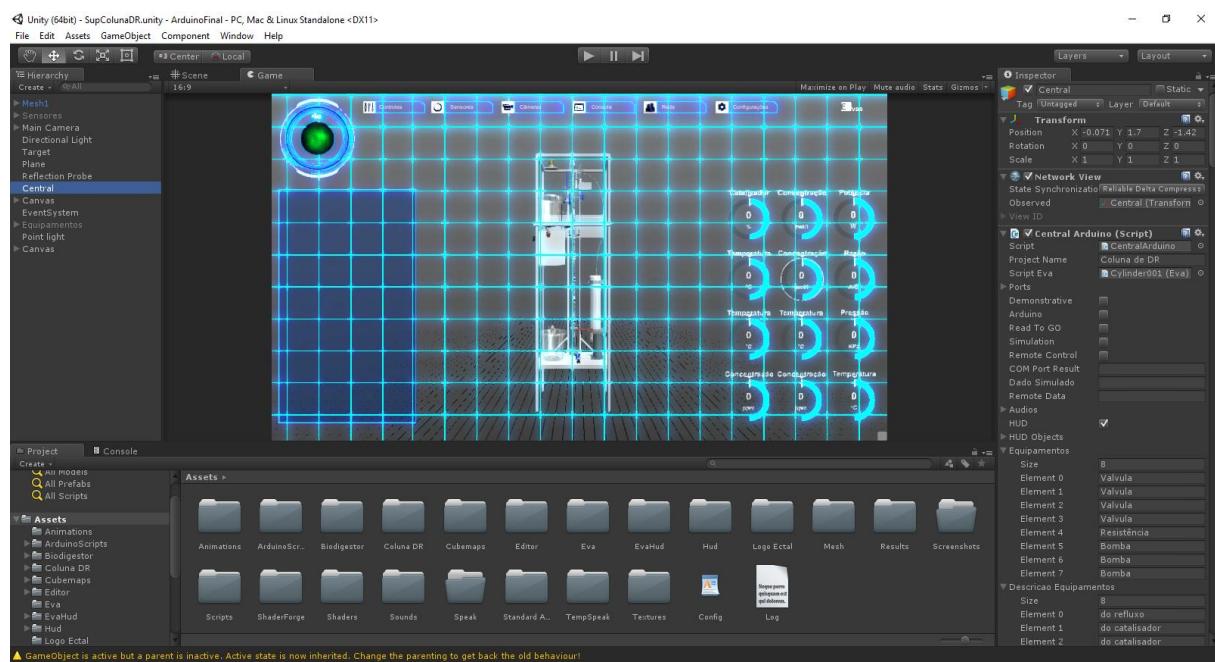
No desenvolvimento da coluna não houve um controle rígido nos custos. Porém estima-se que o custo total para construção e automação da coluna foi de cerca R\$ 5000,00, a montagem custou cerca de R\$ 3000,00 e a automação cerca de R\$ 2000,00. Este valor está muito abaixo do valor de mercado de uma coluna deste tipo (desconsiderando a mão de obra), o que levaria a valor estimado em R\$ 100.000,00, onde não seria incluído muitas das características de automação avançada incluída nela. Portanto, foi possível construir um equipamento funcional e avançado com cerca uma fração do preço comercial. **O que abre portas para o desenvolvimento de equipamentos semelhantes de baixo custo**, permitindo um melhor aproveitamento da verba disponível em projetos de pesquisa.

5.3 Desenvolvimento do supervisório

Para controle e supervisão dos equipamentos e sensores citados é necessário o auxílio de um supervisório, graças a capacidade de processamento de um computador é possível gerenciar milhares de informações simultaneamente, tornando o processo produtivo mais seguro, confiável e de fácil gerenciamento.

O supervisório desenvolvido pelo autor deveria permitir o total controle sobre a coluna, com o mínimo de esforço do usuário, para isso utilizou-se um software chamado Unity 3D®, o mesmo é utilizado no desenvolvimento de jogos e aplicativos interativos em 3D. O Unity 3D® utiliza como linguagem de programação o C#, dentro de um framework da Microsoft®, o .NET® framework, que inclui diversos comando e funções já estabelecidos, permitindo por exemplo a conexão via porta serial com o arduino, por isto ele foi escolhido. A interface de desenvolvimento do Unity 3D® pode ser vista na Figura 15:

Figura 15 - Interface de desenvolvimento do Unity 3D®



Fonte: autor.

Na Interface do Unity 3D® é possível acessar todos os arquivos incluídos no projeto, como imagens, modelos, materiais, scripts entre outros, chamados de “assets”, é possível incluí-los na “cena” que representa o ambiente atual de criação, a tela maior e central. Do lado direito pode-se observar as características dos objetos selecionados, assim como posição, rotação, escala e scripts associados.

O Unity 3D® permite o desenvolvimento de softwares em 3D para controles e visualização de processos, como os desenvolvidos pelo LSDPQ (SILVA et al., 2016, TSCHOEKE et al., 2016a, 2016b e LIMA et al., 2015). Este software permite desenvolver programas de computador que incluem visualizações 3D, ampliando o nível de realismo na representação computacional. A capacidade de processamento atual dos computadores

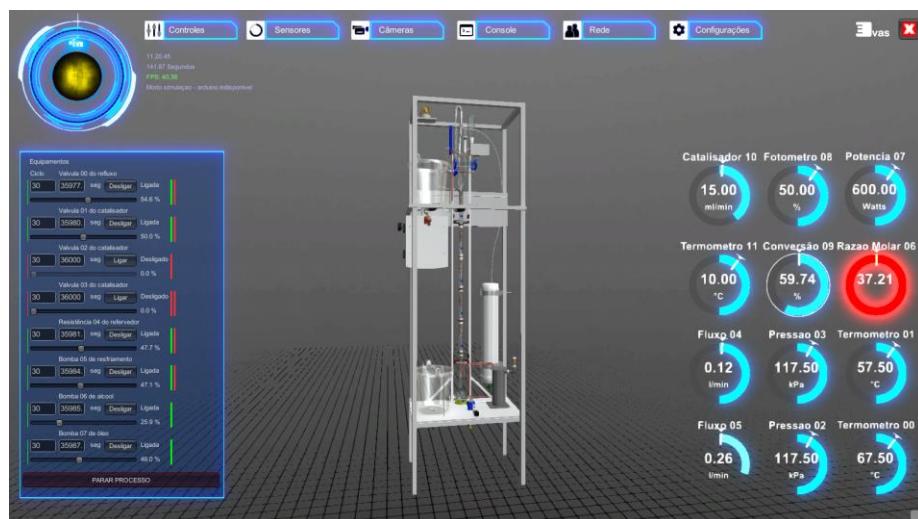
permite representações mais fiéis aos processos, do que as Figuras estáticas em 2D que comumente são utilizadas nos supervisórios. A utilização de supervisórios em 3D é algo pouco explorado academicamente até o momento, o que indica inovação.

Partindo do Unity 3D[©] e de suas características adequadas foi possível desenvolver ao longo de três anos o supervisório da coluna de destilação reativa. Para isso foi necessário se criar todo um protocolo de comunicação entre o arduino e o supervisório, para que houvesse troca de informações entre o computador e a coluna. Códigos foram definidos em ambos para que as informações recebidas e emitidas fossem interpretadas corretamente os detalhes podem ser vistos nos apêndices **A** e **B**.

O modelo 3D da coluna foi desenvolvido utilizando o 3ds max[©], um programa de modelagem 3D, muito utilizado em arquitetura, desenvolvimento de jogos e no desenvolvimento de filmes em computação gráfica. O modelo 3D da coluna foi atualizado a partir das modificações que eram feitas na coluna, cada componente que era adicionado, era modelado, e consequentemente disponível para visualização e interação no supervisório.

A interface de usuário do supervisório foi desenvolvida utilizando funções em 2D e 3D do Unity 3D[©], os equipamentos podem ser controlados segundo um menu na lateral esquerda, como na Figura 16, ciclos de trabalhos podem ser definidos para cada equipamento, para que eles possam trabalhar de forma intermitente, um indicador colorido demonstra se o equipamento está ligado, cronômetros podem ser definidos para cada equipamento. Os detalhes da programação podem ser vistos no apêndice **B**.

Figura 16 - Interface do supervisório da coluna de destilação reativa



Fonte: autor

Os sensores são visualizados do lado direito do supervisório, indicadores circulares indicam o valor referente a cada sensor, algumas variáveis expostas pelos sensores são calculadas por inferências, utilizando o histórico de operação da coluna. O valor médio de cada variável é indicado por um traço branco que se move no sentido circular da medida. Cada sensor é definido com seus limites máximos e mínimos, caso o sensor ultrapasse o mesmo muda de cor. Se o usuário clicar em cima do sensor um relatório é exposto na forma de voz, pela assistente virtual de supervisão desenvolvida pelo autor, chamada de **EVA** (*Extensible Virtual Assistant*).

Para auxílio na supervisão da coluna de destilação reativa uma assistente virtual foi desenvolvida, no supervisório EVA fica representada pelo círculo colorido do lado superior esquerdo, ela apresenta na forma de voz os fatos importantes ocorridos na coluna, como quando um equipamento é ligado/desligado. Ou quando uma variável ultrapassa os limites definidos, ou aconselha checagens, quando necessárias, entre muitas funções desempenhadas por ela. EVA também permite o reconhecimento de fala para comandos e relatórios.

EVA foi desenvolvida neste trabalho como um programa a parte que poderia falar e ouvir, utilizando conceitos básicos de inteligência artificial. Ela foi integrada no supervisório como uma ajudante de alto nível, a mesma foi desenvolvida no Microsoft Visual Studio®, também em C#, utilizando as funções mais recentes do Windows® 10, que permitem o reconhecimento e a síntese de voz em português, a integração com o Unity 3D® foi feita utilizando o *clipboard* (área de transferência) do Windows. Com EVA é possível operar a coluna apenas com comandos de voz, por exemplo: usuário diz: “Eva. Ligue a válvula de refluxo”, EVA responde: “Com que fração do ciclo?”, Usuário responde: “10 por cento”, daí a válvula operará com 10% do tempo aberta. Entre muitas outras funções programadas.

O menu na parte central superior do supervisório é possível observar uma série de botões, que incluem diversas funções: controles, sensores, câmeras, console e rede, como na Figura 17:

Figura 17 - Funções do supervisório da coluna de destilação reativa, da esquerda para direita, de cima para baixo tem-se: Sensores, câmeras, console e rede.



Fonte: autor.

O uso do botão de controles permite visualizar ou esconder os botões de operação do menu de equipamentos, na esquerda. O botão de sensores permite a visualização de gráficos que contém as informações recentes dos sensores, como gráficos ao longo do tempo, valores máximos, mínimos, média e taxa de variação instantânea (derivada), com isso é possível haver um acompanhamento detalhado dos sensores. O botão de câmeras permite a visualização de qualquer câmera que esteja conectada ao computador, três câmeras foram colocadas na coluna em pontos estratégicos, e com elas é possível acompanhar de perto, pontos críticos do equipamento. No botão de console é possível observar um registro contínuo dos eventos da coluna, também chamado de “LOG”, além do controle de funções do supervisório, como permitir que EVA fale ou não, e enviar automaticamente um e-mail com os dados gerados. No botão de rede é possível conectar o supervisório pela Internet com o mesmo programa de supervisão em outro lugar do mundo, ou seja, é possível fazer supervisão remota. Onde, à distância, outros podem verificar e setar equipamentos via Internet. Um aplicativo de celular também foi desenvolvido para este propósito. Algumas telas do mesmo podem ser observadas a seguir:

Figura 18 - Visão geral das telas do aplicativo desenvolvido.



Fonte: autor.

Na Figura 18 é possível observar o aplicativo de celular foi desenvolvido a partir do zero, assim como o supervisório, para que a supervisão pudesse ser feita de forma portátil e remota, o aplicativo foi desenvolvido em Unity 3D[©] também, pois o Unity permite o desenvolvimento para diversas plataformas como: Windows[©], Mac[©], Linux[©], Android[©], IOS[©], Windows Phone[©], Xbox[©], Playstation[©], e Wii[©]. Partindo do projeto base do supervisório modificações foram feitas para que ele se tornasse o aplicativo.

O aplicativo permite a conexão, visualização, e supervisão da coluna, basta que a conexão seja estabelecida como na Figura 18, após isso os dados ficam disponíveis em tempo real. O aplicativo desenvolvido também permite a conversação entre os supervisores, basta que seja habilitado no microfone representado na Figura 18.

O desenvolvimento destas tecnologias tem como intuito o melhor desempenho, confiabilidade e conforto na operação da coluna de destilação reativa. Aprimorando os métodos de automação é possível gerar resultados com um esforço mínimo.

5.4 Testes na coluna de destilação reativa

Após as melhorias na coluna terem sido feitas, e o processo de produção de biodiesel via etanólica ser compreendido, foi necessário testar a coluna de destilação reativa. Para isso um ensaio foi planejado seguindo as condições experimentais expostas na Tabela 7:

Tabela 8 - Condições de operação no teste inicial da coluna de destilação reativa.

Vazão de óleo	0,05 L/min
Vazão de álcool	0,1 L/min
Razão molar resultante (álcool/óleo)	38,14 mol/mol
Concentração de catalisador	1% mássico com relação a vazão de óleo
Vazão de catalisador	18,2 mL/min
Razão de refluxo	0
Carga térmica	500 W (em média)
Pressão	103,20 kPa

As condições expostas na Tabela 7 foram definidas considerando vários pontos:

A coluna tem limites nas condições operacionais, como por exemplo as vazões de óleo e álcool, elas precisam ser baixas para garantir o tempo de residência necessário, valores muito inferiores aos encontrados na Tabela 7 tornariam as medidas de vazão mais imprecisas, porém maximizam a conversão do óleo. A razão de refluxo foi utilizada como sendo zero devido a limitações durante este teste.

Outro ponto levado em consideração é que certas condições operacionais podem maximizar a conversão. Silva (2013) estudou por meio de simulações, as condições que maximizam a conversão do óleo em biodiesel numa coluna de destilação reativa. Segundo o trabalho de Silva (2013) se observou que as variáveis operacionais que mais influenciam a conversão em ordem decrescente são: razão molar álcool/óleo, carga térmica no revedor e razão de refluxo. Portanto maximizando estas variáveis é possível obter a máxima conversão do óleo na coluna.

O teste ocorreu seguindo o procedimento operacional do apêndice E, e utilizando os reagentes expostos no anexo A. No procedimento operacional os sensores e os sistemas de atuação foram testados, assim como o abastecimento de óleo, álcool e catalisador foi efetuado, uma vez que as condições de operação são satisfatórias o experimento é iniciado,

após atingir o estado estacionário, as amostras são coletadas, analisadas, purificadas e analisadas novamente, seguindo as Normas AOCS e EN 14103, a limpeza e drenagem é feita na coluna, assim como uma posterior lavagem com etanol. A coluna foi operada digitalmente utilizando computadores com o supervisório desenvolvido, assim como o aplicativo, desta forma, foi operada e monitorada por diversas pessoas ao mesmo tempo, incluindo a assistente virtual, EVA.

Levando em consideração estes pontos foi possível definir as condições que se encontram na Tabela 9, como resultado as seguintes informações foram obtidas:

Tabela 9 - Resultados experimentais obtidos.

Temperatura média da fase vapor no topo	76,44 °C
Temperatura do líquido resultante	72,25 °C
Conversão do óleo resultante (método de purificação 1)	93 %
Conversão do óleo resultante (método de purificação 2)	88 %
Concentração de catalisador na amostra de fundo	0,0 g/mL
Concentração de sabão na amostra de fundo	0,039 g/mL

Na Tabela 9 é possível observar os resultados obtidos. A temperatura do topo e da base da coluna foi suficiente para garantir que todo meio reacional estava sendo aquecido, favorecendo a reação de transesterificação.

O biodiesel produzido foi purificado por dois métodos distintos no método 1 a decantação era feita por meio de centrifugação, e as lavagens eram feitas em maior número com água a pH neutro. O método 2, com decantação gravitacional e com a adição de 4 lavagens, como visto anteriormente, com duas lavagens extras e com água a pH <4,5. como foi descrito na metodologia. O método 1 foi realizado no LAC (Laboratório de Combustíveis), e se mostrou mais eficaz pois atingiu um teor de éster mais elevado, o método 2 foi executado devido as limitações de equipamentos e material que existem em nosso laboratório.

As análises de catalisador residual e sabão foram realizadas em duplicata, os valores da Tabela 9, são os valores médios. De início é possível observar que a concentração de

catalisador foi de 0 g/mL algo aparentemente incoerente, este fato se deve a dois fatores: a acetona é utilizada como solvente na titulação, em um teste com ela pura, como um ‘branco’, demonstrou-se que o seu pH era ácido, que faz com que a acidez neutralize parte do catalisador, foi decidido utilizar um solvente neutro. A norma não é restrita com relação ao solvente utilizado, desde que ele solubilize todos os produtos e reagentes da transesterificação, em todos os ensaios posteriores foi utilizado o etanol. Outro fator decisivo para encontrar a concentração de catalisador como sendo 0 g/mL foi que a análise foi feita muito tempo depois de coletada, cerca de meia hora, nesse tempo o catalisador residual já tem reagido com o biodiesel e com o óleo se tornando completamente sabão. Como exposto nas Equações 11, 12 e 13, deste modo as análises posteriores foram executadas logo após a retirada da amostra da coluna. Com relação a quantidade de sabão encontrada o valor foi de 0,039 g/mL segundo a Tabela 8, foram gastos 6,4 mL de HCl 0,1mol/L, em média, para titular a amostra, este valor obtido servirá de referência para os experimentos posteriores.

Nos testes iniciais foi possível identificar falhas e possíveis melhorias, para que um planejamento fatorial completo fosse executado. Problemas operacionais foram encontrados: a bomba de resfriamento do condensador não suportou a vazão e foi trocada, para que isso não ocorresse novamente. A bomba trabalhou nos experimentos posteriores com um ciclo de trabalho reduzido, mas que garantia o resfriamento do condensador. A interferência causada pelos solenoides se mostrou crítica no monitoramento do processo, diversos sensores oscilavam constantemente devido ao colapso do campo magnético nos solenoides, por este motivo os *snubbers* foram adicionados, como descrito no tópico 5.2 deste trabalho.

Com relação ao supervisório, melhorias são aplicadas constantemente, porém o comportamento do supervisório foi satisfatório, uma das poucas melhorias necessárias foi de que a temperatura no topo da coluna estava acima do limite superior esperado pelo supervisório. Para que não houvessem alertas constantes a faixa de trabalho do sensor de temperatura no topo da coluna foi ampliada.

Resolvendo os problemas encontrados e tomando a condição do experimento como adequada foi possível desenvolver um planejamento fatorial que indicasse quais os parâmetros são mais críticos no controle da produção de sabão durante a transesterificação.

5.5 Planejamento fatorial

Após verificar que a coluna converte óleo de algodão em biodiesel foi possível planejar o experimento para se estudar em detalhes a influência dos parâmetros operacionais na produção indesejada de sabão. Segundo o resultado obtido anteriormente foi definido um planejamento fatorial próximo da região de máxima conversão do óleo, foi escolhido um planejamento fatorial 2^3 , variando o refluxo, a carga térmica e a razão molar, e os outros parâmetros foram mantidos fixos, como exposto na Tabela 10:

Tabela 10 - Condições de operação experimental da coluna de destilação reativa.

Vazão de óleo	0,02 L/min
Vazão de álcool	Variável para controlar a razão molar
Razão molar resultante (Álcool/óleo)	Variável
Concentração de catalisador	1% mássico com relação a vazão de óleo
Vazão de catalisador	15 mL/min
Razão de refluxo	Variável
Carga térmica	Variável
Pressão	103,20 kPa

Segundo a Tabela 10 o catalisador foi alimentado com a vazão de 15mL/min de hidróxido de sódio diluído em etanol e seco na concentração de 20g/L, para atender a necessidade de 1% de catalisador para a vazão de óleo de 20mL/min. A temperatura da fase vapor do topo variou entre 30 e 80°C dependendo das condições experimentais, e lembrando que a coluna trabalha em condições atmosféricas. A temperatura do líquido de saída no fundo variou de 52 a 60°C. Para avaliar a influência destes parâmetros operacionais o refluxo a carga térmica e a razão molar foram variadas segundo a Tabela 11:

Tabela 11 - Parâmetros variados para o estudo da produção de sabão em uma coluna de destilação reativa.

Variável	Nível (-)	Nível (0)	Nível (+)
Refluxo	0	0,5	Total
Carga térmica	200W	300W	400W
Razão molar (Álcool/Óleo)	15/1	20/1	25/1

Os valores da Tabela 10 foram escolhidos levando em consideração diversos critérios, tais como limites operacionais possíveis e condições favoráveis segundo Silva (2013), após definição das variáveis a serem estudadas foi construído um planejamento fatorial 2^3 , com 2 repetições para a titulação que define a quantidade de catalisador residual e concentração de sabão (norma AOCS métodos cc 17-79), foram escolhidos 3 níveis, incluindo pontos centrais. A coluna foi operada digitalmente utilizando um computador com o supervisório desenvolvido instalado seguindo o procedimento operacional do apêndice E, sendo monitorada também a distância via Internet e pelo celular com o aplicativo desenvolvido. O resumo dos resultados obtidos pode ser visualizado na Tabela 12.

Tabela 12 - Planejamento fatorial e resultados obtidos.

Experimento	Refluxo	Carga térmica	Razão molar	Conversão (%)	Concentração de catalisador (g/mL)	Concentração de sabão (g/mL)
1	-	+	+	82	0,00048	0,064
2	+	+	+	91	0,00061	0,058
3	-	-	+	76	0,0013	0,024
4	+	-	+	81	0,0026	0,022
5	-	+	-	80	0,0040	0,038
6	+	+	-	88	0,0040	0,031
7	-	-	-	64	0,0087	0,020
8	+	-	-	73	0,0088	0,008
9	0	0	+	90	0,00075	0,051
10	0	+	0	87	0,00062	0,061
11	0	0	0	80	0,0037	0,035

Na Tabela 12 é possível observar os resultados provenientes do planejamento fatorial, com relação à conversão do óleo os valores obtidos foram provenientes de cromatografia gasosa seguindo a norma EN 14103, os valores obtidos variaram em uma faixa de 64 a 91% de éster no biodiesel purificado.

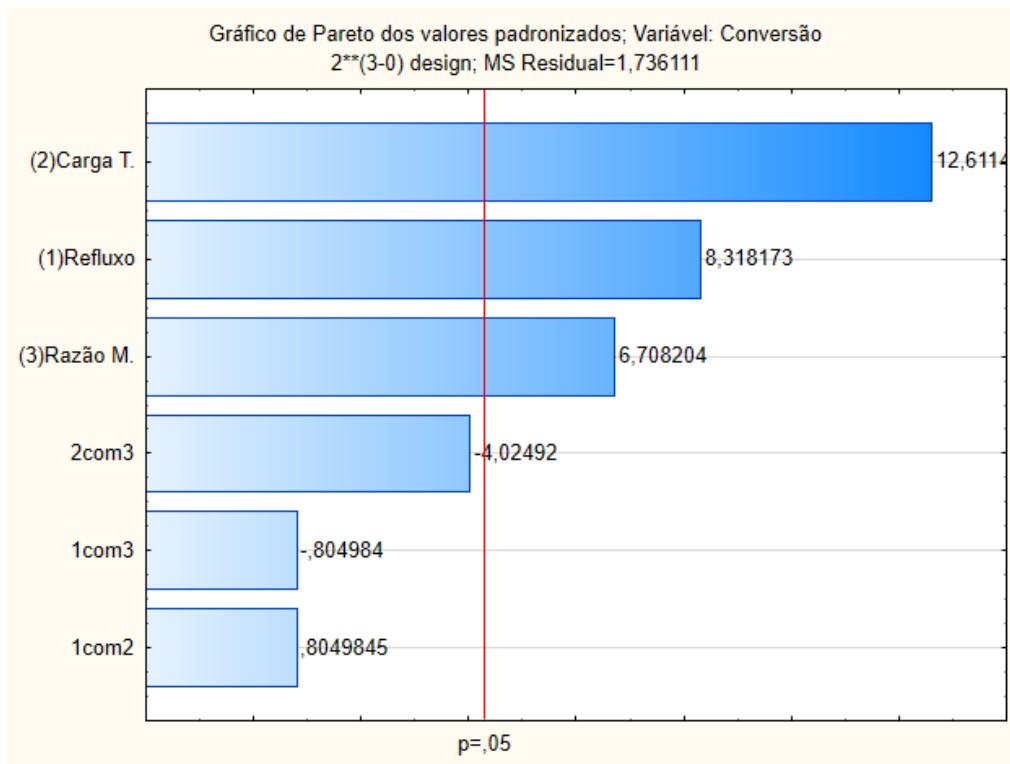
A menor conversão foi obtida com o refluxo mínimo, carga térmica mínima e razão molar mínima. Resultado coerente, pois reduzindo a disponibilidade de etanol na coluna e diminuindo a carga térmica a reação de transesterificação não é favorecida no sentido dos produtos. A condição de menor conversão do óleo é coerente com os resultados de Silva (2013). Trabalho no qual a menor conversão foi obtida com os três parâmetros estudados em seus valores mínimos.

A maior conversão do óleo foi obtida com o refluxo total, carga térmica máxima e maior razão molar etanol/óleo. Segundo Silva (2013) e He (2006) o maior refluxo promove a maior disponibilidade de álcool na coluna, deslocando a reação no sentido dos produtos, assim como a maior carga térmica e maior razão molar. Sendo que uma maior razão molar nem sempre indica maior conversão: com razões molar altas, o excesso de álcool pode gerar efeitos colaterais, com o aumento de volume de líquidos circulando pela coluna, que leva também a uma redução do tempo de residência dos reagentes, assim como ocorre a redução da concentração de catalisador, como exposto por Ding et al. (2014).

5.5.1 Resultados do planejamento com relação a conversão

Os dados obtidos foram tratados estatisticamente, com relação a conversão do óleo. Utilizou-se o software Statistica[©], a partir da metodologia citada no apêndice F, obtendo dentre muitos resultados o gráfico de Pareto da Figura 19:

Figura 19 - Gráfico de Pareto para a conversão do óleo.



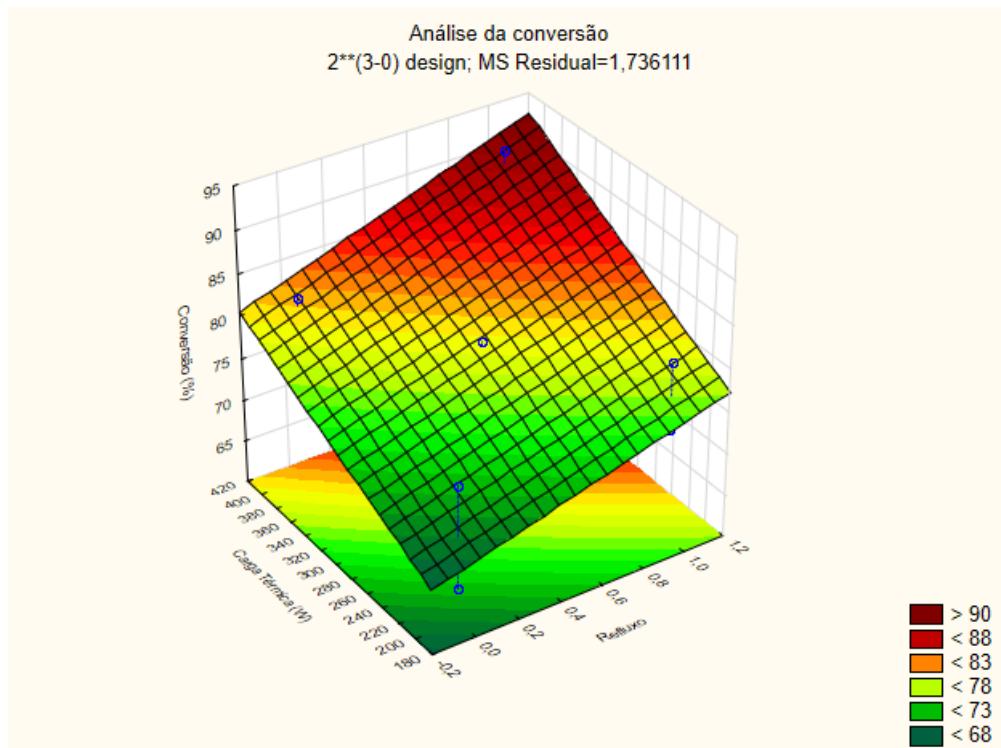
Fonte: autor.

A Figura 19 indica que os efeitos de todos os parâmetros estudados são significativos, porém suas combinações não o são. A figura 19 define uma ordem de significância entre os parâmetros. A variável 2 representada pela carga térmica se mostrou mais significativa dentre os parâmetros estudados, seguido do refluxo, na sequencia a razão molar, sendo a interação razão molar/carga térmica muito próxima de ser considerada significativa. Os resultados obtidos estão parcialmente coerentes com o que foi encontrado por Silva (2013), que encontrou a razão molar como sendo mais importante que a carga térmica dada, tal diferença pode ser atribuída as diferenças operacionais e com relação as diferenças da coluna simulada e da coluna real, nos testes da coluna real a carga térmica foi testada em três níveis, sendo o superior duas vezes maior que o inferior, enquanto na simulação de Silva (2013) havia um aumento de apenas 20%, tornando a influência da carga térmica superior. Da análise estatística também foi obtida a equação que descreve a interrelação entre os fatores e a conversão:

$$\text{Co} = 8.5 \cdot x + 0.13103 \cdot y + 1.9734 \cdot z + 0.0075 \cdot xy - 0.00375 \cdot yz - 0.15 \cdot xz + 20.043 \quad (24)$$

Na Equação 24 os parâmetros x, y e z são respectivamente razão de refluxo, carga térmica e razão molar (álcool/óleo), e “Co” é o teor de éster no biodiesel produzido, e o coeficiente de determinação, R^2 , é de 0,924. Com a Equação 24 é possível generalizar e representar os experimentos da Tabela 12, com relação a conversão. O apêndice **H** demonstra uma forma alternativa de se encontrar a Equação 24 utilizando o Matlab[®]. O script do apêndice **H** ainda verifica o espaço amostral do experimento procurando valores de máximo e mínimo, que equivalem a x=0, y=200, z=15 e x=1, y=400, z=25 respectivamente. Que equivalem aos valores de máximo e mínimo da Tabela 12. O que indica que não existem pontos de inflexão dentro dos intervalos estudados, e o teor de ester produzido é proporcional ao aumento do refluxo, aumento da carga térmica e o aumento da razão molar, e que o contrário também é verdade. Os gráficos de superfície que validam estas informações, serão expostos a seguir. A relação entre a carga térmica e o refluxo pode ser observado no gráfico de superfície a seguir.

Figura 20 - Gráfico de resposta da conversão para carga térmica e refluxo.

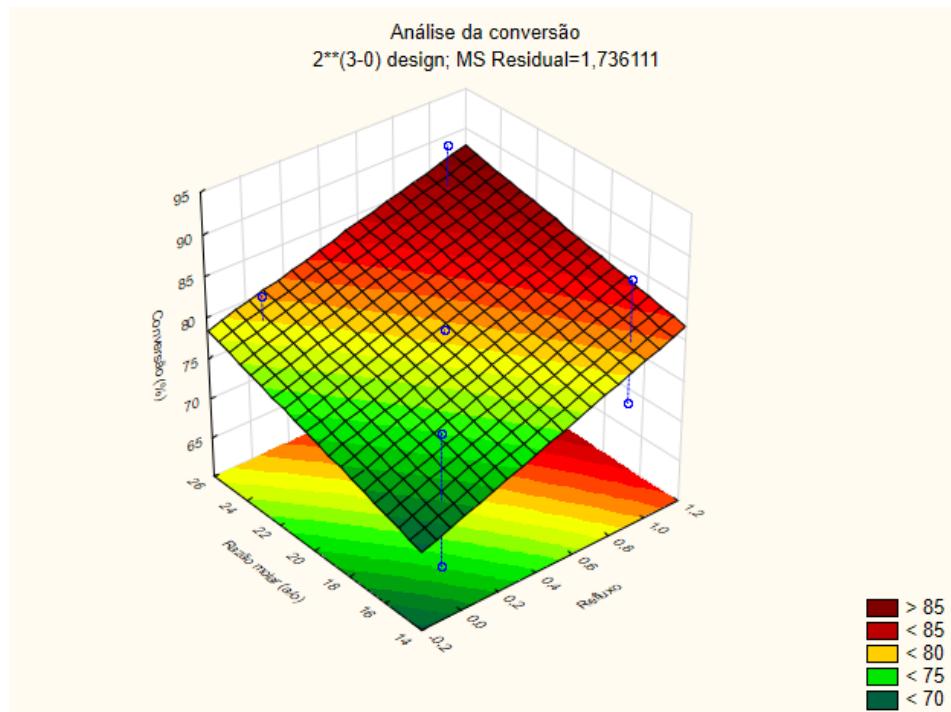


Fonte: autor.

Segundo a Figura 20 um maior refluxo assim como uma maior carga térmica levam a uma maior conversão do óleo, fato que está de acordo com o trabalho de Silva (2013) e também por Boon-Anuwat et al. (2015), pois a reação de transesterificação é favorecida pelo aquecimento e pelo excesso de álcool cedido pelo condensador, consequentemente o inverso

também ocorre. Observando o gradiente da Figura 20 é constatado que a carga térmica exerce uma influência de grandeza similar ao refluxo sobre a conversão. A relação entre o refluxo e a razão molar pode ser visualizada na Figura 21 a seguir.

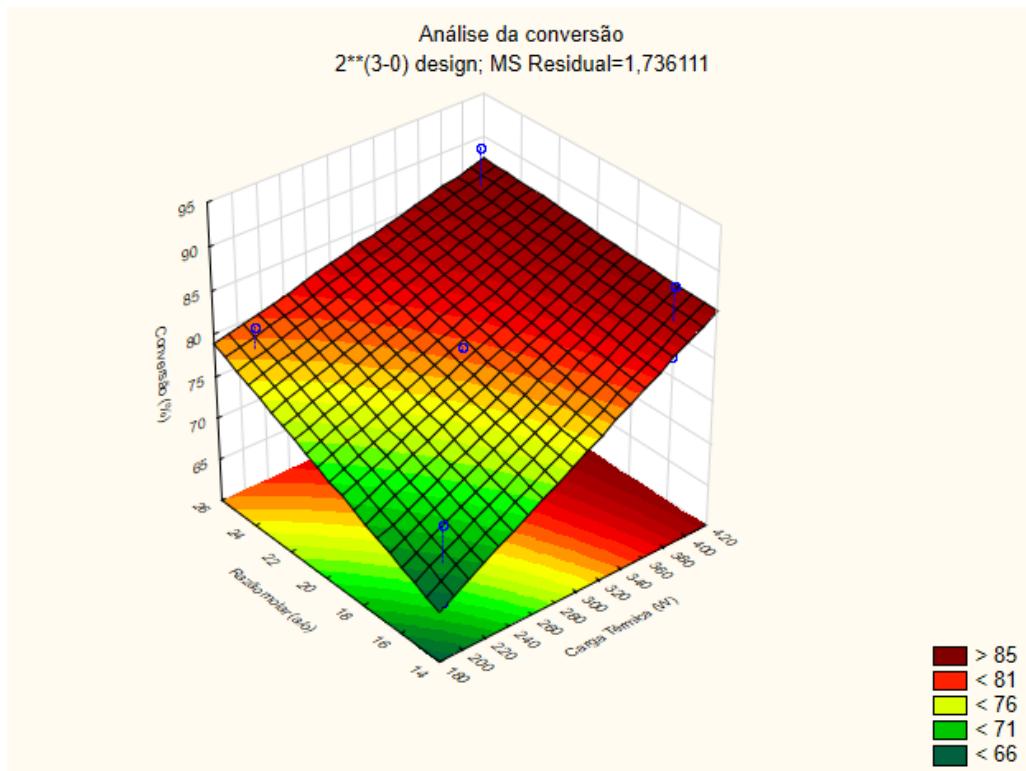
Figura 21 - Gráfico de resposta da conversão para razão molar e refluxo.



Fonte: autor.

A Figura 21 demonstra que mantendo a carga térmica a 300W a conversão do óleo é favorecida, tanto pelo refluxo quanto pela razão molar, o que está de acordo com os trabalhos de Silva (2013) e Boon-Anuwat et al. (2015). Na Figura 21 é possível observar que quanto menor for a razão molar etanol/óleo, menor será a conversão. E que quanto menor for o refluxo, menor será a conversão do óleo, os teores de éster observados estão entre 72% e 90% nestas condições, conforme citado anteriormente uma carga térmica maior precisa ser dada para atingir o resultado máximo obtido nos experimentos, ou seja, 400W. Na Figura 22 é exposta a influência da razão molar e carga térmica na conversão.

Figura 22 - Gráfico de resposta da conversão para Razão molar e carga térmica.

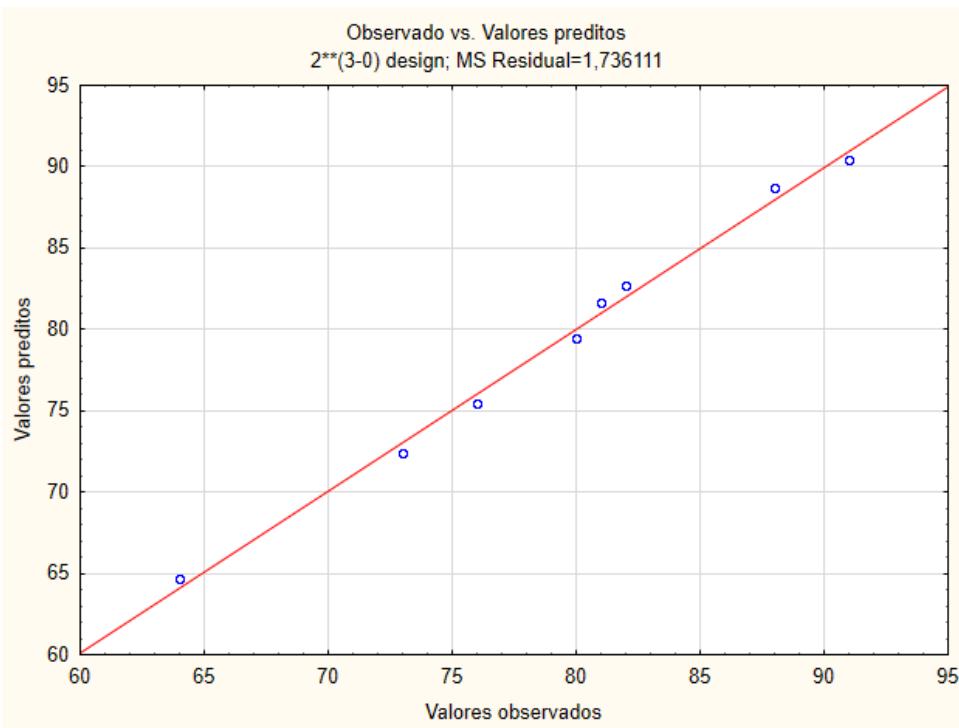


Fonte: autor

Observando a Figura 22 é verificada que valores altos para razão de molar e carga térmica é obtida uma conversão do óleo de 88%. Por outro lado, quando se reduz a carga térmica e mantém-se uma grande razão molar alta, uma conversão de aproximadamente 80% é atingida, e para uma razão molar e carga térmica baixa, uma conversão do óleo de cerca de 66% é atingida. Esses fatos estão de acordo com os trabalhos de Silva (2013), He (2006) e Boon-Anuwat et al. (2015).

Na Figura 22 é possível observar que o gradiente da razão molar é mais acentuado para regiões de baixa carga térmica, enquanto que valores altos de carga térmica levam a uma influência menor da razão molar. Este fato se deve ao favorecimento da conversão quando uma maior carga térmica é utilizada no processo, atingindo mais facilmente altas conversões, definindo um comportamento assintótico, já que a conversão máxima é de 100%. Por outro lado, quando a carga térmica é menor, a conversão do óleo é controlada principalmente pela razão molar.

Figura 23 - Valores observados em função dos valores previstos.



Fonte: autor.

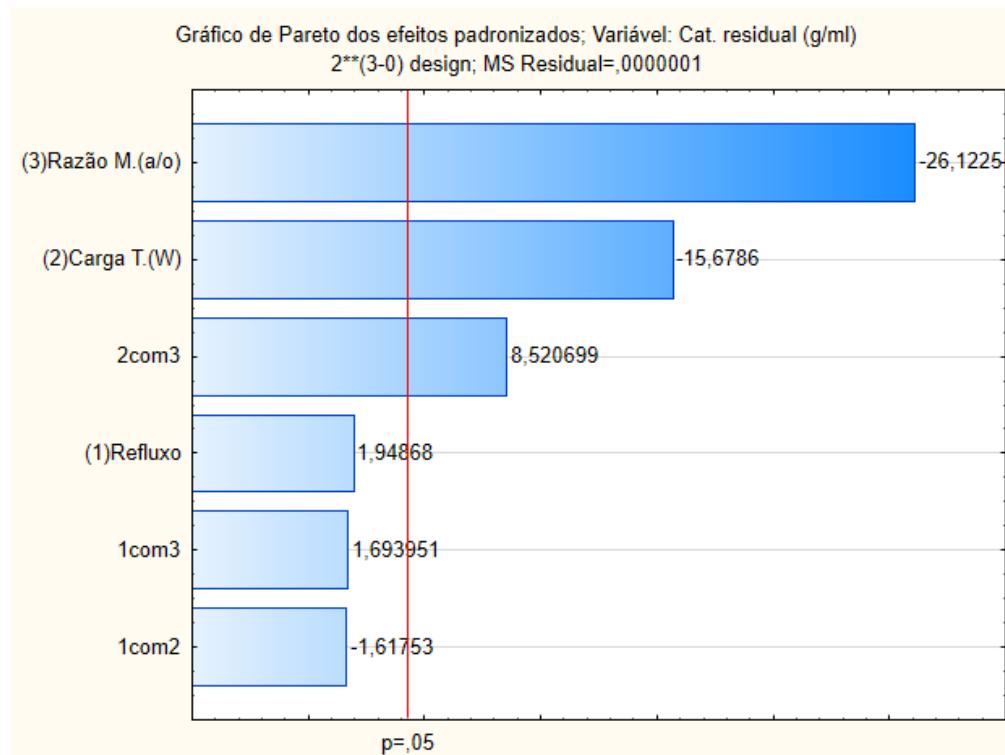
A Figura 23 demonstra os valores observados em função dos valores previstos da conversão do óleo. Cada círculo azul equivale a um resultado experimental, a linha vermelha equivale ao modelo resultante do planejamento fatorial, é possível observar uma concordância adequada entre o modelo e os dados experimentais, com desvios quadráticos médios de 0,4027. O mesmo estudo estatístico foi realizado também para as concentrações de catalisador residual e de sabão produzidos.

5.5.2 Resultados do planejamento com relação ao catalisador residual

Segundo a norma AOCS métodos cc 17-79 disponível no item 5.4 desta tese foi possível quantificar o quanto de catalisador residual e de sabão foram produzidos em cada experimento, na Tabela 11 estão presentes os resultados resumidos, cada valor foi obtido por uma análise em duplicata. O catalisador adicionado tem apenas dois caminhos a seguir segundo o balanço material: Na corrente de fundo da coluna ele pode sair na forma de catalisador inalterado, ou reage e forma sabão. Um valor alto de catalisador na saída da coluna é positivo para a reação, pois mostra que o mesmo continuou catalisando a reação ao máximo até sua saída. É possível observar que diversos fatores influenciaram a concentração de catalisador residual, principalmente a razão molar. A condição que favoreceu a concentração

de catalisador foi: refluxo total, carga térmica e razão molar baixas. O refluxo favorece a diluição do catalisador, diminuindo a velocidade da reação de saponificação, e diminuindo a temperatura da coluna, e favorece também a transesterificação, mantendo o catalisador. A carga térmica baixa desfavorece a saponificação. A razão molar baixa reduz a quantidade de álcool vaporizado no interior da coluna. A partir disto foi possível realizar um estudo estatístico sobre a influência dos parâmetros operacionais.

Figura 24 - Gráfico de Pareto para a concentração de catalisador residual.



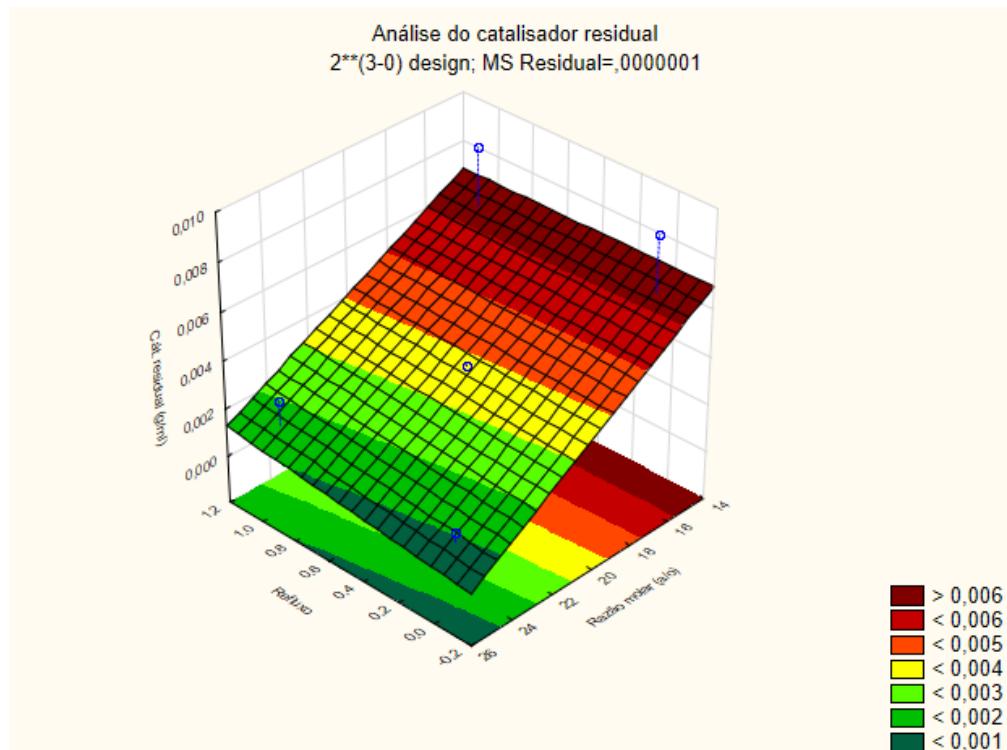
Fonte: autor

Na Figura 24 é possível observar a influência dos parâmetros operacionais na concentração de catalisador residual. O parâmetro mais significativo foi a razão molar com valor de -26,12, significando que são grandezas inversamente proporcionais. Este parâmetro é seguido da carga térmica, e posteriormente da relação entre razão molar e carga térmica. Sendo os outros parâmetros não influentes no processo. A concordância do modelo estatístico com os resultados experimentais apresenta ser adequada, devido ao resíduo de apenas 10^{-7} . Da análise estatística também foi obtida a equação que descreve a interrelação entre os fatores e o teor de catalisador residual, como exposto na Equação 25.

$$\begin{aligned} Ca = & 4.9996 \cdot 10^{-6} \cdot x - 4.8878 \cdot 10^{-5} \cdot y - 0.0010546 \cdot z - 3.175 \cdot 10^{-6} \cdot xy \\ & + 1.6725 \cdot 10^{-6} \cdot yz + 6.65 \cdot 10^{-5} \cdot xz + 0.0291 \end{aligned} \quad (25)$$

Na Equação 25 os parâmetros x, y e z são respectivamente razão de refluxo, carga térmica e razão molar (álcool/óleo), e “Ca” é a concentração de catalisador residual em g/mL, e o coeficiente de determinação, R^2 , é de 0,974. Com a Equação 25 é possível generalizar e representar os experimentos da Tabela 12, com relação a concentração de catalisador residual. O apêndice I demonstra uma forma alternativa de se encontrar a Equação 25 utilizando o Matlab[®]. O script do apêndice I ainda verifica o espaço amostral do experimento procurando valores de máximo e mínimo, que equivalem a x=1, y=200, z=15 e x=0, y=400, z=25 respectivamente. Que equivalem aos valores de máximo e mínimo da Tabela 12. O que indica que não existem pontos de inflexão dentro dos intervalos estudados, e que a quantidade de catalisador residual é proporcional ao aumento do refluxo, redução da carga térmica e a redução da razão molar, e que o contrário também é verdade. Os gráficos de superfície que validam estas informações serão expostos a seguir:

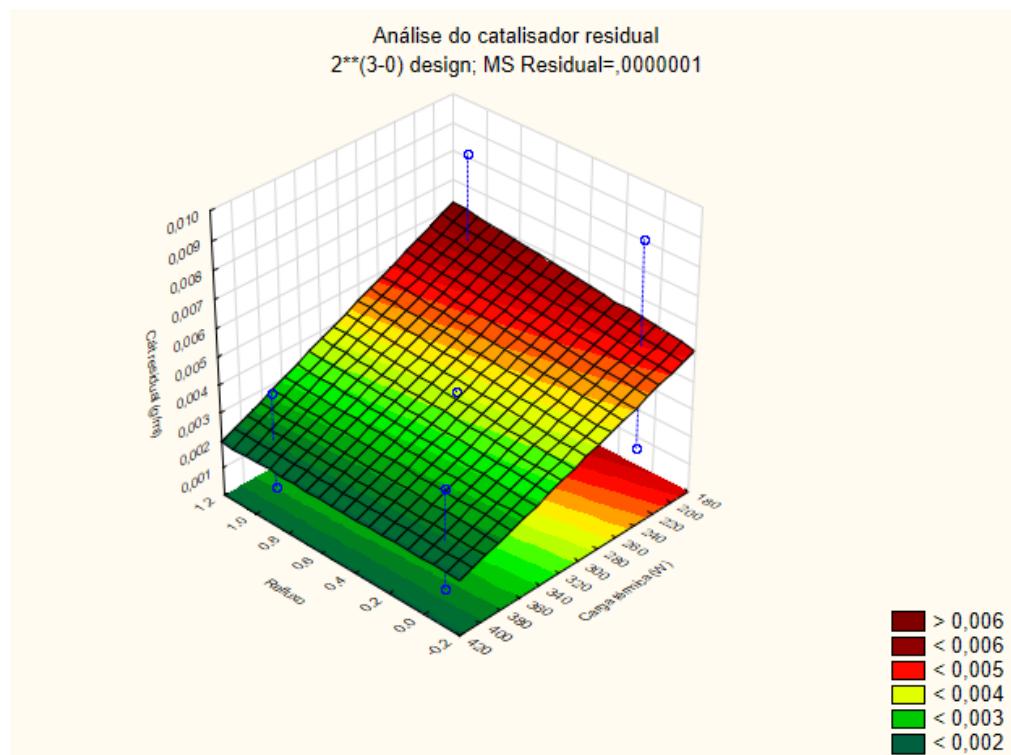
Figura 25 - Gráfico de resposta da concentração de catalisador residual para razão molar e refluxo.



Fonte: autor.

Na Figura 25 é possível observar os efeitos do refluxo e da razão molar no catalisador residual. Verifica-se que o refluxo não afeta significativamente a quantidade de catalisador na saída da coluna. Por outro lado, a razão molar, como visto na Figura 25, é o fator mais influente, seguido da interação entre a razão molar e da carga térmica. Este fato se deve a disponibilidade de álcool quente no interior da coluna. Como será visto posteriormente, o calor é um dos fatores que mais favorece a formação de sabão, quanto mais álcool aquecido entrar na coluna mais sabão será formado, consequentemente a quantidade de catalisador será reduzida. Não há estudos similares sobre esse resultado na literatura.

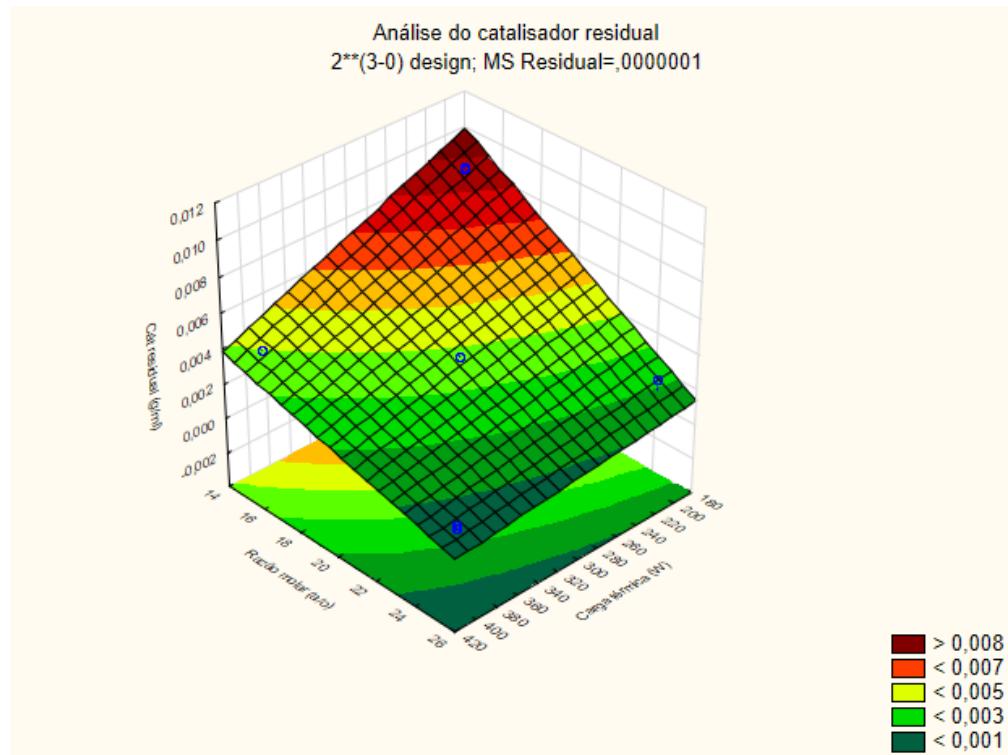
Figura 26 - Gráfico de resposta da concentração de catalisador residual para carga térmica e refluxo.



Fonte: autor.

Na Figura 26 é possível observar a influência carga térmica e do refluxo na concentração de catalisador, o gradiente em relação ao refluxo é menor, o que demonstra sua fraca influência na conversão do óleo, enquanto a carga térmica se mostra influente sobre a conversão. Um alto refluxo e uma baixa carga térmica levam a uma alta disponibilidade de catalisador na saída da coluna. As condições que indicam uma baixa concentração de catalisador na saída da coluna são: refluxo alto ou baixo, e alta carga térmica. Sendo assim, estas condições devem ser evitadas do ponto de vista operacional.

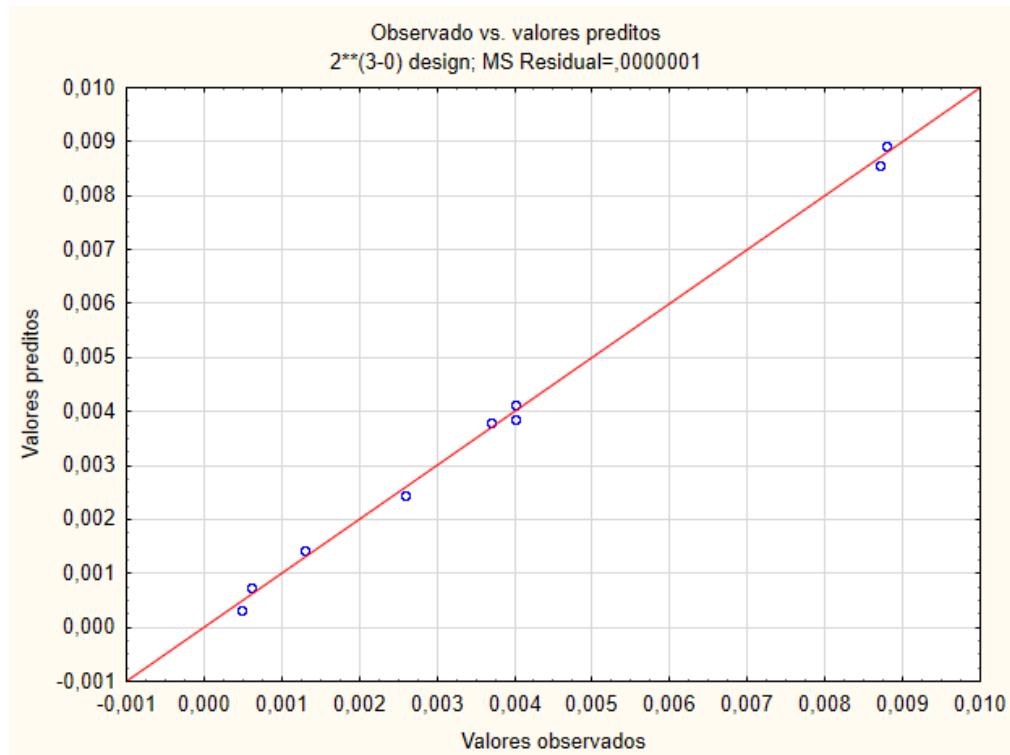
Figura 27 - Gráfico de resposta da concentração de catalisador residual para carga térmica e da razão molar.



Fonte: autor.

Na Figura 27 é possível observar a influência da carga térmica e da razão molar na concentração de catalisador no fundo da coluna. Como dito anteriormente a razão molar é o fator mais importante. Sua influência é superior a carga térmica, pois uma grande quantidade de álcool, sem calor não favorece a saponificação. Sendo assim, a carga térmica associada com a alta razão molar também mostrou ser um efeito significativo sobre a concentração de catalisador residual. Com isso é possível concluir que **a reação de saponificação não é significativamente influenciada pela reação de transesterificação**, pois o aumento da razão molar deveria favorecer a reação concorrente, de transesterificação, diminuindo a concentração final de sabão e aumentando a quantidade de catalisador residual, mas ocorre o contrário. Segundo (Leung et al., 2010) conforme a Equação 13. A reação de saponificação ocorre inclusive com o biodiesel, podendo, então, formar sabão a partir do ácido graxo, do triacilglicerol (óleo) e do biodiesel.

Figura 28 - Valores observados em função dos valores previstos.



Fonte: autor.

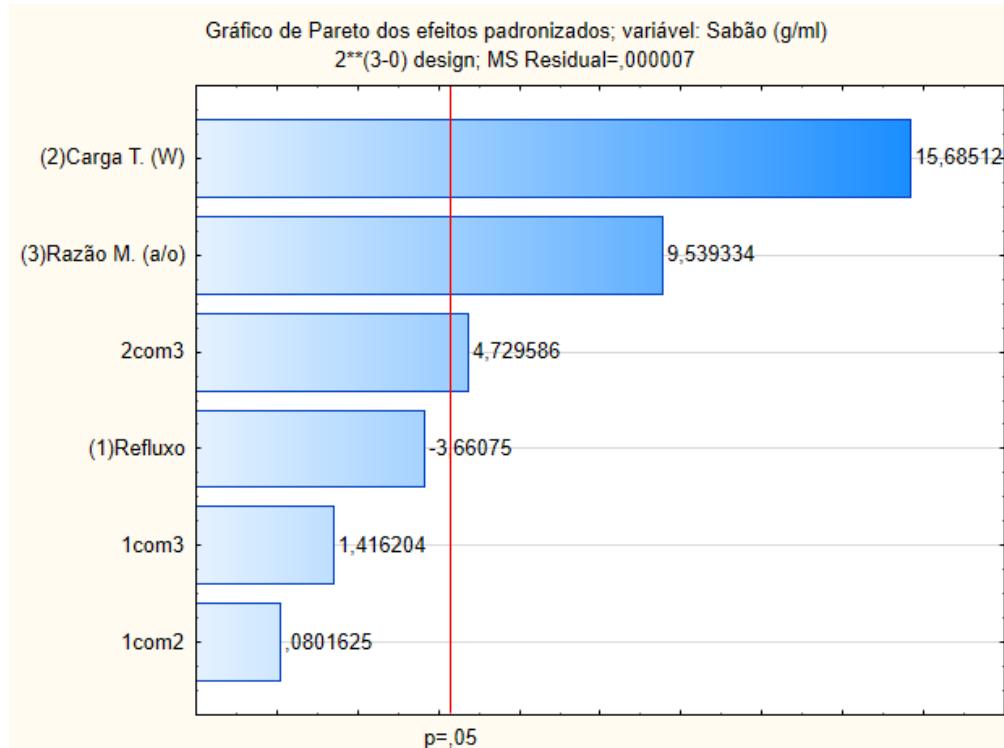
Na Figura 28 é possível observar a concordância entre os dados experimentais e o resultado do modelo estatístico, é possível observar uma ótima concordância entre o modelo e os dados experimentais, com desvios quadráticos médios de 10^{-7} . O mesmo estudo estatístico foi realizado também para as concentrações de sabão no fundo da coluna.

5.5.3 Resultados do planejamento com relação ao sabão

Como exposto anteriormente, para se quantificar catalisador residual e sabão a norma AOCS métodos cc 17-79 foi seguida, a mesma está disponível no item 5.4 desta tese. Na Tabela 11 estão presentes os resultados resumidos, e cada valor foi obtido por uma análise em duplicata. O sabão produzido gera uma série de efeitos indesejados, dificultando o processo de purificação do biodiesel e tornando o biodiesel gelatinoso ou até mesmo sólido. Na Tabela 11 é possível observar que diversos fatores influenciaram o sabão formado, principalmente a razão molar. A condição que favoreceu a formação de sabão foi: refluxo baixo, carga térmica alta e alta razão molar. A carga térmica foi um efeito significante como será visto adiante, pois calor favorece a saponificação, assim como a disponibilidade de álcool aquecido no interior da coluna também influencia a temperatura. Pela Tabela 11 é possível concluir que uma baixa carga térmica pode reduzir a quantidade de sabão, o que é desejado. A partir dos

dados disponíveis um planejamento fatorial foi realizado, que gerou o gráfico de Pareto da Figura 29.

Figura 29 - Gráfico de Pareto para a concentração de sabão residual.



Fonte: autor.

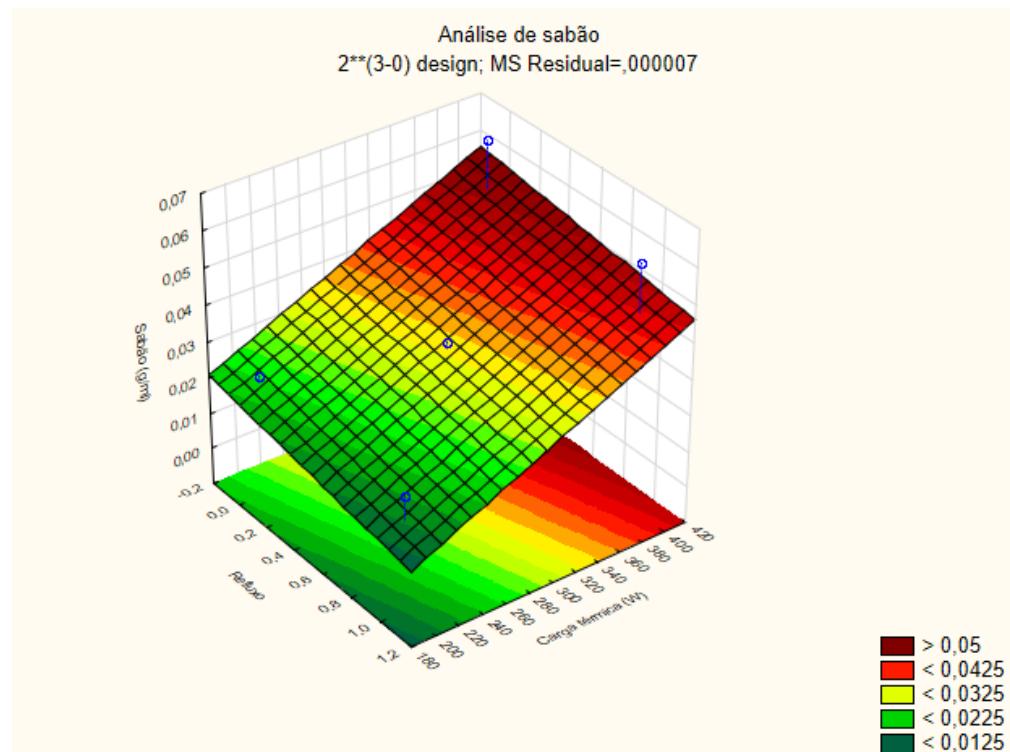
A Figura 29 indica que os efeitos da carga térmica, da razão molar e sua combinação são estatisticamente significativos, os outros parâmetros estudados sobre o sabão produzido não são significativos. A relação entre a razão molar e a carga térmica também se mostrou significativa dentre os parâmetros estudados. Este fato é devido ao aquecimento da coluna como exposto anteriormente. A relação entre a carga térmica e o refluxo pode ser observado no gráfico de superfície a seguir. Da análise estatística também foi obtida a equação que descreve a interrelação entre os fatores e a concentração de sabão, como exposto na Equação 26.

$$Sa = -0.0185 \cdot x - 1.751 \cdot 10^{-5} \cdot y - 0.0010546 \cdot z - 2.5 \cdot 10^{-6} \cdot xy + 8.75 \cdot 10^{-6} \cdot yz + 0.00055 \cdot xz + 0.010655 \quad (26)$$

Na Equação 26 os parâmetros x, y e z são respectivamente razão de refluxo, carga térmica (W) e razão molar (álcool/óleo), e “Sa” é a concentração de sabão residual em g/mL, e o coeficiente de determinação, R^2 , é de 0,943. Com a Equação 26 é possível generalizar e

representar os experimentos da Tabela 12, com relação a concentração de sabão. O apêndice **J** demonstra uma forma alternativa de se encontrar a Equação 26 utilizando o Matlab[®], O script do apêndice **J** ainda verifica o espaço amostral do experimento procurando valores de máximo e mínimo, que equivalem a $x=0$, $y=400$, $z=25$ e $x=1$, $y=200$, $z=15$ respectivamente. Que equivalem aos valores de máximo e mínimo da Tabela 12. Isto indica que não existem pontos de inflexão dentro dos intervalos estudados, e que a concentração de sabão residual é proporcional a redução do refluxo, aumento da carga térmica e a aumento da razão molar, e que o contrário também é verdade. Os gráficos de superfície que validam estas informações serão expostos a seguir:

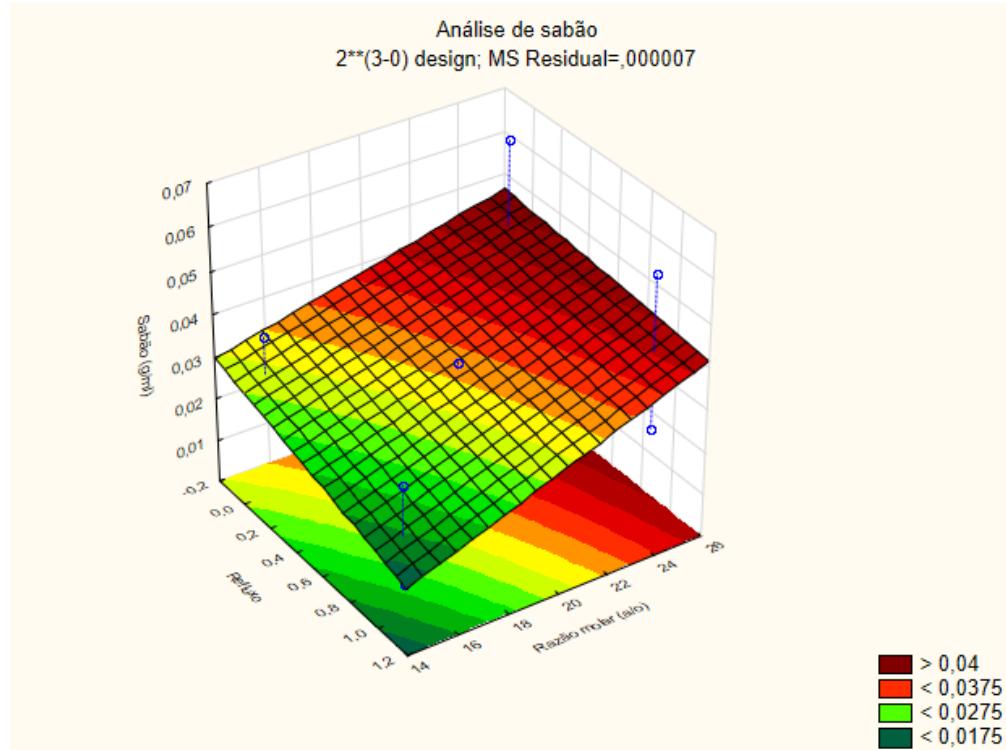
Figura 30 - Gráfico de resposta da concentração de sabão para carga térmica e refluxo.



Fonte: autor.

Na Figura 30 é possível observar a influência da carga térmica e do refluxo na formação de sabão. Segundo o trabalho de Eze, Harvey e Phan (2015) a constante cinética de reação na saponificação tem seu valor aumentado quando a temperatura aumenta, então a saponificação é favorecida por uma temperatura maior, o que também pode ser observado na Figura 30, onde **um aumento na carga térmica levou a uma maior concentração de sabão**, tanto para alto ou baixo refluxo. Na Figura 31 será verificada a influência do refluxo e razão molar.

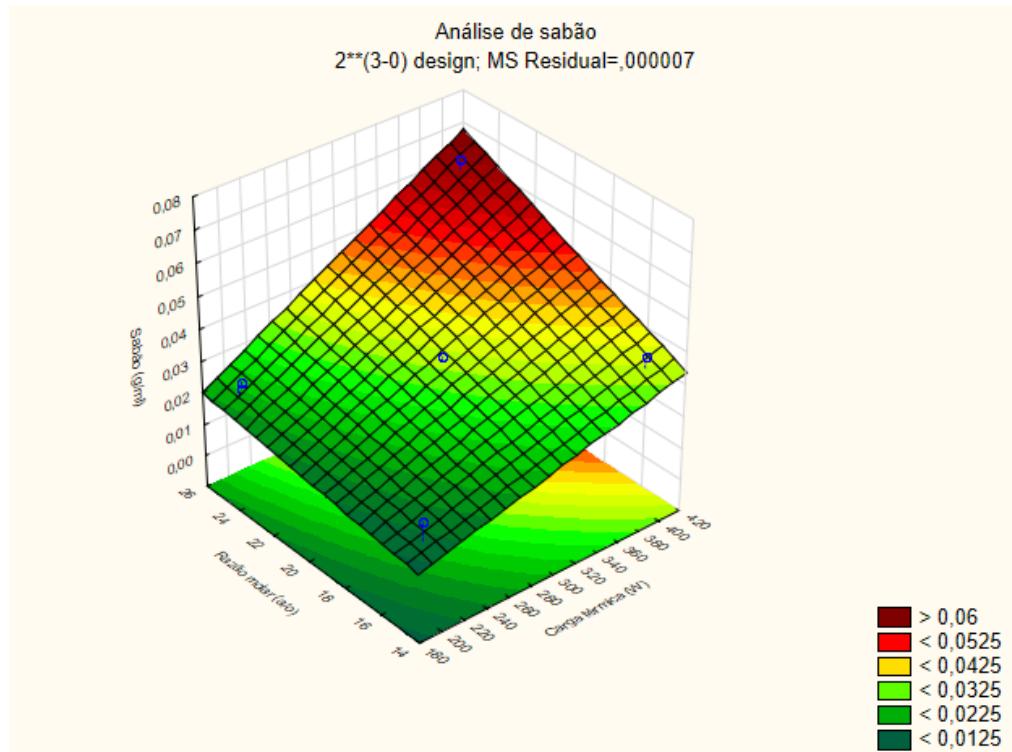
Figura 31 - Gráfico de resposta da concentração de sabão para o refluxo e a razão molar.



Fonte: autor.

Como visto no gráfico de Pareto da Figura 29 o fator direto que menos influencia a produção de sabão é o refluxo. Este fato se reproduz na Figura 31, onde o gradiente com relação a razão molar é muito pequeno. Estes efeitos podem ser observados na Figura 31, onde uma alta razão molar e um baixo refluxo promovem a formação de sabão. A alta razão molar promove uma maior conversão do óleo, conforme observado na Figura 21, sendo assim, **a condição que favorece a formação de biodiesel também favorece a formação de sabão**. Na Figura 32 será verificada a influência da carga térmica e razão molar.

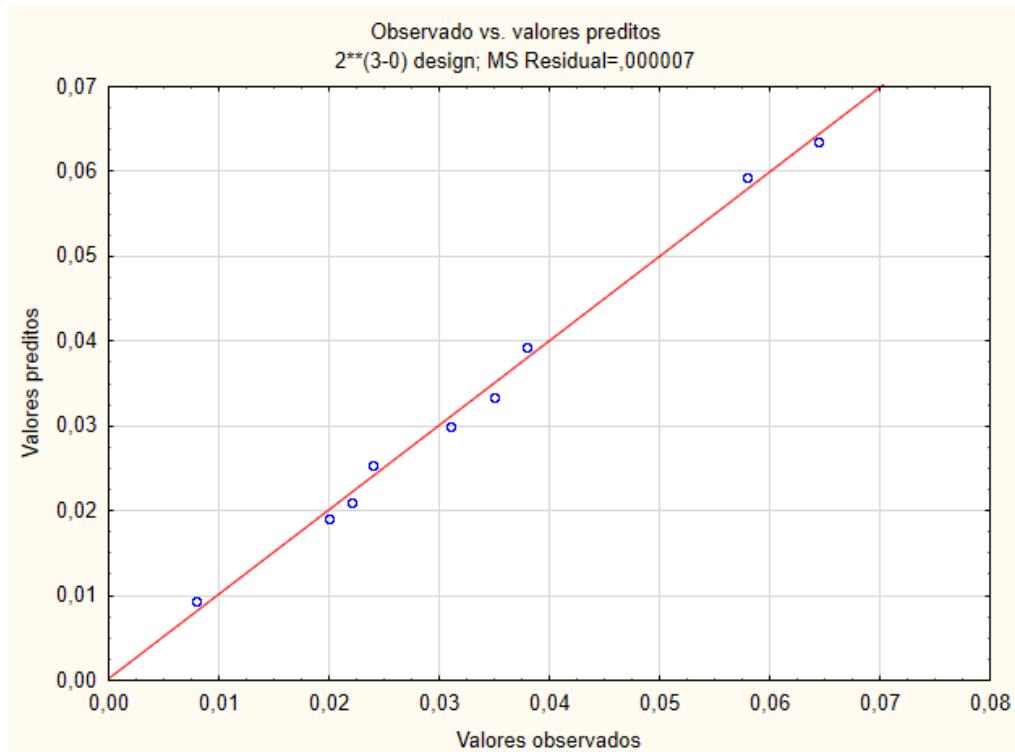
Figura 32 - Gráfico de resposta da concentração de sabão para a razão molar e a carga térmica.



Fonte: autor.

A Figura 32 reforça o que foi observado anteriormente, em que nas regiões de baixa razão molar existe pouco álcool aquecido disponível, o que torna o efeito na saponificação pouco intenso. Contudo com excesso de etanol, existe um maior transporte de calor ao longo da coluna, como mostrado no gráfico de Pareto da Figura 29. O efeito combinado da razão molar e da carga térmica é um efeito significativo. É possível observar que para valores de alta temperatura (alta carga térmica e alta razão molar) tem-se os valores máximos na concentração de sabão, como exposto por Eze, Harvey e Phan (2015). Na Figura 33 será mostrado o efeito dos resultados experimentais versus o calculado.

Figura 33 - Valores observados em função dos valores previstos.



Fonte: autor.

Na Figura 33 é possível observar a concordância entre os dados experimentais e o resultado do modelo estatístico, é possível observar a concordância entre o modelo e os dados experimentais, com desvio quadrático de $7 \cdot 10^{-6}$.

Segundo os fatores citados a saponificação é uma reação que também é favorecida pelos mesmos parâmetros operacionais que a transesterificação. Dessa forma, para contornar este fato existem algumas possibilidades. Segundo Simasatitkul et al. (2010) um aumento excessivo de temperatura pode causar vaporização completa do álcool no interior da coluna, reduzindo a taxa de reação de transesterificação, pois a reação de transesterificação acontece na fase líquida. Trabalhar com temperaturas pouco acima do ponto de ebulição do álcool pode garantir uma retenção líquida adequada, e ser vantajoso de diversas formas: garante aquecimento da coluna, promove o fluxo de vapor, reduz o custo energético, e reduz a quantia de sabão produzido, com relação a se trabalhar em condições bem acima da temperatura de saturação do álcool.

Outro fato é o de que a saponificação não influencia significativamente a saponificação, mas influencia a transesterificação, o aumento do refluxo pode favorecer a reação de produção do biodiesel, sem favorecer a produção de sabão.

6 CONCLUSÕES E PERSPECTIVAS

Diante dos resultados obtidos é possível definir por linhas gerais o que se pode concluir.

6.1 Conclusões

O estudo realizado neste trabalho levou à indicação das variáveis de operação que afetam de forma significativa a saponificação no processo para a produção de biodiesel via etanólica.

Os testes iniciais em batelada revelaram as condições e situações onde o processo de transesterificação é favorecido, assim como as condições que reduzem a concentração de sabão residual, conforme citados abaixo:

- A temperatura, concentração de catalisador e o tempo de residência em uma transesterificação utilizando etanol devem ser maiores que com metanol, e a quantidade de catalisador necessária deve ser superior a 1% mássico em relação ao óleo;
- O catalisador deve ser adicionado gradativamente, pois uma concentração local alta de catalisador pode formar uma grande quantidade de sabão;
- O método de purificação, no caso estudado, deve conter mais que 4 lavagens para ser eficiente. Garantindo uma quantidade mínima de álcool e glicerina residuais;
- O processo de purificação deve utilizar água de lavagem com pH entre 3,0 e 4,5. Valores superiores levam a formação intensa de sabão, valores inferiores fazem o biodiesel ser mais estável, porém sair da especificação;
- A decantação quando efetuada a 50°C pode reduzir o tempo de decantação à metade.

A coluna de destilação desenvolvida e automatizada se mostrou eficaz no estudo da produção de biodiesel. Utilizando materiais de baixo custo e com muito empenho e criatividade foi possível desenvolver uma coluna com sensores e atuadores que pode ser

controlada por computador e até mesmo por celular, onde os comandos podem ser dados inclusive via voz, devido ao desenvolvimento de uma assistente virtual de supervisão, EVA, que auxilia na operação e monitoramento da coluna, tornando todo o processo mais ágil e eficiente.

Com a coluna concluída e o estudo realizado em batelada foi possível utilizar as informações obtidas para traçar um experimento teste para avaliar a produção de biodiesel pela coluna, foi possível obter um biodiesel com 93% de teor de éster etílico. Após melhorias e ajustes feitos foi possível realizar um planejamento fatorial 2³.

Os resultados sobre a conversão e parâmetros que otimizem a produção de biodiesel por destilação reativa são bem estabelecidos na literatura. Porém informações sobre saponificação em colunas de destilação não são encontradas na literatura até o presente momento. Após as análises dos gráficos de superfície uma série de resultados foram obtidos com relação a saponificação, que podem ser expostos resumidamente da seguinte forma:

- O balanço mássico define que o catalisador adicionado pode sair da coluna intacto ou na forma de sabão.
- O teor de ester produzido é proporcional ao aumento do refluxo, aumento da carga térmica e o aumento da razão molar, e que o contrário também é verdade. E não existem pontos de inflexão dentro dos intervalos estudados.
- A quantidade de catalisador residual é proporcional ao aumento do refluxo, redução da carga térmica e a redução da razão molar, e que o contrário também é verdade. E não existem pontos de inflexão dentro dos intervalos estudados.
- A concentração de sabão residual é proporcional a redução do refluxo, aumento da carga térmica e a aumento da razão molar, e que o contrário também é verdade. E não existem pontos de inflexão dentro dos intervalos estudados
- A reação de saponificação ocorre tanto com o AGL, óleo e o biodiesel, tornando a influência da transesterificação mínima sobre a saponificação.
- Os parâmetros operacionais; carga térmica e razão molar álcool/óleo se mostraram estatisticamente influentes na concentração de sabão formado durante a transesterificação.

- Um aumento da carga térmica e da razão molar favorece o aquecimento da coluna e consequentemente a reação de saponificação.
- As condições que favorecem a formação de biodiesel também favorecem a formação de sabão.

6.2 Sugestões para trabalhos futuros

O estudo desenvolvido aqui abre diversas possibilidades para trabalhos futuros. As condições de produção de biodiesel utilizando a destilação reativa foram automatizadas ao ponto de se gastar o menor tempo possível na execução de experimentos, implementando assim o conceito de indústria 4.0. Propiciando mais facilidade no desenvolvimento de trabalhos futuros. Dentre as diversas possibilidades de novos trabalhos tem-se:

- Alguns parâmetros operacionais não foram avaliados, quando a sua influencia na reação de saponificação, tais como: diferentes concentrações de catalisador, teor de água no etanol utilizado, pressões diferentes da atmosférica. Na literatura é possível verificar que diversos trabalhos citam a dependência do teor final de sabão como função destes parâmetros em batelada.
- A utilização de óleos diferentes dos de algodão podem gerar resultados diferentes, principalmente devido a um índice de acidez diferente, outros óleos vegetais podem ser estudados.
- Simulações podem ser desenvolvidas para levar em consideração as informações obtidas e assim encontrar uma janela “ideal” de operação, minimizando a saponificação e maximizando a transesterificação. Levando em consideração também as condições que permitem a reação e a separação simultaneamente.
- O teor de sabão após o processo de purificação do biodiesel também pode ser quantificado, a influencia do processo produtivo e de purificação sobre o sabão no biodiesel final pode ser alvo de trabalhos futuros.

Com relação a automação os trabalhos continuam, estudos estão ainda sendo feitos utilizando a coluna de destilação reativa, e dia após dia melhorias são incorporadas. Como o desenvolvimento de conexões mais seguras, redução dos riscos de vazamento, melhoria na

robustez dos experimentos, assim como o desenvolvimento de melhorias no supervisório. Com relação a programação estudos futuros visam a melhor utilização de conceitos de inteligência artificial em processos químicos, como a predição de variáveis de processo; propiciando a previsão de condições de temperatura, pressão e etc. que fujam as condições desejadas. O uso de inteligência artificial tem como intuito o desenvolvimento de interação homem-máquina aperfeiçoada, permitindo a interação mais intuitiva do uso e operação de equipamentos, como demonstrado com EVA. Atualmente o uso de inteligência artificial vem sendo bastante estudada entre os acadêmicos, e é algo que cresce a cada ano.

REFERÊNCIAS

- AGREDA, V.; PARTIN, L. Reactive distillation process for the production of methyl acetate. **United States Patent Office**. Patente N° 4435595, 1984.
- AGREDA, V. H.; PARTIN, L. R.; HEISE, W. H. High-purity methyl acetate via reactive distillation. **Chemical Engineering Progress**. 86, 40–46. 1990.
- ALCANTARA, R.; AMORES, J.; CANOIRA, L.; FIDALGO, E.; FRANCO, M.; NAVARRO, A. Catalytic production of biodiesel from soy-bean oil, used frying oil and tallow. **Biomass Bioenergy**. 18, 6, 515-527. 2000.
- ANASTOPOULOS, G.; ZANNIKOU, Y.; STOURNAS, S.; KALLIGEROS, S. Transesterification of Vegetable Oils with Ethanol and Characterization of the Key Fuel Properties of Ethyl Esters. **Energies**, 2, 362-376. 2009.
- ATADASHI, I.M.; AROUA, M.K.; AZIZ, A. A. Biodiesel separation and purification: A review. **Renewable Energy**. 36, 2, 437-443. 2011.
- AUTOR DESCONHECIDO, High-Power Control: Arduino + TIP120 Transistor, <http://bildr.org/2011/03/high-power-control-with-arduino-and-tip120/>, 2011, acessado em 1/02/2016.
- BACKHAUS A. A. Continuous process for the manufacture of esters. **United States Patent Office**, Patente N° 1400849, USA, Baltimore, Maryland, 1921.
- BACKHAUS A. A. Apparatus for the manufacture of esters. **United States Patent Office**, Patente N° 1425624, USA, Baltimore, Maryland, 1922.
- BACKHAUS A. A. Process of producing high-grade esters. **United States Patent Office**, Patente N° 1454462, USA, Baltimore, Maryland, 1923.
- BALAT, M.; BALAT, H.; A critical review of bio-diesel as a vehicular fuel. **Energy Convers. Manage.** 49, 2727–2741. 2008.
- BOWMAN, M.; HILLIGOSS, D.; RASMUSSEN, S.; THOMAS, R. Biodiesel: a renewable and biodegradable fuel. **Hydrocarb. Process.** 85, 103–106. 2006.
- BARBOSA, D. C.; SERRA, T. M.; MENEGHETTI, S. M. P.; MENEGHETTI, M. R. Biodiesel production by ethanolysis of mixed castor and soybean oils. **Fuel**. 89, 12, 3791-3794, 2010.
- BARBOSA, D.; DOHERTY, M. F. A new set of composition variables for the representation of reactive phase diagrams. **Proc. Royal Society London**, 413, 1847, 459-464, 1987.
- BOON-ANUWAT, N.; KIATKITTIPONG, W.; AIOUACHE, F.; ASSABUMRUNGRAT, S., Process design of continuous biodiesel production by reactive distillation: Comparison

between homogeneous and heterogeneous catalysts. **Chemical Engineering and Processing** 92, 33–44. 2015.

BOYER, S. A. Scada: Supervisory Control And Data Acquisition, ISA, USA, 4^a Edição, 2009.

CHAVES A. T. C. A. Otimização do processo de produção de biodiesel etílico do óleo de girassol (*helianthus annus*) aplicando um delineamento composto central rotacional (DCCR). **UFPB**, dissertação, 2008.

CUNICO, M. W. M.; CUNICO, M. M.; MIGUEL, O. G.; ZAWADZKI, S. F.; PERALTA-ZAMORA, P.; VOLPATO, N. Planejamento fatorial: uma ferramenta estatística valiosa para a definição de parâmetros experimentais empregados na pesquisa científica. **UFPR**, revista digital de periódicos, v. 9, n. 1, 2008.

DALE, B. Greening the chemical industry: research and development priorities for bio-based industrial products. **J. Chem. Technol. Biotechnol.** 78, 10, 1093–1103. 2003.

DEMIRBAS, A. Biodiesel fuels from vegetable oils via catalytic and non-catalytic supercritical alcohol transesterifications and others methods: a survey. **Energy Convers Manage**, 44, 13, 2093-109. 2003

DIMIAN A. C.; BILDEA C. S.; OMOTA F.; KISS A. A. Innovative process for fatty acid esters by dual reactive distillation. **Computers and Chemical Engineering**, 33, 3, 743-750. 2008.

DING, H.; XIANG, W.; SONG, N.; LIU, C.; YUAN, X. Hydrodynamic Behavior and Residence Time Distribution of Industrial-Scale Bale Packings. **Chemical Engineering Technology**, 37, 7, 1127–1136. 2014.

DIZGE N.; AYDINER C.; IMER D.Y.; BAYRAMOGLU M.; TANRISEVEN A.; KESKINLER B. Biodiesel production from sunflower, soybean, and waste cooking oils by transesterification using lipase immobilized onto a novel microporous polymer. **Bioresour Technol**, 100, 6, 1983–91. 2009

DOHERTY, M.F. Design and synthesis of reactive separation systems. Final report. **Massachusetts Univ.**, Amherst, MA (United States). Dept. of Chemical Engineering, 1992.

DUBILIER, C. Application guide snubber capacitors, www.cde.com, 2011, acessado em 09/07/2017.

ENCINAR, J.M.; GONZALEZ, J.F.; RODRIGUEZ-REINARES, A. Biodiesel from used frying oil. Variables affecting the yields and characteristics of the biodiesel. **Ind. Eng. Chem. Res.** 44, 5491–5499. 2005.

EZE, V.C.; HARVEY, A. P.; PHAN, A.N. Determination of the kinetics of biodiesel saponification in alcoholic hydroxide solutions. **Fuel**, 140, 724-730, 2015.

EZE V. C.; PHAN A. N.; HARVEY A. P. A more robust model of the biodiesel reaction, allowing identification of process conditions for significantly enhanced rate and water tolerance. **Bioresource Technology**, 156, 222–231, 2014.

FRAGA, I. M.; PENHA, B. F.; JACOBINA, M. C. M. Estudo comparativo de três diferentes equações empregadas no cálculo da massa molecular de óleos e gorduras, **Congresso brasileiro de química**, 2014.

FRANCHI, C. M. Controle de Processos Industriais - Princípios e Aplicações, **Editora Érica**, 2008

GASPARINI, F. Avaliação e Adaptação das Condições da EN 14103 para Quantificação de Ésteres em Biodiesel Etílicos puros de Soja, Babaçu, Palma e Sebo Bovino e em suas Misturas, **UNESP**, dissertação de mestrado, 2010.

GHAEMI A.; SHAHROSSEINI, S.; MARAGHEH, M.G.; Nonequilibrium modeling of reactive absorption processes, **Chemical Engineering Communications**. 196, 9, 1076–1089. 2009.

GERPEN, J. V.; SHANKS, B.; PRUSZKO, R.; CLEMENTS, D.; KNOTHE, G. Biodiesel production technology. 1617 Cole Boulevard, Golden, CO: **National Renewable Energy Laboratory**; 2004.

GERPEN J. V. Biodiesel processing and production. **Fuel Processing Technology**. 86, 10, 1097 – 1107. 2005.

GOERING, C.E.; SCHWAB, A.W.; DAUGHERTY, M. J.; PRYDE, E. H.; HEAKIN, A. J. Fuel properties of eleven oils. **Trans ASAE**, 25, 6, 1472-1483, 1982.

HANNA, M. A.; ISOM, L.; CAMPBELL, J. Biodiesel: current perspectives and future. **J. Sci. Ind. Res.** 64, 854–857. 2005.

HE B. B. A.; SINGH, A. P.; THOMPSON, J. C. novel continuous-flow reactor using a reactive distillation technique for biodiesel production. **National institute for advanced transportation technology**, 49, 107-112, 2006.

HERNÁNDEZ, J. G. S.; HERNÁNDEZ, S.; PETRICOLET, A. B. Reactive distillation: A review of optimal design using deterministic and stochastic techniques. **Chem. Eng. And Proc.: Proc. Intensification**. 97, 134-143. 2015.

KISS, A. A.; BILDEA, C. S. A review of biodiesel production by integrated reactive separation technologies, **J Chem Technol Biotechnol**, 87, 7, 861–879, 2012.

KISS, A. A.; DIMIAN, A. C.; ROTENBERG, G. Biodiesel by catalytic reactive distillation powered by metal oxides. **Energy Fuels** 22, 1, 598–604. 2008.

KISS, A. A.; DIMIAN, A. C.; ROTENBERG, G. Solid acid catalysts for biodiesel production – towards sustainable energy. **Adv. Synth. Catal.** 348, 1-2, 75–81. 2006a.

KISS, A. A.; ROTHENBERG, G.; DIMIAN, A.C.; OMOTA, F. The heterogeneous advantage: biodiesel by catalytic reactive distillation. **Top. Catal.** 40, 1-4, 141–150. 2006b.

KITSON, P. J.; GLATZEL, S.; CHEN, W.; LIN, C. G.; SONG, Y. F.; CRONIN, L. 3D printing of versatile reactionware for chemical synthesis. **Nature Protocols.** 11, 5, 920-936. 2016.

KREUL, L.U.; GORAK, A.; BARTON, P.I. Modeling of homogeneous reactive separation processes in packed columns. **Chemical Engineering Science.** 54, 1, 19–34. 1999.

KUBÍNOVA, S.; SLEGR, J. ChemDuino: Adapting Arduino for Low-Cost Chemical Measurements in Lecture and Laboratory. **J. Chem. Educ.** 92, 10, 1751–1753, 2015

KULKARNI, M. G.; DALAI, A. K. Waste cooking oil-an economical source for biodiesel: a review. **Ind. Eng. Chem. Res.** 45, 9, 2901–2913. 2006.

KULPRATHIPANJA S. Reactive Separation Processes. **Taylor & Francis**, New York, 2002.

KUSDIANA, D.; SAKA, S. Two-step preparation for catalyst-free biodiesel fuel production – hydrolysis and methyl esterification. **Appl Biochem Biotechnol** 115, 1-3, 781–791. 2004.

LEUNG, D. Y. C.; WU, X.; LEUNG, M. K. H. A review on biodiesel production using catalyzed transesterification, **Applied Energy**, 87, 4, 1083–1095, 2010.

LIMA, D. V. F.; SILVA, J. P.; AZEVEDO, F. G.; CAVALCANTI, D. E. S.; OLIVEIRA, W. A. L. Desenvolvimento de softwares de visualização avançada (3d) de equipamentos. **Jornada de Iniciação Científica**, recife. 2015.

LOTERO, E.; LIU, Y. J.; LOPEZ, D. E.; SUWANNAKARN, K.; BRUCE, D. A.; GOODWIN, J. G. Synthesis of biodiesel via acid catalysis. **Ind. Eng. Chem. Res.** 44, 14, 5353–5363. 2005.

MACCARANA, Y.; COLOGNI, A.; PREVIDI, F. Design of a HTML5 SCADA system, 2nd International Forum on research and Technologies for Society and Industry Leveraging a better tomorrow, Bologna, Italy, 2016

MACHADO, G. D. Produção de biodiesel por esterificação em coluna de destilação reativa: Modelagem matemática. Dissertação (Mestrado em Engenharia Química), **Universidade Estadual de Maringá**, UEM, Maringá, Paraná, 2009.

MANSOURI, S. S.; SALES-CRUZ, M.; HUUSOM, J. K.; GANI, R. Integrated Process Design and Control of Multi-element Reactive Distillation Processes. **IFAC-PapersOnLine**, 49, 7, 735-740. 2016

MARCHETTI, J. M.;ERRAZU, A. F. Technoeconomic study of supercritical biodiesel production plant. **Energy Conversion and Management**, 49, 8, 2160–2164. 2008

MEHER, L. C.; VIDYASAGAR, D.; NAIK, S. Technical aspects of biodiesel production by trans-esterification: a review. **Renewable Sustainable Energy Rev.** 10, 3, 248–268. 2006.

- MEHER, L. C.; KULKARNI, M. G.; DALAI, A. K.; NAIK, S. N. Transesterification of karanja (*Pongamia pinnata*) oil by solid basic catalysts. **Eur J Lipid Sci Technol.** 108, 5, 389-397. 2006.
- MINAMI, E.; SAKA, S. Kinetics of hydrolysis and methyl esterification for biodiesel production in two-step supercritical methanol process. **Fuel** 85, 17-18, 2479–2483. 2006.
- MINERVA, R.; BIRU, A.; ROTONDI, D. Towards a definition of the Internet of Things (IoT). **IEEE Internet Initiative**, n. 1, 2015
- MUEANMAS, C.; PRASERTSIT, K.; TONGURAI, C. Transesterification of triolein with methanol in reactive distillation column: simulation studies. **International Journal of Chemical Reactor Engineering**, 8, 141, 2010.
- NEJATIMOHARRAMI, F.; FAÍNA, A.; CEJKOVA, J.; HANCZYC, MM.; STOY K. Robotic Automation to Augment Quality of Artificial Chemical Life Experiments. **Proceedings of the Artificial Life Conference**, 634-635, 2016.
- NOBLE, D. Forces of Production: A Social History of Industrial Automation, **Routledge**; 1 edition, 2011.
- NOERES, C.; KENIG, E. Y.; GORAK, A. Modelling of reactive separation processes: reactive absorption and reactive distillation. **Chem Eng Process.** 42, 3, 157–178. 2003.
- NOSHADI, I.; AMIN N. A. S.; PARNAS, R. S. Continuous production of biodiesel from waste cooking oil in a reactive distillation column catalyzed by solid heteropolyacid: Optimization using response surface methodology (RSM). **Fuel**. 94, 156-164. 2012.
- NGUYEN, N.; DEMIREL, Y. Using thermally coupled reactive distillation columns in biodiesel production. **Energy**. 36, 8, 4838–4847. 2011.
- PHANKOSOL, S.; SUDAPRASERT, K.; KRISNANGKURA, K. Estimation Density and Viscosity of Ethyl-Biodiesel from Saponification Number and Iodine Value. **Siam Physics Congress**. 2015.
- PISARENKO Y. A.; SERAFIMOV, L. A.; CARDONA, C. A.; EFREMOV, D. L.; SHUWALOV, A. S. Reactive distillation design: analysis of the process statics, **Reviews in Chemical Engineering**. 17, 4, 253-260. 2011.
- PODDAR, T.; JAGANNATH, A.; ALMANSOORI, A. Use of reactive distillation in biodiesel production: A simulation-based comparison of energy requirements and profitability indicators. **Applied Energy**. 185, 2, 985–997. 2017.
- RAMADHAS, A. S.; JAYARAJ, S.; MURALEEDHARAN, C. Biodiesel production from high FFA rubber seed oil. **Fuel**. 84, 4, 335–340. 2005.
- RASTEGARI, H.; GHAZIASKAR, H. S.; YALPANI, M. SHAFIEI, A. Development of a Continuous System Based on Azeotropic Reactive Distillation to Enhance Triacetin

Selectivity in Glycerol Esterification with Acetic Acid. **Energy Fuels**, 31, 8, 8256–8262. 2017.

RAY, N. M.; RAY, A. K. Determination of adsorption and kinetic parameters for methyl oleate (biodiesel) esterification reaction catalyzed by Amberlyst 15 resin. **The Canadian Journal of Chemical Engineering**. 94, 4, 738–744. 2016.

REYERO, I.; ARZAMENDI, G.; ZABALA, S.; GANDÍA, L. M. Kinetics of the NaOH-catalyzed transesterification of sunflower oil with ethanol to produce biodiesel. **Fuel Processing Technology**. 129, 147–155. 2015.

SATHYANARAYANA, K.; KUMAR, G. V. V. R. Evolution of Computer Graphics and Its impact on Engineering Product Development, **Fifth International Conference on Computer Graphics, Imaging and Visualisation**, 2008.

SAYURI, T. Planejamento de Experimentos (DOE). **USP – NUMA**. 2008.

SEADER, J. D.; HENLEY, E. J. Separation process principles. **John Wiley & Sons**, 631 – 635, 1998.

SEBORG, D. E.; EDGAR T. F.; MELLICHAMP, D. A.; DOYLE III F. J. Process Dynamics an Control, **Ed. Willey**, 4ed, 2016.

SHARMA, A.; MONDAL, S.; MONDAL, A. K.; BAKSI, S.; PATEL, R. K.; CHU, W. S. 3D printing: It's microfluidic functions and environmental impacts, **Inter. J. of Precision Eng. and Manufacturing-Green Technol.** 4, 3, 323–334. 2017.

SHEEHAN, J.; CAMOBRECO, V.; DUFFIELD, J.; GRABOSKI M.; SHAPOURI H. life cycle inventory of biodiesel and petroleum diesel for use in an urban bus. **U.S. department of energy's office of fuels development**, 1998.

SILVA, G. F.; CAMARGO, F. L.; FERREIRA, A. L.O. Application of response surface methodology for optimization of biodiesel production by transesterification of soybean oil with ethanol. **Fuel Processing Technology**. 92, 3, 407–413. 2011.

SILVA, J. P. Avaliação por planejamento fatorial da produção de biodiesel via destilação reativa, **UFPE, DEQ**, dissertação, 2013.

SILVA, J. P.; SOUZA, T. P. C.; CAVALCANTI, D. E. S.; SILVA, J. M. F.; TSCHOEKE, I. C. P.; SANTOS, B. F. Desenvolvimento de software 3d para controle de processos, aplicado a produção de biodiesel. **COBEQ**, Fortaleza, 2016.

SILVA, N. D.; SANTANDER, C. M.; BATISTELLA, C. M.; MACIEL, R.; MACIEL, M. R. W. Biodiesel production from integration between reaction and separation system: reactive distillation process. **Appl Biochem Biotechnol**. 161, 1-8, 245–254, 2010.

SIMASATITKUL, L.; SIRICHARNSAKUNCHAI, P.; PATCHARAVORACHOT, Y.; ASSABUMRUNGRAT, S.; ARPORNWICHANOP, A. Reactive distillation for biodiesel

production from soybean oil. **Korean Journal of Chemical Engineering.** 28, 3, 649–655. 2011.

SMITH, H. A.; LEVENSON, H. S. Kinetics of the esterification of cyclohexanoic acid and saponification of its ethyl ester. **Journal of the American Chemical Society.** 62, 2733-2735. 1940.

SONG, W.; HUSS, R. S.; DOHERTY, M. F.; MALONE, M. F. Discovery of a reactive azeotrope. **Nature,** 388, 561–563. 1997.

SOUZA, T. P. C.; STRAGEVITCH, L.; KNOECHELMANN, A.; PACHECO, J. G. A.; SILVA, J. M. F. Simulation and preliminary economic assessment of a biodiesel plant and comparison with reactive distillation. **Fuel Processing Technology.** 123, 75–81. 2014.

SUNDMACHER, K.; KIENLE, A. Reactive distillation, Status and future directions, **Wiley-VHC Verlag**, Germany, 2002.

TAHER, H.; AL-ZUHAIR, S. The use of alternative solvents in enzymatic biodiesel production: a review. **Biofuels bioproducts & biorefining,** 11, 1, 168–194. 2016.

TAN, T.; SHANG, F.; ZHANG, X. Current development of biorefinery in China. **Biotechnol Adv.** 28, 5, 543-555. 2010.

TSCHOEKE, I. C. P.; SOUTO, M. E. M. M.; LEITAO, N. P. P. A.; CAVALCANTI, J. J. V. F.; SILVA, J. P.; SOUZA, T. P. C. Sistema de controle automático para fermentação de cervejas. **COBEQ**, Fortaleza. 2016(a).

TSCHOEKE, I. C. P.; CAVALCANTI, J. V. F. L.; DIAS, F. F. S.; SANTOS, A. F. M. S.; SOUZA, T. P. C.; SILVA, J. P. Sistema de controle automático para biodigestor. **COBEQ**, Fortaleza. 2016(b).

UNG S.; DOHERTY M. F. Necessary and sufficient conditions for reactive azeotropes in multireaction mixtures. **AIChE Journal,** 41, 11, 2383-2392. 1995.

VENIMADHAVAN, G.; BUZAD, G.; DOHERTY, M. F.; MALONE, M. F. Effect of kinetics on residue curve maps for reactive distillation. **AIChE Journal,** 40, 11, 1814-1824. 1994.

WIGGERS, V. R.; ZONTA, G. R.; FRANÇA, A. P.; SCHARF, D. R.; SIMIONATTO, E. L.; ENDER L.; MEIER H. F. Challenges associated with choosing operational conditions for triglyceride thermal cracking aiming to improve biofuel quality. **Fuel,** 107, 601–608. 2013.

XIAO, Y.; LI, H.; XIAO, G.; GAO, L.; PAN X. Simulation of the Catalytic Reactive Distillation Process for Biodiesel Production via Transesterification. **2013 International Conference on Materials for Renewable Energy and Environment.** Chengdu, China. 2013.

YEE, K. F.; KANSEDO, J.; LEE, K. T. Biodiesel production from palm oil via heterogeneous transesterification: Optimization study. **Chemical engineering communications**, 197, 12, 1597-1611, 2010.

ZHOU, C. H.; BELTRAMINI, J. N.; FAN, Y. X.; LU, G. Q. Chemoselective catalytic conversion of glycerol as a biorenewable source to valuable commodity chemicals. **Chemical society reviews**. 37, 527-549. 2007.

APÊNDICE A - SCRIPT NO ARDUINO

```

1. // ----- PROGRAMA DE CONTROLE DA COLUNA DE DESTILAÇÃO REATIVA -----
2. //xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
3.
4. //Desenvolvido por Josivan Pedro e Thiberio Souza
5. //De 2015 a 2017
6.
7. #include <OneWire.h> //Se importan las librerías
8. #include <DallasTemperature.h>
9.
10. int Pins[] = {52,44,46,48,50,32,11,12};
11. bool PWMPins[] = {false,false,false,false,false,false,true,true};
12. int PinsIntensity[] = {1,1,1,1,1,1,0,0};
13. //52 Válvula de refluxo
14. //44 Válvula de catalisador
15. //46 Válvula de catalisador
16. //48 Válvula de catalisador
17. //50 Refervedor
18. //32 Bomba de resfriamento
19. //13 Bomba do alcool
20. //12 Bombas de óleo
21.
22. int EquipNumber;
23. int EquipIntensity;
24.
25. //Inicialização dos sensores de presão
26. int sensorPin1 = A0;      // Select input pin for the potentiometer
27. int sensorPin2 = A1;      // Select input pin for the potentiometer
28. int sensorValue1 = 0;     // Variable stores value coming from the sensor
29. float P1 =0;
30. int sensorValue2 = 0;     // Variable stores value coming from the sensor
31. float P2 =0;
32.
33. //Inicialização dos sensores de temperatura
34.
35. float T1 =0;
36. float T2 =0;
37. float T3 =0;
38.
39. OneWire ds_1(4);    //sensor 1 on pin 4
40. OneWire ds_2(5);    //sensor 2 on pin 5
41. OneWire ds_3(6);    //sensor 2 on pin 5
42.
43. DallasTemperature sensor_1(&ds_1);
44. DallasTemperature sensor_2(&ds_2);
45. DallasTemperature sensor_3(&ds_3);
46.

```

```
47. //Inicialização dos sensores de fluxo
48. volatile int NbTopsFanC; //measuring the rising edges of the signal
49. float CalcC;
50. int hallsensorC = 2; //The pin location of the sensor
51.
52. volatile int NbTopsFanB; //measuring the rising edges of the signal
53. float CalcB;
54. int hallsensorB = 3; //The pin location of the sensor
55.
56. //Inicialização dos sensores de corrente
57. const int sensorPin = A5;
58. float sensorValue_aux = 0;
59. float sensorValue = 0;
60. float currentValue = 0;
61. float voltporUnidade = 0.0048828125;
62.
63.
64. void setup()
65. {
66.     Serial.begin(9600); //ConFigurando taxa de transferência porta serial
67.
68.     for (int i=0;i<8;i++){//16 possíveis saídas, se alterar, corrigir
69.         delay(150);
70.         pinMode(Pins[i], OUTPUT); //Declara que o pino 12 do arduino é de Saída. Vai mandar dados
71.         , energia...
72.         if(PWMPins[i]==true){analogWrite(Pins[i],PinsIntensity[i]);}
73.         if(PWMPins[i]==false){digitalWrite(Pins[i],PinsIntensity[i]);}
74.     }
75.
76.     //Sensores de fluxo
77.     //2 e 3
78.     pinMode(hallsensorC, INPUT); //initializes digital pin 2 as an input
79.     attachInterrupt(0, rpmC, RISING); //and the interrupt is attached
80.
81.     pinMode(hallsensorB, INPUT); //initializes digital pin 3 as an input
82.     attachInterrupt(1, rpmB, RISING); //and the interrupt is attached
83.
84.     //iniciar a biblioteca dos sensores de temperatura
85.     sensor_1.begin();
86.     sensor_2.begin();
87.     sensor_3.begin();
88.
89. }
90.
91. void loop()
92. { // -----
93.     //digitalWrite(30,HIGH);
```

```

95.      //digitalWrite(36,HIGH);
96.      //Confirmação
97.      Serial.println(" ");
98.      Serial.println("Arduino");
99.      Serial.println(" ");
100.     delay(20);
101.
102.     //Lê os sinais que chegam do computador
103.     ReadFromSerial();
104.
105.    //Sensor de fluxo
106.    //sensor 1
107.    NbTopsFanC = 0;    //Set NbTops to 0 ready for calculations
108.    NbTopsFanB = 0;
109.    sei();           //Enables interrupts
110.    delay (1000);   //Wait 1 second
111.    cli();          //Disable interrupts
112.    delay(20);
113.
114.
115.    CalcC = ((float)NbTopsFanC * 60 / 2500); //((Pulse frequency x 60) / 2500 = flow rate in L/min
116.    //Óleo
117.    if (CalcC<=0.275 && CalcC!=0){CalcC = -3.2405*CalcC*CalcC+1.6311*CalcC+0.0787*(1-exp(-
118.      200*CalcC));} //Correção para baixas vazões
119.    if (CalcC<=0.07){CalcC = (float)PinsIntensity[7] *(-0.12 /(20-56))+ (0.12+(0.12*56)/(20-
120.      56));} //Correção para abaixo do limite de detecção: 0.12l/min (limite de detecção) 56 (Intensidade PWM no limite) 20 valor PWM de corte
121.    if (CalcC<0.00){CalcC =0.0;}
122.    Serial.println("-");
123.    Serial.print("Fluxo04=");
124.    Serial.println(CalcC);
125.    delay(20);
126.
127.    //sensor 2
128.    CalcB = ((float)NbTopsFanB * 60 / 5432); //((Pulse frequency x 60) / 5432 = flow rate in L/min
129.    //Alcool
130.    if (CalcB<=1){CalcB = ((float)NbTopsFanB * 60 / 4836);} //Correção para baixas vazões
131.    if (CalcB<=0.07){CalcB = (float)PinsIntensity[6] *(-0.07 /(18-23))+ (0.07+(0.07*23)/(18-
132.      23));} //Correção para abaixo do limite de detecção: 0.07l/min (limite de detecção) 23 (Intensidade PWM no limite) 18 valor PWM de corte
133.    if (CalcB<0.00){CalcB =0.0;}
134.    Serial.println("-");
135.    Serial.print("Fluxo05=");
136.    Serial.println(CalcB);
137.    delay(20);
138.
139.    //Sensor de pressão
140.    sensorValue1 = analogRead(sensorPin1); // Read sensor
141.    P1=100+(((float)sensorValue1/1023)-0.04)/0.00369;
142.
```

```
138.     sensorValue2 = analogRead(sensorPin2); // Read sensor
139.     P2=100+((float)sensorValue2/1023)-0.04)/0.00369;
140.
141.     Serial.println("-");
142.     Serial.print("Pressao02=");
143.     Serial.println(P1);
144.     delay(20);
145.
146.     Serial.println("-");
147.     Serial.print("Pressao03=");
148.     Serial.println(P2);
149.     delay(20);
150.
151. //mostrar resultados dos sensores
152.     sensor_1.requestTemperatures();
153.     sensor_2.requestTemperatures();
154.     sensor_3.requestTemperatures();
155.     T1=sensor_1.getTempCByIndex(0);
156.     T2=sensor_2.getTempCByIndex(0);
157.     T3=sensor_3.getTempCByIndex(0);
158.
159.     Serial.println("-");
160.     Serial.print("Termometro00=");
161.     Serial.println(T1);
162.     delay(20);
163.     Serial.println("-");
164.     Serial.print("Termometro01=");
165.     Serial.println(T2);
166.     delay(20);
167.     Serial.println("-");
168.     Serial.print("Termometro11=");
169.     Serial.println(T3);
170.     delay(20);
171.
172. //Sensor de corrente
173. for(int i=500; i>0; i--)
174.
175. {
176.     sensorValue_aux = (analogRead(sensorPin) -511);
177.     sensorValue += pow(sensorValue_aux,2);
178. }
179. sensorValue = (sqrt(sensorValue/ 500)) * voltsporUnidade;
180. currentValue = (sensorValue/66)*1000;
181. currentValue = 220*currentValue; //220v e corrente alternada
182. if (currentValue<200){currentValue=0;}//correção para ruidos quando desligado
183. ...
184.
```

APÊNDICE B - SCRIPT PRINCIPAL DO SUPERVISÓRIO

```

1.  using UnityEngine;
2.  using System.Collections;
3.  //Interface de usuario
4.  using UnityEngine.UI;
5.  //Conexão com as portas
6.  using System.IO.Ports;
7.  //Conseguir Data e Hora
8.  using System;
9.  //criar pastas
10. using System.IO;
11. //chamar executavel
12. using System.Diagnostics;
13. //Arquivos de texto
14. using System.Text;
15. //E-mail
16. using System.Net;
17. using System.Net.Mail;
18. //Cubic interpolation
19. using TestMySpline;
20.
21.
22. public class CentralArduino : MonoBehaviour {
23.
24.     public string ProjectName;    //Nome do equipamento
25.     public Eva ScriptEva;
26.     private bool ClipboardOcupado = false;
27.     private bool MicrofoneOcupado = false;
28.
29.     public static bool Speaking=true;    //Fala de Eva, ativada ou nao pelo usuario.
30.     public static bool ControlSpeaking=true; //Controle pelo programa
31.
32.     public static SerialPort sp; //Porta do Arduino
33.     public string[] ports;        //Todas as portas
34.
35.     public bool Demonstrative = false;
36.     public bool Arduino=false;      //Arduino Conectado
37.     public bool ReadToGO = false;    //tudo pronto
38.     private bool ReadToGOMark = false; //tudo pronto marcador
39.     public bool Simulation;        //Modo Simulação
40.     public bool RemoteControl=false; //Modo controle remoto
41.     private int PassoCounter;       //Contador para geração de passo simulado
42.     public string COMPortResult;    //Resultado da porta COM
43.     public string DadoSimulado;    //Dado simulado
44.     public string RemoteData;      //Dado simulado
45.
46.     public AudioSource[] Audios;
```

```

47.     public static AudioSource[] StaticAudios;
48.
49.     public bool HUD=true;
50.     public GameObject[] HUDObjects;
51.
52.     //EQUIPAMENTOS
53.     public string[] Equipamentos;           //Nomes dos equipamentos
54.     public string[] DescricaoEquipamentos; //Descrição dos equipamentos
55.     public bool[] StatusEquipamentos;      //Status dos equipamentos, Ligado\Desligado
56.     public Coroutine[] CorrotinasEquipamentos;
57.     public bool[] MarkStatusEquipamentos;   //Marcador do Status dos equipamentos, Ligado\
58.                                         Desligado
59.     public float[] TempoEquipamentos;       //Tempos de operação dos equipamentos
60.     public bool[] EquipamentosModulados;    //Equipamentos que usam PWM
61.     public bool[] EquipamentosLargePWM;     //Equipamentos que usam PWM simulado de baixa
62.                                         frequencia
63.     public bool[] RealStatusPWM;            //um equipamento de large PWM esta ligado ou d
64.                                         esligado
65.     public float[] ValoresEquipamentos;    //Valores atualizados dos equipamentos
66.     public float[] ValoresAntigosEquipamentos; //Valores antigos dos equipamentos
67.     public float[] LimiteMinEquipamentos;   // Limites minimos de operad
68.     public float[] LimiteMaxEquipamentos;   // Limites maximos de segurança dos sensores
69.
70.     public GameObject[] Indicador3DEquipamentos; //Indicadores 3D
71.     public TextMesh[] Titulo3DIndEquipamentos; //titulos dos indicadores 3D
72.     public TextMesh[] Status3DIndEquipamentos; //Status dos indicadores 3D
73.     public TextMesh[] Info1Ind3DEquipamentos; //Informação adicional 1 dos indicadores 3D
74.     public TextMesh[] Info2Ind3DEquipamentos; //Informação adicional 2 dos indicadores 3D
75.
76.     //BATELADA
77.     public bool Batelada=false;             //Modo Batelada
78.     public float TempoInicioBatelada;      //Tempo do inicio da batelada
79.     private bool MarkBatelada=false;        //Marcador do inicio da batelada
80.     public bool[] EquipConcluidoBatelada;   //Equipamentos em batelada
81.     public string[] TempoInicialEquip;     //Tempo inicial para cada equipamento
82.     public string[] TempoCicloEquip;
83.     public string[] TempoHUEDEquipamentos; //Tempos de operação no HUD
84.     public string[] TempoEquipamentosInicial; //Tempos inicial de operação no HUD
85.     public bool HUDChanged;                //O HUD foi modificado
86.     public int ContHUDChange;              //Contador de atualização
87.     public float TimeHUDChange;            //Tempo para o contador de atualização
88.
89.     //SENSORES
90.     public string[] Sensores;              //Nomes dos Sensores
91.     public string[] DescricaoSensor;       //Descrição dos Sensores
92.     public float[] ValoresAtuaisSensores; //Ultimo valor para cada sensor
93.     public string[] TipoSensores;          //Tipo dos Sensores

```

```

94.     public string[] UnidadeSensores;           //Unidade das medidas dos Sensores
95.     public string[] UnidadeFaladaSensores; //Unidade falada das medidas dos Sensores
96.     public float[, ,] ValoresSensores;        //Resultados dos sensores [sensor I] [intervalos de
97.             tempo J (10min, 1h, 1dia, 1semana)] [Tempos da medição K (10 seg, 1 min, 24 min, 168 min (2h4
98.             8min))]
99.     public float[, ,] TempoValoresSensores; //Escalas de tempo para os valores registrados
100.    public bool[] SensorPrincipal;
101.    public bool[] SensoresImportantes;
102.    //Intervalos utilizados
103.    public float TVerificarSensores=30;
104.    public bool VerifiqueTempoAtualizacao=false;
105.    public float[] SensTempoUltimaAtualizacao;
106.    public float[] SensTempoDesdeUltimaAtualizacao;
107.    private float TAddResultsMark=7f;
108.    private float TAddResults=17f;
109.    private float TSendDataMark=3f;
110.    public int SendDataCounter=0;
111.    private float T10minMark=10f;
112.    private float T10min=10f;
113.    private float T1horMark=60f;
114.    private float T1hor=60f;
115.    private float T1diaMark=1440f;
116.    private float T1dia=1440f;
117.    private float T1semMark=10080f;
118.    public TextMesh[] TituloHUDSensores;      //Titulos dos Mostradores 3D do titulo
119.    public TextMesh[] ValorHUDSensores;       //Valor nos Mostradores 3D dos sensores
120.    public TextMesh[] UnidadeHUDSensores;     //Unidade dos Mostradores 3D dos sensores
121.    public GameObject[] Obj3DSensores;
122.    public TextMesh[] ValorObj3DSensores;     //Valor no objeto no indicador 3D
123.    public TextMesh[] StatusObj3DSensores;   //Status de cada sensor no indicador 3D
124.    public TextMesh[] TituloObj3DSensores;    //Titulo de cada sensor no indicador 3D
125.    public float[][] ValoresSensoresDesde0Inicio; //Valores dos sensores desde o inicio (100
126.               pontos)
127.    public float[] LimiteMinSensores;          // Limites minimos de segurança dos sensores
128.    public float[] LimiteMaxSensores;          // Limites maximos de segurança dos sensores
129.    public GameObject[] CircIndiSensores;      //Medidores circulares do sensor I
130.    public GameObject[] MedIndiSensores;       //Medidores circulares da media no sensor I
131.
132.    public float[] ValoresMediaSensores; //media ao longo da ultima hora do sensor I
133.    public String[] MediaDataSensores; //data da primeira medição da media
134.    public float[] ValoresMinSensores; //valor minimo do sensor I desde o inicio
135.    public String[] MinDataSensores; //data do momento de minimo
136.    public float[] ValoresMaxSensores; //valor maximo do sensor I desde o inicio
137.    public String[] MaxDataSensores; //data do momento de maximo
138.    public bool[] RelatFaladoSensorI; //Utilizar relatorio falado para o sensor I
139.
140.    public Texture Save_Icon;                 //Textura disquete

```

```
141.     public Texture TexturaFundoGrapH;
142.     public Texture LinhasoGrapH;
143.     public Texture TexturaChart;
144.     public Texture TexturaIndicators;
145.     public float[,] AxisXToPlot;
146.     public float[,] AxisYToPlot;
147.     [HideInInspector]
148.     public float[] XInterp;
149.     [HideInInspector]
150.     public float[] YInterp;
151.
152.     //REDE
153.     public string StatusRPC="";
154.     public string testNetworkStatus="";
155.     public string testNetworkMessage="";
156.     public string shouldEnableNatMessage="";
157.
158.     //E-MAILS
159.     public int EmailsEnviados=0;
160.     private string AssuntoEmail="";
161.     private string CorpoEmail="";
162.     public string[] destinatarios;
163.     public string[] AtachedFiles;
164.
165.     public float FPScounter;
166.     public float updateInterval = 0.5f;
167.     private float accum    = 0; // FPS accumulated over the interval
168.     private int   frames   = 0; // Frames drawn over the interval
169.     private float timeleft; // Left time for current interval
170.
171.     public GameObject MainCam;
172.     public Text Alerta1;
173.     public Text Alerta2;
174.     public Boolean Alerta=false;
175.     public float TempoDeAtualizacao=0.1f;
176.     private Color DefinedGuiColor=Color.gray;
177.
178.     public GameObject[] HUDs;
179.
180.     public float Tempo;
181.     public float TempoRelatorio=30f;
182.     private float TempoRelatorioMark=600f;
183.
184.     //programa
185.     public Boolean Desligando=false;
186.     public Boolean Desligar=false;
187.
188.     //Parar processo
189.     public Boolean PararTudo=false;
190.
```

```
191. // GUI
192. private int PosicaoXNaTela;
193. private int PosicaoYNaTela;
194.
195. public float TempoDeDesligamento=3.0f;
196. private float TempoDeDesligar;
197.
198. private Boolean Stats=true;
199. private Boolean StatsAux=false;
200.
201. public bool Graficos=false;
202. private int LarguraDoGrafico;
203. public bool Console=false;
204. private bool ConsoleAux=false;
205. public bool WebCam=false;
206. public Image[] WebCamTex;
207. public bool BigWebCam=false;
208. public bool SmallWebCam=true;
209. public GUITexture GUITextweb1;
210. public WebcamToGUI Scriptweb1;
211. public Texture TextureClose;
212. public Texture TextureDisquete;
213.
214. public string ConsoleString = "INICIANDO ";
215. public bool DebugSerial=false;
216. public bool AutoScroll=false;
217.
218. private string Abertura="";
219.
220. public bool HUDoperacao=false;
221.
222. [HideInInspector]
223. public Vector2 scrollPosition;
224. //HideInInspector
225. public Vector2 scrollGrafPosition;
226. [HideInInspector]
227. public Vector2 scrollEquipPosition;
228.
229. public string URLContent;
230.
231. // START|||||||||||||||||||||||||||||||||||||
232. void Start () {
233.
234.     //Setando os audios estaticos
235.     StaticAudios = new AudioSource[Audios.Length];
236.     for (int i=0; i<Audios.Length; i++) {
237.         StaticAudios[i]=Audios[i];
238.     }
239.
```

```

240.
241.
242.    //Contador de fps
243.    timeleft = updateInterval;
244.
245.    //Desempenho
246.    //Profiler.enabled = true;
247.
248.    //Graficos dos sensores
249.    LarguraDoGrafico = Mathf.RoundToInt(0.41f * (float)Screen.width - 35f);
250.
251.    //Criar Pastas
252.    if (!Directory.Exists(Application.dataPath+"/Screenshots")){
253.        System.IO.Directory.CreateDirectory(Application.dataPath+"/Screenshots");}
254.    //Speak
255.    if (Directory.Exists (Application.dataPath + "/TempSpeak")) {
256.        DeleteFolderContent (Application.dataPath + "/TempSpeak");} //Limpia tudo que houver
257.    if (!Directory.Exists(Application.dataPath + "/TempSpeak")){
258.        System.IO.Directory.CreateDirectory(Application.dataPath + "/TempSpeak");}
259.    //Resultados
260.    if (!Directory.Exists(Application.dataPath + "/Results")){
261.        System.IO.Directory.CreateDirectory(Application.dataPath + "/Results");}
262.    String Results = Application.dataPath + "/Results/Resultados.txt";
263.    if (!File.Exists(Results)){File.Create(Results);}
264.    //Criar arquivos
265.    String Log = Application.dataPath+"/Log.txt";
266.    if (!File.Exists(Log)){File.Create(Log);}
267.
268.    //Iniciando console
269.    ConsoleString = "INICIANDO CONSOLE"+" "+System.DateTime.Now.ToString("dd-MM-yyyy");
270.    //Iniciando Log
271.    AddText(Log, "");
272.    AddText(Log, "[PROGRAMA INICIADO]");
273.    AddText(Log, "");
274.    //Iniciando Resultados
275.    AddText(Results, "");
276.    AddText(Results, "[PROGRAMA INICIADO]");
277.    AddText(Results, "");
278.    AddText(Results,ShowArrayStrings(Sensores));
279.    AddText(Results, "");
280.    //Iniciando Eva
281.
282.        //Tentando se conectar ao arduino
283.        if (PlayerPrefs.GetString ("Mode") == "LocalControl") {ConnectToArduino (true);}
284.        if (PlayerPrefs.GetString ("Mode") == "RemoteControl") {ConnectToArduino (false);RemoteControl=true;}
285.        if (PlayerPrefs.GetString ("Mode") == "Simulation") {ConnectToArduino (false);Simulation=true;}
286.
287.        //Testar conexao com a Internet

```

```

288.         if (CheckForInternetConnection ()) {
289.             UnityEngine.Debug.Log("Computador conectado com a Internet");
290.             AddText(Log, "Computador conectado com a Internet");
291.             ConsoleString += "\r\n"+System.DateTime.Now.ToString("hh.mm.ss")+" "+ "Computador
292.             conectado com a Internet";
293.         }
294.         if (!CheckForInternetConnection ()) {
295.             UnityEngine.Debug.Log("Computador desconectado com a Internet");
296.             AddText(Log, "Internet indisponivel");
297.             ConsoleString += "\r\n"+System.DateTime.Now.ToString("hh.mm.ss")+" "+ "Internet in
298.             disponivel";
299.             Abertura+="A Internet está indisponivel. verifique a conexão ...";
300.         }
301.         //setando os indicadores 3D
302.         for (int i = 0; i < Equipamentos.Length; i++) {
303.             int k=i+1;
304.             Titulo3DIndEquipamentos[i].text=Equipamentos[i]+" "+k.ToString("D2"); //titulos d
305.             os indicadores 3D
306.             Status3DIndEquipamentos[i].text=StatusEquipString[i]; //Status dos indicadores 3D
307.         }
308.         //Tempo inicial de envio dos e-mails
309.         TempoRelatorioMark=TempoRelatorio;
310.         //Arquivos anexados
311.         AtachedFiles[0] = "Log.txt";
312.         AtachedFiles[1] = "Results/Resultados.txt";
313.         //Atribuir titulos e as unidades aos sensores
314.         for (int i = 0; i < Sensores.Length; i++) {
315.             TituloHudsensores [i].text = Sensores[i]+" "+i.ToString("D2");
316.             UnidadeHudsensores [i].text = UnidadeSensores[i];
317.         }
318.         //Desligando tudo
319.         if (!Simulation && !RemoteControl) {
320.             if (sp.IsOpen) {
321.                 for (int i = 0; i < Equipamentos.Length; i++) {
322.                     StartCoroutine(ReliableSetEquipment("e" + i.ToString ("D2") + "000"));
323.                 }
324.             }
325.         }
326.         //Preenchimento com os dados estatisticos iniciais
327.         for (int i = 0; i < Sensores.Length; i++)
328.         {
329.             ValoresMediaSensores[i]=0.5f*LimiteMinSensores[i]+0.5f*LimiteMaxSensores[i];
330.
331.
332.
333.
```

```

334.         ValoresMinSensores[i]=LimiteMaxSensores[i];
335.         ValoresMaxSensores[i]=LimiteMinSensores[i];
336.     }
337.
338.     //Preenchimento dos estados dos relatorios
339.     for (int i = 0; i < Sensores.Length; i++)
340.     {
341.         RelatFaladoSensorI[i]=false;
342.     }
343.
344.     //Preenchendo as corrotinas dos equipamentos
345.     CorrotinasEquipamentos = new Coroutine[Equipamentos.Length];
346.     for (int i = 0; i < Equipamentos.Length; i++) {
347.         CorrotinasEquipamentos[i]=null;
348.     }
349.
350.     XInterp = new float[LarguraDoGrafico - 120];
351.     YInterp = new float[LarguraDoGrafico - 120];
352.     ValoresSensores = new float[Sensores.Length, 4, 60];
353.     TempoValoresSensores = new float[Sensores.Length, 4, 60];
354.     for (int l = 0; l < Sensores.Length; l++)
355.     {
356.         for (int m = 0; m < 4; m++)
357.         {
358.             for (int n = 0; n < 60; n++)
359.             {
360.                 TempoValoresSensores[l, m, n] = 0.1667f * (float)n;
361.                 ValoresSensores[l, m, n] = LimiteMinSensores[l];
362.             }
363.         }
364.     }
365.     float num2 = (TempoValoresSensores[1, 1, TempoValoresSensores.GetLength(2) - 1] -
TempoValoresSensores[1, 1, 0]) / (float)(LarguraDoGrafico - 120 - 1);
366.     AxisXToPlot = new float[Sensores.Length, 4, LarguraDoGrafico - 120];
367.     AxisYToPlot = new float[Sensores.Length, 4, LarguraDoGrafico - 120];
368.     for (int num3 = 0; num3 < Sensores.Length; num3++)
369.     {
370.         for (int num4 = 0; num4 < 4; num4++)
371.         {
372.             for (int num5 = 0; num5 < LarguraDoGrafico - 120; num5++)
373.             {
374.                 AxisXToPlot[num3, num4, num5] = (float)num5 * num2;
375.                 AxisYToPlot[num3, num4, num5] = LimiteMinSensores[num3];
376.             }
377.         }
378.     }
379.
380.     Speak (Abertura);
381.
382. }
```

```

383.
384.      // UPDATE|||||||||||||||||||||||||||||||||||||||||||||||||||||
385.      void Update () {
386.
387.          //string bridge = Application.dataPath + "/Eva/bridge.ini";
388.          //AP_INIFile ini = new AP_INIFile(bridge);
389.
390.          if (!Simulation && !RemoteControl) {
391.              if (!sp.IsOpen) {
392.                  OpenConnection ();
393.              }
394.          }
395.
396.          String Results = Application.dataPath + "/Results/Resultados.txt";
397.          String Log = Application.dataPath + "/Log.txt";
398.
399.          //Ler entradas de sensores
400.          if (!RemoteControl && !Simulation ){
401.              if (sp.IsOpen){
402.                  try {
403.                      COMPortResult = sp.ReadLine ();
404.                  }
405.                  catch {
406.                  }
407.              }
408.          }
409.
410.          //Gerador de dados simulados
411.          if(Simulation){
412.              if (Time.time>=Tempo){
413.                  int l = PassoCounter;
414.                  //DadoSimulado=Sensores [1] + 1.ToString ("D2")+"="+(LimiteMaxSensores[1]-
415.                  LimiteMinSensores[1])*0.25f*(Mathf.Cos (0.001f*(l+1)*Tempo)+1)*(Mathf.Sin(0.005f*Tempo)+1f)/(l-
416.                  +1)+UnityEngine.Random.Range (-0.01f*(LimiteMaxSensores[1]-
417.                  LimiteMinSensores[1]), 0.01f*(LimiteMaxSensores[1]-LimiteMinSensores[1])).ToString();
418.                  //DadoSimulado=Sensores [1] + 1.ToString ("D2")+"="+(LimiteMaxSensores[1]-
419.                  LimiteMinSensores[1])*1f-Mathf.Exp(-0.005f*Tempo))+UnityEngine.Random.Range (-
420.                  0.01f*(LimiteMaxSensores[1]-LimiteMinSensores[1]), 0.01f*(LimiteMaxSensores[1]-
421.                  LimiteMinSensores[1])).ToString();
422.                  //DadoSimulado=Sensores [1] + 1.ToString ("D2")+"="+(LimiteMaxSensores[1]-
423.                  LimiteMinSensores[1])*1.5f*Mathf.Sin(0.01f*Tempo)+UnityEngine.Random.Range (-
424.                  0.1f*(LimiteMaxSensores[1]-LimiteMinSensores[1]), 0.1f*(LimiteMaxSensores[1]-
425.                  LimiteMinSensores[1])).ToString();
426.                  //DadoSimulado=Sensores [1] + 1.ToString ("D2")+"="+(LimiteMinSensores[1]+(LimiteM-
427.                  axSensores[1]-LimiteMinSensores[1])*1.0f*(1f-Mathf.Exp(-
428.                  0.05f*Tempo))+UnityEngine.Random.Range (-0.01f*(LimiteMaxSensores[1]-
429.                  LimiteMinSensores[1]), 0.01f*(LimiteMaxSensores[1]-LimiteMinSensores[1])).ToString();
430.
431.          if(PassoCounter!=9 && PassoCounter!=6 && PassoCounter!=10){

```

```

420.             DadoSimulado=Sensores [1] + 1.ToString ("D2")+"="+(LimiteMinSensores[1]+L
imiteMaxSensores[1])/2f).ToString();
421.         }
422.
423.         if(PassoCounter==4){
424.             DadoSimulado=Sensores [1] + 1.ToString ("D2")+"="+(LimiteMinSensores[1]+(L
imiteMaxSensores[1]-
LimiteMinSensores[1])*(ValoresEquipamentos [7] / LimiteMaxEquipamentos [7])+0.011f).ToString()
;
425.         }
426.         if(PassoCounter==5){
427.             DadoSimulado=Sensores [1] + 1.ToString ("D2")+"="+(LimiteMinSensores[1]+(L
imiteMaxSensores[1]-
LimiteMinSensores[1])*(ValoresEquipamentos [6] / LimiteMaxEquipamentos [6])+0.011f).ToString()
;
428.         }
429.
430.         PassoCounter+=1;
431.     }
432.     if (PassoCounter>Sensores.Length-1){PassoCounter=0;}
433. }
434.
435.
436. //Arduino conectado
437. if (!Simulation && Arduino) {
438.     GrabResult (COMPortResult);//distribuir resultados em seus devidos lugares
439. }
440. //Simulação habilitada
441. if (Simulation) {
442.     GrabResult (DadoSimulado);//distribuir resultados em seus devidos lugares
443. }
444. //Controle Remoto habilitado
445. if (RemoteControl) {
446.     GrabResult (RemoteData);//distribuir resultados em seus devidos lugares
447. }
448.
449.
450. //UnityEngine.Debug.Log (COMPortResult);
451.
452. if (!Simulation && !Arduino && String.Equals(COMPortResult, "Arduino")) {//receber ex:
Termometro01 (alterar código no arduino)
453.     Speak("confirmação efetuada, arduino conectado a porta "+ ports [ports.Length -
1].ToString()+"");
454.     AddText (Log, "confirmação efetuada, arduino conectado a porta "+ ports [ports.Len
gth - 1].ToString());
455.     ConsoleString += "\r\n" + System.DateTime.Now.ToString ("hh:mm:ss") + " " + "conf
irmação efetuada, arduino conectado a porta "+ ports [ports.Length - 1].ToString();
456.     scrollPosition += new Vector2 (0, 15000);
457.     Arduino=true;
458. }

```

```
459.  
460.        //Status da conexao para o console e para o Log  
461.        if (testNetworkStatus != "") {  
462.            AddText (Log, testNetworkStatus);  
463.            AddText (Log, testNetworkMessage);  
464.            AddText (Log, shouldEnableNatMessage);  
465.            ConsoleString += "\r\n" + System.DateTime.Now.ToString ("hh.mm.ss") + " " + testN  
etworkStatus;  
466.            ConsoleString += "\r\n" + System.DateTime.Now.ToString ("hh.mm.ss") + " " + testN  
etworkMessage;  
467.            ConsoleString += "\r\n" + System.DateTime.Now.ToString ("hh.mm.ss") + " " + shoul  
dEnableNatMessage;  
468.            scrollPosition += new Vector2 (0, 15000);  
469.            testNetworkStatus="";  
470.        }  
471.  
472.        //Atualizacao dos dados a cada 17 segundos  
473.  
474.        if (Time.time >= TAddResultsMark)  
475.        {  
476.            if(Arduino || RemoteControl){  
477.                AddText(Results,ShowArray(ValoresAtuaisSensores));  
478.            }  
479.            TAddResultsMark = TAddResults + Time.time;  
480.        }  
481.  
482.        //verificar status dos sensores a cada 120 segundos  
483.        if (Time.time >= TVerificarSensores)  
484.        {  
485.            if (VerifiqueTempoAtualizacao) {  
486.                CheckSensorStatus();  
487.            }  
488.            //Lembretes pre programados  
489.            Lembretes();  
490.  
491.            TVerificarSensores = 240f + Time.time;  
492.        }  
493.  
494.        // Envio de dados para os clientes a cada 1.1 segundo  
495.        if (Time.time >= TSendDataMark)  
496.        {  
497.            if (Simulation || Arduino) {  
498.                if(Network.isServer){  
499.                    int i=SendDataCounter;  
500.                    GetComponent<NetworkView>().RPC("SendData", RPCMode.Others,Sensores [i] +  
i.ToString ("D2")+"="+ValoresAtuaisSensores[i]);  
501.                    //UnityEngine.Debug.Log(Sensores [i] + i.ToString ("D2")+"="+ValoresAtuais  
Sensores[i]);  
502.                    SendDataCounter+=1;  
503.                    if (SendDataCounter>Sensores.Length-1){SendDataCounter=0;}  
504.                }  
505.            }  
506.        }  
507.    }  
508.}
```

```

504.         }
505.     }
506.
507.     //Atualiza o tempo de funcionamento dos sensores
508.     for (int i = 0; i < Sensores.Length; i++) {
509.         SensTempoDesdeUltimaAtualizacao[i]=Time.time-SensTempoUltimaAtualizacao[i];
510.     }
511.
512.     //resultados calculados em função de outras variaveis
513.     ResultadosCalculados ();
514.
515.     TSendDataMark = 1.1f + Time.time;
516. }
517.
518.     //Verificar o Bridge a cada 0,7 segundos
519.     if (Time.time >= TWatchBridge) {
520.         //StartCoroutine(BridgeWatcher());
521.         TWatchBridge+=0.7f;
522.
523.         //Atualiza indicador de media
524.         for (int j = 0; j < ValoresSensores.GetLength(0); j++)
525.         {
526.             float ValuePercentMedia = (Media(PartOfArray(AxisYToPlot, j, 0)) -
527.             LimiteMinSensores [j]) / (LimiteMaxSensores [j] - LimiteMinSensores [j]);
528.             if (!float.IsNaN(ValuePercentMedia) && !float.IsInfinity(ValuePercentMedia)){
529.                 MedIndiSensores[j].transform.localEulerAngles = new Vector3(0.0f, 180.0f,
530.                 360f*ValuePercentMedia);}
531.         }
532.
533.
534.         if (Time.time >= T10minMark)
535.         {
536.             for (int j = 0; j < ValoresSensores.GetLength(0); j++)
537.             {
538.                 for (int k = 0; k < ValoresSensores.GetLength(2) - 1; k++)
539.                 {
540.                     ValoresSensores[j, 0, k] = ValoresSensores[j, 0, k + 1];
541.                 }
542.                 ValoresSensores[j, 0, ValoresSensores.GetLength(2) -
543. 1] = ValoresAtuaisSensores[j];
544.                 InterpolationToPlot(j, 0, TempoValoresSensores, ValoresSensores, AxisXToPlot,
545. AxisYToPlot, out AxisXToPlot, out AxisYToPlot);
546.             }
547.             //Tudo pronto?-----
548.             if (!ReadToGOMark && Time.time>100f && Arduino){
549.                 int LocalErrorCounter=0;

```

```

549.         for (int i = 0; i < Sensores.Length; i++)
550.         {
551.             if (SensTempoDesdeUltimaAtualizacao[i]>50f){
552.                 LocalErrorCounter+=1;
553.             }
554.         }
555.         if (LocalErrorCounter==0){
556.             Speak("Prerequisites de operação e automação atingidos. a coluna está pronta para operar");
557.             ReadToGO=true;
558.             ReadToGOMark=true;
559.         }
560.         if (LocalErrorCounter>0){
561.             Speak("Temos "+LocalErrorCounter.ToString()+" sensores com mal funcionamento.");
562.             ReadToGOMark=true;
563.         }
564.     }
565.
566.     T10minMark = T10min + Time.time;
567. }
568. if (Time.time >= T1horMark)
569. {
570.     for (int l = 0; l < ValoresSensores.GetLength(0); l++)
571.     {
572.         for (int m = 0; m < ValoresSensores.GetLength(2) - 1; m++)
573.         {
574.             ValoresSensores[l, 1, m] = ValoresSensores[l, 0, m + 1];
575.         }
576.         ValoresSensores[l, 1, ValoresSensores.GetLength(2) - 1] = ValoresAtuaisSensores[l];
577.     }
578.     //Relatórios das variáveis importantes
579.
580.     for (int i = 0; i < Sensores.Length; i++) {
581.         if (SensorPrincipal[i] && Time.time>3.1f*T1hor){
582.             if (Arduino || Simulation){
583.                 StartCoroutine(RelatorioSensor(i,"previsão"));
584.             }
585.         }
586.     }
587.
588.     T1horMark = T1hor + Time.time;
589. }
590. if (Time.time >= T1diaMark)
591. {
592.     for (int n = 0; n < ValoresSensores.GetLength(0); n++)
593.     {
594.         for (int num3 = 0; num3 < ValoresSensores.GetLength(2) - 1; num3++)
595.         {

```

```

596.                     ValoresSensores[n, 2, num3] = ValoresSensores[n, 0, num3 + 1];
597.                 }
598.                 ValoresSensores[n, 2, ValoresSensores.GetLength(2) -
1] = ValoresAtuaisSensores[n];
599.             }
600.
601.             Speak("Aconselho a checagem visual completa da coluna");
602.
603.             T1diaMark = T1dia + Time.time;
604.         }
605.         if (Time.time >= T1semMark)
606.         {
607.             for (int num4 = 0; num4 < ValoresSensores.GetLength(0); num4++)
608.             {
609.                 for (int num5 = 0; num5 < ValoresSensores.GetLength(2) - 1; num5++)
610.                 {
611.                     ValoresSensores[num4, 3, num5] = ValoresSensores[num4, 0, num5 + 1];
612.                 }
613.                 ValoresSensores[num4, 3, ValoresSensores.GetLength(2) -
1] = ValoresAtuaisSensores[num4];
614.             }
615.             T1semMark = T1sem + Time.time;
616.         }
617.
618.         //Relatorio falado se clicado no indicador do sensor
619.         for (int i = 0; i < Sensores.Length; i++) {
620.             if (RelatFaladoSensorI[i]){
621.                 string FalaTemp="";
622.                 FalaTemp="No "+Sensores[i]+" "+i.ToString("D2")+" "+DescriçaoSensor[i]+" a "+T
ipoSensores[i]+"é de "+ValoresAtuaisSensores[i].ToString("F2")+" "+UnidadeFaladaSensores[i]+".
..";
623.                 if (Media(PartOfArray(AxisYToPlot, i, 0))<LimiteMaxSensores[i] && Media(PartOf
Array(AxisYToPlot, i, 0))>LimiteMinSensores[i]){FalaTemp+=" A "+TipoSensores[i]+" média está
dentro dos limites previstos"+...;}
624.                 if (Media(PartOfArray(AxisYToPlot, i, 0))<LimiteMinSensores[i]){FalaTemp+=" A
"+TipoSensores[i]+" média está abaixo do limite inferior. a variável está fora do controle"+"
..";}
625.                 if (Media(PartOfArray(AxisYToPlot, i, 0))>LimiteMaxSensores[i]){FalaTemp+=" A
"+TipoSensores[i]+" média está acima do limite superior. a variável está fora do controle"+".
..";}
626.
627.                 if (ValoresMaxSensores[i]<LimiteMinSensores[i]){FalaTemp+=" A "+TipoSensores[
i]+" máxima está abaixo do limite inferior. a variável está fora do controle"+...;}
628.                 if (ValoresMaxSensores[i]>LimiteMaxSensores[i]){FalaTemp+=" A "+TipoSensores[
i]+" máxima está acima do limite superior."+...;}
629.
630.                 if (ValoresMinSensores[i]>LimiteMaxSensores[i]){FalaTemp+=" A "+TipoSensores[
i]+" mínima está acima do limite superior. a variável está fora do controle"+...;}
631.                 if (ValoresMinSensores[i]<LimiteMinSensores[i]){FalaTemp+=" A "+TipoSensores[
i]+" mínima está abaixo do limite inferior."+...;}

```

```

632.
633.           if (TaxaFinal(PartOfArray(TempoValoresSensores, i, 0),PartOfArray(ValoresSenso
res, i, 0))>0.1f){FalaTemp+=". seu valor está aumentando"+"...";}
634.           if (TaxaFinal(PartOfArray(TempoValoresSensores, i, 0),PartOfArray(ValoresSenso
res, i, 0))>-
0.1f && TaxaFinal(PartOfArray(TempoValoresSensores, i, 0),PartOfArray(ValoresSensores, i, 0))<
0.1f){FalaTemp+=". seu valor está se mantendo estável"+"...";}
635.           if (TaxaFinal(PartOfArray(TempoValoresSensores, i, 0),PartOfArray(ValoresSenso
res, i, 0))<-0.1f){FalaTemp+=". seu valor está diminuindo"+"...";}
636.
637.           Speak (FalaTemp);
638.           RelatFaladoSensorI[i]=false;
639.
640.       }
641.   }
642. //RELATORIO ENVIADO A CADA INTERVALO "TempoRelatorio"
643. if (Time.time >= TempoRelatorioMark) {
644.     EnviarResultadosViaEmail();
645.     TempoRelatorioMark = TempoRelatorio + Time.time;
646. }
647.
648. if (Time.time >= Tempo) { //A cada decimo de segundo
649.
650.     //Equipamento i a cada Tempo de atualização
651.     for (int i = 0; i < Equipamentos.Length; i++) {
652.         if (StatusEquipamentos [i]) {
653.             TempoEquipamentos [i] = Mathf.RoundToInt (100.0f * (float.Parse (TempoHUE
quipamentos [i]) - TempoDeAtualizacao)) / 100.0f;
654.             TempoHUEquipamentos [i] = TempoEquipamentos [i].ToString ();
655.             if (TempoEquipamentos [i] <= 0) {
656.                 if (Network.isServer){GetComponent<NetworkView>().RPC("LigaDesligaEqui
pamento", RPCMode.Others, i,0);}
657.                 ValoresEquipamentos[i]=0f;
658.                 MarkStatusEquipamentos [i] = true;
659.                 TempoEquipamentos [i] = 0.0f;
660.                 TempoHUEquipamentos [i] = Mathf.RoundToInt(TempoEquipamentos [i]).ToS
tring ();
661.                 Speak (Equipamentos [i] + " " + i.ToString ("D2") +" "+DescricaoEquipa
mentos[i]+ " desligado depois de " + IntuitiveTime(float.Parse(TempoEquipamentosInicial [i]))
);
662.                 AddText (Log, Equipamentos [i] + " " + i.ToString ("D2") + " desligado
apos: " + TempoEquipamentosInicial [i] + " seg");
663.                 ConsoleString += "\r\n" + System.DateTime.Now.ToString ("hh:mm:ss") +
" " + Equipamentos [i] + " " + i.ToString ("D2") + " desligado apos: " + TempoEquipamentosIni
cial [i] + " seg";
664.                 scrollPosition += new Vector2 (0, 15000);
665.             }
666.         }
667.     }
668.

```

```

669.         //Desabilita o Alerta (GUI.text) quando 'Alerta' = false
670.         if (!Alerta) {
671.             Alerta1.enabled = false;
672.             Alerta2.enabled = false;
673.         }
674.
675.         Tempo = TempoDeAtualizacao + Time.time;
676.
677.     }
678.
679.     //Aplica modificações quando o HUD é modificado-----
680.
681.
682.     if (HUDChanged) {
683.
684.         if (Time.time>=TimeHUDChange){
685.             if (Network.isServer || Network.isClient) {
686.                 GetComponent<NetworkView> ().RPC ("UpdateCicloEquip", RPCMode.Others, Temp
oCicloEquip[ContHUDChange], ContHUDChange);
687.                 GetComponent<NetworkView> ().RPC ("UpdateEquipTime", RPCMode.Others, Tempo
HUDEquipamentos[ContHUDChange], ContHUDChange);
688.             }
689.             ContHUDChange+=1;
690.             TimeHUDChange=Time.time+0.1f;
691.             if (ContHUDChange>Equipamentos.Length-1){
692.                 HUDChanged=false;
693.                 ContHUDChange=0;
694.             }
695.             //Atualiza valores do slider e usa como gatilho para ativar o equipame
696.             nto (Importante)
697.             for (int i = 0; i < Equipamentos.Length; i++) {
698.                 if (ValoresAntigosEquipamentos[i]!=ValoresEquipamentos[i]){
699.                     if (Network.isServer || Network.isClient){GetComponent<Network
View>().RPC("LigaDesligaEquipamento", RPCMode.Others, i, Mathf.RoundToInt(ValoresEquipamentos[
i]));}
700.
701.                     MarkStatusEquipamentos [i] = true;
702.                 }
703.             }
704.
705.             //Batelada
706.             if (MarkBatelada && !Batelada){
707.                 TempoInicioBatelada=Tempo;
708.                 Batelada=true;
709.                 MarkBatelada=false;
710.             }
711.             if (MarkBatelada && Batelada){
712.                 Batelada=false;

```

```

713.         PararTudo=true;
714.         MarkBatelada=false;
715.     }
716.
717.     if (Batelada) {
718.         for (int i = 0; i < Equipamentos.Length; i++) {
719.             if (Tempo-
    TempoInicioBatelada>=float.Parse(TempoInicialEquip [i]) && !EquipConcluidoBatelada[i]){
720.                 MarkStatusEquipamentos[i]=true;
721.                 if (Network.isServer || Network.isClient){GetComponent<NetworkView>().RPC(
    "LigaDesligaEquipamento", RPCMode.Others, i,LimiteMaxEquipamentos[i]);}
722.                 EquipConcluidoBatelada[i]=true;}
723.             }
724.         }
725.
726.         //Desligando Alarme
727.         if (Tempo > 30) {
728.             MainCam.GetComponent< AudioSource > ().mute = true;
729.         }
730.
731.         for (int i = 0; i < Equipamentos.Length; i++) {
732.             //Ligar equipamento "I"
733.             if (MarkStatusEquipamentos [i] && ValoresEquipamentos[i]!=0f) {
734.                 StatusEquipamentos [i] = true;
735.                 if ((Arduino || Simulation) && EquipamentosLargePWM[i]) {
736.                     if (CorrotinasEquipamentos[i]!=null){StopCoroutine(CorrotinasEquipamentos[
    i]);}
737.                     CorrotinasEquipamentos[i] = StartCoroutine(LargePWM(i,ValoresEquipamentos[
    i]));
738.                 }
739.                 if ((Arduino || Simulation) && EquipamentosModulados[i]) {StartCoroutine(Relia-
    bleSetEquipment("e" + i.ToString ("D2") + Mathf.RoundToInt(ValoresEquipamentos[i]).ToString("D
    3")));}}
740.                 //UnityEngine.Debug.Log ("e" + i.ToString ("D2") + Mathf.RoundToInt(Valore
    sEquipamentos[i]).ToString("D3"));
741.                 TempoEquipamentosInicial [i] = TempoHUEquipamentos [i];
742.                 MarkStatusEquipamentos [i] = false;
743.                 if (ValoresAntigosEquipamentos[i]==0f){Speak (Equipamentos [i] + " " + i.ToStr
    ing ("D2") +" "+DescricaoEquipamentos[i]+ " ligado");}
744.                 AddText (Log, Equipamentos [i] + " " + i.ToString ("D2") + " ligado");
745.                 ConsoleString += "\r\n" + System.DateTime.Now.ToString ("hh.mm.ss") + " "
    + Equipamentos [i] + " " + i.ToString ("D2") + " ligado a "+((ValoresEquipamentos[i]/LimiteMa
    xEquipamentos[i])*100f).ToString("F2")+"%";
746.                 scrollPosition += new Vector2 (0, 15000);
747.
748.                 Status3DIndEquipamentos [i].text = "Ligado";
749.                 Info1Ind3DEquipamentos [i].text = "";
750.                 Info2Ind3DEquipamentos [i].text = " ";
751.
752.                 TextButtonEquip [i] = "Desligar";

```

```

753.             StatusEquipString [i] = "Ligada";
754.
755.             ValoresAntigosEquipamentos[i]=ValoresEquipamentos[i];
756.         }
757.
758.             //Desligar equipamento "I"
759.             if (MarkStatusEquipamentos [i] && StatusEquipamentos [i] && ValoresEquipamentos[i]==0f) {
760.                 StatusEquipamentos [i] = false;
761.                 RealStatusPWM[i]=false;
762.                 if (CorrotinasEquipamentos[i]!=null){StopCoroutine(CorrotinasEquipamentos[i]);
763.                 }
764.                 if (Arduino || Simulation) {StartCoroutine(ReliableSetEquipment("e" + i.ToString ("D2") + "000"));}
765.                 MarkStatusEquipamentos [i] = false;
766.                 Speak (Equipamentos [i] + " " + i.ToString ("D2") +" "+DescricaoEquipamentos[i]+ " desligado");
767.                 AddText (Log, Equipamentos [i] + " " + i.ToString ("D2") + " Desligada");
768.
769.                 ConsoleString += "\r\n" + System.DateTime.Now.ToString ("hh:mm:ss") + " "
770.                 + Equipamentos [i] + " " + i.ToString ("D2") + " Desligada";
771.                 scrollPosition += new Vector2 (0, 15000);
772.
773.
774.                 Status3DIndEquipamentos [i].text = "Desligado";
775.                 Info1Ind3DEquipamentos [i].text = "";
776.                 Info2Ind3DEquipamentos [i].text = " ";
777.
778.             }
779.         }
780.
781.         ///////////////////////
782.
783.         //Clipboard
784.         StartCoroutine(WatchClipboard());
785.         StartCoroutine (PauseSpeakTimer (10f));
786.
787.         ///////////////////////
788.
789.         //Estatísticas
790.
791.         if (Input.GetKeyDown("f12"))
792.         {
793.             StatsAux=true;
794.             if (!Stats && StatsAux){
795.                 Speak ("Mostrando estatísticas do programa");
796.                 Stats=true;StatsAux=false;} 
```

```
797.         if (Stats && StatsAux){  
798.             Speak ("Desabilitando estatísticas do programa");  
799.             Stats=false;StatsAux=false;}  
800.     }  
801.  
802.     //Console via botões  
803.     if (Input.GetKeyDown(KeyCode.LeftControl) || Input.GetKeyDown(KeyCode.RightControl))  
804.     {ConsoleAux=true;  
805.         if (!Console && ConsoleAux){Console=true;ConsoleAux=false;}  
806.         if (Console && ConsoleAux){Console=false;ConsoleAux=false;}  
807.     }  
808.  
809.  
810.     //Desligando tudo  
811.     if (PararTudo) {  
812.         //Desligando emergencialmente  
813.         if (Arduino) {  
814.             for (int i = 0; i < Equipamentos.Length; i++) {  
815.                 if (CorrotinasEquipamentos[i]!=null){StopCoroutine(CorrotinasEquipamentos[i]);}  
816.                 if (StatusEquipamentos[i]) StartCoroutine(ReliableSetEquipment("e" + i.ToString ("D2") + "000")); //Codigo e=equipamento, numero do equipamento com dois digitos, "d" de desligado (alterar codigo no arduino)  
817.             }  
818.         }  
819.         ConsoleString += "\r\n" + System.DateTime.Now.ToString ("hh.mm.ss") + " " + "Desligando os equipamentos...";  
820.         scrollPosition += new Vector2 (0, 15000);  
821.         AddText (Log, "Desligando os equipamentos...");  
822.         Speak ("Desligando os equipamentos...");  
823.         for (int i = 0; i < Equipamentos.Length; i++) {  
824.             if (StatusEquipamentos[i]){MarkStatusEquipamentos[i]=true;ValoresEquipamentos[i]=0f;}  
825.             if ((Network.isServer || Network.isClient) && StatusEquipamentos[i]){GetComponent<NetworkView>().RPC("LigaDesligaEquipamento", RPCMode.Others, i, Mathf.RoundToInt(ValoresEquipamentos[i]));}  
826.             ConsoleString += "\r\n" + System.DateTime.Now.ToString ("hh.mm.ss") + " " + Equipamentos [i] + " " + i.ToString ("D2") + " Desligado";  
827.             AddText (Log, Equipamentos [i] + " " + i.ToString ("D2") + " Desligado");  
828.         }  
829.  
830.         PararTudo = false;  
831.     }  
832.  
833.  
834.     if (Desligar){  
835.  
836.         if (Desligando) {  
837.             Speak ("Fechando o programa...");
```

```

838.         TempoDeDesligar=Time.time+TempoDeDesligamento;
839.         PararTudo=true;
840.         Desligando=false;
841.     }
842.
843.     if (Time.time>=TempoDeDesligar) {
844.         Application.Quit();
845.     }
846.
847. }
848.
849.
850. //Contador de FPS
851. if (Stats) {
852.     timeleft -= Time.deltaTime;
853.     accum += Time.timeScale/Time.deltaTime;
854.     ++frames;
855.
856.     // Interval ended - update GUI text and start new interval
857.     if( timeleft <= 0.0 )
858.     {
859.         // display two fractional digits (f2 format)
860.         FPScounter = accum/frames;
861.         timeleft = updateInterval;
862.         accum = 0.0F;
863.         frames = 0;
864.     }
865. }
866.
867. }
868. //ONGUI.....
869. .....
870. .....
871. void OnGUI () {
872.
873.     if (Tempo>3f){
874.
875.         PosicaoXNaTela = Mathf.RoundToInt (0.02f * Screen.width);
876.         PosicaoYNaTela = Mathf.RoundToInt (0.28f * Screen.height);
877.
878.         //Graficos com ScrollBar
879.         if (Graficos) {
880.             //GUI.Box(new Rect(0.25f*Screen.width, 0.27f*Screen.height, 660, 620),"");
881.             GUILayout.BeginArea (new Rect (0.25f * Screen.width + 10, 0.27f * Screen.height +
882.               10, 0.41f * Screen.width + 5, 0.69f * Screen.height - 50));
883.             scrollGrafPosition = GUILayout.BeginScrollView (scrollGrafPosition, GUILayout.Widt
884.               h (0.41f * Screen.width - 10), GUILayout.Height (0.69f * Screen.height - 60));
885.             for (int i = 0; i < Sensores.Length; i++)
886.             {

```

```

883.             if(scrollGrafPosition.y<=560*(i+1) && scrollGrafPosition.y>=360*(i-
884.               1)){//otimização: só desenhe o que é mostrado
885.               if (LimiteMinSensores[i]<ValoresMinSensores[i]){
886.                 DisplayDebugGUI("Resultado do "+Sensores[i]+ " "+i.ToString("D2"), "Tem-
887. po (m)", "Valor\n"+("+"+UnidadeSensores[i]+")", 0, 40 + 480 * i, LarguraDoGrafico, 260, PartOfA-
888. rray(AxisXToPlot, i, 0), PartOfArray(AxisYToPlot, i, 0), LinhasoGraphH, TexturaFundoGraphH, Text-
889. uraChart, LimiteMinSensores[i], ValoresMaxSensores[i], Color.white, Color.red, 0.6f * Color.bl-
890. ue + 0.3f * Color.green + 0.1f*Color.white);
891.               }
892.             }
893.             GUILayout.Label ("Graficos");
894.             for (int i = 0; i < 14f*Sensores.Length; i++) {
895.               GUILayout.Label ("\r\n");
896.             }
897.             GUILayout.EndScrollView ();
898.             GUILayout.EndArea ();
899.           }
900.
901.           String Log = Application.dataPath + "/Log.txt";
902.
903.           GUI.backgroundColor = Color.white;
904.           var centeredStyle = GUI.skin.GetStyle ("Label");
905.           centeredStyle.alignment = TextAnchor.UpperLeft;
906.           centeredStyle.fontSize = 10 + Screen.width / 1000;
907.
908.           var gs = GUI.skin.GetStyle ("Button");
909.           gs.fontSize = 10 + Screen.width / 1000;
910.
911.           GUI.color = Color.white;
912.
913.           if (Desligar && Stats) {
914.             GUI.Label (new Rect (0.15f * Screen.width, 0.18f * Screen.height, 500, 40), "Desli-
915. gando...");
916.
917.             //Mostrar estatísticas
918.             if (Stats) {
919.               GUI.contentColor = Color.blue + 0.8f * Color.white; //Cor

```

```

920.             GUI.Label (new Rect (0.15f * Screen.width, 0.08f * Screen.height, 130, 30), System
921.                 .DateTime.Now.ToString ("hh.mm.ss"));
922.                 GUI.Label (new Rect (0.15f * Screen.width, 0.1f * Screen.height, 130, 30), Time.ti
923.                     me.ToString ("f2") + " Segundos");
924.                     if (FPScounter < 30)
925.                         GUI.contentColor = Color.yellow + 0.5f * Color.white; //Cor
926.                     else
927.                         if (FPScounter < 20)
928.                             GUI.contentColor = Color.red + 0.5f * Color.white; //Cor
929.                         GUI.Label (new Rect (0.15f * Screen.width, 0.12f * Screen.height, 130, 30), "FPS:
930.                             " + FPScounter.ToString ("f2"));
931.                             GUI.contentColor = Color.blue + 0.8f * Color.white; //Cor
932.                             if (!Simulation && !RemoteControl) {
933.                                 GUI.Label (new Rect (0.15f * Screen.width, 0.14f * Screen.height, 500, 40), po
934.                                     rts [ports.Length - 1].ToString ());
935.                                 GUI.Label (new Rect (0.15f * Screen.width, 0.16f * Screen.height, 500, 40), CO
936.                                     MPortResult);
937.                             }
938.                             if(Network.isClient || Network.isServer){GUI.Label (new Rect (0.25f * Screen.w
939. idth, 0.08f * Screen.height, 130, 30), " Supervisores: " + (Network.connections.Length+1).ToSt
940. ring());}
941.                             if(Network.isClient){GUI.Label (new Rect (0.25f * Screen.width, 0.1f * Screen.
942. height, 130, 30), " Ping: " + Network.GetAveragePing(Network.connections[0])+" ms" );}
943.                             //GUI.Label (new Rect (0.28f * Screen.width, 0.08f * Screen.height, 500, 40),
944.                             "Uso da CPU: "+Profiler.GetMonoUsedSize());
945.                             //GUI.Label (new Rect (0.28f * Screen.width, 0.1f * Screen.height, 500, 40), "
946.                             Uso da RAM: "+ Profiler.GetTotalAllocatedMemory());
947.                             //Feed test
948. // WebClient Cliente = new WebClient();
949. // GUI.Label (new Rect (0.35f * Screen.width, 0.12f * Screen.height, 130, 30),Cli
950. ente.DownloadString("https://scholar.google.com.br/scholar?hl=pt-
951. BR&q=biodiesel&btnG=&lr="));
952.             GUI.contentColor = Color.white; //Cor
953.         }
954.         //Console
955.         if (Console) {
956.             //GUI.Box(new Rect(0.25f*Screen.width, 0.27f*Screen.height, 0.41f*Screen.width, 0.
957. 69f*Screen.height),"");
958.             GUILayout.BeginArea (new Rect (0.25f * Screen.width + 10, 0.27f * Screen.height +
959. 10, 0.41f * Screen.width + 5, 0.69f * Screen.height - 100));
960.             scrollPosition = GUILayout.BeginScrollView (scrollPosition, GUILayout.Width (0.41f
961. * Screen.width - 2), GUILayout.Height (0.69f * Screen.height - 110));

```

```

954.         GUILayout.Label (ConsoleString);
955.         GUILayout.EndScrollView ();
956.         GUILayout.EndArea ();
957.
958.         if (GUI.Button (new Rect (0.25f * Screen.width + 40, 0.96f * Screen.height -
959.             40, 120, 20), "Limpar texto"))
960.             ConsoleString = "";
961.         if (GUI.Button (new Rect (0.25f * Screen.width + 10, 0.96f * Screen.height -
962.             40, 20, 20), TextureDisquete)) {
963.             ConsoleString += "\r\n" + System.DateTime.Now.ToString ("hh.mm.ss") + " "
964.             + "Log salvo em: " + Log;
965.             scrollPosition += new Vector2 (0, 15000);
966.         }
967.         AutoScroll = GUI.Toggle(new Rect (0.25f * Screen.width + 170, 0.96f * Screen.h
968.             eight - 70, 120, 20), AutoScroll, " Autoscroll");
969.         DebugSerial = GUI.Toggle(new Rect (0.25f * Screen.width + 10, 0.96f * Screen.h
970.             eight - 70, 180, 20), DebugSerial, " Debug Serial COM");
971.         Speaking = GUI.Toggle(new Rect (0.25f * Screen.width + 170, 0.96f * Screen.hei
972.             ght - 40, 120, 20), Speaking, " Relatorio falado");
973.
974.         //HUD do usuário
975.         if (HUDoperacao) {
976.             //GUI.Box(new Rect(0.25f*Screen.width, 0.27f*Screen.height, 660, 620),"");
977.             GUILayout.BeginArea (new Rect (PosicaoXNaTela, PosicaoYNaTela, 0.21f * Screen.
978.                 width, 0.69f * Screen.height-40));
979.             scrollEquipPosition = GUILayout.BeginScrollView (scrollEquipPosition, GUILayout.
980.                 Width (0.21f * Screen.width), GUILayout.Height (0.69f * Screen.height-50));
981.
982.             GUI.Label (new Rect (5,25, 500, 40), "Ciclo");
983.             //Equipamentos
984.             for (int i = 0; i < Equipamentos.Length; i++) {
985.                 int k = i;
986.                 GUI.Label (new Rect (55, 25 + 65 * i, 500, 40), Equipamentos [i] + " " + k
987.                     .ToString ("D2")+" "+DescricaoEquipamentos[i]);
988.                 TempoCicloEquip [i] = GUI.TextField (new Rect (5, 45 + 65 * i, 45, 22), Te
989.                     mpoCicloEquip [i], 25);

```

```

990.             if (Network.isServer || Network.isClient){GetComponent<NetworkView>().
991.                 RPC("LigaDesligaEquipamento", RPCMode.Others, i, Mathf.RoundToInt(ValoresEquipamentos[i]));}
992.                 MarkStatusEquipamentos [i] = true;
993.             }
994.             GUI.Label (new Rect (198, 45 + 65 * i, 70, 22), StatusEquipString [i]);
995.             //indicador cores
996.             GUI.color = 0.8f*Color.red+0.2f*Color.white;
997.             if(StatusEquipamentos[i]) {GUI.color = Color.green;}
998.             GUI.DrawTexture(new Rect(250, 45 + 65 * i, 3, 44), TexturaChart);
999.             GUI.DrawTexture(new Rect(0, 45 + 65 * i, 1, 44), TexturaChart);
1000.
1001.             GUI.color = 0.8f*Color.red+0.2f*Color.white;
1002.             if(RealStatusPWM[i]) {GUI.color = Color.green;}
1003.             if(EquipamentosLargePWM[i])GUI.DrawTexture(new Rect(256, 45 + 65 * i, 3, 4
1004.             4), TexturaChart);
1005.             GUI.color = Color.white;
1006.             //Slider
1007.             if(EquipamentosModulados[i] || EquipamentosLargePWM[i]){ValoresEquipamento
1008. s[i] = GUI.HorizontalSlider(new Rect(5, 75 + 65 * i, 186, 30), ValoresEquipamentos[i], 0.0F, L
1009. imiteMaxEquipamentos[i]);}
1010.
1011.             if(!EquipamentosModulados[i]){GUI.enabled = false;ValoresEquipamentos[i] =
1012. GUI.HorizontalSlider(new Rect(5, 75 + 65 * i, 186, 30), ValoresEquipamentos[i], 0.0F, LimiteM
1013. axEquipamentos[i]);GUI.enabled = true;}
1014.             GUI.Label (new Rect (198, 70 + 65 * i, 40, 22), (ValoresEquipamentos[i]*(1
1015. 00f/LimiteMaxEquipamentos[i])).ToString("f1")+" %");
1016.
1017.
1018.
1019.             //habilita/desabilita Batelada
1020.             if (!Batelada) {
1021.                 if (GUI.Button (new Rect (Screen.width * 0.015f, Screen.height * 0.91f, 10
1022. 0, 22), "Iniciar Batelada")){
1023.                     Speak ("Batelada iniciada");
1024.                     MarkBatelada = true;}
1025.                 }
1026.                 if (Batelada) {
1027.                     if (GUI.Button (new Rect (Screen.width * 0.015f, Screen.height * 0.91f, 10
1028. 0, 22), "Parar Batelada")){
1029.                         Speak ("Batelada interrompida");
1030.                         MarkBatelada = true;}
1031.                     }

```

```
1031.  
1032.    }  
1033.  
1034.    DefinedGuiColor = GUI.backgroundColor;  
1035.    GUI.backgroundColor = Color.white;  
1036.  
1037.    //Alerta  
1038.    if (Alerta) {  
1039.        GUI.depth = 1;  
1040.        GUI.Box (new Rect (Screen.width * 0.5f - 240, Screen.height * 0.5f -  
1041.            30, 480, 140), "");  
1041.        if (GUI.Button (new Rect (Screen.width * 0.5f -  
1042.            75, Screen.height * 0.5f + 60, 150, 30), "OK"))  
1042.            Alerta = false;  
1043.        }  
1044.  
1045.        GUI.backgroundColor = DefinedGuiColor;  
1046.  
1047.    //Fechar programa  
1048.    if (GUI.Button (new Rect (Screen.width - 10 -  
1049.        Screen.height/25, Screen.height/60, Screen.height/24, Screen.height/24), TextureClose)) {  
1049.        Desligar = true;  
1050.        Desligando = true;  
1051.    }  
1052.  
1053.    //Simulação  
1054. //    if (!Simulation) {  
1055. //        if (GUI.Button (new Rect (Screen.width - 118, 5, 70, 25), "Simulation")) {  
1056. //            Simulation = true;  
1057. //            Speak ("Modo simulação habilitado");  
1058. //        }  
1059. //    }  
1060. //    if (Simulation && !RemoteControl) {  
1061. //        if (GUI.Button (new Rect (Screen.width - 118, 5, 70, 25), "Simulation")) {  
1062. //            if (ports.Length > 0) {  
1063. //                Simulation = false;  
1064. //                Speak ("Modo simulação desabilitado");  
1065. //            }  
1066. //            if (ports.Length == 0) {  
1067. //                Speak ("O Modo simulação não pode ser desabilitado, Não há arduino conecta-  
1068. //                    do");  
1068. //            }  
1069. //        }  
1070. //    }  
1071.  
1072.    //Parar processo  
1073.    GUI.backgroundColor = Color.red;  
1074.    if (HUDoperacao) {  
1075.        if (GUI.Button (new Rect (Screen.width * 0.015f, Screen.height * 0.9f, Screen.  
width * 0.215f, 26), "PARAR PROCESSO")) {
```

```
1076.             PararTudo = true;
1077.         }
1078.     }
1079.
1080. }
1081. //O HUD foi modificado
1082. if (GUI.changed) {
1083.     HUDChanged=true;
1084.     ContHUDChange=0;
1085.
1086. }
1087. }
1088.
1089. void OnApplicationQuit()
1090. {
1091.     String Log = Application.dataPath+"/Log.txt";
1092.     if (ports.Length > 0) {
1093.         Speak ("Programa fechado, até a próxima");
1094.         ConsoleString += "\r\n" + System.DateTime.Now.ToString ("hh.mm.ss") + " Programa f
echado";
1095.         AddText (Log,"Programa fechado");
1096.
1097.         if (!Simulation && !RemoteControl){sp.Close ();}
1098.     }
1099.     if (Network.isClient || Network.isServer) {Network.Disconnect();}
1100. }
1101.
1102. public void ConnectToArduino(bool Connect){
1103. ...
1104. }
1105.
1106. public void OpenConnection()
1107. {
1108.     if (sp !=null){
1109.         if (sp.isOpen){
1110.             sp.Close();
1111.             UnityEngine.Debug.Log("Fechando Porta, pois ja esta aberta");
1112.         }
1113.         else{
1114.             sp.Open();
1115.             sp.ReadTimeout = 1;
1116.             UnityEngine.Debug.Log("Porta Aberta!");
1117.         }
1118.     }
1119.     else{
1120.         if (sp.isOpen){UnityEngine.Debug.Log("Porta Aberta!");
1121.         else{UnityEngine.Debug.Log("Porta Nula");}
1122.     }
1123. }
1124.
```

```
1125.     //Arquivos de texto
1126. //  private static void AddTextToTxt(FileStream fs, string value)
1127. //  {
1128. //      byte[] info = new UTF8Encoding(true).GetBytes(value);
1129. //      fs.Write(info, 0, info.Length);
1130. //  }
1131.
1132.
1133.     //Arquivos de texto
1134.     public static void AddTextToFile(string adress,string value)
1135.     {
1136.         FileStream fs=new FileStream(adress, FileMode.Append, FileAccess.Write, FileShare.Write);
1137.         fs.Close();
1138.         StreamWriter sw=new StreamWriter(adress, true, Encoding.Unicode);
1139.         string NextLine="\r\n"+value;
1140.         sw.WriteLine(NextLine);
1141.         sw.Close();
1142.     }
1143.
1144.     //Arquivos de texto
1145.     public static void AddText(string adress,string value)
1146.     {
1147.         FileStream fs=new FileStream(adress, FileMode.Append, FileAccess.Write, FileShare.Write);
1148.         fs.Close();
1149.         StreamWriter sw=new StreamWriter(adress, true, Encoding.UTF8);
1150.         string NextLine="\r\n"+System.DateTime.Now.ToString("hh.mm.ss")+" "+System.DateTime.Now.ToString("dd-MM-yyyy")+" "+value;
1151.         sw.WriteLine(NextLine);
1152.         sw.Close();
1153.     }
1154.
1155.     //Delete todo o conteudo de uma pasta
1156.     public static void DeleteFolderContent(string LocalPath)
1157.     {
1158.         System.IO.DirectoryInfo di = new DirectoryInfo (LocalPath);
1159.
1160.         foreach (FileInfo file in di.GetFiles()) {
1161.             file.Delete ();
1162.         }
1163.         foreach (DirectoryInfo dir in di.GetDirectories()) {
1164.             dir.Delete (true);
1165.         }
1166.     }
1167.
1168.
1169.
1170.     //Fala
1171.     public static void Speak(string texto)
```

```

1172.    {
1173.        if (Speaking) {
1174.            if (ControlSpeaking) {
1175.                if (!Directory.Exists (Application.dataPath + "/TempSpeak")) {
1176.                    Directory.CreateDirectory (Application.dataPath + "/TempSpeak");
1177.                }
1178.                string str = UnityEngine.Random.Range (0, 99999999).ToString ("D9");
1179.                string text = Application.dataPath + "/TempSpeak/Temp" + str + ".vbs";
1180.                AddTextToFile (text, "\r\nSet voice = CreateObject(\"SAPI.Spvoice\")");
1181.                AddTextToFile (text, "\r\nvoice.Rate = 1.2");
1182.                AddTextToFile (text, "\r\nvoice.Volume = 100");
1183.                AddTextToFile (text, "\r\nvoice.Speak\"" + texto + "\"");
1184.                AddTextToFile (text, "Set obj = CreateObject(\"Scripting.FileSystemObject\")");
1185.            ;
1186.            AddTextToFile (text, "obj.DeleteFile(\"" + text + "\")");
1187.            new Process
1188.            {
1189.                StartInfo =
1190.                {
1191.                    FileName = "cscript",
1192.                    Arguments = "/B /Nologo \"\" + text + "\"",
1193.                    WindowStyle = ProcessWindowStyle.Hidden
1194.                }
1195.            }.Start ();
1196.        }
1197.    }
1198.
1199. //Seta o Tempo "largo" de PWM dos equipamentos com Large PWM
1200. IEnumerator LargePWM(int i,float CycleDuty){
1201.     float CycleTime = float.Parse(TempoCicloEquip[i]); //Segundos
1202.     float TimeOn=(CycleDuty*CycleTime)/LimiteMaxEquipamentos[i];
1203.     float TimeOff=CycleTime-TimeOn;
1204.     while(StatusEquipamentos[i]){
1205.         //Ligar
1206.         if(TimeOn>0f){
1207.             RealStatusPWM[i]=true;
1208.             if(Arduino){sp.Write ("e" + i.ToString ("D2") + Mathf.RoundToInt(LimiteMaxEquipamentos[i]).ToString("D3")));
1209.                 yield return new WaitForSeconds(TimeOn);
1210.             }
1211.             //Desligar
1212.             if(TimeOff>0f){
1213.                 RealStatusPWM[i]=false;
1214.                 if(Arduino){sp.Write ("e" + i.ToString ("D2") + "000");}
1215.                 yield return new WaitForSeconds(TimeOff);
1216.             }
1217.         }
1218.         //Desligar
1219.         RealStatusPWM[i]=false;

```

```

1220.        if (Arduino) {
1221.            sp.Write ("e" + i.ToString ("D2") + "000");
1222.        }
1223.        //UnityEngine.Debug.Log("Equipamento "+i.ToString("D2")+" Desligado");
1224.    }
1225.
1226.
1227.
1228.    //Tempo intuitivo
1229.    public string IntuitiveTime(float TimeInSeconds){
1230.        string Result = "";
1231.        if (TimeInSeconds>=0f && TimeInSeconds<60f)Result=TimeInSeconds.ToString ("F1")+" segundos";
1232.        if (TimeInSeconds>=60f && TimeInSeconds<3600f)Result=(TimeInSeconds/60f).ToString ("F1")+" minutos";
1233.        if (TimeInSeconds>=3600f && TimeInSeconds<86400f)Result=(TimeInSeconds/3600f).ToString ("F1")+" horas";
1234.        if (TimeInSeconds>=86400f && TimeInSeconds<604800f)Result=(TimeInSeconds/86400f).ToString ("F1")+" dias";
1235.        if (TimeInSeconds>=604800f && TimeInSeconds<2721600f)Result=(TimeInSeconds/604800f).ToString ("F1")+" semanas";
1236.        if (TimeInSeconds>=2721600f && TimeInSeconds<31536000f)Result=(TimeInSeconds/2721600f).ToString ("F1")+" meses";
1237.        if (TimeInSeconds>=31536000f)Result=(TimeInSeconds/31536000f).ToString ("F1")+" anos";
1238.
1239.        return Result;
1240.
1241.        IEnumerator PauseSpeakTimer(float Seconds){
1242.            if (ClipboardHelper.clipBoard == "EvaMicOcu") {
1243.                ControlSpeaking=false;
1244.                yield return new WaitForSeconds(Seconds);
1245.                ControlSpeaking=true;
1246.            }
1247.        }
1248.
1249.        IEnumerator FreeSpeakTimer(){
1250.            if (ClipboardHelper.clipBoard == "EvaMicFre") {
1251.                ControlSpeaking=true;
1252.            }
1253.            yield break;
1254.        }
1255.
1256.        //Verifica alterações do clipboard para comunicação com Eva
1257.        IEnumerator WatchClipboard(){
1258.            if (ClipboardHelper.clipBoard.Length==9 && !ClipboardOcupado) {
1259.                if(ClipboardHelper.clipBoard.Substring(0,3) == "Eva"){
1260.                    ClipboardOcupado = true;
1261.                    yield return new WaitForSeconds(0.1f);
1262.                    //Setar Equipamentos

```

```

1263.                 if (ClipboardHelper.clipBoard.Substring(3,1) == "E" && ClipboardHelper.clipBoa
    rd != "EvaEnvEma") {
1264.                     int EquipIntensity=int.Parse(ClipboardHelper.clipBoard.Substring(6,3));
1265.                     int EquipNumber=int.Parse(ClipboardHelper.clipBoard.Substring(4,2));
1266.                     ControlSpeaking=true;
1267.                     ValoresEquipamentos [EquipNumber] = ((float)EquipIntensity*LimiteMaxEquipa
    mentos[EquipNumber])/100f; //recebe valor em porcentagem
1268.                     MarkStatusEquipamentos[EquipNumber]=true;
1269.                 }
1270.                 //Relatorio de sensores
1271.                 if (ClipboardHelper.clipBoard.Substring(3,1) == "S") {
1272.                     int SensNumber=int.Parse(ClipboardHelper.clipBoard.Substring(4,2));
1273.                     PlaySound(3);
1274.                     ControlSpeaking=true;
1275.                     RelatFaladoSensorI[SensNumber]=true;
1276.                 }
1277.                 if (ClipboardHelper.clipBoard.Substring(3,1) == "D") {
1278.                     Desligar = true;
1279.                     Desligando = true;
1280.                     PlaySound(3);
1281.                     ControlSpeaking=true;
1282.                 }
1283.                 if (ClipboardHelper.clipBoard == "EvaRelGer") {
1284.                     ScriptEva.RelatorioGeral=true;
1285.                     PlaySound(3);
1286.                     ControlSpeaking=true;
1287.                 }
1288.                 if (ClipboardHelper.clipBoard == "EvaEnvEma") {
1289.                     EnviarResultadosViaEmail();
1290.                     PlaySound(3);
1291.                     ControlSpeaking=true;
1292.                 }
1293.
1294.                 ClipboardHelper.clipBoard = "";
1295.                 ClipboardOcupado=false;
1296.                 yield break;
1297.             }
1298.         }
1299.     }
1300.
1301.     //Verifica alterações no arquivo Eva/bridge.ini
1302.     IEnumerator BridgeWatcher(){
1303.         string bridge = Application.dataPath + "/Eva/bridge.ini";
1304.         AP_INIFile ini = new AP_INIFile(bridge);
1305.         string change = ini.ReadString("Control", "Change");
1306.         yield return new WaitForSeconds(0.01f);
1307.         if (change == "True") {
1308.
1309.             //Relatorio geral
1310.             string BridgeRelatorio = ini.ReadString("Control", "Relatorio");

```

```

1311.         yield return new WaitForSeconds(0.01f);
1312.         if (BridgeRelatorio=="True"){
1313.             ScriptEva.RelatorioGeral=true;
1314.             ini.WriteString("Control", "Relatorio", "False");
1315.             yield return new WaitForSeconds(0.01f);
1316.         }
1317.
1318.         //Enviando email
1319.         string BridgeEmail = ini.ReadString("Control", "EnviarEmail");
1320.         yield return new WaitForSeconds(0.01f);
1321.         if (BridgeEmail=="True"){
1322.             EnviarResultadosViaEmail();
1323.             ini.WriteString("Control", "EnviarEmail", "False");
1324.             yield return new WaitForSeconds(0.01f);
1325.         }
1326.
1327.         //Parando equipamentos
1328.         string BridgeStopAll = ini.ReadString("Control", "DesligaTudo");
1329.         yield return new WaitForSeconds(0.01f);
1330.         if (BridgeStopAll=="True"){
1331.             PararTudo = true;
1332.             ini.WriteString("Control", "DesligaTudo", "False");
1333.             yield return new WaitForSeconds(0.01f);
1334.         }
1335.
1336.         //Controlar equipamento I
1337.         for (int i = 0; i < Equipamentos.Length; i++) {
1338.             string BridgeEquipStatus = ini.ReadString("Control", "Equipamento"+i.ToString(
    "D2"));
1339.             yield return new WaitForSeconds(0.01f);
1340.             if (BridgeEquipStatus=="True" && !StatusEquipamentos[i]){
1341.                 MarkStatusEquipamentos[i]=true;
1342.                 ValoresEquipamentos [i] = LimiteMaxEquipamentos[i];
1343.             }
1344.             if (BridgeEquipStatus=="False" && StatusEquipamentos[i]){
1345.                 MarkStatusEquipamentos[i]=true;
1346.                 ValoresEquipamentos [i] = 0;
1347.             }
1348.         }
1349.
1350.         //Relatorio do sensor I
1351.         for (int i = 0; i < Sensores.Length; i++) {
1352.             string BridgeSensorRelat = ini.ReadString("Control", "Sensor"+i.ToString("D2"));
1353.             yield return new WaitForSeconds(0.01f);
1354.             if (BridgeSensorRelat=="True" && !RelatFaladoSensorI[i]){
1355.                 RelatFaladoSensorI[i]=true;
1356.                 ini.WriteString("Control", "Sensor"+i.ToString("D2"), "False");
1357.                 yield return new WaitForSeconds(0.01f);
1358.             }
1359.         }

```

```

1360.
1361.         ini.WriteString("Control", "Change", "False");
1362.     }
1363. }
1364.
1365. //Call a vbs file
1366. public static void CallVbs()
1367. {
1368.     String Voice = "C:/TempSpeak/Temp.vbs";
1369.     if (File.Exists (Voice)) {
1370.         Process scriptProc = new Process ();
1371.         scriptProc.StartInfo.FileName = @"cscript";
1372.         scriptProc.StartInfo.Arguments = "//B //Nologo " + "C://TempSpeak//Temp.vbs";
1373.         scriptProc.StartInfo.WindowStyle = ProcessWindowStyle.Hidden; //evita de mostrar a
janela de prompt
1374.         scriptProc.Start ();
1375.     }
1376. }
1377.
1378. //Conseguir versao do windows
1379. public void WindowsVersion(out string Version)
1380. {
1381.     int foundS1 = SystemInfo.operatingSystem.IndexOf("(");
1382.     Version = SystemInfo.operatingSystem.Substring(0,foundS1-2);
1383. }
1384.
1385. public void DataArduino(string Data, out string DataVariable, out float DataValue, out Boolean Useful )
1386. {
1387. ...
1388. }
1389.
1390. //pegar os resultados e distribuir em seu devido lugar
1391. public void GrabResult (string Resultado)
1392. {
1393.     string a = "";
1394.     float b = 0.0f;
1395.     Boolean c = false;
1396.     DataArduino (Resultado, out a, out b, out c);
1397.     for (int i = 0; i < Sensores.Length; i++) {
1398.         if (String.Equals (a, Sensores [i] + i.ToString ("D2")) && c) { //receber ex: Termo
metro01 (alterar codigo no arduino)
1399.             if (b>ValoresMaxSensores[i]) {ValoresMaxSensores[i]=b;MaxDataSensores[i]=System
.DateTime.Now.ToString("hh.mm.ss")+" "+System.DateTime.Now.ToString("dd/MM/yyyy");}
1400.             if (b<ValoresMinSensores[i]) {ValoresMinSensores[i]=b;MinDataSensores[i]=System
.DateTime.Now.ToString("hh.mm.ss")+" "+System.DateTime.Now.ToString("dd/MM/yyyy");}
1401.             if(i!=6){
1402.                 ValoresAtuaisSensores [i] = (ValoresAtuaisSensores [i]+b)/2f;
1403.                 ValorHUDSensores [i].text = ((ValoresAtuaisSensores [i]+b)/2f).ToString (
" F2");

```

```

1404.             ValorObj3DSensores [i].text = ((ValoresAtuaisSensores [i]+b)/2f).ToString
1405.                 ("F2") + " " + UnidadeSensores [i];
1406.             }
1407.             ValoresAtuaisSensores [i] = b;
1408.             ValorHUDSensores [i].text = b.ToString ("F2");
1409.             ValorObj3DSensores [i].text = b.ToString ("F2") + " " + UnidadeSensores [i
1410. ];
1411.         }
1412.         StatusObj3DSensores [i].text = "Funcionando";
1413.         TituloObj3DSensores [i].text = Sensores [i];
1414.         SensTempoDesdeUltimaAtualizacao[i]=0f;
1415.         SensTempoUltimaAtualizacao[i]=Time.time;
1416.         VerifiqueTempoAtualizacao=true;
1417.         float ValuePercent = (ValoresAtuaisSensores [i] -
1418.             LimiteMinSensores [i]) / (LimiteMaxSensores [i] - LimiteMinSensores [i]);
1419.         CircIndiSensores [i].GetComponent<Renderer> ().material.SetFloat ("_Percent",
1420.             ValuePercent);
1421.     }
1422. //resultados calculados em função de outras variaveis
1423. public void ResultadosCalculados(){
1424.     //Razao molar
1425.     float MMOLeo = 859.1648f; //massa molar media do oleo de algodao
1426.     float MMAcool = 46.06844f; //massa molar do alcool utilizado (Etanol = 46.06844 Metano
1427.     1 = 32.04)
1428.     float RoOleo = 912f; //Densidade media do oleo de algodao Kg/m3
1429.     float RoAlcool = 789f; //Densidade Kg/m3 (Etanol = 789 Metanol = 792)
1430.     float FluxoOleo = ValoresAtuaisSensores [4]; //(1/min)
1431.     float FluxoAlcool = ValoresAtuaisSensores [5]+ValoresAtuaisSensores [10]/1000f; //(Vaz
1432.     ao de alcool mais vazao de catalisador em l/min)
1433.     //UnityEngine.Debug.Log(FluxoAlcool.ToString("f4"));
1434.     float VazaoMolarOleo = RoOleo * FluxoOleo / MMOLeo; //(mol/min)
1435.     float VazaoMolarAlcool = RoAlcool * FluxoAlcool / MMAcool; //(mol/min)
1436.     if (VazaoMolarOleo < 0.005) {VazaoMolarOleo=0f;}
1437.     if (VazaoMolarAlcool < 0.005) {VazaoMolarAlcool=0f;}
1438.     float Razao = (VazaoMolarAlcool / VazaoMolarOleo); //Razao molar alcool/oleo
1439.     if (!(float.IsNaN(ValoresAtuaisSensores[6])) &&
1440.         ValoresAtuaisSensores[6] != float.PositiveInfinity &&
1441.         ValoresAtuaisSensores[6] != float.NegativeInfinity) {
1442.         Razao=(ValoresAtuaisSensores[6]+Razao)/2f;
1443.         //if (Razao != Razao) {Razao = 0.0f;}
1444.         //UnityEngine.Debug.Log("Razao molar "+Razao.ToString("f4"));
1445.         GrabResult ("Razao Molar06=" + Razao.ToString ());
1446.         //Dosagem de catalisador
1447.         float PorcCatalisador = 0.91f;//padrao devido ao vazamento
1448.         if (StatusEquipamentos[1]) {PorcCatalisador=30f*(ValoresEquipamentos[1]/LimiteMaxEquipa
1449.         mentos[1]);}

```

```

1447.         if (StatusEquipamentos[3]) {PorcCatalisador=36.36f*(ValoresEquipamentos[3]/LimiteMaxEqui
   pamentos[3]);}
1448.         if (StatusEquipamentos[1] && StatusEquipamentos[3]) {PorcCatalisador=30f*(ValoresEquipamento
   s[1]/LimiteMaxEquipamentos[1])+36.36f*(ValoresEquipamentos[3]/LimiteMaxEquipamentos[3]);}

1449.         GrabResult ("Catalisador10=" + PorcCatalisador.ToString ());
1450.         //Estimativa da conversao
1451.         float Conv = 0f;
1452.         if (StatusEquipamentos [6] && (StatusEquipamentos [1] || StatusEquipamentos [3]) && St
   atusEquipamentos [7] {
1453.             float x = ValoresEquipamentos [0] / LimiteMaxEquipamentos [0]; //Refluxo (Normaliz
   ado 0-1)
1454.             //UnityEngine.Debug.Log("Refluxo "+x.ToString("f4"));
1455.             float y = (ValoresEquipamentos [4] / LimiteMaxEquipamentos [4]) * 1000f; //Carga t
   ermica (Watts)
1456.             //UnityEngine.Debug.Log("Carga termica "+y.ToString("f4"));
1457.             float z = Razao; //Razao molar
1458.             if (z != z) {z = 0.0f;} //verifica se e NaN
1459.             //float z = Media (PartOfArray (ValoresSensores, 6, 0)); //Razao molar
1460.             //UnityEngine.Debug.Log("Razao Molar "+z.ToString("f4"));
1461.             Conv = 2.9122f * x + 0.22245f * y + 3.1834f * z + 0.017755f * x * y -
   0.008923f * y * z - 0.27153f * x * z; //ajuste dos dados experimentais no matlab
1462.             Conv = Conv/(0.9f+(FluxoOleo+FluxoAlcool)/1.85f); //minhas correcoes
1463.             if (Conv<0f){Conv=0f;}
1464.             if (Conv>99.9f){Conv=99.9f;}
1465.         }
1466.         GrabResult ("Conversão09=" + Conv.ToString ());
1467.
1468.     }
1469.
1470.     //Lembretes ao usuario
1471.     public void Lembretes(){
1472.
1473.         string FalaLembrete = null;
1474.         if (StatusEquipamentos [1] || StatusEquipamentos [3]) {
1475.             FalaLembrete+=" Catalisador";
1476.         }
1477.         if (StatusEquipamentos [6]) {
1478.             if (StatusEquipamentos [1] || StatusEquipamentos [3]) {FalaLembrete += " e";}
1479.             FalaLembrete+="Alcool";
1480.         }
1481.         if (StatusEquipamentos [7]) {
1482.             if (StatusEquipamentos [6] || StatusEquipamentos [1] || StatusEquipamentos [3]) {Fa
   laLembrete += " e";}
1483.             FalaLembrete+="Oleo";
1484.         }
1485.         if (ValoresAtuaisSensores[11]>LimiteMaxSensores[11]){//verificando tanque de resfriame
   nto
1486.             if (FalaLembrete!=null){FalaLembrete += "... e também";}
1487.             FalaLembrete+="Adicione mais gelô ao tanque de resfriamento";

```

```
1488.        }
1489.        if (FalaLembrete != null) {
1490.            if (StatusEquipamentos [6] || StatusEquipamentos [1] || StatusEquipamentos [3] ||
1491.                StatusEquipamentos [7]){
1492.                    Speak ("Verifique os níveis de" + FalaLembrete);
1493.                }
1494.            }
1495.
1496.        private int COMPortNumber(string COMPortName)
1497.        {
1498.            int num = COMPortName.IndexOf("M");
1499.            string s = string.Empty;
1500.            s = COMPortName.Substring(num + 1, COMPortName.Length - 1 - num);
1501.            return int.Parse(s);
1502.        }
1503.
1504.        //Testar conexão com a Internet
1505.        //using System.Net;
1506.        public static bool CheckForInternetConnection()
1507.        {
1508.            try{using (var client = new WebClient())
1509.                {using (var stream = client.OpenRead("http://www.google.com"))
1510.                    {return true;}}
1511.                }
1512.            }
1513.            catch
1514.            {return false;}
1515.        }
1516.
1517.        public void EnviarResultadosViaEmail(){
1518.            String Log = Application.dataPath + "/Log.txt";
1519.            //Com Internet
1520.            if (CheckForInternetConnection()){
1521.                Speak ("relatório enviado para os e-mails cadastrados");
1522.                AddText (Log, "relatório enviado para os e-mails cadastrados");
1523.                ConsoleString += "\r\n"+System.DateTime.Now.ToString("hh.mm.ss")+" "+"relatório e
1524.                nviado para os e-mails cadastrados";
1525.                scrollPosition += new Vector2 (0, 15000);
1526.
1527.                AssuntoEmail = "Relatorio do supervisório da "+ProjectName;
1528.                CorpoEmail = "Resultados do supervisório da "+ProjectName;
1529.                SendEmail(destinatarios,AssuntoEmail, CorpoEmail, AtachedFiles);
1530.                EmailsEnviados+=1;
1531.            }
1532.            //Sem Internet
1533.            if (!CheckForInternetConnection()){
1534.                Speak ("relatório não enviado devido a problemas de conexão com a Internet");
1535.                AddText (Log, "relatório não enviado devido a problemas de conexão com a Internet"
1536.            );
1537.
```

```

1535.             ConsoleString += "\r\n"+System.DateTime.Now.ToString("hh.mm.ss")+" "+ "relatório n
    ào enviado devido a problemas de conexão com a Internet";
1536.             scrollPosition += new Vector2 (0, 15000);
1537.         }
1538.     }
1539.
1540.     //enviar e-mail
1541.     public void SendEmail(string[] receivers, string Assunto, string Corpo, string[] PathAttachedFile)
1542.     {
1543.         using (var mail = new MailMessage {
1544.             From = new MailAddress("eva.supervisorio@gmail.com"),
1545.             Subject = Assunto,
1546.             Body = Corpo
1547.         }) {
1548.             for(int i = 0; i < receivers.Length; i++)
1549.             {
1550.                 mail.To.Add(receivers[i]);
1551.             }
1552.
1553.             for(int j = 0; j < PathAttachedFile.Length; j++)
1554.             {
1555.                 System.Net.Mail.Attachment attachment = new System.Net.Mail.Attachment (Application.dataPath+"\\"+PathAttachedFile[j]);
1556.                 mail.Attachments.Add (attachment);
1557.             }
1558.
1559.             var smtpServer = new SmtpClient("smtp.gmail.com") {
1560.                 Port = 587,
1561.                 Credentials = (ICredentialsByHost)new NetworkCredential("eva.supervisorio@gmail.com", "*****")
1562.             };
1563.             ServicePointManager.ServerCertificateValidationCallback = delegate { return true;
1564.             };
1565.             smtpServer.EnableSsl = true;
1566.             smtpServer.Send(mail);
1567.         }
1568.
1569.         //Checar o status dos sensores
1570.         public void CheckSensorStatus()
1571.         {
1572.             string FalaTemp = "";
1573.             int SensoresSemResposta = 0;
1574.             for (int i = 0; i < Sensores.Length; i++) {
1575.                 SensTempoDesdeUltimaAtualizacao[i]=Time.time-SensTempoUltimaAtualizacao[i];
1576.                 if (SensTempoDesdeUltimaAtualizacao[i]>43f){
1577.                     SensoresSemResposta+=1;
1578.                     FalaTemp+=" O "+Sensores [i] + " " + i.ToString ("D2") +" "+DescriçaoSensor[i]
1579.                     + " parou de funcionar a "+IntuitiveTime(SensTempoDesdeUltimaAtualizacao[i])+"...";
1580.                 }
1581.             }
1582.             if (SensoresSemResposta>=10)
1583.             {
1584.                 FalaTemp+="\n\n";
1585.                 FalaTemp+="\n";
1586.                 FalaTemp+="\n";
1587.             }
1588.         }
1589.     }
1590.
```

```

1579.         }
1580.     }
1581.     //Condições específicas
1582.     if (StatusEquipamentos [4] && ValoresAtuaisSensores [7] < 600f) {FalaTemp+=" O referve
1583.         dor está ligado, mas não há corrente significativa passando por ele...";}
1584.         if (StatusEquipamentos [7] && ValoresAtuaisSensores [4] == 0f) {FalaTemp+=" A bomba de
1585.         óleo está ligada, mas não há vazão no sensor de fluxo de óleo...";}
1586.         if (StatusEquipamentos [6] && ValoresAtuaisSensores [5] == 0f) {FalaTemp+=" A bomba de
1587.         álcool está ligada, mas não há vazão no sensor de fluxo de álcool...";}
1588.
1589.     }
1590.
1591.     //Graficos
1592.     public Rect DisplayDebugGUI(String Titulo, String Xlabel, String Ylabel, int PositionX, int
1593.     PositionY, int Largura, int Altura, float[] Xvalues, float[] YValues, Texture LinhasoGraphH, T
1594.     exture TexturaFundoGraphH, Texture graphHTexture, float MinValue, float MaxValue, Color ColorDe
1595.     fault, Color ColorLow, Color ColorHigh)
1596.     {
1597.         if (graphHTexture != null)
1598.         {
1599.             int labelWidth = 150;
1600.             int labelHeight = 51;
1601.             int ButtonSize = 32;
1602.             int XLeftOver = 60;
1603.             int YLeftOver = 40;
1604.             int padding = 2;
1605.             int frameWidth = Largura;
1606.             int frameHeight = Altura;
1607.             int barWidth=Mathf.RoundToInt((frameWidth-2*XLeftOver)/Xvalues.Length);
1608.             int barHeight=frameHeight - 2*YLeftOver;
1609.             Rect frameRect = new Rect(PositionX - padding, PositionY -
1610.             padding, frameWidth, frameHeight);
1611.             GUI.BeginGroup(frameRect);
1612.             var centeredStyle = GUI.skin.GetStyle("Label");
1613.             centeredStyle.alignment = TextAnchor.MiddleCenter;
1614.             centeredStyle.fontSize=labelHeight/3-6;
1615.             var gs = GUI.skin.GetStyle("Button");
1616.             gs.fontSize=labelHeight/3-7;
1617.             GUI.color = new Color(0.5f, 0.5f, 0.5f, 1f);
1618.             GUI.DrawTexture(new Rect(0, 0, frameRect.width, frameRect.height), TexturaFundoGra
1619.             ph);

```

```

1619.         GUI.color = Color.white;
1620.
1621.         //Titulo
1622.         GUI.Label(new Rect(XLeftOver, YLeftOver/2-labelHeight/2, frameWidth-
1623.             2*XLeftOver, labelHeight + 4), Titulo);
1623.
1624.         //Valores do eixo Y
1625.         GUI.Label(new Rect(XLeftOver/2-labelWidth/2, frameHeight/2-
1626.             labelHeight/2, labelWidth, labelHeight + 3), Ylabel);
1626.         GUI.Label(new Rect(XLeftOver/2-labelWidth/2, YLeftOver-
1627.             labelHeight/2, labelWidth, labelHeight + 3), MaxValue.ToString("F2"));
1627.         GUI.Label(new Rect(XLeftOver/2-labelWidth/2, frameHeight - YLeftOver-
1628.             labelHeight/2, labelWidth, labelHeight + 3), MinValue.ToString("F2"));
1628.
1629.         //Valores do eixo X
1630.         GUI.Label(new Rect(frameWidth/2-labelWidth/2,frameHeight-
1631.             YLeftOver/2+labelHeight/10-labelHeight/2, labelWidth, labelHeight + 3), Xlabel);
1631.         GUI.Label(new Rect(XLeftOver-labelWidth/2, frameHeight - YLeftOver+labelHeight/6-
1632.             labelHeight/2, labelWidth, labelHeight + 3), Xvalues[Xvalues.Length-1].ToString("F2"));
1632.         GUI.Label(new Rect(XLeftOver+(frameWidth-2*XLeftOver)/4-
1633.             labelWidth/2, frameHeight - YLeftOver+labelHeight/6-
1634.             labelHeight/2, labelWidth, labelHeight + 3), Xvalues[3*Mathf.RoundToInt(Xvalues.Length/4)-
1635.             1].ToString("F2"));
1635.         GUI.Label(new Rect(frameWidth/2-labelWidth/2, frameHeight -
1636.             YLeftOver+labelHeight/6-
1637.             labelHeight/2, labelWidth, labelHeight + 3), Xvalues[Mathf.RoundToInt(Xvalues.Length/2)-
1638.             1].ToString("F2"));
1638.
1639.         //Grid
1640.         //Horizontal
1641.         GUI.DrawTexture(new Rect(XLeftOver, YLeftOver, frameWidth-
1642.             2*XLeftOver, 1), LinhasoGraphH);
1642.         GUI.DrawTexture(new Rect(XLeftOver, frameHeight/2, frameWidth-
1643.             2*XLeftOver, 1), LinhasoGraphH);
1643.         //Vertical
1644.         GUI.DrawTexture(new Rect((frameWidth-
1645.             2*XLeftOver)/4+XLeftOver, YLeftOver, 1, frameHeight-2*YLeftOver), LinhasoGraphH);
1645.         GUI.DrawTexture(new Rect(frameWidth/2, YLeftOver, 1, frameHeight-
1646.             2*YLeftOver), LinhasoGraphH);

```

```

1646.             GUI.DrawTexture(new Rect(3*(frameWidth-
    2*XLeftOver)/4+XLeftOver, YLeftOver, 1, frameHeight-2*YLeftOver), LinhasoGraphH);
1647.             //Margins
1648.             GUI.color = 0.8f*Color.blue+0.5f*Color.green;
1649.             GUI.DrawTexture(new Rect(0, 0, frameWidth, 1), LinhasoGraphH);
1650.             GUI.DrawTexture(new Rect(0, frameHeight - 1, frameWidth, 1), LinhasoGraphH);
1651.             GUI.DrawTexture(new Rect(0, 0, 1, frameHeight), LinhasoGraphH);
1652.             GUI.DrawTexture(new Rect(frameWidth - 1, 0, 1, frameHeight), LinhasoGraphH);
1653.             GUI.DrawTexture(new Rect(frameWidth - 1-ButtonSize-
    4*padding, 0, 1, frameHeight), LinhasoGraphH);
1654.
1655.             GUI.color = 0.5f*Color.blue+0.3f*Color.green+0.3f*Color.white;
1656.             //Botões
1657.             if (GUI.Button(new Rect(frameWidth-ButtonSize-padding/2, YLeftOver/2-
    ButtonSize/2+4, ButtonSize-4, ButtonSize-
    4),Save_Icon)){SavePartOfScreen(Mathf.RoundToInt(0.24f * Screen.width), Mathf.RoundToInt(0.04f
    * Screen.height+5), Mathf.RoundToInt(0.43f * Screen.width), Mathf.RoundToInt(0.68f * Screen.h
    eight), Titulo+" "+System.DateTime.Now.ToString("hh.mm.ss")+" "+System.DateTime.Now.ToString(
    "dd-MM-yyyy"));}
1658.
1659.             if (GUI.Button(new Rect(frameWidth-ButtonSize-padding/2, YLeftOver/2-
    ButtonSize/2+4+2*padding+ButtonSize, ButtonSize-4, ButtonSize/2+2),"10m")){}
1660.             if (GUI.Button(new Rect(frameWidth-ButtonSize-padding/2, YLeftOver/2-
    ButtonSize/2+4+3*padding+3*ButtonSize/2, ButtonSize-4, ButtonSize/2+2),"1h")){}
1661.             if (GUI.Button(new Rect(frameWidth-ButtonSize-padding/2, YLeftOver/2-
    ButtonSize/2+4+4*padding+4*ButtonSize/2, ButtonSize-4, ButtonSize/2+2),"1d")){}
1662.             if (GUI.Button(new Rect(frameWidth-ButtonSize-padding/2, YLeftOver/2-
    ButtonSize/2+4+5*padding+5*ButtonSize/2, ButtonSize-4, ButtonSize/2+2),"1s")){}
1663.
1664.             for(int i = 0; i < YValues.Length; i++)
1665.             {
1666.                 float perc = ((YValues[i] - MinValue) / (.MaxValue - MinValue));
1667.                 GUI.color = new Color(1f, 1f, 1f, 0.25f);
1668.                 //GUI.color = ColorDefault;
1669.                 if (perc<=0.1f){GUI.color = ColorLow;}
1670.                 if (perc>=0.9f){GUI.color = ColorHigh;}
1671.                 if (!float.IsNaN(perc) && !float.IsInfinity(perc)){
1672.                     GUI.DrawTexture(new Rect(XLeftOver+barWidth*i,frameHeight -
    YLeftOver, barWidth, -(int)((float)barHeight)*perc)), graphTexture);
1673.                     GUI.color = Color.white;
1674.                     GUI.DrawTexture(new Rect(XLeftOver+barWidth*i,frameHeight-YLeftOver-
    (int)((float)barHeight)*perc), barWidth,1), graphTexture);
1675.                 }
1676.             }
1677.
1678.             GUI.color = Color.white;

```

```

1679.
1680.        GUI.EndGroup();
1681.
1682.        return frameRect;
1683.    }
1684.    return new Rect(0,0,0,0);
1685. }
1686.
1687. // Display sensor stats
1688. public Rect DisplaySensorStats(int SensorNumber, String Titulo, int PositionX, int Positio
   nY, int Largura, int Altura, float[] Xvalues, float[] YValues, float Max, float Min, Texture
   ColorIndicator)
1689. {
1690.     int labelWidth = 200;
1691.     int labelHeight = 11 + Screen.width / 1000;
1692.     int IndicatorSize = 32;
1693.     int XLeftOver = 30;
1694.     int YLeftOver = 30;
1695.     int padding = 2;
1696.     int frameWidth = Largura;
1697.     int frameHeight = Altura;
1698.     Rect frameRect = new Rect(PositionX - padding, PositionY -
   padding, frameWidth, frameHeight);
1699.
1700.     GUI.BeginGroup(frameRect);
1701.
1702.     var centeredStyle = GUI.skin.GetStyle("Label");
1703.     centeredStyle.alignment = TextAnchor.MiddleCenter;
1704.     centeredStyle.fontSize=11 + Screen.width / 1000;
1705.
1706.     GUI.color = new Color(0.5f, 0.5f, 0.5f, 1f);
1707.     GUI.DrawTexture(new Rect(0, 0, frameRect.width, frameRect.height), TexturaFundoGra
   pH);
1708.     GUI.color = Color.white;
1709.
1710.     //Titulo
1711.     GUI.Label(new Rect(XLeftOver, YLeftOver/2-labelHeight/2, frameWidth-
   2*XLeftOver, labelHeight + 4), Titulo);
1712.
1713.     centeredStyle.alignment = TextAnchor.MiddleLeft;
1714.     //Informações
1715.     GUI.Label(new Rect(IndicatorSize+XLeftOver+5, YLeftOver + 3f*labelHeight/2, labelWidth
   , labelHeight + 3), "Media: "+Media(YValues).ToString("F3")+" "+UnidadeSensores[SensorNumber])
   ;
1716.     GUI.Label(new Rect(IndicatorSize+XLeftOver+5, YLeftOver + 3f*labelHeight, labelWidth,
   labelHeight + 3), "Minimo: "+Min.ToString("F3")+" "+UnidadeSensores[SensorNumber]);
1717.     GUI.Label(new Rect(IndicatorSize+XLeftOver+5, YLeftOver + 9f*labelHeight/2, labelWidt
   h, labelHeight + 3), "Maximo: "+Max.ToString("F3")+" "+UnidadeSensores[SensorNumber]);

```

```

1718.         GUI.Label(new Rect(IndicatorSize+XLeftOver+5, YLeftOver + 6f*labelHeight, labelWidth,
1719.                         labelHeight + 3), "Taxa: "+TaxaFinal(Xvalues,YValues).ToString("F4")+" "+UnidadeSensores[SensorNumber]+"/min");
1720.         //Datas
1721.         centeredStyle.alignment = TextAnchor.MiddleRight;
1722.         GUI.Label(new Rect(frameWidth/2, YLeftOver + 3f*labelHeight/2, labelWidth, labelHeight
1723.                         + 3), "Ultimos 10 minutos");
1724.         GUI.Label(new Rect(frameWidth/2, YLeftOver + 3f*labelHeight, labelWidth, labelHeight
1725.                         + 3), MinDataSensores[SensorNumber]);
1726.         GUI.Label(new Rect(frameWidth/2, YLeftOver + 9f*labelHeight/2, labelWidth, labelHeight
1727.                         + 3), MaxDataSensores[SensorNumber]);
1728.         centeredStyle.alignment = TextAnchor.MiddleCenter;
1729.         GUI.color = Color.white;
1730.         //Indicadores
1731.         GUI.color = Color.green;
1732.         if (Media (YValues) >= LimiteMaxSensores [SensorNumber] || Media (YValues) <= LimiteMi
1733.             nSensores [SensorNumber]) {GUI.color = Color.red;}
1734.         GUI.DrawTexture(new Rect(XLeftOver, YLeftOver + 3f*labelHeight/2, IndicatorSize, label
1735.                         Height), ColorIndicator);
1736.         GUI.color = Color.green;
1737.         if (Min >= LimiteMaxSensores [SensorNumber] || Min <= LimiteMinSensores [SensorNumber]
1738. ) {GUI.color = Color.yellow;}
1739.         GUI.DrawTexture(new Rect(XLeftOver, YLeftOver + 3f*labelHeight, IndicatorSize, labelHe
1740.                         ght), ColorIndicator);
1741.         GUI.color = Color.green;
1742.         GUI.DrawTexture(new Rect(0, YLeftOver, frameWidth, 1), LinhasoGraphH);
1743.         GUI.DrawTexture(new Rect(0, 0, frameWidth, 1), LinhasoGraphH);
1744.         GUI.DrawTexture(new Rect(0, frameHeight - 1, frameWidth, 1), LinhasoGraphH);
1745.         GUI.DrawTexture(new Rect(0, 0, 1, frameHeight), LinhasoGraphH);
1746.         GUI.DrawTexture(new Rect(frameWidth - 1, 0, 1, frameHeight), LinhasoGraphH);
1747.
1748.         GUI.color = Color.white;
1749.
1750.         GUI.EndGroup();
1751.
1752.         return frameRect;
1753.     }
1754.
1755.     public static float Media(float[] Entrada)

```

```

1756.    {
1757.        float CumResult = 0;
1758.        for (int i = 0; i < Entrada.Length; i++) {
1759.            CumResult+=Entrada[i];
1760.        }
1761.        CumResult = CumResult / Entrada.Length;
1762.        return CumResult;
1763.    }
1764.
1765.    public static float TaxaFinal(float[] EntradaX, float[] EntradaY)
1766.    {
1767.        float Taxa = 0;
1768.        Taxa = 0.5f*(EntradaY[EntradaY.Length-1]-EntradaY[EntradaY.Length-
2])/((EntradaX[EntradaX.Length-1]-EntradaX[EntradaX.Length-2])+0.5f*(EntradaY[EntradaY.Length-
2]-EntradaY[EntradaY.Length-3])/((EntradaX[EntradaX.Length-2]-EntradaX[EntradaX.Length-3]));
1769.        return Taxa;
1770.    }
1771.    public float SegundaDerivadaFinal(float[] EntradaX, float[] EntradaY)
1772.    {
1773.        float Taxa = 0;
1774.        Taxa = ((EntradaY[EntradaY.Length-1]-EntradaY[EntradaY.Length-
2])/((EntradaX[EntradaX.Length-1]-EntradaX[EntradaX.Length-2])-((EntradaY[EntradaY.Length-2]-
EntradaY[EntradaY.Length-3])/((EntradaX[EntradaX.Length-2]-EntradaX[EntradaX.Length-
3])))/((EntradaX[EntradaX.Length-2]-EntradaX[EntradaX.Length-3]));
1775.        return Taxa;
1776.    }
1777.
1778.    public static void CubicInterpolation(float[] x, float[] y, out float[] Xout, out float[]
Yout, int NumberOfPoints)
1779.    {
1780.        //using TestMySpline
1781.
1782.        // Create the upsampled X values to interpolate
1783.        int n = NumberOfPoints;
1784.        Xout = new float[n];
1785.        Yout = new float[n];
1786.        float stepSize = (x[x.Length - 1] - x[0]) / (n - 1);
1787.
1788.        for (int i = 0; i < n; i++)
1789.        {
1790.            Xout[i] = x[0] + i * stepSize;
1791.        }
1792.
1793.        // Fit and eval
1794.        CubicSpline spline = new CubicSpline();
1795.        Yout = spline.FitAndEval(x, y, Xout);
1796.
1797.    }
1798.
1799.    public static float[] PartOfArray(float[,] InputArray, int Line, int Column)

```

```
1800.    {
1801.        float[] array = new float[InputArray.GetLength(2)];
1802.        for (int i = 0; i < InputArray.GetLength(2); i++)
1803.        {
1804.            array[i] = InputArray[Line, Column, i];
1805.        }
1806.        return array;
1807.    }
1808.    public void InterpolationToPlot(int Linha, int Coluna, float[,] XInputArray, float[,] YI
nputArray, float[,] XOutputArray, float[,] YOutputArray, out float[,] XResult, out float[,]
YResult)
1809.    {
1810.        XResult = XOutputArray;
1811.        YResult = YOutputArray;
1812.        float[] x = new float[YInputArray.GetLength(2)];
1813.        float[] y = new float[YInputArray.GetLength(2)];
1814.        float[] array = new float[YOutputArray.GetLength(2)];
1815.        float[] array2 = new float[YOutputArray.GetLength(2)];
1816.        x = PartOfArray(XInputArray, Linha, Coluna);
1817.        y = PartOfArray(YInputArray, Linha, Coluna);
1818.        CentralArduino.CubicInterpolation(x, y, out array, out array2, YOutputArray.GetLength(
2));
1819.        for (int i = 0; i < YOutputArray.GetLength(2); i++)
1820.        {
1821.            XResult[Linha, Coluna, i] = array[i];
1822.            YResult[Linha, Coluna, i] = array2[i];
1823.        }
1824.    }
1825.    private string ShowArrayLine(float[,] InputArray, int Line, int Column)
1826.    {
1827.        string text = string.Empty;
1828.        for (int i = 0; i < InputArray.GetLength(2); i++)
1829.        {
1830.            text = text + " " + InputArray[Line, Column, i].ToString();
1831.        }
1832.        return text;
1833.    }
1834.
1835.    private string ShowArray(float[] InputArray)
1836.    {
1837.        string text = string.Empty;
1838.        for (int i = 0; i < InputArray.GetLength(0); i++)
1839.        {
1840.            text = text + " " + InputArray[i].ToString("f4");
1841.        }
1842.        return text;
1843.    }
1844.
1845.    private string ShowArrayStrings(string[] InputArray)
1846.    {
```

```

1847.     string text = string.Empty;
1848.     for (int i = 0; i < InputArray.GetLength(0); i++)
1849.     {
1850.         text = text + InputArray[i]+i.ToString("D2")+" ";
1851.     }
1852.     return text;
1853. }
1854.
1855. //Relatorio ddo sensor i (preditivo)
1856. IEnumarator ReliableSetEquipment(string CodToArduino){
1857.     if (Arduino) {
1858.         sp.Write (CodToArduino);
1859.     }
1860.     yield return new WaitForSeconds(0.07f);
1861.     //sp.Write ("\n");
1862. }
1863.
1864. //Relatorio ddo sensor i (preditivo)
1865. IEnumarator RelatorioSensor(int i, string Tipo){
1866. ...
1867. }
1868. //Regressao linear de dados
1869. public void RegressaoLinear(float[] X, float[] Y, out float a, out float b, out float Rsquare){
1870.     float Xmedio = Media (X);
1871.     float Ymedio = Media (Y);
1872.     float Numerador = 0f;
1873.     float Denominador = 0f;
1874.     for (int i = 0; i < X.Length; i++) {
1875.         Numerador+=(X[i]-Xmedio)*(Y[i]-Ymedio);
1876.         Denominador+=Mathf.Pow((X[i]-Xmedio),2f);
1877.     }
1878.     a = Numerador / Denominador;
1879.     b = Ymedio - a * Xmedio;
1880.     float Sresidual = 0f;
1881.     float Stotal = 0f;
1882.     for (int i = 0; i < Y.Length; i++) {
1883.         Sresidual+=Mathf.Pow((Y[i]-a*X[i]-b),2f);
1884.         Stotal+=Mathf.Pow((Y[i]-Ymedio),2f);
1885.     }
1886.     Rsquare = 1f-Sresidual/Stotal;
1887. }
1888.
1889. public static void SavePartOfScreen(int PositionX, int PositionY, int Largura, int Altura,
String Name) {
1890.
1891.     //using System;
1892.
1893.     //criar pastas

```

```
1894.         if (!Directory.Exists (Application.dataPath + "/Screenshots")) {System.IO.Directory.Cr
1895.             eateDirectory (Application.dataPath + "/Screenshots");}
1896.             // Create a texture the size of the screen, RGB24 format
1897.             int width = Largura;
1898.             int height = Altura;
1899.             int X = PositionX;
1900.             int Y = PositionY;
1901.             Texture2D tex = new Texture2D(width, height, TextureFormat.RGB24, false);
1902.
1903.             // Read screen contents into the texture
1904.             tex.ReadPixels(new Rect(X, Y, width, height), 0, 0);
1905.             tex.Apply();
1906.
1907.             // Encode texture into PNG
1908.             byte[] bytes = tex.EncodeToPNG();
1909.             Destroy(tex);
1910.
1911.             //also write to a file in the project folder
1912.             File.WriteAllBytes(Application.dataPath + "/Screenshots/" + Name + ".png", bytes);
1913.
1914.         }
1915.
1916.         public void ShowHideHUD(){
1917.             bool Marker = true;
1918.             Audios [10].Play ();
1919.             if (HUD && Marker) {
1920.                 for (int i = 0; i < HUDO.Objects.Length; i++) {
1921.                     HUDO.Objects[i].SetActive(false);
1922.                 }
1923.                 Demonstrative=true;
1924.                 HUDO.operacao=false;
1925.                 Console=false;
1926.                 Stats=false;
1927.                 Graficos=false;
1928.                 HUD=!HUD;
1929.                 Marker = false;
1930.             }
1931.             if (!HUD && Marker) {
1932.                 for (int i = 0; i < HUDO.Objects.Length; i++) {
1933.                     HUDO.Objects[i].SetActive(true);
1934.                 }
1935.                 Demonstrative=false;
1936.                 HUDO.operacao=false;
1937.                 Console=false;
1938.                 Stats=true;
1939.                 Graficos=false;
1940.                 HUD=!HUD;
1941.                 Marker = false;
1942.             }
```

```
1943.    }
1944.
1945.    public static void PlaySound(int i)
1946.    {
1947.        StaticAudios[i].Play();
1948.    }
1949.
1950.    [RPC]
1951.    private void LigaDesligaEquipamento(int Numero, int intensity)
1952.    {
1953.        MarkStatusEquipamentos[Numero]=true;
1954.        ValoresEquipamentos [Numero] = intensity;
1955.    }
1956.    [RPC]
1957.    private void SendData(string ReceivedData)
1958.    {
1959.        RemoteData=ReceivedData;
1960.    }
1961.    [RPC]
1962.    private void UpdateEquipTime(string T, int Equip)
1963.    {
1964.        TempoHDEquipamentos[Equip]=T;
1965.    }
1966.    [RPC]
1967.    private void UpdateCicloEquip(string T, int Equip)
1968.    {
1969.        TempoCicloEquip[Equip]=T;
1970.    }
1971.    [RPC]
1972.    private void UpdateEquipValues()
1973.    {
1974.        HUDChanged = true;
1975.    }
1976.
1977.
1978. }
```

APÊNDICE C - CÁLCULO DA ACIDEZ DE ÓLEOS VEGETAIS

- 1) Em um erlenmeyer (50mL) previamente tarado misturar: 12,5 mL de álcool etílico, 0,5 mL de solução de fenolftaleína a 1% e uma gota da solução de NaOH a 0,25 mol/L;
- 2) Pesar entre 0,5 e 1,0 g de óleo e adicionar no erlenmeyer;
- 3) Titular o conjunto com uma solução de NaOH a 0,25 mol/L, com o auxilio de uma bureta de 10 mL;
- 4) A determinação do índice de acidez é realizada segundo a norma American Oil Chemists' Society (AOCS, Ca-5a-40), que consiste na determinação de ácidos graxos livres existentes em amostras de óleo brutos e refinados, e gorduras animais. O índice de acidez (I) é calculado conforme a equação:

$$I = 7,05 * V/m$$

Em que, V é o volume de hidróxido de sódio gasto na titulação e m é a massa de óleo e I;

Exemplos:

Óleo

Densidade de óleo = 0,912g/mL

Volume de óleo = 1 mL

Utilizado NaOH 3/20 mL = 0,15 mL

Índice de acidez = $7,05 * 0,15 / 0,912 = 1,15\%$

Biodiesel

Densidade de biodiesel = 0,865g/mL

Volume de biodiesel = 1 mL

Utilizado NaOH 1/20 mL = 0,05 mL

Índice de acidez = $7,05 * 0,05 / 0,865 = 0,387\%$

Biodiesel 90%

Densidade de biodiesel = 0,865g/mL

Volume de biodiesel = 1 mL

Utilizado NaOH (6.5 mL)/10 = 0,65 mL

Índice de acidez = $7,05 * 0,65 / 0,865 = 5,30\%$

APÊNDICE D - SCRIPT EM MATLAB PARA CÁLCULO DA QUANTIDADE DE REAGENTES E CATALISADOR EM UMA TRANSESTERIFICAÇÃO

```
%planejamento de transesterificação de óleo de algodão em biodiesel
clc;
clear;

%informações do usuário
VolumeMistura=200; %mL de Oleo e Biodiesel misturados
FracaoConvertida=0; %% de Biodiesel
RazaoMolar=12; %Alcool/Oleo
PorcentagemCatalisador=0.8; %% mássico, para etanol mínimo de 1.2%
AcidezMistura=1; %

%Informações prévias
%Oleo de Algodão
MassaMolarOleo=859.16; %g/mol
DensidadeOleo=0.912; %g/cm3

%Etanol
MassaMolarAlcool=46.06844; %g/mol
DensidadeAlcool=0.789; %g/cm3
PurezaAlcool=0.95; %

%NaOH
MassaMolarCatalisador=39.9971; %g/mol
ValenciaCatalisador=1; %hidroxilas ionizaveis
PurezaCatalisador=0.95; %

%Esteres
MassaMolarBiodiesel=301.76; %g/mol
DensidadeBiodiesel=0.865; %g/cm3

%H2SO4
MassaMolarNeutralizador=98.078; %g/mol
ValenciaNeutralizador=2; %hidrogênios ionizaveis
PurezaNeutralizador=0.92; %
DensidadeNeutralizador=1.84; %g/cm3

%Resultados
DensidadeMistura=DensidadeOleo*(1-
FracaoConvertida/100)+DensidadeBiodiesel*(FracaoConvertida/100); %g/mL
MassaMistura=VolumeMistura/DensidadeMistura; %g
MassaOleo=MassaMistura*(1-FracaoConvertida/100); %g
MolsOleo=MassaOleo/MassaMolarOleo; %mols
MassaAlcool=RazaoMolar*MolsOleo*MassaMolarAlcool; %g
VolumeAlcool=MassaAlcool/DensidadeAlcool; %mL
MassaCatalisador=MassaOleo*(PorcentagemCatalisador/100)+VolumeMistura*0.001
3*AcidezMistura; %g

MolsCatalisador=MassaCatalisador/MassaMolarCatalisador;
MolsNeutralizador=(ValenciaCatalisador/ValenciaNeutralizador)*MolsCatalisador;
MassaNeutralizador=MolsNeutralizador*MassaMolarNeutralizador;
VolumeNeutralizador=MassaNeutralizador/DensidadeNeutralizador;

disp('-----');
disp('Dados:');

```

```

fprintf('Volume Inicial %4.4f mL. Fração inicial convertida %4.4f
porcento\n',VolumeMistura,FracaoConvertida);
fprintf('Razão molar %4.4f mol/mol. Porcentagem de catalisador %4.4f
porcento\n',RazaoMolar,PorcentagemCatalisador);
fprintf('Acidez da mistura %4.4f porcento\n',AcidezMistura);
disp(' ');
fprintf('O volume necessário de álcool é %4.6f mL \n',VolumeAlcool);
disp(' ');
fprintf('A massa total de catalisador necessária é %4.6f g
\n',MassaCatalisador);
disp('Se for utilizar etóxido seque primeiro com silício gel.');
disp(' ');
fprintf('A massa de catalisador pra compensar a acidez é %4.6f g
\n',VolumeMistura*0.0013*AcidezMistura);
disp(' ');
fprintf('A massa de neutralizador pra compensar o catalisador é %4.6f g
\n',MassaNeutralizador);
fprintf('Equivalente a %4.6f mL \n',VolumeNeutralizador);
disp('A quantidade mínima de ácido a ser adicionada deve ser o dobro deste
valor.');
disp('Aconselha-se entre 5 e 10% de ácido na solução de lavagem.');
disp(' ');

```

EXEMPLO DE RESULTADO:

Dados:

Volume Inicial 200.0000 mL. Fração inicial convertida 0.0000 porcento
 Razão molar 12.0000 mol/mol. Porcentagem de catalisador 0.8000 porcento
 Acidez da mistura 1.0000 porcento

O volume necessário de álcool é 178.841728 mL

A massa total de catalisador necessária é 2.014386 g
 Se for utilizar etóxido seque primeiro com silício gel.

A massa de catalisador pra compensar a acidez é 0.260000 g

A massa de neutralizador pra compensar o catalisador é 2.469766 g
 Equivalente a 1.342264 mL
 A quantidade mínima de ácido a ser adicionada deve ser o dobro deste valor.
 Aconselha-se entre 5 e 10% de ácido na solução de lavagem.

APÊNDICE E - PROCEDIMENTO OPERACIONAL DE UTILIZAÇÃO DA COLUNA DE DESTILAÇÃO REATIVA

Preparativos iniciais

- Ligar computador 30 minutos antes
- Ligar a coluna e conectar ela ao computador simultaneamente
- Abrir supervisório
- Verificar todos os sensores e atuadores
- Esperar a confirmação de que tudo está OK de EVA
- Verificar posição das webcams
- Abastecer o recipiente de óleo com a quantia desejada
- Abastecer o recipiente de álcool com a quantia desejada
- Ligar refervedor
- Ligar bomba de álcool na vazão desejada
- Abastecer o tanque de resfriamento com gelo
- Verificar o nível do condensador
- Verificar temperaturas e níveis de álcool
- Alimentar o topo da coluna com catalisador
- Alterar a vazão do álcool para a desejada

Início da produção

- Ligar válvula 1 de catalisador por 5 minutos
- Ligar bomba de óleo na vazão desejada
- Modificar a vazão de álcool para atingir a razão molar desejada
- Coletar biodiesel no fundo da coluna
- Ligar válvula de catalisador na proporção desejada
- Esperar atingir o estado estacionário (15 minutos)
- Coletar biodiesel final na saída durante o período desejado

Termino do experimento

- Analisar/purificar a amostra como desejado
- Drenar excesso de álcool e óleo nos recipientes
- Lavar tanque de catalisador e coluna com álcool
- Lavar tanque de óleo com álcool e bombear álcool pelas tubulações de óleo

APÊNDICE F - FAZENDO A ANÁLISE ESTATÍSTICA DE UM PLANEJAMENTO FATORIAL NO STATISTICA 12.0

- 1- Abrir o programa Statistica 10.0
- 2- Clicar em arquivo novo, selecionar o número de variáveis e casos, clicar em OK.
- 3- Montar a planilha de dados com os fatores e a resposta.
- 4- Clicar no menu “DOE”, e clicar em “2**($k-p$)standard design”
- 5- Selecionar o tipo de planejamento experimental, clicar em OK. Selecionar o número de fatores clicar em OK.
- 6- No menu “Design of an experiment with two levels”, clicar no ícone “add to design” e indicar o número de pontos centrais, caso existam.
- 7- Clicar em “Summary” para observar o experimento aleatorizado pelo programa.
- 8- No ícone “Quick” selecionar “Change the factor names”, “values” etc e introduzir o nome dos fatores e os valores de máximo, mínimo e no ponto central.
- 9- Ainda na janela “Design of an experiment with two levels” clicar em cancelar e na nova janela clicar em “Analyze design” em seguida clicar em “variables” e selecionar a variável dependente e as independentes e por fim clicar em OK.
- 10- Na janela dos resultados ir ao ícone “Quick” e clicar em “ANOVA table” para análise estatística dos resultados. Nesta janela observa-se a importância dos fatores e das interações no modelo.
- 11- Clicar no ícone “Summary: effect estimates” e observar resultados.
- 12- Voltar na janela “Analyzes of an experiment” e clicar no ícone “Pareto Chart of effects”.

APÊNDICE G - DETALHAMENTO DOS CÁLCULOS ESTEQUIOMÉTRICOS

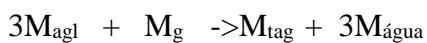
Os cálculos estequiométricos para reações de transesterificação são complexos e extensos, e serão detalhados neste apêndice.

Na Tabela 4 é possível observar Composição mássica do óleo de algodão, e também a semelhança entre os dados expostos. Os dados físicos segundo (RAMADHAS et al., 2005) estão expostos na Tabela 5.

Tendo em vista essas informações e levando em consideração as massas molares dos ácidos em questão, na Tabela 6 é possível observar a massa molar dos ácidos que compõem o óleo de algodão.

A massa molar média do ácido foi calculada pela transformação em fração molar e se retirou a média ponderada, o que leva a 273.8 g/mol.

A massa molar média de um triacilglicerol proveniente de algodão pode ser dada seguindo a estequimetria da Equação 16:



$$3 \times 273,7 \text{ g/mol} + 92,0776 \text{ g/mol} \rightarrow X + 3 \times 18,01528 \text{ g/mol}$$

$$X = 859,2 \text{ g/mol}$$

Massa molar média dos triacilglicerois presentes no óleo de algodão 859,2 g/mol

Para a transesterificação se tem a Equação 17, que resulta nos seguintes cálculos estequiométricos:

Segundo os cálculos estequiométricos:

Massa molar media do triacilglicerol: 859,1648 g/mol

Massa molar etanol: 46,06844 g/mol

Massa molar glicerina: 92,09382 g/mol

Massa molar media do éster: X

Por tanto a massa molar média de um éster proveniente de algodão é:

$$859,1648 + 3 \times 46,06844 \rightarrow 92,09382 + 3 \cdot X$$

$$X = 301,7588 \text{ g/mol}$$

Massa molar media do éster: 301,7588 g/mol

O método utilizado e o valor encontrado estão coerentes com o que é encontrado na literatura (DEMIRBAS, 2003).

Para definição da quantidade de reagente utilizado, em um caso específico, na conversão uma série de fatores tem que ser levada em consideração:

Partindo de 200mL em volume de óleo:

$$\text{Densidade média} = 0,912 = 0,912 \text{ g/cm}^3 = 0,912 \text{ kg/L}$$

$$\text{Massa estimada} = 0,2\text{L} \cdot 0,912\text{kg/L} = 0,1824 \text{ kg} = 182,4 \text{ g}$$

Tomando então 180 g:

$$\text{Número de mols do óleo} = 180 \text{ g} / 859,16 \text{ g/mol} = 0,2095 \text{ mols}$$

Será utilizada uma razão molar de 12/1 (álcool/ óleo) para converter o óleo.

Necessitando de uma quantidade de álcool de:

$$\text{Massa de etanol necessária} = 12 \text{ mol/mol} * 0,2095\text{mol} * 46,06844 \text{ g/mol} = 115,8 \text{ gramas}$$

$$\text{Em volume} = 115,8 \text{ g} / 0,789\text{g/cm}^3 = 146,8 \text{ mL de etanol}$$

A massa de catalisador necessária será de 0,8% em massa em relação ao óleo:

$$182,4 \text{ g} \cdot (0,8\% / 100) = 1,5 \text{ g de NaOH (dissolvido em parte do álcool adicionado)}$$

Mais o NaOH para compensar a acidez do óleo, no caso 1%, onde 0,0013g/mL é o fator correspondente para neutralização de 1 grama de óleo com 1% de teor de acidez:

$$200\text{mL} \cdot 0,0013\text{g/mL} \cdot 1\% = 0,26\text{g}$$

O índice de acidez foi calculado seguindo a norma padrão, exposta no Apêndice C.

Equivalente a:

$$1,50\text{g} + 0,26\text{g} = 1,76 \text{ g de NaOH}$$

$$1,76 \text{ g} / 39,9971\text{g/mol} = 0,0440 \text{ mol de NaOH}$$

Tanto o álcool adicionado, quanto a mistura catalisador/álcool precisam ser secadas para eliminação da uma eventual fração de água, o CaO pode ser utilizado pois na presença de água como na Equação 18. O hidróxido resultante além de remover a água é insolúvel em etanol, uma simples decantação pode remover o álcool anidro do precipitado branco. Definindo f como fração em massa de água no álcool. Deve-se utilizar a seguinte quantidade de CaO:

M = massa de álcool hidratado que se quer desidratar

$f \cdot M$ = massa de água que se quer eliminar.

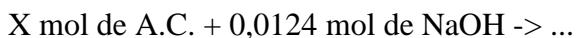
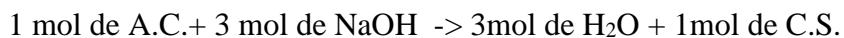
P=pureza do CaO (0 a 1)

Proporção da reação: 1 mol de CaO para 1 mol de H₂O, em massa 56,0774 de CaO para 18,0152g de água

$$\frac{56,0774}{P \cdot X} = \frac{18,0152}{f \cdot M}$$

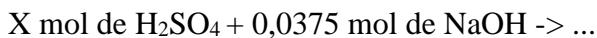
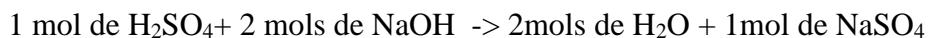
$$X = \frac{3,1129 \cdot f \cdot M}{P}$$

Após a reação o catalisador precisa ser neutralizado, é possível utilizar ácido cítrico monohidratado (C₆H₈O₇·H₂O MM=192,123+18,015=210,138g/mol) segundo a reação da Equação 20. A água da hidratação não participa da reação. Devido a quantidade de catalisador usado e pela estequiometria tem-se:



$$X = 0,004133 \text{ mol equivale a } 0,004133 \text{ g} \cdot 210,138 \text{ g/mol} = 0,8686 \text{ g de C}_6\text{H}_8\text{O}_7\text{H}_2\text{O}$$

Ou também pode ser neutralizada utilizando ácido sulfúrico (MM=98,078 g/mol) segundo a reação da Equação 21. A água da hidratação não participa da reação. Devido a quantidade de catalisador usado e pela estequiometria tem-se:



$$X = 0,01875 \text{ mol equivale a } 0,01875 \text{ mol} \cdot 98,078 \text{ g/mol} = 1,839 \text{ g de H}_2\text{SO}_4$$

APÊNDICE H - AVALIAÇÃO DO MODELO DE GENERALIZAÇÃO DO TEOR DE ÉSTER

```

clc
clear
%dados
x=[0      400 25;
1      400 25;
0      200 25;
1      200 25;
0      400 15;
1      400 15;
0      200 15;
1      200 15;
0.5 300 25;
0.5 400 20;
0.5 300 20]
y=[82;
91;
76;
81;
80;
88;
64;
73;
90;
87;
80]
modelfun = @(b,x)b(1).*x(:,1)+b(2).*x(:,2)+b(3).*x(:,3) + ...
b(4).*x(:,1).*x(:,2)+b(5).*x(:,2).*x(:,3)+b(6).*x(:,1).*x(:,3)+b(7);
beta0 = [0.1 0.1 0.1 0.1 0.1 0.1 0.1];
mdl = fitnlm(x,y,modelfun,beta0)

%computar e testar
x=1;
y=400;
z=25;
conv=91
R = 8.5*x+0.13103*y+1.9734*z + ...
0.0075*x*y-0.00375*y*z-0.15*x*z+20.043

%procurar o máximo
max=0;
param_max=zeros(1,3);
for m = 0:0.1:1
    for n = 200:400
        for o = 15:25
            R = 8.5*m+0.13103*n+1.9734*o + ...
            0.0075*m*n-0.00375*n*o-0.15*m*o+20.043;
            if R>max
                max=R;
                param_max=[m n o];
            end
        end
    end
end

max
param_max

%procurar o mínimo
min=100;

```

```
param_min=zeros(1,3);
for m = 0:0.1:1
    for n = 200:400
        for o = 15:25
            R = 8.5*m+0.13103*n+1.9734*o + ...
                0.0075*m*n-0.00375*n*o-0.15*m*o+20.043;
        if R<min
            min=R;
            param_min=[m n o];
        end
    end
end
min
param_min
```

APÊNDICE I - AVALIAÇÃO DO MODELO DE GENERALIZAÇÃO DA CONCENTRAÇÃO DE CATALISADOR RESIDUAL

```

clc
clear
%dados
x=[0      400 25;
1      400 25;
0      200 25;
1      200 25;
0      400 15;
1      400 15;
0      200 15;
1      200 15;
0.5 300 25;
0.5 400 20;
0.5 300 20]
y=[0.00048;
0.00061;
0.0013;
0.0026;
0.0040;
0.0040;
0.0087;
0.0088;
0.00075;
0.00062;
0.0037]
modelfun = @(b,x)b(1).*x(:,1)+b(2).*x(:,2)+b(3).*x(:,3) + ...
b(4).*x(:,1).*x(:,2)+b(5).*x(:,2).*x(:,3)+b(6).*x(:,1).*x(:,3)+b(7);
beta0 = [0.01 0.01 0.01 0.01 0.01 0.01 0.01];
mdl = fitnlm(x,y,modelfun,beta0)

%computar e testar
x=1;
y=400;
z=25;
conv=0.00061
R = 4.9996e-06*x-4.8878e-05*y-0.0010546*z ...
-3.175e-06*x*y+1.6725e-06*y*z+6.65e-05*x*z+0.0291

%procurar o máximo
max=0;
param_max=zeros(1,3);
for m = 0:0.1:1
    for n = 200:400
        for o = 15:25
            R = 4.9996e-06*m-4.8878e-05*n-0.0010546*o...
            -3.175e-06*m*n+1.6725e-06*n*o+6.65e-05*m*o+0.029153;
            if R>max
                max=R;
                param_max=[m n o];
            end
        end
    end
end

max
param_max

```

```
%procurar o minimo
min=100;
param_min=zeros(1,3);
for m = 0:0.1:1
    for n = 200:400
        for o = 15:25
            R = 4.9996e-06*m-4.8878e-05*n-0.0010546*o...
                -3.175e-06*m*n+1.6725e-06*n*o+6.65e-05*m*o+0.029153;
            if R<min
                min=R;
                param_min=[m n o];
            end
        end
    end
end
min
param_min
```

APÊNDICE J - AVALIAÇÃO DO MODELO DE GENERALIZAÇÃO DA CONCENTRAÇÃO DE SABÃO

```

clc
clear
%dados
x=[0      400 25;
1      400 25;
0      200 25;
1      200 25;
0      400 15;
1      400 15;
0      200 15;
1      200 15;
0.5 300 25;
0.5 400 20;
0.5 300 20]
y=[0.064;
0.058;
0.024;
0.022;
0.038;
0.031;
0.020;
0.008;
0.051;
0.061;
0.035]
modelfun = @(b,x)b(1).*x(:,1)+b(2).*x(:,2)+b(3).*x(:,3) + ...
b(4).*x(:,1).*x(:,2)+b(5).*x(:,2).*x(:,3)+b(6).*x(:,1).*x(:,3)+b(7);
beta0 = [0.01 0.01 0.01 0.01 0.01 0.01 0.01];
mdl = fitnlm(x,y,modelfun,beta0)

%computar e testar
x=1;
y=400;
z=25;
conv=0.058
R = -0.0185*x-1.751e-05*y-0.00096964*z ...
+2.5e-06*x*y+8.75e-06*y*z+0.00055*x*z+0.010655

%procurar o máximo
max=0;
param_max=zeros(1,3);
for m = 0:0.1:1
    for n = 200:400
        for o = 15:25
            R = -0.0185*m-1.751e-05*n-0.00096964*o...
            +2.5e-06*m*n+8.75e-06*n*o+0.00055*m*o+0.010655;
            if R>max
                max=R;
                param_max=[m n o];
            end
        end
    end
end

max
param_max

```

```
%procurar o minimo
min=100;
param_min=zeros(1,3);
for m = 0:0.1:1
    for n = 200:400
        for o = 15:25
            R = -0.0185*m-1.751e-05*n-0.00096964*o...
                +2.5e-06*m*n+8.75e-06*n*o+0.00055*m*o+0.010655;
            if R<min
                min=R;
                param_min=[m n o];
            end
        end
    end
end
min
param_min
```

ANEXO A - TABELA DE REAGENTES UTILIZADOS E SUAS INFORMAÇÕES

Reagente	Fornecedor	Pureza
Etanol	Neon	Mín 99,5%
Óleo de algodão	Usina de Caetés	-
Hidróxido de sódio	Dinâmica	Mín 98%
Óxido de cálcio	Dinâmica	Mín 95%
Ácido sulfúrico	Synth	98%
Ácido clorídrico	Dinâmica	0.1M(N)
Sílica Gel	Dinâmica	-
Acetona	Química moderna	Mín 99,5%
Fenoftaleína	Dinâmica	P.A.
Azul de bromofenol	Dinâmica	P.A.