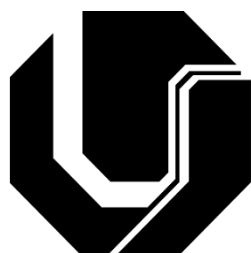


**UNIVERSIDADE FEDERAL UBERLÂNDIA
FACULDADE DE ENGENHARIA QUÍMICA
CURSO DE GRADUAÇÃO EM ENGENHARIA QUÍMICA**



**DESENVOLVIMENTO DE KITS EXPERIMENTAIS DE OPERAÇÃO
AUTOMÁTICA E BAIXO CUSTO PARA REATORES QUÍMICOS**

Maria Paula Virgílio Damis

UBERLÂNDIA-MG

2020

UNIVERSIDADE FEDERAL UBERLÂNDIA
FACULDADE DE ENGENHARIA QUÍMICA
CURSO DE GRADUAÇÃO EM ENGENHARIA QUÍMICA

**DESENVOLVIMENTO DE KITS EXPERIMENTAIS DE OPERAÇÃO
AUTOMÁTICA E BAIXO CUSTO PARA REATORES QUÍMICOS**

Maria Paula Virgílio Damis

Monografia de graduação apresentada a Universidade Federal de Uberlândia como parte dos requisitos necessários para a aprovação na disciplina Trabalho de Conclusão de curso, do curso de Engenharia Química.

UBERLÂNDIA-MG

2020

MEMBROS DA BANCA EXAMINADORA DA MONOGRAFIA DA DISCIPLINA TRABALHO DE CONCLUSÃO DE CURSO DE MARIA PAULA VIRGILIO DAMIS APRESENTADA À UNIVERSIDADE FEDERAL DE UBERLÂNDIA EM 09/10/2020.

BANCA EXAMINADORA:

Prof. Dr. Humberto Molinar Henrique

Orientador – FEQUI/UFU

Prof. Dr. Sérgio Mauro da Silva Neiro

FEQUI/UFU

Prof. Dr. Rubens Gedraite

FEQUI/UFU

AGRADECIMENTOS

Gostaria de agradecer, primeiramente, a minha família. Aos meus pais e ao meu irmão pelo apoio e confiança durante toda essa trajetória. Aos meus avós pelo exemplo que são para mim, principalmente nesse caminho de formação profissional.

Agradeço, em especial, meu orientador, Prof. Dr. Humberto Molinar, por toda a confiança depositada, pelo auxílio e compreensão durante esse último ano.

Agradeço, também, aos professores da Faculdade de Engenharia Química da Universidade Federal de Uberlândia que contribuíram para a minha formação.

SUMÁRIO

LISTA DE FIGURAS.....	v
LISTA DE TABELAS.....	ix
RESUMO.....	x
1 INTRODUÇÃO	1
2 A Plataforma Arduino	3
2.1 Introdução ao Arduino	3
2.2 Placas Arduino	4
2.2.1 Arduino UNO.....	4
2.2.2 Arduino Mega 2560	8
2.3 A IDE Arduino.....	8
3 Integração LabView – Arduino.....	12
3.1 Introdução	12
3.2 Interface LabVIEW – Arduino (LIFA).....	13
4 Atuadores e sensores para arduino	17
4.1 Introdução aos Atuadores, Sensores e <i>Shields</i> Utilizados	17
4.2 Manipulação de Vazão Usando a Minibomba RS385 12V DC.....	17
4.2.1 O Hardware	17
4.2.2 O Software	19
4.3 Medição de Vazão	24
4.3.1 O Hardware	24
4.3.2 O Software	27
4.4 Medição de Condutividade Elétrica.....	28
4.4.1 O Hardware	29
4.4.2 O Software	30
4.5 Medição de Massa (Célula de Carga)	32
4.5.1 O Hardware	32

4.5.2	O Software	34
5	Calibração dos sensores de vazão	36
5.1	Introdução	36
5.2	Modelo de Regressão Linear Simples.....	36
5.2.1	Estimação de Parâmetros em um Modelo de Regressão Linear Simples .	36
5.2.2	Estimando a Variância do Erro de Modelagem	38
5.2.3	Valor Esperado dos Parâmetros e Variância dos Estimadores de Mínimos Quadrados	39
5.2.4	Testando a Significância da Regressão Linear	39
5.2.5	Intervalo de Confiança para β_0 e β_1	40
5.3	Materiais	41
5.4	O <i>Software</i> de Calibração	42
5.5	Resultados e Discussão	43
6	Construção das unidades experimentais.....	50
6.1	Introdução	50
6.2	Unidade Experimental – Bateria de Reatores de Mistura.....	50
6.2.1	Projeto dos reatores	50
6.2.2	Materiais da unidade experimental	53
6.2.3	Custo estimado	55
6.3	Unidade Experimental – Reator tubular	55
6.3.1	Projeto do reator.....	56
6.3.2	Materiais da unidade experimental	57
6.3.3	Custo estimado	58
7	Distribuição de tempo de residência dos reatores CSTR e PFR.....	59
7.1	Introdução	59
7.2	Fundamentação teórica	59
7.2.1	Distribuição de tempo de residência	59

7.2.2	Tempo médio de residência	60
7.2.3	Técnica estímulo-resposta.....	61
7.2.4	DTR experimental.....	61
7.3	Materiais	62
7.4	Interface implementada para obtenção da DTR.....	62
7.4.1	Bateria de reatores CSTR.....	62
7.4.2	Reator PFR.....	63
7.5	Resultados e Discussão	64
7.5.1	Cálculo do Tempo Médio de Residência Experimental para a bateria de CSTR	64
7.5.2	Cálculo do Tempo Médio de Residência Experimental para o PFR.....	70
7.5.3	DTR Experimental e teórica da bateria de reatores de mistura ou tanque	75
7.5.4	DTR Experimental e teórico PFR	78
8	Determinação da Cinética Química da reação de Acetato de Etila e Hidróxido de Sódio em Reator Batelada	81
8.1	Introdução	81
8.2	Cinética da reação	81
8.3	Materiais e métodos	83
8.4	Interface implementada para a obtenção dos dados cinéticos	84
8.5	Resultados e discussões	85
8.5.1	Cálculo da constante da taxa de reação.....	85
8.5.2	Cálculo da Energia de ativação e do Fator pré-exponencial (A)	88
9	Conclusão.....	89
10	Sugestões para estudos futuros	91
11	Referências	92

LISTA DE FIGURAS

Figura 1. Foto de alguns modelos de placas Arduino.....	4
Figura 2. Arduino UNO e seus componentes.....	8
Figura 3. Arduino Mega 2560 e seus componentes.....	9
Figura 4. Janela inicial da IDE Arduino.....	9
Figura 5. Instalação do <i>toolkit</i> LIFA utilizando o gerenciador de pacotes VIPM.....	13
Figura 6. Bibliotecas usadas no painel frontal do <i>script</i> do LabVIEW para configurar o <i>hardware</i> Arduino.....	14
Figura 7. Bibliotecas usadas no diagrama de blocos do <i>script</i> do LabVIEW para acessar os recursos da placa Arduino.....	14
Figura 8. Bibliotecas usadas no diagrama de blocos do <i>script</i> do LabVIEW para: a) acessar valores de leitura e escrita nos pinos do Arduino; b) configurar sensores conectados ao Arduino.....	15
Figura 9. Detalhes do <i>motor shield L293D</i> : a). <i>Pinout</i> . b). Acoplada na Arduino UNO.....	18
Figura 10. Detalhes da minibomba RS385 e esquema de ligação elétrica ao <i>motor shield L293D</i>	19
Figura 11. Adicionando bibliotecas na IDE Arduino.....	20
Figura 12. Linhas de código para adicionadas à LIFA para configurar os sentidos de rotação dos motores 1 a 4.....	21
Figura 12 (<i>cont.</i>). Linhas de código para adicionadas à LIFA para configurar os sentidos de rotação dos motores 1 a 4 (continuação).....	22
Figura 14. Linhas de código para ajustar as velocidades de rotação dos motores 1 a 4.....	23
Figura 15. Linhas de código adicionadas à LIFA para parar os motores 1 a 4.....	23
Figura 16. Tensão Hall produzida pelo efeito Hall.....	24
Figura 17. Arranjo com ímã fixo com pá de rotor móvel de um material ferromagnético entre um sensor Hall e um ímã permanente.....	25
Figura 18. (a). Sensor de efeito Hall modelo YF-S402. (b). Sensor aberto mostrando detalhe do rotor.....	25
Figura 19. (a). Conexão do sensor YF-S402 ao Arduino. (b). Pinagem do sensor.....	26
Figura 20. Linhas de código para imprimir a duração de um pulso.....	27
Figura 21. Linhas de código incluídas no <i>Sketch</i> da LIFA para adquirir os sinais do sensor.....	28
Figura 22. (a). Condutímetro Digimed DM-32. (b). Conectores de entra e saída.....	29

Figura 23. Conversor/Adaptador USB para Serial RS232/DB9.	29
Figura 24. Entradas e saídas das sub-VI: (a). VISA Configure Serial Port VI. (b). VISA Read Function.	30
Figura 25. VI construída para ler dados do condutivímetro DM-32. (a). Diagrama de blocos. (b). Painel frontal.	31
Figura 26. (a). Célula de carga de 5 kg. (b). Amplificador HX711.....	33
Figura 27. Detalhes das ligações entre célula de carga, amplificador e placa Arduino.	33
Figura 28. Detalhes da balança construída, (a) foto superior (b) foto lateral.	34
Figura 29. Linhas de código adicionadas à LIFA para adquirir os sinais do amplificador HX711.	34
Figura 30. Linhas de código inserindo a função <i>ReadCount</i>	35
Figura 31. Diagrama esquemático do sistema de calibração do sensor de efeito Hall YF-S402.	41
Figura 32. <i>Script</i> LabVIEW para calibração dos sensores de vazão YF-S402. a). Painel frontal. b). Diagrama de blocos.	42
Figura 33. Volume <i>versus</i> tempo para cada sinal PWM enviado para a minibomba do sensor 1 da unidade PFR.	43
Figura 34. Curva de calibração do sensor 1 da unidade PFR.	44
Figura 35. Volume <i>versus</i> tempo para cada sinal PWM enviado para a minibomba do sensor 2 da unidade PFR.	45
Figura 36. Curva de calibração do sensor 2 da unidade PFR.	45
Figura 37. Volume <i>versus</i> tempo para cada sinal PWM enviado para a minibomba do sensor 1 da unidade de 3 CSTR em série.	46
Figura 38. Curva de calibração do sensor 1 da unidade de 3 CSTR em série.	47
Figura 39. Volume <i>versus</i> tempo para cada sinal PWM enviado para a minibomba do sensor 2 da unidade de 3 CSTR em série.	47
Figura 40. Curva de calibração do sensor 2 da unidade de 3 CSTR em série.	48
Figura 41. Distribuição do tempo de residência para o modelo n-CSTR.	51
Figura 42. Desenho técnico em escala do reator de mistura.	53
Figura 43. Representação esquemática da bateria de reatores de mistura.	54
Figura 44. Representação esquemática do reator Tubular.	57
Figura 45. <i>Script</i> LabVIEW para a obtenção da DTR para os CSTRs. a) - Painel frontal e (b) – Diagrama de blocos.	62

Figura 46. <i>Script</i> LabVIEW para a obtenção da DTR para o reator PFR. (a). Painel frontal e (b). Diagrama de blocos.	63
Figura 47. Distribuição tempo de residência experimental e ajustada para vazão de 400 mL/min	66
Figura 48. Distribuição tempo de residência experimental e ajustada para vazão de 600 mL/min	67
Figura 49. Distribuição tempo de residência experimental e ajustada para vazão de 800 mL/min	68
Figura 50. Distribuição tempo de residência experimental e ajustada para vazão de 800 mL/min	69
Figura 51. Distribuição tempo de residência experimental e ajustada para a vazão de 400 mL/min ($\lambda_0 = 0,0448\text{mS/cm}$ e $\lambda_\infty = 11,433\text{mS/cm}$)	71
Figura 51. Distribuição tempo de residência experimental e ajustada para a vazão de 600 mL/min ($\lambda_0 = 0,580\text{mS/cm}$ e $\lambda_\infty = 11,433\text{mS/cm}$)	71
Figura 53. Distribuição tempo de residência experimental e ajustada para a vazão de 800 mL/min ($\lambda_0 = 0,119\text{mS/cm}$ e $\lambda_\infty = 11,433\text{mS/cm}$)	72
Figura 54. Distribuição tempo de residência experimental e ajustada para a vazão de 1000 mL/min ($\lambda_0 = 0,513\text{mS/cm}$ e $\lambda_\infty = 11,433\text{mS/cm}$)	72
Figura 55: Determinação do tempo médio de residência por meio da função de distribuição de tempo de residência.	73
Figura 56: Distribuição do tempo de residência na bateria de CSTR experimental e teórico com vazão total de 400mL/min	76
Figura 57: Distribuição do tempo de residência na bateria de CSTR experimental e teórico com vazão total de 600mL/min.	76
Figura 58: Distribuição do tempo de residência na bateria de CSTR experimental e teórico com vazão total de 800mL/min.	77
Figura 59: Distribuição do tempo de residência na bateria de CSTR experimental e teórico com vazão total de 1000mL/min.	77
Figura 60: Distribuição do tempo de residência no PFR experimental e teórico com vazão total de 400mL/min	78
Figura 61: Distribuição do tempo de residência no PFR experimental e teórico com vazão total de 600mL/min.	79

Figura 62: Distribuição do tempo de residência no PFR experimental e teórico com vazão total de 800mL/min.	79
Figura 63: Distribuição do tempo de residência no PFR experimental e teórico com vazão total de 1000mL/min.	80
Figura 64: Unidade experimental do Reator batelada.	83
Figura 65. <i>Script</i> LabVIEW para a obtenção dos parâmetros cinéticos. (a). Painel frontal. (b). Diagrama de blocos.	84
Figura 66. Cálculo da constante da taxa de reação à tempertura de 20°C.....	85
Figura 67. Cálculo da constante da taxa de reação na tempertura de 25°C.....	86
Figura 68. Cálculo da constante da taxa de reação na temperatura de 30°C.....	86
Figura 69. Cálculo da constante da taxa de reação na temperatura de 35°C.....	86
Figura 70. Cálculo da constante da taxa de reação na temperatura de 40°C.....	87
Figura 71. Calculo da constante da taxa de reação na temperatura de 45°C.....	87
Figura 72. Ajuste da Lei de Arrhenius.	88

LISTA DE TABELAS

Tabela 1. Características de Placas Arduino e Compatíveis.	5
Tabela 2. Dados resumidos adquiridos da balança na calibração do sensor 1 da unidade PFR.	43
Tabela 3. Vazão (mL/min) <i>versus</i> frequência (Hz) do sensor 1 da unidade PFR.	44
Tabela 4. Vazão (mL/min) <i>versus</i> frequência (Hz) do sensor 2 da unidade PFR.	45
Tabela 5. Vazão (mL/min) <i>versus</i> frequência (Hz) do sensor 1 da unidade 3 CSTR em série.	46
Tabela 6. Vazão (mL/min) <i>versus</i> frequência (Hz) do sensor 2 da unidade 3 CSTR em série.	48
Tabela 7. Análise estatística das equações de regressão da calibração dos sensores (parâmetros)	49
Tabela 8. Análise estatística das equações de regressão da calibração dos sensores. (equações e nível de confiança).....	49
Tabela 9. Volume real da unidade experimental CSTR.	52
Tabela 10. Custo estimado a Unidade experimental da bateria de CSTR.	55
Tabela 11. Volume da unidade experimental PFR e reatores CSTR.	56
Tabela 12. Custo estimado a Unidade experimental do PFR.	58
Tabela 13. Dados de DTR experimental e ajustada para vazão de 400 mL/min.....	65
Tabela 14. Dados de DTR experimental e ajustada para vazão de 600 mL/min.....	66
Tabela 15. Dados de DTR experimental e ajustada para vazão de 800 mL/min.....	67
Tabela 16. Dados de DTR experimental e ajustada para vazão de 1000 mL/min.....	68
Tabela 17. Valores de tempo médio de residência experimental e teórico para cada vazão para a bateria de reatores CSTR.	70
Tabela 18. Valores de tempo médio de residência experimental e teórico para cada vazão para o reator PFR.....	74
Tabela 19. Valores estimados da constante em função da temperatura.	87
Tabela 20. Cálculo da Energia de ativação e do Fator pré-exponencial.....	88

RESUMO

O presente trabalho visou desenvolver *kits* experimentais para laboratórios de engenharia química para operação automática de reatores de mistura, tubular e batelada, utilizando *hardware* e *softwares* livres e de baixo custo. O aplicativo *LabVIEW Student Software Suite* (gratuito por seis meses) foi utilizado para desenvolver as interfaces gráficas dos *kits* e a plataforma de prototipagem eletrônica Arduino (*open source*) foi utilizada para desenvolver o *hardware* das montagens experimentais e realizar a aquisição/controlado dos dados experimentais. Além de abordar as questões teóricas e conceituais do Arduino, o presente trabalho apresenta descritivos das montagens com foco na operação automática das unidades experimentais. Os *kits* serão utilizados nos cursos de graduação e pós-graduação da Faculdade de engenharia Química da Universidade Federal de Uberlândia para ensino e pesquisa em temas como cinética, reatores, modelagem matemática de sistemas de engenharia, dinâmica de processos, instrumentação, controle e automação de processos e outros.

Palavras-chave: Arduino, Kits Experimentais, Reatores Químicos, Interface Arduino/LabVIEW, Aulas Práticas de Engenharia.

ABSTRACT

This work aimed to develop experimental kits for chemical engineering laboratories in order to operate stirred tank, tubular and batch reactors automatically by using low cost and free hardware and software. LabVIEW Student Software Suite application (free for six months) was used to develop graphic interfaces of the kits and the electronic prototyping platform Arduino (open source) was used to develop the hardware of the experimental setups and perform acquisition/control of experimental data. In addition to addressing the theoretical and conceptual issues of Arduino, the present work presents descriptions of the setups with a focus on the automatic operation of the experimental units. The kits will be used in undergraduate and graduate courses at Faculty of Chemical Engineering at the Federal University of Uberlândia for teaching and research on topics such as kinetics, reactors, mathematical modeling of engineering systems, process dynamics, instrumentation, control and automation of processes and others.

Keywords: Arduino, Experimental Kits, Chemical Reactors, Arduino/LabVIEW Interface, Practical Engineering Lessons.

CAPÍTULO 1

1 INTRODUÇÃO

O curso de graduação de Engenharia Química na Universidade Federal de Uberlândia, assim como a maioria dos cursos de engenharia no mundo, possui várias disciplinas práticas no decorrer dos anos de formação do aluno. O principal objetivo dessas aulas práticas é facilitar o entendimento do aluno, de modo que a assimilação do conhecimento passado em aulas teóricas se tornar mais efetivos durante uma atividade experimental.

As disciplinas práticas presentes nesse curso na UFU seguem um cronograma de experimentos que encaixam no cronograma das matérias teóricas estudadas pelos alunos. Não é novidade que em universidades públicas brasileiras existem limitações financeiras para aparelhamento de seus laboratórios de engenharia. Em muitos casos, as aulas práticas que fazem uso de reagentes e equipamentos específicos podem se tornar uma barreira intransponível para realizar as aulas práticas.

Pensando nisso, esse trabalho visa desenvolver *kits* experimentais de custo acessível para o estudo de sistemas de reações com reatores de mistura, tubular e batelada, para uso em aulas práticas do curso de graduação em Engenharia Química.

O objetivo do trabalho é construir uma plataforma experimental de baixo custo, direta e surpreendentemente poderosa para automação de práticas de laboratório, determinar experimentalmente a distribuição tempo de residência (DTR) em reator tubular e em reatores de mistura em série, além de determinar cinética de reação em reator batelada. Pode-se declarar especificamente que os objetivos são:

- Construir uma plataforma experimental de baixo custo, direta e surpreendentemente poderosa para automação unidades experimentais;
- Determinar experimentalmente a distribuição do tempo de residência (DTR) em reatores de mistura (CSTR) dispostos em série;
- Determinar experimentalmente a distribuição do tempo de residência (DTR) em um reator tubular (PFR);
- Determinar os parâmetros cinéticos de uma reação homogênea, ou seja, a constante específica da taxa de reação, a ordem da reação e a energia de ativação da reação, utilizando um reator batelada;

Os *kits* foram baseados na plataforma de prototipagem eletrônica *open source* Arduino para desenvolver o *hardware* das montagens experimentais e fazer a aquisição e controle de dados. As interfaces gráficas foram desenvolvidas usando *LabVIEW Student Software Suite*, que é uma *suite* de *softwares* voltada para aplicações acadêmicas e de engenharia. Essa *suite* fornece o *LabVIEW Student Edition*, bem como seis *add-ons* de *software* que abrangem recursos de modulação, medição, inspeção e desenvolvimento. A *suite* está disponível gratuitamente no site do desenvolvedor e pode ser registrada gratuitamente por um período de seis meses para avaliação. Além de abordar as questões teóricas e conceituais do Arduino, o presente trabalho apresenta descritivos das montagens com foco no desenvolvimento de práticas de engenharia química.

O capítulo 2 traz uma descrição detalhada da plataforma Arduino, suas principais placas no mercado, as funções de seus pinos e do funcionamento da IDE Arduino, que é a interface de *software* que fornece um ambiente completo para todos os projetos baseados em Arduino.

O capítulo 3 trata da integração entre os ambientes da IDE Arduino e do LabVIEW, destacando vantagens e outros aspectos práticos dessa associação.

O capítulo 4 trata da descrição detalhada dos sensores e atuadores usados neste trabalho, bem suas configurações no ambiente IDE Arduino. Neste trabalho o interesse se voltou para medição, manipulação e controle de vazões, de condutividade elétrica de soluções e de pesagem de massa com células de carga.

O capítulo 5 trata da calibração dos sensores utilizados na montagem dos *kits* e na validação estatística das curvas de calibração.

O capítulo 6 trata do projeto, dimensionamento e montagem dos *kits* experimentais para o estudo de DTR em reator de mistura (CSTR – Continuous Stirred Tank Reactor) e reator tubular (PFR - Plug Flow Reactor).

O capítulo 7 trata das corridas experimentais para determinação das DTR do reator tubular e para uma bateria de três reatores de mistura operando em série.

O capítulo 8 trata do estudo cinético de reação entre acetato de etila e hidróxido de sódio em reator batelada.

O capítulo 9 apresenta as conclusões do trabalho.

CAPÍTULO 2

2 A PLATAFORMA ARDUINO

2.1 Introdução ao Arduino

A plataforma Arduino é uma plataforma de prototipagem eletrônica *open source* de placa única composta de *hardware* e *software* específicos. O *hardware* é composto por diversas placas de circuito impresso com um microcontrolador que executam tarefas programáveis pelo *software* de interface ARDUINO IDE que utiliza essencialmente a linguagem C/C++ para se comunicar com as placas. Este *software* é um ambiente de desenvolvimento integrado (IDE) do Arduino de plataforma cruzada (para Windows, macOS, Linux) escrito na linguagem de programação Java. A interface ARDUINO IDE é usada para escrever e fazer *upload* de programas escritos em linguagem C/C++ para placas Arduino e compatíveis. Ela pode ser baixada gratuitamente de <https://www.arduino.cc/en/Main/Software>.

O Arduino é, então, um tipo básico de computador programável que permite enviar sinais de entradas e receber sinais de saídas para um dispositivo externo ou vice-versa. O Arduino é um pouco mais avançado do que algumas outras microplacas disponíveis no mercado, mas pode se tornar poderosa nas mãos de um programador experiente.

A plataforma foi lançada na Itália em 2005 pelos seus criadores Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. Ela tem uma comunidade forte e crescente de usuários. Nessa comunidade há muitas pessoas oferecendo projetos, ferramentas e recursos que os usuários podem usar gratuitamente para construir seus projetos e protótipos. O Arduino é de código aberto, o que significa que o usuário pode usar a tecnologia para fazer livremente o que quiser. O usuário pode usar o *hardware* que está nos dispositivos, como os chips, e criar sua própria empresa para projetar sua própria versão da placa, que pode ser usada em projetos específicos e até criar algo para revender.

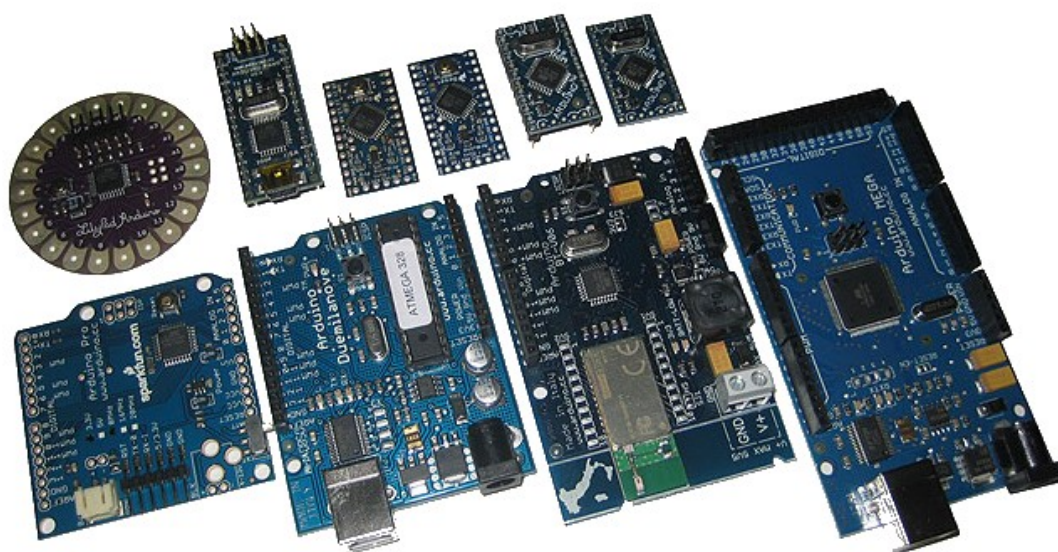
É importante mencionar que existem muitas versões diferentes das placas no mercado. A própria empresa Arduino lançou várias versões de suas próprias placas e, por ser de código aberto, várias outras empresas também lançaram modelos compatíveis com as placas Arduino. No entanto, essa profusão de placas as vezes pode fazer com que ocorram alguns problemas de compatibilidade com o código e com a tecnologia.

Outro aspecto muito relevante da tecnologia é o que diz respeito aos custos das placas. Eles são incrivelmente baixos. É possível comprar placas Arduino e compatíveis entre US\$5 a US\$40 e *kits* completos entre US\$50 a US\$300.

2.2 Placas Arduino

Existem diversos modelos diferentes de placas Arduino que diferem entre si basicamente pelos números de canais de entrada/saída, fator de forma e microprocessador. Elas são a UNO, Mega 2560, Leonardo, Due, Romeo, Julieta, Ethernet, Fio, Micro, Mini, Pro Micro, Nano, LilyPad, Esplora, Red Board, Duemilanove e Diecimila, entre outras. A Figura 1 mostra algumas dessas placas e a Tabela 1 resume algumas características delas. As placas mais populares são as placas Arduino UNO e Arduino Mega 2560 e foram as placas usadas neste trabalho. A seguir apresenta-se uma descrição mais detalhada dessas placas.

Figura 1. Foto de alguns modelos de placas Arduino.



Fonte: <https://aprendafazer.net/saiba-mais-sobre-a-familia-arduino/>

2.2.1 Arduino UNO

A placa Arduino Uno é o modelo mais popular das placas Arduino, resultado de suas características de entrada/saída, simplicidade de seu projeto e principalmente seu custo muito baixo, aproximadamente Us\$22 a original e Us\$2.75 as compatíveis. Esta placa é baseada no microcontrolador ATmega328p. Possui 14 pinos de entrada/saída digitais (em alguns casos,

quando necessários, os pinos analógicos também podem ser usados como pinos digitais). Dos 14 pinos digitais 6 podem ser usados como saídas PWM (*pulse width modulation*) de 8 bits. Ela possui 6 entradas analógicas de 10 bits, um oscilador de cristal de 16 MHz, uma tomada de energia, um botão de *reset*, uma conexão USB e 6 pinos ICSP, que são usados para programar diretamente os microcontroladores da placa usando o protocolo serial SPI, que além dos pinos de tensão e controle (VCC, GND e RESET) estão presentes os 3 pinos adicionais necessários para uma conexão ponto-a-ponto usando o protocolo SPI (MOSI, MISO e SCK). Através destes pinos o mestre envia dados ao periférico, o mestre recebe dados do periférico e o mestre controla o pulso/clock da conexão. A Arduino UNO tem memória flash de 32KB (0,5KB usado no *bootloader*), memória SRAM de 2KB, memória EEPROM de 1KB e velocidade de clock de 16MHz.

Ela vem pronta para ser conectada diretamente a um computador com um cabo USB e para começar a programá-la. Esta placa pode ser alimentada por uma bateria externa, por um adaptador AC/DC ou ainda pela própria conexão USB (ver Figura 2). Essa placa é diferente de todas as demais placas Arduino, pois não usa o *chip* de driver USB para serial FTDI. A tensão recomendada fornecida ao Arduino não deve exceder mais de 12V e deve estar entre 6V e 12V.

Tabela 1. Características de Placas Arduino e Compatíveis.

Modelo	Vin	Vpin	Clock	Digitais	Analógicas	PWM	UART	Flash	Bootloader	Interface de Programação
Arduino Uno R3	7-12V	5V	16MHz	14	6	6	1	32Kb	Optiboot	USB via ATmega16U2
BlackBoard UNO R3	7-12V	5V	16MHz	14	6	6	1	32Kb	Optiboot	USB via FTDI / FTDI
Arduino Leonardo	7-12V	5V	16MHz	20 ⁽¹⁾	12	7	1	32Kb	Leonardo	USB nativo
Romeo V2.0	7-12V	5V	16MHz	20 ⁽¹⁾	12	7	1	32Kb	Leonardo	USB nativo
Julieta	7-12V	5V	16MHz	14	8 ⁽²⁾	6	1	32Kb	Optiboot	USB via FTDI / FTDI
Arduino Mega 2560 R3	7-12V	5V	16MHz	54	16	15	4	256Kb	STK500v2	USB via ATmega16U2
BlackBoard Mega 2560 R3	7-12V	5V	16MHz	54	16	15	4	256Kb	STK500v2	USB via FTDI / FTDI
Arduino Mega ADK	7-12V	5V	16MHz	54	16	15	4	256Kb	STK500v2	USB via ATmega16U2
Arduino Due	7-12V	3.3V	84MHz	54	12	12	4	512Kb	Due	USB nativo
Arduino Esplora	5V	5V	16MHz	-	-	-	-	32Kb	Esplora	USB nativo
Arduino Ethernet	7-12V	5V	16MHz	14	6	4	1	32Kb	Optiboot	USB via ATmega16U2
Arduino Fio	3,3-12V	3,3V	8MHz	14	8	6	1	32Kb	AtmegaBOOT	FTDI / Xbee
Arduino Micro	7-12V	5V	16MHz	20	12	7	1	32Kb	Micro	USB nativo
Arduino Pro Micro	5-12V	5V	16MHz	12	4	5	1	32Kb	DiskLoader	USB nativo
Arduino Pro Mini 3,3V	3,3-12V	3,3V	8MHz	14	6	6	1	32Kb	AtmegaBOOT	FTDI
Arduino Pro Mini 5V	5-12V	5V	16MHz	14	8	6	1	32Kb	AtmegaBOOT	FTDI
BlackBoard Pro Mini 5V	5-12V	5V	16MHz	14	8	6	1	32Kb	AtmegaBOOT	FTDI

⁽¹⁾ A quantidade de portas no Arduino Leonardo é a mesma que a do Arduino UNO ou da BlackBoard, mas alguns pinos possuem dupla funcionalidade fazendo que a quantidade de pinos seja um pouco maior.

⁽²⁾ A Julieta possui 2 entradas analógicas extras no barramento de sensores (A6 e A7)

2.2.1.1 Pinos Analógicos, Digitais, PWM e AREF

São os pinos responsáveis por conectar a placa aos sensores externos usando fios. Eles se dividem em:

Pinos digitais: Existem 14 pinos digitais em um Arduino Uno rotulados de 0 a 13 (ver Figura 2). Esses pinos podem ser usados para ligar/desligar um circuito digital e, adicionalmente, para receber entrada digital de dispositivos externos, ou seja, esses pinos podem ser configurados e usados como entrada digital ou como saída digital. Esses pinos podem produzir dois estados HIGH e LOW, ou seja, o estado HIGH produz 5V e o estado LOW produz 0V nos pinos. Cada um desses pinos digitais pode fornecer ou receber no máximo 40 mA e possui um resistor de *pull-up* interno (por padrão, eles são desconectados dos pinos internamente) de 20-50 k Ω .

Pinos PWM: São pinos de saída digital que podem ser configurados como pinos de sinais PWM (modulação por largura de pulso). São os pinos 3, 5, 6, 9, 10 e 11 que são marcados com o símbolo “~” (ver Figura 2). Esses pinos geram um sinal analógico no formato de onda quadrada de 0 a 5V com frequência modulada.

Pino AREF: Pino de referência analógica (ver Figura 2). Esse pino quando usado serve para definir uma referência de tensão externa entre 0V e 5V como o limite superior para os pinos de entrada analógicos.

Pinos Serial RX e Serial TX: São os pinos responsáveis pela comunicação serial entre o Arduino e um computador para transferir programas da IDE do Arduino para a placa Arduino. Usado para transmitir (pino 1) e receber (pino 0) dados seriais (ver Figura 2).

Pinos analógicos: Esses pinos recebem sinais analógicos entre 0 a 5V e são rotulados como A0 a A5 na placa Arduino Uno (ver Figura 2). Esses pinos recebem um sinal analógico dos sensores e o convertem em um valor digital que pode ser lido por computadores via comunicação USB.

2.2.1.2 Pinos Fonte de Energia Elétrica (5V, 3.3V, GND)

A placa UNO pode ser usada para fornecer energia elétrica para sensores nos níveis 3.3 e 5 V. Existem 3 pinos de aterramento, denominados como GND na placa UNO (ver Figura 2) e qualquer um dos pinos pode ser usado para aterrar seu circuito. O pino 5V (ver Figura 2) fornece 5V de energia que pode ser usado para alimentar um circuito externo. O pino 3.3V (ver Figura 2) fornece 3.3V de energia e, também, pode ser usado para alimentar um circuito externo.

2.2.1.3 Botão de Reset

O Arduino possui um botão de reinicialização (ver Figura 2) que desliga a placa e a liga novamente. Quando pressionado, o programa armazenado na memória EEPROM do Arduino é reiniciado e o LED no pino 13 pisca por alguns instantes antes de executar novamente o programa armazenado na EEPROM. Se esse LED não piscar quando se pressiona o botão de *reset*, então há uma falha grave na placa que requer diagnóstico adicional.

2.2.1.4 LED Indicador de Energia

O LED indicador de energia está localizado próximo aos pinos do ICSP (ver Figura 2). Esse LED deve acender quando se conecta a placa Arduino a uma fonte de energia. Se essa luz não acender, há algo errado com placa Arduino.

2.2.1.5 LED TX e RX

TX é a abreviatura para transmissor e RX para o receptor. Esses pinos são responsáveis pela comunicação serial. Esses LEDs (ver Figura 2) fornecem indicações visuais sempre que o Arduino estiver recebendo ou transmitindo dados via conexão USB.

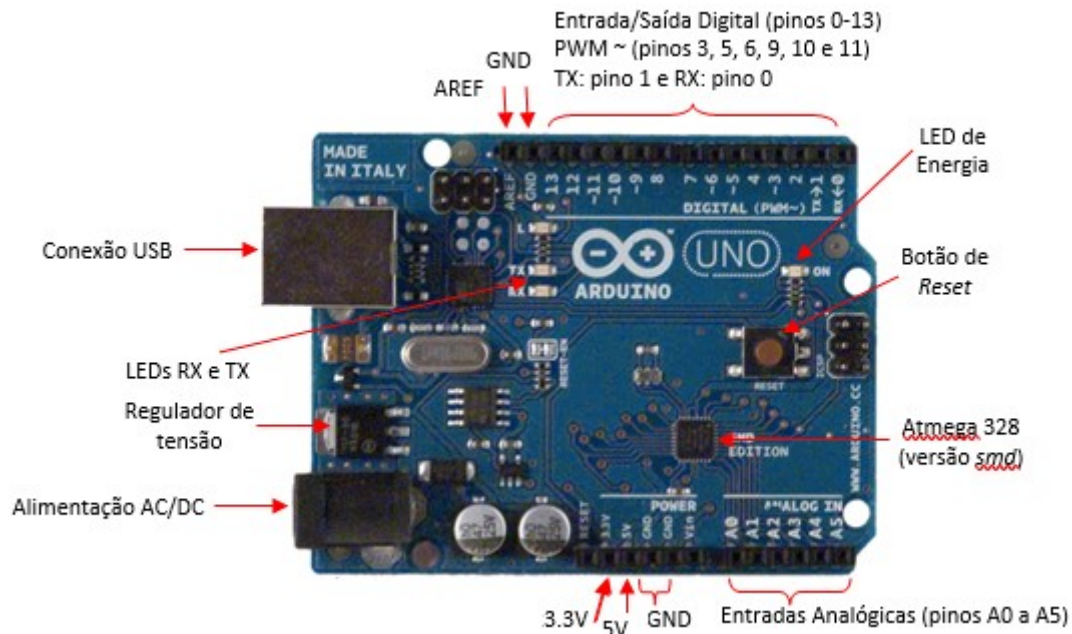
2.2.1.6 Microprocessador ATmega328p

Cada placa Arduino possui seu circuito integrado ou o microcontrolador que é o componente principal, o cérebro do Arduino. O circuito integrado do Arduino pode diferir de placa para placa. Os microcontroladores são geralmente da empresa ATMEL. É necessário saber qual microcontrolador a placa possui antes de carregar um programa a partir do IDE do Arduino. Esta informação está disponível na parte superior do microcontrolador. O microcontrolador da Arduino UNO é o Atmega 328 (ver Figura 2).

2.2.1.7 Regulador de Tensão

O regulador de tensão é o dispositivo que controla a quantidade de tensão permitida na placa Arduino. Os reguladores de tensão estão atuando como um *gate keeper*. Dessa forma, o regulador elimina tensão extra que pode danificar o circuito. Esses reguladores de tensão têm seu limite e ele não pode controlar a tensão além de sua capacidade. Portanto, não se pode conectar o Arduino a fontes maiores do que 20 V.

Figura 2. Arduino UNO e seus componentes.



Fonte: Autor

2.2.2 Arduino Mega 2560

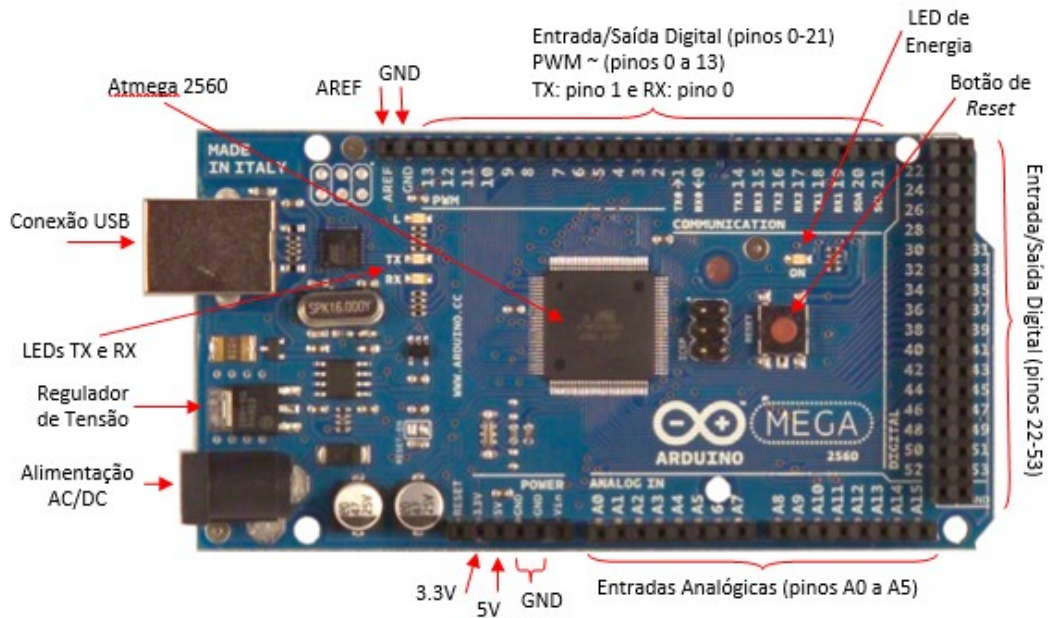
A Arduino Mega 2560 é uma placa bem mais poderosa que a UNO. Ela tem memória flash de 256KB (8KB usado no *bootloader*), memória SRAM de 8KB, memória EEPROM de 4KB e velocidade de clock de 16MHz e é baseada no microcontrolador ATmega2560. Possui 54 pinos de entrada/saída digitais dos quais 14 podem ser usados como saídas PWM, 16 entradas analógicas, 4 UARTs (portas seriais de hardware), um oscilador de cristal de 16 MHz, uma conexão USB, um conector de energia, uma conexão ICSP e um botão de *reset*. Todos esses itens podem ser vistos na Figura 3. Basta conectá-la a um computador com um cabo USB ou alimentá-la com um adaptador AC/DC ou bateria para começar a usá-la. A Mega 2560 é compatível com a maioria dos *shields* projetados para as Arduino Duemilanove ou Diecimila.

2.3 A IDE Arduino

A IDE Arduino é a interface de *software* que fornece um ambiente completo para todos os projetos baseados em Arduino. A IDE é minimalista, mas fornece toda as bibliotecas e funções necessárias para criar qualquer projeto mais elaborado baseado em Arduino. A barra de menus superior possui opções padrão como Arquivo, Editar, Ferramentas, Esboço, Ajuda. A Figura 4 revela a janela criada ao se abrir a IDE. Nela pode-se destacar alguns detalhes importantes. A seção do meio do Arduino IDE é como um editor de texto, onde toda a codificação de programação é feita. A seção inferior do IDE é denominada console de saída e

é usada para ver o *status* da compilação, quaisquer erros no programa, quanta memória foi usada e onde várias mensagens úteis são exibidas.

Figura 3. Arduino Mega 2560 e seus componentes.



Fonte: Autor

Figura 4. Janela inicial da IDE Arduino





Fonte: Autor

Os códigos feitos usando a IDE do Arduino são chamados de *sketchs* e geralmente são escritos em uma versão simplificada do C++ (vários recursos do C++ não estão incluídos na IDE Arduino). Como programar um microcontrolador é diferente de programar um computador, existem algumas bibliotecas incluídas no *hardware* do Arduino, como ler e escrever valores em pinos digitais, ler valores em pinos analógicos e outras funções específicas do *hardware*. Isso geralmente confunde muitas pessoas e elas acham que o Arduino está programado em uma "linguagem Arduino". No entanto, o Arduino é programado em uma versão simplificada do C++ adicionada de algumas bibliotecas específicas do *hardware* do Arduino. Os programas precisam de duas funções, a *void setup()* e a *loop void ()*. A *void setup()* é executada primeiro e uma única vez. A *loop void()* é executada repetidamente até que a placa seja desligada ou um novo *sketch* seja carregado. Algumas funções são específicas do C do Arduino, tais como:




- *pinMode(pin, mode)*: Designa o pino especificado como entrada ou saída.
- *digitalWrite(pin, valor)*: Envia um nível de tensão para o pino designado 0 (LOW) ou 5V (HIGH).
- *digitalRead(pin)*: Lê o nível de tensão atual do pino designado 0 (LOW) ou 5V (HIGH).
- *analogRead(pin)*: Lê o nível de tensão atual do pino designado entre de 0 a 1023 (0 a 5V) – 10 bits (2^{10}).
- *analogWrite(pin, Duty_Cycle)*: Envia sinal PWM para o pino designado com ciclo de trabalho de 0 a 255 (0 a 100%) – 8 bits (2^8).
- *serial.write, serial.read, serial.print, serial.println*: Comandos seriais.

A maioria dos projetos no Arduino escritos usando a IDE usa os 6 botões encontrados na barra de menus que podem ser vistos na Figura 4. Eles são descritos a seguir:

1. : Ícone de marca de escala que é usado para compilar o *sketch* do Arduino. Ao terminar de escrever o *sketch* o usuário pode clicar nele uma vez para verificar o código escrito.

2. : Ícone de seta para a direita que usado para transferir (*upload*) o código compilado para a EEPROM do Arduino. Se o *sketch* não tiver sido compilado, primeiro ele compila o código e depois transfere (*upload*) para a EEPROM da placa.

3. : ícone de papel pontilhado que cria um arquivo novo.

4. : Ícone de seta para cima que é usado para abrir um esboço *sketch* do Arduino.
5. : Ícone de seta para baixo que é usado para salvar o *sketch* atual do Arduino.
6. : Ícone de lupa localizado na extrema direita da janela que é usado para abrir o monitor serial. Ferramenta muito útil para enviar e receber dados do Arduino para a tela do computador. Utilizado para fins de depuração do código.

CAPÍTULO 3

3 INTEGRAÇÃO LABVIEW – ARDUINO

3.1 Introdução

O Arduino é uma poderosa plataforma de prototipagem eletrônica usada por milhões de pessoas em todo o mundo para criar projetos de automação. Usando o Arduino é possível conectar facilmente sensores e objetos físicos a um microcontrolador sem ser um especialista em eletrônica.

No entanto, o uso do Arduino ainda exige que o usuário saiba escrever códigos em C/C++ o que não é fácil para todos. É aqui que o LabVIEW entra em cena. O LabVIEW é um *software* usado por muitos profissionais e universidades de todo o mundo, principalmente para automatizar medições sem precisar escrever uma única linha de código. Graças a uma interface chamada LIFA, é relativamente fácil fazer a interação entre o Arduino e o LabVIEW.

Outro aspecto em que o uso do LabVIEW em interação com o Arduino pode levar a muitas vantagens é possibilidade de usar a interface gráfica poderosa do LabVIEW para construir diagramas, gráficos e animações em tempo real dos sinais enviados e recebidos de sensores, motores e dispositivos conectados ao Arduino. Adicionalmente, essa interface também permite usar as bibliotecas matemáticas, estatísticas, processamento de sinais, comunicação de dados, conectividade, controle e simulação, disponíveis no ambiente LabVIEW.

O LabVIEW é um *software* proprietário desenvolvido pela National Instruments Corporation (NI). A NI é líder mundial em equipamentos automatizados de teste e *software* de instrumentação virtual. O LabVIEW é um produto que eles desenvolveram e está sendo usado em muitos laboratórios em todo o mundo. LabVIEW, sigla para *Laboratory Virtual Instrument Engineering Workbench*, é programado com uma linguagem gráfica conhecida como “G”. O LabVIEW pode ser gerenciado pelo VIPM (*Virtual Instrument Package Manager*). O VIPM contém todas as ferramentas e *kits* para aprimorar o produto LabVIEW. O termo VI (*Virtual Instrument*) refere-se ao bloco de construção dos programas que utilizam a linguagem G.

Apesar de ser um *software* proprietário, a NI disponibiliza desde versões antigas até as atuais do LabVIEW para *download* a partir do sítio <https://www.ni.com/pt-br/shop/labview.html>. Estas versões são versões de teste (*trial*) e funcionam por 45 dias. A

versão do LabVIEW utilizada neste trabalho foi versão do estudante 2020 e foi baixada deste sítio:

<https://www.ni.com/pt-br/support/downloads/software-products/download.labview-student-software-suite.html#352831>

Já a ativação da suíte pode ser feita no sítio:

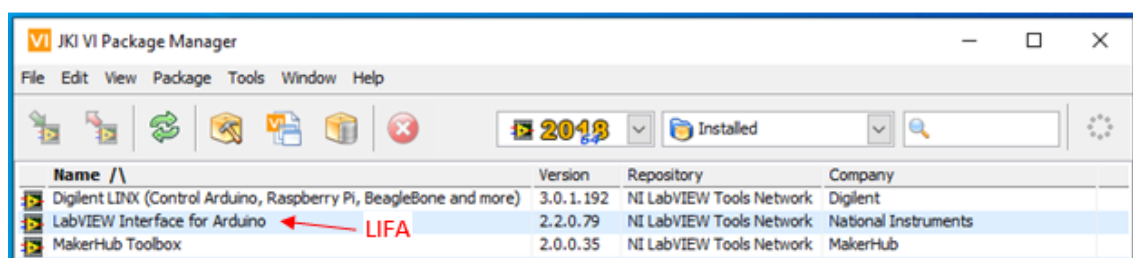
<https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z000000g1ALSAY&l=pt-BR>

O LabVIEW *Student Software Suite* é uma suíte de *softwares* voltada para aplicações acadêmicas e de engenharia. Ela fornece o LabVIEW *Student Edition*, bem como seis *add-ons* de software que abrangem recursos de modulação, medição, inspeção e desenvolvimento.

3.2 Interface LabVIEW – Arduino (LIFA)

A interface LabVIEW/Arduino foi realizada utilizando o *toolkit* gratuito LIFA (*LabVIEW Interface for Arduino*). Este *toolkit* pode ser baixado e instalado no ambiente LabVIEW utilizando o gerenciador de pacotes do LabVIEW (VIPM). A Figura 5 mostra a tela do VIPM com a interface LIFA já instalada.

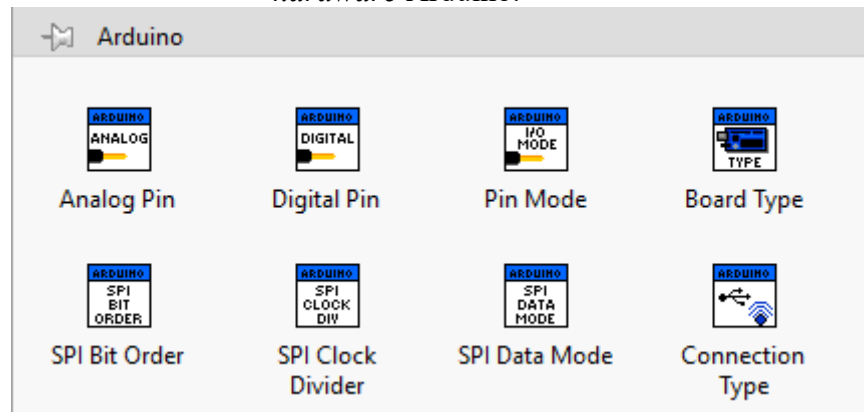
Figura 5. Instalação do *toolkit* LIFA utilizando o gerenciador de pacotes VIPM.



Fonte: Autor

A LIFA permite que os desenvolvedores adquiram dados do microcontrolador Arduino e os processe no ambiente de programação gráfica do LabVIEW. A Figura 6 mostra as bibliotecas de configuração das placas Arduino que foram inseridas no LabVIEW pelo LIFA e que são usadas no painel frontal do *script* do LabVIEW.

Figura 6. Bibliotecas usadas no painel frontal do *script* do LabVIEW para configurar o *hardware* Arduino.

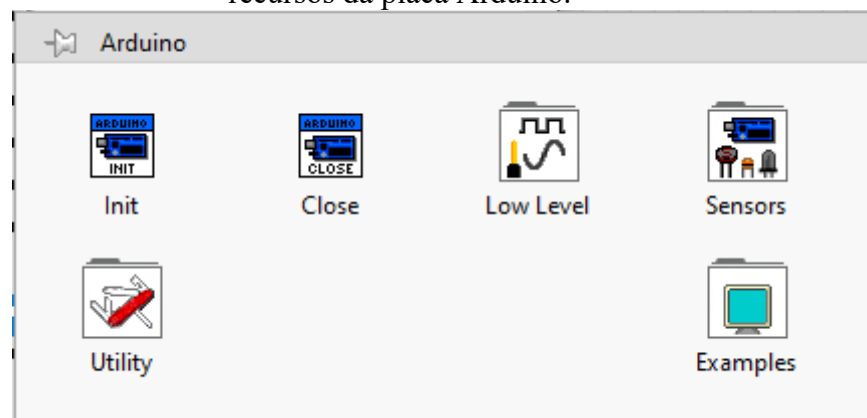


Fonte: Autor

As bibliotecas “*Board Type*” e “*Connection Type*” servem para informar ao LabVIEW qual placa será usada (UNO, Mega etc.) e como ela está conectada ao computador (serial, USB, XBEE etc.), respectivamente. A biblioteca “*Pin Mode*” serve para configurar um determinado pino como entrada ou saída. Já as bibliotecas “*Analog Pin*” e “*Digital Pin*” servem para informar ao LabVIEW em qual pino analógico ou qual pino digital ele deve ler ou escrever valores, respectivamente.

A Figura 7 mostra as bibliotecas de acesso à placa Arduino que foram inseridas no LabVIEW pela LIFA e que são usadas no diagrama de blocos do *script* do LabVIEW.

Figura 7. Bibliotecas usadas no diagrama de blocos do *script* do LabVIEW para acessar os recursos da placa Arduino.

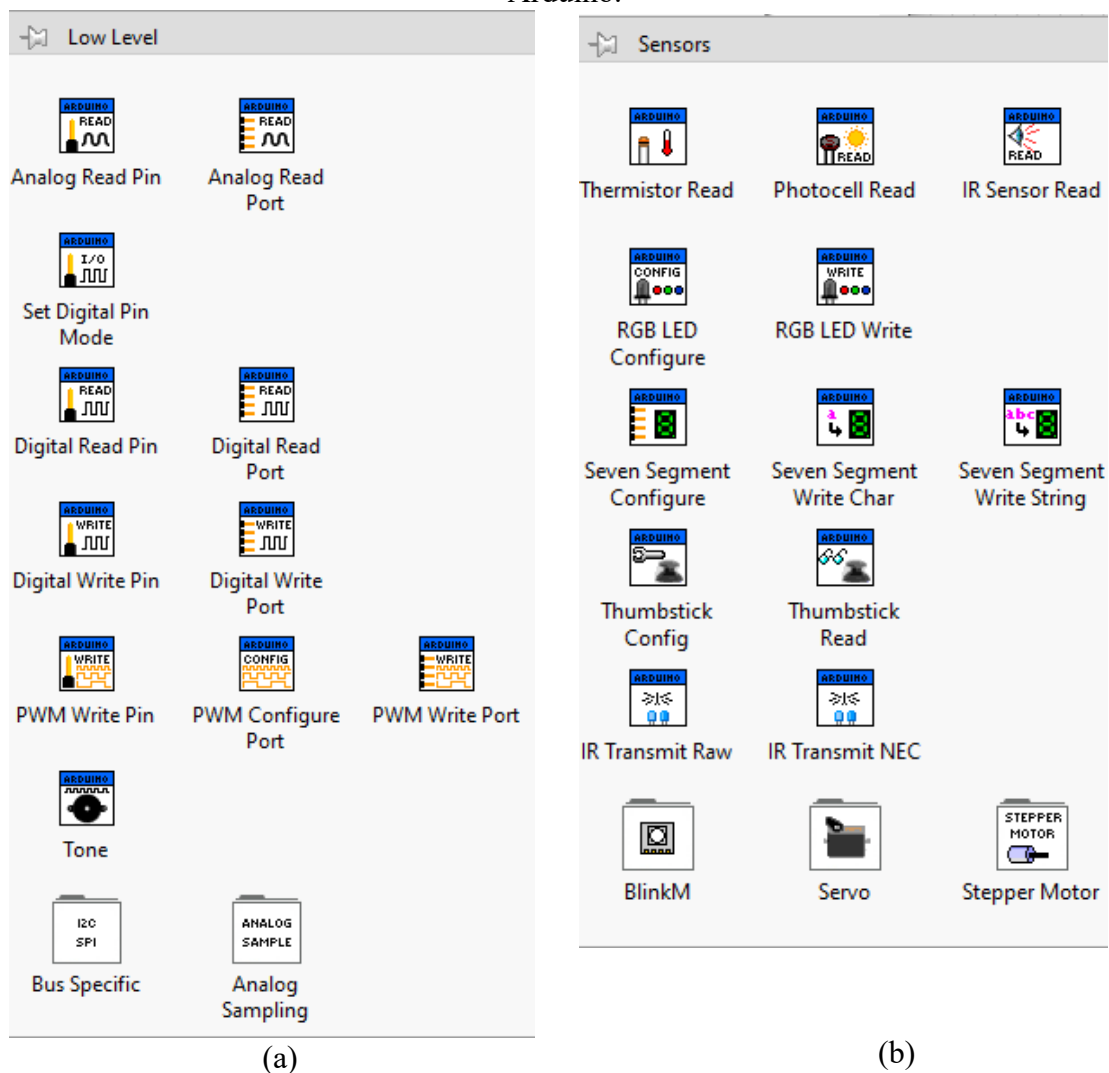


Fonte: Autor

As bibliotecas “*Init*” e “*Close*” servem para abrir e fechar uma determinada porta serial no computador, respectivamente. A placa Arduino deverá estar conectada nesta porta. A biblioteca “*Low Level*” é na verdade um conjunto de bibliotecas que operam no nível da máquina e que realizam as leituras em pinos analógicos ou digitais, enviam valores para os

pinos digitais e para os pinos PWM. Essas rotinas podem operar sobre um determinado pino ou sobre uma sequência de pinos simultaneamente. São essas rotinas que verdadeiramente fazem a conexão/comunicação da placa Arduino com o computador. A Figura 8a mostra essas bibliotecas. Já a Figura 8b mostra as bibliotecas que fazem a comunicação dos sensores com a placa Arduino. Os sensores que já são configurados nas bibliotecas da LIFA são os termistores, fotocélulas, infra-vermelho, LEDs, servo motores e motores de passo.

Figura 8. Bibliotecas usadas no diagrama de blocos do *script* do LabVIEW para: a) acessar valores de leitura e escrita nos pinos do Arduino; b) configurar sensores conectados ao Arduino.



Fonte: Autor

Neste trabalho, usou-se além dos sensores incorporados à LIFA, sensores de efeito *hall* para medir vazões, motores de corrente contínua (DC) para manipular vazões e leituras de portas seriais para adquirir sinais provenientes de condutivímetro com saída serial padrão tipo

RS-232. Infelizmente, eles não fazem parte das bibliotecas do LIFA. Para contornar esta dificuldade, foram desenvolvidas neste trabalho e incorporadas à LIFA novas bibliotecas para contemplar os sensores de efeito *hall* e motores DC. Essas bibliotecas serão discutidas em mais detalhes nos próximos capítulos.

CAPÍTULO 4

4 ATUADORES E SENSORES PARA ARDUINO

4.1 Introdução aos Atuadores, Sensores e *Shields* Utilizados

O que faz a plataforma Arduino ser uma ferramenta poderosa para automação e controle é sua habilidade em lidar com vasta gama de sensores e atuadores igualmente de baixo custo. Sensores são dispositivos eletrônicos que permitem ao microcontrolador Arduino fazer leituras do ambiente, reagindo de acordo com as instruções programadas no *sketch*. Por exemplo, pode-se programar o Arduino para ligar um ventilador quando a temperatura atingir um determinado valor ou desligar uma lâmpada quando o sol nascer. Neste trabalho, o interesse está voltado para medição, manipulação e controle de vazões de entradas em reatores tipo tanques agitados (CSTR) e tubular (PFR), acionamento de agitadores mecânicos dos reatores tanque, acionamento de motores de corrente contínua (12 VDC), medição de temperatura e de condutividade elétrica de soluções para acompanhamento do avanço de reação química.

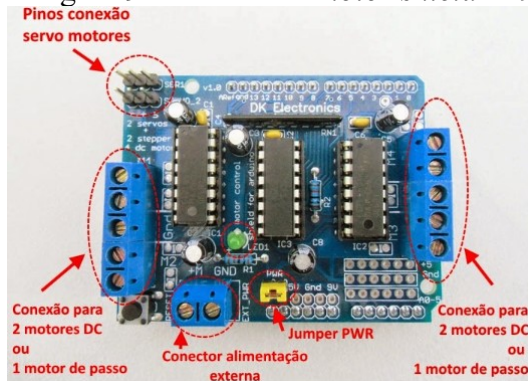
4.2 Manipulação de Vazão Usando a Minibomba RS385 12V DC

4.2.1 O Hardware

Neste trabalho a atenção está voltada para o acionamento e controle de rotação de uma minibomba modelo RS385 de corrente contínua de 12V. O objetivo é manipular automaticamente a rotação do motor 12V DC desta minibomba utilizando o Arduino. Este objetivo é alcançado alterando a tensão de 0 a 12 V nos terminais da minibomba utilizando um *motor shield* para Arduino baseado no chip L293D. *Shields* são placas de circuito impresso que podem ser conectadas ao Arduino, encaixando-se perfeitamente por cima dele e expandindo suas capacidades. A Figura 9a mostra o *motor shield L293D* revelando os pinos de conexão com os motores e a Figura 9b mostra o *shield* montado sobre uma placa Arduino UNO.

O *motor shield L293D* tem 2 chips L293D e um 74HC595, o que permite controlar simultaneamente 4 motores DC ou 2 motores de passo e, ainda, suporta adicionalmente 2 servomotores. Pode ser usado para acionar motores com tensões de 4,5 a 16 VDC. A Figura 10 mostra detalhes da minibomba e o esquema de ligação elétrica dela ao *motor shield L293D*. O custo dessa minibomba é de US\$3.37 e do *motor shield L293D* é de US\$1.90 aproximadamente.

Figura 9. Detalhes do *motor shield L293D*: a). *Pinout*. b). Acoplada na *Arduino UNO*.



(a)



(b)

Fonte: Autor

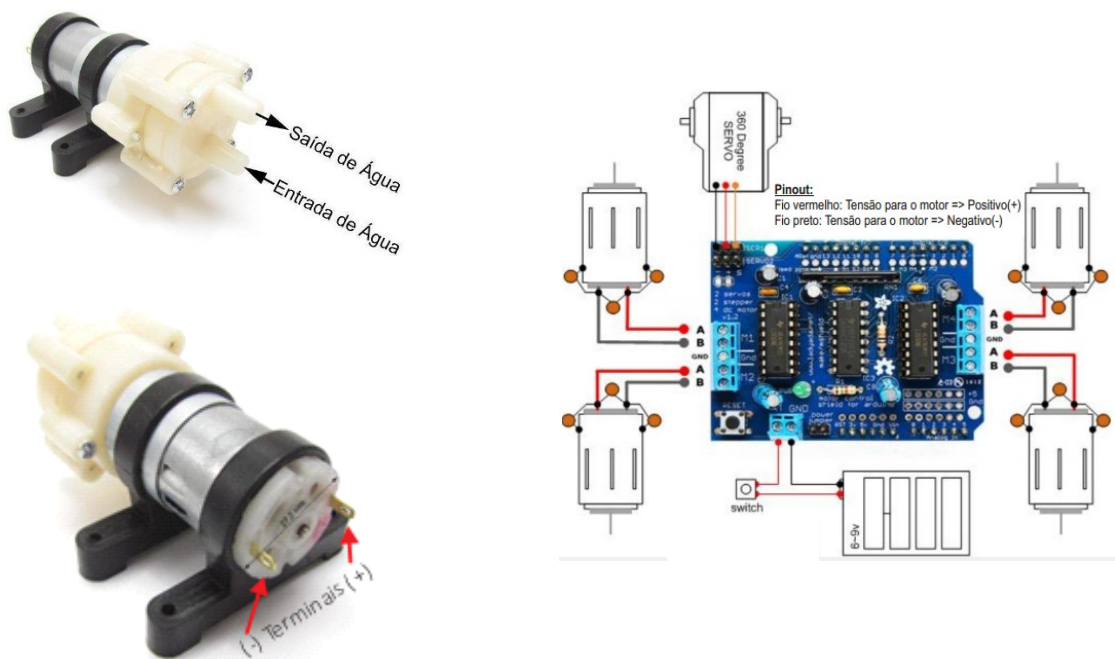
As características da minibomba são:

- Modelo: RS385;
- Material: metal e plástico;
- Vazão: máxima de 2L/min;
- Altura: de sucção máxima de 2 metros e de recalque máxima de 3 metros;
- Tamanho: 90 mm x 40 mm x 35 mm;
- Peso: 106 g;
- Potência: para alimentação de 6 a 12V consumo é de 5W a 10 W;
- Tensão recomendada: 9 a 12VDC;
- Corrente: de trabalho: 0,5-0,7A e carga a vazio: 0,18A
- Diâmetro de saída: diâmetro 6 mm, um diâmetro exterior de 9 mm;
- Vida útil: até 2500 h;
- Temperatura da água: até 80 °C;
- Diâmetro tubulação de entrada e saída: 7,6mm;
- Diâmetro do motor: 28,6mm.

Para conectar um motor basta soldar dois fios nos terminais “+” e “-” da minibomba e depois conectá-los aos terminais M1 ou M2 ou M3 ou M4 do *Shield*, respeitando a polaridade indicada nos terminais “A” e “B” (ver Figura 10). O *motor shield L293D* pode acionar até 4 motores DC bidirecionalmente. Isso significa que eles podem ser movidos no sentido horário e anti-horário. A velocidade também pode ser variada em incrementos de 0,39% (8 bits) usando os pinos PWM do Arduino. A ponte H do *Shield* não suporta cargas

acima de 0,6 A ou picos acima de 1,2 A. Portanto, esse *Shield* é para motores pequenos. Para motores maiores acionados por Arduino pode-se usar o *motor shield VNH5019* que possui dois canais (aciona dois motores simultaneamente), funciona com tensões de 5,5V a 24V e pode fornecer corrente de até 12A em cada canal e 30A de corrente de pico.

Figura 10. Detalhes da minibomba RS385 e esquema de ligação elétrica ao *motor shield L293D*.



Fonte: Autor

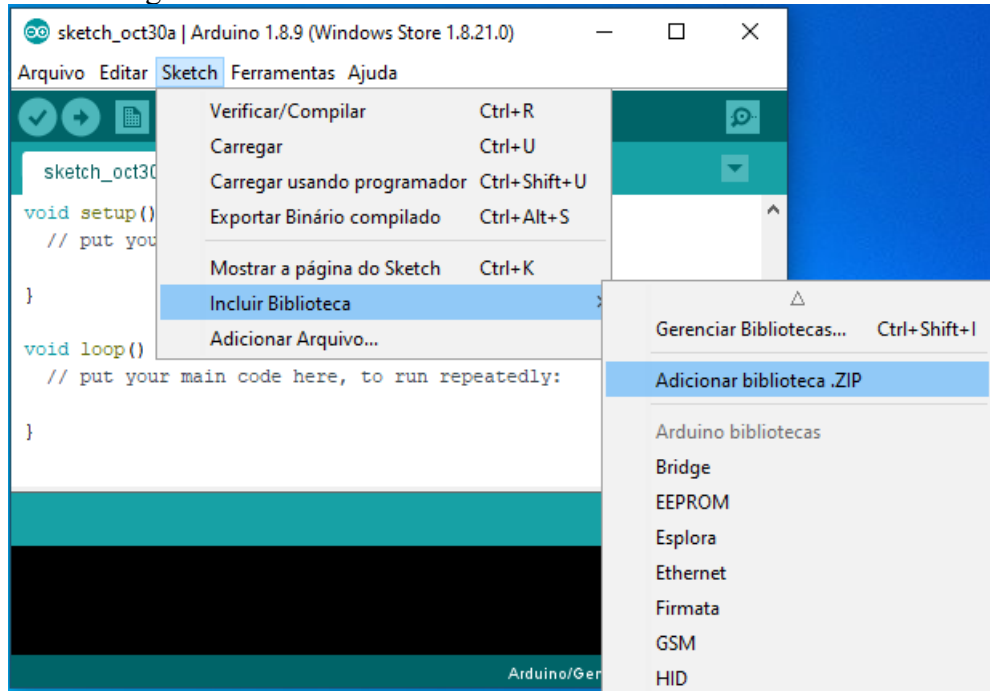
4.2.2 O Software

A *LabVIEW Interface for Arduino* (LIFA) elaborada pela *National Instruments* (NI) não contempla biblioteca para o *motor shield L293D*. Logo, antes de poder usar o *shield* é necessário instalar a biblioteca “*AFMotor.h*” na IDE Arduino, compilar novamente a LIFA e fazer o *upload* dela na placa Arduino. Essa biblioteca instruirá o Arduino a “conversar” com o *Shield*. Esta biblioteca foi desenvolvida pela empresa *Adafruit Industries*¹ e está disponível gratuitamente nos sítios <https://github.com/adafruit/Adafruit-Motor-Shield-library> ou <https://www.arduino-libraries.info/libraries/adafruit-motor-shield-library>.

¹ É uma empresa de hardware de código aberto sediada em Nova York e fundada em 2005. A empresa projeta, fabrica e comercializa uma série de produtos eletrônicos, componentes eletrônicos, ferramentas e acessórios.

Para adicionar uma biblioteca à IDE Arduino abra a IDE, siga as abas *Sketch >> Incluir Biblioteca >> Adicionar biblioteca.ZIP* e aponte para o arquivo “.zip” baixado. A Figura 11 mostra os passos na IDE.

Figura 11. Adicionando bibliotecas na IDE Arduino.



Fonte: Autor

Adicionalmente, a linha a seguir foi incluída no *sketch* da LIFA. Esta instrução carrega a biblioteca "*AFMotor.h*" durante a compilação do *sketch*.

```
#include "AFMotor.h"
```

O comando *AF_DCMotor* desta biblioteca configura o *motor Shield* com os parâmetros *número do motor* (1 a 4) e a *frequência da ponte H do Shield* (1, 2, 8 e 64 KHz). As linhas a seguir foram incluídas no *Sketch* da LIFA e configuram as minibombas 1 a 4 nas portas 1 a 4 do *Shield*:

```
AF_DCMotor DCM1(1, MOTOR12_64KHZ); /* dc motor #1, 64KHz pwm */  
AF_DCMotor DCM2(2, MOTOR12_64KHZ); /* dc motor #2, 64KHz pwm */  
AF_DCMotor DCM3(3, MOTOR12_64KHZ); /* dc motor #3, 64KHz pwm */  
AF_DCMotor DCM4(4, MOTOR12_64KHZ); /* dc motor #4, 64KHz pwm */
```

Para definir o sentido de rotação do motor usou-se o comando *run* com o parâmetro “FORWARD”. As linhas a seguir foram incorporadas ao *Sketch* da LIFA e configuram minibombas para rotação no sentido horário:

```
DCM1.run(FORWARD); //Motor 1 configurado para girar sentido horário
DCM2.run(FORWARD); //Motor 2 configurado para girar sentido horário
DCM3.run(FORWARD); //Motor 3 configurado para girar sentido horário
DCM4.run(FORWARD); //Motor 4 configurado para girar sentido horário
```

As linhas de código a seguir foram, então, adicionadas à LIFA para configurar os sentidos de rotação dos motores 1 a 4:

Figura 12. Linhas de código para adicionadas à LIFA para configurar os sentidos de rotação dos motores 1 a 4.

```

/*****
** Configura os motores DC
*****/
case 0x35: // DC Motor Configure
  if(command[2] == 1){
    if(command[3] == 0){
      DCM1.setSpeed(0); //Velocidade mínima
      DCM1.run(FORWARD); //Motor 1 configurado para girar sentido horário
    }
    else if(command[3] == 1){
      DCM1.setSpeed(0); //Velocidade mínima
      DCM1.run(BACKWARD); //Motor 1 configurado para girar sentido anti-horário
    }
    else {
      DCM1.setSpeed(0); //Velocidade mínima
      DCM1.run(RELEASE); //Motor 1 configurado para parar
    }
  }
  else if(command[2] == 2){
    if(command[3] == 0){
      DCM2.setSpeed(0); //Velocidade mínima
      DCM2.run(FORWARD); //Motor 2 configurado para girar sentido horário
    }
    else if(command[3] == 1){
      DCM2.setSpeed(0); //Velocidade mínima
      DCM2.run(BACKWARD); //Motor 2 configurado para girar sentido anti-horário
    }
    else {
      DCM2.setSpeed(0); //Velocidade mínima
      DCM2.run(RELEASE); //Motor 2 configurado para parar
    }
  }
  else if(command[2] == 3){
    if(command[3] == 0){
      DCM3.setSpeed(0); //Velocidade mínima
      DCM3.run(FORWARD); //Motor 3 configurado para girar sentido horário
    }
  }
}

```

Fonte: Autor

Figura 13 (cont.). Linhas de código para adicionadas à LIFA para configurar os sentidos de rotação dos motores 1 a 4 (continuação).

```
else if(command[3] == 1){
  DCM3.setSpeed(0); //Velocidade mínima
  DCM3.run(BACKWARD);//Motor 3 configurado para girar sentido anti-horário
}
else {
  DCM3.setSpeed(0); //Velocidade mínima
  DCM3.run(RELEASE);//Motor 3 configurado para parar
}
}
else {
  if(command[3] == 0){
    DCM4.setSpeed(0); //Velocidade mínima
    DCM4.run(FORWARD);//Motor 4 configurado para girar sentido horário
  }
  else if(command[3] == 1){
    DCM4.setSpeed(0); //Velocidade mínima
    DCM4.run(BACKWARD);//Motor 4 configurado para girar sentido anti-horário
  }
  else {
    DCM4.setSpeed(0); //Velocidade mínima
    DCM4.run(RELEASE);//Motor 4 configurado para parar
  }
}
}
Serial.write('0');
break;
```

Fonte: Autor

Para acionar o motor com uma velocidade que pode variar de 0 a 100% da rotação máxima do motor, usou-se o comando *setSpeed* com o parâmetro “*Speed*” que é a velocidade de rotação que varia de 0 (parado) a 255 (velocidade máxima). O usuário pode alterar a velocidade sempre que quiser. Por exemplo, os comandos a seguir acionam as minibombas para a velocidade desejada:

```
DCM1.setSpeed(64); //ajusta a velocidade do Motor 1 para 25% da máxima
DCM2.setSpeed(128); //ajusta a velocidade do Motor 2 para 50% da máxima
DCM3.setSpeed(192); //ajusta a velocidade do Motor 3 para 75% da máxima
DCM4.setSpeed(255); //ajusta a velocidade do Motor 4 para 100% da máxima
```

Então, para ajustar as velocidades de rotação dos motores 1 a 4 foram adicionadas ao *sketch* da LIFA as linhas de código mostradas na Figura 14.

As linhas de código mostradas na Figura 15 foram também adicionadas à LIFA para parar os motores 1 a 4.

Figura 14. Linhas de código para ajustar as velocidades de rotação dos motores 1 a 4.

```

/*****
** Set Speed to DC motors
*****/
case 0x36: // Set Speed to DC motors
    DCMotor_Speed = (int)(command[3] << 8) + command[4];
    if(DCMotor_Speed < 0){
        DCMotor_Speed = 0;
    }
    else if(DCMotor_Speed >255){
        DCMotor_Speed = 255;
    }
    if(command[2] == 1){
        DCM1.setSpeed(DCMotor_Speed); //Set speed for DC Motor 1
    }
    else if(command[2] == 2){
        DCM2.setSpeed(DCMotor_Speed); //Set speed for DC Motor 2
    }
    else if(command[2] == 3){
        DCM3.setSpeed(DCMotor_Speed); //Set speed for DC Motor 3
    }
    else {
        DCM4.setSpeed(DCMotor_Speed); //Set speed for DC Motor 4
    }
    Serial.write('0');
    break;

```

Fonte: Autor

Figura 15. Linhas de código adicionadas à LIFA para parar os motores 1 a 4.

```

/*****
** Stop DC motors
*****/
case 0x37: // Release DC Motors
    DCMotor_Speed = (int)command[3];
    if(command[2] == 1){
        DCM1.setSpeed(0); //Velocidade mínima
        DCM1.run(RELEASE); //Motor 1 configurado para parar
    }
    else if(command[2] == 2){
        DCM2.setSpeed(0); //Velocidade mínima
        DCM2.run(RELEASE); //Motor 2 configurado para parar
    }
    else if(command[2] == 3){
        DCM3.setSpeed(0); //Velocidade mínima
        DCM3.run(RELEASE); //Motor 3 configurado para parar
    }
    else {
        DCM4.setSpeed(0); //Velocidade mínima
        DCM4.run(RELEASE); //Motor 4 configurado para parar
    }
    Serial.write('0');
    break;

```

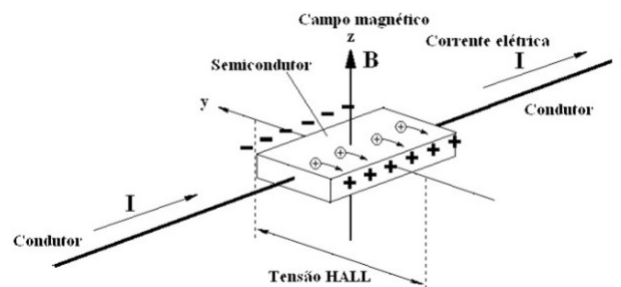
Fonte: Autor

4.3 Medição de Vazão

4.3.1 O Hardware

Neste trabalho usou-se um sensor de vazão modelo YF-S402 desenvolvido para atuar em conjunto com diversas plataformas de prototipagem, sejam elas, Arduino, PIC, AVR, ARM, entre outros. Este sensor é baseado no efeito Hall. Um sensor de efeito Hall é um transdutor em que cargas elétricas atravessando um campo magnético aplicado perpendicularmente ao escoamento das cargas responde com uma variação em sua tensão de saída. O efeito Hall está relacionado ao surgimento de uma diferença de potencial elétrico em um condutor elétrico colocado transversalmente em um campo magnético perpendicular e por ele fazer circular uma corrente elétrica, conforme esquematizado na Figura 16.

Figura 16. Tensão Hall produzida pelo efeito Hall.



Fonte: Autor

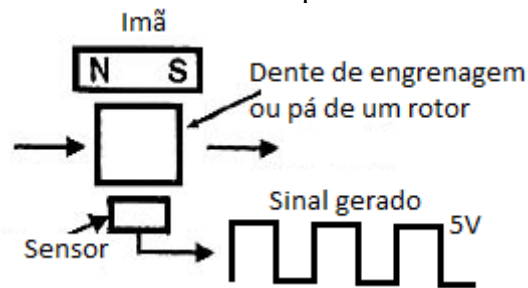
Essa diferença de potencial elétrico é proporcional à intensidade do campo elétrico e pode ser medida por um circuito externo. Sensores de efeito Hall são utilizados em diversos contextos, como medidores de rotação (rodas de bicicleta, dentes de engrenagens, indicador de velocidade para automóveis, sistemas de ignição eletrônica), sensores de vazão de fluidos, sensores de corrente e pressão.

A forma do sinal que é obtido como saída de um sensor de efeito Hall é determinado pelo posicionamento do sensor em relação ao campo magnético. Desse modo, existem diversas configurações relativas ao posicionamento do sensor em relação ao campo que podem ser utilizadas pelo projetista e/ou engenheiro ao planejar o circuito detector e os dispositivos em que o sensor será instalado.

No caso do sensor utilizado (YF-S402) a configuração é de sensor tipo imã fixo com passagem de um material ferromagnético móvel entre o imã e o sensor. A movimentação do material ferromagnético deforma o campo magnético que é detectado por um sensor. No sensor YF-S402 o material ferromagnético é colocado na extremidade das pás de um rotor que é

impulsionado pela passagem de líquido pelo dispositivo. A rotação produzida pela passagem do líquido através do rotor faz com que o material ferromagnético se aproxime e se distancie do ímã em movimentos circulares, provocando alterações no campo magnético. Um detector é capaz, então, de perceber quando o campo é alterado. Um circuito auxiliar gera em decorrência uma onda quadrada de tensão com níveis 0 e 5V conforme esquematizado na Figura 17.

Figura 17. Arranjo com ímã fixo com pá de rotor móvel de um material ferromagnético entre um sensor Hall e um ímã permanente.



Fonte: Autor

A frequência do sinal de onda passa a ser função da frequência de alteração do campo magnético, que é função da velocidade angular do rotor do sensor que, por sua vez, dependerá da velocidade do líquido que atravessa o sensor e impulsiona o rotor. Desta forma, a onda quadrada de 0-5V gerada pelo sensor pode ser conectada diretamente em um pino digital da placa Arduino. Usando a IDE do Arduino pode-se, então, medir a frequência ou período da onda gerada. Finalmente, uma curva de calibração entre frequência ou período com a vazão pode ser construída.

Figura 18. (a). Sensor de efeito Hall modelo YF-S402. (b). Sensor aberto mostrando detalhe do rotor.



(a)



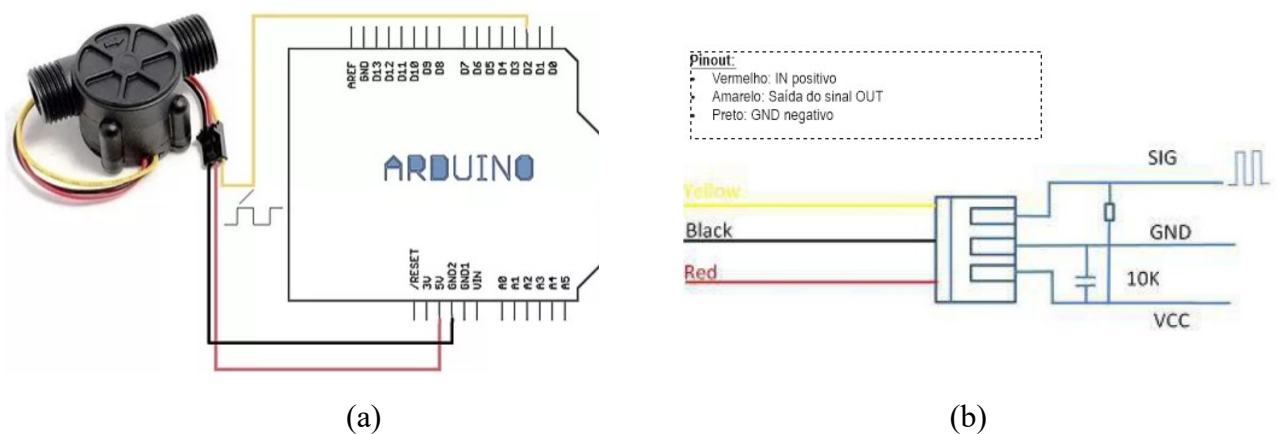
(b)

Fonte: Autor

A Figura 18 mostra detalhes do sensor e a Figura 19 mostra o esquema de ligação elétrica do sensor YF-S402 ao Arduino. O custo desse sensor é de US\$3.47 aproximadamente. As características dele são:

- Tensão de trabalho: DC 5V a 24V
- Capacidade de carga $\leq 10\text{mA}$ (DC 5V)
- Tensão mínima de trabalho: DC 4.5V
- Corrente máxima de trabalho: 15mA (DC 5V)
- Range de vazão: 0,5 ~ 5 L/min
- Temperatura de operação $\leq 80^\circ\text{C}$
- Temperatura do líquido $\leq 120^\circ\text{C}$
- Umidade de operação: 35% a 90% de UR
- Pressão da água: 0,35 MPa
- Temperatura de armazenamento: -25°C a $+80^\circ\text{C}$
- Umidade de Armazenamento: 25% a 95% de UR
- Resistência de isolamento $> 100\text{M}\Omega$
- Bitola do tubo: 1/4" diâmetro externo
- Erro: $\pm 0,1$ L/min
- Líquido de trabalho: água, óleo, cerveja.

Figura 19. (a). Conexão do sensor YF-S402 ao Arduino. (b). Pinagem do sensor.



Fonte: Autor

4.3.2 O Software

A função `pulseIn()` da IDE foi usada para medir a duração de um pulso, isto é, o período da onda quadrada gerada pelo sensor de vazão. Esta função mede a duração de um pulso em um pino digital (que pode ser *HIGH* ou *LOW*). Por exemplo, se o valor *HIGH* é passado para a função, a função `pulseIn()` espera o pino ir do estado *LOW* para *HIGH*, começa a temporizar, então espera o pino ir para o estado *LOW* e para de temporizar. Retorna o tamanho do pulso em microssegundos ou desiste e retorna 0 (zero) se não receber nenhum pulso dentro de um tempo máximo de espera especificado. Funciona com pulsos de 10 microssegundos a 3 minutos de duração. A sintaxe da função é:

Sintaxe:

`pulseIn(pino, valor)` ou `pulseIn(pino, valor, tempo_de_espera)`

Parâmetros:

pino: o número do pino no qual se quer capturar a duração de um pulso. Inteiro de 16 bits.

valor: tipo de pulso a ser lido. Pode ser *HIGH* ou *LOW*.

tempo_de_espera (opcional): tempo de duração em microssegundos do pulso. Inteiro de 32 bits.

Retorna:

A duração do pulso (em microssegundos) ou 0 se nenhum pulso começar antes de se esgotar o tempo de espera. Inteiro de 32 bits.

O exemplo a seguir (Figura 20) imprime a duração de um pulso no pino 7 na tela (*serial monitor*) da IDE:

Figura 20. Linhas de código para imprimir a duração de um pulso.

```
int pino = 7;           //pino para a entrada do pulso
unsigned long duracao; //variável para guardar a duração do pulso

void setup()
{
  Serial.begin(9600);
  pinMode(pino, INPUT);
}

void loop()
{
  duracao = pulseIn(pino, HIGH);
  Serial.println(duracao);
}
```

Fonte: Autor

A *LabVIEW Interface for Arduino* (LIFA) elaborada pela *National Instruments* (NI) também não contempla biblioteca para o sensor YF-S402. Então, antes de usar o sensor foi necessário alterar a LIFA para adquirir os sinais deste sensor. As linhas de programa da Figura 21 foram incluídas no *Sketch* da LIFA:

Figura 21. Linhas de código incluídas no *Sketch* da LIFA para adquirir os sinais do sensor.

```
/**
 * Read flowrate (Hall Effect Flowmeter Sensor - Model Number YF-S402)
 */
case 0x3B: // Measure Flowrate only
  Nsample = 0;
  Soma_Pulsos = 0;
  for(int i=0; i<Nsampling; i++){
    Pulsos = pulseIn(command[2], command[3],100000); //Período em micros;
    if (Pulsos > 3500 && Pulsos < 35000) {
      Soma_Pulsos = Soma_Pulsos + Pulsos;
      Nsample = Nsample + 1;
    }
  }
  if (Nsample != 0){
    Pulsos = Soma_Pulsos/Nsample;
  }
  else{
    Pulsos = 0;
  }
  Serial.write((Pulsos >> 24 & 0xFF));
  Serial.write((Pulsos >> 16 & 0xFF));
  Serial.write((Pulsos >> 8 & 0xFF));
  Serial.write(Pulsos & 0xFF); // Send flow sensor frequency to LabVIEW in 24 bits
  break;

```

Fonte: Autor

4.4 Medição de Condutividade Elétrica

A corrente elétrica pode ser entendida como o movimento ordenado de partículas eletricamente carregadas que circulam por um condutor ou íons que se deslocam por uma solução eletrolítica, quando entre as extremidades desse condutor há uma diferença de potencial, ou seja, tensão. Eletrólitos são soluções eletricamente carregadas em um meio de elevada constante dielétrica como a água. O eletrólito se dissocia em íons cátions e ânions que se movem na solução, simultaneamente e em direções opostas, estabelecendo a corrente elétrica. O íon positivo (cátion) é atraído pelo polo negativo (catodo) e o íon negativo (ânion) é atraído pelo polo positivo (anodo). A dissociação iônica pode ser total ou parcial.

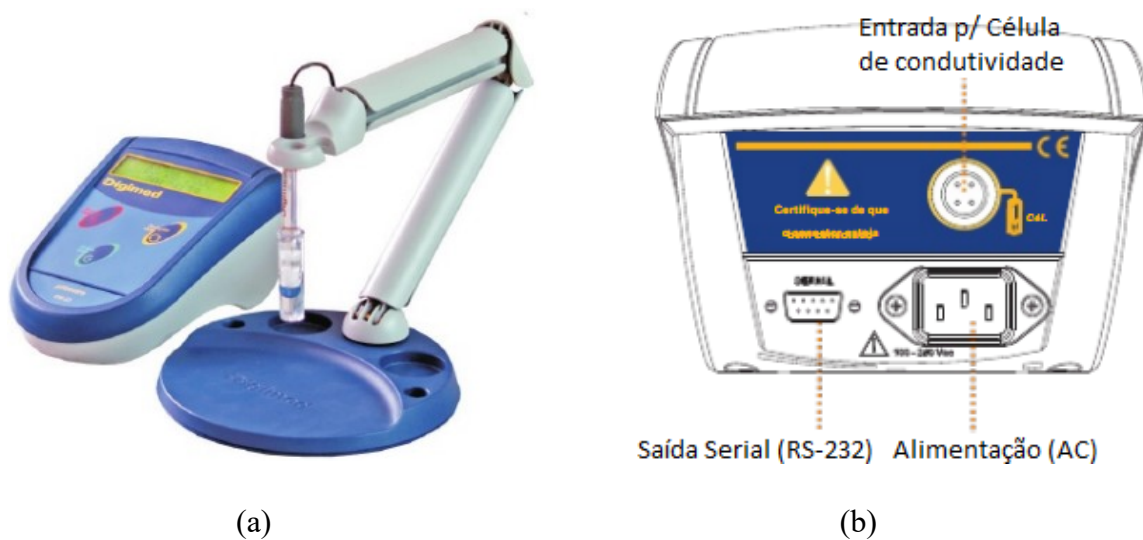
A condutividade elétrica de uma solução é definida como o recíproco da resistividade elétrica da solução. É indicativa da facilidade com que corrente elétrica atravessa a solução. A

unidade é o *Siemens* por metro ($S.m^{-1}$). A condutividade de soluções iônicas tem importância prática na indústria por causa da facilidade de sua medida em linha (*on line*). No presente estudo ela é usada para acompanhar o grau de avanço de uma reação de saponificação entre hidróxido de sódio e acetato de etila.

4.4.1 O Hardware

A condutividade elétrica foi medida em linha usando um condutímetro marca Digimed modelo DM-32 disponível no laboratório da Faculdade de Engenharia Química da Universidade Federal de Uberlândia (FEQUI/UFU). A Figura 22.a mostra o condutímetro usado e a Figura 22.b mostra os conectores de energia, saída serial padrão RS-232 e de entrada da célula de condutividade.

Figura 22. (a). Condutímetro Digimed DM-32. (b). Conectores de entrada e saída.



Fonte: <https://docgo.net/download/documents/manual-dm-32>.

Figura 23. Conversor/Adaptador USB para Serial RS232/DB9.



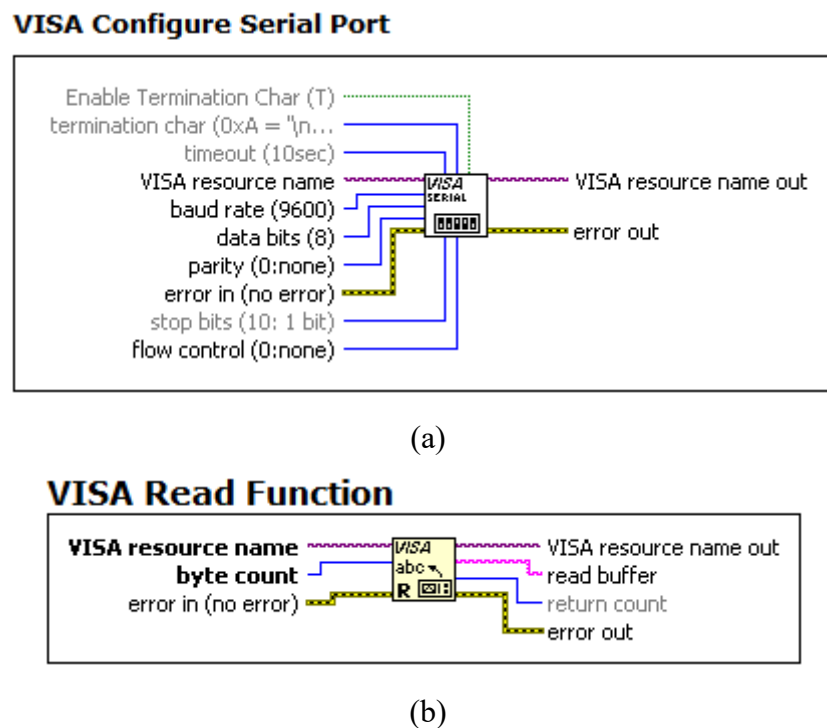
Fonte: Autor

Como este instrumento tem uma saída serial padrão RS-232, não há necessidade de levar esse sinal para o Arduino para que ele chegue até o computador. Isso pode ser realizado diretamente utilizando uma entrada USB do computador. Para isso, basta um conversor serial para USB (ver Figura 23) para que o sinal do condutivímetro possa ser lido diretamente no LabVIEW.

4.4.2 O Software

O sinal disponível na porta USB do computador pode ser facilmente adquirido usando o *software* LabVIEW. O LabVIEW já traz várias *sub-VI* de comunicação serial embutidas em suas bibliotecas. No presente estudo, usou-se a *sub-VI* “VISA Configure Serial Port VI” e a *sub-VI* “VISA Read Function”. Essas *sub-VI* configuram a porta USB a ser lida e lê dados dessa porta, respectivamente. A Figura 24.a e 24.b mostram as entradas e saídas dessas *sub-VI*. O DM-32 usa uma velocidade de comunicação (*baud rate*) de 9600 *bits* por segundo, sem paridade, número de bits igual a 8 e *bit* de parada igual a 1.

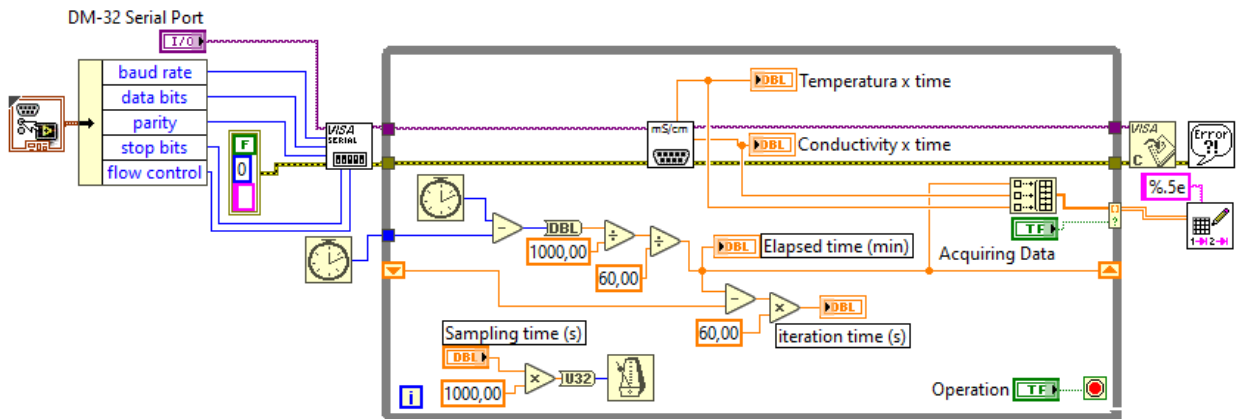
Figura 24. Entradas e saídas das sub-VI: (a). VISA Configure Serial Port VI. (b). VISA Read Function.



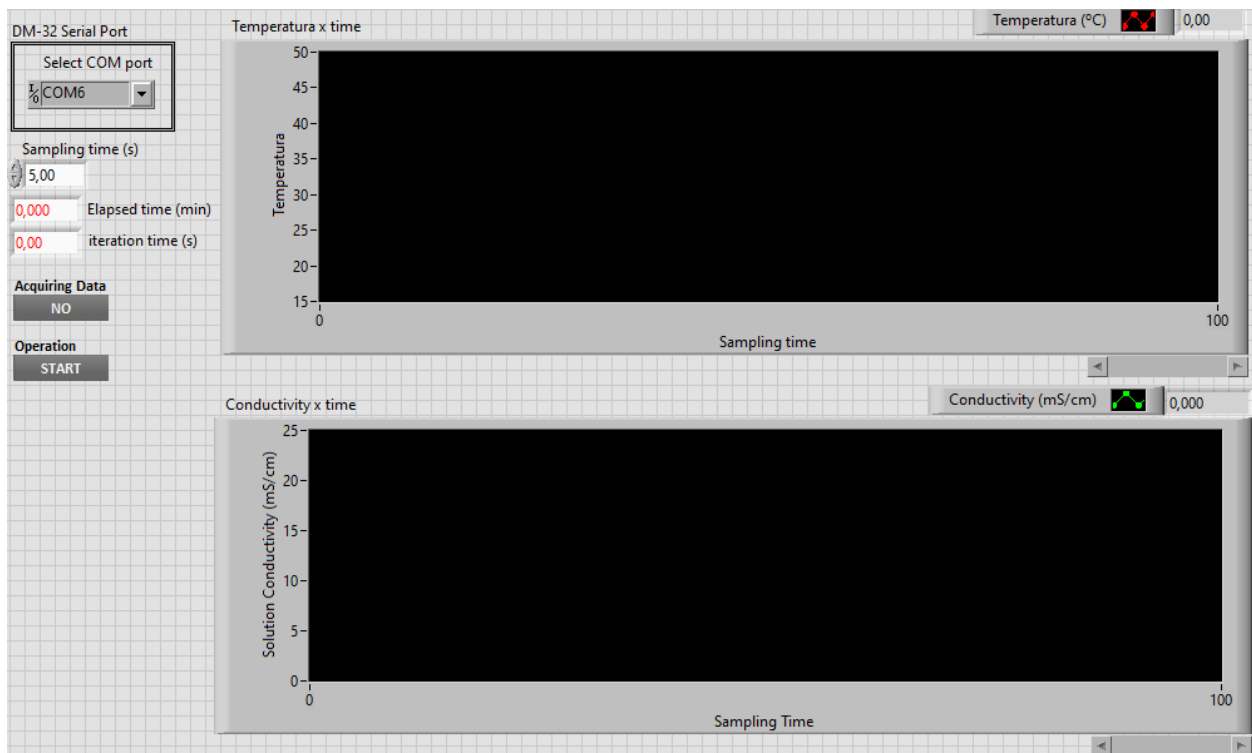
Fonte: Autor

A Figura 25.a e 25.b mostra o diagrama de blocos e o painel frontal, respectivamente, da VI construída para ler dados de condutividade elétrica e de temperatura medidas pelo DM-32 ao longo de tempo.

Figura 25. VI construída para ler dados do condutivímetro DM-32. (a). Diagrama de blocos. (b). Painel frontal.



(a)



(b)

Fonte: Autor

4.5 Medição de Massa (Célula de Carga)

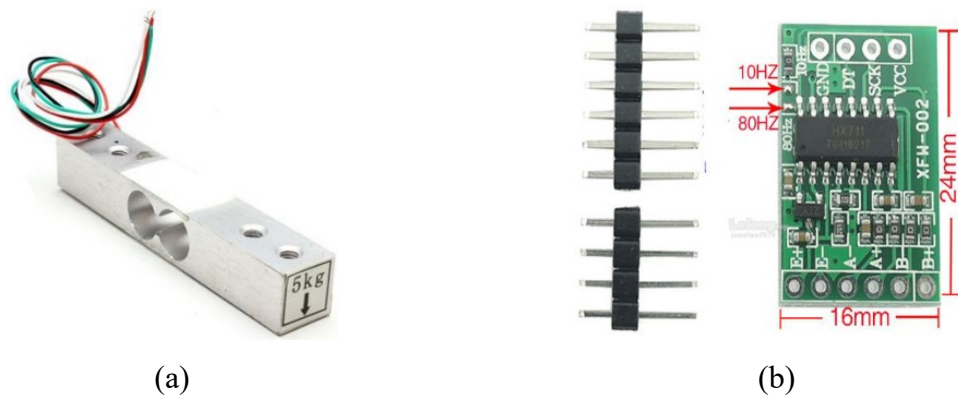
Neste trabalho os sensores de vazão utilizados (YF-S402) foram calibrados utilizando uma balança construída na oficina da FEQUI/UFU. Esta balança foi conectada à placa Arduino Mega 2560 e pela aquisição do volume/tempo determinou-se a vazão. Curvas de calibração dos sensores de vazão foram, então, construídas para todos os sensores. O próximo capítulo mostra os resultados das calibrações.

4.5.1 O Hardware

A massa de líquido das correntes foi medida utilizando uma célula de carga de 5 kg tipo *strain gauge*. Esta célula de carga é constituída de um invólucro de metal e uma pastilha *strain gauge* instalada junto ao seu corpo. *Strain gauge* é um dispositivo de modifica sua resistência elétrica sempre que submetido à um esforço físico de compressão ou tração. Um módulo amplificador de sinal é necessário para amplificar o sinal de saída do sensor para a faixa de 0 a 5V. O módulo amplificador utilizador foi o modelo HX711. A Figura 26.a e 26.b mostra a célula de carga e o amplificador utilizados. As especificações da célula de carga são:

- Carga nominal: 0 a 5 kg;
- Potência nominal de saída: $1.0\text{mv/v} \pm 0.15\text{mv/v}$;
- Zero saída: $\pm 0,1\text{mv/v}$;
- Precisão: 0,03% f.s./30 min;
- Extremidade de entrada: Vermelho + (energia), Preto - (energia);
- Extremidade de saída: Verde + (sinal), Branco - (sinal);
- Tensão de funcionamento recomendada: 3 ~ 12 VDC;
- Tensão máxima de operação: 15 VDC;
- Impedância de entrada: 410 ± 30 ohms;
- Impedância de saída: 350 ± 3 ohms;
- Isolamento: $> 2000 \text{ M}\Omega/50 \text{ VDC}$;
- Faixa de compensação de temperatura: $-10\text{C} \sim 40\text{C}$;
- Faixa de temperatura de operação: $-20\text{C} \sim 60\text{C}$;
- Classe de proteção: IP65;
- Material: liga de alumínio;
- Dimensões (Comprimento x Largura x Espessura): $80 \times 12,7 \times 12,7\text{mm}$;
- Peso: 31g.

Figura 26. (a). Célula de carga de 5 kg. (b). Amplificador HX711



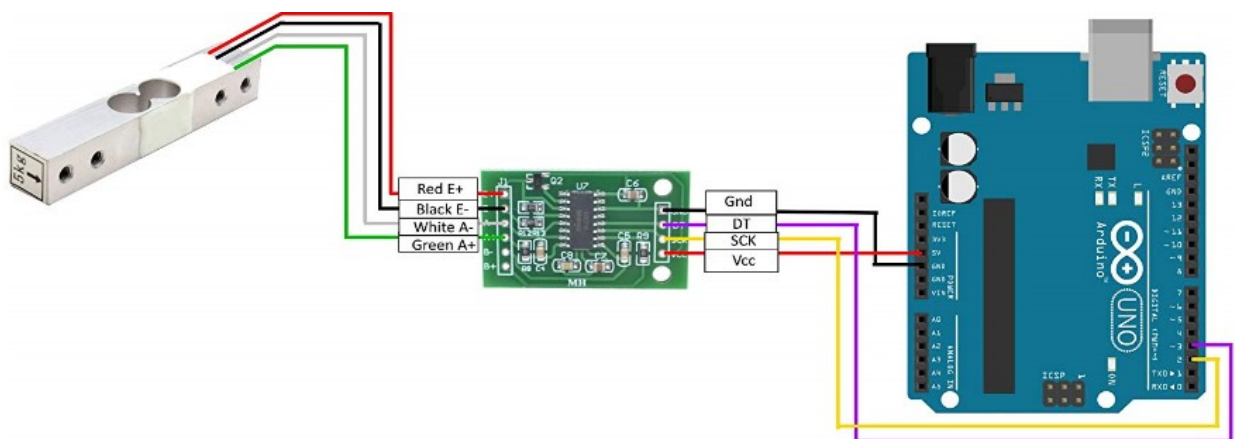
Fonte: Autor

As especificações do amplificador são:

- Modelo: HX711
- Tensão de operação: 4,8 a 5,5V DC
- Corrente de operação: 1,6mA
- Temperatura de operação: -20 a 85°C
- Interface SPI
- Dimensões: 29 x 17 x 4mm (sem os pinos)

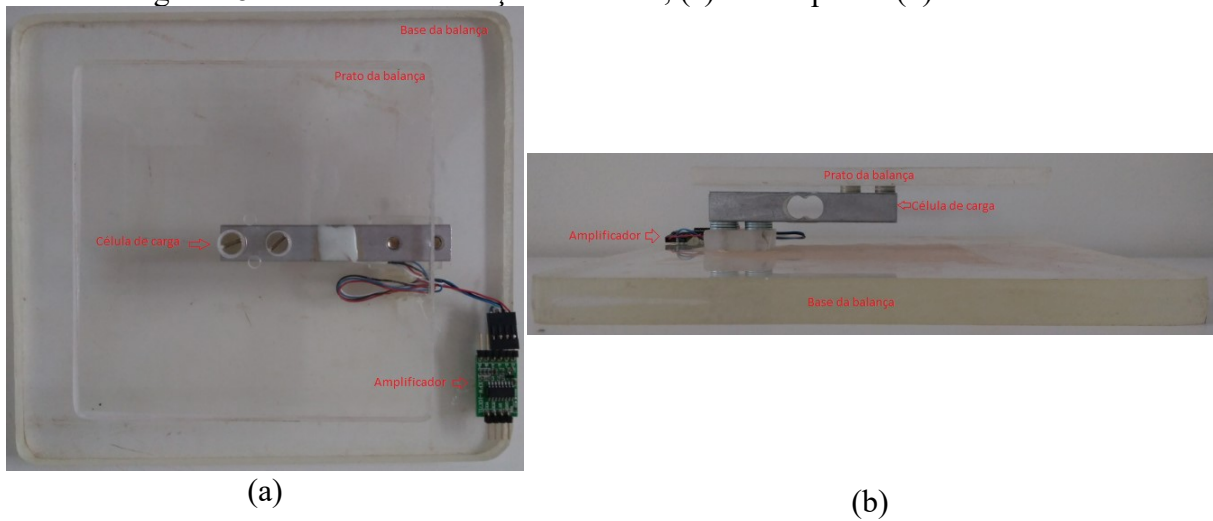
A Figura 27 mostra as ligações entre célula de carga, amplificador e placa Arduino. A Figura 28 mostra fotos superior e lateral da balança construída. Duas placas de acrílico foram usadas como base e prato da balança.

Figura 27. Detalhes das ligações entre célula de carga, amplificador e placa Arduino.



Fonte: Autor

Figura 28. Detalhes da balança construída, (a) foto superior (b) foto lateral.



Fonte: Autor

4.5.2 O Software

A *LabVIEW Interface for Arduino* (LIFA) elaborada pela *National Instruments* (NI) também não contempla biblioteca para o amplificador HX711. Então, antes de usar o sensor foi necessário alterar a LIFA para adquirir os sinais deste amplificador. As linhas de código a seguir foram, então, adicionadas à LIFA para configurar e ler dados do amplificador HX711 (Figura 29).

Figura 29. Linhas de código adicionadas à LIFA para adquirir os sinais do amplificador HX711.

```

/*****
** Set Pins to Weighting scale (A LoadCell with HX711 Dual-channel 24-bit amplifier)
*****/
case 0x39: // Set Pins as Input and Output to LoadCell with HX711 amplifier
    pinMode(command[2], INPUT_PULLUP); //Pin DT do HX711 (entrada para receber os dados do pino DT)
    pinMode(command[3], OUTPUT); //Pin SCK do HX711 (saída para o pino SCK)
    Serial.write('0');
    break;

/*****
** Read data from LoadCell with HX711 Dual-channel 24-bit amplifier)
*****/
case 0x3A: // Read data from LoadCell with HX711 amplifier
    ICount = ReadCount(command);
    Serial.write((ICount >> 24 & 0xFF));
    Serial.write((ICount >> 16 & 0xFF));
    Serial.write((ICount >> 8 & 0xFF));
    Serial.write(ICount & 0xFF);
    break;

```

Fonte: Autor

A função *ReadCount* é uma subrotina que também foi incorporada à *LIFA* e quem efetivamente faz a comunicação com o amplificador HX711 (Figura 30).

Figura 30. Linhas de código inserindo a função *ReadCount*.

```
unsigned long ReadCount(unsigned char command[])
{
    unsigned long Count = 0;
    unsigned char i;
    digitalWrite(command[3], LOW);
    while(digitalRead(command[2]));
    for(i=0;i<24;i++){
        digitalWrite(command[3], HIGH);
        Count = Count << 1;
        digitalWrite(command[3], LOW);
        if(digitalRead(command[2])) Count++;
    } //end for
    digitalWrite(command[3], HIGH);
    Count = Count^0x800000;
    digitalWrite(command[3], LOW);
    return(Count);
} //end ReadCount
```

Fonte: Autor

CAPÍTULO 5

5 CALIBRAÇÃO DOS SENSORES DE VAZÃO

5.1 Introdução

As unidades experimentais de reação, três reatores de mistura e um reator tubular, são dotadas de duas correntes de entrada cada. Para a operação das unidades, as vazões volumétricas dessas correntes serão controladas automaticamente. Para isso, é necessário calibrar os 4 sensores tipo Hall modelo YF-S402, obter as curvas de calibração por regressão linear, validar estatisticamente as curvas de calibração, obter os intervalos de confiança dos parâmetros da curva. A calibração é realizada medindo-se o volume de líquido coletado em um intervalo de tempo. A medida do volume é realizada automaticamente utilizando-se a balança construída com a célula de carga de 5 kg discutida no Capítulo 4. Os dados são adquiridos via Arduino e armazenados em arquivo texto (*.txt). O tempo também é medido utilizando bibliotecas de temporização do LabVIEW.

5.2 Modelo de Regressão Linear Simples

Muitos problemas de engenharia e ciências envolvem explorar as relações entre duas ou mais variáveis. Análise de regressão é uma técnica estatística para modelar e investigar a relação entre duas ou mais variáveis. A análise de regressão pode ser usada na instrumentação para construir modelos matemáticos para prever o valor de uma variável que se deseja medir, a partir do conhecimento da medida direta de outra variável. Como exemplo, pode-se citar a medição da vazão dada por um sensor/transmissor a partir do sinal de 4-20 mA ou 0-10V enviado pelo transmissor, ou ainda, a medição do nível de líquido num tanque dado por um sensor/transmissor de pressão diferencial com célula capacitiva a partir do sinal de 4-20 mA ou 0-10V enviado pelo transmissor. Como geralmente os instrumentos de medição são desenvolvidos para se obter uma relação linear entre variáveis medidas e transmitidas, os modelos de regressão linear serão aqui abordados.

5.2.1 *Estimação de Parâmetros em um Modelo de Regressão Linear Simples*

A regressão linear simples considera um único regressor ou preditor “ x ” e uma única variável dependente ou variável resposta “ y ”. Suponha que a relação verdadeira entre “ y ” e “ x ” seja uma linha descrita como:

$$y(x) = \beta_0 + \beta_1 x \quad (5.1)$$

As medidas que são efetuadas, no entanto, nunca são exatas pois o valor verdadeiro de uma variável é sempre corrompido por ruídos durante o processo de medida. O erro é inerente ao próprio processo de medida, isto é, nunca será completamente eliminado. Desta forma a Equação 5.1 pode ser mais convenientemente escrita em termos da variável medida como:

$$y(x) = \beta_0 + \beta_1 x + \varepsilon \quad (5.2)$$

Em que $y(x)$ é a medida da variável realizada com algum instrumento de medida para um determinado valor “ x ” da variável independente e “ ε ” é o erro cometido na medida de “ y ”. Na ausência de erros sistemáticos o erro de medida “ ε ” deveria ser um ruído branco, isto é, uma variável aleatória normalmente distribuída de média zero e de variância finita σ^2 . Se modelo for perfeito o erro de modelagem será reduzido ao ruído branco. Todo desenvolvimento a seguir será com a finalidade de testar essa hipótese e desenvolver uma maneira de estimar os parâmetros β_0 e β_1 . Seja y_i o valor medido da variável $y(x)$ para um determinado valor x_i de x . Então, pode-se definir o erro de modelagem como:

$$\varepsilon_i = y_i - y(x_i) \quad (5.3)$$

Suponha que se tenha agora “ n ” pares de observações experimentais $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. No método de mínimos quadrados os parâmetros do modelo (β_0 e β_1) são determinados a fim de se minimizar o somatório do quadrado dos erros de modelagem, isto é:

$$L = \sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (y_i - y(x_i))^2 = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2 \quad (5.4)$$

Os parâmetros β_0 e β_1 são agora determinados de forma que a função “ L ” esteja num ponto de mínimo da função. Matematicamente esse mínimo é obtido fazendo as derivadas $(\partial L/\partial \beta_0)$ e $(\partial L/\partial \beta_1)$ iguais a zero e resolvendo o sistema de duas equações para as incógnitas β_0 e β_1 . Obtém-se como resultado:

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \quad (5.5)$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n y_i x_i - \frac{(\sum_{i=1}^n y_i)(\sum_{i=1}^n x_i)}{n}}{\sum_{i=1}^n x_i^2 - \frac{(\sum_{i=1}^n x_i)^2}{n}} \quad (5.6)$$

Os valores $\hat{\beta}_0$ e $\hat{\beta}_1$ são as estimativas de mínimos quadrados dos parâmetros verdadeiros β_0 e β_1 . Na Equação 5.5 \bar{y} é a média dos y_i , isto é, $\bar{y} = (1/n) \sum_{i=1}^n y_i$ e \bar{x} é a média dos x_i , isto é, $\bar{x} = (1/n) \sum_{i=1}^n x_i$. Por conveniência de notação define-se os valores de S_{xx} e S_{xy} como:

$$S_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2 = \sum_{i=1}^n x_i^2 - \frac{(\sum_{i=1}^n x_i)^2}{n} \quad (5.7)$$

$$S_{xy} = \sum_{i=1}^n y_i(x_i - \bar{x})^2 = \sum_{i=1}^n y_i x_i - \frac{(\sum_{i=1}^n y_i)(\sum_{i=1}^n x_i)}{n} \quad (5.8)$$

Então a Equação 5.6 pode ser reescrita como:

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \quad (5.9)$$

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} \quad (5.10)$$

5.2.2 Estimando a Variância do Erro de Modelagem

Com os parâmetros $\hat{\beta}_0$ e $\hat{\beta}_1$ já determinados a variância amostral do erro de modelagem pode agora ser calculada usando a definição de variância, isto é:

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^n \varepsilon_i^2}{n - 2}$$

O número “ $n-2$ ” no denominador é devido a diminuição de um grau de liberdade uma vez que o somatório dos erros de modelagem é sempre zero. Por motivos de facilidade de nomenclatura introduz-se aqui o termo SS_E como somatório dos erros de modelagem ao quadrado:

$$SS_E = \sum_{i=1}^n \varepsilon_i^2$$

Mas como:

$$\begin{aligned} SS_E &= \sum_{i=1}^n (y_i - y(x_i))^2 = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2 \\ &= \sum_{i=1}^n y_i^2 - n\bar{y}^2 - \hat{\beta}_1 S_{xy} = \sum_{i=1}^n (y_i - \bar{y})^2 - \hat{\beta}_1 S_{xy} \end{aligned}$$

Então:

$$SS_E = S_{yy} - \hat{\beta}_1 S_{xy} \quad (5.11)$$

$$\hat{\sigma}^2 = \frac{SS_E}{n-2} \quad (5.12)$$

5.2.3 Valor Esperado dos Parâmetros e Variância dos Estimadores de Mínimos Quadrados

A esperança matemática e a variância dos estimadores $\hat{\beta}_0$ e $\hat{\beta}_1$ de mínimos quadrados podem ser facilmente determinados (MONTGOMEY & RUNGER, 2016) e são dados por:

$$E(\hat{\beta}_1) = \beta_1 \quad \text{e} \quad E(\hat{\beta}_0) = \beta_0 \quad (5.13)$$

$$V(\hat{\beta}_1) = \frac{\hat{\sigma}^2}{S_{xx}} \quad \text{e} \quad V(\hat{\beta}_0) = \hat{\sigma}^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} \right) \quad (5.14)$$

A Equação 5.13 revela que os estimadores $\hat{\beta}_0$ e $\hat{\beta}_1$ são estimadores não tendenciosos dos parâmetros verdadeiros do modelo β_0 e β_1 , respectivamente.

5.2.4 Testando a Significância da Regressão Linear

Uma importante parte da verificação da adequação de um modelo de regressão linear é a realização de um teste estatístico de hipóteses em relação aos parâmetros do modelo e a construção de intervalos de confiança para eles. Para testar as hipóteses sobre a inclinação e a interseção do modelo de regressão, é necessário fazer a suposição adicional de que o erro do modelo (e_i) é uma variável aleatória normalmente distribuída com média “zero” e variância finita, representado abreviadamente como $N(0, \sigma^2)$.

Para testar a significância do modelo é preciso testar a hipótese da inclinação β_1 ser uma constante diferente de zero. Assim:

$$H_0: \beta_1 = 0$$

$$H_1: \beta_1 \neq 0$$

Se o teste estatístico levar a rejeição da hipótese nula, isto é $\beta_1 = 0$, a hipótese alternativa é válida e assim o modelo é estatisticamente significativo. Falha na rejeição de H_0 é equivalente a concluir que não existe relação linear entre “ x ” e “ y ”.

Uma vez que os erros de modelagem são $N(0, \sigma^2)$, segue diretamente que as observações y_i são $N(\beta_0 + \beta_1 x_i, \sigma^2)$. Da Equação 5.6 observa-se que $\hat{\beta}_1$ é uma combinação linear de variáveis aleatórias e, conseqüentemente, $\hat{\beta}_1$ é $N(\beta_1, \sigma^2/S_{xx})$. Em decorrência disso, é possível mostrar que a razão $(n-2)\hat{\sigma}^2/\sigma^2$ tem uma distribuição chi-quadrado com $(n-2)$ graus de liberdade e $\hat{\beta}_1$ é independente de $\hat{\sigma}$. Como resultado destas propriedades, a estatística:

$$t_0 = \frac{\hat{\beta}_1}{\sqrt{\hat{\sigma}^2/S_{xx}}} \quad (5.15)$$

Segue uma distribuição *t-student* com $(n-2)$ graus de liberdade sob a hipótese $H_0: \beta_1 = 0$. Esta hipótese é rejeitada quando:

$$|t_0| > t_{\alpha/2, n-2} \quad (5.16)$$

Em que “ α ” é a probabilidade de se aceitar hipótese H_0 como verdadeira, também chamado de nível de significância do teste, valor tipicamente assumido como 1% a 5% em aplicações práticas. Um procedimento similar pode ser desenvolvido para testar a relevância estatística do intercepto do modelo (β_0):

$$H_0: \beta_0 = 0$$

$$H_1: \beta_0 \neq 0$$

$$t_0 = \frac{\hat{\beta}_0}{\sqrt{\hat{\sigma}^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} \right)}} \quad (5.17)$$

O valor $t_{\alpha/2, n-2}$ é o valor da estatística “*t-student*” em que “ α ” é o nível de significância do teste e “ $n-2$ ” é o número de graus de liberdade do teste (“ n ” é o número de pontos experimentais medidos).

5.2.5 Intervalo de Confiança para $\hat{\beta}_0$ e $\hat{\beta}_1$

Além de estimar o valor do intercepto ($\hat{\beta}_0$) e o valor da inclinação ($\hat{\beta}_1$) da reta de mínimos quadrados, pode-se também determinar o intervalo de confiança para estes parâmetros. A largura desses intervalos de confiança é uma medida da qualidade global da linha de regressão. Se os erros (ε_i) do modelo de regressão forem normal e independentemente distribuídos, então as variáveis,

$$\frac{\hat{\beta}_1}{\sqrt{\hat{\sigma}^2/S_{xx}}} \quad \text{e} \quad \frac{\hat{\beta}_0}{\sqrt{\hat{\sigma}^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} \right)}}$$

são ambas variáveis randômicas com distribuição *t-student* com $(n-2)$ graus de liberdade. Isto conduz à seguinte definição de intervalos de confiança de $100(1-\alpha)\%$ para a inclinação e para a interseção:

$$\hat{\beta}_1 - t_{\alpha/2, n-2} \sqrt{\hat{\sigma}^2/S_{xx}} \leq \beta_1 \leq \hat{\beta}_1 + t_{\alpha/2, n-2} \sqrt{\hat{\sigma}^2/S_{xx}} \quad (5.18)$$

$$\hat{\beta}_0 - t_{\alpha/2, n-2} \sqrt{\hat{\sigma}^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} \right)} \leq \beta_0 \leq \hat{\beta}_0 + t_{\alpha/2, n-2} \sqrt{\hat{\sigma}^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} \right)} \quad (5.19)$$

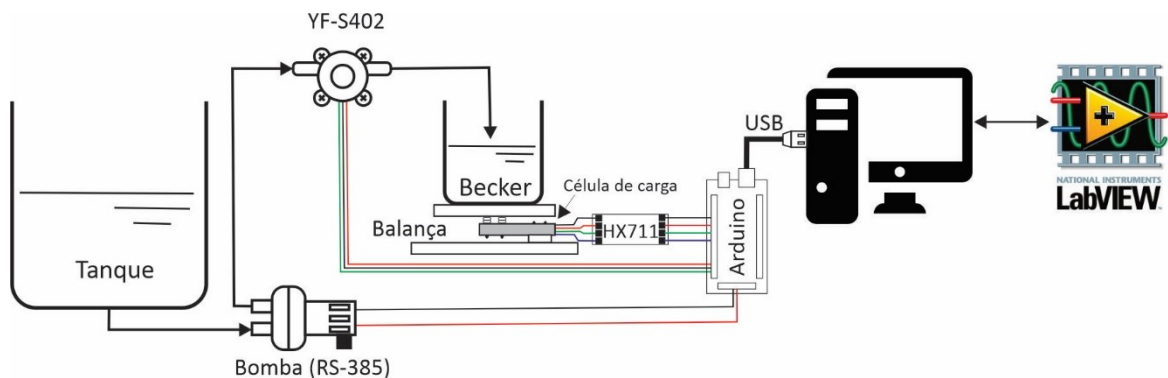
O valor $t_{\alpha/2, n-2}$ é o valor da estatística “*t-student*” em que “ α ” é o nível de significância do teste e “ $n-2$ ” é o número de graus de liberdade do teste (“ n ” é o número de pontos experimentais medidos).

5.3 Materiais

A Figura 31 apresenta de maneira esquemática o sistema de calibração construído e usado na calibração dos sensores de vazão YF-S402. Os materiais a seguir foram usados:

- Microcontrolador baseado no ATMEL ATmega2560 modelo Arduino Mega 2560;
- Minibomba 12 VDC modelo RS485 vazão máxima de 2 L/min;
- Sensor de efeito Hall modelo YF-S402 vazão máxima de 6 L/min;
- Célula de carga 5 Kg - Sensor de Peso;
- Conversor/amplificador HX711 de 24 bits;
- Notebook tipo IBM-PC Intel Core i7 rodando sob plataforma Windows 10;
- Fonte de tensão modelo ATX 12 VDC e 20A.
- *Software* LabVIEW 2020 versão do estudante;
- *Software* Arduino IDE versão 1.8.9 (Windows Store 1.8.21.0)
- Tubulações e acessórios;
- Cabos e conexões para interligação dos instrumentos;
- Tanque 30 L para armazenamento de água;
- *Becker* 5 L para recolhimento de água.

Figura 31. Diagrama esquemático do sistema de calibração do sensor de efeito Hall YF-S402.

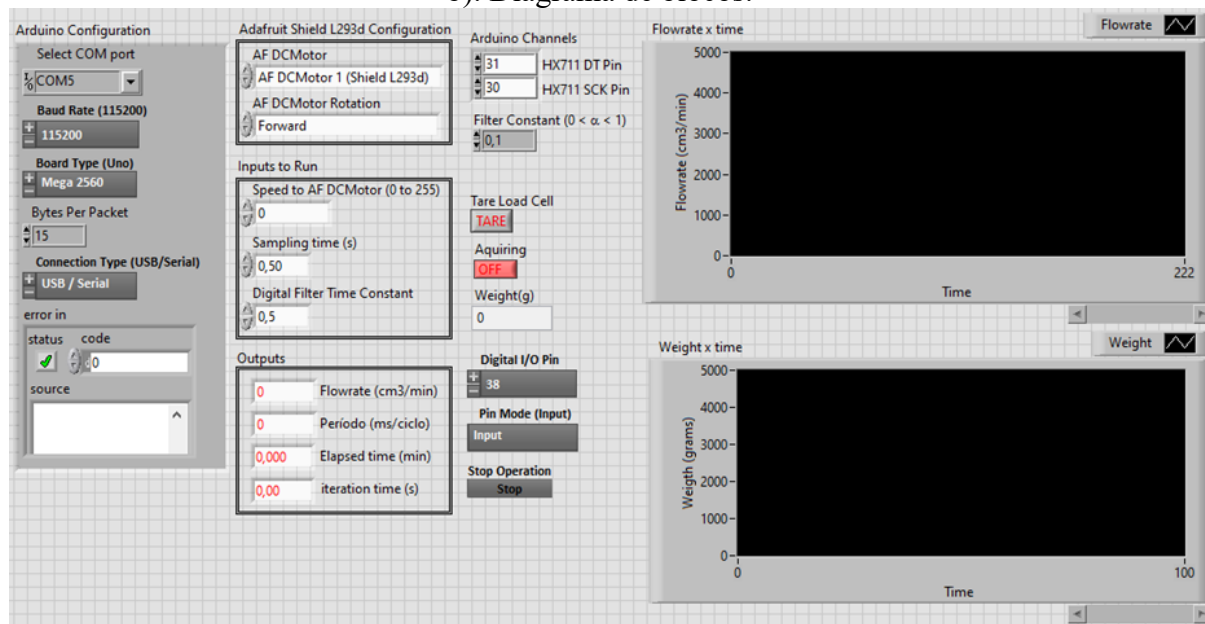


Fonte: Autor

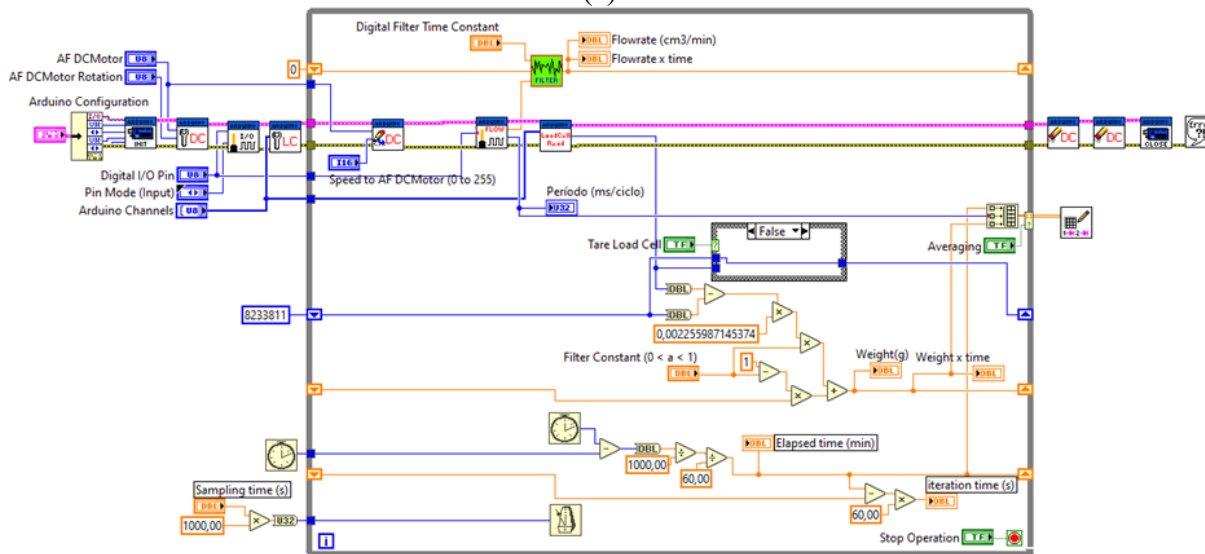
5.4 O Software de Calibração

A calibração dos sensores foi realizada acionando a minibomba com um sinal PWM (0-255) enviado manualmente de uma tela do LabVIEW através da placa Arduino. O sinal onda de 0-5V enviado pelo sensor tem seu período medido em *bits* (0-1024) pelo Arduino e enviado de volta ao LabVIEW que a transforma em período em microssegundos. O líquido bombeado é recolhido em um *becker* localizado no prato da balança construída, conforme revela a Figura 31.

Figura 32. Script LabVIEW para calibração dos sensores de vazão YF-S402. a). Painel frontal. b). Diagrama de blocos.



(a)



(b)

Fonte: Autor

O LabVIEW registra o volume coletado ao longo do tempo associado ao sinal PWM enviado à minibomba. Ao se atingir aproximadamente 2000 g de líquido coletado, interrompe-

se o processo e um arquivo de dados no formato texto (*.txt) é gerado contendo o tempo transcorrido, o sinal PWM enviado à minibomba, o volume coletado e o período da onda gerada pelo sensor YF-S402. O processo foi repetido com sinal PWM variando de 130 a 250. A Figura 32a. e 32b. mostram o painel frontal e o diagrama de blocos do *script* LabVIEW desenvolvido.

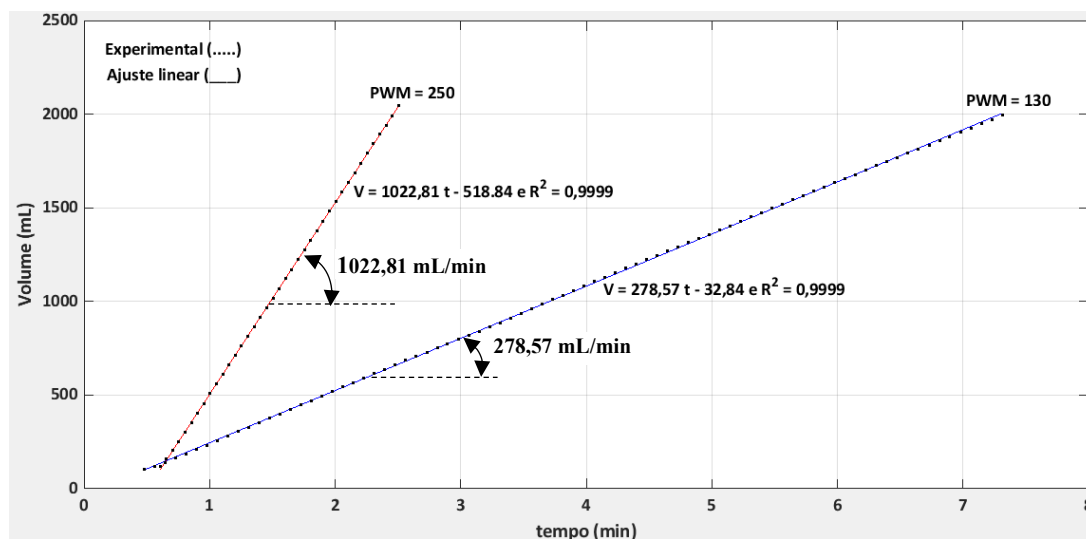
5.5 Resultados e Discussão

Por economia de espaço a Tabela 2 apresenta somente parte dos dados obtidos na calibração dos dois sensores YF-S402 da unidade do reator PFR. O sinal PWM enviado para as minibombas variam de 130 à 250. A Figura 33 mostra esses dados na forma gráfica.

Tabela 2. Dados resumidos adquiridos da balança na calibração do sensor 1 da unidade PFR.

PWM = 130			PWM = 250		
tempo (min)	Período (ms)	Peso (g)	tempo (min)	Período (ms)	Peso (g)
...
3,255	23564	869,338	2,105	5771	1636,201
3,263	23571	872,163	2,114	5770	1644,758
3,271	23577	874,343	2,122	5803	1652,939
3,280	23508	876,326	2,130	5786	1661,385
3,288	23578	878,852	2,139	5812	1669,914
3,296	23225	880,915	2,147	5805	1678,904
3,305	23607	883,397	2,155	5795	1687,438
3,313	23664	885,738	2,164	5837	1696,017
3,321	23520	888,382	2,172	5839	1704,222
...

Figura 33. Volume *versus* tempo para cada sinal PWM enviado para a minibomba do sensor 1 da unidade PFR.



Fonte: Autor

Uma regressão dos dados de volume *versus* tempo para cada sinal PWM enviado para a minibomba foi realizada. A Figura 33 mostra os dados de volume *versus* tempo e as retas de regressão. O coeficiente angular dessas retas é a vazão. Repetindo-se esse procedimento para PWM de 130 a 250 foi possível construir a Tabela 3 para cada vazão em função do período e, conseqüentemente da frequência da onda quadrada gerada pelo sensor.

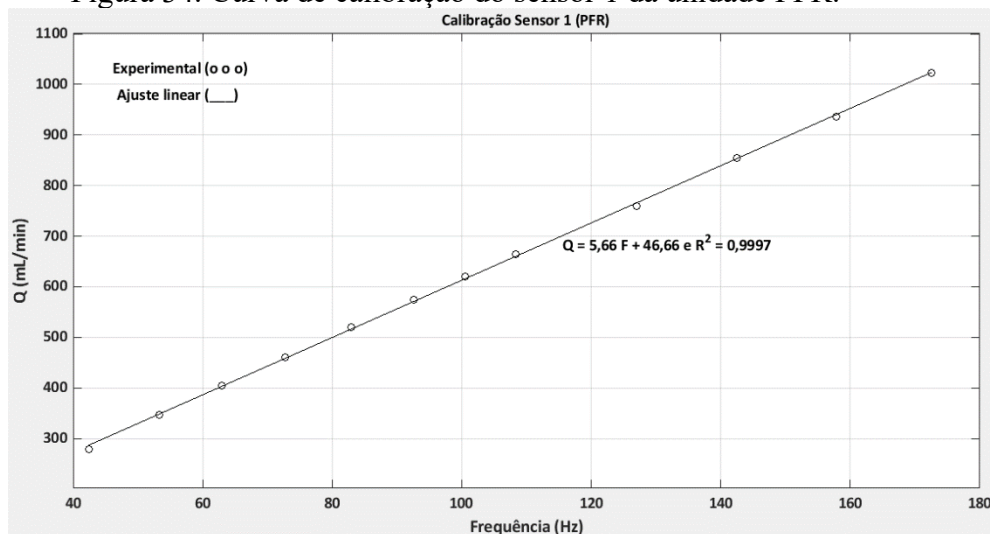
Tabela 3. Vazão (mL/min) *versus* frequência (Hz) do sensor 1 da unidade PFR.

PWM	Período (μs/ciclo)	Frequência (Hz)	Q (mL/min)
130	23542,62	42,48	278,57
140	18750,81	53,33	346,59
150	15887,72	62,94	404,89
160	13741,02	72,77	461,28
170	12064,29	82,89	519,30
180	10801,09	92,58	575,06
190	9951,58	100,49	619,81
200	9235,60	108,28	663,81
220	7873,52	127,01	759,74
230	7017,85	142,49	855,39
240	6334,02	157,88	936,26
250	5795,14	172,56	1022,81

Com os dados da Tabela 3 realizou-se uma regressão linear da vazão *versus* frequência da onda quadrada gerada pelo sensor. Essa reta de regressão é a curva de calibração do sensor. Essa curva pode ser vista na Figura 34.

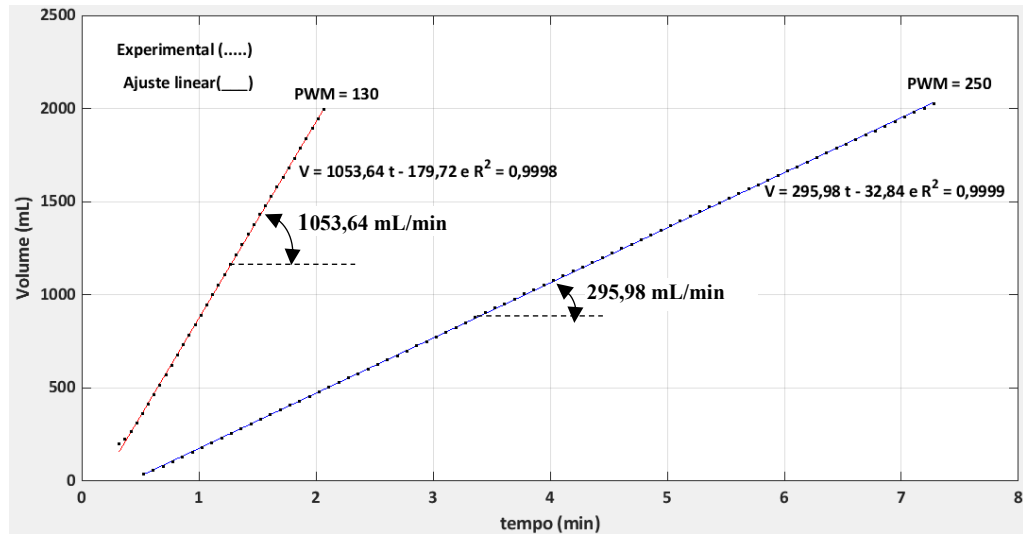
Dados semelhantes foram obtidos para o sensor 2 da unidade do PFR e para os sensores 1 e 2 da unidade de 3 CSTR em série. As Figuras e Tabelas a seguir mostram os dados obtidos e as curvas de regressão geradas para os demais sensores.

Figura 34. Curva de calibração do sensor 1 da unidade PFR.



Fonte: Autor

Figura 35. Volume *versus* tempo para cada sinal PWM enviado para a minibomba do sensor 2 da unidade PFR.

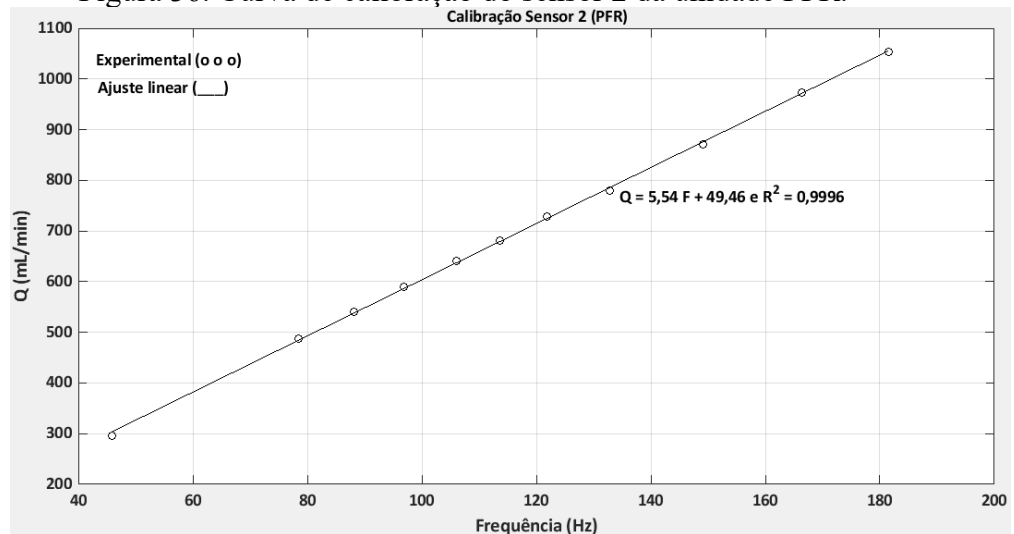


Fonte: Autor

Tabela 4. Vazão (mL/min) *versus* frequência (Hz) do sensor 2 da unidade PFR.

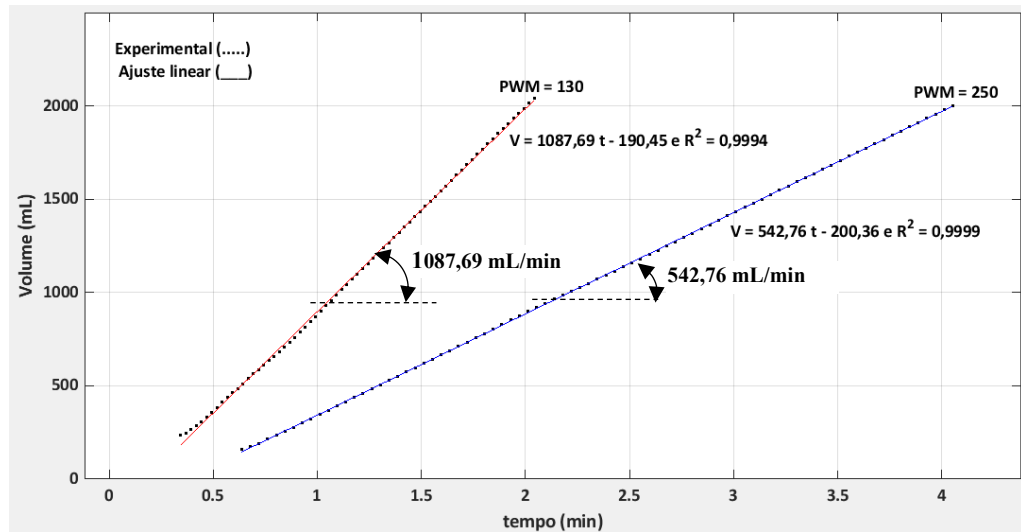
PWM	Período (μ s/ciclo)	Frequência (Hz)	Q (mL/min)
130	21757,24	45,96	295,98
160	12746,02	78,46	487,06
170	11348,21	88,12	540,92
180	10318,26	96,92	590,01
190	9424,59	106,11	640,88
200	8801,32	113,62	681,79
210	8207,19	121,84	728,46
220	7532,17	132,76	780,23
230	6705,23	149,14	870,04
240	6013,35	166,30	973,39
250	5508,04	181,55	1053,64

Figura 36. Curva de calibração do sensor 2 da unidade PFR.



Fonte: Autor

Figura 37. Volume *versus* tempo para cada sinal PWM enviado para a minibomba do sensor 1 da unidade de 3 CSTR em série.

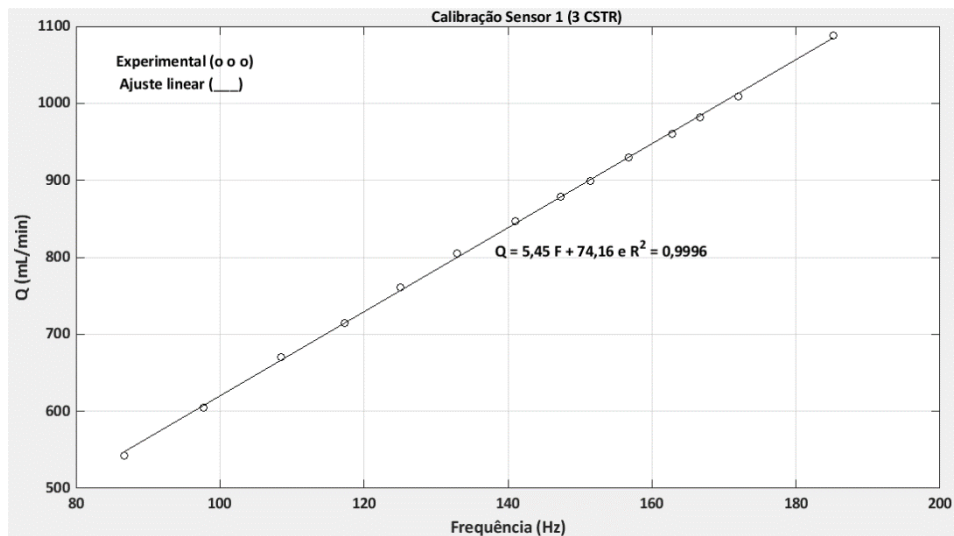


Fonte: Autor

Tabela 5. Vazão (mL/min) *versus* frequência (Hz) do sensor 1 da unidade 3 CSTR em série.

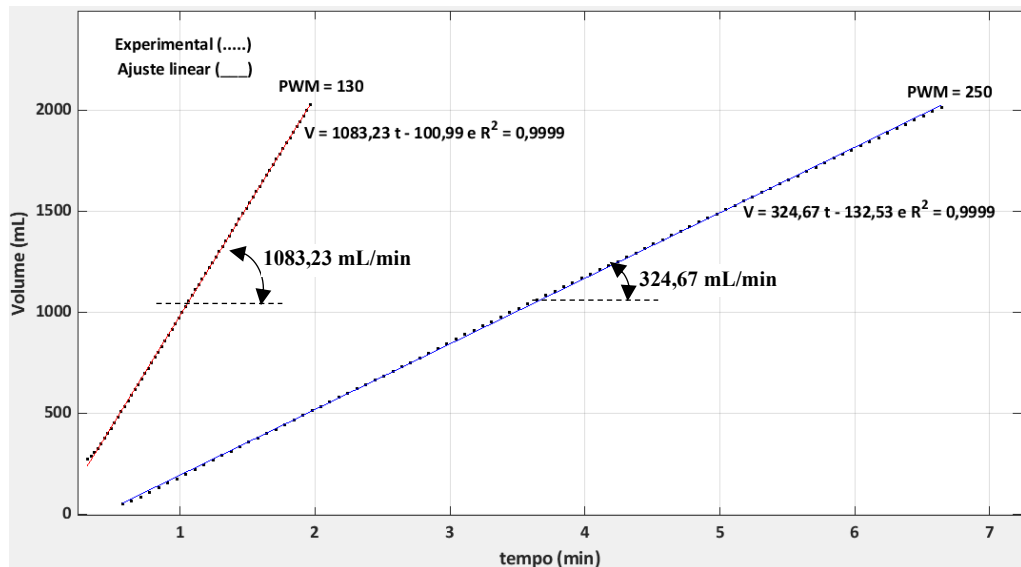
PWM	Período (µs/ciclo)	Frequência (Hz)	Q (mL/min)
120	11528,48	86,74	542,76
130	10231,37	97,74	605,11
140	9216,93	108,50	670,06
150	8528,04	117,26	714,49
160	7996,52	125,05	760,95
170	7521,59	132,95	804,75
180	7089,87	141,05	846,98
190	6790,85	147,26	878,76
200	6605,43	151,39	898,64
210	6378,49	156,78	929,18
220	6143,52	162,77	960,50
230	5999,57	166,68	981,44
240	5814,33	171,99	1008,69
250	5401,33	185,14	1087,69

Figura 38. Curva de calibração do sensor 1 da unidade de 3 CSTR em série.



Fonte: Autor

Figura 39. Volume *versus* tempo para cada sinal PWM enviado para a minibomba do sensor 2 da unidade de 3 CSTR em série.

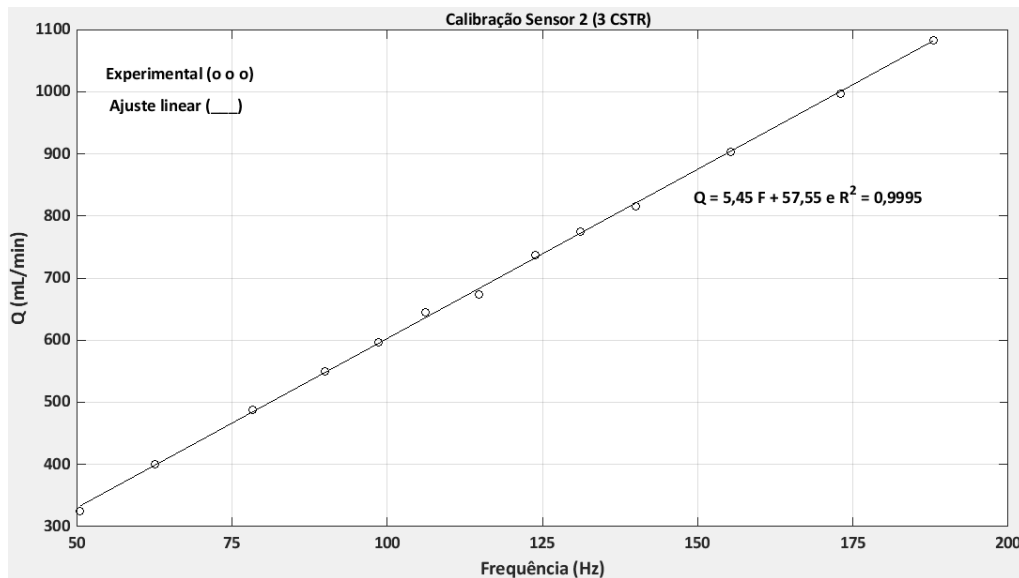


Fonte: Autor

Tabela 6. Vazão (mL/min) versus frequência (Hz) do sensor 2 da unidade 3 CSTR em série.

PWM	Período (μs/ciclo)	Frequência (Hz)	Q (mL/min)
130	19780,46	50,55	324,67
140	15953,92	62,68	400,92
150	12755,76	78,40	488,15
160	11114,65	89,97	549,80
170	10146,53	98,56	596,21
180	9422,33	106,13	645,38
190	8715,47	114,74	674,15
200	8074,00	123,85	736,84
210	7628,32	131,09	775,03
220	7143,75	139,98	815,60
230	6441,29	155,25	902,80
240	5780,44	173,00	997,02
250	5320,45	187,95	1083,23

Figura 40. Curva de calibração do sensor 2 da unidade de 3 CSTR em série.



As Figuras 33, 35, 37 e 39 revelam que o volume acumulado no *becker* é claramente uma função linear do tempo de bombeamento das minibombas. Os dados apresentam nível de ruído baixo ainda que o *script* do LabVIEW tenha implementado um filtro digital de primeira ordem com $\alpha = 0,1$ (constante do filtro).

As vazões correspondentes às frequências da onda quadrada 0-5V enviada pelo sensor YF-S402 foram, então, determinadas como o coeficiente angular da reta de regressão do volume versus tempo (dV/dt). As Tabelas 3, 4, 5 e 6 mostram as frequências e as vazões determinadas experimentalmente para os 4 sensores de vazão de efeito Hall YF-S402.

Os dados de vazão versus frequência foram então usados em regressões lineares para a determinação das curvas de calibração dos sensores. As Figuras 34, 36, 38 e 40 mostram as curvas de regressão, revelando um ajuste linear das curvas de calibração.

Com a finalidade de se fazer uma análise mais rigorosa acerca da validade das curvas de calibração, foi realizado o teste estatístico de hipóteses representado pelas Equações 5.15, 5.16 e 5.17, em relação aos parâmetros do modelo. Adicionalmente, determinou-se a validação estatística e os intervalos de confiança para os parâmetros das curvas de calibração para nível de confiança de 95% ($\alpha = 5\%$). Tabela 7 e 8 resume os parâmetros estatísticos das quatro equações de regressão dos sensores de vazão de efeito Hall.

Tabela 7. Análise estatística das equações de regressão da calibração dos sensores (parâmetros)

Sensor	β_0	β_1	$\Delta\beta_0$	$\Delta\beta_1$	$t_{\alpha/2, n-2}$	$t_0(\beta_0)$	$t_0(\beta_1)$
PFR 1	46,664	5,662	8,042	0,074	2,228	12,928	170,827
PFR 2	49,457	5,542	10,196	0,083	2,228	10,807	148,293
CSTR 1	74,156	5,459	10,116	0,071	2,228	16,332	170,972
CSTR 2	57,549	5,450	10,336	0,084	2,228	12,405	144,385

Tabela 8. Análise estatística das equações de regressão da calibração dos sensores. (equações e nível de confiança)

Sensor	Equação	Nível de confiança 95% ($\alpha = 5\%$)
PFR 1	$Q = 5,662 F + 46,664$	Sim
PFR 2	$Q = 5,542 F + 49,459$	Sim
CSTR 1	$Q = 5,459 F + 74,156$	Sim
CSTR 2	$Q = 5,450 F + 57,549$	Sim

Destas tabelas é possível verificar que as estatísticas “t”, especificamente $t_0(\beta_0)$ e $t_0(\beta_1)$, são todas maiores do que a estatística $t_{\alpha/2, n-2}$. Portanto, as hipóteses nulas H_0 são rejeitadas, o que leva a aceitação das hipóteses alternativas, confirmando estatisticamente a validade das equações de regressão. É possível também afirmar que as incertezas são bem menores para os parâmetros β_1 do que para os parâmetros β_0 , uma vez que $t_0(\beta_1) \gg t_0(\beta_0) > t_{\alpha/2, n-2}$.

Cabe aqui ressaltar que os sensores utilizados são surpreendentemente precisos uma vez que são sensores de baixo custo, o que viabiliza seu uso em montagens de laboratórios.

CAPÍTULO 6

6 CONSTRUÇÃO DAS UNIDADES EXPERIMENTAIS

6.1 Introdução

Com o intuito de desenvolver kits experimentais para o estudo de sistemas de reações em reator de mistura (CSTR – *Continuous Stirred Tank Reactor*) e reator tubular (PFR - *Plug Flow Reactor*), três reatores CSTR e um PFR em escala de laboratório foram construídos. Devido a finalidade do projeto ser uso em aulas práticas de cursos de graduação e pós-graduação, foi considerada a necessidade de montar as unidades experimentais de baixo custo.

Visando comparar resultados entre as unidades adotou-se a soma dos volumes dos três reatores de mistura como é ao volume do reator tubular. Assim é possível comparar resultados obtidos, analisar aspectos que os diferenciam e estudar mais detalhadamente as características dos dois tipos de sistemas.

6.2 Unidade Experimental – Bateria de Reatores de Mistura

Um reator de mistura consiste teoricamente em um tanque agitado em que as concentrações das espécies químicas são uniformemente distribuídas em todo volume do reator e o regime de escoamento é contínuo. A unidade experimental é composta de uma bateria de três reatores de mistura em série. Detalhes da construção e montagem serão mostrados nas próximas sessões.

6.2.1 Projeto dos reatores

Os reatores foram projetados baseados na premissa de se executar pelo menos 4 corridas no decorrer de três aulas de cinquenta minutos cada, com 15 minutos de intervalo entre as corridas para preparação, abastecimento dos tanques de armazenamento de solução, ajuste no LabVIEW e partida da corrida. Como a corrida mais longa será aquela executada com a menor vazão e tomando-se a menor vazão dada pelas bombas de alimentação do primeiro reator (vazão nominal de 400 mL/min) segue que:

$$t_{aula} = 3 \text{ aulas} \times 50 \frac{\text{min}}{\text{aula}} = 150 \text{min} \quad (6.1)$$

$$t_{total} = \frac{150 \text{ min}}{4 \text{ corridas}} = 37,5 \text{ min por corrida} \quad (6.2)$$

$$t_{líquido} = 37,5 \text{ min} - 15 \text{ min} = 22,5 \text{ min} \quad (6.3)$$

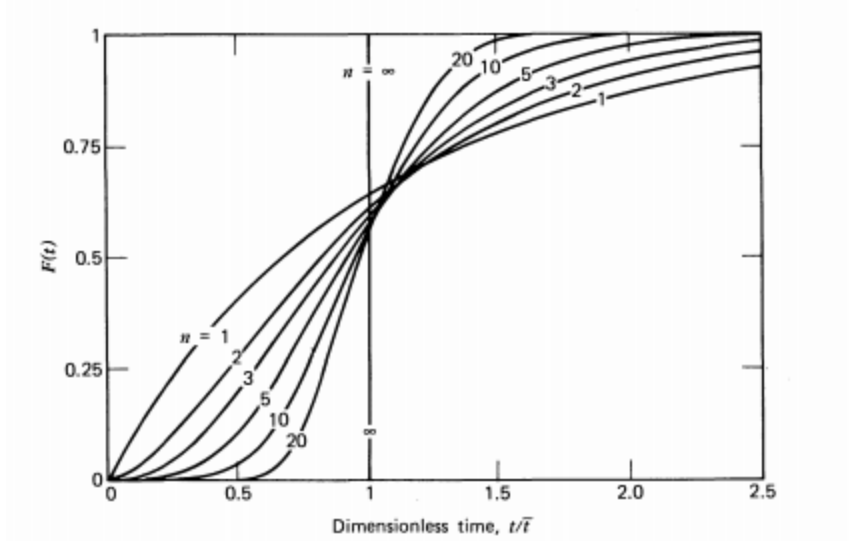
Em que:

t_{aula} : Tempo de aula

t_{total} : Tempo total de cada corrida;

$t_{líquido}$: Tempo líquido de cada corrida.

Figura 41. Distribuição do tempo de residência para o modelo n-CSTR



Fonte: HILL; CHARLES, 1977, p.406.

Segundo o modelo teórico de escoamento ideal para uma bateria de três reatores CSTR, num tempo de corrida igual a 5 vezes o tempo médio de residência da unidade, a função DTR atinge aproximadamente o valor final de 1, como mostrado na Figura 41. Dessa forma tem-se que:

$$\tau_{bateria} = \frac{t_{líquido}}{5} = \frac{22,5}{5} = 4,5 \text{ min} \quad (6.4)$$

Em que $\tau_{bateria}$ é tempo médio de residência da bateria de 3 reatores. Como a bateria tem três reatores, o tempo médio de residência de cada reator é dado por:

$$\tau_{reator} = \frac{\tau_{bateria}}{3} = \frac{4,5}{3} = 1,5 \text{ min} \quad (6.5)$$

Em que τ_{reator} é tempo médio de residência de um reator. Como o primeiro reator é alimentado por duas bombas, a vazão total (Q_{total}) de entrada no reator será de 2 x 400 mL/min = 800 mL/min. Assim, tem-se que:

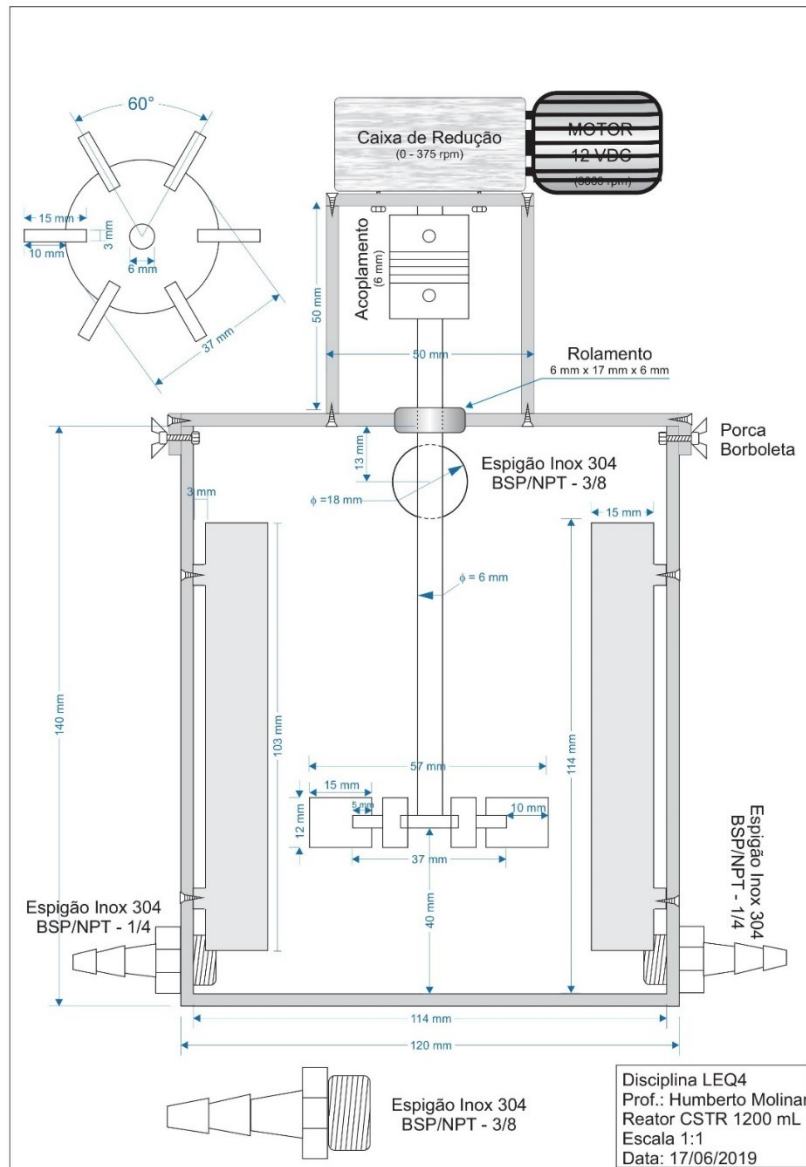
$$\tau_{reator} = \frac{V_{reator}}{Q_{total}} \rightarrow V_{reator} = \frac{1,5min}{800mL/min} = 1200mL \quad (6.5)$$

Em que V_{reator} é o volume de cada reator. Após a construção da unidade experimental fez-se a verificação dos volumes de cada um dos tanques. A Tabela 9 mostra os valores reais do volume de cada CSTR. A Figura 42 mostra o desenho técnico em escala utilizado na construção dos reatores.

Tabela 9. Volume real da unidade experimental CSTR.

Reator CSTR	Volume (cm³)
CSTR1	1246,6
CSTR2	1222,6
CSTR3	1186,3
CSTR _{Médio}	1218,5

Figura 42. Desenho técnico em escala do reator de mistura.



Fonte: Autor

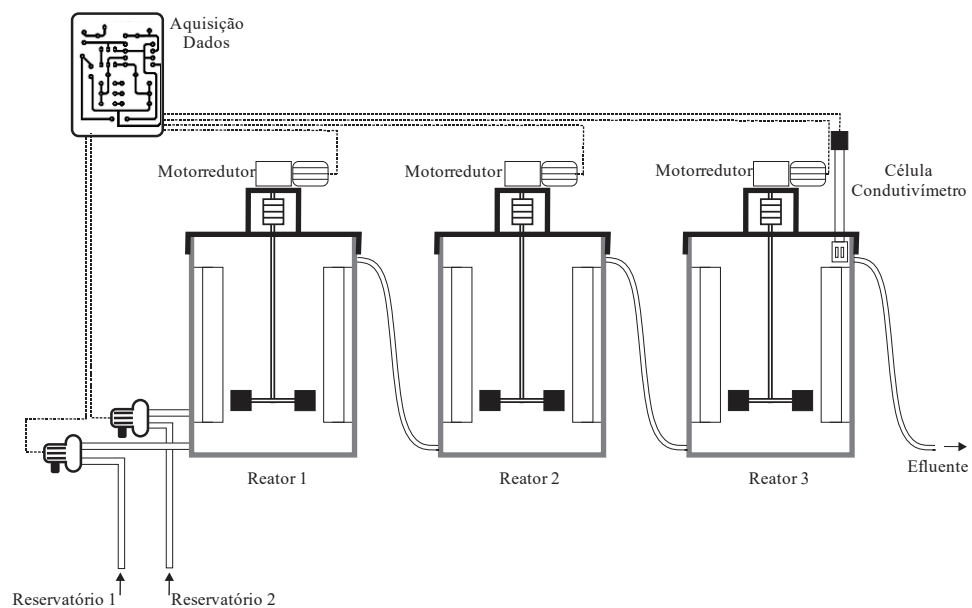
6.2.2 Materiais da unidade experimental

Para a montagem da unidade experimental dos três reatores de mistura usou-se os materiais a seguir:

- 2 tanques de 30L para armazenamento de água e de solução de NaCl;
- 3 reatores de acrílico com volume de aproximadamente 1200ml;
- Tubulações (mangueira de silicone $D_i=8\text{mm}$);
- 2 minibombas 12 VDC modelo RS485 com vazão máxima de 2 L/min;
- 2 sensores de efeito Hall modelo YF-S402 vazão máxima de 6 L/min;

- 1 microcontrolador baseado no ATMEL ATmega2560 modelo compatível Arduino Mega 2560;
- 1 microcontrolador baseado no ATMEL ATmega328 modelo compatível Arduino UNO R3;
- 2 *motorshields* Arduino L293D Driver Ponte H
- Motorreductor (Motor 12 VDC + Caixa de redução 0-375 rpm)
- Acoplamento de alumínio (6mm DI)
- Rolamento (6mm x 17mm x 6mm)
- Espigão Inox 304 BSP/NPT – 1/4”
- Porca borboleta 3/16”
- Barra Roscada UNC 3/16”
- Fonte de tensão modelo ATX 12 VDC e 20A;
- *Software* LabVIEW 2020 versão do estudante;
- *Software* Arduino IDE versão 1.8.9 (Windows Store 1.8.21.0);
- Cabos e conexões para interligação dos instrumentos (Cabo Flexível 3 vias 26 AWG);
- Condutímetro Digimed DM32;
- Conversor R232 para USB.

Figura 43. Representação esquemática da bateria de reatores de mistura.



Fonte: Autor.

6.2.3 Custo estimado

A Tabela 10 apresenta os elementos que compõem a unidade experimental da bateria de reatores CSTR com o custo estimado. O cálculo dos preços foi feito com a cotação de dólar a R\$3,30 (cotação do período de compras de materiais para este trabalho - março de 2018).

Tabela 10. Custo estimado a Unidade experimental da bateria de CSTR.

Elementos - Bateria de reatores CSTR	QTD	Valor Unit.	Valor (R\$)
Tanque de 30L para armazenamento das substâncias;	2	R\$ 24,00	R\$ 48,00
Tube acrílico (reator)	1	R\$ 68,14	R\$ 68,14
Tubulações (mangueira de silicone Di=8mm) (metros)	4	R\$ 22,50	R\$ 90,00
Minibomba 12 VDC modelo RS485 vazão máxima de 2 L/min;	2	R\$ 39,80	R\$ 79,60
Sensor de efeito Hall modelo YF-S402 vazão máxima de 6 L/min;	2	R\$ 13,22	R\$ 26,44
Microcontrolador baseado no ATMEL ATmega328P modelo Arduino UNO R3;	1	R\$ 22,44	R\$ 22,44
Microcontrolador baseado no ATMEL ATmega2560 modelo Arduino Mega 2560;	1	R\$ 76,90	R\$ 76,90
Motor Shield Arduino L293D Driver Ponte H	2	R\$ 19,90	R\$ 39,80
Fonte de tensão modelo ATX 12 VDC e 20A;	1	R\$ 79,00	R\$ 79,00
Software LabVIEW versão 2020;	1	R\$ 0,00	R\$ 0,00
Software Arduino IDE versão 1.8.9 (Windows Store 1.8.21.0);	1	R\$ 0,00	R\$ 0,00
Cabos e conexões para interligação dos instrumentos (Cabo Flexível 3 vias 26 AWG);	10	R\$ 0,80	R\$ 8,00
Conduvímeter Digimed DM32.	1	R\$ 3.336,00	R\$ 3.336,00
Motorreductor (Motor 12 VDC + Caixa de redução (0-375rpm))	3	R\$ 29,70	R\$ 89,10
Acoplamento de alumínio (6mm DI)	3	R\$ 2,71	R\$ 8,13
Rolamento (6mm x 17mm x 6mm)	3	R\$ 2,49	R\$ 7,47
Espigão Inox 304 BSP/NPT - 1/4"	7	R\$ 8,00	R\$ 56,00
Porca Borboleta 3/16"	12	R\$ 0,47	R\$ 5,64
Barra Roscada UNC 3/16" (metro)	3	R\$ 4,91	R\$ 14,73
Conversor R232 para USB	1	R\$ 16,90	R\$ 16,90
Total			R\$ 4.072,29

6.3 Unidade Experimental – Reator tubular

Um reator tubular consiste um tubo onde espécies químicas alimentadas no reator reagem à medida que avançam ao longo do comprimento do reator. Com o intuito de atingir

características próximas de um PFR e conservar a busca por baixo custo de montagem, o sistema montado foi simples e está apresentado nas sessões seguintes.

6.3.1 Projeto do reator

Como o objetivo do estudo, além de comparar os resultados experimentais com os teóricos, é também comparar os resultados obtidos entre os dois modelos de reatores, esse reator PFR foi projetado para que tivesse o mesmo volume que a soma de volumes da bateria de CSTR. Dessa forma, a partir dos cálculos já demonstrados do projeto dos reatores de mistura, definiu-se qual seria o volume do PFR. Uma mangueira de material plástico tipo cristal de uma polegada de diâmetro foi usada para construir o reator. O comprimento do tubo então calculado:

Tabela 11. Volume da unidade experimental PFR e reatores CSTR.

Reator	Volume (cm ³)
CSTR1	1246,6
CSTR2	1222,6
CSTR3	1186,3
PFR	3655,5

$$V_{PFR} = \text{Área}_{base-tubo} \times \text{Comprimento}_{tubo}$$

$$\text{Comprimento}_{tubo} = \frac{V_{PFR}}{\text{Área}_{base-tubo}}$$

Sendo,

$$\text{Área}_{base-tubo} = \pi \times r^2$$

Em que “r” é o raio do tubo.

$$\text{Comprimento}_{tubo} = \frac{3655,5}{\pi \times (2,54/2)^2}$$

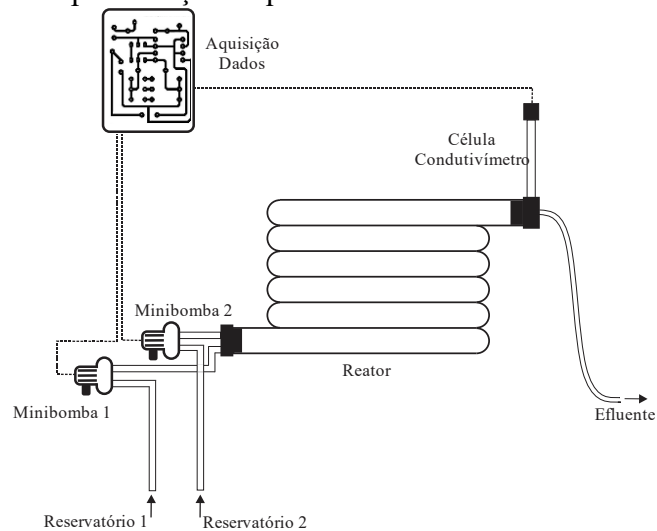
$$\text{Comprimento}_{tubo} = 722,14 \text{ cm}$$

6.3.2 Materiais da unidade experimental

- 2 tanques de 30L para armazenamento de água e de solução de NaCl;
- Mangueira plástica tipo cristal de 1" de diâmetro e 722 cm de comprimento;
- 2 minibombas 12 VDC modelo RS485 com vazão máxima de 2 L/min;
- 2 sensores de efeito Hall modelo YF-S402 vazão máxima de 6 L/min;
- 1 microcontrolador baseado no ATMEL ATmega2560 modelo compatível Arduino Mega 2560;
- 1 *motorshield* Arduino L293D Driver Ponte H;
- Fonte de tensão modelo ATX 12 VDC e 20A;
- *Software* LabVIEW 2020 versão do estudante;
- *Software* Arduino IDE versão 1.8.9 (Windows Store 1.8.21.0);
- Cabos e conexões para interligação dos instrumentos (Cabo Flexível 3 vias 26 AWG);
- Condutivímetro Digimed DM32;
- Conversor R232 para USB.

A Figura 44 mostra a representação esquemática do reator tubular.

Figura 44. Representação esquemática do reator Tubular.



Fonte: Autor.

6.3.3 Custo estimado

A Tabela 12 apresenta os elementos que compõem a unidade experimental do reator PFR com o custo estimado. O cálculo dos preços foi feito com a cotação de dólar a R\$3,30 (cotação do período de compras de materiais para este trabalho - março de 2018).

Tabela 12. Custo estimado a Unidade experimental do PFR.

Elementos - Reator PFR	QTD	Valor Unit.	Valor (R\$)
Minibombas 12 VDC modelo RS485 vazão máxima de 2 L/min;	2	R\$ 39,80	R\$ 79,60
Sensor de efeito Hall modelo YF-S402 vazão máxima de 6 L/min;	2	R\$ 13,22	R\$ 26,44
Condutivímetro Digimed DM32;	1	R\$ 3.336,00	R\$ 3.336,00
Conversor R232 para USB;	1	R\$ 16,90	R\$ 16,90
Microcontrolador baseado no ATMEL ATmega2560 modelo Arduino Mega 2560;	1	R\$ 76,90	R\$ 76,90
Mangueira cristal 1 polegada (metro)	8	R\$ 12,50	R\$ 100,00
Tanque de 30L para armazenamento das substâncias;	2	R\$ 24,00	R\$ 48,00
Cabos e conexões para interligação dos instrumentos (Cabo Flexível 3 vias 26 AWG);	10	R\$ 0,80	R\$ 8,00
<i>Software</i> LabVIEW versão 2020;	1	R\$ 0,00	R\$ 0,00
<i>Software</i> Arduino IDE versão 1.8.9 (Windows Store 1.8.21.0);	1	R\$ 0,00	R\$ 0,00
Motor Shield Arduino L293D Driver Ponte H	1	R\$ 19,90	R\$ 19,90
Fonte de tensão modelo ATX 12 VDC e 20A;	1	R\$ 79,00	R\$ 79,00
Tubulações (mangueira de silicone Di=8mm) - (metros)	4	R\$ 22,50	R\$ 90,00
Total			R\$ 3.880,74

CAPÍTULO 7

7 DISTRIBUIÇÃO DE TEMPO DE RESIDÊNCIA DOS REATORES CSTR E PFR

7.1 Introdução

A distribuição de tempo de resistência (DTR) estuda o desempenho do reator através da maneira com que o fluido escoar dentro de um determinado dispositivo, ou seja, o tempo que cada porção de fluido permanece dentro de um dispositivo e nesse caso pode ser tanto um reator de mistura quanto um reator tubular.

Análises teóricas fazem suposições de idealidade do escoamento das espécies químicas dentro do reator, isto é, escoamento empistonado para o PFR e mistura perfeita para o CSTR. Contudo, em reatores reais. Portanto, devido a importância de conhecer o real comportamento do fluido para dimensionar equipamentos e para avaliar projetos com clareza, deve-se buscar maneiras para medir a DTR de modo experimental. Nesse trabalho foi feito estudo da distribuição do tempo de residência utilizando a técnica estímulo-resposta.

7.2 Fundamentação teórica

7.2.1 Distribuição de tempo de residência

O tempo de residência de um elemento de fluido é o intervalo de tempo em que ele permanece no interior do dispositivo (reatores, nesse caso), desde a entrada até a saída. Sabendo que há diversos possíveis caminhos a serem percorridos por cada elemento de fluido, conclui-se que terão diferentes tempos de resistência dentro do reservatório.

A distribuição dos tempos de residência das partículas de fluido saindo do sistema, $E(t)$, é calculada conforme a Equação 7.1. (LEVENSPIEL, 2000)

$$E(t) = \frac{C(t)}{\int_0^{\infty} C(t)dt} \quad (7.1)$$

Em que:

$C(t)$: concentração de saída do traçador no instante t .

Para sistemas com escoamento ideal, pode-se determinar a função da distribuição dos tempos de residência sem necessidade de trabalhos experimentais, dentre esses sistemas estão o reator tubular (PFR) e o reator de mistura perfeita (CSTR). (HILL, 1977)

O reator tubular (PFR – *Plug Flow Reactor*) tem perfil de velocidade plano e sem mistura longitudinal. Seguindo essas idealidades, todos os elementos do fluido deixam o reator com a mesma idade (\bar{t}). A partir disso, a $E(t)$ para esse sistema é calculada conforme a Equação 7.2 (HILL, 1977):

$$\begin{aligned} E(t) &= 0 \text{ for } 0 < t < \bar{t} \\ E(t) &= 1 \text{ for } t > \bar{t} \end{aligned} \quad (7.2)$$

Para um reator de mistura perfeita (CSTR - *Continuous Stirred Tank Reactor*), assume-se que imediatamente após a entrada de um elemento de volume no sistema, é possível encontrá-lo em qualquer porção do reator. Dessa forma, qualquer elemento de fluido possui uma chance igual de sair do reator em um curto espaço de tempo. Com base nisso, a resposta de um CSTR ideal para uma entrada ideal no sistema é calculada conforme a equação 7.3:

$$E(t) \frac{dE(t)}{dt} = \frac{e^{-t/\bar{t}}}{\bar{t}} \quad (7.3)$$

Para reatores CSTR, há também um outro modelo frequentemente utilizado para simular e comparar reatores reais com ideias. O modelo teórico da curva da distribuição tempo de residência para uma bateria de n -CSTR simula que um reator ideal é substituído por n reatores de mistura perfeita idênticos, que somados tem o mesmo volume que o reator real. Um determinado valor de n irá fornecer melhor resposta para o reator real sendo estudado. A partir disso, tem-se a equação 7.4 que descreve a $E(t)$ para esse modelo (HILL, 1977):

$$E(t) = 1 - e^{-nt/\bar{t}} \left[1 + \frac{nt}{\bar{t}} + \frac{1}{2!} \left(\frac{nt}{\bar{t}} \right)^2 + \dots + \frac{1}{(n-1)!} \left(\frac{nt}{\bar{t}} \right)^{n-1} \right] \quad (7.4)$$

7.2.2 Tempo médio de residência

Sendo θ , o tempo médio teórico de residência do fluido dentro do reator, tem-se:

$$\theta = \frac{V}{Q} \quad (7.5)$$

Em que:

V: volume total do reator;

Q: vazão do reator (sendo que as vazões de entrada e saída são iguais, ou seja, volume constante).

7.2.3 Técnica estímulo-resposta

A distribuição do tempo de residência do fluido escoando pode ser facilmente e diretamente determinada pelo método de investigação muito utilizado, o método experimental estímulo-resposta (LEVENSPIEL, 2000).

A técnica estímulo-resposta é comumente usada para determinar e estudar DTR de um fluido em um reator, a prática experimental em que um traçador é injetado no ponto de entrada do equipamento, podendo ser de duas formas: *off-line* ou *on-line*.

Na primeira maneira é feita a coleta de volumes definidos do fluido na saída dos reatores para determinar a concentração média do traçador. Já o método *on-line*, utilizada nesse trabalho, consiste em obter um número maior de pontos para compor a curva de DTR e consomem menor tempo para consolidação dos dados. (LEE S. W., HANNA M.A., JONES D.D., 2009)

Alguns traçadores são amplamente utilizados em experimentos de determinação de DTR, entre eles estão os corantes, soluções salinas e compostos radioativos (RAO & LONCIN, 1974) e podem ser injetados de diferentes formas: aleatória, pulso, degrau e periódica. Para seguir a técnica estímulo-resposta foi utilizada uma solução salina como traçador e medição da condutividade elétrica foi usada para determinar a curva da DTR.

7.2.4 DTR experimental

Por meio da equação 7.1 calcula-se a distribuição dos tempos de residência das partículas do fluido que saem do sistema utilizando a função $C(t)$ referente a concentração de traçador na saída do reator. Há outras maneiras experimentais de medir a quantidade de traçador, nesse caso foi utilizada a condutividade elétrica como variável na saída do sistema.

Através da técnica experimental estímulo-resposta para determinar a curva de residência e usando como traçador uma substância salina (NaCl), tem-se a função resposta abaixo (HILL, 1977):

$$E(t)_{Exp} = \frac{\lambda - \lambda_0}{\lambda_\infty - \lambda_0} \quad (7.6)$$

Em que:

λ : condutividade elétrica no tempo t ;

λ_∞ : condutividade elétrica medida após um intervalo de tempo suficientemente elevado);

λ_0 : condutividade elétrica inicial (valor obtido no primeiro instante de tempo).

7.3 Materiais

Para a determinação da distribuição do tempo de residência foram utilizadas as unidades experimentais dos reatores de mistura e do reator tubular detalhadas no Capítulo 6. Além disso, os materiais seguintes também foram utilizados:

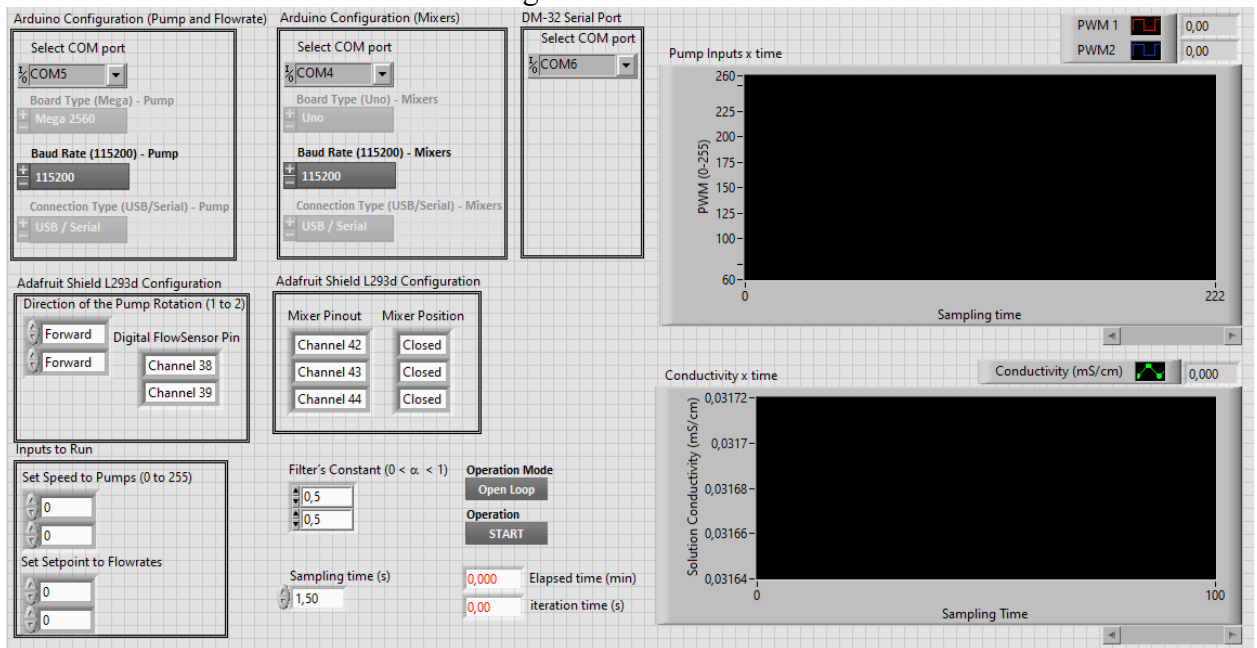
- Água;
- Solução de NaCl a 0,1mol/L.

7.4 Interface implementada para obtenção da DTR

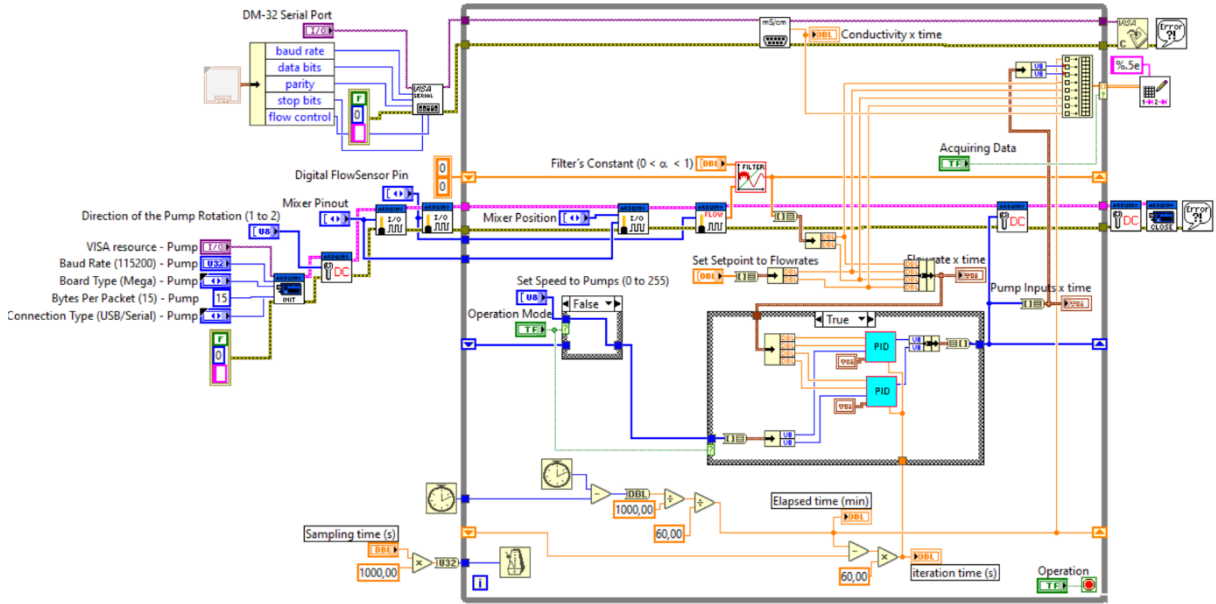
7.4.1 Bateria de reatores CSTR

Para a obtenção da DTR da série de três reatores CSTR utilizou-se a unidade experimental previamente mostrada no Capítulo 6. Foi implementada uma interface especificamente para adquirir dados da DTR. A Figura 45 mostra o painel frontal e o diagrama de blocos do *script* LabVIEW desenvolvido.

Figura 45. *Script* LabVIEW para a obtenção da DTR para os CSTRs. a) - Painel frontal e (b) – Diagrama de blocos.



(a)



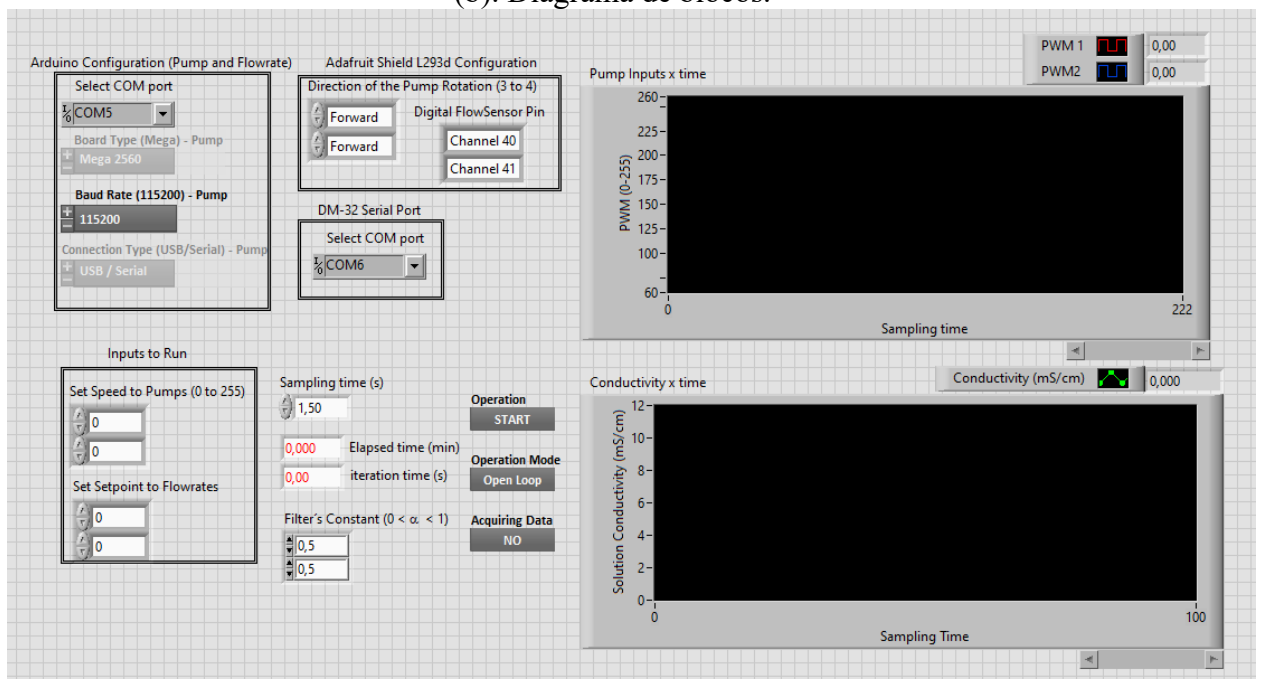
(b)

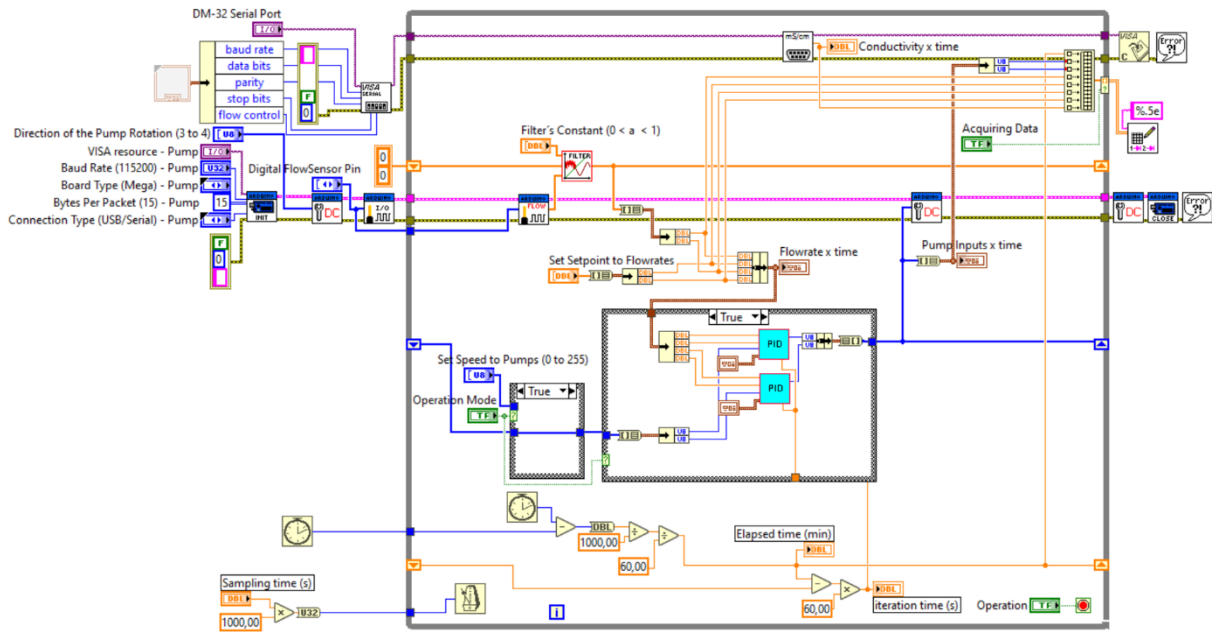
Fonte: Autor.

7.4.2 Reator PFR

Para a obtenção da DTR do reator tubular fez-se uso da unidade experimental previamente demonstrada no Capítulo 6. Foi implementada uma interface especificamente para a aquisição dos dados da DTR, a Figura 46 mostra o painel frontal do *script* LabVIEW desenvolvido.

Figura 46. *Script* LabVIEW para a obtenção da DTR para o reator PFR. (a). Painel frontal e (b). Diagrama de blocos.





(b)
Fonte: Autor.

7.5 Resultados e Discussão

7.5.1 Cálculo do Tempo Médio de Residência Experimental para a bateria de CSTR

O tempo médio de residência (θ) da bateria de CSTR foi calculado realizando um ajuste por mínimos quadrados da expressão da distribuição tempo de residência de uma bateria de três reatores CSTR em série aos dados experimentais de cada corrida.

$$\min_{\theta} L = \sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (E_i(\theta) - (E_i)_{exp})^2 \quad (7.7)$$

Em que $E(t, \theta)$ é dada por:

$$E(t, \theta) = \frac{C_n}{C_0} = 1 - e^{-3t/\theta} \left[1 + \frac{nt}{\theta} + \frac{1}{2!} \left(\frac{nt}{\theta} \right)^2 \right] \quad (7.8)$$

L é a função objetivo do método de mínimos quadrados e ε é desvio dos dados experimentais à curva ajustada. Dessa forma, um problema de otimização não linear foi resolvido no parâmetro θ , utilizando o pacote *solver* do Excel 2019 com o método do gradiente reduzido generalizado (GRG). Este método foi desenvolvido por Wolfe (1963) e posteriormente foi generalizado por Carpentier e Abadie (1969). As Tabelas 13, 14, 15 e 16 apresentam os dados experimentais e ajustados. As Figuras 47, 48, 49 e 50 mostram graficamente a DTR experimental e a ajustada com o valor do tempo médio de residência experimental calculado

para vazão de 400 mL/min, 600 mL/min, 800 mL/min e 1000 mL/min de cada reagente (NaCl e água).

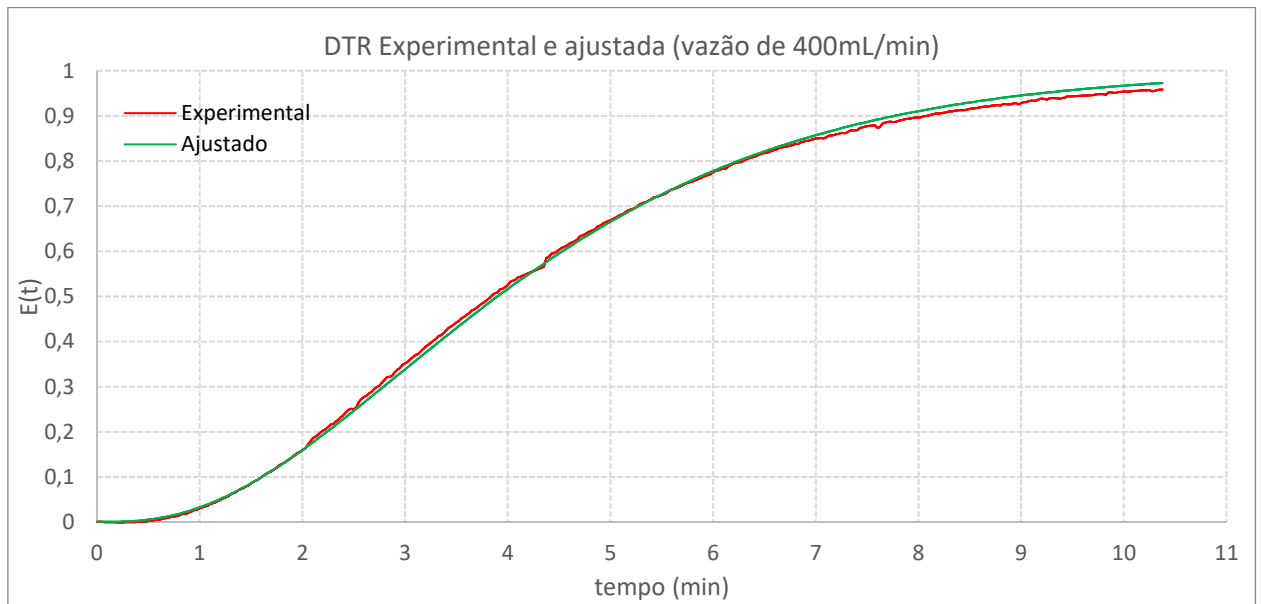
Na obtenção da $E(t)_{Exp}$ foi utilizada a Equação 7.6, a qual necessita dos valores de de condutividade elétrica inicial (λ_0) e infinita (λ_∞). Estes valores foram determinados experimentalmente. Os valores das condutividades inicial λ_0 e λ_∞ foram obtidos extrapolando os valores experimentais de condutividade em função do tempo (λ versus tempo) medidos experimentalmente para os tempos $t \rightarrow 0$ e $t \rightarrow \infty$, respectivamente. O tempo $t \rightarrow \infty$ foi determinando quando observou-se que as medidas experimentais de $\lambda(t)$ não mais apresentavam variação com o tempo, tornando-se constante. Tomou-se, então, esse valor como λ_∞ . Os valores de λ_0 e λ_∞ foram os mesmos para todas as corridas, uma vez a razão das duas vazões de entrada nos sistemas de reação (reatores tanque e tubular) foi mantida constante, aumentado-se ambas vazões na mesma proporção de corrida para corrida. Dessa forma os valores de λ_0 e λ_∞ permaneceram os mesmos em todas as corridas. Os valores obtidos experimentalmente foram $\lambda_0 = 0,346$ mS/cm e $\lambda_\infty = 11,660$ mS/cm.

Tabela 13. Dados de DTR experimental e ajustada para vazão de 400 mL/min.

tempo (min)	$E(t)_{Exp}$	$E(t)_{Ajuste}$	Desvio ²
0,00E+00	0,00E+00	0,00E+00	0,00E+00
2,50E-02	1,15E-04	8,25E-07	1,30E-08
5,00E-02	4,41E-04	6,52E-06	1,89E-07
⋮	⋮	⋮	⋮
7,40E+00	8,68E-01	8,81E-01	1,68E-04
7,42E+00	8,72E-01	8,83E-01	1,12E-04
7,45E+00	8,74E-01	8,84E-01	9,74E-05
7,47E+00	8,76E-01	8,85E-01	8,38E-05
7,50E+00	8,77E-01	8,87E-01	8,24E-05
7,52E+00	8,78E-01	8,88E-01	1,00E-04
7,55E+00	8,79E-01	8,89E-01	1,05E-04
7,57E+00	8,79E-01	8,90E-01	1,27E-04
7,60E+00	8,74E-01	8,92E-01	3,33E-04
7,62E+00	8,75E-01	8,93E-01	3,14E-04
7,65E+00	8,83E-01	8,94E-01	1,37E-04
⋮	⋮	⋮	⋮
1,02E+01	9,57E-01	9,70E-01	1,84E-04
1,02E+01	9,57E-01	9,70E-01	1,87E-04

1,02E+01	9,57E-01	9,71E-01	1,79E-04
1,03E+01	9,55E-01	9,71E-01	2,81E-04
1,03E+01	9,56E-01	9,72E-01	2,58E-04
1,03E+01	9,57E-01	9,72E-01	2,13E-04
1,03E+01	9,58E-01	9,72E-01	2,12E-04
1,04E+01	9,58E-01	9,73E-01	2,13E-04

Figura 47. Distribuição tempo de residência experimental e ajustada para vazão de 400 mL/min



Fonte: Autor.

Tabela 14. Dados de DTR experimental e ajustada para vazão de 600 mL/min.

tempo (min)	$E(t)_{Exp}$	$E(t)_{Ajuste}$	Desvio ²
0,00E+00	0,00E+00	0,00E+00	0,00E+00
2,50E-02	-2,92E-04	2,82E-06	8,67E-08
5,00E-02	-9,17E-04	2,21E-05	8,83E-07
⋮	⋮	⋮	⋮
4,03E+00	7,84E-01	7,84E-01	1,64E-07
4,05E+00	7,88E-01	7,88E-01	1,50E-07
4,08E+00	7,89E-01	7,91E-01	4,26E-06
4,10E+00	7,93E-01	7,94E-01	2,80E-06
4,13E+00	7,96E-01	7,98E-01	4,43E-06
4,15E+00	8,00E-01	8,01E-01	7,40E-07
4,18E+00	8,03E-01	8,04E-01	2,37E-06
4,20E+00	8,07E-01	8,08E-01	9,67E-07
4,23E+00	8,09E-01	8,11E-01	1,95E-06
4,25E+00	8,09E-01	8,14E-01	2,19E-05

4,28E+00	8,15E-01	8,17E-01	4,13E-06
⋮	⋮	⋮	⋮
8,13E+00	9,83E-01	9,90E-01	5,28E-05
8,15E+00	9,82E-01	9,90E-01	5,87E-05
8,17E+00	9,82E-01	9,90E-01	6,93E-05
8,20E+00	9,83E-01	9,91E-01	5,72E-05
8,22E+00	9,84E-01	9,91E-01	4,37E-05
8,25E+00	9,85E-01	9,91E-01	3,86E-05
8,28E+00	9,85E-01	9,91E-01	4,30E-05
8,30E+00	9,84E-01	9,91E-01	5,81E-05

Figura 48. Distribuição tempo de residência experimental e ajustada para vazão de 600 mL/min



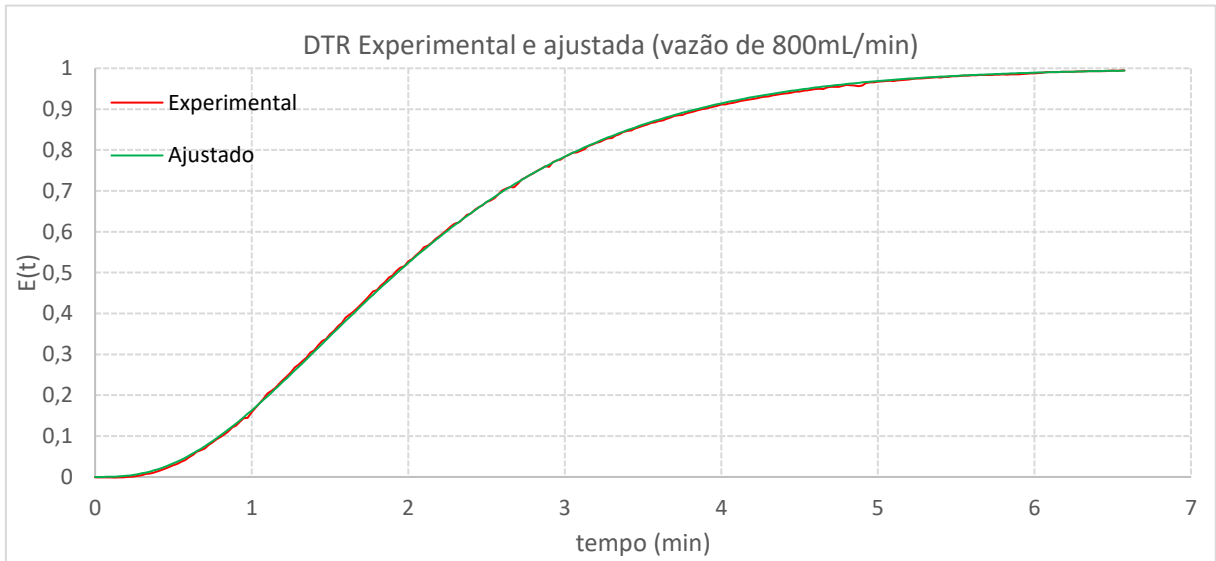
Fonte: Autor.

Tabela 15. Dados de DTR experimental e ajustada para vazão de 800 mL/min.

tempo (min)	$E(t)_{Exp}$	$E(t)_{Ajuste}$	Desvio ²
0,00E+00	-9,79E-04	0,00E+00	9,59E-07
2,50E-02	-1,31E-03	6,74E-06	1,73E-06
5,00E-02	-7,94E-04	5,26E-05	7,16E-07
⋮	⋮	⋮	⋮
3,15E+00	8,09E-01	8,10E-01	1,05E-06
3,18E+00	8,13E-01	8,15E-01	3,87E-06
3,20E+00	8,17E-01	8,19E-01	1,72E-06
3,23E+00	8,20E-01	8,23E-01	9,87E-06
3,25E+00	8,25E-01	8,27E-01	3,64E-06

3,28E+00	8,28E-01	8,30E-01	4,88E-06
3,30E+00	8,30E-01	8,34E-01	2,27E-05
3,33E+00	8,35E-01	8,38E-01	7,75E-06
3,35E+00	8,39E-01	8,42E-01	6,49E-06
3,38E+00	8,44E-01	8,45E-01	1,84E-06
3,40E+00	8,47E-01	8,49E-01	3,74E-06
⋮	⋮	⋮	⋮
6,40E+00	9,93E-01	9,93E-01	6,76E-08
6,43E+00	9,93E-01	9,93E-01	1,52E-08
6,45E+00	9,93E-01	9,93E-01	3,75E-09
6,48E+00	9,94E-01	9,94E-01	5,59E-08
6,50E+00	9,94E-01	9,94E-01	2,90E-07
6,53E+00	9,94E-01	9,94E-01	4,39E-08
6,55E+00	9,95E-01	9,94E-01	2,36E-07
6,58E+00	9,95E-01	9,94E-01	2,51E-07

Figura 49. Distribuição tempo de residência experimental e ajustada para vazão de 800 mL/min



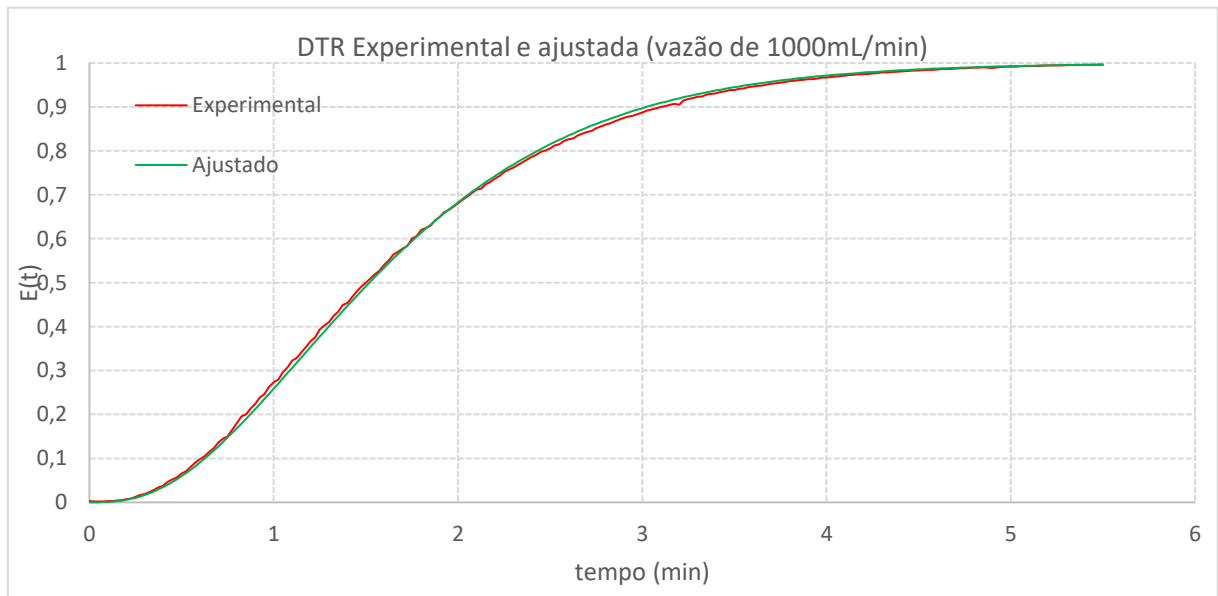
Fonte: Autor.

Tabela 16. Dados de DTR experimental e ajustada para vazão de 1000 mL/min.

tempo (min)	$E(t)_{Exp}$	$E(t)_{Ajuste}$	Desvio ²
0,00E+00	2,72E-03	0,00E+00	7,38E-06
2,50E-02	2,17E-03	1,37E-05	4,67E-06
5,00E-02	1,99E-03	1,06E-04	3,55E-06
⋮	⋮	⋮	⋮
2,50E+00	8,05E-01	8,15E-01	8,84E-05

2,52E+00	8,12E-01	8,20E-01	6,47E-05
2,55E+00	8,15E-01	8,25E-01	9,81E-05
2,58E+00	8,22E-01	8,30E-01	5,86E-05
2,60E+00	8,26E-01	8,35E-01	7,95E-05
2,63E+00	8,28E-01	8,39E-01	1,24E-04
2,65E+00	8,35E-01	8,44E-01	8,16E-05
2,67E+00	8,39E-01	8,49E-01	9,21E-05
2,70E+00	8,43E-01	8,53E-01	9,48E-05
2,73E+00	8,45E-01	8,57E-01	1,38E-04
2,75E+00	8,51E-01	8,61E-01	9,62E-05
⋮	⋮	⋮	⋮
5,33E+00	9,95E-01	9,95E-01	3,56E-07
5,35E+00	9,95E-01	9,96E-01	5,75E-07
5,38E+00	9,95E-01	9,96E-01	8,36E-07
5,40E+00	9,96E-01	9,96E-01	7,27E-08
5,43E+00	9,96E-01	9,96E-01	5,69E-08
5,45E+00	9,96E-01	9,96E-01	4,09E-08
5,48E+00	9,96E-01	9,96E-01	4,60E-08
5,50E+00	9,96E-01	9,96E-01	1,08E-07

Figura 50. Distribuição tempo de residência experimental e ajustada para vazão de 1000 mL/min



Fonte: Autor.

Como já explicado anteriormente, foram feitos os cálculos de tempo médio de residência experimental e teórico para cada vazão, além do cálculo de desvio. Os resultados estão apresentados na Tabela 17.

Tabela 17. Valores de tempo médio de residência experimental e teórico para cada vazão para a bateria de reatores CSTR.

Q_{total} (mL/min)	$\theta_{\text{Teórico}}$ (min)	θ_{Exp} (min)	Desvio (%)
400	4,569	4,378	4,362
600	3,046	2,904	4,906
800	2,285	2,166	5,491
1000	1,828	1,705	7,229

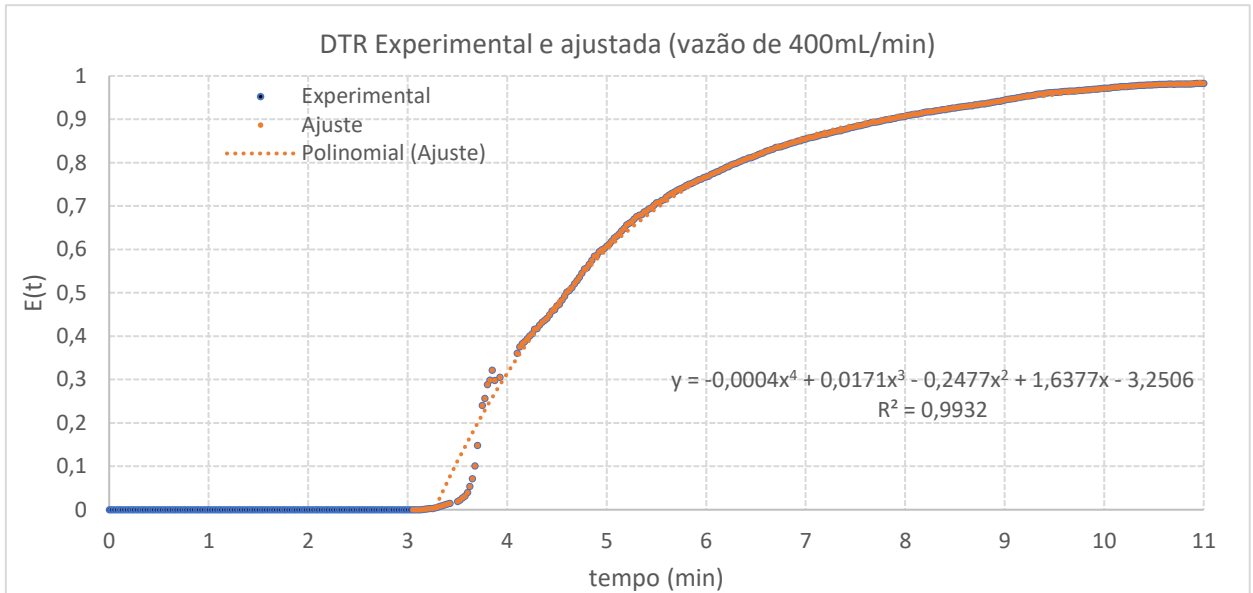
Considerando os valores de desvio apresentados na Tabela 17, pode-se assegurar que a bateria de reatores tanque se comportou satisfatoriamente como uma bateria de reatores CSTR, isto é, com escoamento bastante próximo do escoamento ideal. Nesse caso, ficou evidente que a medida que a vazão de entrada da bateria aumenta, o tempo de residência experimental se distancia do teórico. Essa relação pode ser explicada pela forma com que o kit experimental foi projetado e montado, onde o aparecimento de bolhas no interior mostrou-se mais intenso quando a vazão aumenta.

7.5.2 Cálculo do Tempo Médio de Residência Experimental para o PFR

Para a obtenção da $E(t)_{\text{Exp}}$ do reator tubular foi utilizada a Equação 7.6, assim como explicado anteriormente na para a bateria de reatores tanque. Além disso, no caso de reator tubular foi feito um ajuste polinomial da DTR experimental considerando o grau do polinômio em que o coeficiente de correlação (R^2) foi mais próximo da unidade.

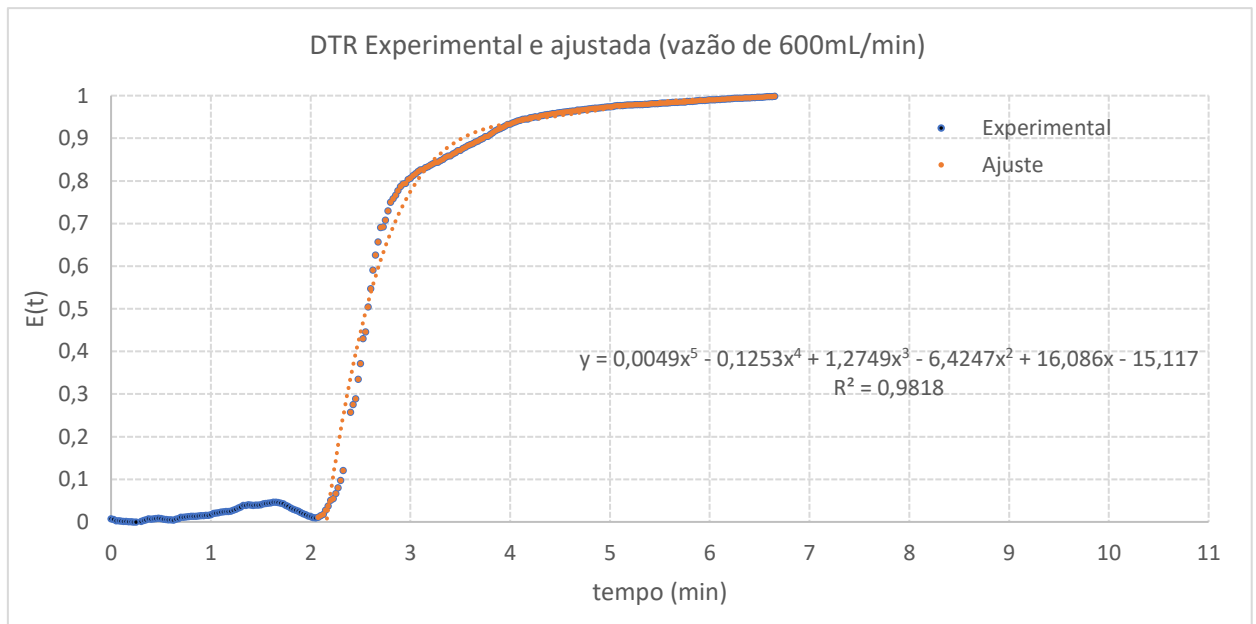
Como citado anteriormente, para utilizar a equação de obtenção de DTR experimentalmente é necessário obter os valores de condutividade inicial e finita. Seguiu-se o mesmo procedimento explicado para a bateria de reatores tanque. Mesmo sendo a mesma concentração, foram feitas as medidas da condutividade inicial para cada corrida, com o intuito de diminuir os possíveis erros experimentais (pesagem, preparação da solução e variação de temperatura). As curvas de distribuição de tempo de residência para cada vazão estão apresentadas a seguir, juntamente com a curva ajustada e o polinômio obtido (Figuras 51, 52, 53 e 54).

Figura 51. Distribuição tempo de residência experimental e ajustada para a vazão de 400 mL/min ($\lambda_0 = 0,0448\text{mS/cm}$ e $\lambda_\infty = 11,433\text{mS/cm}$)



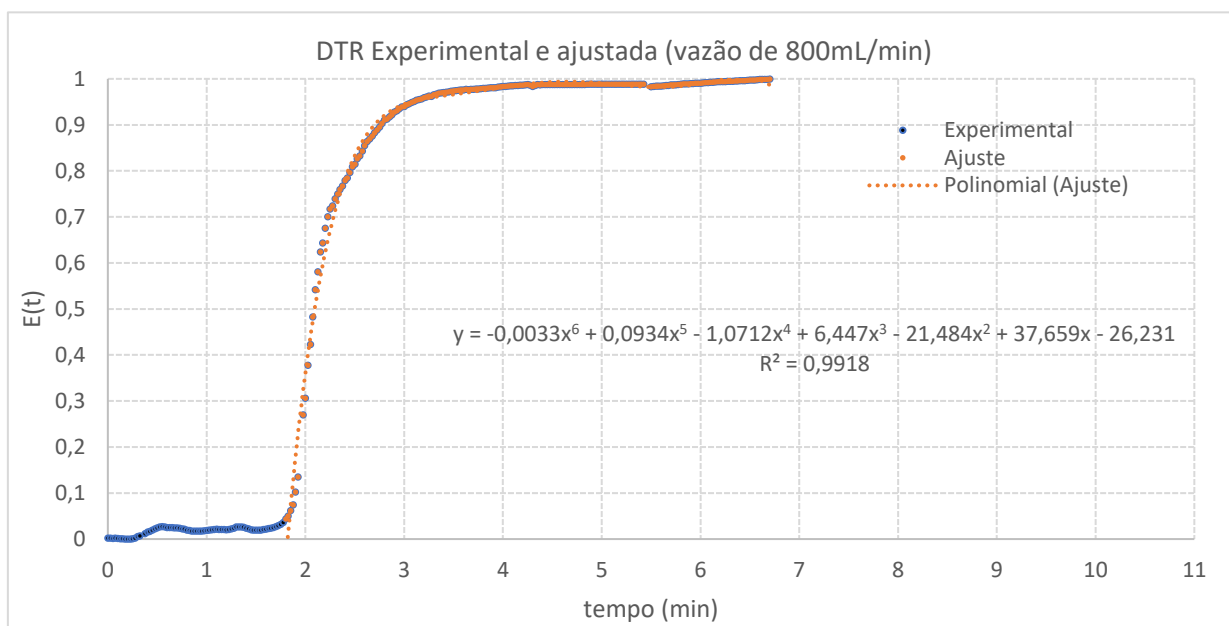
Fonte: Autor.

Figura 52. Distribuição tempo de residência experimental e ajustada para a vazão de 600 mL/min ($\lambda_0 = 0,580\text{mS/cm}$ e $\lambda_\infty = 11,433\text{mS/cm}$)



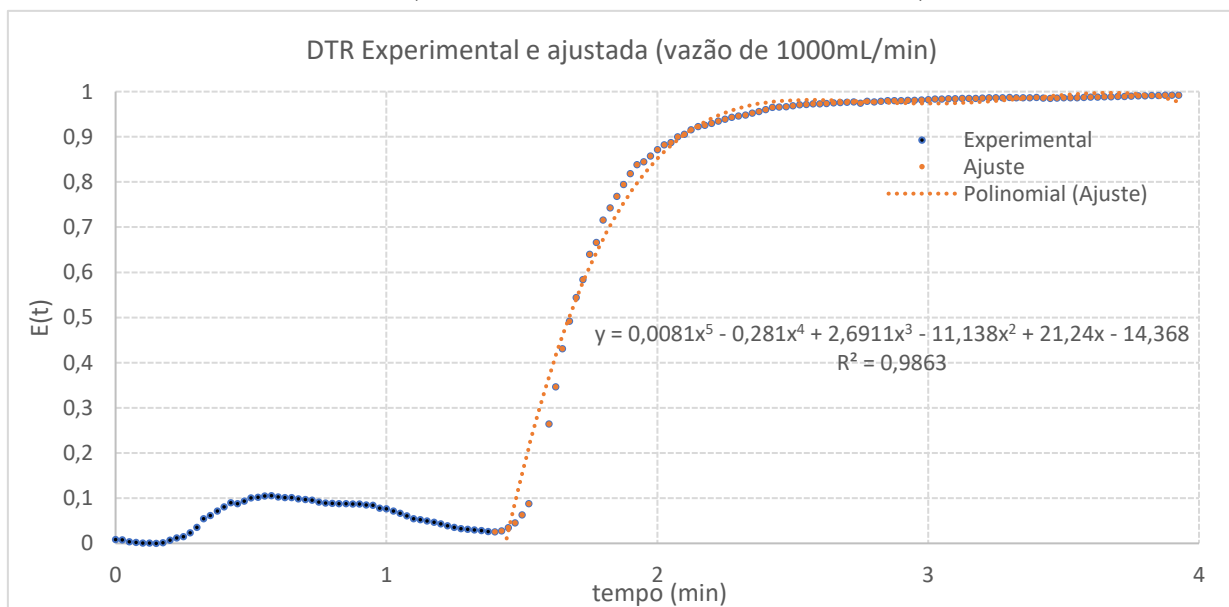
Fonte: Autor.

Figura 53. Distribuição tempo de residência experimental e ajustada para a vazão de 800 mL/min ($\lambda_0 = 0,119\text{mS/cm}$ e $\lambda_\infty = 11,433\text{mS/cm}$)



Fonte: Autor.

Figura 54. Distribuição tempo de residência experimental e ajustada para a vazão de 1000 mL/min ($\lambda_0 = 0,513\text{mS/cm}$ e $\lambda_\infty = 11,433\text{mS/cm}$)



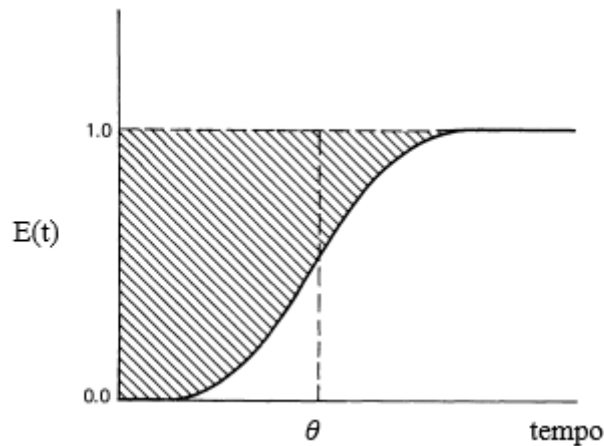
Fonte: Autor.

A determinação do tempo de residência médio experimental do reator PFR foi feita através da função de distribuição de tempo de residência proposto por Hill (1977). Logo, tem-se a Equação 7.9:

$$\theta = \int_{E(t)=0}^{E(t)=1} t dE(t) \quad (7.9)$$

A partir desse conceito e da curva de $t \times E(t)$, sabe-se que a área sombreada é o tempo médio de residência (θ), como mostra a Figura 55.

Figura 55: Determinação do tempo médio de residência por meio da função de distribuição de tempo de residência.



Fonte: (HILL, 1977)

Com posse do polinômio ajudado para cada curva, o cálculo do tempo médio de residência do PFR a partir dos dados experimentais foi feito através do conceito de área sombreada mostrado acima. Sabe-se que a função $E(t)$ tem limite superior de 1 e conhecendo o tempo final de passagem do traçador, é possível calcular a área total do gráfico como um retângulo. Ao subtrair a área abaixo da curva desse retângulo, tem-se a área sombreada.

$$\theta = \text{Área Retângulo} - \text{Área abaixo da curva} \quad (7.10)$$

$$\theta = E(t_{final}) \times t_{final} - \int_{t_{inicial}}^{t_{final}} E(t) dt \quad (7.11)$$

Em que:

t_{final} : tempo final de passagem de traçador, quando $E(t) = 1$;

$t_{inicial}$: tempo inicial;

Os resultados dos cálculos de tempo médio de residência experimental e teórico para cada vazão e o cálculo de desvio estão apresentados na Tabela 18.

Tabela 18. Valores de tempo médio de residência experimental e teórico para cada vazão para o reator PFR.

Q_{total} (mL/min)	θ Teórico (min)	θ Exp (min)	Desvio (%)
400	4,400	5,154	14,62
600	2,933	2,783	5,39
800	2,285	2,200	2,25
1000	1,760	1,761	0,05

A partir dos resultados apresentados é possível concluir que à medida que se aumenta a vazão no reator tubular maior a proximidade do tempo médio de residência experimental com teórico. Isso se deve às características de escoamento do fluido, isto é, com uma vazão maior o escoamento se aproxima cada vez mais do escoamento empistonado, onde as moléculas passam pelo reator na mesma velocidade. Já em casos em que a vazão é baixa como 400mL/min, o perfil parabólico de velocidade característico de escoamentos laminares em tubulações e/ou o efeito da dispersão axial do soluto (NaCl) parecem ser significativos.

Através do número de Reynolds é possível caracterizar qual o tipo de escoamento em cada uma das vazões (laminar, transiente ou turbulento), a fim de entender a diferença de desvios em cada situação. O número de Reynolds é adimensional e segue a Equação 7.12. Nesse caso, como todas as outras variáveis presentes no cálculo são iguais, a velocidade do fluido define o tipo de escoamento dentro do reator.

Para um escoamento laminar em um reator tubular, o perfil de velocidade é parabólico e o fluido que escoar pelo centro do reator é aquele que permanece menos tempo no reator (FOGLER, 2002). Isso indica que nesse tipo de escoamento o desvio da condição de escoamento empistonado do PFR será mais significativo.

$$Re = \frac{\rho v d}{\mu} \quad (7.12)$$

Em que:

ρ : massa específica do fluido;

v : velocidade média do fluido;

d : diâmetro para o fluxo no tubo;

μ : viscosidade dinâmica do fluido;

A fim de calcular o número de Reynolds para os casos extremos de vazão, 400mL/min e 1000mL/min, foi considerado a viscosidade dinâmica e a densidade do fluido como a da água pura, já que a solução usada era bastante diluída. A velocidade do fluido foi calculada pela Equação 7.13, em que a área da base do tubo foi citada no Capítulo 6 e Q é a vazão do fluido.

$$v = \frac{Q}{Área_{base-tubo}} \quad (7.13)$$

A baixo tem-se os valores utilizados para o cálculo do número de Reynolds.

$$\rho = 1000kg/m^3$$

$$d = 0,0254 m^2$$

$$\mu = 1,0030 \times 10^{-3} Pa/s$$

A partir do que foi apresentado, calculou-se o Reynolds para a maior e a menor vazão (resultados apresentados a baixo). Conforme apresentado em Brunetti (2007), o escoamento será laminar se $Re < 2000$ e será turbulento para $Re > 2400$, entre esses limites, o escoamento será de transição. Os valores calculados foram:

$$Re (400 mL/min) = 333,20$$

$$Re (1000 mL/min) = 833,09$$

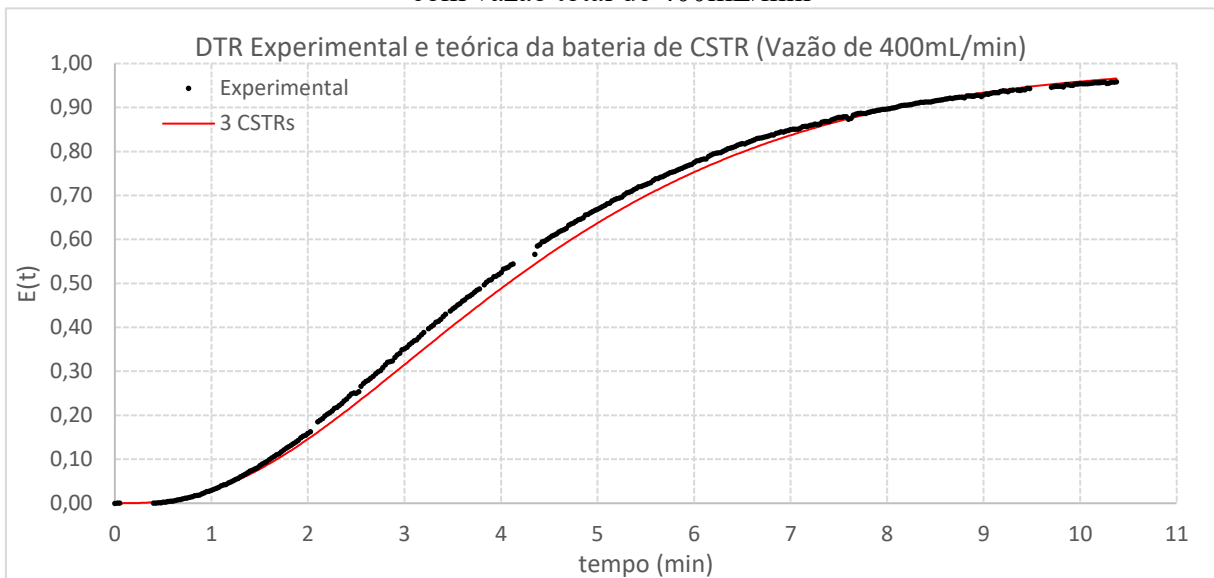
Logo, conclui-se em todas as vazões estudadas o escoamento laminar é o escoamento predominante. Em decorrência disso era de se esperar desvios apreciáveis entre o tempo de residência experimental e o predito pela literatura. Isto foi o ocorrido para a vazão de 400 mL/min com desvio de 14,62%. Contudo, surpreendentemente para as outras vazões os desvios foram pequenos, havendo boa concordância com os valores preditos pela literatura. Uma possível explicação para isso é o fato de que em vazões maiores, mesmo estando em regime laminar, o perfil de velocidades parabólico se achata um pouco mais e se aproxima do perfil empistonado. Veja que quanto maior é a vazão menor é o desvio entre o valor experimental e o teórico, o que corrobora com a justificativa apresentada.

7.5.3 DTR Experimental e teórica da bateria de reatores de mistura ou tanque

Como já explicado anteriormente, utilizando a técnica experimental estímulo-resposta para determinar a curva de tempos de residência e usando como traçador uma substância salina

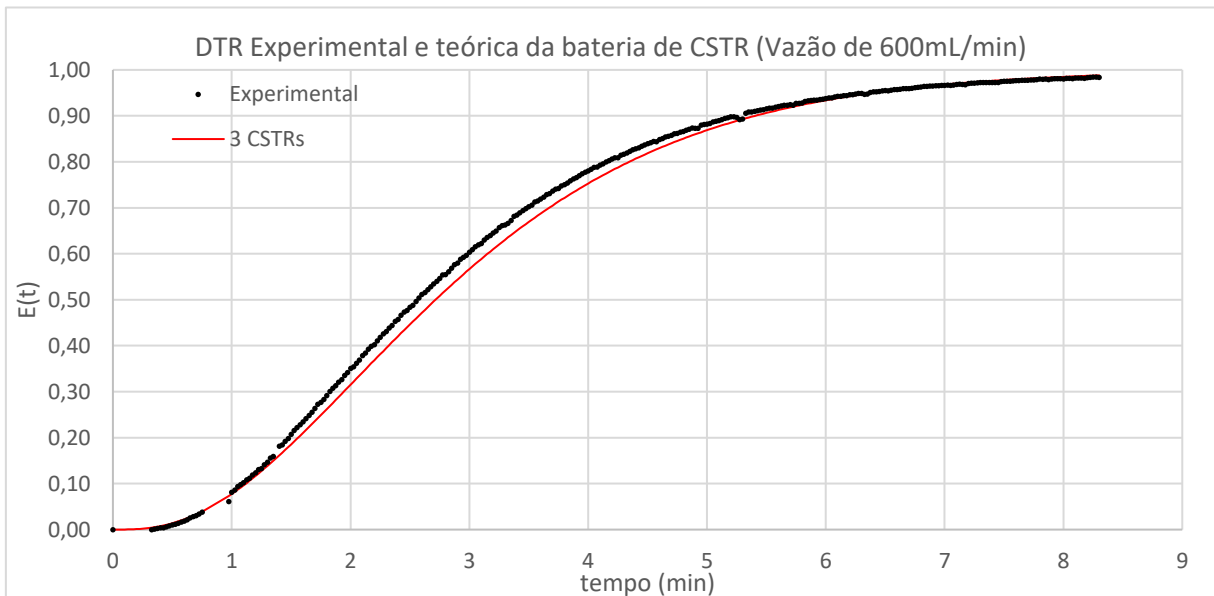
(NaCl) foi determinada a DTR para as quatro vazões já citadas. Os resultados experimentais em comparativo com a DTR teórica estão apresentados nas Figuras 56, 57, 58 e 59.

Figura 56: Distribuição do tempo de residência na bateria de CSTR experimental e teórico com vazão total de 400mL/min



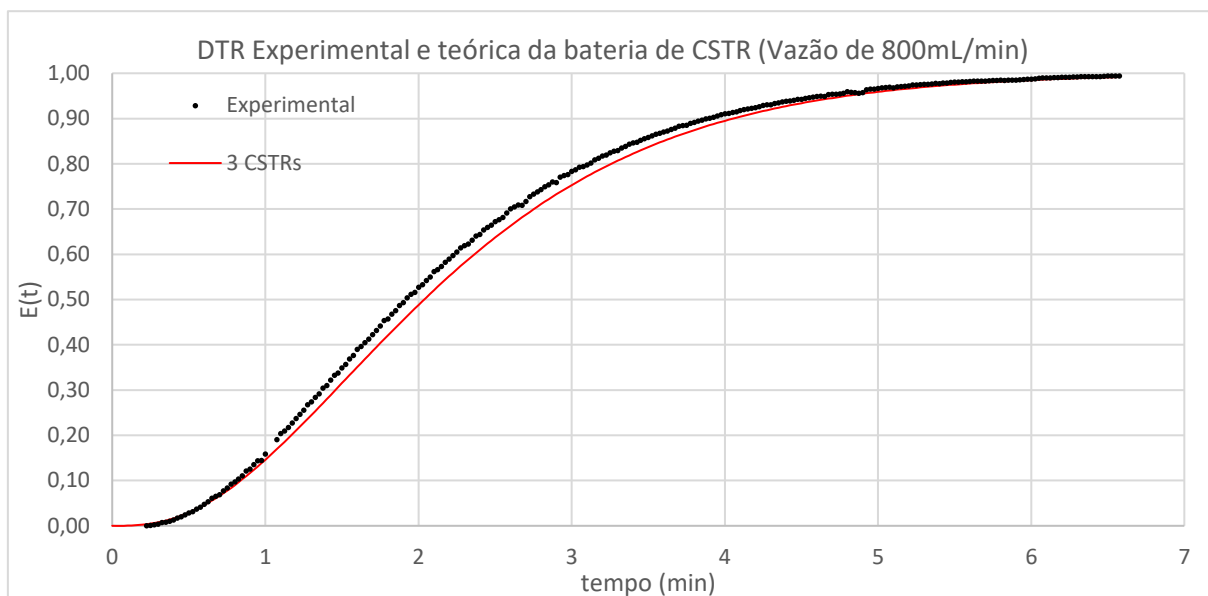
Fonte: Autor.

Figura 57: Distribuição do tempo de residência na bateria de CSTR experimental e teórico com vazão total de 600mL/min.



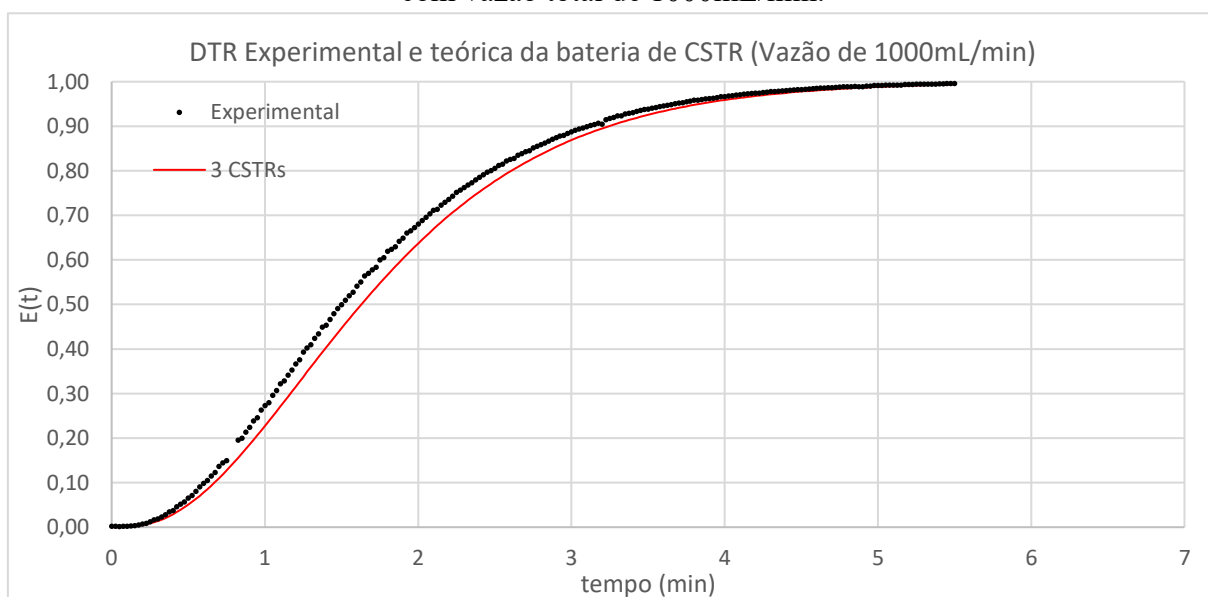
Fonte: Autor.

Figura 58: Distribuição do tempo de residência na bateria de CSTR experimental e teórico com vazão total de 800mL/min.



Fonte: Autor.

Figura 59: Distribuição do tempo de residência na bateria de CSTR experimental e teórico com vazão total de 1000mL/min.



Fonte: Autor.

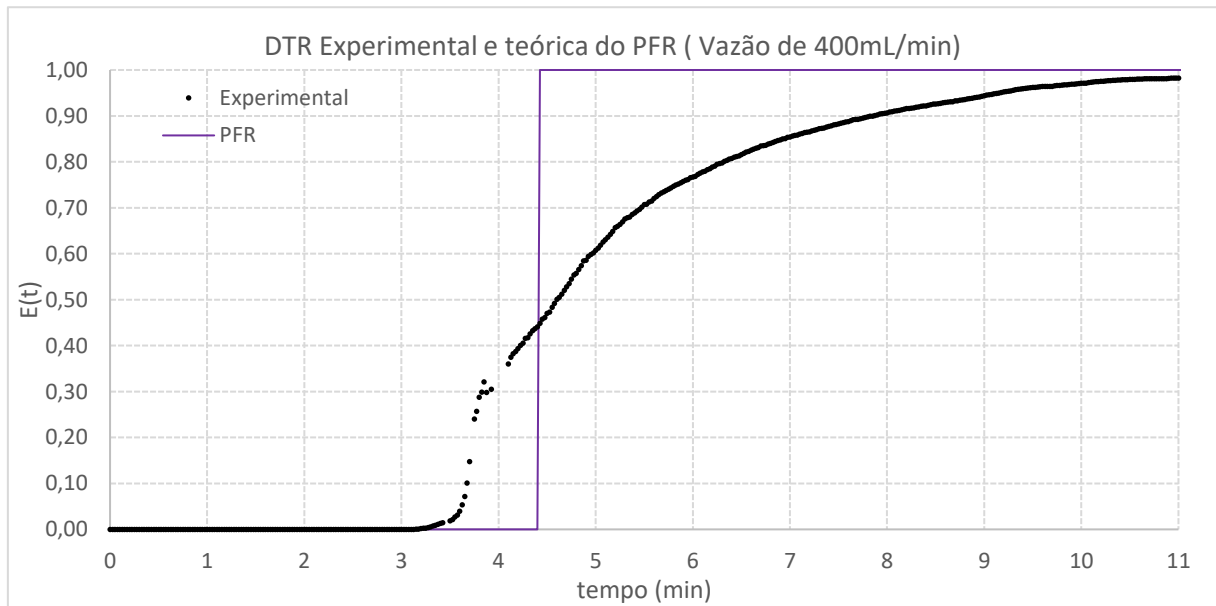
A partir dos resultados apresentados é possível concluir que a bateria de reatores de mistura construídos se comportou com uma bateria de CSTR de mesmo volume, o que comprova que no interior dos reatores a mistura assemelha-se bastante a uma mistura perfeita.

Analisando os gráficos de DTR juntamente com os valores de desvio do tempo médio de residência teórico e experimental, assegura-se que o kit experimental se aproxima significativamente de uma bateria de reatores CSTR.

7.5.4 DTR Experimental e teórico PFR

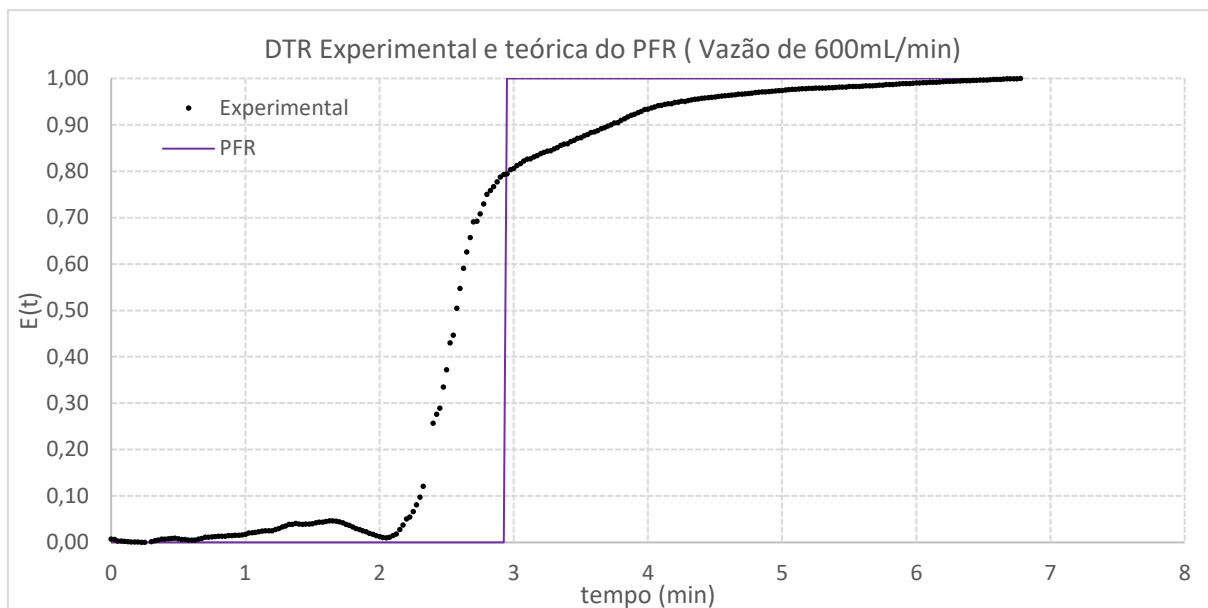
Como já dito anteriormente na sessão de resultados da bateria de CSTR, para o reator PFR também foi utilizado a técnica experimental estímulo-resposta para determinar a curva de tempos de residência. Os resultados experimentais juntamente com a DTR teórica estão apresentados nas Figuras 60, 61, 62 e 63.

Figura 60: Distribuição do tempo de residência no PFR experimental e teórico com vazão total de 400mL/min



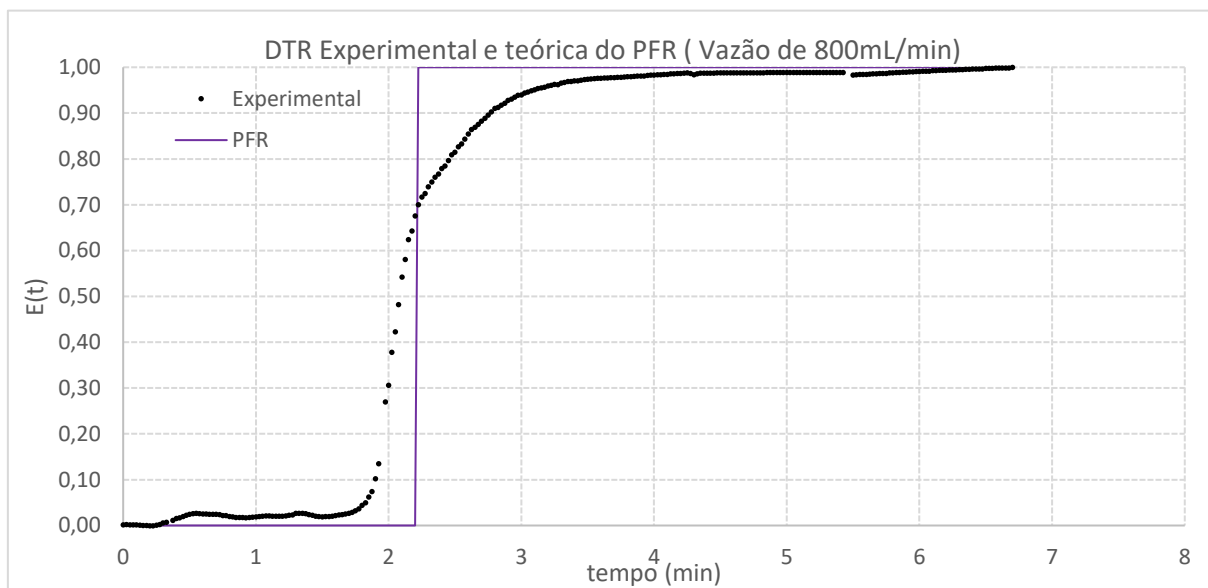
Fonte: Autor.

Figura 61: Distribuição do tempo de residência no PFR experimental e teórico com vazão total de 600mL/min.



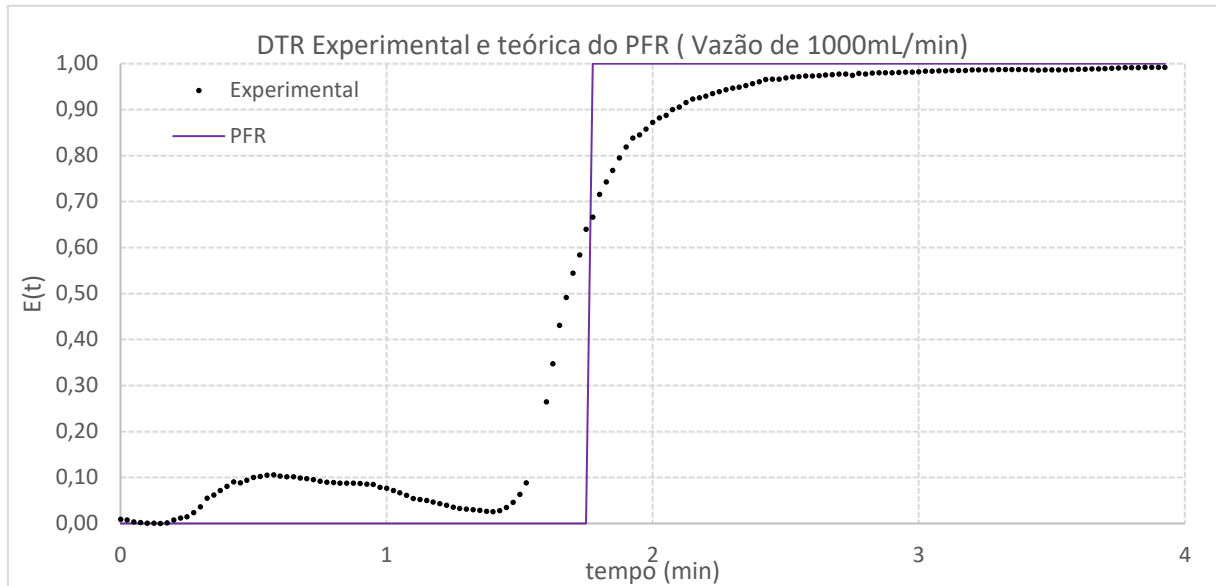
Fonte: Autor.

Figura 62: Distribuição do tempo de residência no PFR experimental e teórico com vazão total de 800mL/min.



Fonte: Autor.

Figura 63: Distribuição do tempo de residência no PFR experimental e teórico com vazão total de 1000mL/min.



Fonte: Autor.

Os resultados obtidos nas corridas do reator tubular demonstram que em vazões maiores a DTR se aproxima do comportamento da DTR teórica de um reator PFR. Porém, na vazão de 400 mL/min (a menor vazão estudada) a curva da distribuição de tempos de residência está mais distante da DTR teórica.

Adicionalmente, conforme apresentado por Hill (1977) e Fogler (2002), pode-se também citar o efeito de difusão radial e longitudinal como fator que contribui para o afastamento da condição ideal de escoamento, podendo inclusive ambos os efeitos estarem presentes.

CAPÍTULO 8

8 DETERMINAÇÃO DA CINÉTICA QUÍMICA DA REAÇÃO DE ACETATO DE ETILA E HIDRÓXIDO DE SÓDIO EM REATOR BATELADA

8.1 Introdução

Como apresentado nos capítulos anteriores foi desenvolvido um aparato experimental de reatores tanque e reator tubular para estudar distribuição de tempo de residência. Com o intuito de explorar também cinéticas de reações químicas, fez-se um estudo da reação entre acetato de etila e hidróxido de sódio. Esta reação foi escolhida pela possibilidade de acompanhar o andamento da reação (grau de avanço) medindo-se a condutividade elétrica da solução, o que torna a plataforma Arduino já desenvolvida útil também para este caso. Foi implementada uma interface específica para adquirir esses dados cinéticos utilizando um reator tipo batelada.

A cinética química é, basicamente, uma descrição matemática da taxa de reação de um processo químico, podendo existir diversas descrições matemáticas para caracterizar as várias situações reais, sendo elas complexas ou simples (BERGER, HUGH STITT, KAPTEIJN, & MOULIJN, 2001).

8.2 Cinética da reação

Alguns trabalhos da literatura (Wijayarathne & Wasalathilake, 2014; Tsujikawa H., Inone, 1966 e Walker, 1906) têm tratado a saponificação de acetato de etila com hidróxido de sódio como de 2ª ordem geral e 1ª ordem em relação aos reagentes. Contudo, outros trabalhos mais recentes têm mostrado que a ordem de reação diminui e se torna sequencial em vez de 2ª ordem quando concentrações equimolares de ambos os reagentes são usadas (Mukhtar et. all., 2017). Estes pesquisadores mostraram que dados experimentais da hidrólise alcalina de acetato de etila são bem ajustados por uma ordem de reação em altas concentrações, mas por outra ordem de reação diferente em baixas concentrações. Mais especificamente, eles demonstraram que esta reação tem uma ordem geral de reação 1,3118 e não pode ser expressa satisfatoriamente como uma reação de 2ª ordem, especialmente quando proporções equimolares de ambos os reagentes são usadas. Eles mostraram também que os dados experimentais se ajustam à uma cinética de 2ª ordem somente quando um dos reagentes está em excesso. Partindo deste ponto,

usou-se aqui neste trabalho concentrações de 0,1 molar de hidróxido de sódio e de acetato de etila, mas com volumes de reagentes de 45 mL e 25 mL de hidróxido de sódio e acetato de etila, respectivamente. Dessa forma, o hidróxido de sódio estará presente em excesso e uma cinética como mostrada na Equação (8.1) pode ser proposta para esta reação:

$$-r_A = kC_A C_B \quad (8.1)$$

Em que k é a constante da taxa e é dada pela Lei de Arrhenius:

$$k = A e^{-E/RT} \quad (8.2)$$

Em que:

A : Fator pré-exponencial ou fator frequência, algumas vezes chamado de k_0 ;

E : Energia de ativação, J/mol ou cal/mol;

R : Constante dos gases, 8,314 J/mol.K ou 1,987 cal/mol.K;

T : Temperatura absoluta, K.

A energia de ativação será determinada experimentalmente realizando a reação em várias temperaturas diferentes. A reação em estudo ocorre a densidade constante uma vez que ela se dá essencialmente em fase líquida. O grau de avanço a densidade constante (ξ^*) (HILL, 1977) da reação será usado para acompanhar o andamento da reação e será determinado pela medida da condutividade da reação, isto é:

$$\xi^* = \xi_\infty^* \frac{(\lambda - \lambda_0)}{(\lambda_\infty - \lambda_0)} \quad (8.5)$$

O valor de ξ_∞^* pode ser facilmente determinado a partir da constante de equilíbrio da reação se a reação for reversível. Se a reação for irreversível, ela pode ser determinada a partir da concentração inicial do reagente limitante, isto é:

$$\xi_\infty^* = -\frac{C_{A0}}{\nu_A} \quad (8.6)$$

Em que ν_A é o coeficiente estequiométrico da espécie química na reação, sendo negativo para reagentes e positivo para produtos. Mukhtar et. all. (2017) mostraram que essa reação pode ser considerada irreversível em ampla faixa de temperaturas. Logo, adotando a espécie A como o

reagente limitante, no caso o acetato de etila, então $v_A = -1$. Dessa forma, a Equação (8.5) pode ser reescrita como:

$$\xi^* = C_{A0} \frac{(\lambda - \lambda_0)}{(\lambda_{\infty} - \lambda_0)} \quad (8.7)$$

A condutividade do meio reacional, $\lambda(t)$, foi medida experimentalmente em função do tempo usando um condutivímetro, conectado a um computador via porta serial e os valores foram adquiridos via LabVIEW. O reator foi mantido a temperatura constante pela passagem água proveniente de um banho termostatizado. Corridas em várias temperaturas diferentes foram realizadas. Para cada corrida, uma constante da taxa da Equação (8.1) foi determinada por regressão dos dados experimentais à expressão da taxa de reação proposta.

8.3 Materiais e métodos

Para a determinação dos parâmetros cinéticos da reação química em estudo foi utilizada uma unidade experimental com reator encamisado de vidro, agitador magnético e condutivímetro, como mostrado da Figura 64, além de um banho termostatizado. As soluções utilizadas foram:

- Hidróxido de sódio (NaOH) a 0,1 mol/L;
- Acetato de etila a 0,1 mol/L.

Figura 64: Unidade experimental do Reator batelada.



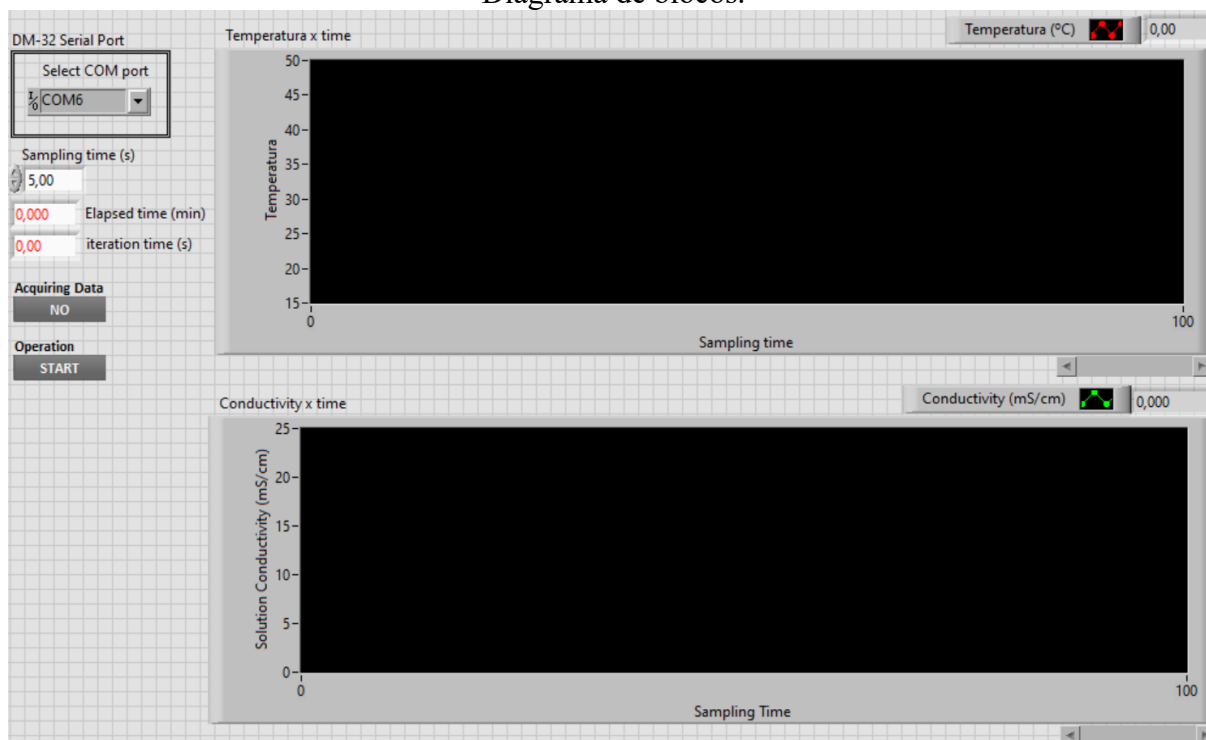
Fonte: Autor.

Para iniciar as corridas experimentais, mediu-se 45 mL de solução de NaOH e 25 mL de solução de acetato de etila, depositando separadamente em béqueres de 50 mL. Com o banho termostatzado ligado e com temperatura relativamente constante, adicionou-se o NaOH no reator e ligou-se o agitador magnético. Em seguida, adicionou a solução de acetato de etila de forma rápida. A condutividade da solução foi automaticamente medida pelo condutímetro acoplado à saída USB de um computador. Este sinal foi adquirido por um *script* do LabVIEW que fazia a leitura da porta serial que estava conectada ao condutímetro. Esse procedimento foi realizado para seis temperaturas (20°C, 25°C, 30°C, 35°C, 40°C e 45°C).

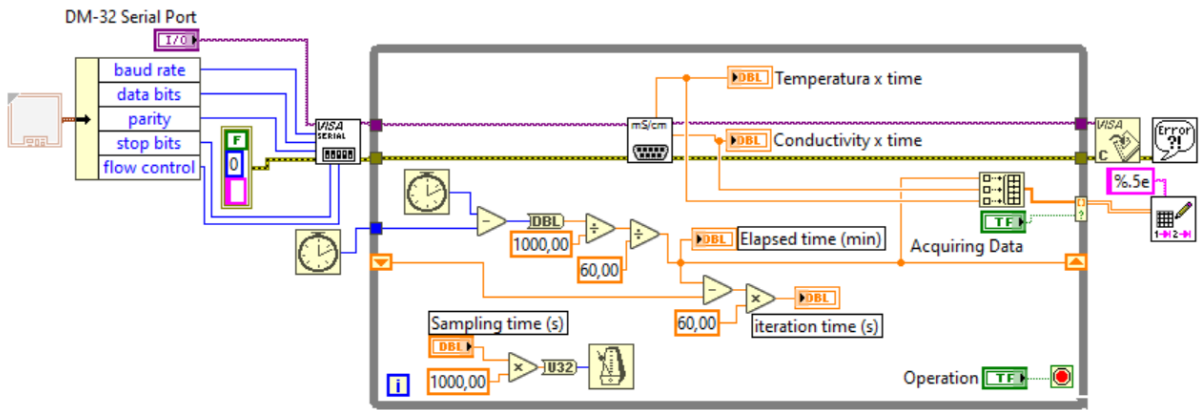
8.4 Interface implementada para a obtenção dos dados cinéticos

Para a obtenção dos dados cinéticos da reação no reator batelada utilizou-se uma interface implementada especificamente para adquirir valores de temperatura, condutividade e tempo de reação. A Figura 65 mostra o painel frontal do *script* LabVIEW desenvolvido.

Figura 65. *Script* LabVIEW para a obtenção dos parâmetros cinéticos. (a). Painel frontal. (b). Diagrama de blocos.



(a)



(b)

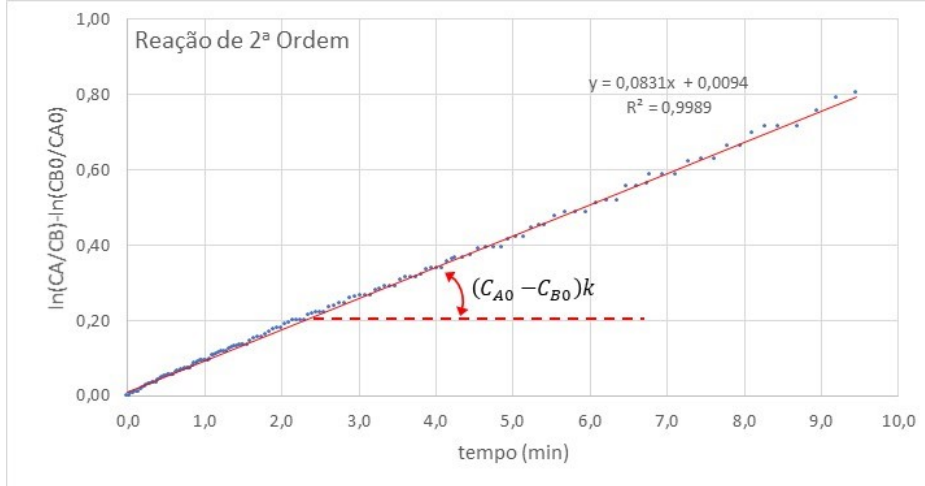
Fonte: Autor.

8.5 Resultados e discussões

8.5.1 Cálculo da constante da taxa de reação

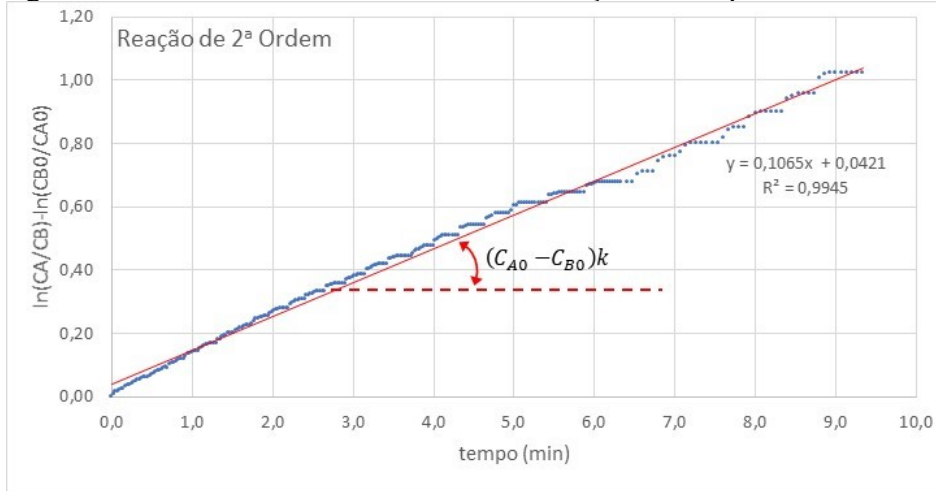
A partir da regressão dos dados experimentais obtidos para cada temperatura foi possível determinar a constante da taxa de reação. O procedimento experimental foi realizado nas temperaturas 20°C, 25°C, 30°C, 35°C, 40°C e 45°C. Para a estimação do valor da constante da taxa da reação foi utilizado o método integral e a equação da taxa de reação (Equação 8.1), conforme apresentado em Hill (1977). Os resultados estão apresentados na Tabela 19 e nas Figuras 66, 67, 68, 69, 70 e 71, respectivamente.

Figura 66. Cálculo da constante da taxa de reação à temperatura de 20°C.



Fonte: Autor.

Figura 67. Cálculo da constante da taxa de reação na temperatura de 25°C.



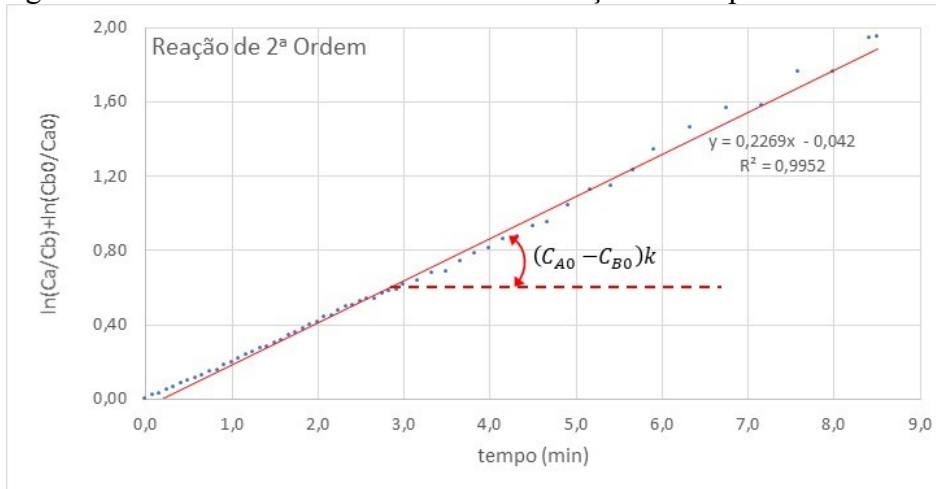
Fonte: Autor.

Figura 68. Cálculo da constante da taxa de reação na temperatura de 30°C.



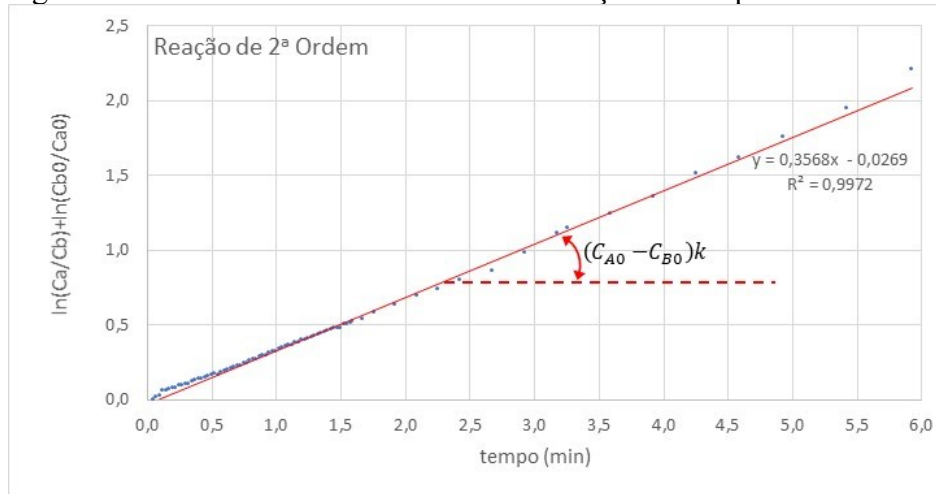
Fonte: Autor.

Figura 69. Cálculo da constante da taxa de reação na temperatura de 35°C.



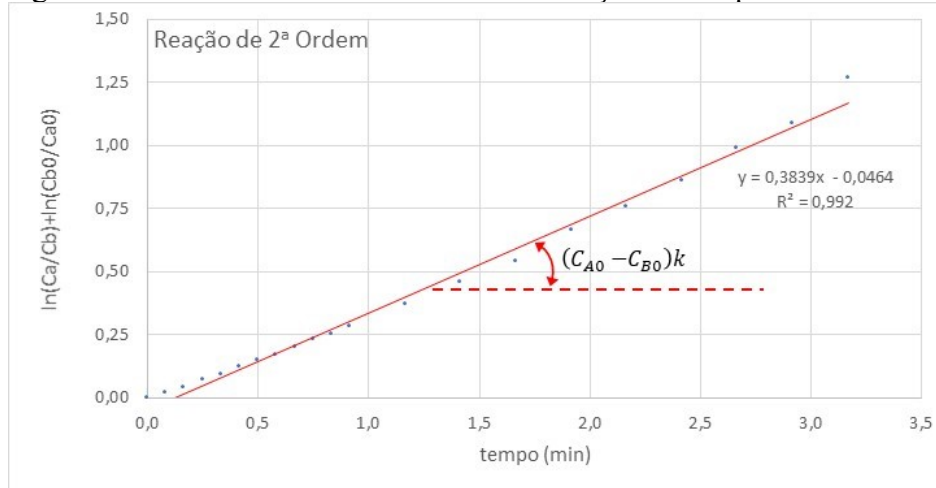
Fonte: Autor.

Figura 70. Cálculo da constante da taxa de reação na temperatura de 40°C



Fonte: Autor.

Figura 71. Cálculo da constante da taxa de reação na temperatura de 45°C



Fonte: Autor.

Tabela 19. Valores estimados da constante em função da temperatura.

Temp (°C)	k (L/mol.min)
20	2,91
25	3,73
30	5,58
35	7,94
40	12,49
45	13,44

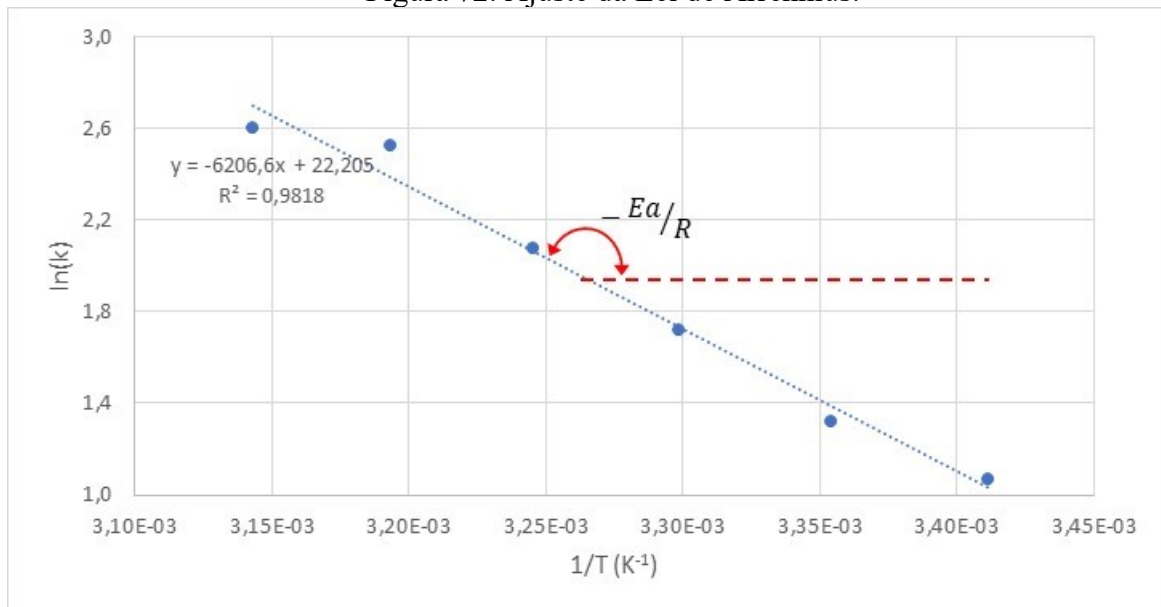
8.5.2 Cálculo da Energia de ativação e do Fator pré-exponencial (A)

A energia de ativação da reação e o fator pré-exponencial foram determinados por regressão linear de $\ln(k)$ versus $1/T$. A Tabela 20 e a Figura 72 apresentam os valores.

Tabela 20. Cálculo da Energia de ativação e do Fator pré-exponencial.

Temp (°C)	Temp (K)	1/T (K ⁻¹)	ln(k)
20	293,15	0,003411223	1,07
25	298,15	0,003354016	1,32
30	303,15	0,003298697	1,72
35	308,15	0,003245173	2,07
40	313,15	0,003193358	2,52
45	318,15	0,003143171	2,60

Figura 72. Ajuste da Lei de Arrhenius.



Fonte: Autor.

Os valores encontrados foram:

$$\ln(A) = 22,205 \quad \rightarrow \quad A = 4,4 \cdot 10^{09} \text{ L}/(\text{mol} \cdot \text{min})$$

$$-Ea/R = 6206,6 \text{ K} \quad \rightarrow \quad Ea = 12.332,61 \text{ cal/mol}$$

O valor de Ea calculado neste trabalho apresentou boa concordância com o valor de $Ea = 11.500$ cal/mol reportado por Tsujikawa H. & Inone (1966).

CAPÍTULO 9

9 CONCLUSÃO

É notório que universidades públicas brasileiras enfrentam limitações financeiras. Considerando isso, aparelhar laboratórios de engenharia e realizar aulas práticas que fazem uso de reagentes e equipamentos específicos pode ser um desafio. Portanto, esse trabalho desenvolveu *kits* experimentais de custo acessível para o estudo de sistemas de reações com reatores de mistura, tubular e batelada, para uso em aulas práticas do curso de graduação em Engenharia Química.

A construção de uma plataforma experimental de baixo custo, direta e surpreendentemente poderosa para automação de práticas de laboratório foi realizada e o desenvolvimento detalhado está apresentado nesse trabalho.

Para desenvolver os *kits* foi utilizada a plataforma de prototipagem eletrônica Arduino para criar o *hardware* das montagens experimentais e fazer a aquisição de dados e controle do experimento. Foi utilizado o *LabVIEW Student Software Suite* para criação das interfaces gráficas. O presente trabalho também apresentou as montagens detalhadas com foco no desenvolvimento de práticas de engenharia química.

Com o intuito de avaliar a qualidade e eficiência desses *kits* e das unidades experimentais criadas foram feitos experimentos para determinação da distribuição de tempo de residência (DTR) em reator tubular e em reatores de mistura em série, além da determinação de parâmetros cinéticos de reação em reator batelada.

Os resultados experimentais mostraram que a plataforma proposta (*hardware* mais *software*) é eficiente para uso em configurações de práticas de laboratório. A instrumentação possui fácil configuração, baixo nível de ruído e baixo custo. Portanto, configurações experimentais como essa podem ser construídas para fins didáticos e de pesquisa em um ambiente de aprendizagem. A plataforma pode ser eficaz para o aprendizado de engenharia química suportado por evidências experimentais, o que permite que alunos sejam capazes de:

- Explicar suas ideias e maneiras de pensar sobre questões, problemas e fenômenos.
- Discutir entre si e com o professor sobre a base das teorias aprendidas em sala de aula.
- Submeter suas ideias e teorias estudadas a evidências experimentais.
- Usar experimentação de baixo custo para desenvolver planos de pesquisa, relatórios e observações de campo para consolidar as lições aprendidas.

Portanto, os objetivos do trabalho foram atingidos e os resultados obtidos foram satisfatórios e indicam que os *kits* experimentais desenvolvidos foram adequados para as montagens e práticas, permitindo que os estudantes obtenham uma clara e condizente comparação entre o estudo teórico e o estudo experimental dos temas abordados. Apesar do estudo ter se limitado a estudos cinéticos e de reatores químicos, a plataforma pode ser estendida e aplicada a vários outros campos da Engenharia Química como, instrumentação, controle, otimização e modelagem de processos, operações unitárias e outros.

10 SUGESTÕES PARA ESTUDOS FUTUROS

- Estudo da performance de reatores CSTR em reações;
- Estudo da performance de reatores PFR em reações;
- Estudo de reatores CSTR e PFR com reação em regime transiente;
- Estudo de reatores CSTR e PFR com reação em condições não isotérmicas.

11 REFERÊNCIAS

ABADIE, J., & CARPENTIER, J. (1969). *Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints*.

BANZI, M. (2011). *Getting Started with Arduino*. O'Reilly, USA.

BERGER, R., HUGH STITT, E., KAPTEIJN, F., & MOULIJN, J. (2001). *Chemical reaction kinetics in practice*. CATTECH.

BRUNETTI, F. (2007). *Mecânica dos Fluidos*.

CANDELAS, F. A., GARCÍA, G. J., PUENTE, S., POMARES, J., JARA, C., PÉREZ, J., TORRES, F. (2015). Experiences on using Arduino for laboratory experiments of Automatic Control and Robotics. *IFAC-Papers OnLine* 48-29, 105-110.

FOGLER, H. S. (2002). *Elementos de Engenharia das Reações Químicas*. LTC.

HILL, C. G. (1977). *Introduction to Chemical Engineering Kinetics and Reactor Design*.

ISHIKAWA, M., & MARUTA, I. (2009). Rapid prototyping for control education using Arduino and open-source technologies. *Proceedings of 8th IFAC Symposium on Advances in Control Education*.

LEE S. W., HANNA M.A., JONES D.D. (2009). *Residence Time Distribution Determination Using On-Line Digital Image Processing*.

LEVENSPIEL, O. (2000). *Engenharia das Reações Químicas*.

MONTGOMEY, D. C., & RUNGER, G. C. (2016). *Estatística Aplicada e Probabilidade para Engenheiros*. Rio de Janeiro: LTC.

MOONAN, W. J., & WOLFE, J. H. (1963).

Mukhtar, A., Shafi, U., Qazi, M., Qadir, H., Qizilbash, M., & Awan, B. (2017). Kinetics of Alkaline Hydrolysis of Ethyl Acetate by Conductometric Measurement Approach Over Temperature Ranges (298.15-343.15K). *Austin Chem Eng*, 1046.

RAO, M., & LONCIN, M. (1974). *Lebensmittel-Wissenschaft und Technologie*.

TSUJIKAWA, H., & INONE, H. (1966). The Reaction Rates of Alkaline Hydrolysis of Ethyl Acetate. *Bulletin of Chemical Society of Japan*, 1837-1842.

ÚBEDA, D., GIL, A., LUCAS, J. A., JIMÉNEZ, L. M., REINOSO, Ó., & ARDILLA, Ó. (2009). Plataforma de prácticas docentes on-line mediante Arduino. *Proc. of XXX Jornadas de Automática*.

WALKER, J. (1906). Method for Determining the Velocities of Saponification. *Proceedings of the Royal Society of London*, 157-160.

WIJAYARATHNE, U., & WASALATHILAKE, K. (2014). Aspen Plus Simulation of Saponification of Ethyl Acetate in the Presence of Sodium Hydroxide in a Plug Flow Reactor. *Journal of Chemical Engineering and Process Technology*, 1062-1069.

