

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO**

**VISEDU - QUÍMICA: VISUALIZADOR DE MATERIAL**  
**EDUCACIONAL, MÓDULO DE QUÍMICA**

**ARTHUR HENRIQUE EGGERT**

**BLUMENAU**  
**2015**

**2015/2-05**

**ARTHUR HENRIQUE EGGERT**

**VISEDU - QUÍMICA: VISUALIZADOR DE MATERIAL  
EDUCACIONAL, MÓDULO DE QUÍMICA**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Dalton Solano dos Reis, M. Sc. - Orientador

**BLUMENAU  
2015**

**2015/2-05**

**VISEDU - QUÍMICA: VISUALIZADOR DE MATERIAL  
EDUCACIONAL, MÓDULO DE QUÍMICA**

Por

**ARTHUR HENRIQUE EGGERT**

Trabalho de Conclusão de Curso aprovado  
para obtenção dos créditos na disciplina de  
Trabalho de Conclusão de Curso II pela banca  
examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Dalton Solano dos Reis, M. Sc. – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Alexandre Roberto Valdameri, M. Sc. – FURB

Membro: \_\_\_\_\_  
Prof. Aurélio Faustino Hoppe, M. Sc. – FURB

Blumenau, 09 de dezembro de 2015

Dedico este trabalho aos meus familiares e todos os meus amigos.

## **AGRADECIMENTOS**

Agradeço primeiramente aos meus pais, Moacir Henrique Eggert e Irani Meister Eggert, por me apoiarem durante a faculdade, agradeço ainda a minha namorada Samira da Rocha pelo apoio, paciência e auxílio no desenvolvimento deste trabalho. E por fim agradeço ao professor Dalton Solano do Reis, por acreditar em mim e pelo auxílio prestado.

O bom senso, que um dia foi professor, agora, em nossas escolas, está totalmente morto. A ciência sua filha, o matou para ver como ele era feito.

Antônio Gramsci

## RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação web para o auxílio de alunos e professores no processo de ensino e aprendizagem de química, mais especificamente do conceito de geometria molecular. A aplicação tem como principal objetivo o desenho e validação da fórmula estrutural de uma molécula com um átomo central e a sua representação em um ambiente tridimensional. Para que isso seja possível à aplicação conta com quatro áreas principais, caixa de entrada da fórmula química, a lista de elementos onde os elementos, a área de desenho e a área de visualização. Para a criação dos objetos na área de desenho é utilizado o *framework* Fabric.js, que facilita a criação e manipulação de objetos em um elemento `canvas`, já para a criação dos objetos tridimensionais faz-se o uso do Three.js por intermédio do motor de jogos desenvolvido por Montibeler (2014). Este motor de jogos permite uma arquitetura de componentes facilitando assim sua utilização e possível extensão. Foram executados testes da aplicação com alunos de primeiro e segundo ano do ensino médio de uma escola particular da região de Blumenau.

Palavras-chave: Computação gráfica. Química. Geometria molecular.

## **ABSTRACT**

This paper presents the development of a web application for the assistance of students and teachers in the teaching and learning of chemistry, specifically the concept of molecular geometry. The application's main objective is the design and validation of the structural formula of a molecule with a central atom and its representation in a three dimensional environment. For this to be possible the application has four main areas, the input of the chemical formula, the list of elements where the elements in the drawing area and the viewing area. For the creation of objects in the drawing area is used Fabric.js framework that facilitates the creation and manipulation of objects in a canvas element, as for the creation of three-dimensional objects is done using Three.js through the engine games developed by Montibeler (2014). This game engine provides a component architecture thereby facilitating their use and possible extension. Application tests were performed with first and second year students of high school of a private school in Blumenau region.

**Key-words:** Computer graphics. Chemistry. Molecular geometry.

## LISTA DE FIGURAS

Figura 1 – Tabela Periódica dos Elementos .....	19
Figura 2 – Estrutura de Lewis da Molécula F <sub>2</sub> .....	21
Figura 3 – Geometrias Moleculares.....	22
Figura 4 – Pipeline do WebGL.....	26
Figura 5 – Ambiente 3D de edição .....	28
Figura 6 - Ambiente 2D de edição.....	29
Figura 7 – Diagrama de casos de uso .....	31
Figura 8 – Diagrama de classes .....	32
Figura 9 – Diagrama de classe completo.....	33
Figura 10 – Diagrama de classes completo .....	36
Figura 11 – Diagrama de Sequência.....	38
Figura 12 – Tela principal da aplicação.....	47
Figura 13 – Erro ao executar parse da fórmula química.....	48
Figura 14 – Parse da fórmula química sem erros .....	48
Figura 15 – Molécula de água (H <sub>2</sub> O).....	49
Figura 16 – Visualização tridimensional .....	49
Figura 17 – Gráfico dos resultados obtidos na Questão 1 .....	52
Figura 18 – Gráfico dos resultados obtidos na Questão 2 .....	52
Figura 19 – Gráfico dos resultados obtidos na Questão 3 .....	53
Figura 20 – Gráfico dos resultados obtidos na Questão 4.....	54
Figura 21 – Gráfico dos resultados obtidos na Questão 5 .....	54
Figura 22 – Gráfico dos resultados obtidos na Questão 6 .....	55
Figura 23 – Gráfico dos resultados obtidos na Questão 7 .....	56
Figura 24 – Formulário respondido por uma dos estudantes entrevistados .....	68
Figura 25 – Formulário respondido por uma dos estudantes entrevistados .....	69
Figura 26 – Formulário respondido por uma dos estudantes entrevistados .....	70
Figura 27– Formulário respondido por uma dos estudantes entrevistados .....	71
Figura 28– Formulário respondido por uma dos estudantes entrevistados .....	72
Figura 29– Formulário respondido por uma dos estudantes entrevistados .....	73
Figura 30 - Diagrama de Classes do motor de jogos.....	74

## LISTA DE QUADROS

Quadro 1– Tipos de Ligações Químicas.....	20
Quadro 2– Relação entre os pares e o tipo de geometria.....	23
Quadro 3 – Three.js compatibilidade entre navegadores .....	27
Quadro 4 – Relação de Requisitos Funcionais e Não Funcionais .....	30
Quadro 5– Classificação dos grupos e grandes grupos da tabela periódica .....	34
Quadro 6 – Método click da caixa de entrada de <i>submit</i> da fórmula .....	40
Quadro 7 – Método de <i>Parse</i> da Fórmula Química.....	41
Quadro 8 – Método para adicionar átomo na molécula.....	42
Quadro 9 – Método para a seleção do modo de ligação a ser utilizado .....	42
Quadro 10 – Métodos <i>onMouseUp()</i> e <i>onMouseDown()</i> .....	43
Quadro 11 – Identificação dos Tipos de Ligação .....	44
Quadro 12 – Cálculo da Regra do Octeto.....	45
Quadro 13 – Método <i>valida()</i> .....	46
Quadro 14 – Cálculo para seleção do próximo ponto .....	47
Quadro 15 – Perguntas aplicadas aos usuários.....	51
Quadro 16 – Testes de Memória e Desempenho no Google Chrome .....	57
Quadro 17 – Testes de Memória e Desempenho no Firefox .....	57
Quadro 18 – Gráfico de processamento de FPS da aplicação.....	58
Quadro 19 – Gráfico de uso de memória da aplicação.....	58
Quadro 20 – Testes de Memória e Desempenho no Google Chrome .....	59
Quadro 21 – Testes de Memória e Desempenho no Firefox .....	59
Quadro 22 – Gráfico de processamento de FPS da aplicação.....	60
Quadro 23 – Gráfico de uso de memória da aplicação.....	60
Quadro 24 – Características dos trabalhos correlatos e do trabalho desenvolvido. ....	60
Quadro 25 – Descrição do caso de uso UC01 .....	65
Quadro 26 – Descrição do caso de uso UC02 .....	65
Quadro 27 – Descrição do caso de uso UC03 .....	65
Quadro 28 – Descrição do caso de uso UC04 .....	66
Quadro 29 – Descrição do caso de uso UC05 .....	66
Quadro 30 – Descrição do caso de uso UC06 .....	67
Quadro 31 – Interação com o ambiente tridimensional.....	67



## LISTA DE ABREVIATURAS E SIGLAS

API – *Application Programming Interface*

CML – *Chemical Markup Language*

DOM – *Document Object Model*

FPS – *Frames por Segundo*

GPU – *Graphics Processor Unit*

HTML – *Hyper Text Markup Language*

HTML5 – *Hyper Text Markup Language Versão 5*

InChI – *International Chemical Identifier*

IUPAC – *International Union of Pure and Applied Chemistry*

OpenGL – *Open Graphics Library*

OpenGL ES – *Open Graphics Library for Embedded Systems*

SVG – *Scalable Vector Graphics*

UML – *Unified Modeling Language*

VBO – *Vertex Buffer Object*

W3C – *World Wide Web Consortium*

WebGL – *Web Graphics Library*

WHATWG – *Web Hypertext Application Technology Working Group*

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>14</b>
1.1 OBJETIVOS.....	15
1.2 ESTRUTURA.....	15
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>17</b>
2.1 INFORMÁTICA NA EDUCAÇÃO .....	17
2.2 QUÍMICA.....	18
2.2.1 Fórmulas.....	19
2.2.2 Estrutura de Lewis.....	20
2.3 WEBGL.....	23
2.3.1 Pipeline do WebGL.....	25
2.3.2 Three.JS.....	26
2.3.3 Fabric.JS.....	27
2.4 TRABALHOS CORRELATOS.....	27
2.4.1 Avogadro.....	27
2.4.2 BKChem.....	28
<b>3 DESENVOLVIMENTO .....</b>	<b>30</b>
3.1 REQUISITOS.....	30
3.2 ESPECIFICAÇÃO .....	30
3.2.1 Diagrama de casos de uso .....	30
3.2.2 Diagrama de pacotes e classes .....	31
3.2.3 Diagrama de Sequência.....	37
3.3 IMPLEMENTAÇÃO .....	39
3.3.1 Técnicas e ferramentas utilizadas.....	39
3.3.2 O visualizador de material educacional (VisEdu-Química).....	40
3.3.3 Operacionalidade da implementação .....	47
3.4 RESULTADOS E DISCUSSÕES.....	49
3.4.1 Testes da aplicação.....	50
3.4.2 Pesquisa de opinião sobre a usabilidade .....	50
3.4.3 Teste de desempenho .....	56
3.4.4 Comparativo entre trabalho desenvolvido e seus correlatos .....	60
<b>4 CONCLUSÕES.....</b>	<b>62</b>

4.1 EXTENSÕES .....	62
APÊNDICE A – DETALHAMENTO DOS CASOS DE USO DO MOTOR.....	65
APÊNDICE B – FORMULÁRIOS SOBRE A EXPERIÊNCIA DE UTILIZAÇÃO DO VISEDU-QUÍMICA .....	68
ANEXO A – DIAGRAMA DE CLASSES DO MOTOR DE JOGOS DESENVOLVIDO POR MONTIBELER .....	74

## 1 INTRODUÇÃO

Atualmente a humanidade vive em processo acelerado de modificações e rupturas, que se reflete em todos os setores da sociedade. Assim sendo, a educação e a informação assumem papel significativo neste processo (CARVALHO, 1997). Segundo Silva (2011), das disciplinas ministradas tanto no ensino fundamental como no ensino médio, a química é citada pelos alunos como uma das mais difíceis e complicadas de se estudar, e que sua dificuldade aumenta por conta de ser abstrata e complexa.

Percebe-se que os alunos muitas vezes não conseguem aprender, por falharem em associar o conteúdo estudado com seu cotidiano, tornando-os desinteressados pelo tema (NUNES; ARDONI, 2010). No entanto, nem sempre o professor está preparado para atuar de forma interdisciplinar, relacionando o conteúdo com a realidade dos alunos (LOBATO, 2007).

O uso da informática por professores e alunos ajuda a disseminar oportunidades e estimula-os em extrair e aproveitar o máximo de seus talentos naturais, desenvolvendo a consciência histórica, compreendida enquanto o domínio progressivo e científico (VINCINGUERA, 2002). Com a expansão da internet em escala mundial, passou-se a analisar o potencial de seus recursos para o enriquecimento das técnicas educacionais conhecidas até o momento (BAPTISTA, 2004).

Citam-se como principais facilitadores de acesso à internet os navegadores, que devido a sua evolução nos últimos quinze anos, rodam em diversos tipos de hardware, de computadores a celulares e *tablets*. Os documentos são geralmente escritos utilizando *Hyper Text Markup Language* (HTML), o que permite incorporar *links* para outros documentos ou para lugares diferentes do mesmo documento (GROSSKURTH; GODFREY, 2006).

De forma mais atual verifica-se que os navegadores também permitem a criação de representações em um ambiente 3D utilizando recursos gráficos da *Graphics Processing Unit* (GPU) e com auxílio da *Application Programming Interface* (API) WebGL (DIRKSEN, 2013). Em particular usando o HTML, versão cinco (HTML5), que pode ser considerado o mais recente padrão para HTML. Trata-se de uma cooperação entre o *World Wide Web Consortium* (W3C) e a *Web Hypertext Application Technology Working Group* (WHATWG). Foi especialmente concebido para proporcionar um conteúdo rico, sem a necessidade de *plugins* adicionais. A versão atual oferece desde animações para gráficos, músicas para filmes e também pode ser usado para construir aplicações web. Os principais navegadores (Chrome, Firefox, Internet Explorer, Safari e Opera) suportam as novas APIs e os novos elementos do

HTML5, e continuam a adicionar os novos recursos a suas últimas versões (W3SCHOOLS, 2015).

Diante do exposto acima, verifica-se que a necessidade de ferramentas dinâmicas para o auxílio da educação são mais que necessárias, tanto para os alunos como para os professores. Motivado por este fator, neste trabalho é desenvolvido o aplicativo VisEdu-Química para a plataforma web. O aplicativo explora diversos recursos dos navegadores modernos para tornar a experiência do usuário mais interativa. As principais funcionalidades desenvolvidas se dão pela validação da fórmula química utilizada, assim como a validação das moléculas desenhadas e a geração de uma visualização tridimensional da molécula.

## 1.1 OBJETIVOS

O objetivo deste trabalho é a implementação de uma ferramenta para o auxílio de alunos e professores, especialmente os do ensino fundamental e médio, no processo de ensino e aprendizagem da química, utilizando a API WebGL, juntamente com a especificação do HTML5.

Os objetivos específicos do trabalho são:

- a) criar um ambiente 2D para a modelagem de moléculas em suas fórmulas estruturais completas;
- b) validar a molécula criada no ambiente 2D;
- c) criar uma visualização 3D da molécula.

## 1.2 ESTRUTURA

Este trabalho está estruturado em quatro capítulos, sendo que o primeiro deles contém uma introdução ao tema e em seguida são apresentados os objetivos e a estrutura do trabalho. O segundo capítulo apresenta a fundamentação teórica necessária para o entendimento deste. Inicialmente é apresentada uma visão da informática na educação, da influência da mesma no ensino da química.

No capítulo três são apresentadas as etapas de desenvolvimento deste trabalho. Primeiramente são apresentados os principais requisitos da aplicação. Posteriormente é exposta a especificação da aplicação, onde são descritos os casos de uso e os diagramas de classe e de sequência. Na especificação também é discutido como ocorreu a implementação do trabalho a partir das técnicas e ferramentas utilizadas, onde se pode visualizar alguns trechos de código. O capítulo é finalizado com a exibição dos resultados e das discussões ocorridas durante o processo de desenvolvimento da aplicação.

Por fim, o capítulo quatro descreve as conclusões deste trabalho e apresenta sugestões para extensões futuras.

## 2 FUNDAMENTAÇÃO TEÓRICA

Na seção 2.1 é apresentada uma visão da informática na educação, onde também será discutida a influência da informática no ensino da química. A seção 2.2 apresenta conceitos relacionados à Química. Na seção 2.3 são definidas as principais características da biblioteca multiplataforma WebGL. Por fim, na seção 2.4 são relacionados os trabalhos correlatos ao desenvolvido.

### 2.1 INFORMÁTICA NA EDUCAÇÃO

A escola deve aproveitar o momento de inovações tecnológicas e modernizar suas práticas e propostas de ensino e aprendizagem, tanto na forma quanto no conteúdo, atendendo às novas necessidades impostas pelo mundo dinâmico e globalizado (AMARAL; AMARAL, 2008). Sendo a escola considerada, tradicionalmente, uma fonte de cultura e conhecimento, as novas diretrizes a colocam também como fonte de competências que devem ser adquiridas ou reconhecidas e desenvolvidas (ALARCÃO, 2003, p.12). E dentre essas competências se encontra o uso da informática.

Diante disso, é importante colocar que o uso da informática, só faz sentido quando o professor a entende como forma de auxílio, como elemento motivador que provoca o surgimento de novas práticas pedagógicas, no desenvolvimento das atividades, dinamizando assim o processo de ensino e aprendizagem. Porém, a utilização da informática em sala de aula exige a preparação do professor, pois o mesmo deve ser capaz de elaborar estratégias que atraiam os estudantes para o conteúdo a ser lecionado, para que aconteça assim, a construção do conhecimento, distanciando-se da ideia da utilização de softwares ou programas multimídias apenas voltadas para a visualização sequencial de textos, figuras ou vídeos (MEDEIROS, 2008).

A química foi muito beneficiada com o uso dos computadores, pois se trata de uma ciência bastante experimental e a visualização de processos é de suma importância para assimilação do conteúdo (FERREIRA, 1998). O problema de compreensão da química pelos alunos poderia ser parcialmente resolvido com a utilização de softwares específicos. Por exemplo: software para demonstração de moléculas em três dimensões e jogos educativos dão novos significados às tarefas de ensino, atendendo as propostas ditadas para a nova educação (BONA, 2009).

## 2.2 QUÍMICA

Para Atkins e Jones (2001) a química é a ciência que estuda a matéria e as mudanças que esta sofre, sendo assim, a química está ao nosso redor, todo o tempo. Ainda segundo os autores, a química é uma ciência de três níveis: o primeiro nível, macroscópico, trata da matéria e de suas transformações, onde se podem verificar as mudanças ocorridas. O segundo nível, microscópico, trata de eventos que não podem ser diretamente observadas, neste ponto, a química interpreta essas mudanças como um rearranjo dos átomos. Já o terceiro nível, simbólico, trata da descrição destes fenômenos por intermédio de símbolos e fórmulas, sendo este o nível que mantém unidos os outros dois. Toda matéria é feita de várias combinações, moléculas, de formas mais simples da matéria, chamadas de elementos químicos. Um elemento químico é uma substância formada por um único tipo de átomo (ATKINS; JONES, 2001).

Atualmente, muitas pessoas estão familiarizadas como o conceito de átomo e estão cientes de que átomos são partículas microscópicas de que toda a matéria é composta (RUSSELL, 1994). Sendo assim pode-se afirmar que os átomos existem, embora não possam ser vistos, e que eles são as unidades que formam os elementos químicos. Por sua vez, a química utiliza a unidade fundamental “átomo” para definir o elemento, sendo este uma substância que é composta por um único tipo de átomo (ATKINS; JONES, 2001). Segundo Russel (1994), em muitas substâncias, os átomos são agrupados ou reunidos em agregados de dois ou mais. Este arranjo é uma molécula, e segundo Atkins e Jones (2001), uma molécula é um grupo discreto de átomos ligados em um arranjo específico.

Os elementos químicos são agrupados em um arranjo chamado de Tabela Periódica (ATKINS; JONES, 2001). Dentro deste os elementos são listados na ordem crescente do seu número atômico e arranjados em linhas contendo certo número de elementos. Sendo assim estes elementos formam famílias cujas propriedades têm tendências reguladoras. A Tabela Periódica ainda apresenta uma classificação por cores de elementos segundo o seu comportamento, conforme apresentado na Figura 1.

Figura 1 – Tabela Periódica dos Elementos

1A		Elementos de Transição										3A a 8A									
1 <b>H</b> Hidrogênio	2 <b>He</b> Hélio											3 <b>B</b> Boro	4 <b>C</b> Carbono	5 <b>N</b> Nitrogênio	6 <b>O</b> Oxigênio	7 <b>F</b> Flúor	8 <b>Ne</b> Neônio				
3 <b>Li</b> Lítio	4 <b>Be</b> Berílio											13 <b>Al</b> Alumínio	14 <b>Si</b> Silício	15 <b>P</b> Fósforo	16 <b>S</b> Enxofre	17 <b>Cl</b> Cloro	18 <b>Ar</b> Argônio				
11 <b>Na</b> Sódio	12 <b>Mg</b> Magnésio	21 <b>Sc</b> Escândio	22 <b>Ti</b> Titânio	23 <b>V</b> Vanádio	24 <b>Cr</b> Cromo	25 <b>Mn</b> Manganês	26 <b>Fe</b> Ferro	27 <b>Co</b> Cobalto	28 <b>Ni</b> Níquel	29 <b>Cu</b> Cobre	30 <b>Zn</b> Zinco	31 <b>Ga</b> Gálio	32 <b>Ge</b> Germânio	33 <b>As</b> Arsênio	34 <b>Se</b> Selênio	35 <b>Br</b> Bromo	36 <b>Kr</b> Criptônio				
19 <b>K</b> Potássio	20 <b>Ca</b> Cálcio	39 <b>Y</b> Ítrio	40 <b>Zr</b> Zircônio	41 <b>Nb</b> Nióbio	42 <b>Mo</b> Molibdênio	43 <b>Tc</b> Tecnécio	44 <b>Ru</b> Rutênio	45 <b>Rh</b> Ródio	46 <b>Pd</b> Paládio	47 <b>Ag</b> Prata	48 <b>Cd</b> Cádmio	49 <b>In</b> Índio	50 <b>Sn</b> Estanho	51 <b>Sb</b> Antimônio	52 <b>Te</b> Telúrio	53 <b>I</b> Iodo	54 <b>Xe</b> Xenônio				
37 <b>Rb</b> Rubídio	38 <b>Sr</b> Estrôncio	57-71 Lantanídeos	72 <b>Hf</b> Háfio	73 <b>Ta</b> Tântalo	74 <b>W</b> Tungstênio	75 <b>Re</b> Rênio	76 <b>Os</b> Ósmio	77 <b>Ir</b> Írídio	78 <b>Pt</b> Platina	79 <b>Au</b> Ouro	80 <b>Hg</b> Mercúrio	81 <b>Tl</b> Tálio	82 <b>Pb</b> Chumbo	83 <b>Bi</b> Bismuto	84 <b>Po</b> Polônio	85 <b>At</b> Astato	86 <b>Rn</b> Radônio				
55 <b>Cs</b> Césio	56 <b>Ba</b> Bário	89-103 Atinídeos	104 <b>Rf</b> Rutherfordio	105 <b>Db</b> Dúbnio	106 <b>Sg</b> Seabórgio	107 <b>Bh</b> Bório	108 <b>Hs</b> Hássio	109 <b>Mt</b> Meitnério	110 <b>Uun</b> Ununílio	111 <b>Uuu</b> Ununúnio	112 <b>Uub</b> Ununbício	113 <b>Uut</b> Ununtrio	114 <b>Uuq</b> Ununquádro	115 <b>Uup</b> Ununpértio	116 <b>Uuh</b> Ununhexio	117 <b>Uus</b> Ununseptio	118 <b>Uuo</b> Ununoctio				
		* 6 57 <b>La</b> Lantânio	58 <b>Ce</b> Cério	59 <b>Pr</b> Praseodímio	60 <b>Nd</b> Neodímio	61 <b>Pm</b> Promécio	62 <b>Sm</b> Samário	63 <b>Eu</b> Európio	64 <b>Gd</b> Gadolínio	65 <b>Td</b> Térbio	66 <b>Dy</b> Disprósio	67 <b>Ho</b> Hólmio	68 <b>Er</b> Érbio	69 <b>Tm</b> Túlio	70 <b>Yb</b> Ítérbio	71 <b>Lu</b> Lutécio					
		** 7 89 <b>Ac</b> Actínio	90 <b>Th</b> Tório	91 <b>Pa</b> Protactínio	92 <b>U</b> Urânio	93 <b>Np</b> Netúnio	94 <b>Pu</b> Plutônio	95 <b>Am</b> Americio	96 <b>Cm</b> Cúrio	97 <b>Bk</b> Berquélio	98 <b>Cf</b> Califórnio	99 <b>Es</b> Einstênio	100 <b>Fm</b> Férmio	101 <b>Md</b> Mendelévio	102 <b>No</b> Nobelio	103 <b>Lw</b> Laurêncio					

Fonte: IUPAC (2015).

## 2.2.1 Fórmulas

Existem pelo menos duas formas de se expressar uma molécula: a fórmula química e a fórmula estrutural. A fórmula química de um composto representa sua composição em termos de símbolos químicos, e os valores subscritos mostram o número de átomos de cada elemento presente (ATKINS; JONES, 2001). Já a fórmula estrutural de uma molécula não apresenta somente o número de cada tipo de átomo, mas também mostra visualmente a forma como eles estão ligados entre si, no interior da molécula (RUSSELL, 1994).

Essas duas representações são importantes, e sempre que possível devem ser apresentadas ao aluno em conjunto, pois a fórmula química fornece uma visão não espacial da molécula, que permite inúmeras interpretações sobre o composto. Por exemplo, apresenta-se a fórmula de um brometo como  $C_3H_7Br$ . Ele pode apresentar duas estruturas: uma onde o Bromo se liga ao Carbono radicalar, e outra onde o Bromo se liga ao Carbono central. Conforme a complexidade da molécula aumenta, as possibilidades de estruturas crescem exponencialmente (SOLOMONS, 2000). Para um estudante é um conceito abstrato imaginar que uma mesma fórmula química corresponde a vários compostos diferentes, definidos apenas pela fórmula estrutural, e isso reforça a necessidade de uma visão prática do conteúdo.

Para que os átomos de uma fórmula estrutural se unam, faz-se o uso de uma ligação química. Forma-se uma ligação entre dois átomos se o arranjo resultante dos dois núcleos e seus elétrons tem menos energia que a energia total dos átomos separados (ATKINS; JONES, 2001). Ainda segundo os autores se a redução de energia for obtida por transferência completa de um ou mais elétrons de um átomo para o outro, formam-se íons e o composto mantém-se

pela atração eletrostática. Este tipo de arranjo é conhecido como ligação iônica. Porém se a redução de energia for atingida pelo compartilhamento de elétrons, os átomos unem-se por meio de uma ligação covalente para formar moléculas discretas. Tem-se um terceiro tipo de ligação, a metálica, na qual cátions em grande número são mantidos juntos por um mar de elétrons, cada um dos quais vem de um átomo do arranjo. O Quadro 1 demonstra quais as ligações que ocorrem entre os grupos de elementos da tabela periódica e as condições para que ocorram, assim como alguns exemplos de moléculas onde esta ligação está presente.

Quadro 1– Tipos de Ligações Químicas

Grupos	Tipo Ligação	Condição	Exemplo de Molécula
Metais Alcalinos ou Metais Alcalinos Terrosos e Hidrogênio	Iônica	Metais com eletronegatividade menor ou igual a 1,0	LiH, NaH, KH, CaH <sub>2</sub> , SrH <sub>2</sub> , BaH <sub>2</sub>
	Covalente	Metais com eletronegatividade maior que 1,0	BeH <sub>2</sub> , MgH <sub>2</sub>
Não Metais ou Semi-metais e Hidrogênio	Covalente	Nenhuma	HF, HCl, HBr, HI, H <sub>2</sub> O, H <sub>2</sub> S, H <sub>2</sub> Se, PH <sub>3</sub> , Si <sub>4</sub> , CH <sub>4</sub>
Não Metais ou Semi-metais e Não Metais ou Semi-metais	Covalente	Nenhuma	F <sub>2</sub> , Cl <sub>2</sub> , CO, CO <sub>2</sub> , P <sub>4</sub> , Cl <sub>2</sub> O
Qualquer Metal e Qualquer Metal	Metálica	Nenhuma	Fe <sub>(n)</sub> , Al <sub>(n)</sub>
Qualquer Metal e Não Metais ou Semi-metais	Iônica	Diferença de eletronegatividade maior ou igual a 1,7	NaCl, KCl, AlF <sub>3</sub> , K <sub>2</sub> O
	Covalente	Diferença de eletronegatividade menor que 1,7	AlCl <sub>3</sub> , HgCl <sub>2</sub>

Fonte: Atkins e Jones (2001).

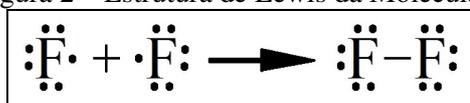
### 2.2.2 Estrutura de Lewis

Em uma ligação iônica, um átomo perde elétrons de sua camada de valência, e outro recebe elétrons, até que ambos atinjam a configuração de um gás nobre, um duplete para os elementos vizinhos do hélio e um octeto para os demais. A mesma constatação pode ser feita para as ligações covalentes, porém, nas ligações covalentes os átomos compartilham elétrons até atingir a configuração necessária (ATKINS; JONES, 2001). Este conceito é chamado de Regra do Octeto. Por exemplo, o nitrogênio possui cinco elétrons na valência e utiliza mais três para completar o octeto. O cloro possui sete elétrons na valência, e usa mais um para completar o octeto. Já o argônio, que é um gás nobre, possui um octeto completo e não compartilha seus elétrons. O hidrogênio utiliza mais um elétron para chegar ao duplete do hélio. De forma geral a valência do átomo é igual ao número de ligações que ele pode formar (ATKINS; JONES, 2001).

A estrutura de Lewis de uma molécula representa os átomos por seus símbolos químicos, as ligações por linhas e os pares isolados por pares de pontos. Sendo assim, a estrutura de Lewis ajuda no entendimento das propriedades das moléculas incluindo suas formas e suas possíveis reações. Deve-se deixar claro que a estrutura de Lewis não define a forma da molécula, somente seus padrões de ligação (ATKINS; JONES, 2001).

A Figura 2 mostra a estrutura de Lewis para uma molécula com dois átomos de flúor, que possui sete elétrons na camada de valência e por isso precisa de mais um elétron para completar o octeto, neste caso o elétron está sendo compartilhado entre os dois por meio de uma ligação covalente.

Figura 2 – Estrutura de Lewis da Molécula F<sub>2</sub>



Fonte: Atkins e Jones (2001).

Quando se possui apenas um par de elétrons emparelhados, conforme a Figura 2, tem-se uma ligação simples, porém dois átomos podem compartilhar dois ou três pares de elétrons. Quando dois pares de elétrons são compartilhados ocorre uma ligação dupla e três pares de elétrons formam uma ligação tripla. Uma ligação dupla é descrita com duas linhas entre os átomos e a ligação tripla é representada por três linhas entre os átomos. Assim como na ligação simples cada linha representa um par de elétrons (ATKINS; JONES, 2001).

Para que seja possível a escrita de uma estrutura de Lewis é preciso saber quais átomos estão ligados entre si na molécula, desta forma tem-se o átomo terminal que é aquele que se liga somente a outro átomo, e o átomo central se liga em pelo menos dois outros átomos. Como exemplos de átomos centrais têm-se o átomo de oxigênio (O) da molécula da água (H<sub>2</sub>O), e o átomo de carbono (C) do metano (CH<sub>4</sub>) (ATKINS, JONES, 2001).

A estrutural geral da molécula e a identidade do átomo central são conhecidas de antemão, mas se houver dúvida, uma boa regra é escolher como átomo central o elemento com a mais baixa energia de ionização. Este arranjo frequentemente conduz ao mínimo de energia, porque um átomo central compartilha mais elétrons do que um átomo terminal (ATKINS; JONES, 2001 p.171).

A forma de uma molécula determina várias de suas qualidades, como seu sabor, seu odor ou mesmo sua função como fármaco. Esta forma também governa as reações que ocorrem no organismo dos seres humanos a fim de mantê-los vivos. Sendo assim as propriedades dos materiais também são afetadas por sua geometria (ATKINS; JONES, 2001). A estrutura de Lewis vista anteriormente, é a representação em duas dimensões das ligações, ou seja, conectividade, entre os átomos, e, exceto em casos simples não representam o arranjo das moléculas no espaço. A teoria da Repulsão dos Pares de Elétrons da Camada de Valência

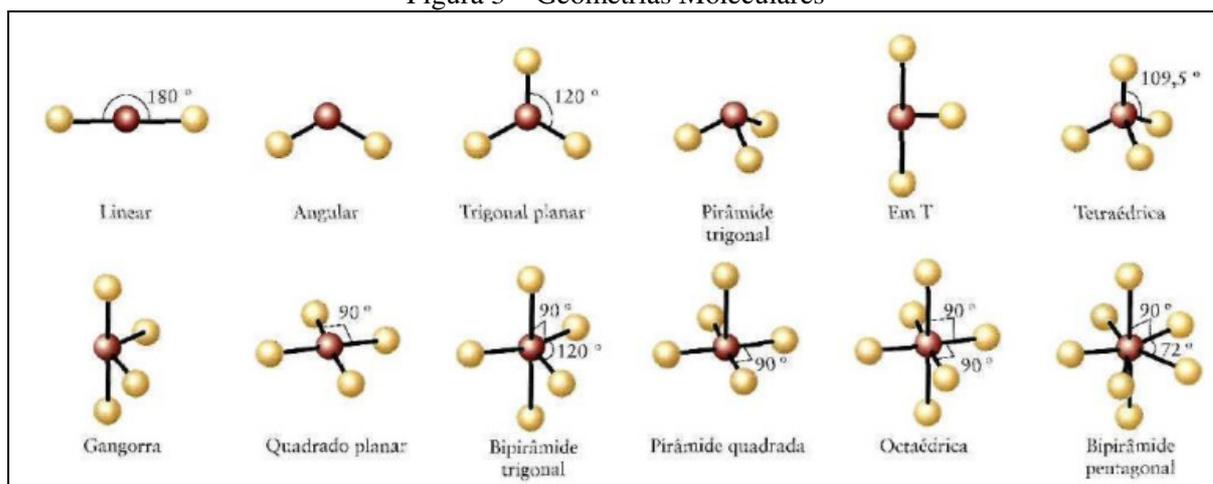
(VSEPR) amplia a teoria de Lewis para que seja possível explicar as formas das moléculas adicionando regras para explicarem os ângulos de cada ligação (ATKINS; JONES, 2001).

Segundo os autores, uma série de cinco regras deve ser seguida, conforme a ordem abaixo:

- a primeira regra determina que as regiões de altas concentrações de elétrons se repelem, ou seja, tentem de se afastar o máximo possível, mantendo a mesma distância do átomo central;
- a segunda regra identifica que não existem diferenças entre ligações simples ou múltiplas, uma ligação múltipla é tratada apenas como uma outra região de alta concentração de elétrons;
- a terceira regra define que quando existem dois ou mais átomos centrais, as ligações são tratadas de forma independente;
- a quarta regra impõe que todos os pares, ligantes ou isolados, devem ser analisados para a definição do arranjo, porém apenas os pares ligantes são considerados ao descrevermos a forma do arranjo;
- a quinta e última regra define que os pares isolados devem ser tratados como exercendo uma maior repulsão do que os pares de ligação.

A Figura 3 mostra as formas de moléculas simples e seus ângulos de ligação, os pares não ligantes não foram incluídos, pois não são considerados na representação das formas geométricas.

Figura 3 – Geometrias Moleculares



Fonte: Atkins e Jones (2001).

O Quadro 2 demonstra a relação que existe entre os tipos de geometria e a quantidade de pares existentes na molécula.

Quadro 2– Relação entre os pares e o tipo de geometria

Pares	Pares Ligantes	Pares não Ligantes	Geometria	Ângulo das Ligações	Exemplo
2	2	0	Linear	180°	CO <sub>2</sub>
3	2	1	Angular	120°	SO <sub>2</sub>
4	2	2	Angular	110°	H <sub>2</sub> O
3	3	0	Trigonal Plana	120°	BF <sub>3</sub>
4	4	0	Tetraédrica	109,5°	CH <sub>4</sub>
4	3	1	Piramidal	110°	NH <sub>3</sub>
5	5	0	Bipiramidal Trigonal	90°, 120°	PCl <sub>5</sub>
5	4	1	Gangorra	180°, 120°	SF <sub>4</sub>
5	3	2	Forma de T	90°, 180°	ClF <sub>3</sub>
5	2	3	Linear	180°	XeF <sub>2</sub>
6	6	0	Octaédrica		
6	5	1	Piramidal Quadrada	90°	BrF <sub>5</sub>
6	4	2	Quadrada Planar	90°	XeF <sub>4</sub>
7	7	0	Bipiramidal Pentagonal	90°, 72°	IF <sub>7</sub>

Fonte: Atkins e Jones (2001).

Após o domínio e utilização de todas essas informações, deve-se estar apto a tarefa de prever em três dimensões a visão espacial de uma molécula, a partir do seu esboço bidimensional.

### 2.3 WEBGL

O HTML é uma linguagem de marcação, utilizada para descrever páginas da internet, sendo composto por uma série de *tags* de marcação, onde cada uma descreve um conteúdo diferente na página. As *tags* são prefixadas e sufixadas pelos sinais de `menor` que e `maior` que respectivamente (W3CSCHOOLS, 2015). A atual versão do HTML é a cinco, que nasceu em 2004 de uma cooperação entre duas grandes empresas a W3C e a WHATWG (W3SCHOOLS, 20015).

Esta versão inclui diversas novas funcionalidades, como semântica e acessibilidade. Foram incluídos novos recursos que antes só eram possíveis com a utilização de tecnologias de terceiros por meio de *plug-ins*, como o suporte à reprodução de áudio e vídeo. Dentre os novos recursos disponibilizados, destaca-se a inclusão do componente `<canvas>` para desenhos, componentes para reprodução de áudio e vídeo, armazenamento local de arquivos e o WebGL, para gráficos 3D.

Já o WebGL, é uma API para renderização 3D de modo imediato desenvolvida para web, derivada da biblioteca gráfica *OpenGL for Embedded Systems* (OpenGL ES) 2.0. O OpenGL ES é uma API livre e multi-plataforma para funções gráficas 2D e 3D em sistemas

embarcados, baseada no sistema gráfico do OpenGL 2.0, mas concebido principalmente para *hardwares* gráficos rodando em dispositivos móveis e embarcados (KHRONOS, 2015). Por ser baseado no OpenGL ES 2.0 a WebGL possui uma funcionalidade de renderização similar, porém com foco no HTML, principalmente no componente `<canvas>` do HTML5 (ANYURU, 2012).

Por ser uma API de modo imediato, a WebGL não salva nenhum modelo interno da cena que deveria ser desenhada, em vez disso a WebGL mantém uma representação própria da cena em memória. Por isto toda a cena precisa ser redesenhada a cada *frame*, independente dela ter sido alterada ou não (ANYURU, 2012).

Por ser planejada como uma API para ser utilizada na web, a WebGL necessariamente está de acordo como os princípios de segurança da plataforma web, e foi concebida com esta ideia desde o princípio (KHRONOS, 2015). Atualmente os maiores vendedores de navegadores do mercado, como Mozilla, Apple e Google, fazem parte de um grupo de trabalho conhecido por Khronos, que visa à criação de padrões abertos e livres de *royalties* (KHRONOS, 2015).

O elemento `<canvas>` é uma área retangular dentro da página web, onde podem ser desenhados objetos gráficos, utilizando Javascript. Originalmente a integração com o `<canvas>` ocorre por meio da interface de renderização `CanvasRenderingContext2D`. Já a chamada para a renderização por WebGL se dá por meio da interface `WebGLRenderingContext` (ANYURU, 2012).

Ambas as interfaces de integração ocorrem por meio da chamada do método `getContext()` do elemento `<canvas>`. O WebGL gerencia vários tipos de recursos que são representados como objetos do tipo *Document Object Model* (DOM). Cada objeto é manipulado a partir da interface `WebGLObject` (KHRONOS, 2015).

Segundo Cantor e Jones (2012), o WebGL, utiliza de uma técnica de renderização local, ou seja, os elementos que são parte do objeto 3D, normalmente serão baixados do servidor, porém todo o processo requerido para obter o objeto em 3D serão feitas localmente. Cantor e Jones (2012) ainda colocam que em comparação com outras tecnologias como Java 3D, Flash e o UnityWeb Player, a WebGL apresenta algumas vantagens:

- a) programação em Javascript: o Javascript permite ao desenvolvedor acesso direto a todas as partes do DOM. E por ser padrão para a web, permite uma fácil integração com outras bibliotecas Javascript;
- b) gerenciamento automático de memória: diferentemente do OpenGL e outras tecnologias, o WebGL não possui nenhum comando para alocação e deslocação de

memória;

- c) desempenho: o desempenho do WebGL pode ser comparada a de aplicações desktop, isso ocorre pois o WebGL possui acesso aos *hardwares* gráficos instalados na máquina da onde ele está sendo executado;
- d) sem compilação: como o WebGL é escrito em Javascript não é preciso compilar o código antes de executar, isto permite com que as edições sejam feitas e visualizadas rapidamente.

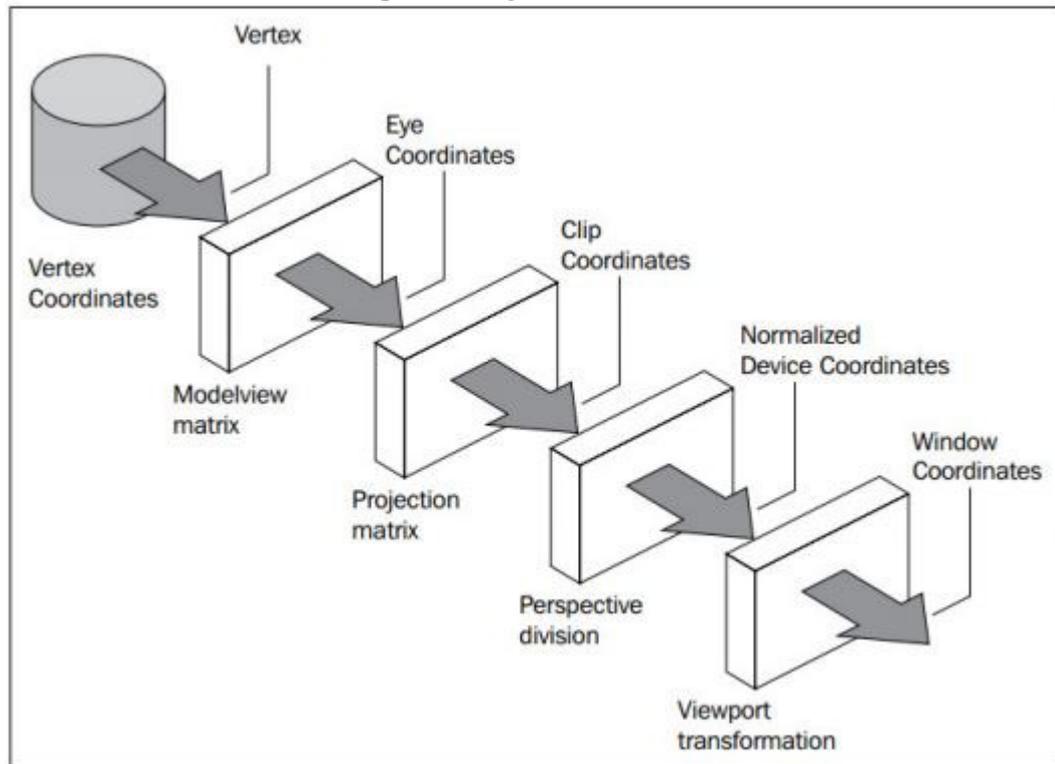
O processo de renderização local que o WebGL utiliza faz com que os polígonos sejam decompostos em triângulos, linhas e primitivas. Assim cada primitiva é processada de forma paralela pela GPU, passando por uma série de passos, conhecida como *pipeline* de renderização, até que finalmente a cena final é criada e exposta no elemento `<canvas>` (CANTOR; JONES, 2012).

### 2.3.1 Pipeline do WebGL

Em um ambiente 3D o objeto modelo é chamado de *mesh*, cada face que ele possui é chamada de polígono, onde cada polígono possui três ou mais cantos conhecidos como vértices. As GPUs mais antigas não permitiam a manipulação dos vértices, mas elas possuíam funções embutidas para, por exemplo, aplicar rotação e escala. Já as GPUs modernas permitem a utilização de um *pipeline* programável, com as *pipelines* programáveis o desenvolvedor possui total flexibilidade para manipular os vértices, criando funções próprias para descrever como os objetos serão renderizados (ARORA, 2014).

A Figura 4 mostra a *pipeline* do WebGL, onde tem-se como entrada as *Vertex Coordinates*, também chamadas de *Vertex Buffer Objects* (VBO), os dois passos seguintes, *Modelview Matrix* e *Project Transformation*, são realizados pelo *Vertex Shader*. Cada vértice de um objeto tem que ser transformado em relação à sua localização bem como a localização da câmera. Em seguida os vértices que não estão na área de visualização serão cortados (*Perspective Division*). E por fim a *Viewport Transformation* define a localização e o tamanho do objeto processado.

Figura 4 – Pipeline do WebGL



Fonte: Arora (2014).

### 2.3.2 Three.JS

Com o WebGL pode-se usar diretamente os recursos da placa de vídeo para criar objetos gráficos de alta performance, sendo eles 2D ou 3D. Porém programar WebGL diretamente do Javascript é um processo complexo e muito suscetível a erros. Neste contexto o Three.js é uma biblioteca que torna este processo mais simples (DIRKSEN, 2013). Segundo (FHTR, 2012) o Three.js é uma biblioteca, leve, de fácil utilização que permite a renderização de WebGL, mas não somente este, como também permite o uso de *canvas* e *Scalabe Vector Graphics* (SVG), permitindo ainda o uso de alguns *Shaders*.

O Three.js possui 4 componentes básicos sendo eles, o renderizador, a câmera, a cena e por último os objetos. Cada um deles possui uma classe e são acessíveis através da instanciação destas classes (FHTR, 2012). O Three.js funciona na maioria dos navegadores modernos, exceto algumas versões do Internet Explorer, conforme demonstra o Quadro 3 abaixo (DIRKSEN, 2013).

Quadro 3 – Three.js compatibilidade entre navegadores

Navegadores	Suportado desde
Mozilla Firefox	Versão 4.0
Google Chrome	Versão 9.0
Safari	Versão 5.1
Opera	Versão 12.0
Internet Explorer	Parcialmente desde versão 11

### 2.3.3 Fabric.JS

Com o elemento `<canvas>` do HTML5 pode-se criar uma grande gama de gráficos para web, porém a API que é fornecida é de baixo nível, sendo assim após o desenho de uma forma é extremamente difícil aplicar qualquer tipo de interação sobre a mesma, pois se deve manipular o contexto do `<canvas>` e não seus objetos (ZAYTSEV, 2012).

Ainda segundo Zaytsev (2012) a biblioteca Fabric.js tem como objetivo facilitar o processo de interação sobre o que for desenhado, fornecendo uma API que permite controle sobre os objetos e não sobre o contexto, fornecendo métodos para manipulação destes objetos, enquanto a API se encarrega de cuidar do contexto do canvas e de seu processo de renderização.

## 2.4 TRABALHOS CORRELATOS

A seguir serão apresentados dois softwares de desenho químico. O Avogadro (HANWELL et al, 2012) voltado para edição e visualização avançada de moléculas em 3D. E o BKChem, uma ferramenta *opensource* para desenho químico 2D (DANNE, 2010).

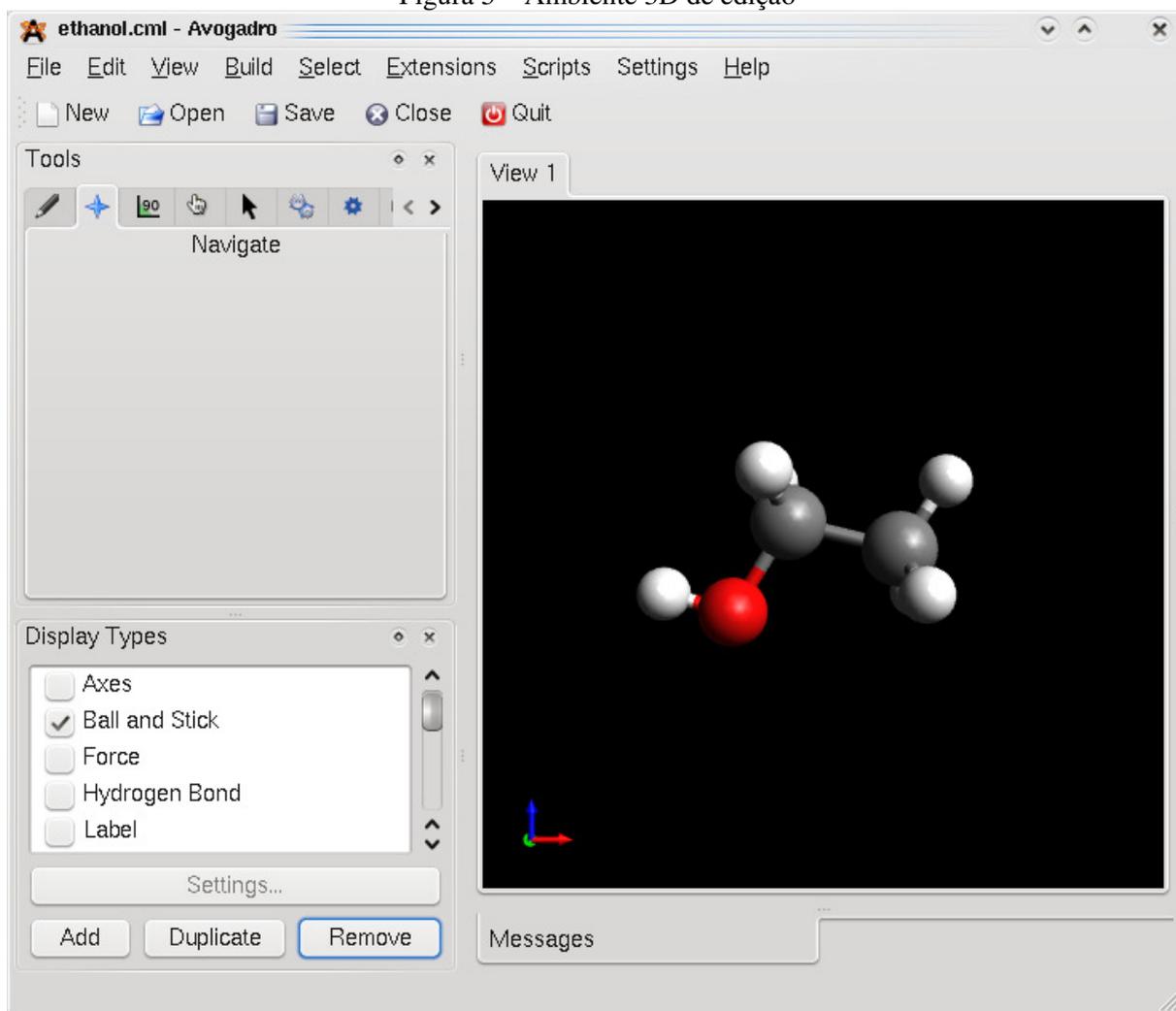
### 2.4.1 Avogadro

O projeto Avogadro constitui-se de um editor e visualizador de moléculas, desenvolvido para ser multi-plataforma e utilizado na química computacional, modelagem química e bioinformática. Ele oferece uma renderização de qualidade e uma arquitetura modular permitindo a utilização de vários *plug-ins*. Seus usos típicos incluem a modelagem de estruturas moleculares, formatação de arquivos de entrada e análise da saída de vários programas utilizados na química computacional (HANWELL et al, 2012). O Avogadro não pretende apenas ser utilizado em pesquisa, como também pode ser utilizado para ensino. O Avogadro é desenvolvido com a linguagem de programação C++ e permite a criação de *plug-ins*, utilizando também a linguem C++ ou ainda Python, sendo assim uma ferramenta multi-plataforma.

A Figura 5 demonstra em um ambiente 3D a fórmula estrutural de uma molécula de etanol, assim como algumas das funções disponíveis no Avogadro, como a escolha do tipo de

apresentação da molécula. Para os usuários, ele oferece uma interface fácil de usar, suporte integrado para *download* a partir de bases de dados comuns, como PubChem e Protein Data Bank. A extração de dados químicos a partir de uma ampla variedade de formatos, incluindo a saída de química computacional, e suporte nativo, para o formato de arquivo *Chemical Markup Language* (CML). Para os desenvolvedores, ele pode ser facilmente estendido através de um poderoso mecanismo de *plug-in* para suportar novos recursos em química orgânica, complexos inorgânicos, a concepção de medicamentos, materiais, biomoléculas e simulações.

Figura 5 – Ambiente 3D de edição



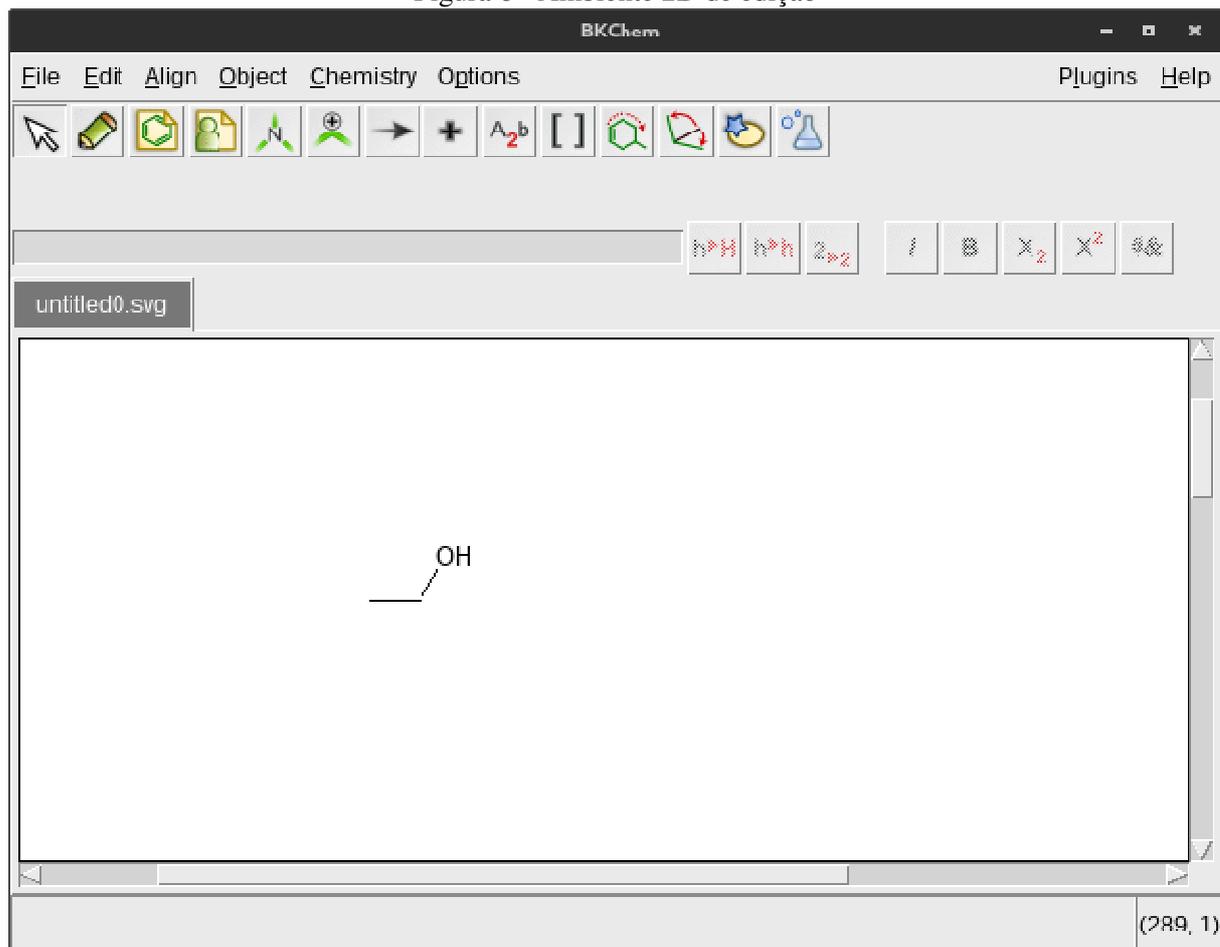
Fonte: Hanwell, et al. (2012).

#### 2.4.2 BKChem

BKChem é um software livre de desenho químico em 2D. Foi concebido e escrito por Beda Kosata e é atualmente mantido por Reinis Danne. O BKChem é escrito em Python, isto implica algumas das características do programa, como a independência de plataforma e a performance. Como Python é interpretado não possui o mesmo desempenho que um aplicativo compilado em código nativo (DANNE, 2010). O BKChem permite ainda a

importação de arquivos no formato *International Chemical Identifier* (InChI), que foi desenvolvido pela *International Union of Pure and Applied Chemistry* (IUPAC) para estabelecer uma nomenclatura única para substâncias químicas, a fim de facilitar a sua acessibilidade, fornecendo ainda um padrão de entrada de dados para softwares da área química (IUPAC, 2015). A Figura 6 demonstra a molécula de etanol em sua fórmula estrutural completa em um ambiente 2D.

Figura 6 - Ambiente 2D de edição



Fonte: Danne (2010).

### 3 DESENVOLVIMENTO

Este capítulo apresenta todas as etapas de desenvolvimento do aplicativo. Nele são apresentados os principais requisitos, a especificação, a implementação e, por fim, os resultados e discussões.

#### 3.1 REQUISITOS

O Quadro 4 apresenta os Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) do VisEdu-Química.

Quadro 4 – Relação de Requisitos Funcionais e Não Funcionais

Nº	Descrição	Tipo
RF1	Permitir a digitação de uma fórmula química.	RF
RF2	Permitir a criação da fórmula estrutural com base nos componentes da fórmula química.	RF
RF3	Validar a fórmula estrutural.	RF
RF4	Exibir o resultado da fórmula estrutural em um ambiente 3D.	RF
RF5	Permitir interação da câmera no ambiente 3D.	RF
RNF1	Utilizar JavaScript e HTML5 como linguagens para implementação.	RNF
RNF2	Utilizar a API WebGL por meio da biblioteca Three.js.	RNF
RNF3	Utilizar o IntelliJ Idea versão 14 como Integrated Development Environment (IDE) para desenvolvimento.	RNF

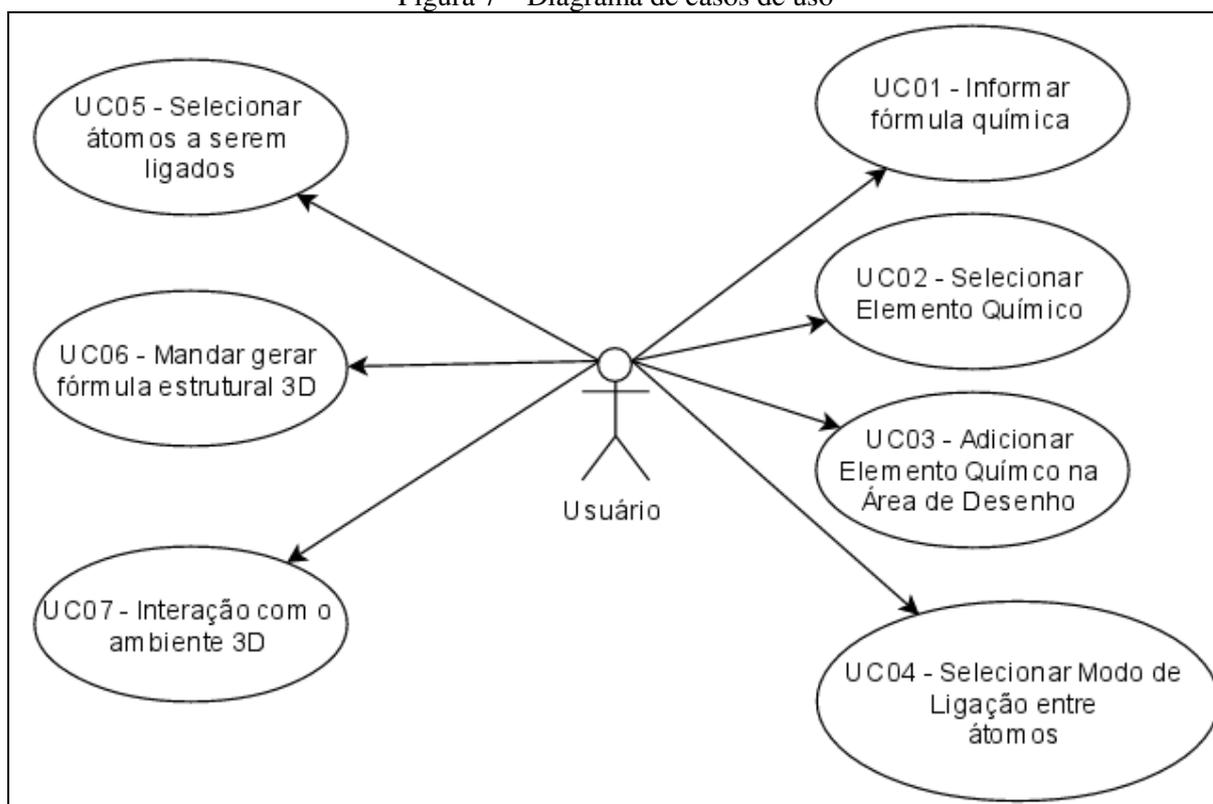
#### 3.2 ESPECIFICAÇÃO

Para especificar a implementação do projeto foram utilizados diagramas da *Unified Modeling Language* (UML) em conjunto com a ferramenta Draw.io. Para a modelagem do visualizador de material educacional foram utilizados os diagramas de casos de uso, de classes e de sequência, conforme apresentado nas seções seguintes.

##### 3.2.1 Diagrama de casos de uso

Nessa seção são apresentados os casos de uso que descrevem as funcionalidades do VisEdu-Química. De acordo com os requisitos funcionais foram definidos sete casos de uso, que permitem o ator executar operações na aplicação. Por se tratar de uma aplicação voltada para a aprendizagem da disciplina de química, obteve-se somente um ator, o usuário da aplicação sendo ele aluno ou professor. O mesmo foi definido como *Usuário*, conforme ilustrado na Figura 7. As informações de cada caso de uso estão disponíveis

Figura 7 – Diagrama de casos de uso



Para compreensão dos casos de uso é necessário ter conhecimento previo das partes que compõem a aplicação. O VisEdu-Química é formado por quatro itens principais, sendo eles:

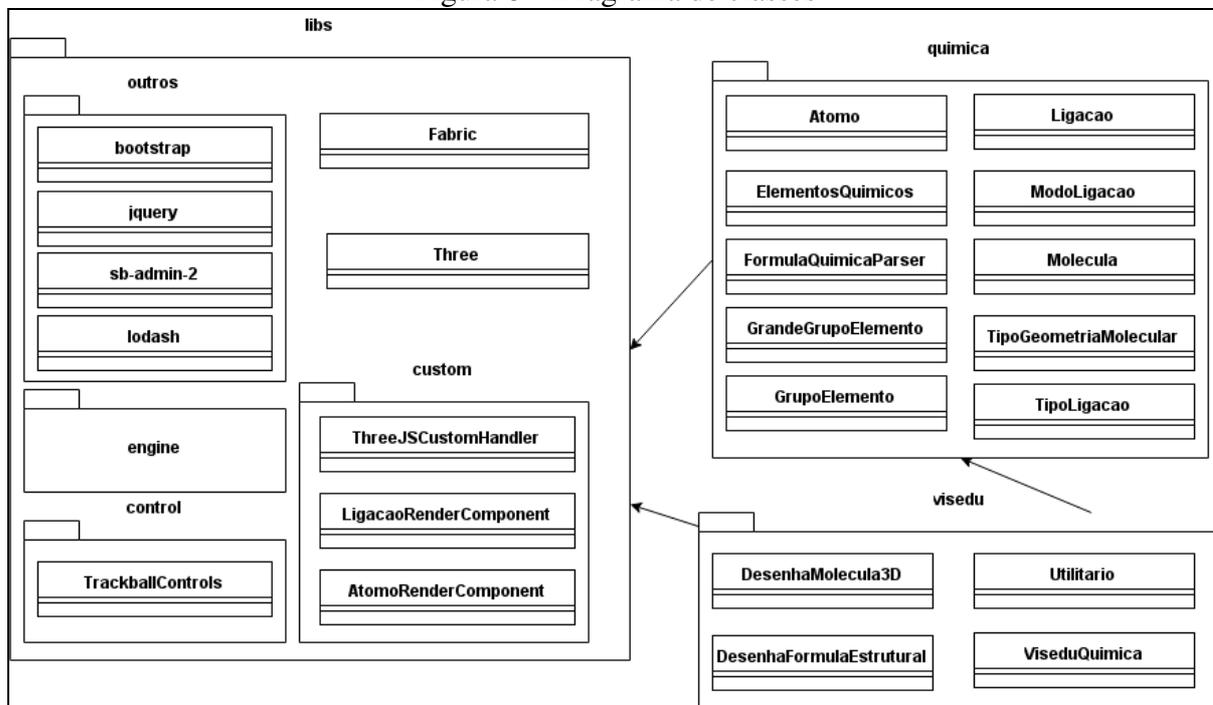
- a) Área de Entrada: local onde a fórmula química é informada;
- b) Lista de Elementos: local onde os elementos químicos presentes na fórmula informada encontram-se disponíveis para seleção, assim como os modos de ligação (simples ou dupla);
- c) Área de Desenho: local onde o Usuário, pode fazer a modelagem bidimensional da fórmula química informada;
- d) Área de Visualização: local onde o Usuário poderá interagir tridimensionalmente com a fórmula estrutural.

### 3.2.2 Diagrama de pacotes e classes

Nesta seção são descritas as classes que compõem a aplicação VisEdu-Química. A Figura 8 apresenta o diagrama contendo o conjunto de pacotes que compõem a aplicação e o relacionamento entre eles. Na mesma pode-se ainda observar os *frameworks* e classes da aplicação dentro de seus respectivos pacotes. Esse diagrama não apresenta as classes

pertencentes aos *frameworks*, salvo os que são compostos de apenas uma classe. Abaixo cada um dos pacotes é apresentado e suas classes devidamente definidas.

Figura 8 – Diagrama de classes



### 3.2.2.1 Pacote `js.libs`

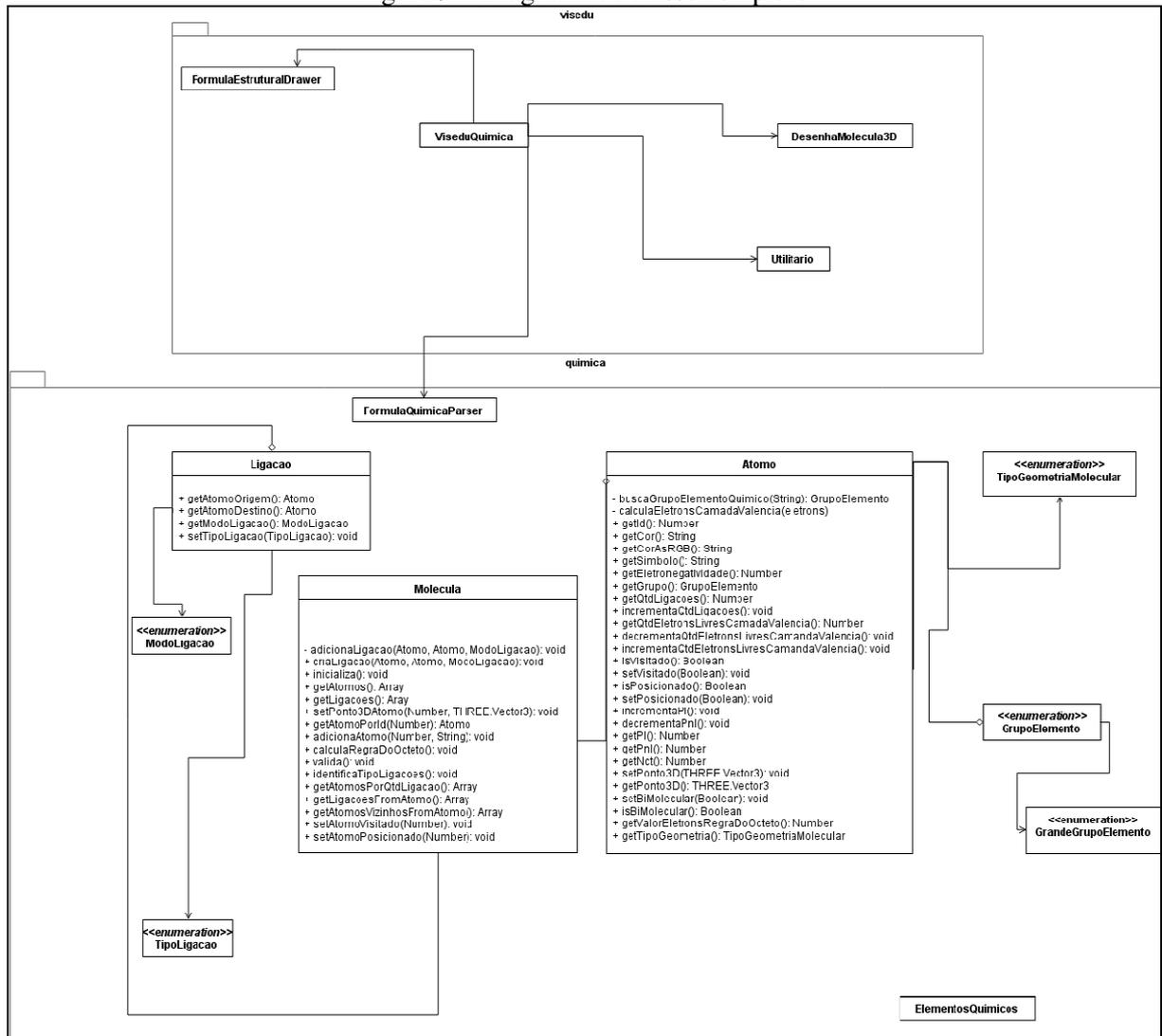
O pacote `js.libs` apresenta os *frameworks* utilizados para o desenvolvimento da aplicação. Através da Figura 8 observa-se que este pacote possui vários subpacotes, dentre eles o `js.libs.engine`, que contém o código fonte desenvolvido por Montibeler (2014). Devido à autoria do código, o mesmo não é apresentado de forma detalhada, no entanto suas referências podem ser visualizadas no Anexo A. Pode-se verificar ainda na Figura 8 a existência das classes `Three` e `Fabric`, que são responsáveis pela interação com o elemento de `<canvas>` nos ambientes tridimensional e bidimensional respectivamente.

Tem-se por fim a visualização dos subpacotes `js.libs.outros`, `js.libs.control` e `js.libs.custom`, sendo o primeiro deles responsável por armazenar bibliotecas utilitárias comuns em vários sistemas desenvolvidos em Javascript. O segundo pacote com a classe `TrackBallControl` é responsável pela interação do usuário com o ambiente tridimensional gerado pelo `Three`. Por fim, tem-se o pacote `js.libs.custom` com as classes utilizadas para adaptar as funcionalidades do motor de jogos para utilização na aplicação VisEdu-Química.

### 3.2.2.2 Pacote `js.quimica`

O pacote `js.quimica` apresenta as classes responsáveis pela manipulação e validação das regras de negócio vinculadas a área da química. Abaixo segue uma explicação de cada uma das classes, partindo das mais simples para finalizar nas mais complexas. A Figura 9 mostra as classes e seus relacionamentos dentro e fora do pacote.

Figura 9 – Diagrama de classe completo



A classe `ElementosQuimicos` é responsável por armazenar as informações de: nome, sigla, nox, elétrons, número atômico, eletronegatividade, configuração eletrônica, cor atribuída e grupo de todos os elementos químicos existentes na tabela periódica. Sendo que essas informações são necessárias para que as devidas verificações, validações e aferições possam ser feitas no decorrer do uso da aplicação.

A enumeração `TipoLigacao` é responsável por armazenar os tipos de ligações químicas existentes, classificadas em Iônica, Covalente e Metálica, conforme descrito na

tabela periódica. A enumeração `ModoLigacao`, armazena informações referentes aos dois modos de ligação utilizados pela aplicação: simples e dupla. As enumerações `GrupoElemento` e `GrandeGrupoElemento` armazenam os dados referentes aos grupos e grandes grupos da tabela periódica, conforme exposto no Quadro 5

Quadro 5– Classificação dos grupos e grandes grupos da tabela periódica

Grupos	Grandes Grupos
Hidrogênio	Hidrogênio
Metais Alcalinos	Metais
Metais Alcalinos Terrosos	
Metais de Transição	
Metais de Transição Interna	
Outros Metais	
Não Metais	Ametais
Gases Nobres	Gases Nobres

A classe `FormulaQuimicaParser` é responsável por validar e verificar a existência de todos os elementos químicos informados em uma fórmula química, disponibilizando os mesmos para uso subsequente. Para que este processo ocorra é necessária a chamada do método público `parse(String)`. Este método recebe como parâmetro a fórmula química informada pelo usuário, e com base nesta fórmula a aplicação irá verificar se todos os elementos químicos descritos nela são válidos por meio do método privado `isElementoQuimicoValido(String)`. Por sua vez, este método percorre a lista denominada `SIMBOLOS` da enumeração `ElementosQuimicos`, e verifica se a `String` recebida como parâmetro consta nesta lista. No fim da execução do método `parse(String)` obtém-se como retorno uma lista com todos os elementos químicos presentes na fórmula. Por exemplo, no caso da fórmula  $\text{CO}_2$ , dois elementos de Oxigênio estarão presentes neste retorno.

As classes `Atomo`, `Ligacao` e `Molecula`, são responsáveis por armazenar e validar os dados da fórmula estrutural que o usuário informar na Área de Desenho. Esta validação ocorre em várias etapas, sendo a primeira delas a identificação dos tipos de ligações. Esta etapa é executada através do método `identificaTiposLigacoes()`, que é disponibilizado de forma pública pela classe `Molecula`. Neste método todas as ligações são percorridas, e com base nos dois átomos que compõem a ligação, o `TipoLigacao` entre os átomos é identificada. No Quadro 1 demonstrado na seção 2.2.1, observa-se quais os tipos de ligações possíveis de acordo com as enumerações `GrupoElemento` e `GrandeGrupoElemento`. Caso não ocorra a identificação de algum tipo de ligação constante no Quadro 1 uma mensagem de erro é apresentada ao usuário.

A segunda etapa do processo de validação é o cálculo da regra do octeto. Esta validação é feita através da chamada do método público `calculaRegraDoOcteto()` da

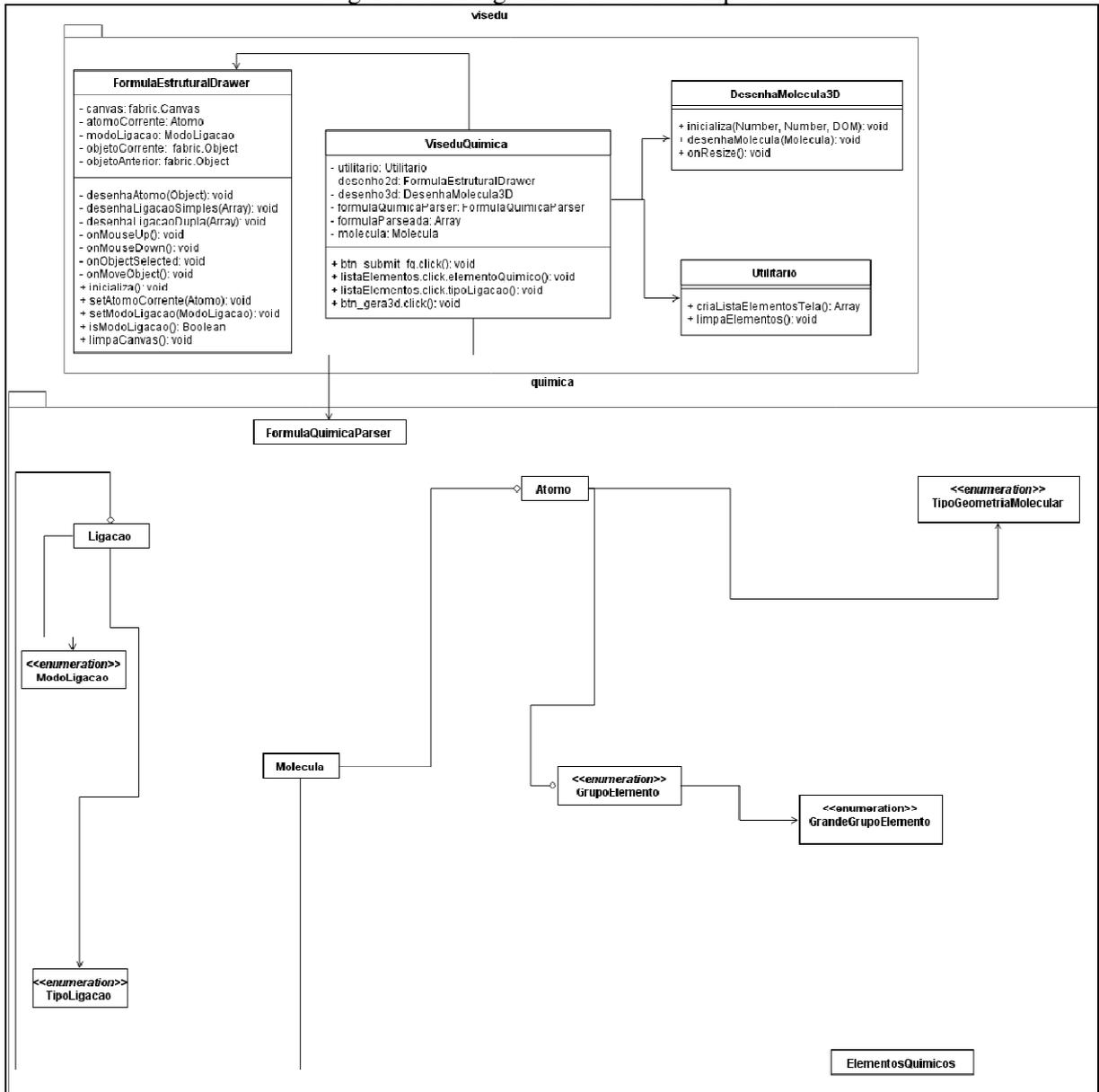
classe `Molecula`. Nesta etapa as ligações são novamente percorridas, e em caso do `TipoLigacao` ter sido identificada como `COVALENTE`, ambos os átomos da ligação têm seus elétrons na camada de valência incrementados em um, em caso de `ModoLigacao.DUPLA` em dois. Isto ocorre, pois, como descrito anteriormente na seção 2.2.1, neste tipo de ligação os elétrons são compartilhados entre os átomos. Em caso do `TipoLigacao` ter sido identificada como `IONICA`, uma validação adicional é executada. Essa validação é necessária, pois este tipo de ligação aceita apenas o `ModoLigacao` como `SIMPLES`. Dando continuidade ao processo principal, faz-se o cálculo da diferença de eletronegatividade, valor que determina nestas ligações qual dos átomos decreta um elétron, e qual dos átomos incrementa este elétron.

A terceira e última etapa do processo de validação ocorre no método `valida(Number)` da classe `Molecula`. Este método recebe como parâmetro a quantidade de elementos químicos apresentados na fórmula, e utiliza esta informação para verificar se todos estes elementos estão aplicados no desenho bidimensional. Em caso negativo uma mensagem de erro é apresentada ao usuário para que o mesmo finalize a representação utilizando os elementos faltantes. Ainda neste método faz-se uma validação para confirmar a existência de ao menos uma ligação em cada um dos átomos, formando uma espécie de grafo. E por fim é verificado se o valor calculado para regra do octeto está de acordo com o descrito no capítulo 2.2.

### 3.2.2.3 Pacote `js.visedu`

Este pacote é responsável pelas classes que controlarão o fluxo da aplicação, assim como as principais funções de desenho que serão utilizadas. A Figura 10 mostra tanto as classes deste pacote, como os seus relacionamentos.

Figura 10 – Diagrama de classes completo



A classe `ViseduQuimica`, é responsável por criar o fluxo de utilização da aplicação, com os métodos `btn_submit_fq.click()`, `listaElementos.click.elementoQuimico()`, `listaElementos.click.tipoLigacao()` e `btn_gera3d.click()`.

O método `btn_submit_fq.click()` faz a chamada do métodos `parse(String)` da classe `FormulaQuimicaParser` e `criaListaElementosTela(Array)` da classe `Utilitario`. Este último é responsável pela geração da Lista de Elementos, que é disponibilizada para o usuário com todos os elementos químicos. O método recebe como parâmetro a lista que é retornada através do método `parse(String)`.

Já os métodos `listaElementos.click.elementoQuimico()` e `listaElementos.click.tipoLigacao()`, fazem a seleção do elemento químico e do tipo de

ligação que irá ocorrer entre estes elementos. Essas informações são passadas para a classe `FormulaEstruturalDrawer` que faz a interação com a Área de Desenho através da utilização do *framework* `Fabric.js`. O desenho bidimensional é viabilizado ao usuário principalmente através das funções do mouse atreladas diretamente aos métodos de `onMouseUp()` e `onMouseDown()`, responsáveis pela criação das representações.

Por fim o método `btn_gera3D.click()` permite a geração da visualização da molécula no ambiente tridimensional. Para isto, faz-se uso da classe `DesenhaMolecula3D`. A referida classe possui apenas dois métodos, descritos na sequência.

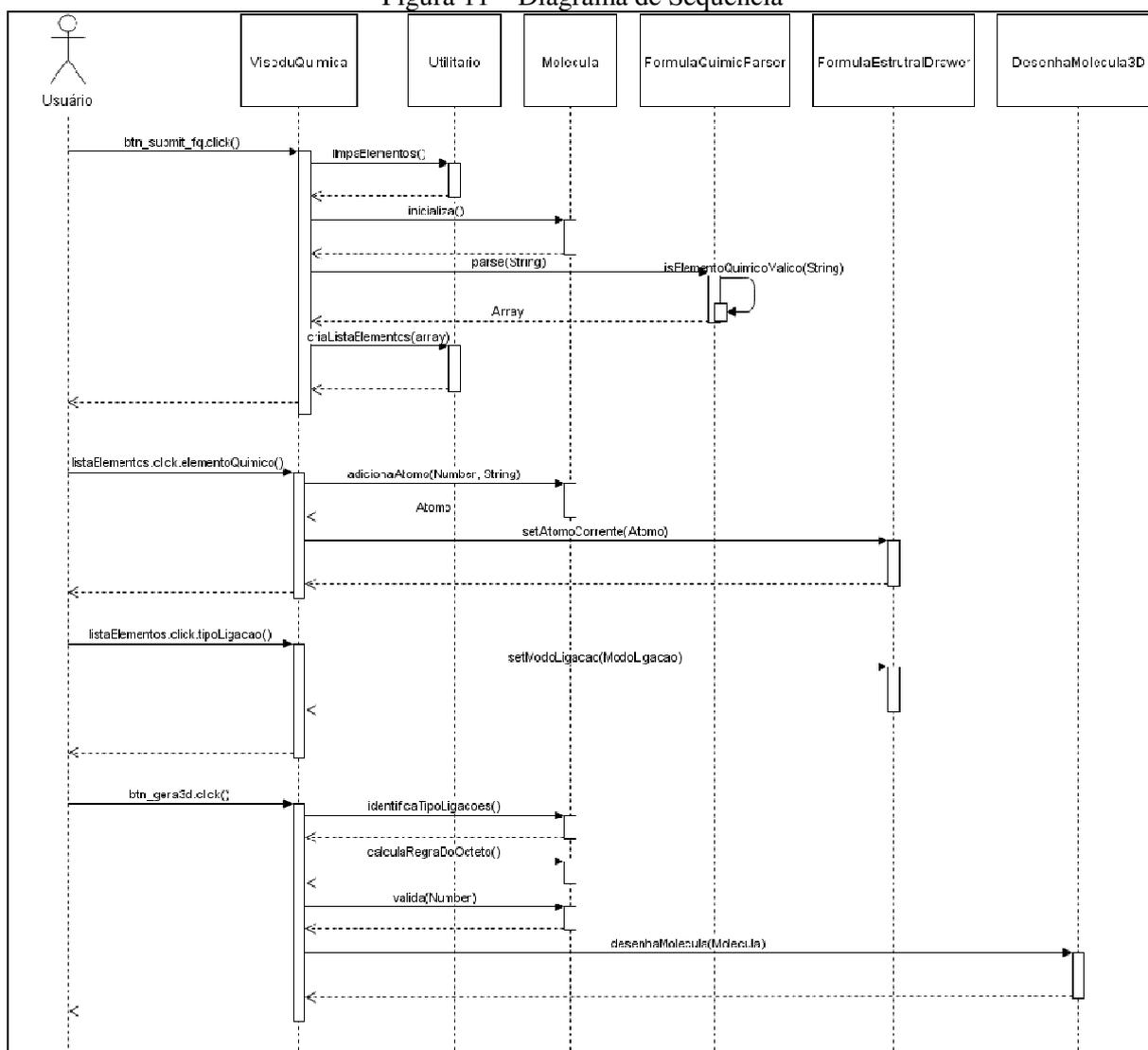
O primeiro método, nomeado `inicializa()`, é responsável por erguer todo o ambiente necessário para o funcionamento do motor de jogos, pela criação da cena principal, pela criação do objeto de iluminação, assim como pelo carregamento da classe customizada `ThreeJSHandler` que lida com os eventos de interação na área de visualização. Na sequência ocorre a chamada o método `Game.init()` que dá início ao fluxo de renderização do referido motor de jogos, e por consequência do `Three.js`.

O segundo método desenha molécula com base no `TipoGeometria` que é associada a mesma. Os tipos de geometria e seus ângulos podem ser visto no Quadro 2 (Seção 0). Caso nenhum tipo de geometria possa ser associado uma mensagem de erro é apresentada para o usuário.

### 3.2.3 Diagrama de Sequência

Esta seção apresenta um diagrama de sequência da aplicação `Visedu Química` que ilustra o processo de validação, desenho e geração de uma molécula tridimensional. Tal diagrama pode ser visto em detalhes na Figura 11.

Figura 11 – Diagrama de Sequência



O fluxo da aplicação tem início quando o usuário digita a fórmula química e realiza o envio da mesma por meio do método `btn_submit_fq.click()` da classe `Visedu`. Esta classe por sua vez faz a chamada dos métodos `inicializa()` da classe `Molecula`, arranjando a criação de uma nova molécula sem nenhuma ligação ou átomo. Em seguida o método `parse(String)` da classe `FormulaQuimicaParser` é invocado para que os elementos químicos sejam validados e retornados para o `Visedu`, com o objetivo de encaminhar para a classe `Utilitario` a fim de que seja feita a criação da Lista de Elementos.

O método `listaElementos.click.elementoQuimico()` permite que, ao usuário clicar/selecionar um elemento químico, ocorra a adição do mesmo na classe `Molecula` por intermédio do método `adicionaAtomo(Number, String)`. Esta ação resultará na criação de um objeto do tipo átomo, que será enviado para a classe `FormulaEstruturalDrawer` que por sua vez possibilitará a adição deste elemento na Área de Desenho. Este processo, assim

como a chamada do método `listaElementos.click.tipoLigacao()`, ocorrem várias vezes, até que se esgotem os elementos químicos disponíveis.

Por fim o método `btn_gera3d.click()` é chamado para que a visualização tridimensional ocorra. Este método executa os métodos `identificaTipoLigacao()`, `calculaRegraDoOcteto()`, `valida()` e `desenhaMolecula(Molecula)`, para que a molécula possa ser validada conforme descrito anteriormente, gerando então a visualização tridimensional.

### 3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas ferramentas e técnicas utilizadas na implementação da aplicação e a operacionalidade desta implementação.

#### 3.3.1 Técnicas e ferramentas utilizadas

O desenvolvimento desta aplicação foi realizado utilizando a linguagem de programação Javascript para a implementação das classes e a linguagem de marcação HTML5 para a implementação da interface web. Também foi utilizado o *framework* Three.js em sua versão 71, assim como o *framework* Fabric.js em sua versão 1.6.0-rc.1. O ambiente de programação utilizado foi o IntelliJ IDEA Ultimate, em sua versão 14.

O método de desenvolvimento seguido na implementação deste trabalho assemelha-se com o processo incremental da engenharia de software, onde foi criada uma lista de tarefas a serem feitas, uma lista de tarefas feitas e uma lista de tarefas sendo feitas. Conforme uma tarefa fosse concluída, a mesma era retirada de uma lista e adicionada na outra, e assim sucessivamente até que todas as tarefas fossem concluídas. Desta forma, inicialmente foi realizado o *download* do motor de jogos desenvolvido por Montibeler (2014). Em seguida foi criado um novo repositório hospedado pelo serviço BitBucket, para manter o repositório atualizado bem como fazer o registro das alterações, utilizou-se a ferramenta GIT em sua versão 2.5.3.

Os navegadores utilizados para teste foram Google Chrome versão 45.0.2454.85 (64-bit) e Mozilla Firefox versão 41. O computador utilizado em todo o desenvolvimento da aplicação foi um notebook Acer, modelo V3-571, com sistema operacional Arch Linux, com processador Intel Core i7, memória RAM de 6GB e placa de vídeo Intel Graphics 4000.

### 3.3.2 O visualizador de material educacional (VisEdu-Química)

Esta seção descreve a implementação do visualizador de material educacional, módulo de Química, na sua versão 1.0 apresentando as principais rotinas desenvolvidas.

#### 3.3.2.1 Fórmula Química Parser

Para que seja possível a criação de uma molécula no ambiente bidimensional, primeiramente se faz necessário a inserção de uma fórmula química. Uma vez inserida, esta fórmula será validada para que possíveis erros de estrutura ou elementos não existentes possam ser identificados. Para que esta validação ocorra a classe `ViseduQuimica` possui um método que é diretamente ligado ao `submit` da caixa de entrada da fórmula química. Este método faz a chamada dos métodos `parse(String)` da classe `FormulaQuimicaParser` e `criaListaElementos(Array)` da classe `Utilitario`, sendo este último o método responsável pela geração da lista de elementos químicos com os quais o usuário poderá interagir. O Quadro 6, demonstra a chamada do método.

Quadro 6 – Método click da caixa de entrada de `submit` da fórmula

```

$( '#btn_submit_fq' ).on( 'click', function () {
    try {
        formulaParseada =
formulaQuimicaParser.parse( $( '#formula_quimica' ).val() );
        utilitario.criaListaElementosTela( formulaParseada );
    } catch ( e ) {
        alert( e );
    }
});

```

O método `parse(String)` recebe como parâmetro a fórmula química informada para validação. A partir deste momento, são executadas algumas etapas descritas abaixo.

Na primeira etapa, a `String` é separada em uma lista com base em cada letra maiúscula que a fórmula química apresenta. Na segunda etapa cada elemento separado na lista, agora composto por um elemento químico e a quantidade de vezes que o mesmo aparece na fórmula, é percorrida e separada novamente. Desta vez o elemento químico é separado da quantidade, e caso não seja possível fazer essa separação, pois nenhum número foi identificado, assume-se que a quantidade do mesmo é um. Na terceira etapa cada elemento é submetido a uma verificação, o mesmo deve estar presente na lista `SIGLAS` da classe `ElementosQuimicos`.

Se todas estas etapas ocorrerem sem falhas, o elemento químico é adicionado a uma lista que será o retorno do método `parse(String)`. Caso alguma validação não esteja de

acordo uma exceção é imediatamente lançada para que o usuário seja informado do ocorrido. O Quadro 7 mostra o método acima descrito.

Quadro 7 – Método de Parse da Fórmula Química

```

var isElementoQuimicoValido = function (simbolo) {
    return _.includes(ElementosQuimicos.SIGLAS, simbolo);
};
self.parse = function (formula) {
    var formulaSplited;
    var elementos = [];
    if (formula) {
        formulaSplited = formula.split(/(?=[A-Z])/);
        formulaSplited.forEach(function (elemento) {
            var elementoSplited, elementoQuimico, qtdElemento;
            elementoSplited = elemento.split(/(?=[0-9])/);
            elementoQuimico = _.get(elementoSplited, 0);
            qtdElemento = _.get(elementoSplited, 1, 1);
            for (var i = 2; i < elementoSplited.length; i++) {
                qtdElemento += _.get(elementoSplited, 2, 1);
            }
            if (isNaN(qtdElemento)) {
                throw new EvalError('Elemento Químico ' +
                    elementoQuimico + ' com quantidade inválida ' +
                    '(' + qtdElemento + ')');
            } else {
                if (isElementoQuimicoValido(elementoQuimico) === true)
                {
                    for (i = 0; i < parseInt(qtdElemento); i++) {
                        elementos.push({sigla: elementoQuimico});
                    }
                } else {
                    throw new EvalError('Elemento Químico ' +
                        elementoQuimico + ' não encontrado!');
                }
            }
        });
    } else {
        throw new EvalError('Valor não pode ser nula/branco')
    }
    return elementos;
};

```

### 3.3.2.2 Desenho da Fórmula Estrutural

Após a criação da lista de elementos, o usuário poderá escolher um elemento da lista. Cada elemento da lista está diretamente associado a uma função por meio do evento de *click*, e uma vez que o elemento for selecionado, a função `listaElementos.click.elemento()` será executada. Esta função adiciona o elemento selecionado a molécula, por meio do método `adicionaAtomo(Number, String)` e estabelece este átomo como o `atomoCorrente` da classe `FormulaEstruturalDrawer`, pelo método `setAtomoCorrente()`. O Quadro 8 mostra os detalhes do método.

Quadro 8 – Método para adicionar átomo na molécula

```
$('#listaElementos').on('click', '.elementoQuimico', function () {
    var atomo = molecula.adicionaAtomo(this.id, $(this).text());
    desenho2d.setAtomoCorrente(atomo);
});
```

O outro método que poderá ser chamado pelo usuário neste momento é o `listaElementos.click.tipoLigacao()`, que determina qual o modo de ligação que será utilizada pela classe `FormulaEstruturalDrawer`. O Quadro 9 mostra detalhes da implementação do método.

Quadro 9 – Método para a seleção do modo de ligação a ser utilizado

```
$('#listaElementos').on('click', '.tipoLigacao', function () {
    if (this.id === 'ligacaoSimples') {
        desenho2d.setModoLigacao(ModoLigacao.SIMPLES);
    } else {
        desenho2d.setModoLigacao(ModoLigacao.DUPLA);
    }
});
```

O método `adicionaAtomo(Number, String)` recebe como parâmetro o *id* do elemento químico e a sigla do mesmo. Após, os dados deste elemento são retirados da lista `ELEMENTOS` da classe `ElementosQuimicos`, e então um objeto de átomo é criado com base no elemento citado, e este objeto é então retornado para que possa ser utilizado pelo método `setAtomoCorrente(Atomo)`. Uma vez que o `atomoCorrente` é estabelecido, o usuário faz a interação com a Área de Desenho, selecionando a coordenada que o átomo deve ser posicionado. Essa interação ocorre, pois a classe `FormulaEstruturalDrawer` possui dois métodos que estão ligados os eventos de `onMouseUp` e `onMouseDown` do mouse dentro da Área de Desenho. Caso o `atomoCorrente` não esteja estabelecido, mas sim um `ModoLigação`, o usuário deve selecionar dois átomos que já estão previamente posicionados para que o desenho da ligação seja feito. O Quadro 10 mostra detalhes dos dois métodos de interação com o mouse.

Quadro 10 – Métodos onMouseUp() e onMouseDown()

```

var onMouseUp = function () {
  if (atomoCorrente) {
    var atomoId = atomoCorrente.getId();
    atomoCorrente = null;
    onMouseUpCallback(atomoId);
  }
};

var onMouseDown = function (evento) {
  if (self.isModoLigacao()) {
    if (objetoAnterior !== null && objetoCorrente !== null) {
      if (_.isEqual(modoSimples, ModoSimples.SIMPLES)) {
        desenhaLigacaoSimples([objetoAnterior.left,
objetoAnterior.top, objetoCorrente.left, objetoCorrente.top]);
      } else {
        desenhaLigacaoDupla([objetoAnterior.left,
objetoAnterior.top, objetoCorrente.left, objetoCorrente.top])
      }
      onMouseDownCallback(modoSimples, objetoCorrente.atomoId,
objetoAnterior.atomoId);
      objetoAnterior = null;
      objetoCorrente = null;
      modoSimples = null;
    }
  } else {
    if (atomoCorrente) {
      desenhaAtomo(canvas.getPointer(evento.e));
    }
  }
  canvas.renderAll();
};

```

Pode-se observar que dependendo de qual método foi previamente chamado o desenho feito será diferente. O método `desenhaAtomo(Object)`, irá chamar o *framework* `Fabric.js` e adicionar um objeto do tipo `Circle` ao `canvas`. Já o método `desenhaLigacaoSimples()` e `desenhaLigacaoDupla()` irá adicionar um objeto do tipo `Line` ao `canvas`.

### 3.3.2.3 Geração tridimensional da Molécula

Para que a molécula tridimensional seja visualizada o usuário deve fazer a chamada do método `btn_gera3d.click()` da classe `ViseduQuimica`. Este método executará todas as chamadas de validações necessárias. Os métodos de validação são: `identificaTipoLigacoes()`, `calculaRegraDoOcteto()` e `valida(Number)` todos da classe `Molecula`, e em seguida a aplicação irá fazer a chamada do método `desenhaMolecula(Molecula)` da classe `DesenhaMolecula3D`.

A primeira validação ocorre com a invocação do método `identificaTipoLigacoes()`, que percorre todas as ligações e faz uma relação com o Quadro 1. O Quadro 11 mostra os detalhes da implementação do método.

Quadro 11 – Identificação dos Tipos de Ligação

```

self.identificaTipoLigacoes = function () {
    ligacoes.forEach(function (ligacao) {
        ...

        if ((origem.getGrupo() === GrupoElemento.METAIS_ALCALINOS
|| origem.getGrupo() === GrupoElemento.METAIS_ALCALINOS_TERROSOS)
        && destino.getGrupo() === GrupoElemento.HIDROGENIO) {
            ...
        }
        if ((destino.getGrupo() === GrupoElemento.METAIS_ALCALINOS
|| destino.getGrupo() === GrupoElemento.METAIS_ALCALINOS_TERROSOS)
        && origem.getGrupo() === GrupoElemento.HIDROGENIO) {
            if (destino.getEletronegatividade() > 1.0) {
                ligacao.setTipoLigacao(TipoLigacao.COVALENTE);
            } else {
                ligacao.setTipoLigacao(TipoLigacao.IONICA);
            }
        }
        ...
        if (origem.getGrupo().grandeGrupo === GrandeGrupo.METAIS
&& destino.getGrupo().grandeGrupo === GrandeGrupo.METAIS) {
            ligacao.setTipoLigacao(TipoLigacao.METALICA);
        }
        if ((origem.getGrupo().grandeGrupo === GrandeGrupo.METAIS
&& (destino.getGrupo() === GrupoElemento.OUTROS_METAIS
|| destino.getGrupo() === GrupoElemento.NAO_METAIS))
||
        (destino.getGrupo().grandeGrupo === GrandeGrupo.METAIS
&& (origem.getGrupo() === GrupoElemento.OUTROS_METAIS
|| origem.getGrupo() === GrupoElemento.NAO_METAIS))) {
            if (origem.getEletronegatividade() -
destino.getEletronegatividade() < 1.7
|| destino.getEletronegatividade() -
origem.getEletronegatividade() < 1.7) {
                ligacao.setTipoLigacao(TipoLigacao.COVALENTE);
            } else {
                ligacao.setTipoLigacao(TipoLigacao.METALICA);
            }
        }
        if (ligacao.getTipoLigacao() === null) {
            throw Error('Tipo de Ligação entre ' +
origem.getSimbolo() + ' e ' + destino.getSimbolo() + ' não existe!');
        }
    }
};

```

A validação segue através do método `calculaRegrsadsdaDoOcteto()`. Este método percorre as ligações verificando os seus tipos, para que os elétrons possam ser incrementados ou decrementados de acordo com o que foi anteriormente descrito na seção 2.2.1. O Quadro 12 mostra os detalhes da implementação.

Quadro 12 – Cálculo da Regra do Octeto

```

self.calculaRegraDoOcteto = function () {

    ligacoes.forEach(function (ligacao) {

        var origem = self.getAtomoPorId(ligacao.getAtomoOrigem());
        var destino = self.getAtomoPorId(ligacao.getAtomoDestino());

        if (ligacao.getTipoLigacao() === TipoLigacao.COVALENTE) {
            origem.incrementaQtdEletronsLivresCamandaValencia();
            destino.incrementaQtdEletronsLivresCamandaValencia();
            if (ligacao.getModoLigacao() === ModoLigacao.DUPLA) {
                origem.incrementaQtdEletronsLivresCamandaValencia();
                destino.incrementaQtdEletronsLivresCamandaValencia();
            }
        } else {
            if (ligacao.getTipoLigacao() === TipoLigacao.IONICA &&
                ligacao.getModoLigacao() !== ModoLigacao.SIMPLES) {
                throw new Error('Ligações Iônicas são sempre
                simples');
            } else {
                if (origem.getEletronegatividade() >
                    destino.getEletronegatividade()) {
                    origem.incrementaQtdEletronsLivresCamandaValencia();
                    destino.decrementaQtdEletronsLivresCamandaValencia();
                } else {
                    origem.decrementaQtdEletronsLivresCamandaValencia();
                    destino.incrementaQtdEletronsLivresCamandaValencia();
                }
            }
        }
    });
};

```

A última validação a ser executada é a do método `valida(Number)`. Este método recebe como parâmetro a quantidade de elementos que a fórmula química possui. Esta informação é enviada para que a primeira verificação identifique se todos os elementos foram utilizados no desenho. Em seguida faz-se uma validação para certificar que todos os átomos estão presentes em ao menos uma ligação. No último passo é verificado se o valor calculado para regra do octeto está de acordo com o que deveria, conforme descrito na seção 2.2. O Quadro 13 mostra os detalhes da implementação.

Quadro 13 – Método valida()

```

self.valida = function (qtdElementosFormula) {
    if (qtdElementosFormula !== atomos.length) {
        throw new Error('Nem todos os elementos da formula foram
utilizados! [Qtd Elementos: ' + qtdElementosFormula +
        ' ] [Qtd Atomos: ' + atomos.length + ' ]');
    }

    atomos.forEach(function (atomo) {

        if (atomos.length === 2) {
            atomo.setBiMolecular(true);
        }

        var achou = false;
        ligacoes.forEach(function (ligacao) {
            if (ligacao.getAtomoOrigem() === atomo.getId() ||
ligacao.getAtomoDestino() === atomo.getId()) {
                achou = true;
            }
            if (ligacao.getTipoLigacao() === TipoLigacao.IONICA &&
ligacao.getModoLigacao() !== ModoLigacao.SIMPLES) {
                throw new Error('Ligações Iônicas são sempre
simples');
            }
            if (ligacao.getTipoLigacao() === TipoLigacao.METALICA) {
                throw new Error('Ligações Metálicas não são tratadas
pelo aplicativo');
            }
        });
        if (atomo.getQtdEletronsLivresCamadaValencia() !== 0) {
            if (atomo.getValorEletronsRegraDoOcteto() !==
atomo.getQtdEletronsLivresCamadaValencia()) {
                throw new Error('Molécula não esta obedecendo a regra
do octeto ' + ' [ Atomo => ' + atomo.getSimbolo() +
                ' Valor Regra do Octeto => ' +
atomo.getValorEletronsRegraDoOcteto() + ' Eletrons Livres => ' +
atomo.getQtdEletronsLivresCamadaValencia() + ' ]')
            }
        }
        if (!achou) {
            throw new Error('Atomo ' + atomo.getSimbolo() + ' não é
usado em nenhuma ligação! [1]');
        }
        if (atomo.getQtdLigacoes() === 0) {
            throw new Error('Atomo ' + atomo.getSimbolo() + ' não é
usado em nenhuma ligação! [2]');
        }
    });
};

```

Desta forma ao ser executado o método `desenhaMolecula()` percorre todos os átomos da molécula, ordenados de forma decrescente pela quantidade de ligações. O primeiro átomo é posicionado nas coordenadas 0, 0, 0. Os próximos átomos a serem posicionados são aqueles ligados ao átomo principal. Suas coordenadas são obtidas considerando o tipo de geometria conforme o Quadro 2 (Seção 0). O Quadro 14 mostra os detalhes da implementação

Quadro 14 – Cálculo para seleção do próximo ponto

```

if (tipoGeometria.eixos === "xyz") {
    x2 = x + 4 * Math.cos(tipoGeometria.angulos[countLigacoes][1]) *
Math.sin(tipoGeometria.angulos[countLigacoes][0]);
    y2 = y + 4 * Math.sin(tipoGeometria.angulos[countLigacoes][1]);
    z2 = z + 4 * Math.cos(tipoGeometria.angulos[countLigacoes][1]) *
Math.cos(tipoGeometria.angulos[countLigacoes][0])
} else {
    x2 = x + Math.cos(tipoGeometria.angulos[countLigacoes]) * 4;
    y2 = y + Math.sin(tipoGeometria.angulos[countLigacoes]) * 4;
    z2 = z;
}
}

```

### 3.3.3 Operacionalidade da implementação

Ao iniciar a aplicação o usuário é remetido à interface principal da aplicação. Esta interface é formada por três áreas principais: a área de entrada que é formada por uma caixa de texto utilizada para a inserção da fórmula química, assim como uma relação com os elementos químicos retirados da fórmula. É formada ainda por uma área de desenho, para que o usuário possa fazer o desenho da fórmula estrutural e uma área de visualização, onde o usuário pode ver o resultado tridimensional da molécula. A Figura 12 ilustra a interface principal e os elementos da tela.

Figura 12 – Tela principal da aplicação



Após o usuário digitar a fórmula química o mesmo clica no botão à direita da caixa de texto, em seguida se não houver nenhuma falha na fórmula química os elementos são apresentados na lista de elementos, a Figura 14 demonstra essa situação. Em caso de erro na fórmula química uma mensagem de erro é apresentada ao usuário, como pode ser visto na Figura 13. Em seguida o usuário deve selecionar um dos elementos da lista de elementos e

clique em um ponto dentro da área de desenho, desta forma o elemento selecionado será adicionado à área de desenho.

Figura 13 – Erro ao executar parse da fórmula química

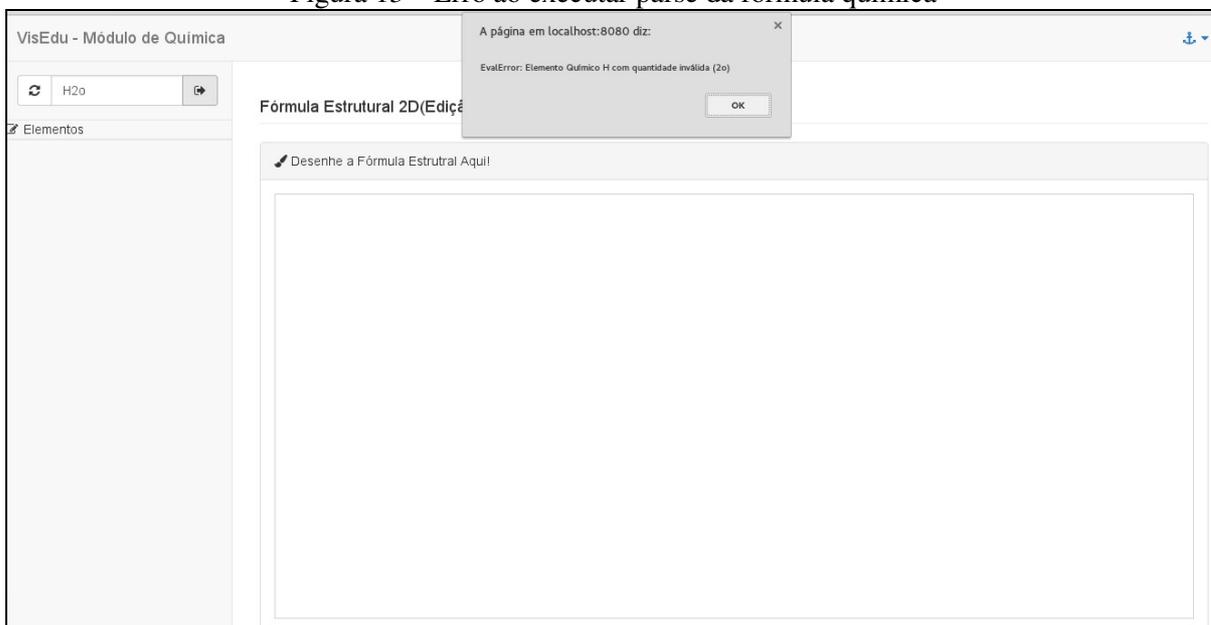


Figura 14 – Parse da fórmula química sem erros



Quando todos os elementos forem adicionados, o usuário deve escolher um modo de ligação, simples ou dupla, e então fazer a seleção de dois átomos na área de desenho para que a ligação entre eles possa ocorrer. A figura Figura 15 demonstra a representação de uma molécula de água ( $H_2O$ ), com todos seus átomos devidamente dispostos, assim como suas ligações feitas. Após a criação de todas as ligações necessárias, o usuário poderá clicar no botão com símbolo de engrenagem, para que então a visualização tridimensional seja apresentada, caso a molécula representada não esteja de acordo com a regra do octeto, suas

ligações não sejam validas ou nem todos os elementos tenham sido utilizados, uma mensagem de erro será apresentada ao usuário. Caso a molécula esteja de acordo com as regras, sua representação tridimensional será gerada a Figura 16 demonstra essa situação.

Figura 15 – Molécula de água (H<sub>2</sub>O)

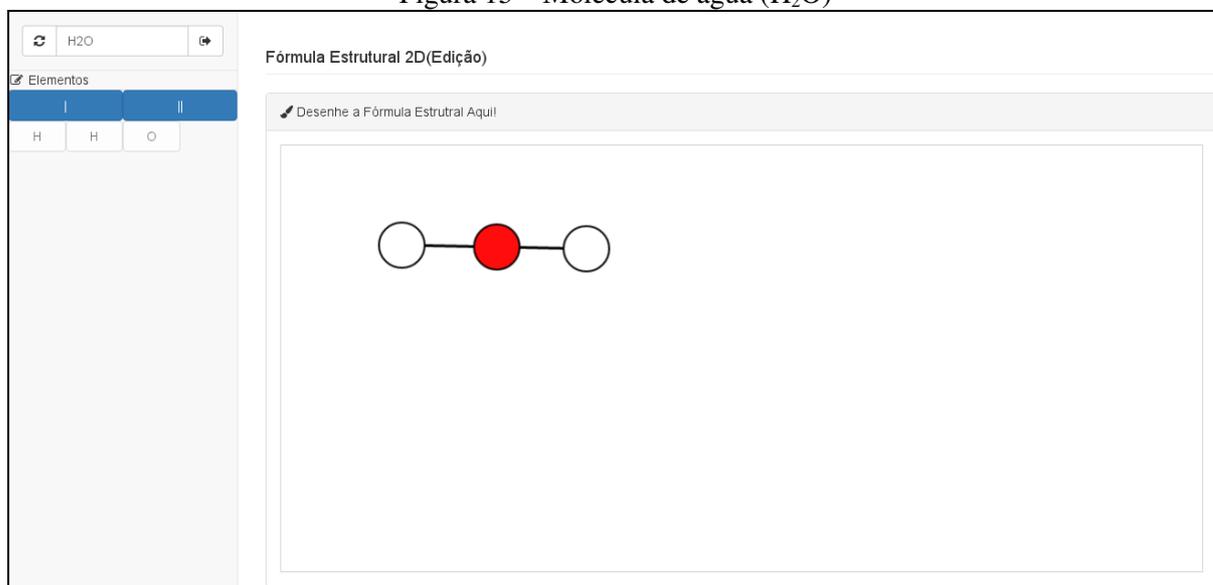
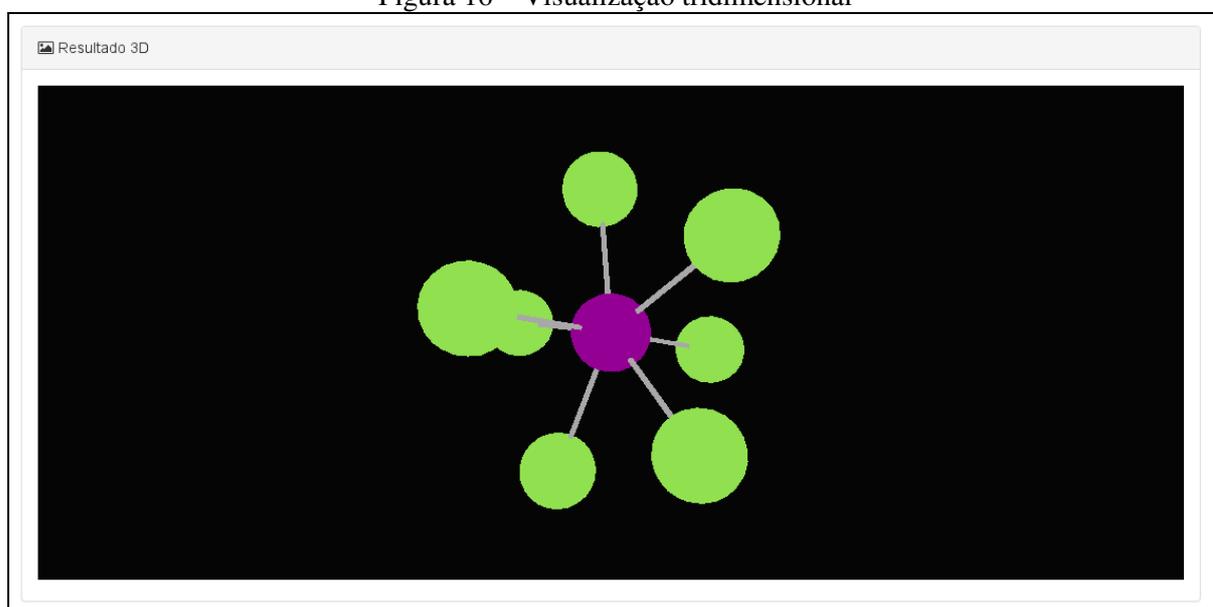


Figura 16 – Visualização tridimensional



### 3.4 RESULTADOS E DISCUSSÕES

Este trabalho apresenta uma aplicação web para a visualização de material educacional, na qual fosse possível exibir de forma tridimensional uma molécula e permitir uma alteração da posição de visualização, facilitando uma análise em vários ângulos diferentes.

O objetivo inicial deste trabalho é a criação de uma aplicação voltada para o ensino da química, em especial da área de geometria molecular. Porém após uma verificação dos livros didáticos utilizados para o ensino da disciplina de química, identificou-se uma enorme abrangência de assuntos nesta área, dificultado assim o desenvolvimento da aplicação, optou-se então pelo ensino da área de geometria molecular apenas de moléculas simples, normalmente com apenas um átomo central. Para que este objetivo fosse atingido se fez uso do motor de jogos utilizado em outros módulos do projeto VisEdu.

Inicialmente foi desenvolvido um protótipo de tela utilizando HTML5, tomando como base os aplicativos BKChem e Avogadro. Este protótipo apenas permitia a exibição tridimensional de moléculas previamente definidas, e o mesmo foi usado para testes do elemento `<canvas>`. Como o objetivo era a interação com um ambiente que permite o desenho bidimensional de uma molécula, fez-se necessária a busca por um *framework* que pudesse auxiliar este processo. Optou-se pelo uso do *framework* Fabric.js, que torna mais simples a manipulação dos eventos e objetos dentro do `<canvas>`.

Conforme proposto nos objetivos, foi desenvolvida uma validação para o desenho realizado no ambiente bidimensional pudesse ser transportado para o tridimensional sem falhas. Para que isso fosse possível fez-se uso dos conceitos químicos de: tipos de ligação, distribuição de Lewis, regra do octeto e geometria molecular. Uma vez finalizada a implementação das validações uma mudança foi necessária no protótipo original, para que agora as novas coordenadas fossem calculadas pela aplicação.

As próximas seções têm por objetivo descrever os testes realizados para avaliar os resultados do trabalho.

#### 3.4.1 Testes da aplicação

Para a execução dos testes da aplicação, foram utilizados computadores presentes na sala de aula, de uma escola de ensino médio da região de Blumenau. Foi aplicado também um questionário que será apresentado nas próximas seções. Os conceitos avaliados foram aspectos da usabilidade, a utilização em um ambiente educacional e também os recursos da aplicação. Além dos testes de usabilidade, foram realizados testes de desempenho.

#### 3.4.2 Pesquisa de opinião sobre a usabilidade

A pesquisa de opinião foi realizada com 53 alunos de primeiro e segundo ano do ensino médio de um colégio particular da região de Blumenau.

### 3.4.2.1 Metodologia

O experimento ocorreu no mês de novembro por meio de testes coletivos da aplicação com os usuários. Para realizar os testes foram utilizados computadores disponíveis em sala de aula. Por se tratar de uma aplicação web nenhum tipo de instalação precisou ser feita nestas máquinas.

### 3.4.2.2 Aplicação do teste

Para o início da avaliação, os participantes foram orientados sobre o funcionamento da aplicação e apresentados à interface da mesma. Foram instruídos na utilização do mesmo e para a execução dos testes as seguintes moléculas foram utilizadas: H<sub>2</sub>O, CO<sub>2</sub>, BF<sub>3</sub>, NaOH, C<sub>2</sub>H<sub>4</sub>, PbCl<sub>4</sub>, BrF<sub>5</sub>, SF<sub>6</sub>, IF<sub>7</sub> e SF<sub>4</sub>.

O questionário utilizado possui sete perguntas, sendo quatro delas de simples escolha e três delas descritivas. As perguntas têm como objetivo avaliar a usabilidade da aplicação e analisar a possibilidade do mesmo ser utilizado como apoio em um ambiente de ensino. Ao responder o questionário os usuários puderam deixar suas reclamações e/ou sugestões. Os resultados são apresentados nas seções seguintes. O Quadro 15 mostra as perguntas utilizadas no questionário.

Quadro 15 – Perguntas aplicadas aos usuários

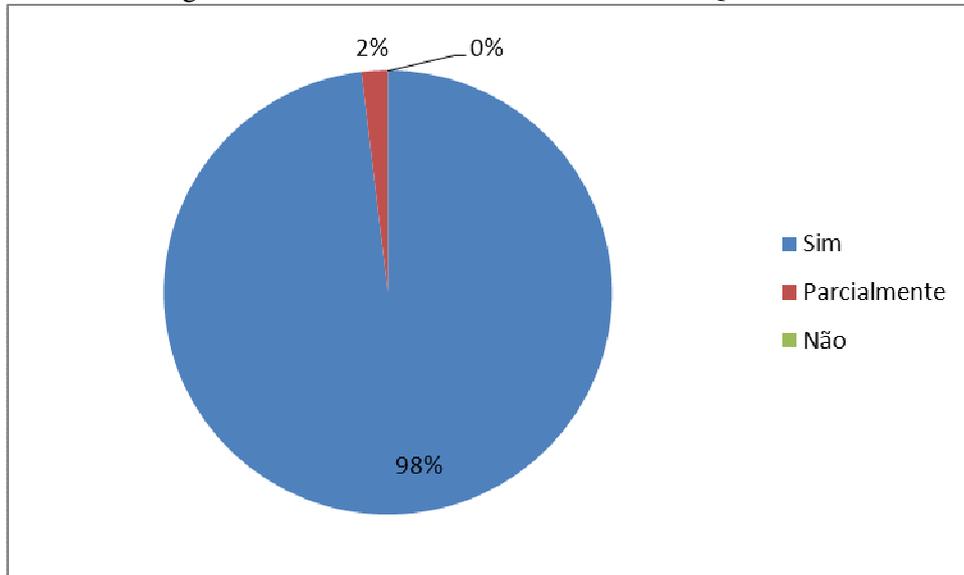
Nº	Questão	Tipo de Resposta
1	Você considera que ferramentas interativas podem auxiliar no processo de educação?	Múltipla Escolha (Sim, Parcialmente e Não).
2	No cenário atual de ensino, é viável para as instituições de ensino disponibilizarem <i>notebooks</i> para grupos de alunos?	Múltipla Escolha (Sim e Não).
3	Você considera a aplicação apresentada como algo que poderia auxiliar a fixação do conteúdo?	
4	Como você classifica a usabilidade da aplicação?	Múltipla Escolha (Ruim, Regular, Boa e Ótima).
5	Com base nas respostas anteriores, quais foram os “pontos positivos” encontrados na aplicação?	Descritiva.
6	Com base nas respostas anteriores, quais foram os “pontos negativos” encontrados na aplicação?	
7	Com base nas respostas anteriores, quais seriam suas sugestões para melhorar o aplicativo?	

### 3.4.2.3 Análise dos resultados dos questionários

As seções abaixo apresentam as opiniões e discussões obtidas partir da aplicação do resultado do questionário.

A primeira questão, “Você considera que ferramentas interativas podem auxiliar no processo de educação?” obteve como resultado percentuais os valores de 98% para Sim; 2% para Parcialmente e 0% para Não. Os resultados são apresentados na Figura 17.

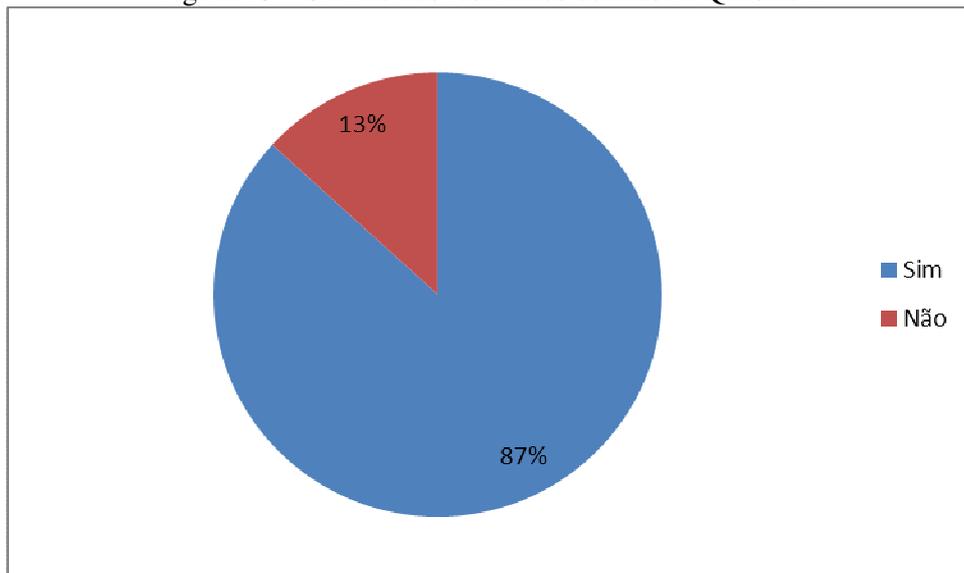
Figura 17 – Gráfico dos resultados obtidos na Questão 1



Conforme apresentado verificou-se uma conformidade com a discussão apresentada na seção 2.1, onde é verificada a importância de ferramentas interativas no processo educacional.

A segunda questão “No cenário atual de ensino, é viável para as instituições de ensino disponibilizarem *notebooks* para grupos de alunos?” os resultados percentuais obtidos foram de 87% para Sim e 13% para Não. Os resultados são apresentados na Figura 18.

Figura 18 – Gráfico dos resultados obtidos na Questão 2

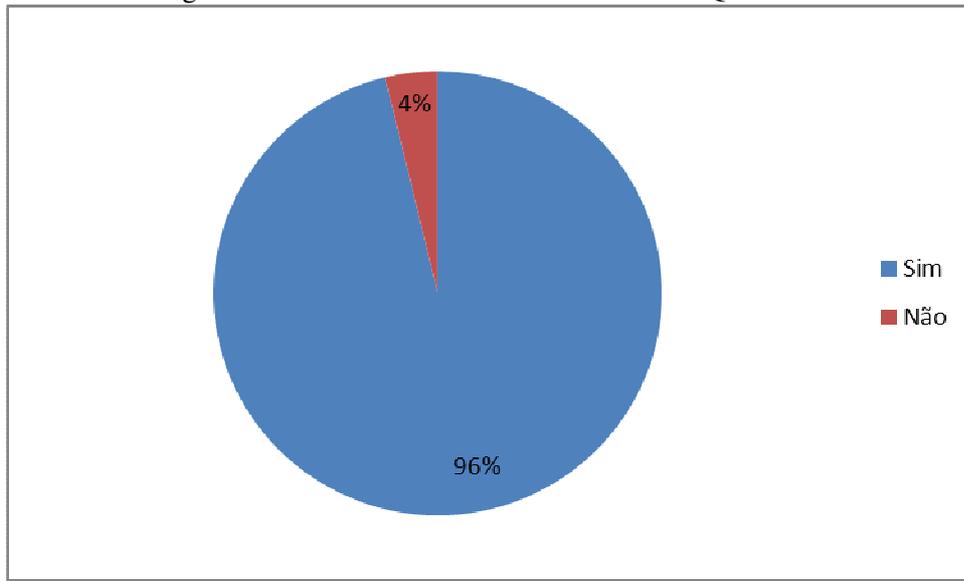


É importante ressaltar neste item que a pesquisa foi realizada em uma escola particular, portanto a maioria de respostas em Sim é condizente com a posição social e o meio em que os

adolescentes vivem. Pode-se questionar neste ponto se a realização do mesmo questionário em uma escola pública de periferia apresentaria os mesmos resultados.

A terceira questão “Você considera a aplicação apresentada como algo que poderia auxiliar a fixação do conteúdo?” os resultados percentuais obtidos foram de 96% para Sim e 4% para Não. Os resultados são apresentados na Figura 19.

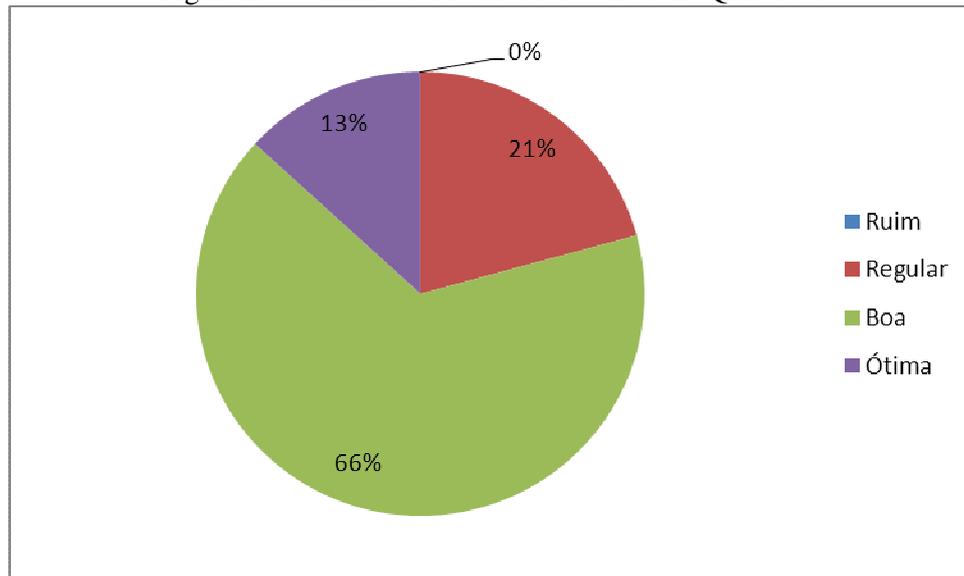
Figura 19 – Gráfico dos resultados obtidos na Questão 3



Novamente, verificou-se uma conformidade com a discussão apresentada na seção 2.1 nas opiniões obtidas. Conforme visto nos resultados das questões um, dois e três, a grande maioria dos estudantes entrevistados concordam que ferramentas interativas provém um auxílio ao processo de ensino, e concordam, dentro de sua atual realidade econômica familiar que a escola conseguiria disponibilizar as ferramentas necessárias para utilização de aplicações interativas. Assim a maioria considera a aplicação testada como um apoio educacional na mediação da aprendizagem.

Na quarta questão “Como você classifica a usabilidade da aplicação?” os resultados percentuais obtidos foram 66% para Boa, 21% para Regular, 13% para Ótima e 0% para Ruim. Os resultados são apresentados na Figura 20.

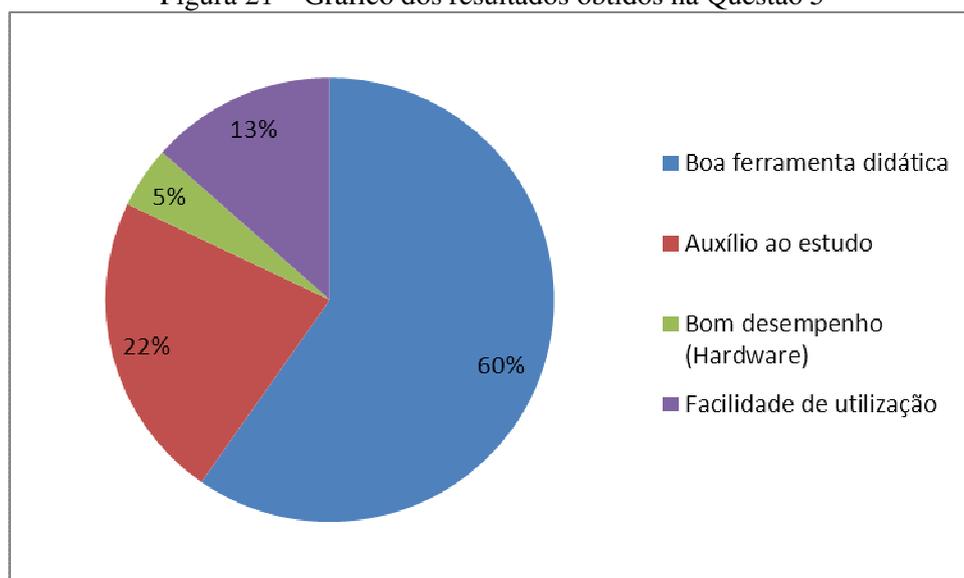
Figura 20 – Gráfico dos resultados obtidos na Questão 4



Pode-se verificar que a maioria dos estudantes entrevistados considera que a aplicação atente as necessidades de usabilidade de acordo com o que foi proposto pela mesma, porém nota-se que há uma necessidade de aprimorar alguns dos pontos da aplicação.

Na quinta questão “Com base nas respostas anteriores, quais foram os pontos positivos encontrados na aplicação?”, fez-se necessária a leitura de todas as respostas para que as mesmas puderam ser agrupadas em quatro pontos principais. Os resultados percentuais obtidos foram 60% para Boa Ferramenta Didática, 22% para Auxílio ao estudo, principalmente em tarefas escolares, 13% para Facilidade de Utilização e 5% para Bom Desempenho. A Figura 21 apresenta os resultados obtidos.

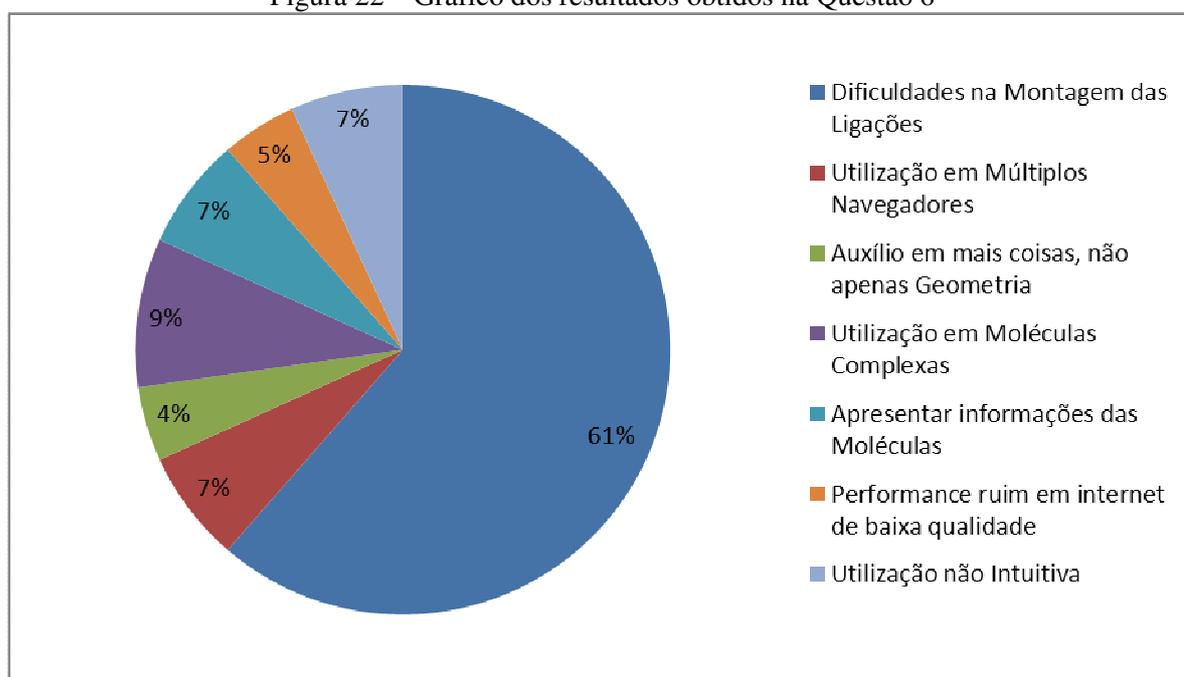
Figura 21 – Gráfico dos resultados obtidos na Questão 5



Por se tratar de uma questão descritiva, os resultados tiveram de ser analisados e agrupados.

Na sexta questão “Com base nas respostas anteriores, quais foram os pontos negativos encontrados na aplicação?”, assim como na questão cinco, precisou-se fazer uma leitura das questões para que fosse possível agrupar os resultados obtidos. Os resultados percentuais obtidos foram 61% para Dificuldades na Montagem das ligações, 9% para Utilização de Moléculas Complexas, 7% Utilização em Múltiplos Navegadores, 7% Apresentar mais Informações das Moléculas, 7% para Utilização não Intuitiva, 5% para Performance ruim em internet de baixa Qualidade e 4% para Auxílio em mais assuntos além da geometria molecular. A Figura 22 apresenta os resultados obtidos.

Figura 22 – Gráfico dos resultados obtidos na Questão 6

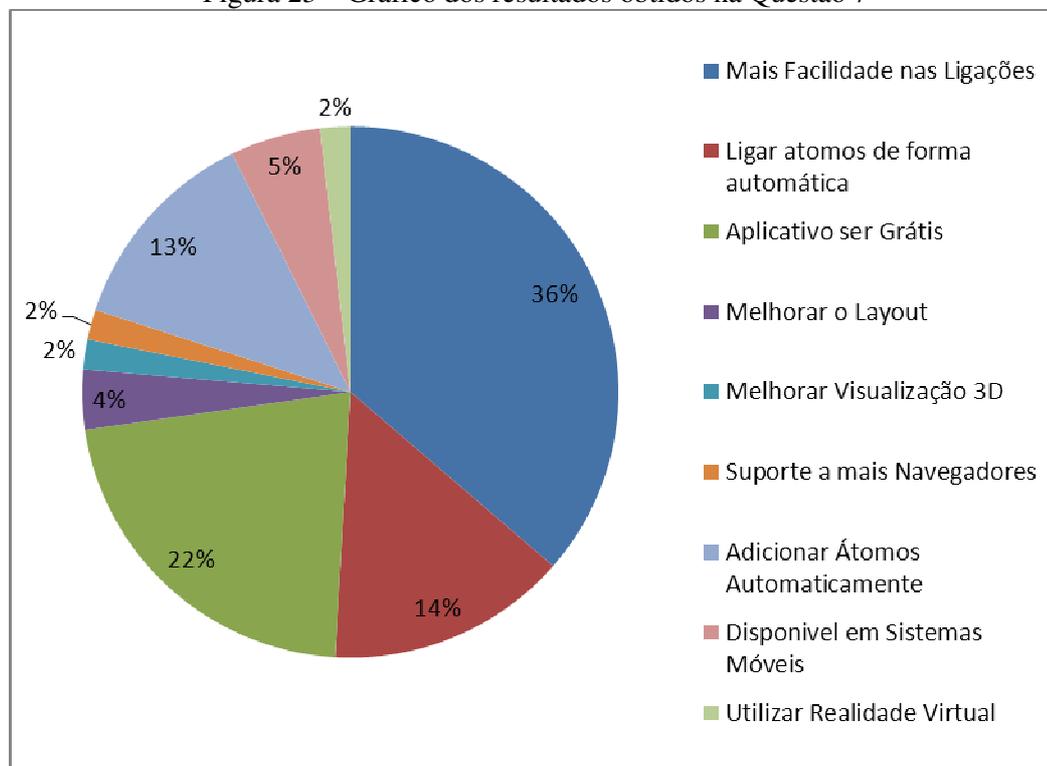


Pode-se observar que a maioria dos estudantes entrevistados, apontou como dificuldade a geração das ligações entre os átomos, sendo este então o principal ponto a ser melhorado na aplicação, porém todos os resultados podem ser vistos como melhorias válidas para uma versão posterior.

Na sétima e última questão “Com base nas respostas anteriores, quais seriam suas sugestões para melhorar o aplicativo?” os resultados percentuais obtidos para essa questão foram 36% para Facilitar o processo de Ligação, 22% para que o aplicativo seja disponibilizado de forma gratuita, 14% para que as ligações sejam feitas de forma automática, 13% para que os átomos sejam adicionados automaticamente após a digitação da fórmula, 5% para que o sistema seja disponibilizado em dispositivos móveis, 4% para melhoras de *layout*, 2% utilização de Realidade Virtual, 2% para suporte a mais navegadores, 2% para que a

visualização tridimensional seja melhorada. A Figura 23 apresenta o gráfico com os resultados obtidos.

Figura 23 – Gráfico dos resultados obtidos na Questão 7



Verifica-se aqui novamente a problemática da forma como a aplicação lida com as ligações, no entanto têm-se outras sugestões interessantes, como por exemplo, o uso de Realidade Virtual para uma maior imersão do aluno na utilização da aplicação.

### 3.4.3 Teste de desempenho

Para avaliar o desempenho da aplicação foi realizada somente a operação de adição de vários átomos na área de visualização tridimensional. Os testes foram feitos em dois navegadores diferentes, o Mozilla Firefox e o Google Chrome, descritos na seção 3.3.1.

As ferramentas utilizadas para fazer o levantamento das informações de Frames Por Segundo (FPS) no Chrome foi o Google Developers Tools, sendo esta uma ferramenta embutida no próprio navegador, e para o Mozilla Firefox foi usado o Firebug em sua versão 2.0.11. Para os testes de memória foi utilizado o utilitário Monitor do Sistema disponível de forma nativa no ArchLinux.

Os critérios utilizados para testar o desempenho da aplicação foram o processamento de FPS em relação à quantidade de objetos adicionados na cena e o consumo de memória durante a utilização dos objetos. Estes testes foram efetuados tanto no ambiente bidimensional como no tridimensional.

Foram criadas seis baterias de testes para cada um dos ambientes, primeiramente serão descritos os testes do ambiente tridimensional para em seguida serem descritos os testes dos objetos bidimensionais.

### 3.4.3.1 Testes no ambiente Tridimensional

No Quadro 16 pode-se visualizar a análise de desempenho feita sobre o navegador Google Chrome. Nos testes feitos pode-se verificar que após a utilização de 1.280 peças o sistema parou de funcionar corretamente, sendo que a taxa de FPS também teve uma queda significativa neste ponto.

Quadro 16 – Testes de Memória e Desempenho no Google Chrome

Qtde de Objetos	FPS	Memória (MB)	Observação
40	60	202	Sistema funcionou normalmente, as taxas de FPS e memória estavam constantes durante a utilização.
80	60	237	
160	60	307	
320	60	432	
640	42~60	732	Sistema funcionou normalmente, porém as taxas de FPS variaram constantemente entre os eventos de zoom e rotação.
1.280	12~42	1300	Sistema não estava funcionando corretamente, o Google Chrome travou por aproximadamente 30 segundos até que conseguiu renderizar todos os objetos, a utilização dos eventos de zoom e rotação ficaram prejudicadas.

No Quadro 17 pode-se observar os resultados obtidos com a análise de desempenho no navegador Firefox, vê-se que a desempenho no Firefox é inferior ao desempenho no Google Chrome.

Quadro 17 – Testes de Memória e Desempenho no Firefox

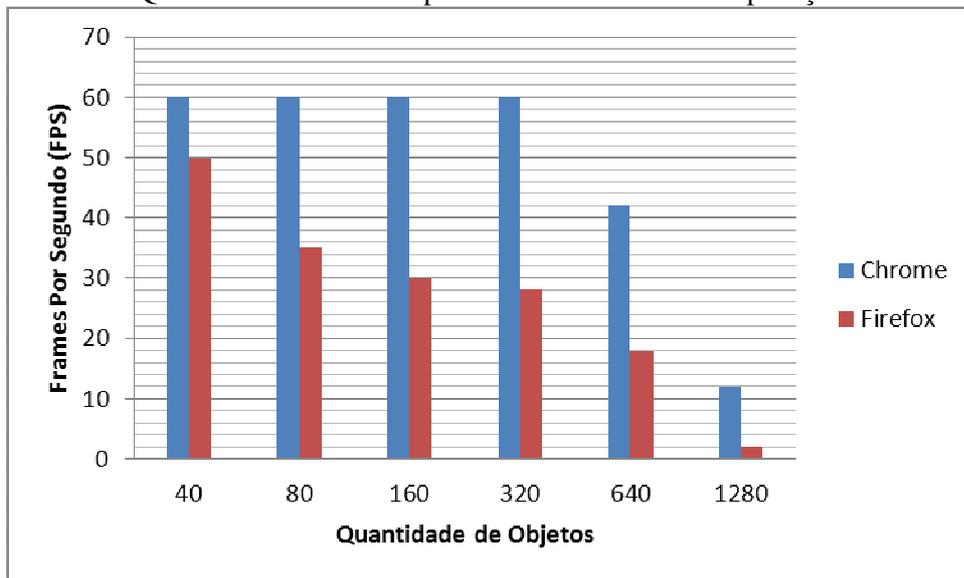
Qtde de Objetos	FPS	Memória (MB)	Observação
40	50	500	Sistema funcionou normalmente, as taxas de FPS e memória estavam constantes durante a utilização.
80	35	600	Sistema funcionou normalmente, porém as .taxas de FPS estava muito baixas mesmo sem nenhuma interação com o ambiente.
160	30	650	
320	28	700	
640	18	1.500	Sistema não funcionou normalmente, e as .taxas de FPS estava muito baixas mesmo sem nenhuma interação com o ambiente, o inverso pode ser dito da memória estava com um consumo excessivo
1.280	2,1	2.000	Sistema não estava funcionando corretamente, o Firefox travou por aproximadamente 60 segundos até que conseguiu renderizar todos os objetos, a utilização dos eventos de zoom e rotação ficaram prejudicadas.

Diante das avaliações efetuadas, percebe-se que o navegador Firefox é o menos indicado para o uso do sistema. Seu consumo de memória é superior e sua taxa de FPS não

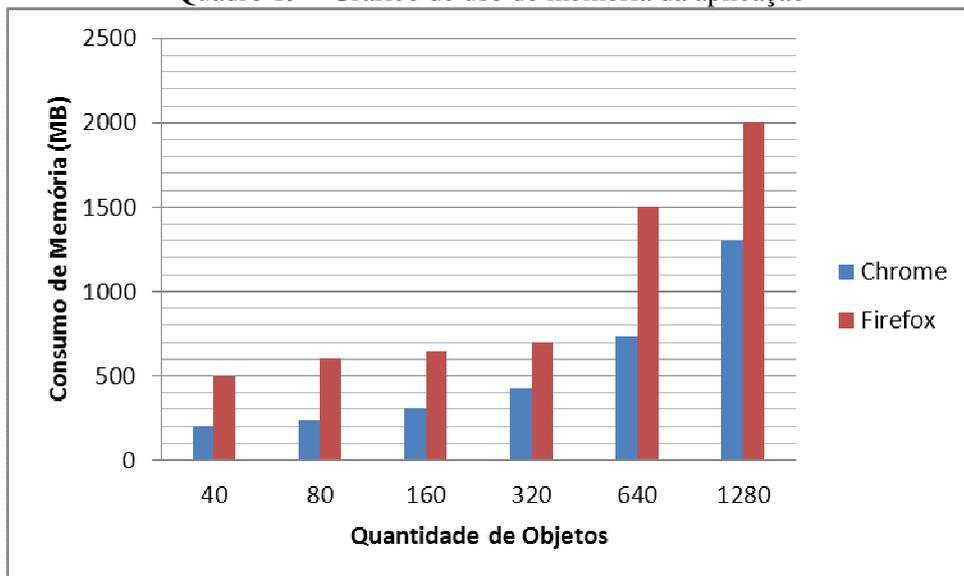
chegou em 60 em nenhum momento durante os testes da aplicação. O Internet Explorer não foi utilizado como ambiente de testes, pois não está disponível de forma nativa no Linux

Já o Google Chrome, tem tanto seu consumo de memória baixo como sua taxa de FPS acima do Firefox, porém sua grande vantagem é que a taxa de FPS se mantém relativamente alta mesmo com uma grande quantidade de objetos em cena. O Quadro 18 demonstra o gráfico de comparação entre os navegadores, com relação aos testes de FPS que foram efetuados, já o Quadro 19 mostra o gráfico de comparação com relação ao consumo de memória.

Quadro 18 – Gráfico de processamento de FPS da aplicação



Quadro 19 – Gráfico de uso de memória da aplicação



Com bases nos resultados obtidos verificou-se que a aplicação funciona melhor no navegador Google Chrome.

### 3.4.3.2 Testes no ambiente bidimensional

No Quadro 20 pode-se visualizar a análise de desempenho feita sobre o navegador Google Chrome. Nos testes feitos pode-se verificar que mesmo após a utilização de 1.280 peças o sistema funcionou normalmente.

Quadro 20 – Testes de Memória e Desempenho no Google Chrome

Qtde de Objetos	FPS	Memória (MB)	Observação
40	60	73	Sistema funcionou normalmente, as taxas de FPS e memória estavam constantes durante a utilização.
80	60	230	
160	60	250	
320	55	260	
640	50	265	
1.280	40	270	

No Quadro 21 pode-se observar os resultados obtidos com a análise de desempenho no navegador Firefox, vê-se que o desempenho no Firefox é inferior ao desempenho no Google Chrome. Neste caso o sistema não pode ser testando acima dos 320 átomos em cena, pois o navegador deixou de responder e teve de ser reiniciado.

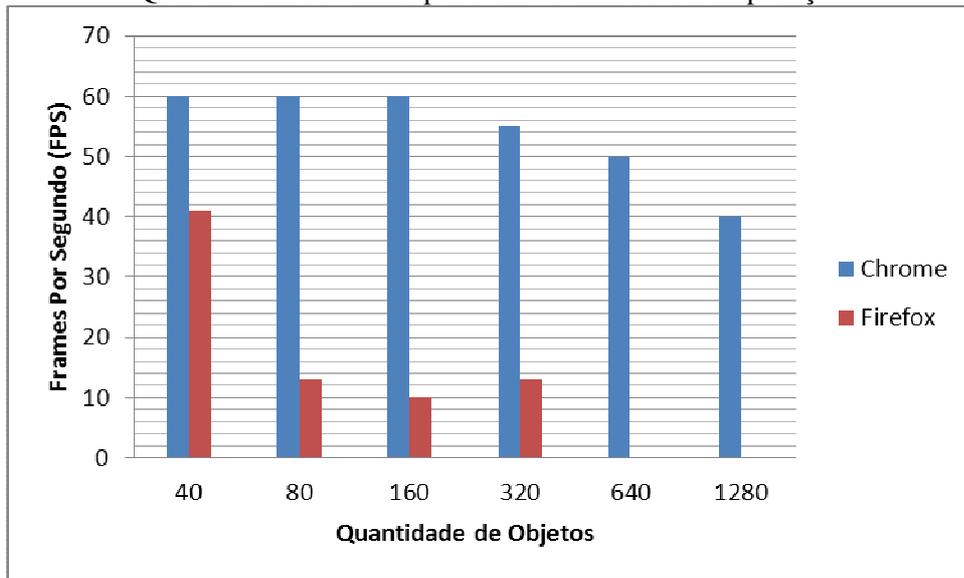
Quadro 21 – Testes de Memória e Desempenho no Firefox

Qtde de Objetos	FPS	Memória (MB)	Observação
40	41	600	Sistema funcionou normalmente, as taxas de FPS e memória estavam constantes durante a utilização.
80	13	1200	Sistema funcionou normalmente, porém as .taxas de FPS estava muito baixas mesmo sem nenhuma interação com o ambiente.
160	10	1300	
320	13	1300	
640	-	-	Sistema travou e não terminou de carregar as peças
1.280	-	-	Sistema travou e não terminou de carregar as peças

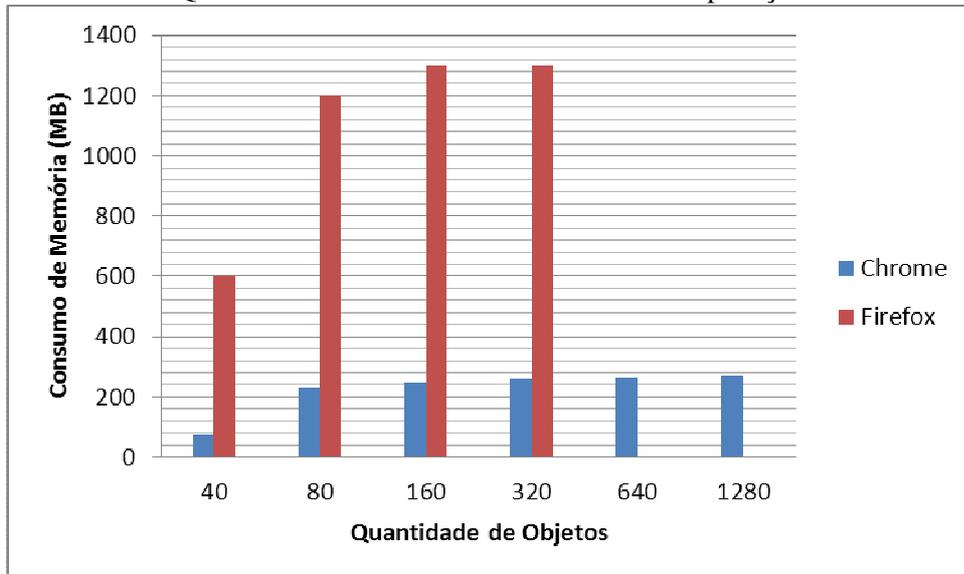
Diante das avaliações efetuadas, percebe-se que o navegador Firefox é o menos indicado para o uso do sistema. Seu consumo de memória é superior e sua taxa de FPS não chegou em 60 em nenhum momento durante os testes da aplicação. O Internet Explorer não foi utilizado como ambiente de testes, pois não está disponível de forma nativa no Linux

Já o Google Chrome, tem tanto seu consumo de memória baixo como sua taxa de FPS acima do Firefox, porem sua grande vantagem é que a taxa de FPS se mantém relativamente alta mesmo com uma grande quantidade de objetos em cena. O Quadro 22 demonstra o gráfico de comparação entre os navegadores, com relação aos testes de FPS que foram efetuados, já o Quadro 23 o mostra o gráfico de comparação com relação ao consumo de memória.

Quadro 22 – Gráfico de processamento de FPS da aplicação



Quadro 23 – Gráfico de uso de memória da aplicação



Assim como os resultados obtidos com os testes no ambiente tridimensional, o navegador mais adequado para o uso é o Google Chrome.

#### 3.4.4 Comparativo entre trabalho desenvolvido e seus correlatos

Nessa seção será apresentada uma comparação entre as principais características da aplicação desenvolvida com as características identificadas nos trabalhos correlatos (Quadro 24).

Quadro 24 – Características dos trabalhos correlatos e do trabalho desenvolvido.

Caraterística	BKChem	Avogadro	Desenvolvido
ambiente de edição 2D	x		x
ambiente de edição 3D		x	
visualização 3D		x	x

validação da estrutura química	x		x
exportar resultado	x	x	
ambiente web			x
leitura de estruturas InChI	x		

A aplicação desenvolvida englobou algumas das características dos trabalhos correlatos, como por exemplo, a validação da fórmula química e a possibilidade da visualização da molécula em um ambiente tridimensional e adicionou um novo recurso. O recurso adicionado foi à possibilidade de a aplicação ser executada em um ambiente web, sem a necessidade de nenhum tipo de instalação. Isso tem com objetivo facilitar o uso da aplicação em qualquer computador.

O novo recurso adicionado também pode ser visto como o principal diferencial da aplicação em relação aos demais softwares. Este diferencial também permite que a aplicação seja executada em um ambiente sem acesso a internet, pois por se tratar de uma aplicação desenvolvida inteiramente com a linguagem Javascript, apenas a execução do arquivo principal da aplicação no navegador, irá permitir o uso da mesma.

## 4 CONCLUSÕES

Este trabalho apresentou o desenvolvimento de um ambiente para o ensino da Química, em especial o conceito de geometria molecular. Para a criação deste ambiente proposto fez-se uso do motor de jogos desenvolvido por Montibeler (2014) assim como do *framework* Fabric.js, ambos se mostraram a excelentes escolhas, tanto durante o desenvolvimento como durante a utilização da aplicação. O desempenho mostrado pela aplicação dentro dos limites do que fora proposto é mais que satisfatório, conseguindo renderizar uma quantidade significativa de objetos em uma cena.

Mesmo tendo atingido os objetivos propostos pelo trabalho, para que a aplicação se torne utilizável em qualquer computador alguns ajustes são necessários. Sendo que durante os testes foi identificado um problema com a utilização de moléculas complexas, com mais de um átomo principal. Além disto, a adição de mais informações no ambiente tridimensional seria um auxílio muito grande para os estudantes, para que os mesmos pudessem entender melhor os resultados gerados.

Percebe-se também com a pesquisa de opinião sobre a usabilidade que o aplicativo teve boa aceitação como uma possibilidade de auxílio no ensino. Dentro da pesquisa nota-se também que para que os alunos se sintam mais dispostos a utilizar a aplicação, alguns ajustes são necessários na parte de usabilidade e de layout.

### 4.1 EXTENSÕES

Abaixo serão descritas algumas das possibilidades de extensões dentro do trabalho desenvolvido, sendo elas:

- a) melhorar a forma com que as ligações são feitas;
- b) adicionar mais informações sobre a molécula no ambiente tridimensional;
- c) adicionar automaticamente os átomos da fórmula no ambiente bidimensional;
- d) melhorar a visualização do ambiente tridimensional;
- e) utilizar iluminação na visualização tridimensional;
- f) permitir a utilização de mais navegadores;
- g) permitir o desenho de moléculas complexas.

## REFERÊNCIAS

- ALARCAO, Isabel. **Professores Reflexivos Em Uma Escola Reflexiva**. São Paulo: Cortez, 2003.
- ANYURU, Andreas. **Professional WebGL programming: developing 3D graphics for the web**. Chichester: Wrox, 2012.
- ATKINS, Peter W.; JONES, Loretta. **Princípios de Química: questionando a vida moderna e o meio ambiente**. 3. ed. Tradução Ricardo Bicca de Alencastro. Porto Alegre: Bookman, 2001.
- BAPTISTA, Michele M. Internet: Auxílio à educação. **Revista Biblos**, Rio Grande, v. 16, n. 1, p. 37-44, 2004. Disponível em: <<http://www.seer.furg.br/biblos/article/view/409>>. Acesso em: 27 mar. 2015.
- BONA, Berenice O. Análise de Softwares Educativos para o Ensino de Matemática nos Anos Iniciais do Ensino Fundamental. **Experiências em Ensino de Ciências**, Carazinho, v.4, n. 1, p. 35-55, 2009. Disponível em: <[http://www.if.ufrgs.br/eenci/artigos/Artigo\\_ID71/v4\\_n1\\_a2009.pdf](http://www.if.ufrgs.br/eenci/artigos/Artigo_ID71/v4_n1_a2009.pdf)>. Acessado em: 25 mar. 2015.
- BARNEA, N. ; DORI, Y. J. High school chemistry students performance and gender differences in a computerized molecular modeling learning environment. **Journal of Science Education and Technology**. v. 8, n. 4, p. 257-271, 1999.
- CANTOR, Diego.; JONES, Brandon. **WebGL: Beginner's Guide**. Birmingham: PACKT Publishing, 2014.
- CARVALHO, Maria G. Tecnologia, Desenvolvimento Social e Educação Tecnológica. **Revista Educação & Tecnologia**, Curitiba, n. 1, jul. 1997. Disponível em: <<http://revistas.utfpr.edu.br/pb/index.php/revedutec-ct/article/download/1011/603>>. Acesso em: 28 mar. 2015.
- DANNE, Renis. **BKChem**. [S.l.], 2010. Disponível em: <<http://bkchem.zirael.org/>>. Acesso em: 28 mar. 2015.
- DIRKSEN, Jos. **Learning Three.js: The JavaScript 3D Library for WebGL**. Birmingham: PACKT Publishing, 2013.
- FERREIRA, Vitor F. As tecnologias Interativas no Ensino. **Revista Química Nova**, São Paulo, v.21, n. 6, nov./dez. 1998. Disponível em: <[http://quimicanova.sbq.org.br/audiencia\\_pdf.asp?aid2=2704&nomeArquivo=Vol21No6\\_780\\_v21\\_n6\\_\(18\).pdf](http://quimicanova.sbq.org.br/audiencia_pdf.asp?aid2=2704&nomeArquivo=Vol21No6_780_v21_n6_(18).pdf)>. Acesso em: 28 mar. 2015.
- FHTR. **Introduction to Three.js**. [S.l.], 2012. Disponível em: <<http://fhtr.org/BasicsOfThreeJS/>>. Acesso em: 28 mar. 2015.
- GROSSKURTH, Alan; GODFREY, Michel W. **Architecture and evolution of the modern web browser**. Waterloo, 2006. Disponível em: <<http://grosskurth.ca/papers/browserarchevol-20060619.pdf>>. Acesso em: 27 mar. 2015.
- HANWELL, Marcus D. et al. Avogadro: An advanced semantic chemical editor, visualization, and analysis platform. **J. Cheminformatics**, v. 4, n. 1, p. 17, 2012.
- IUPAC. **Periodic Table of the Elements**. [S.l.], 2013. Disponível em: <[http://www.iupac.org/fileadmin/user\\_upload/news/IUPAC\\_Periodic\\_Table-1May13.pdf](http://www.iupac.org/fileadmin/user_upload/news/IUPAC_Periodic_Table-1May13.pdf)>. Acesso em: 28 mar. 2015.

KRAUSS, José R. **VISEDU-MAT**: Visualizador de material educacional, módulo de matemática. 2013. 64 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

KHRONOS. **WebGL specification**. [S.l.], 2015. Disponível em: <<https://www.khronos.org/registry/webgl/specs/1.0/>>. Acesso em: 25 mar. 2015.

LOBATO, Anderson C. **A abordagem do efeito estufa nos livros de química**: uma análise crítica. 2007. 32 f. Monografia de especialização (Ensino de Ciências por Investigação). Universidade Federal de Minas Gerais, Belo Horizonte.

MEDEIROS, Miguel A. A informática no ensino da química: análise de um software para ensino de Tabela Periódica. In: ENCONTRO NACIONAL DE ENSINO DE QUÍMICA, 14., 2008, Curitiba. **Anais...** Curitiba: Imprensa Universitária da UFPR, 2008. p. 1-10.

MONTIBELER, James P. **VISEDU-CG**: Visualizador de material educacional, módulo de computação gráfica. 2014. 105 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

NUNES, Amisson S.; ARDONI, Dulcélia S. O ensino de química nas escolas da rede pública de ensino fundamental e médio do município de Itapetininga-BA: O olhar dos alunos. In: **Encontro Dialógico Transdisciplinar** – Enditrans, 2010, Vitória da Conquista, BA – Educação e conhecimento científico, 2010.

RUSSELJ, John B. **Química Geral Volume I**. 2. ed. São Paulo: Makron Books, 1994.

SILVA, Airton M. Proposta para Tornar o Ensino de Química mais Atraente. **Revista de - Química Industrial**, Rio de Janeiro, ano 79. n. 731, p. 7-12, 2º trimestre, 2011. Disponível em: <<http://www.abq.org.br/rqi/Edicao-731.html>>. Acesso em 28 mar. 2015.

SOLOMONS, TW Graham; FRYHLE, Craig B. **Química Orgânica, vol. 1**. 7. ed. Tradução Whei Oh Lin. Rio de Janeiro: LTC Editora, 2000.

VICINGUERA, Maria L. F. **O uso do computador auxiliando no ensino da química**. 2002. 97 f. Dissertação (Mestrado em Engenharia de Produção) – Curso de Pós-graduação em Engenharia da Produção, Universidade Federal de Santa Catarina, Florianópolis. Disponível em: <[http://www.educadores.diaadia.pr.gov.br/arquivos/File/2010/artigos\\_teses/quimica/uso\\_comput\\_ens\\_quim\\_dissert.pdf](http://www.educadores.diaadia.pr.gov.br/arquivos/File/2010/artigos_teses/quimica/uso_comput_ens_quim_dissert.pdf)>. Acesso em: 28 mar. 2015.

W3SCHOOLS. **HTML5 Introduction**. [S.l.], 2015. Disponível em: <[http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)>. Acesso em: 28 mar. 2015.

W3C. **HTML5**. [S.l.], 201. Disponível em: <<http://www.w3.org/TR/html5/>>. Acesso em: 28 mar. 2015.

WU, H., KRAJCIK, J. S., SOLOWAY, E. Promoting understanding of chemical representations: Students use of visualization tool in the classroom. **Journal of research in Science Teaching**, v. 38, n. 7, p. 821-842, 2001.

## APÊNDICE A – Detalhamento dos casos de uso do motor

Este apêndice apresenta o detalhamento dos casos de uso do motor utilizado pela aplicação desenvolvida

Quadro 25 – Descrição do caso de uso UC01

UC01 – Informar fórmula química	
Requisitos atendidos	RF01
Pré-Condição	Nenhuma.
Cenário principal	<ol style="list-style-type: none"> <li>1. O Usuário digita uma fórmula química na Área de Entrada;</li> <li>2. O Usuário clica no botão =&gt;;</li> <li>3. O sistema valida a fórmula química;</li> <li>4. O sistema cria a Lista de Elementos químicos apontados na fórmula;</li> </ol>
Exceção	Na etapa três, caso a fórmula química seja inválida, possua algum elemento que não existe ou esteja semanticamente inválida, uma caixa de diálogo com erro será apresentada.
Pós-Condição	O usuário visualiza uma relação de elementos químicos apontados na fórmula na Lista de Elementos.

Quadro 26 – Descrição do caso de uso UC02

UC02 – Selecionar elemento químico	
Requisitos atendidos	RF02
Pré-Condição	UC01
Cenário principal	O Usuário seleciona um elemento químico na Lista de Elementos;
Exceção	Nenhuma.
Pós-Condição	O elemento químico selecionado é disponibilizado para adição na Área de Desenho.

Quadro 27 – Descrição do caso de uso UC03

UC03 – Adicionar elemento químico na área de desenho	
Requisitos atendidos	RF02
Pré-Condição	UC02
Cenário principal	<p>O Usuário adiciona o elemento químico à Área de Desenho;</p> <p>O elemento químico adicionado é desabilitado da Lista de Elementos pelo sistema;</p> <p>O sistema adiciona este elemento químico à fórmula estrutural;</p>
Exceção	Na etapa um caso o Usuário clique fora da Área de Desenho, nada acontecerá. O elemento químico seguirá permanentemente selecionado até que o Usuário clique em um ponto dentro da Área de Desenho.
Pós-Condição	O Usuário visualiza o elemento químico selecionado dentro da Área de Desenho, onde o mesmo pode ser manipulado, alterando suas coordenadas.

Quadro 28 – Descrição do caso de uso UC04

UC04 – Selecionar tipo de ligação química entre átomos	
Requisitos atendidos	RF02
Pré-Condição	UC03
Cenário principal	1. O Usuário seleciona um modo de ligação na Lista de Elementos;
Exceção	Nenhuma.
Pós-Condição	O modo de ligação fica preparado para ser adicionado entre dois átomos.

Quadro 29 – Descrição do caso de uso UC05

UC05 – Adicionar ligação entre átomos	
Requisitos atendidos	RF02
Pré-Condição	UC04
Cenário principal	<ol style="list-style-type: none"> <li>1. O Usuário seleciona um átomo da Área de Desenho;</li> <li>2. O sistema classifica este átomo como ponto de origem da ligação química.</li> <li>3. O Usuário seleciona outro átomo da Área de Desenho.</li> <li>4. O sistema adiciona este segundo átomo como ponto de destino da ligação química.</li> <li>5. O sistema desenha a ligação entre os dois átomos;</li> </ol>
Exceção	<ol style="list-style-type: none"> <li>1. Na etapa dois caso o Usuário clique fora da Área de Desenho ou não selecione nenhum átomo, o sistema não classificará qualquer átomo como ponto de origem da ligação.</li> <li>2. Na etapa três caso o Usuário clique fora da Área de Desenho, não selecione um átomo ou selecione o átomo que já está marcado como ponto de origem da ligação, o sistema não irá executar nenhuma ação.</li> </ol>
Pós-Condição	O usuário visualiza a ligação entre os dois átomos selecionados.

Quadro 30 – Descrição do caso de uso UC06

UC06 – Geração da fórmula estrutural tridimensional	
Requisitos atendidos	RF03, RF04
Pré-Condição	UC05
Cenário principal	<ol style="list-style-type: none"> <li>1. O Usuário clica no botão com símbolo de engrenagem;</li> <li>2. O sistema verifica se todos os elementos químicos da Lista de Elementos foram utilizados na formula estrutural bidimensional;</li> <li>3. O sistema valida se todos os átomos possuem ao menos uma ligação química;</li> <li>4. O sistema identifica os tipos de ligações existentes entre os átomos;</li> <li>5. O sistema valida a fórmula estrutural de acordo com a regra do octeto;</li> <li>6. O sistema estabelece o tipo de geometria da molécula.</li> </ol>
Exceção	<ol style="list-style-type: none"> <li>1. Na etapa dois, caso o Usuário deixe de utilizar algum elemento químico disponível na Lista de Elementos, uma caixa de diálogo informando o erro será apresentada pelo sistema;</li> <li>2. Na etapa três, caso algum átomo não pertença a pelo menos uma ligação química, uma caixa de diálogo informando o erro será apresentada pelo sistema;</li> <li>3. Na etapa quatro, caso o tipo de ligação seja considerada inválida, ou se a ligação for classificada como metálica, uma caixa de diálogo informando o erro será apresentada pelo sistema;</li> <li>4. Na etapa cinco caso a fórmula não esteja de acordo com a regra do octeto, o sistema apresenta uma caixa de diálogo informando o erro.</li> <li>5. Na etapa seis, caso o tipo de geometria não seja identificado, o sistema apresenta caixa de diálogo informando o erro.</li> </ol>
Pós-Condição	O Usuário visualiza a fórmula estrutural tridimensional na Área de Visualização.

Quadro 31 – Interação com o ambiente tridimensional

UC07 – Interação com o ambiente tridimensional	
Requisitos atendidos	RF05
Pré-Condição	UC06
Cenário principal	<ol style="list-style-type: none"> <li>1. O Usuário pode aplicar o efeito de zoom com o <i>scroll</i> do mouse dentro da Área de Visualização;</li> <li>2. Ao clicar e arrastar o ponteiro do mouse o Usuário pode mudar a posição da molécula dentro da Área de Visualização.</li> </ol>
Exceção	Nenhuma.
Pós-Condição	O Usuário visualiza as mudanças da câmera na Área de Visualização.

## APÊNDICE B – Formulários sobre a experiência de utilização do VisEdu-Química

Nas figuras 24 a 29 a seguir, são apresentados alguns dos formulários preenchidos pelos alunos entrevistados com suas considerações sobre o VisEdu-Química.

Figura 24 – Formulário respondido por uma dos estudantes entrevistados

**Questionário de avaliação**

- 1) Você considera que ferramentas interativas podem auxiliar no processo de educação?  
 Sim  
 Parcialmente  
 Não
- 2) No cenário atual de ensino, é viável para instituições de ensino disponibilizarem notebooks para grupos de alunos?  
 Sim  
 Não
- 3) Você considera a aplicação apresentada como algo que poderia auxiliar a fixação do conteúdo?  
 Sim  
 Não
- 4) Como você classifica a usabilidade da aplicação?  
 Ruim  
 Regular  
 Boa  
 Ótima
- 5) Com base nas respostas anteriores, quais foram os "pontos positivos" encontrados na aplicação?  

Ver os ângulos e geometrias corretamente  
Mostra o que já foi usado
- 6) Com base nas respostas anteriores, quais foram os "pontos negativos" encontrados na aplicação?  

Complicações na tentativa de ver moléculas maiores.
- 7) Com base nas repostas anteriores, quais seriam suas sugestões para melhorar o aplicativo?  

Visualização ~~em~~ 3D melhorada  
Botão de ligação deveria ficar selecionado até terminarem todas as ligações necessárias

Figura 25 – Formulário respondido por uma dos estudantes entrevistados

5  
7

### Questionário de avaliação

- 1) Você considera que ferramentas interativas podem auxiliar no processo de educação?
  - Sim
  - Parcialmente
  - Não
- 2) No cenário atual de ensino, é viável para instituições de ensino disponibilizarem notebooks para grupos de alunos?
  - Sim
  - Não
- 3) Você considera a aplicação apresentada como algo que poderia auxiliar a fixação do conteúdo?
  - Sim
  - Não
- 4) Como você classifica a usabilidade da aplicação?
  - Ruim
  - Regular
  - Boa
  - Ótima
- 5) Com base nas respostas anteriores, quais foram os "pontos positivos" encontrados na aplicação?
 

*A simplicidade e objetividade da interface possibilita uma ótima experiência.*
- 6) Com base nas respostas anteriores, quais foram os "pontos negativos" encontrados na aplicação?
 

*O aplicativo é muito básico, serve apenas para uma função.*
- 7) Com base nas respostas anteriores, quais seriam suas sugestões para melhorar o aplicativo?
 

*Disponibilização gratuita quando o aplicativo estiver pronto, representação de fórmulas complexas*

Figura 26 – Formulário respondido por uma dos estudantes entrevistados

7

## Questionário de avaliação

- 1) Você considera que ferramentas interativas podem auxiliar no processo de educação?
  - Sim
  - Parcialmente
  - Não
- 2) No cenário atual de ensino, é viável para instituições de ensino disponibilizarem notebooks para grupos de alunos?
  - Sim
  - Não
- 3) Você considera a aplicação apresentada como algo que poderia auxiliar a fixação do conteúdo?
  - Sim
  - Não
- 4) Como você classifica a usabilidade da aplicação?
  - Ruim
  - Regular
  - Boa
  - Ótima
- 5) Com base nas respostas anteriores, quais foram os “pontos positivos” encontrados na aplicação?
 

O aplicativo tem um uso ótimo para questões de trabalhos e exercícios e para melhor compreensão dos formatos geométricos
- 6) Com base nas respostas anteriores, quais foram os “pontos negativos” encontrados na aplicação?
 

ficar botando cada molécula e ~~ligar~~ ligando torna-se cansativo ao formar uma molécula grande e ~~algumas~~ algumas moléculas precisam de um certo conhecimento
- 7) Com base nas respostas anteriores, quais seriam suas sugestões para melhorar o aplicativo?
 

Botões Botões de um jeito mais acessível para público do ensino-médio, facilitar a criação, não ficar tão cansativo, e uma customização das cores e tabelas assim com a tabela para uso mais fácil

Figura 27– Formulário respondido por uma dos estudantes entrevistados

6

### Questionário de avaliação

- 1) Você considera que ferramentas interativas podem auxiliar no processo de educação?  
 Sim  
 Parcialmente  
 Não
- 2) No cenário atual de ensino, é viável para instituições de ensino disponibilizarem notebooks para grupos de alunos?  
 Sim  
 Não
- 3) Você considera a aplicação apresentada como algo que poderia auxiliar a fixação do conteúdo?  
 Sim  
 Não
- 4) Como você classifica a usabilidade da aplicação?  
 Ruim  
 Regular  
 Boa  
 Ótima
- 5) Com base nas respostas anteriores, quais foram os “pontos positivos” encontrados na aplicação?  

Para ajudar a entender a estrutura molecular.
- 6) Com base nas respostas anteriores, quais foram os “pontos negativos” encontrados na aplicação?  

Os átomos poderiam ser ligados automaticamente.
- 7) Com base nas repostas anteriores, quais seriam suas sugestões para melhorar o aplicativo?  

Poderia fazer funcionar para celular e tablet!

Figura 28– Formulário respondido por uma dos estudantes entrevistados

7

### Questionário de avaliação

- 1) Você considera que ferramentas interativas podem auxiliar no processo de educação?  
 Sim  
 Parcialmente  
 Não
- 2) No cenário atual de ensino, é viável para instituições de ensino disponibilizarem notebooks para grupos de alunos?  
 Sim  
 Não
- 3) Você considera a aplicação apresentada como algo que poderia auxiliar a fixação do conteúdo?  
 Sim  
 Não
- 4) Como você classifica a usabilidade da aplicação?  
 Ruim  
 Regular  
 Boa  
 Ótima
- 5) Com base nas respostas anteriores, quais foram os “pontos positivos” encontrados na aplicação?  

*Quero de não termos tido muito tempo para testar o programa, é normal presumir que teremos uma ~~uma~~ visualização mais intuitiva e prática da química.*
- 6) Com base nas respostas anteriores, quais foram os “pontos negativos” encontrados na aplicação?  

*Não vi o suficiente para poder apontar algum*
- 7) Com base nas repostas anteriores, quais seriam suas sugestões para melhorar o aplicativo?  

*Não vi o suficiente para dar muitas sugestões, mas tentem investir em realidade virtual, seria fantástica*

Figura 29– Formulário respondido por uma dos estudantes entrevistados

5

### Questionário de avaliação

- 1) Você considera que ferramentas interativas podem auxiliar no processo de educação?
  - Sim
  - Parcialmente
  - Não
- 2) No cenário atual de ensino, é viável para instituições de ensino disponibilizarem notebooks para grupos de alunos?
  - Sim
  - Não
- 3) Você considera a aplicação apresentada como algo que poderia auxiliar a fixação do conteúdo?
  - Sim
  - Não
- 4) Como você classifica a usabilidade da aplicação?
  - Ruim
  - Regular
  - Boa
  - Ótima
- 5) Com base nas respostas anteriores, quais foram os "pontos positivos" encontrados na aplicação?
 

Além do fato de ele corrigir quando erra a imagem em 3D, ele ajuda a desenvolver uma visão geométrica e facilita para aqueles que não possuem a mesma.
- 6) Com base nas respostas anteriores, quais foram os "pontos negativos" encontrados na aplicação?
 

A falta de uma explicação da imagem 3D formada (por ex: pirâmidal, tetraédrica e a demonstração, pois é necessário ir ligando e colorindo as moléculas (por 1, 2, 3) que irá lá e selecionar a ligação novamente.
- 7) Com base nas respostas anteriores, quais seriam suas sugestões para melhorar o aplicativo?
 

Uma legenda emborixo da imagem 3D seria ótima, facilitaria para aqueles que não lembram das formas.

