

**UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE QUÍMICA E BIOTECNOLOGIA
GRADUAÇÃO EM QUÍMICA LICENCIATURA**

RAFAEL ANTONIO SANTOS DA SILVA

**USO DA QUÍMICA COMPUTACIONAL COMO FERRAMENTA AUXILIADORA
NO ENSINO DE QUÍMICA**

Maceió/AL

2019

RAFAEL ANTONIO SANTOS DA SILVA

**USO DA QUÍMICA COMPUTACIONAL COMO FERRAMENTA AUXILIADORA NO
ENSINO DE QUÍMICA**

Trabalho de Conclusão de Curso apresentado ao Instituto de Química e Biotecnologia da Universidade Federal de Alagoas, como requisito parcial para obtenção do título de Licenciatura em Química.

Orientador(a): Prof. Dr. Júlio Cosme Santos da Silva

Co-orientador(a): Prof. Dr. Daniela Santos Anunciação

Maceió/AL

2019



ATA DE APRESENTAÇÃO E DEFESA DE TCC - IQB

1. Data da apresentação do TCC: 28 de Janeiro de 2019

2. Aluno / matrícula: Rafael Antonio Santos da Silva / 11111128

3. Orientador(es) / Unidade Acadêmica:
Júlio Cosme Santos da Silva (IQB/UFAL) - Daniela Santos Amenciano (IQB-UFAL)

4. Banca Examinadora (nome / Unidade Acadêmica):

<u>Júlio Cosme Santos da Silva</u>	(Presidente)	Nota: <u>8,5</u>
<u>Daniela Santos Amenciano</u>	(1º avaliador)	Nota: <u>8,5</u>
<u>[Assinatura]</u>	(2º avaliador)	Nota: <u>8,5</u>
<u>[Assinatura]</u>	(3º avaliador)	Nota: <u>8,5</u>

5. Título do Trabalho: Uso da Química Computacional como Ferramenta Auxiliadora no Ensino de Química

6. Local: Sala de Reunião - IQB

7. Apresentação: Horário início: 14:07 Horário final: 14:39
Arguição: Horário início: 14:40 Horário final: 15:17

8. Nota final: 8,5

Em sessão pública, após exposição do seu trabalho de TCC por cerca de 32 minutos, o candidato foi arguido oralmente pelos membros da banca por 37 minutos, tendo como resultado:

- APROVADO
 APROVADO COM RESTRIÇÕES – mediante modificações no trabalho que foram sugeridas pela banca como condicional para aprovação.
 NÃO APROVADO.

Na forma regulamentar foi lavrada a presente ata que é abaixo assinada pelos membros da banca, na ordem acima determinada, e pelo candidato:

Maceió, 28 de Janeiro de 2019.

Presidente: Júlio Cosme Santos da Silva
1º Avaliador: Daniela Santos Amenciano
2º Avaliador: [Assinatura]
3º Avaliador: [Assinatura]
Candidato: Rafael Antonio Santos da Silva

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me guiado e feito capaz de acreditar, mesmo nos momentos difíceis e me levado pelo caminho certo nesta fase da minha vida.

Agradeço também as minhas mães, Marilene Julia da Silva e Josefa Julia da Silva, por terem me dado todo carinho, educação, acreditando sempre na minha capacidade e me apoiando nas decisões. A todos meus familiares, que estiveram e estão presentes na minha vida, fazendo chegar até este momento.

Agradeço aos meus orientadores, Prof. Dr. Júlio Cosme Santos e Prof. Dr. Daniela Santos Anunciação, por terem acreditado neste trabalho, demonstrando dedicação e empenho, os quais foram essenciais para a conclusão deste trabalho.

Agradeço ao meu namorado Martin da Costa dos Santos, por está do meu lado apoiando e me fazendo feliz, mesmo com todo o estresse. Ao meu amigo Vinicius Moura por estar comigo durante todo o trajeto deste curso e a todos meus amigos que fizeram parte dessa etapa inesquecível.

RESUMO

A química é uma disciplina de grande dificuldade aos alunos, seja no ensino médio ou na graduação, isso deve-se ao fato de apresentar uma significativa quantidade de conteúdos abstratos e necessidade de análises matemáticas. Neste trabalho de conclusão de curso, foi feita uma explanação de como a utilização da química computacional pode intervir no estudo da química de uma forma satisfatória e que faça transparecer suas vantagens. O mesmo foi dividido em dois momentos, um relacionado com a apresentação dos conceitos gerais necessários à sua compreensão e um outro onde foi realizado um estudo de caso que envolveu o desenvolvendo de um código computacional simples, escrito em *Fortran*, para realizar um estudo de cinética química envolvendo processos de 1ª e 2ª Ordem. Por fim, com base no estudo realizado, foi proposto uma ementa para uma disciplina voltada para o estudo computacional de conteúdos fundamentais da química.

Palavras-chave: *Química computacional, Cinética Química, Computação científica.*

ABSTRACT

Chemistry is a discipline of great difficulty to students, whether in high school or undergraduate level, this is due to the fact of presenting a significant amount of abstract content and the need for mathematical analysis. In this completion of course work, was carried out an explanation about how the use of Computational Chemistry can intervene in the study of chemistry in a satisfactory way and that shows its advantages. This work was divided in two moments, one related to the presentation of the general concepts necessary for its understanding and another one where a case study was carried out that involved the development of a simple computational code, written in Fortran, to carry out a study of kinetic chemistry involving 1st and 2nd order processes. Finally, based on the study carried out, was proposed a syllabus for a discipline focused on the computational study of fundamental contents of chemistry.

Keywords: Computational chemistry, Chemical Kinetics, Scientific computing.

LISTA DE FIGURAS

Figura 1 – Exemplo de ações que podem constituir um algoritmo lógico: Passos seguidos pelo autor deste trabalho em atividades matutinas.	16
Figura 2 – Logomarcas de programas disponíveis no mercado que são ambientes de desenvolvimento integrados.	19
Figura 3 – Interface do ambiente de desenvolvimento Plato da empresa <i>SILVERFROST</i>	20
Figura 4 – Modelo de um fluxograma utilizado em um programa para determinar o caráter ácido/base de uma solução.	24
Figura 5 – Representação da compilação em linguagem <i>Fortran</i>	25
Figura 6 – Comandos inicial (<i>PROGRAM</i>) e final (<i>END PROGRAM</i>) para a criação de um programa em linguagem <i>Fortran</i>	25
Figura 7 – Visualização das variáveis no ambiente de desenvolvimento integrado (Plato).	26
Figura 8 – Visualização dos comandos ler (<i>READ</i>) e escrever (<i>WRITE</i>) no ambiente de desenvolvimento integrado (Plato).	27
Figura 9 – Exemplo ilustrado no Plato para um caso situacional, onde uma das variáveis assume valor diferente do esperado.	28
Figura 10 – Exemplo ilustrado no Plato para um caso situacional em que uma das variáveis assume mais de um valor diferente do esperado.	28
Figura 11 – Exemplo ilustrado no Plato para um caso de repetição.	29
Figura 12 – Exemplo ilustrado no Plato para a criação de um arquivo de saída.	30
Figura 13– Típica curva de concentração por tempo para uma reação de 1ª ordem.	32

Figura 14 – Típica curva de concentração por tempo para uma reação de 2ª Ordem.	33
Figura 15 – Primeiro bloco de linhas de comandos do programas para determinar a cinética de uma reação.	34
Figura 16 – Segundo bloco de linhas de comandos do programas para determinar a cinética de uma reação	34
Figura 17 – Terceiro bloco de linhas de comandos do programa para determinar a cinética de uma reação.	35
Figura 18 – Quarto bloco de linhas de comandos do programa para determinar a cinética de uma reação.	35
Figura 19 – Quinto bloco de linhas de comandos do programa para determinar a cinética de uma reação.	36
Figura 20 – Arquivo concA preenchido corretamente para leitura do programa.	37
Figura 21 – Trajeto para a execução do programa na ferramenta plato.	37
Figura 22 – Tela gerada após execução do programa seguindo o passo da figura 21.	38
Figura 23 – Valores obtidos no programa elaborado em linguagem <i>Fortran</i> para cinética de 1ª ordem.....	39
Figura 24 – Perfil gráfico de uma cinética de 1ª Ordem para a relação da concentração de A em função de um tempo t.	39
Figura 25 – Linearização da relação de concentração de A em função do tempo por escala logarítmica para uma reação de 1ª Ordem construído pela ferramenta gráfica xmgrace.....	40
Figura 26 – Valores obtidos no programa elaborado em linguagem <i>Fortran</i> para cinética de 2ª ordem.....	41

Figura 27– Perfil gráfico de uma cinética de 2ª Ordem para a relação da concentração de A em função de um tempo t. construído pela ferramenta gráfica xmgrace.41

LISTA DE TABELAS

Tabela 1 – Algumas das universidades brasileiras e disciplinas presentes em suas grades curriculares que direcionam ao estudo computacional da química.	22
Tabela 2 – Dados reais utilizados para realizar o teste com o programa criado neste trabalho.	38

LISTA DE SIGLAS E ABREVIATURAS

FORTRAN	Formula Translater
IDE	Integrated development environment
UFAL	Universidade Federal de Alagoas
UFRJ	Universidade Federal do Rio de Janeiro
UFRN	Universidade Federal do Rio Grande do Norte
UFS	Universidade Federal de Sergipe
UNICAMP	Universidade Estadual de Campinas
USP	Universidade de São Paulo

SUMÁRIO

1.0	INTRODUÇÃO	13
2.0	FUNDAMENTAÇÃO TEÓRICA	15
2.1	Programas, programação científica e suas estruturas.....	15
2.2	O algoritmo	16
2.3	A linguagem <i>Fortran</i>	18
2.4	Ambiente desenvolvimento integrado (IDE).....	18
3.0	OBJETIVOS	21
3.1	Objetivo Geral	21
3.2	Objetivos Específicos.....	21
4.0	JUSTIFICATIVA	22
5.0	METODOLOGIA PARA A CRIAÇÃO DO PROGRAMA	23
5.1	Passos para a criação de um programa	23
5.2	Comandos básicos necessários para criação de um programa em linguagem Fortran	25
5.2.1	<i>Comandos de Instrução</i>	25
5.2.2	<i>Comandos de Decisão</i>	27
5.2.3	<i>Comandos de Repetição</i>	29
5.3	Criando um arquivo de saída para os dados gerados no programa	29
6.0	ANÁLISES DOS DADOS E DOS RESULTADOS	31
6.1	Formulação Matemática	31
6.1.1	<i>Reações de 1ª Ordem</i>	31
6.1.2	<i>Reações de 2ª Ordem</i>	32
6.2	O Programa	33
6.3	Utilizando o Programa	36
6.4	Dados Obtidos	38

6.5	Proposta de Ementa para uma disciplina de programação em química	43
7.	CONCLUSÃO	45
8.	REFERÊNCIAS	46

1.0 INTRODUÇÃO

A química dada como uma ciência experimental apresenta conteúdos de difícil entendimento por parte dos alunos, agravado pela não visualização de seus conteúdos abstratos. Como alternativa que permita a visualização de fenômenos dificilmente observados experimentalmente ou de difícil compreensão em sala de aula, pesquisadores afirmam que a utilização de recursos diferenciados como *softwares* educacionais podem ser mais significativos e atrativos aos alunos (LOCATELLI et al., 2015).

Segundo BORDINI et al. (2016) o pensamento computacional é um processo de resolução de problemas. Esta área do conhecimento trabalha diversas habilidades cognitivas humanas necessárias para a formação de cidadãos mais confiantes, persistentes, tolerantes, mesmo com as complexidades adversas, e capazes de trabalhar com outros indivíduos para solução de problemas em qualquer área do conhecimento.

Neste trabalho de conclusão de curso, foi feita uma explanação de como a utilização da computação científica pode intervir no estudo da química de uma forma satisfatória e que faça transparecer suas vantagens. O mesmo foi dividido em dois momentos, um voltado a apresentar os conceitos gerais necessários à sua compreensão e um outro onde será feito um estudo de caso, onde foi utilizado um programa criado com base na linguagem computacional simples (*Fortran*) para realizar um estudo de cinética química envolvendo reações de primeira e 2ª ordem.

Sendo assim, utilizou-se ferramentas computacionais simples como é o caso da plataforma Plato, que está disponível gratuitamente na internet para criação deste programa, abrindo deste modo novas portas para que os graduandos do curso possam explorar ainda mais essas ferramentas na criação de seus próprios programas envolvendo diferentes ramos da Química.

O curso de química licenciatura da Universidade Federal de Alagoas (UFAL) não apresenta em sua grade curricular nenhuma disciplina de caráter obrigatório que inclua o uso da computação científica para explorar conceitos de química, o que pode ocasionar certo atraso no acesso dos estudantes do curso ao vasto ambiente tecnológico proporcionado pelos computadores e de como o uso dessa ferramenta pode ajudar na consolidação de conceitos fundamentais da química construídos

durante o curso. Introdução à computação é uma disciplina eletiva, a qual não vem sendo ofertada pelo Instituto de Química Biotecnologia (IQB) desde 2007, segundo dados do projeto pedagógico da unidade.

A disciplina apresenta em sua ementa: um resumo Histórico, organização de um computador, Utilização do Sistema Operacional Windows (Word, Excel, Power Point, bloco de notas etc.). Os quais em nenhum momento são direcionados à Química ou ao seu ensino.

Deste modo este trabalho apresenta como proposta central a elaboração de uma disciplina de química que inclua computação científica como uma ferramenta para auxiliar os estudantes na consolidação de alguns conceitos vistos ao longo do curso apresentando também como seu uso pode ser importante para tratar problemas reais da química.

2.0 FUNDAMENTAÇÃO TEÓRICA

Para COSTA et al. (2012), a natureza microscópica e abstrata dos conhecimentos químicos provoca no meio estudantil uma dificuldade na aprendizagem de seus diversos conceitos, fato que se deve a linguagem química ter uma grande presença simbólica, o que exige capacidade de abstração e generalização.

Na atualidade, o desenvolvimento tecnológico trouxe ferramentas computacionais capazes de simular e exibir situações da realidade, possibilitando aos alunos aplicar o conhecimento teórico. Entre elas se destacam *softwares* educacionais que proporciona criar ambientes virtuais, mapas conceituais, jogos didáticos, vídeos e materiais de apoio, com uma infinidade de temas abordados no ensino de química. Sua maior vantagem está na inserção do aluno e do profissional da educação em um universo tecnológico que estimula cada vez mais a curiosidade, descobertas e possibilidades (PEREIRA, 2014).

2.1 Programas, programação científica e suas estruturas

Um programa computacional é resultado de uma atividade intelectual de seu programador. Esta atividade depende total e exclusivamente do conhecimento prévio deste programador sobre manejo e solução do problema proposto, de acordo com um conjunto de instruções e dados predefinidos, sendo executado por uma máquina, afim de cumprir seu objetivo (SOUZA et al., 2011).

O programa surge da necessidade humana de meios mais rápidos para solucionar seus problemas, por exemplo, um programa que controle outras máquinas, minimizando o manejo humano, a fim de evitar acidentes e diminuir o custo de pessoal e capacitações; ou um programa para gerir seus funcionários, dados e folhas. Tudo parte de uma necessidade cada vez mais notável no mercado. O uso do computador tem acompanhado um novo tipo de educação, seja no âmbito familiar ou das escolas, de universidades ou empresas.

Para SOUZA et al. (2011), a criação de um programa deve seguir quatro passos, sendo eles: A análise, onde se detalha como o programa vai funcionar. O projeto, tendo-se uma noção maior do resultado esperado. A implementação,

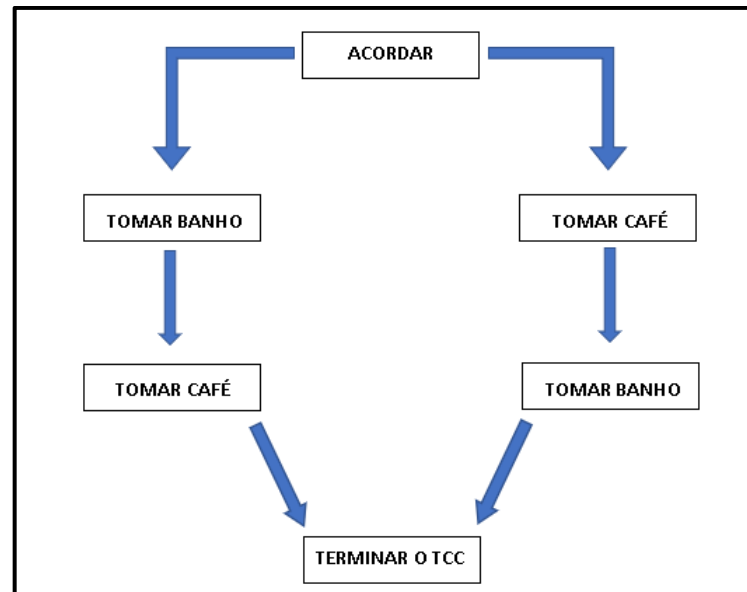
ocorrendo a utilização da linguagem de programação e a construção do programa, e, por fim, os testes realizados após a construção os quais devem satisfazer todas as especificações do usuário destino.

Na construção de um programa, a qual em sua maior parte é dada de forma escrita, é essencial uma visão gráfica de todo o processo. A mesma é dada por um fluxograma, ou seja, uma representação gráfica do passo a passo realizado pelo programa para solucionar um problema e que possa ser entendido por qualquer pessoa que deseja utilizá-lo. No fluxograma são utilizados três tipos de instruções: as sequenciais, sem nenhum tipo de decisão; as de decisão, onde podem ocorrer uma mudança no padrão seguido pelo programa dependendo do resultado; e as de repetição, que pelo nome já pode ser entendida como um conjunto repetitivo de ações (SOUZA et al., 2011).

2.2 O algoritmo

Em nosso cotidiano o ser humano pensa em diferentes formas de organizar suas atividades, tarefas ou deveres. Tomar banho, tomar café e escovar os dentes, por exemplo, são alguns passos da estratégia que geralmente é traçada pelo homem no dia a dia. Alguns podem começar essas atividades em ordens diferentes, mas enfim, existem variadas formas de realizá-las. Esse conjunto de passos seguidos com o objetivo de solucionar determinada atividade também pode ser visto como um algoritmo lógico, conforme exemplo apresentado na Figura 1.

Figura 1 – Exemplo de ações que podem constituir um algoritmo lógico: Passos seguidos pelo autor deste trabalho em atividades matutinas.



Fonte: Autor, 2018.

Segundo SOUSA et al. (2014) algoritmo pode ser definido como:

“Sequência finita, ordenada e não ambígua de passos para solucionar determinado problema ou realizar uma tarefa”

Um algoritmo é de grande importância para a elaboração de um código computacional, uma vez que o computador/máquina não consegue reconhecer instruções desconhecidas ou fazer a conexão entre símbolos ou expressões em uma linguagem que o mesmo não consegue decifrar. Desta forma, todas as etapas precisam ser cuidadosamente pré-definidas.

SOUSA et al. (2014) em seus estudos definem cinco pontos que um algoritmo computacional deve especificamente possuir:

Definição: Os passos de um algoritmo devem ser bem definidos, objetivando a clareza e evitando ambiguidades.

Finitude: Um algoritmo deve chegar ao seu fim após um número definido de passos.

Efetividade: Um algoritmo deve ser efetivo, ou seja, suas operações devem ser básicas o suficiente para que possam, em princípio, serem executadas de maneira exata e em um tempo finito.

Entradas: Um algoritmo deve possuir zero ou mais entradas. Estas são insumos ou quantidades que são processados pelos algoritmos durante a execução de seus passos.

Saídas: Um algoritmo deve possuir uma ou mais saídas as quais representam o resultado do trabalho realizado pelos algoritmos”.

2.3 A linguagem *Fortran*

A linguagem utilizada para a programação, assim como a nossa linguagem natural tem finalidade de fornecer um meio eficiente de comunicação, formada por um conjunto de palavras e com regras para sua utilização.

Estas linguagens diferem em muitos pontos, por exemplo, a computacional não tem apenas o propósito de facilitar a comunicação entre pessoas, mas também entre pessoas e o computador. Erros gramaticais não impossibilitam uma conversa entre duas pessoas, já na linguagem computacional este erro, seja ele por uma simples omissão de sinal, é capaz de impossibilitar que a comunicação seja iniciada.

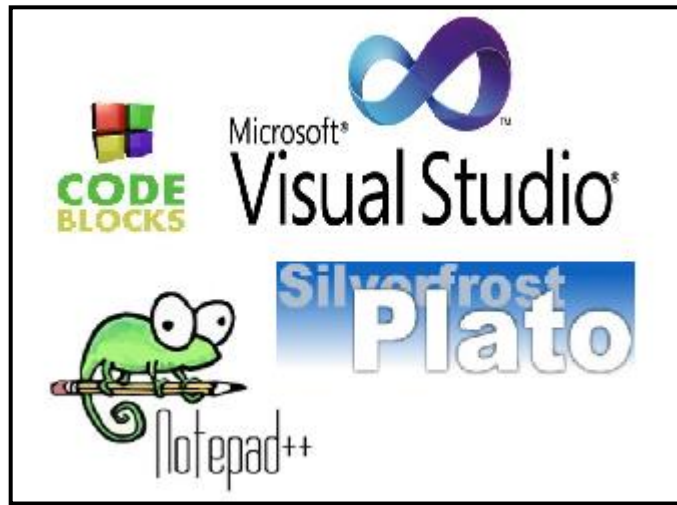
O *Fortran* (*FORmula TRANslation System*), primeira linguagem de programação alto nível a ser proposta surgiu em 1956 e tem como função a resolução de problemas da área científica utilizando computadores. Seu nome é formado pela combinação de duas palavras, *formula* e *translation*, ou seja, tradutor de fórmulas em sua tradução livre e cuja linguagem é bastante difundida e já apresentou diversas versões (CRISTO, 2003).

2.4 Ambiente desenvolvimento integrado (IDE)

O Ambiente de desenvolvimento integrado (IDE - *Integrated development environment*) é uma ferramenta necessária à elaboração, compilação e execução de aplicações em *Fortran*. Há muitas opções no mercado e diversas maneiras de desenvolver em *Fortran*.

Existe uma gama de sistemas que pode deixar o elaborador em dúvida no momento de escolher o ambiente de trabalho, a exemplo de *Code Blocks*, *Microsoft Visual Studio*, *Notepad++* e *Plato*, dentre outras. A Figura 2, representa exemplos de programas computacionais disponíveis no mercado e que se caracterizam como ambientes de desenvolvimento integrado.

Figura 2 – Logomarcas de programas disponíveis no mercado que são ambientes de desenvolvimento integrados.



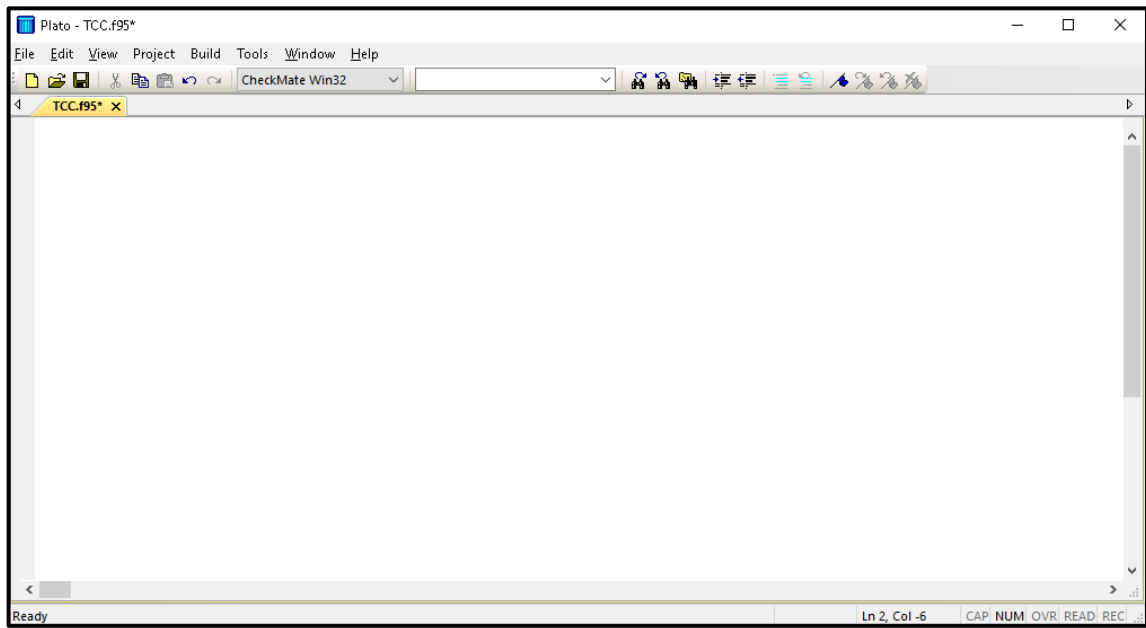
Fonte: Autor, 2018.

Um IDE é uma ferramenta que reúne uma grande quantidade de funcionalidades para a programação, tais como a criação de projetos em diversas plataformas, compilações transparentes, configurações com interface simples e editores de textos com várias características especiais.

Segundo TOLETO e DIAS, os IDE's não apresentam desvantagens, porém podem deixar o usuário mal-acostumado com a transparência do processo, causando-lhe uma dependência e pouca exploração da forma mais básica que inclui a criação do programa.

O Plato da empresa SILVERFROST fornece um ambiente visual de desenvolvimento com uma enorme quantidade de funcionalidades aliadas à facilidade de utilização voltadas à linguagem *Fortran*. Para a digitação de um código computacional (algoritmo), a única ferramenta que seria necessária é o bloco de notas do Windows, entretanto, o Plato já traz em si essa funcionalidade, não necessitando que o programador busque este recurso em outros programas. A Figura 3 representa uma imagem da interface do ambiente de desenvolvimento de acordo com o programa Plato mencionado acima.

Figura 3 – Interface do ambiente de desenvolvimento Plato da empresa SILVERFROST.



Fonte: Autor, 2018.

3.0 OBJETIVOS

3.1 Objetivo Geral

O objetivo deste trabalho corresponde ao uso da química computacional como ferramenta auxiliar na compreensão de conceitos fundamentais da Química.

3.2 Objetivos Específicos

Este trabalho teve como objetivos específicos:

- Destacar a importância da Química computacional;
- Alertar para a necessidade da inclusão digital de alunos da graduação;
- Exibir como é criado um programa computacional simples em linguagem *Fortran*;
- Propor uma disciplina de Química computacional que possa ser implementada nas grades dos cursos de Química da UFAL.

4.0 JUSTIFICATIVA

John Pople em 1998 recebeu o prêmio Nobel por sua contribuição ao avanço de métodos computacionais, levando assim a popularização dos mesmos. Segundo RAUPP et al. (2008) a exploração da potencialidade de ferramentas computacionais no ensino de Química ainda está apenas no começo, mas resultados preliminares indicam a existência de ganhos significativos em aprendizagem devido a interação em computador e aluno, onde a linguagem dada, até então como simbólica pode ser assimilada.

Conforme Tabela 1, podemos observar a predominância de disciplinas exploratórias a computação em cursos de Química Licenciatura de outras universidades brasileiras, podendo-se assim notar a importância desse ensino para a formação dos profissionais.

Sendo assim, entende-se necessidade trazer ao curso de química Licenciatura da UFAL uma disciplina com essa abordagem. Uma vez, que no projeto pedagógico atual do curso de Química Licenciatura da Universidade Federal de Alagoas, não consta nenhuma uma disciplina obrigatória ou eletiva que tenha como foco exploração da Química Computacional, essa informação foi retirada do site oficial do Instituto de Química e Biotecnologia.

Tabela 1 – Algumas das universidades brasileiras e disciplinas presentes em suas grades curriculares que direcionam ao estudo computacional da química.

Universidade	Disciplina	Caráter
USP	Introdução à Tecnologia ou à Pesquisa Científica	Obrigatória
UFRJ	Introdução a Computação	Eletiva
UNICAMP	Tecnologias de Informação e Comunicação Aplicadas ao Ensino de Química	Obrigatória
UFRN	Informática e Educação	Obrigatória
UFS	Ferramentas Computacionais Para o Ensino de Química	Obrigatória

Fonte: Autor, 2019.

5.0 METODOLOGIA PARA A CRIAÇÃO DO PROGRAMA

A metodologia desenvolvida neste trabalho envolveu a criação do programa caracterizado como um IDE em linguagem *Fortran* utilizando como modelo o conteúdo de Cinética Química. Este conteúdo é abordado mais superficialmente na disciplina de Química Geral 2 dos cursos de Química do Instituto de Química e Biotecnologia da UFAL, porém é trabalhado de forma mais aprofundada na disciplina de Físico-Química 2.

A criação do referido programa requer uma série de etapas sequenciais que serão apresentadas detalhadamente a seguir, desde a elaboração aos comandos estabelecidos para que fosse executado.

5.1 Passos para a criação de um programa

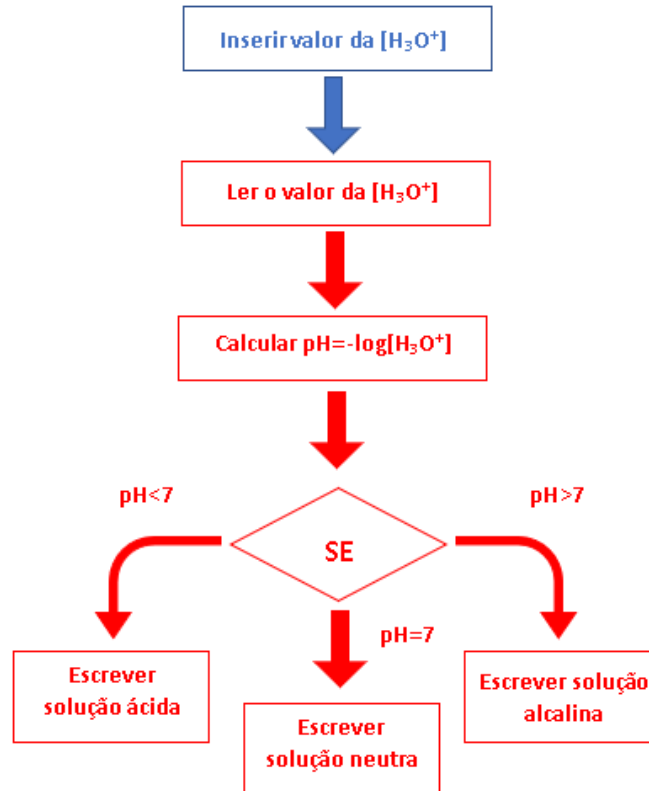
Na criação de um programa utilizando química computacional é necessário atender a alguns critérios os quais implicarão diretamente na sua utilidade:

a) Definir de forma sucinta o problema que se pretende solucionar computacionalmente. Qual sua finalidade? Onde será empregado?

Exemplo: Expressar as raízes de uma função, calcular uma área, volume, etc.

b) Elaborar um diagrama de fluxo, o qual explicará o passo a passo das tarefas que o programa realizará até a obtenção do resultado, a exemplo da Figura 4 que representa um modelo de fluxograma aplicado para a determinação do perfil ácido-base de uma solução sob análise. Em azul marcado o comando inicial, inserido durante a rotina do programa ou recolhido de um arquivo externo (output) e em vermelho, as tarefas que o programa realizará até seu fim, onde será feita a exibição da informação sobre a solução.

Figura 4 – Modelo de um fluxograma utilizado em um programa para determinar o caráter ácido/base de uma solução.



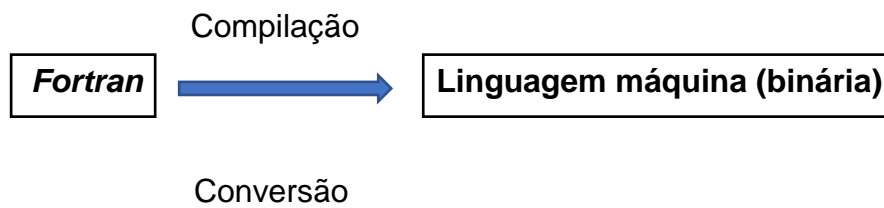
Fonte: Adaptado de UNIVERSIDADE DE COIMBRA. Departamento de Química - Programação em *Fortran 90* - Informática geral, 2010/2011.

c) Escrever o código do programa (Codificar o algoritmo, seguindo o diagrama anterior e recorrendo a uma linguagem de programação (*FORTRAN*));

d) Compilar o programa para correção de eventuais erros de sintaxe.

Compilar é o ato de tradução de um programa feita em uma linguagem de alto nível para uma linguagem máquina, que neste caso trata-se de uma linguagem binária (Figura 5).

Figura 5 – Representação da compilação em linguagem *Fortran*.



Fonte: Autor, 2018.

e) Testar o programa para uma solução conhecida.

5.2 Comandos básicos necessários para criação de um programa em linguagem Fortran

Antes de iniciar a criação efetiva do programa, é necessário conhecer os comandos básicos e a estrutura a ser seguida. No Plato, que será a ferramenta utilizada durante este trabalho, os comandos estarão destacados pela coloração azul.

5.2.1 Comandos de Instrução

Todo programa em linguagem *Fortran* deve apresentar em sua estrutura, obrigatoriamente, um comando de início e fim, conforme destaque em azul representado na figura 6.

Figura 6 – Comandos inicial (PROGRAM) e final (END PROGRAM) para a criação de um programa em linguagem *Fortran*.

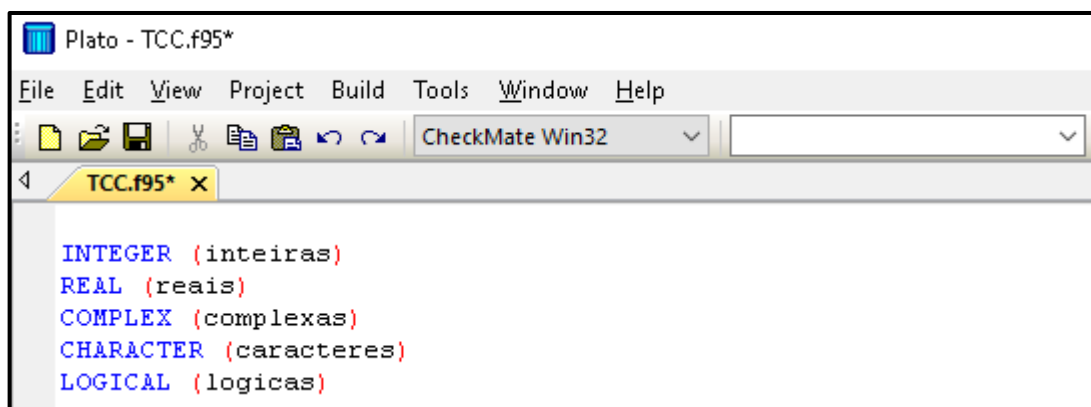
```
PROGRAM Química "Início do programa de nome Química"  
-----  
END PROGRAM Química "Final do programa de nome Química"
```

Fonte: Autor, 2018.

No decorrer do programa, após seu início, deve vir a declaração de variáveis, as quais podem ser dos tipos: inteiras, reais, complexas, caracteres ou lógicas.

O comando *IMPLICIT NONE* (do inglês: nenhum implícito em tradução livre) deve ser utilizado antes da declaração das variáveis, pois o mesmo faz com que todas as variáveis tenham que ser obrigatoriamente definidas durante a estrutura do programa (Figura 7).

Figura 7 – Visualização das variáveis no ambiente de desenvolvimento integrado (Plato).

A screenshot of the Plato IDE interface. The window title is "Plato - TCC.f95*". The menu bar includes "File", "Edit", "View", "Project", "Build", "Tools", "Window", and "Help". The toolbar contains icons for file operations and a dropdown menu set to "CheckMate Win32". The main editor area shows the following code:

```
INTEGER (inteiras)
REAL (reais)
COMPLEX (complexas)
CHARACTER (caracteres)
LOGICAL (logicas)
```

Fonte: Autor, 2018.

A leitura dos dados necessários para a realização de uma tarefa computacional requer o uso do comando *READ* (do inglês: ler).

Exemplo: `READ(*,*) k1,k2`

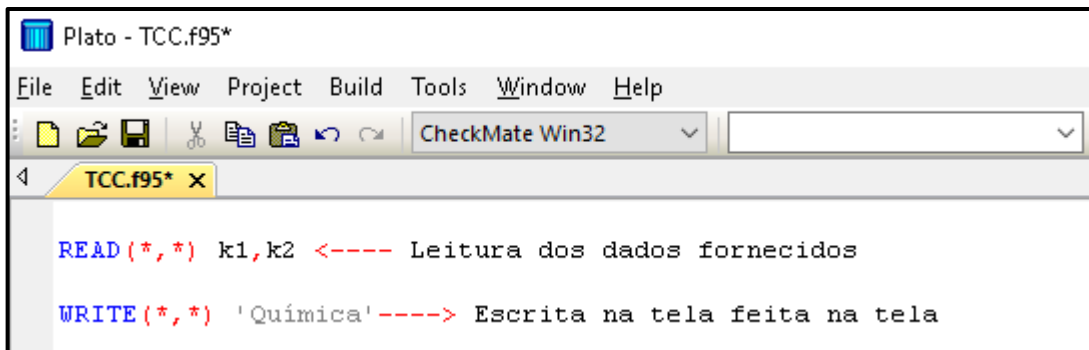
O exemplo acima representa a leitura das quantidades *k1* e *k2* na tela do computador.

Para escrever informações na tela do computador ou num arquivo de saída (*output*), utiliza-se o comando *WRITE* (do inglês: escrever).

Exemplo: `WRITE(*,*) 'Química'`

Os comandos de leitura e escrita gerados de acordo com o programa utilizado neste trabalho estão representados na Figura 8.

Figura 8 – Visualização dos comandos ler (*READ*) e escrever (*WRITE*) no ambiente de desenvolvimento integrado (Plato).



The screenshot shows the Plato IDE interface. The title bar reads 'Plato - TCC.f95*'. The menu bar includes 'File', 'Edit', 'View', 'Project', 'Build', 'Tools', 'Window', and 'Help'. The toolbar contains icons for file operations and a dropdown menu set to 'CheckMate Win32'. The main editor window shows the following code:

```

READ (*, *) k1, k2 <---- Leitura dos dados fornecidos
WRITE (*, *) 'Química' ----> Escrita na tela feita na tela
  
```

Fonte: Autor, 2018.

O (*,*) depois do comando significa que o mesmo irá escrever ou ler a informação na tela do computador. O primeiro (*) está relacionado à unidade onde o mesmo será escrito ou lido, já o segundo (*) está relacionado ao formato utilizado, neste caso, como está o asterisco, o formato é livre.

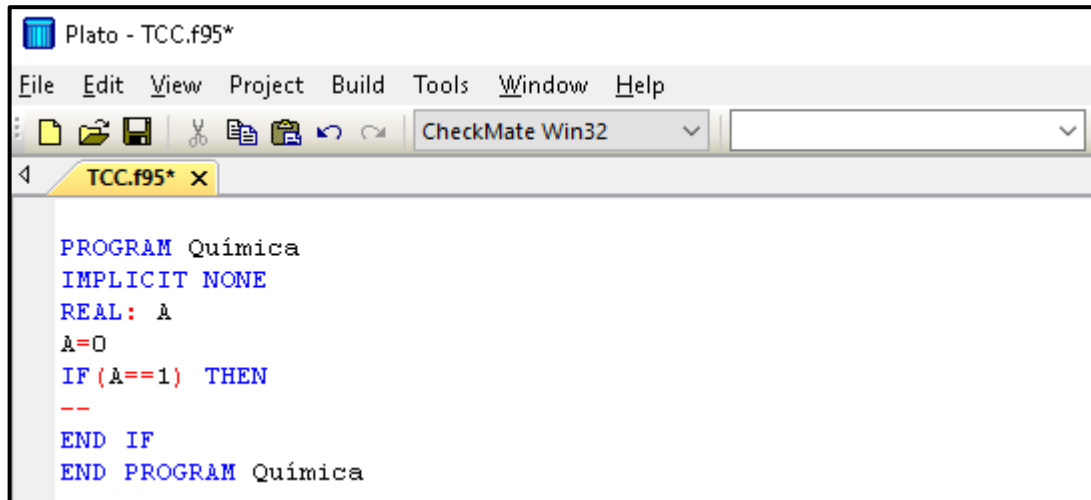
5.2.2 Comandos de Decisão

Outros comandos utilizados neste desenvolvimento do programa, têm como foco casos situacionais. Esses são: *IF*, *ELSE IF*, *ELSE* e *END IF*. O comando *IF* (do inglês: se) será utilizado para iniciar o caso situacional e *END IF* (do inglês: fim do se) será utilizado para encerrar o comando *IF*.

($A=1$) é a sentença lógica que deve acompanhar o comando *IF*. Já o comando *THEN* (do inglês: então), indica os comandos subsequentes a serem realizados pelo programa caso a sentença lógica avaliada a partir do *IF* seja válida.

A Figura 9, representa um caso em que uma das variáveis assume valor diferente do esperado e no qual se faz necessário o uso dos comandos *IF*, *THEN* e *END IF*.

Figura 9 – Exemplo ilustrado no Plato para um caso situacional, onde uma das variáveis assume valor diferente do esperado.

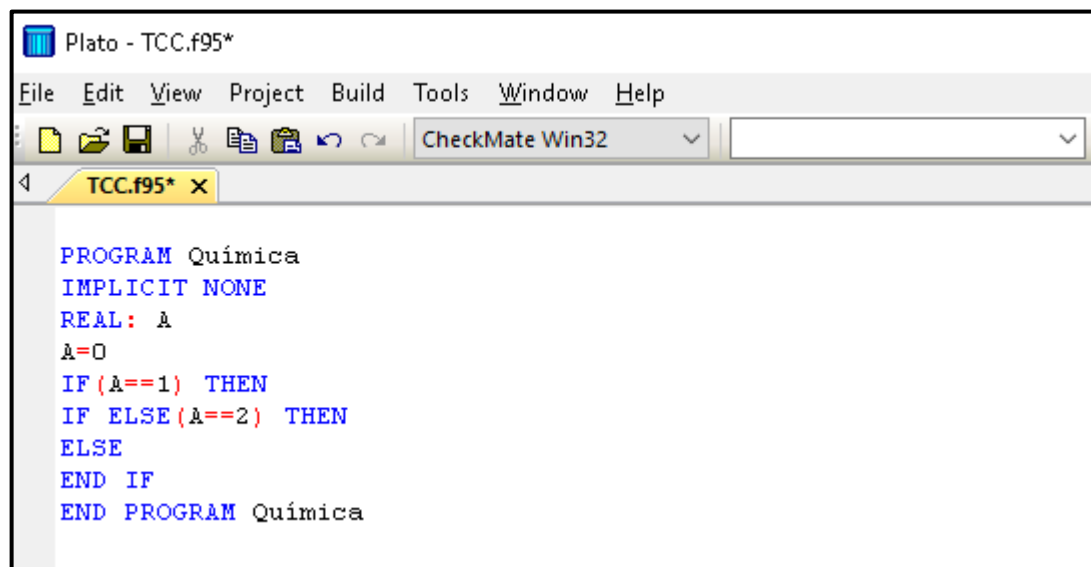


```
Plato - TCC.f95*
File Edit View Project Build Tools Window Help
CheckMate Win32
TCC.f95* x
PROGRAM Química
IMPLICIT NONE
REAL: A
A=0
IF (A==1) THEN
--
END IF
END PROGRAM Química
```

Fonte: Autor, 2018.

O comando *ELSE IF* (do inglês: outro se) é utilizado no caso onde apresentam-se mais de uma situação lógica, já o *ELSE* (do inglês: outro) finaliza o comando *ELSE IF*. Neste caso, uma das variáveis sob análise assume mais de um valor diferente do valor esperado (Figura 10)

Figura 10 – Exemplo ilustrado no Plato para um caso situacional em que uma das variáveis assume mais de um valor diferente do esperado.



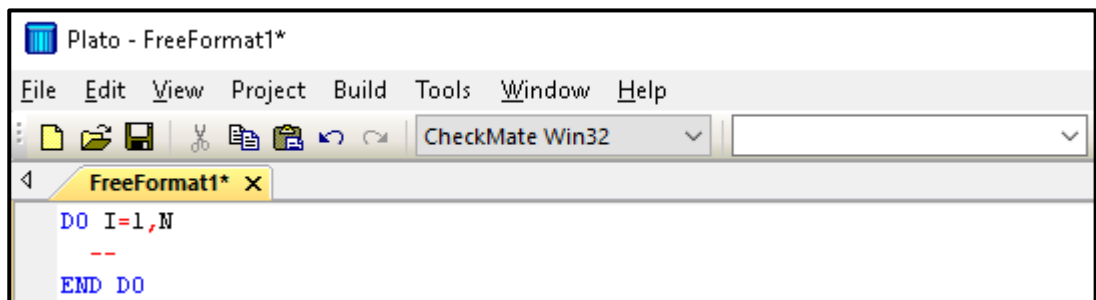
```
Plato - TCC.f95*
File Edit View Project Build Tools Window Help
CheckMate Win32
TCC.f95* x
PROGRAM Química
IMPLICIT NONE
REAL: A
A=0
IF (A==1) THEN
IF ELSE (A==2) THEN
ELSE
END IF
END IF
END PROGRAM Química
```

Fonte: Autor, 2018.

5.2.3 Comandos de Repetição

A estrutura de repetição, ou seja, a capacidade de executar mais de uma vez um determinado conjunto de instruções, é iniciada pelo comando DO (do inglês: fazer) e finalizada pelo comando END DO (do inglês: fim de fazer), conforme representado pela Figura 11.

Figura 11 – Exemplo ilustrado no Plato para um caso de repetição.



Fonte: Autor, 2018.

5.3 Criando um arquivo de saída para os dados gerados no programa

Para a criação de arquivo com dados do programa, foi utilizada a linha de comando a seguir logo após as declarações das variáveis:

```
OPEN (UNIT=10, FILE="dados.txt")
```

O comando *OPEN* (do inglês: abrir) criar um arquivo (FILE) com o nome dados formato .txt na unidade(UNIT) 10. Para fechar esse comando utilizou-se o novo comando *CLOSE*(UNIT=10), conforme representado pela Figura 12.

Figura 12 – Exemplo ilustrado no Plato para a criação de um arquivo de saída.



```
PROGRAM Química
IMPLICIT NONE
REAL A
READ (*, *) A
OPEN (UNIT=10, FILE='dados.txt')
WRITE (10, *) A
CLOSE (UNIT=10)
END PROGRAM Química
```

Fonte: Autor, 2018.

Embora os comandos representados neste capítulo pareçam simples e de fácil execução, vale salientar que a linguagem *FORTRAN* oferece uma série de possibilidades em programação as quais fogem do escopo deste trabalho (REFERÊNCIA CRISTO, H. D. PEREIRA. Programação em Linguagem *Fortran* – Guia básico de programação em Linguagem *Fortran 77* e *90*. Belo Horizonte, 2003).

Os comandos apresentados até o momento correspondem aos principais empregados na linguagem *FORTRAN* os quais foram empregados na elaboração do programa fruto deste trabalho.

6.0 ANÁLISES DOS DADOS E DOS RESULTADOS

Uma vez compreendidos os comandos básicos úteis à programação em linguagem *FORTRAN*, foi desenvolvido um programa apresentado a seguir juntamente com os dados que podem ser obtidos e como podem ser trabalhados junto aos estudantes. Neste caso, o programa foi aplicado ao conteúdo de Cinética Química.

6.1 Formulação Matemática

Para o entendimento do programa e sua utilidade é necessário, primeiramente, conhecer um pouco sobre cinética química, ou seja, um estudo da taxa de velocidade em que reações químicas ocorrem, algumas de forma muito rápida e outras muito lentas dependendo de certos fatores. O programa tem como finalidade o estudo sobre a ordem cinética de uma reação química.

6.1.1 Reações de 1ª Ordem

Considere a reação em que A e B são reagentes:



Essa equação não traz nenhuma informação sobre a ordem da reação. Ao supor que seja de 1ª ordem em [A], tem-se que:

$$v(t) = -\frac{d[A]}{dt} = k[A]$$

Se a concentração de A é $[A]_0$ no tempo zero ($t=0$) e [A] em um tempo t, essa equação pode ser integrada, assim obtendo:

$$\ln \frac{[A]}{[A]_0} = -kt$$

ou por simplificações matemáticas,

$$[A] = [A]_0 e^{-kt}$$

[A] mostra um decaimento exponencial quando o tempo inicial de $[A]_0$ vai a zero, isso pode ser representado pela seguinte expressão:

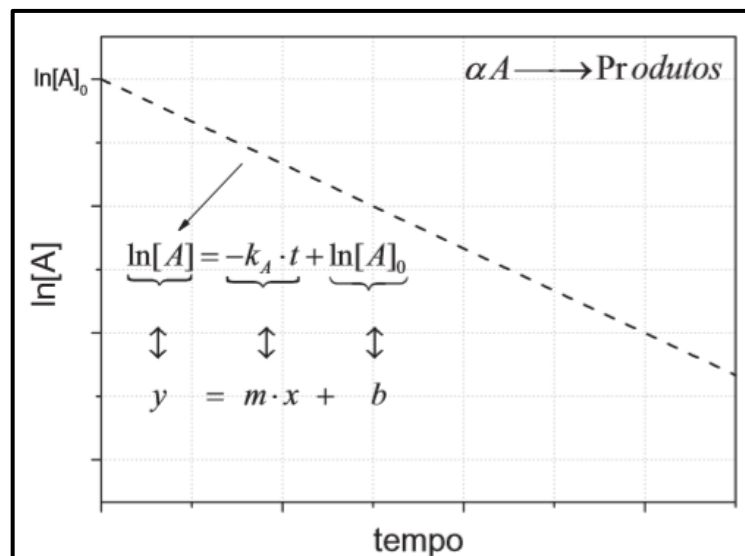
$$\ln[A] = \ln[A]_0 - kt$$

Ao plotar o gráfico do logaritmo neperiano da concentração do reagente A em função do tempo obtém-se uma reta decrescente (Figura 13), onde tem-se o coeficiente angular (-k) e coeficiente linear $\ln[A]_0$ (logaritmo neperiano da concentração inicial de A).

Um conceito importante é o tempo de meia-vida, ou seja, o tempo necessário para que a concentração decaia à metade. Assim, pela equação obtém-se o tempo de meia vida para uma equação de 1ª Ordem como sendo:

$$[A]_t = [A]_0 e^{-kt_{1/2}} \rightarrow \frac{[A]_0}{2} = [A]_0 e^{-kt_{1/2}} \rightarrow e^{kt_{1/2}} = 2 \rightarrow t_{1/2} = \frac{\ln 2}{k}$$

Figura 13– Típica curva de concentração por tempo para uma reação de 1ª ordem.



Fonte: ANGELUCCI, C. A; GODOI G. S; Cinética química. São Cristóvão: Universidade Federal de Sergipe. CESAD,2009.

6.1.2 Reações de 2ª Ordem

Considere a mesma equação utilizada para a reação de 1ª Ordem, se supormos que agora a ordem da reação é 2, temos que:

$$v(t) = -\frac{d[A]}{dt} = k[A]^2$$

Através de métodos matemáticos, a equação transforma-se na seguinte expressão:

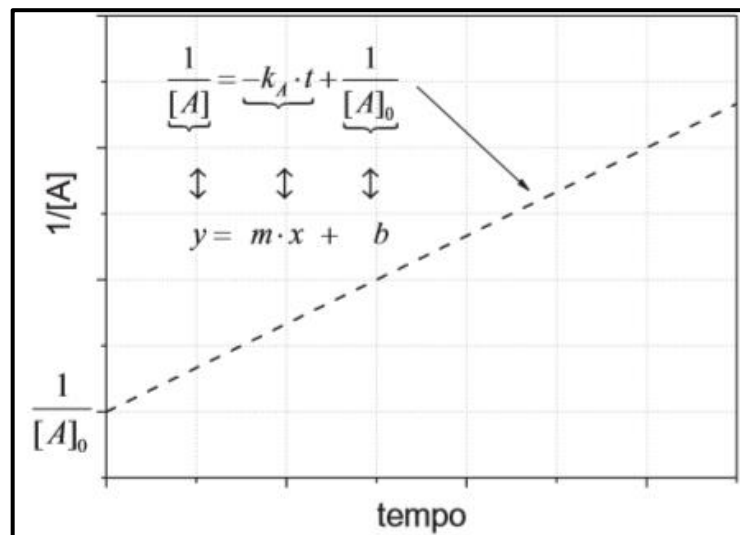
$$\frac{1}{[A]_0} - \frac{1}{[A]_t} = -kt \longrightarrow \frac{1}{[A]_t} = \frac{1}{[A]_0} + kt$$

Ao plotar o gráfico do inverso da concentração do reagente A em função do tempo obtém-se uma reta crescente de coeficiente angular k e coeficiente linear $1/[A]_0$ (inverso da concentração inicial de A). Seu tempo de meia vida é dado pela expressão:

$$t_{1/2} = \frac{1}{k[A]_0}$$

E análise gráfica para esse trabalho é essencial, sendo este o meio de determinar e confirma a ordem da reação estudada. Além disso com ela também é possível determinar a constante k.

Figura 14 – Típica curva de concentração por tempo para uma reação de 2ª Ordem.



Fonte: ANGELUCCI, C. A; GODOI G. S; Cinética química. São Cristóvão: Universidade Federal de Sergipe. CESAD,2009.

6.2 O Programa

O programa criado neste trabalho, tem como função principal estabelecer a

ordem cinética a partir de um conjunto de dados experimentais de concentrações medidos em função do tempo. O mesmo possui um total de 62 linhas, sendo apenas 50 preenchidas por comandos, demonstrando desde já a simplicidade do código.

No primeiro bloco do programa está demarcado seu início, seu nome e suas variáveis, as quais são de tipo reais, inteiras e caracteres (Figura 15).

Figura 15 – Primeiro bloco de linhas de comandos do programas para determinar a cinética de uma reação.

```

program cinetica
implicit none
integer i,j,n
real k, tm, k2, tm2
real, dimension(100,100) :: A, LNA, At, At2
character*100 texto

```

Fonte: Autor, 2018.

No segundo bloco está presente a identificação de um arquivo de entrada de nome concA em extensão .txt, onde devem ser colocados os dados obtidos de concentração em função do tempo. Na sequência está presente a linha de código que permite a criação de um arquivo de saída nomeado lnA_t em extensão .out, onde os novos dados devem ser escritos e utilizados na etapa de análise para decidir sobre qual a cinética do processo (Figura 16).

Figura 16 – Segundo bloco de linhas de comandos do programas para determinar a cinética de uma reação .

```

open(unit=10,file="concA.txt",status="old",action="read")
open(unit=20,file="lnA_t.out",status="replace",action="write")

```

Fonte: Autor, 2018.

No terceiro bloco ocorre a leitura da quantidade de medidas de concentração realizada (n) e dos dados inseridos no arquivo de entrada concA. Tem-se i e j como componentes de uma matriz onde i é a quantidade de linha e j a quantidade de colunas. Assim, i corresponde à quantidade de medidas realizadas e j varia de 1 até 2, que corresponde à quantidade de colunas nesta análise (Figura 17).

Figura 17 – Terceiro bloco de linhas de comandos do programa para determinar a cinética de uma reação.

```

read(10,*) n
read(10,*) texto

do i=1,n
  read(10,*) (A(i,j), j=1,2)
enddo

```

Fonte: Autor, 2018.

No quarto bloco é realizada uma análise supondo se tratar de um comportamento adequado a uma cinética de 1ª Ordem, onde será analisado o logaritmo natural da concentração ($\ln[A]$) em função do tempo (t) (Figura 18). Sendo os dados coerentes a plotagem de um gráfico $\ln[A] \times t$ deverá fornecer uma reta cujo coeficiente angular é igual à constante de velocidade do processo. Neste bloco também está sendo calculada sua constante e tempo de meia vida.

Figura 18 – Quarto bloco de linhas de comandos do programa para determinar a cinética de uma reação.

```

write(20,*) " Dados de Uma Cinetica de Primeira Ordem"
write(20,*) "*****"
write(20,*) "      ln[A]t          t"
write(20,*) "*****"

do i=1,n
  LNA(i,1)=log(A(i,1))
  write(20,*) LNA(i,1), A(i,2)
enddo

k= (-1)*(LNA(n,1)-LNA(1,1))/(A(n,2)-A(1,2))
tm= 0.693/k

write(20,*) "Constante de Velocidade"
write(20,*) k, "min-1"
write(20,*) "Tempo de Meia-Vida"
write(20,*) tm, "min"

```

Fonte: Autor, 2018.

No quinto bloco é apresentada uma análise supondo que o comportamento é adequado a uma cinética de 2ª Ordem, onde será analisado o inverso da concentração

($1/[A]$) em função do tempo (t) (Figura 19). Sendo os dados coerentes, a plotagem de um gráfico $1/[A] \times t$ deve fornecer uma reta. Neste bloco também está sendo calculada sua constante e tempo de meia vida.

Figura 19 – Quinto bloco de linhas de comandos do programa para determinar a cinética de uma reação.

```

write(20,*) "*****"
write(20,*) "      [A]t          t"
write(20,*) "*****"

do i=1,n
  At(i,1)=A(1,1)*exp(-k*A(i,2))
  write(20,*) At(i,1), A(i,2)
enddo

write(20,*) "
write(20,*) "*****"
write(20,*) "Dados de Uma Cinetica de Segunda Ordem"
write(20,*) "*****"
write(20,*) "      1/At          t"
write(20,*) "*****"

do i=1,n
  At2(i,1)=1/A(i,1)
  write(20,*) At2(i,1), A(i,2)
enddo

k2= (At2(n,1)-At2(1,1))/(A(n,2)-A(1,2))
tm2 = 1/k2*(A(1,1))
write(20,*) "Constante de Velocidade"
write(20,*) k2, "M-1.min-1"
write(20,*) tm2, "min"

end program

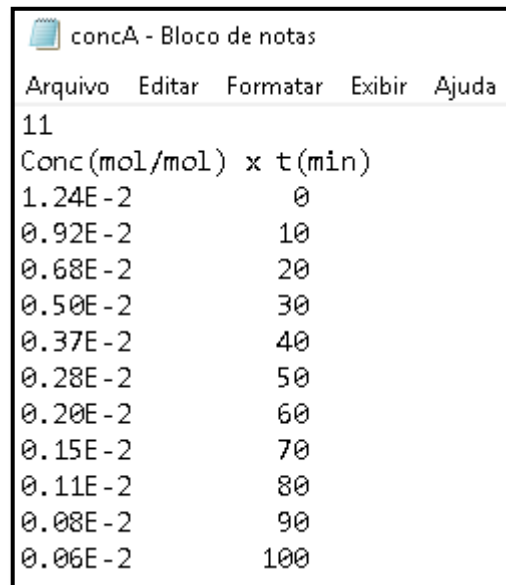
```

Fonte: Autor, 2018.

6.3 Utilizando o Programa

Para utilizar o programa, o primeiro passo é criar e preencher o arquivo de entrada (conca), onde serão inseridos os dados a serem analisados. A primeira coluna corresponde às concentrações medidas em mol L^{-1} e a segunda coluna corresponde aos tempos em minutos nos quais as concentrações foram medidas (Figura 20).

Figura 20 – Arquivo concA preenchido corretamente para leitura do programa.

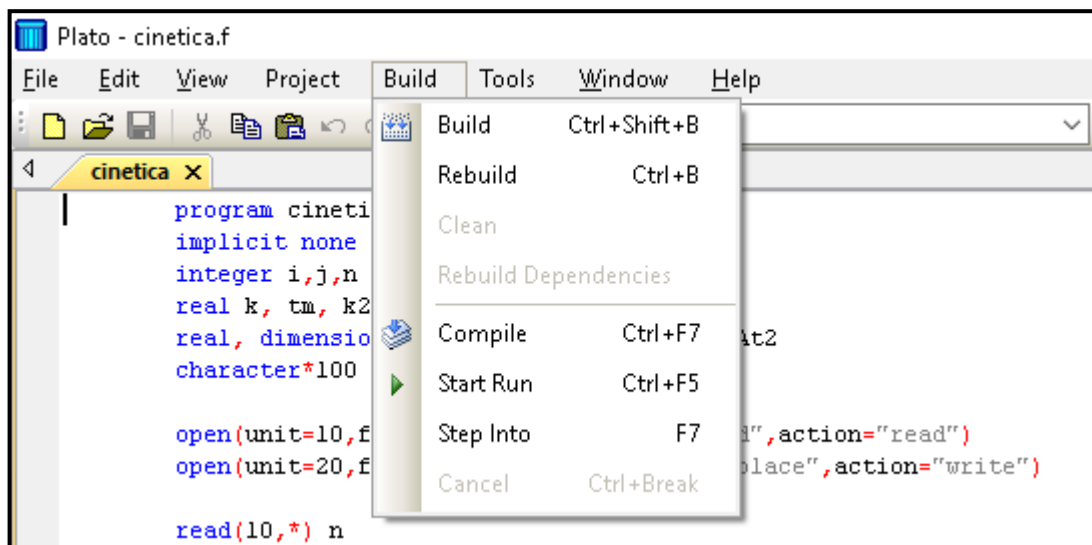


Conc (mol/mol)	t (min)
1.24E-2	0
0.92E-2	10
0.68E-2	20
0.50E-2	30
0.37E-2	40
0.28E-2	50
0.20E-2	60
0.15E-2	70
0.11E-2	80
0.08E-2	90
0.06E-2	100

Fonte: Autor, 2018.

Na etapa seguinte, basta abrir o programa já inscrito no Plato. Em sua barra de tarefas, deve-se selecionar a opção *BUILD* (do inglês: construir) e, em seguida, escolher a opção *START RUN* (do inglês: iniciar corrida) para dar início ao programa, conforme representado pela Figura 21.

Figura 21 – Trajeto para a execução do programa na ferramenta plato.



Fonte: Autor, 2018.

Se estiver tudo correto o Plato deve exibir uma tela conforme a Figura 22:

Figura 22 – Tela gerada após execução do programa seguindo o passo da Figura 21.



Fonte: Autor, 2018.

6.4 Dados Obtidos

Para esse estudo foram utilizados dados reais da reação de decomposição do N_2O_5 gasoso a uma temperatura de 318K obtidos do livro “*Physical Chemistry: A Molecular Approach*” (SIMON, 1997), conforme representação a seguir (Eq. 1 e Tabela 1).

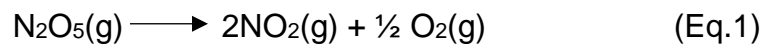


Tabela 2 – Dados reais utilizados para realizar o teste com o programa criado neste trabalho.

Concentração (Mol/L)	Tempo (Minutos)
1.24E-2	0
0.92E-2	10
0.68E-2	20
0.50E-2	30
0.37E-2	40
0.28E-2	50
0.20E-2	60
0.15E-2	70
0.11E-2	80
0.08E-2	90
0.06E-2	100

Fonte: SIMON, 1997.

Constante (k): $3,04 \times 10^{-2} \text{ min}^{-1}$.

Tempo de meia vida: 22,81.

As Figuras 23 a 24 representam os valores obtidos e utilizados no programa desenvolvido, o perfil gráfico da relação de concentração do reagente A em função do tempo antes e após a linearização através de escala logarítmica, respectivamente.

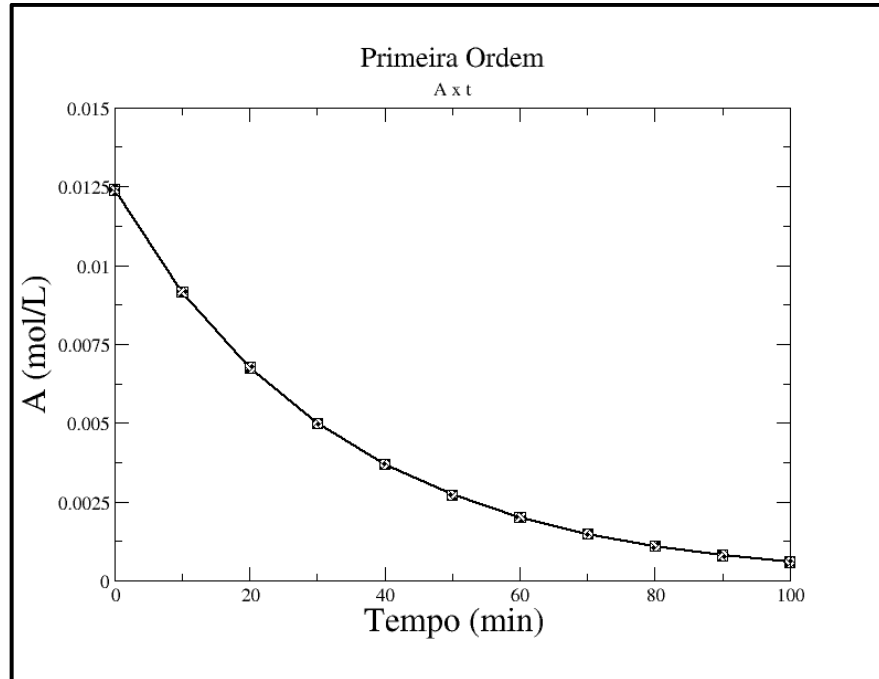
Estão expressos na Figura 23, para cinética de 1ª Ordem e na Figura 26 para cinética de 2ª Ordem.

Figura 23 – Valores obtidos no programa elaborado em linguagem *Fortran* para cinética de 1ª ordem.

Dados de Uma Cinetica de Primeira Ordem	
ln[A]t	t
-4.39006	0.00000
-4.68855	10.0000
-4.99083	20.0000
-5.29832	30.0000
-5.59942	40.0000
-5.87814	50.0000
-6.21461	60.0000
-6.50229	70.0000
-6.81245	80.0000
-7.13090	90.0000
-7.41858	100.000
Constante de Velocidade	
3.028522E-02min-1	
Tempo de Meia-Vida	
22.8824	min

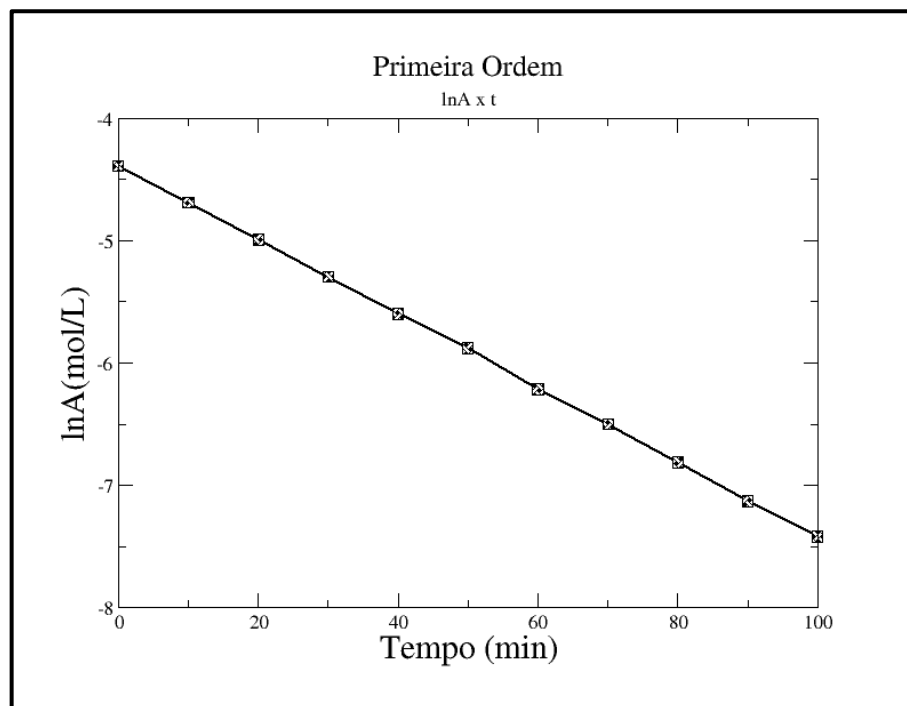
Fonte: Autor, 2018.

Figura 24 – Perfil gráfico de uma cinética de 1ª Ordem para a relação da concentração de A em função de um tempo t.



Fonte: Autor, 2018.

Figura 25 – Linearização da relação de concentração de A em função do tempo por escala logarítmica para uma reação de 1ª Ordem construído pela ferramenta gráfica xmgrace.



Fonte: Autor, 2018.

Após essa análise fica evidente que a cinética para os dados obtidos no

programa é de 1ª Ordem, estes valores podem ser confirmados na literatura onde os dados foram retirados (SIMON, 1997).

O gráfico expresso na Figura 25, representa de forma satisfatória uma cinética de 1ª Ordem, a qual deve ser uma reta decrescente com o passar do tempo(t), confirmando assim sua veracidade.

As Figuras 26 e 27 representam os valores obtidos e utilizados no programa desenvolvido e o perfil gráfico da relação de concentração do reagente A em função do tempo, respectivamente.

Figura 26 – Valores obtidos no programa elaborado em linguagem *Fortran* para cinética de 2ª ordem.

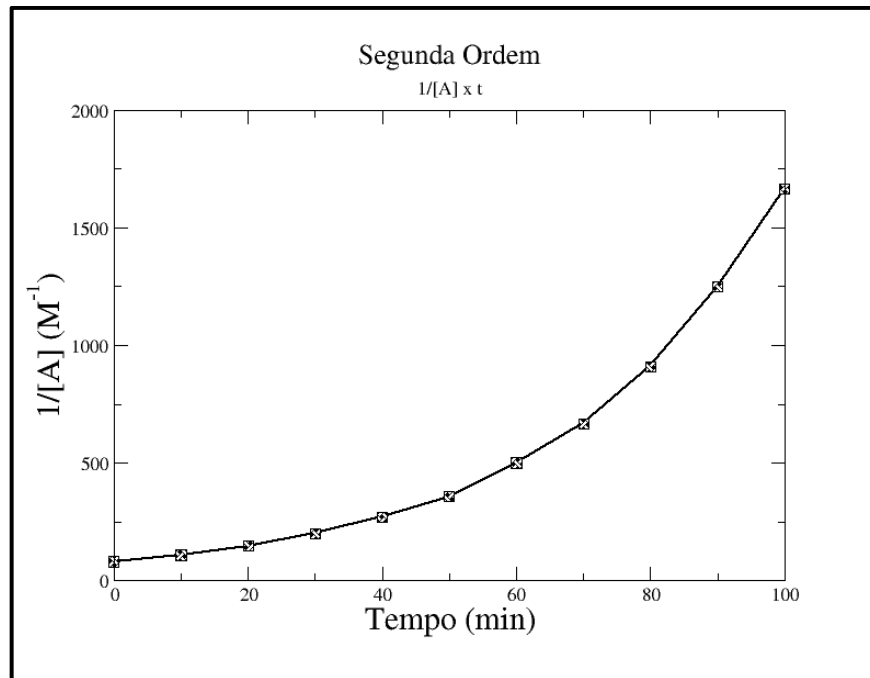
```

*****
Dados de Uma Cinetica de Segunda Ordem
*****
  1/At          t
*****
  80.6452      0.00000
  108.696     10.0000
  147.059     20.0000
  200.000     30.0000
  270.270     40.0000
  357.143     50.0000
  500.000     60.0000
  666.667     70.0000
  909.091     80.0000
  1250.00     90.0000
  1666.67    100.000
Constante de Velocidade
  15.8602    M-1.min-1
  7.818306E-04min

```

Fonte: Autor, 2018.

Figura 27– Perfil gráfico de uma cinética de 2ª Ordem para a relação da concentração de A em função de um tempo t. construído pela ferramenta gráfica xmgrace.



Fonte: Autor, 2018.

Após essa análise fica evidente que a cinética para os dados colhidos é condizente com um processo de 1ª ordem, o que vai de acordo com a literatura de onde os dados foram retirados. Como mostrado na Figura 24, a variação da concentração de reagente com o tempo segue um comportamento do tipo exponencial, o que caracteriza uma cinética de 1ª ordem.

Usualmente, a análise cinética é facilitada quando se faz uma tentativa de linearização dos dados medidos. A Figura 25 representa linearização do tipo $\ln A \times t$, mostra uma dependência linear decrescente a qual é condizente com uma cinética de 1ª ordem. Os dados da Figura 25 foram utilizados para determinar a constante de velocidade e o tempo de meia-vida assumindo uma cinética de 1ª ordem. Os valores da constante de velocidade e tempo de meia vida obtidos pelo programa foram $3,028 \times 10^{-2} \text{ min}^{-1}$ e 22,882 min, respectivamente, os quais estão em ótimo acordo com os valores de $3,04 \times 10^{-2} \text{ min}^{-1}$ e 22,81 min determinados experimentalmente.

Para confirmar de fato que a cinética do processo é de 1ª ordem, é necessário eliminar a hipótese de que o mesmo seja de 2ª ordem. A Figura 27 mostra a variação do inverso da concentração do reagente com o tempo. O esperado pela teoria é que a dependência de $1/[A] \times t$ apresente uma dependência linear crescente, cujo coeficiente linear da reta é igual a constante de velocidade. A Figura 27 mostra claramente que os dados de $1/[A] \times t$ para a decomposição do $\text{N}_2\text{O}_5(\text{g})$ não segue um

comportamento linear, indicando que este processo não é de 2ª ordem.

6.5 Proposta de Ementa para uma disciplina de programação em química

A partir do desenvolvimento deste trabalho, foi feita a proposta de uma disciplina a ser integrada ao projeto pedagógico do curso de Licenciatura em Química da UFAL conforme a seguir:

Nome da disciplina: Introdução à programação aplicada à Química.

Curso: Química Licenciatura.

Carga horaria: 30 horas.

Semestre recomendado: 3º semestre.

Conhecimento base recomendado: Informática básica, Química geral 1 e 2.

Modelo de Ensino: Presencial.

Objetivos da disciplina:

Introdução a conceitos básicos para o estudo da Química computacional.

Competências a Serem Desenvolvidas:

Capacidade de estabelecer o algoritmo mais adequado à resolução de um problema e utilizá-lo na resolução de alguns problemas simples em Química.

Cronograma da disciplina:

- Histórico e fundamentos da Química computacional
- Representação da linguagem maquina;
- Folhas de cálculos e aplicações em química;
- Programação em linguagem simples;
- Resolução de problemas químicos;
-

Método de Ensino:

Aulas teóricas onde serão ministrados os conceitos básicos e introdutórios e aulas de laboratório computacional onde será feita a implementação dos algoritmos e soluções de problemas propostos.

Método de Avaliação:

50% solução dos exercícios propostos;

50% exame teórico.

7. CONCLUSÃO

A realização deste trabalho evidenciou a necessidade da inclusão digital de alunos da graduação dos cursos de Química da UFAL. A Química computacional para o Instituto de Química e Biotecnologia tem grande importância no desenvolvimento das pesquisas, havendo uma necessidade de aguçar os estudantes da graduação, em destaque os alunos do curso de licenciatura a participarem, assim pretendeu-se alcançar o corpo discente com esse trabalho, alertando sobre a necessidade de inclusão digital dos alunos, afim de torna o meio acadêmico mais informatizado.

A criação do programa simples em linguagem Fortran permite inferir que se trata de ferramenta simples e acessível ao graduando de química podendo ser explorada por qualquer estudante no curso para qualquer disciplina. Encontrar *softwares* na internet não é uma tarefa difícil, o grande ápice da criação de um programa de sua autoria, está em trabalhar sua capacidade lógica e cognitiva. Foram exibidos todos os passos básicos para a criação de um programa, mas cabe ao usuário definir sua utilidade, seja como ferramenta de apoio, ou apenas um facilitador. Vale ressaltar que precisa-se cada vez mais de *softwares* flexíveis que permitam professores e alunos criarem ferramentas que se adequem a processos de ensino coletivos e individuais.

A disciplina proposta é passível de ser inserida no projeto pedagógicos dos cursos de Química da UFAL, na qualidade de eletiva ou obrigatória desde que haja uma disciplina preparatória como requisito básico de formação computacional. A proposta/recomendação da disciplina de Introdução à programação aplicada à Química, deve sanar o atraso que o curso de química sofre em relação a formação superior em outros estabelecimentos de ensino onde o mesmo é fornecido. Garantindo base necessária para o desenvolvimento dos alunos, possibilitando buscar novos métodos para soluções mais eficientes que contribuam para a química e seu estudo significativamente.

8. REFERÊNCIAS

ANGELUCCI, C. A; GODOI G. S; Cinética química. São Cristóvão: Universidade Federal de Sergipe. CESAD,2009. Acessado em: 08/01/2019. Disponível em: http://www.cesadufs.com.br/ORBI/public/uploadCatalogo/11400916022012Cinetica_Quimica_Aula_1.pdf.

ARROIO, A; GIORDAN, M; QNEsc, nº24, novembro, 2006; 8-11.

BORDINI, A; AVILA, C. M. O; WEISSHAHN, Y; CUNHA, M. M; CAVALHEIRO, S. A. C; FOSS, L; AGUIAR, M. S; REISER, R. H. S; Computação na Educação Básica no Brasil: o Estado da Arte. Universidade Federal de Pelotas, 2016.

UNIVERSIDADE DE COIMBRA, UC. Departamento de Química. Programação em *Fortran* 90 - Informática geral, 2010/2011. Acessado em:08/01/2019 Disponível em: <https://drive.google.com/open?id=1SXtWQJ7T8fjzkNKai3vTaDfJ7Squley>.

COSTA, R.G., PASSERINO, L.M. e ZARO, M.A. Fundamentos teóricos do processo de formação de conceitos e suas implicações para o ensino e aprendizagem de Química. Ensaio Pesquisa em Educação em Ciências, v.14, n.1, p.271-281. 2012. Acessado em:19/01/2019. Disponível em: <http://www.redalyc.org/articulo.oa?id=129523627018>.

CRISTO, H. D. PEREIRA. Programação em Linguagem *Fortran* – Guia básico de programação em Linguagem *Fortran* 77 e 90. Belo Horizonte, 2003. Acessado em: 08/01/2019. Disponível em: <https://www.inf.ufes.br/~thomas/fortran/tutorials/helder/fortran.pdf>.

SIMON, J. D; Physical Chemistry: A Molecular Approach, cap. 26, página 1056. University Science Books 1997.

LOCATELLI, A; ZOCH, A. N; TRENTIN, M. A. S; TICs no Ensino de Química: Um Recorte do “Estado da Arte”. Revista Tecnologias na Educação. Ano 7 – Ed. 8, 2005.

MARCONI, Marina de Andrade. Metodologia do Trabalho Científico. 7º Edição. São Paulo: Atlas, 2011.

PEREIRA, D.I.S; *Softwares* educacionais no ensino de química. Campina Grande/PB. 2014. Acessado em: 19/01/2019. Disponível em: <http://dspace.bc.uepb.edu.br/jspui/bitstream/123456789/13620/1/PDF%20-%20Deydeby%20Illan%20dos%20Santos%20Pereira.pdf>.

SILVERFROST FTN95, Plato: *Fortran* for Windows. Acessado em: 24/07/2018. Disponível em: <https://www.silverfrost.com/>.

SILVA, Antônio Carlos Ribeiro. Metodologia da Pesquisa Aplicada À Contabilidade. 2º Edição. São Paulo: Atlas, 2006.

SOUSA, Bruno; JUNIOR, José e FORMIGA, Andrei. Introdução Programação Editora da UFPB, João Pessoa, 2014. Acessado em: 24/07/2018. Disponível em: http://biblioteca.virtual.ufpb.br/files/introducao_a_programacao_1463150047.pdf.

SOUZA, M. A. F; GOMES, M. M; SOARES, M.V; Algoritmos e lógica de programação: um texto introdutório para a engenharia. 2.ed rev. Cengage Learning. São Paulo,2011.

RAUPP, D; SERRANO, A; MARTINS, T.L.C; A evolução da química computacional e sua contribuição para a educação em Química. Revista Liberato, Novo Hamburgo, v. 9, n. 12, p. 13-22, jul./dez. 2008. Acessado em: 20/01/2019. Disponível em: http://www.liberato.com.br/sites/default/files/arquivos/Revista_SIER/v.%209%2C%20n.%2012%20%282008%29/2.%20A%20evolu%20E7%E3o%20da%20qu%20EDmica%20computacional%20e%20sua.pdf.

TOLETO, Claudio F.M; DIAS, Maurício A. IDES E PROGRAMAÇÃO. Acessado em 27/07/2018. Disponível em: https://edisciplinas.usp.br/pluginfile.php/349201/mod_resource/content/1/%20IDE%20e%20Programação.pdf.

XMGRACE. Grace Development Team 1996-2008. Acessado em:08/01/2019.

Disponível em: <http://plasma-gate.weizmann.ac.il/Grace/>.