

José  
Maurício  
Schneedorf  
Ferreira  
da Silva

# VIVIFICANDO CONTEÚDOS PARA O ENSINO BÁSICO

Moléculas, Gráficos,  
e Mapas Interativos



José  
Maurício  
Schneedorf  
Ferreira  
da Silva

# VIVIFICANDO CONTEÚDOS PARA O ENSINO BÁSICO

Moléculas, Gráficos,  
e Mapas Interativos



| São Paulo | 2025 |



DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)

S586c

Silva, José Maurício Schneedorf Ferreira da -  
Vivificando Conteúdos Para o Ensino Básico: Moléculas,  
Gráficos, e Mapas Interativos / José Maurício Schneedorf  
Ferreira da Silva. – São Paulo: Pimenta Cultural, 2025.

Livro em PDF

ISBN 978-85-7221-394-3

DOI 10.31560/pimentacultural/978-85-7221-394-3

1. Ensino reproduzível. 2. Objetos interativos educacionais.  
3. Pensamento computacional. 4. Metodologias ativas.  
5. Ensino básico. I. Silva, José Maurício Schneedorf Ferreira da.  
II. Título.

CDD 371.3

Índice para catálogo sistemático:

I. Educação - Metodologias ativas

Simone Sales • Bibliotecária • CRB: ES-000814/0

Copyright © Pimenta Cultural, alguns direitos reservados.

Copyright do texto © 2025 o autor.

Copyright da edição © 2025 Pimenta Cultural.

Esta obra é licenciada por uma Licença Creative Commons:  
*Atribuição-NãoComercial-SemDerivações 4.0 Internacional - (CC BY-NC-ND 4.0).*  
Os termos desta licença estão disponíveis em:  
<<https://creativecommons.org/licenses/>>.  
Direitos para esta edição cedidos à Pimenta Cultural.  
O conteúdo publicado não representa a posição oficial da Pimenta Cultural.

---

Direção editorial	Patricia Biegging Raul Inácio Busarello
Editora executiva	Patricia Biegging
Gerente editorial	Landressa Rita Schiefelbein
Assistente editorial	Júlia Marra Torres
Estagiária editorial	Ana Flávia Pivisan Kobata
Diretor de criação	Raul Inácio Busarello
Assistente de arte	Naiara Von Groll
Editoração eletrônica	Andressa Karina Voltolini
Estagiárias em editoração	Raquel de Paula Miranda Stela Tiemi Hashimoto Kanada
Imagens da capa	user1299111, pikisuperstar - Freepik.com; AMI COMUNICAÇÃO LTDA.- ME (Pabline Cota Felix
Tipografias	Acumin, Arial
Revisão	Patricia Biegging
Autor	José Maurício Schneedorf Ferreira da Silva

---

**PIMENTA CULTURAL**  
São Paulo • SP  
+55 (11) 96766 2200  
[livro@pimentacultural.com](mailto:livro@pimentacultural.com)  
[www.pimentacultural.com](http://www.pimentacultural.com)





# CONSELHO EDITORIAL CIENTÍFICO

## Doutores e Doutororas

**Adilson Cristiano Habowski**

*Universidade La Salle, Brasil*

**Adriana Flávia Neu**

*Universidade Federal de Santa Maria, Brasil*

**Adriana Regina Vettorazzi Schmitt**

*Instituto Federal de Santa Catarina, Brasil*

**Aguimario Pimentel Silva**

*Instituto Federal de Alagoas, Brasil*

**Alaím Passos Bispo**

*Universidade Federal do Rio Grande do Norte, Brasil*

**Alaím Souza Neto**

*Universidade Federal de Santa Catarina, Brasil*

**Alessandra Knoll**

*Universidade Federal de Santa Catarina, Brasil*

**Alessandra Regina Müller Germani**

*Universidade Federal de Santa Maria, Brasil*

**Aline Corso**

*Universidade do Vale do Rio dos Sinos, Brasil*

**Aline Wendpap Nunes de Siqueira**

*Universidade Federal de Mato Grosso, Brasil*

**Ana Rosangela Colares Lavand**

*Universidade Estadual do Norte do Paraná, Brasil*

**André Gobbo**

*Universidade Federal da Paraíba, Brasil*

**André Tanus Cesário de Souza**

*Faculdade Anhanguera, Brasil*

**Andressa Antunes**

*Universidade Federal de Ouro Preto, Brasil*

**Andressa Wiebusch**

*Universidade Federal de Santa Maria, Brasil*

**Andreza Regina Lopes da Silva**

*Universidade Federal de Santa Catarina, Brasil*

**Angela Maria Farah**

*Universidade de São Paulo, Brasil*

**Anísio Batista Pereira**

*Universidade do Estado do Amapá, Brasil*

**Antonio Edson Alves da Silva**

*Universidade Estadual do Ceará, Brasil*

**Antonio Henrique Coutelo de Moraes**

*Universidade Federal de Rondonópolis, Brasil*

**Arthur Vianna Ferreira**

*Universidade do Estado do Rio de Janeiro, Brasil*

**Ary Albuquerque Cavalcanti Junior**

*Universidade Federal de Mato Grosso, Brasil*

**Asterlindo Bandeira de Oliveira Júnior**

*Universidade Federal da Bahia, Brasil*

**Bárbara Amaral da Silva**

*Universidade Federal de Minas Gerais, Brasil*

**Bernadette Beber**

*Universidade Federal de Santa Catarina, Brasil*

**Bruna Carolina de Lima Siqueira dos Santos**

*Universidade do Vale do Itajaí, Brasil*

**Bruno Rafael Silva Nogueira Barbosa**

*Universidade Federal da Paraíba, Brasil*

**Caio Cesar Portella Santos**

*Instituto Municipal de Ensino Superior de São Manuel, Brasil*

**Carla Wanessa do Amaral Caffagni**

*Universidade de São Paulo, Brasil*

**Carlos Adriano Martins**

*Universidade Cruzeiro do Sul, Brasil*

**Carlos Jordan Lapa Alves**

*Universidade Estadual do Norte Fluminense Darcy Ribeiro, Brasil*

**Caroline Chioquetta Lorenset**

*Universidade Federal de Santa Catarina, Brasil*

**Cassia Cordeiro Furtado**

*Universidade Federal do Maranhão, Brasil*

**Cássio Michel dos Santos Camargo**

*Universidade Federal do Rio Grande do Sul, Brasil*

**Cecilia Machado Henriques**

*Universidade Federal de Santa Catarina, Brasil*

**Christiano Martino Otero Avila**

*Universidade Federal de Pelotas, Brasil*

**Cláudia Samuel Kessler**

*Universidade Federal do Rio Grande do Sul, Brasil*

**Cristiana Barcelos da Silva**

*Universidade do Estado de Minas Gerais, Brasil*

**Cristiane Silva Fontes**

*Universidade Federal de Minas Gerais, Brasil*

**Daniela Susana Segre Guertzenstein**

*Universidade de São Paulo, Brasil*

**Daniele Cristine Rodrigues**

*Universidade de São Paulo, Brasil*

**Dayse Centurion da Silva**

*Universidade Anhanguera, Brasil*

**Dayse Sampaio Lopes Borges**

*Universidade Estadual do Norte Fluminense Darcy Ribeiro, Brasil*

**Deilson do Carmo Trindade**

*Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, Brasil*

**Diego Pizarro**

*Instituto Federal de Brasília, Brasil*

**Dorama de Miranda Carvalho**

*Escola Superior de Propaganda e Marketing, Brasil*

**Edilson de Araújo dos Santos**

*Universidade de São Paulo, Brasil*

**Edson da Silva**

*Universidade Federal dos Vales do Jequitinhonha e Mucuri, Brasil*

**Elena Maria Mallmann**  
*Universidade Federal de Santa Maria, Brasil*

**Eleonora das Neves Simões**  
*Universidade Federal do Rio Grande do Sul, Brasil*

**Eliane Silva Souza**  
*Universidade do Estado da Bahia, Brasil*

**Elvira Rodrigues de Santana**  
*Universidade Federal da Bahia, Brasil*

**Estevão Schultz Campos**  
*Centro Universitário Adventista de São Paulo, Brasil*

**Éverly Pegoraro**  
*Universidade Federal do Rio de Janeiro, Brasil*

**Fábio Santos de Andrade**  
*Universidade Federal de Mato Grosso, Brasil*

**Fabricia Lopes Pinheiro**  
*Universidade Federal do Estado do Rio de Janeiro, Brasil*

**Fauston Negreiros**  
*Univrsidade de Brasília, Brasil*

**Felipe Henrique Monteiro Oliveira**  
*Universidade Federal da Bahia, Brasil*

**Fernando Vieira da Cruz**  
*Universidade Estadual de Campinas, Brasil*

**Flávia Fernanda Santos Silva**  
*Universidade Federal do Amazonas, Brasil*

**Gabriela Moysés Pereira**  
*Universidade Federal do Rio de Janeiro, Brasil*

**Gabriella Eldereti Machado**  
*Universidade Federal de Santa Maria, Brasil*

**Germano Ehlerth Pollnow**  
*Universidade Federal de Pelotas, Brasil*

**Geuciane Felipe Guerim Fernandes**  
*Universidade Federal do Pará, Brasil*

**Geymeesson Brito da Silva**  
*Universidade Federal de Pernambuco, Brasil*

**Giovanna Ofretorio de Oliveira Martin Franchi**  
*Universidade Federal de Santa Catarina, Brasil*

**Handherson Leylton Costa Damasceno**  
*Universidade Federal da Bahia, Brasil*

**Hebert Elias Lobo Sosa**  
*Universidad de Los Andes, Venezuela*

**Helciclever Barros da Silva Sales**  
*Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira, Brasil*

**Helena Azevedo Paulo de Almeida**  
*Universidade Federal de Ouro Preto, Brasil*

**Hendy Barbosa Santos**  
*Faculdade de Artes do Paraná, Brasil*

**Humberto Costa**  
*Universidade Federal do Paraná, Brasil*

**Igor Alexandre Barcelos Graciano Borges**  
*Universidade de Brasília, Brasil*

**Inara Antunes Vieira Willerding**  
*Universidade Federal de Santa Catarina, Brasil*

**Jaziel Vasconcelos Dorneles**  
*Universidade de Coimbra, Portugal*

**Jean Carlos Gonçalves**  
*Universidade Federal do Paraná, Brasil*

**Joao Adalberto Campato Junior**  
*Universidade Brasil, Brasil*

**Jocimara Rodrigues de Sousa**  
*Universidade de São Paulo, Brasil*

**Joelson Alves Onofre**  
*Universidade Estadual de Santa Cruz, Brasil*

**Jónata Ferreira de Moura**  
*Universidade São Francisco, Brasil*

**Jonathan Machado Domingues**  
*Universidade Federal de São Paulo, Brasil*

**Jorge Eschriqui Vieira Pinto**  
*Universidade Estadual Paulista Júlio de Mesquita Filho, Brasil*

**Jorge Luís de Oliveira Pinto Filho**  
*Universidade Federal do Rio Grande do Norte, Brasil*

**Juliana de Oliveira Vicentini**  
*Universidade de São Paulo, Brasil*

**Juliano Milton Kruger**  
*Instituto Federal do Amazonas, Brasil*

**Juliano Pizzano Ayoub**  
*Universidade Estadual de Ponta Grossa, Brasil*

**Julierme Sebastião Morais Souza**  
*Universidade Federal de Uberlândia, Brasil*

**Junior César Ferreira de Castro**  
*Universidade de Brasília, Brasil*

**Katia Bruginski Mulik**  
*Universidade de São Paulo, Brasil*

**Laionel Vieira da Silva**  
*Universidade Federal da Paraíba, Brasil*

**Lauro Sérgio Machado Pereira**  
*Instituto Federal do Norte de Minas Gerais, Brasil*

**Leonardo Freire Marino**  
*Universidade do Estado do Rio de Janeiro, Brasil*

**Leonardo Pinheiro Mozdzenski**  
*Universidade Federal de Pernambuco, Brasil*

**Letícia Cristina Alcântara Rodrigues**  
*Faculdade de Artes do Paraná, Brasil*

**Lucila Romano Tragtenberg**  
*Pontifícia Universidade Católica de São Paulo, Brasil*

**Lucimara Rett**  
*Universidade Metodista de São Paulo, Brasil*

**Luiz Eduardo Neves dos Santos**  
*Universidade Federal do Maranhão, Brasil*

**Maikel Pons Giralt**  
*Universidade de Santa Cruz do Sul, Brasil*

**Manoel Augusto Polastrelli Barbosa**  
*Universidade Federal do Espírito Santo, Brasil*

**Marcelo Nicomedes dos Reis Silva Filho**  
*Universidade Estadual do Oeste do Paraná, Brasil*

**Márcia Alves da Silva**

*Universidade Federal de Pelotas, Brasil*

**Marcio Bernardino Sirino**

*Universidade Federal do Estado do Rio de Janeiro, Brasil*

**Marcos Pereira dos Santos**

*Universidad Internacional Iberoamericana del Mexico, México*

**Marcos Uzel Pereira da Silva**

*Universidade Federal da Bahia, Brasil*

**Marcus Fernando da Silva Praxedes**

*Universidade Federal do Recôncavo da Bahia, Brasil*

**Maria Aparecida da Silva Santandel**

*Universidade Federal de Mato Grosso do Sul, Brasil*

**Maria Cristina Giorgi**

*Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Brasil*

**Maria Edith Maroca de Avelar**

*Universidade Federal de Ouro Preto, Brasil*

**Marina Bezerra da Silva**

*Instituto Federal do Piauí, Brasil*

**Marines Rute de Oliveira**

*Universidade Estadual do Oeste do Paraná, Brasil*

**Maurício José de Souza Neto**

*Universidade Federal da Bahia, Brasil*

**Maurício José de Souza Neto**

*Universidade Federal da Bahia, Brasil*

**Michele Marcelo Silva Bortolai**

*Universidade de São Paulo, Brasil*

**Mônica Tavares Orsini**

*Universidade Federal do Rio de Janeiro, Brasil*

**Nara Oliveira Salles**

*Universidade do Estado do Rio de Janeiro, Brasil*

**Neide Araujo Castilho Teno**

*Universidade Estadual de Mato Grosso do Sul, Brasil*

**Neli Maria Mengalli**

*Pontifícia Universidade Católica de São Paulo, Brasil*

**Patricia Bieging**

*Universidade de São Paulo, Brasil*

**Patrícia Flavia Mota**

*Universidade do Estado do Rio de Janeiro, Brasil*

**Patrícia Helena dos Santos Carneiro**

*Universidade Federal de Rondônia, Brasil*

**Rainei Rodrigues Jadejiski**

*Universidade Federal do Espírito Santo, Brasil*

**Raul Inácio Busarello**

*Universidade Federal de Santa Catarina, Brasil*

**Raymundo Carlos Machado Ferreira Filho**

*Universidade Federal do Rio Grande do Sul, Brasil*

**Ricardo Luiz de Bittencourt**

*Universidade do Extremo Sul Catarinense, Brasil*

**Roberta Rodrigues Ponciano**

*Universidade Federal de Uberlândia, Brasil*

**Robson Teles Gomes**

*Universidade Católica de Pernambuco, Brasil*

**Rodiney Marcelo Braga dos Santos**

*Universidade Federal de Roraima, Brasil*

**Rodrigo Amancio de Assis**

*Universidade Federal de Mato Grosso, Brasil*

**Rodrigo Sarruge Molina**

*Universidade Federal do Espírito Santo, Brasil*

**Rogério Rauber**

*Universidade Estadual Paulista Júlio de Mesquita Filho, Brasil*

**Rosane de Fatima Antunes Obregon**

*Universidade Federal do Maranhão, Brasil*

**Samuel André Pompeo**

*Universidade Estadual Paulista Júlio de Mesquita Filho, Brasil*

**Sebastião Silva Soares**

*Universidade Federal do Tocantins, Brasil*

**Silmar José Spinardi Franchi**

*Universidade Federal de Santa Catarina, Brasil*

**Simone Alves de Carvalho**

*Universidade de São Paulo, Brasil*

**Simoni Urnau Bonfiglio**

*Universidade Federal da Paraíba, Brasil*

**Stela Maris Vaucher Farias**

*Universidade Federal do Rio Grande do Sul, Brasil*

**Tadeu João Ribeiro Baptista**

*Universidade Federal do Rio Grande do Norte, Brasil*

**Taiane Aparecida Ribeiro Nepomoceno**

*Universidade Estadual do Oeste do Paraná, Brasil*

**Taíza da Silva Gama**

*Universidade de São Paulo, Brasil*

**Tania Micheline Miorando**

*Universidade Federal de Santa Maria, Brasil*

**Tarcísio Vanzin**

*Universidade Federal de Santa Catarina, Brasil*

**Tascieli Feltrin**

*Universidade Federal de Santa Maria, Brasil*

**Tatiana da Costa Jansen**

*Serviço Nacional de Aprendizagem Comercial, Brasil*

**Tayson Ribeiro Teles**

*Universidade Federal do Acre, Brasil*

**Thiago Barbosa Soares**

*Universidade Federal do Tocantins, Brasil*

**Thiago Camargo Iwamoto**

*Universidade Estadual de Goiás, Brasil*

**Thiago Medeiros Barros**

*Universidade Federal do Rio Grande do Norte, Brasil*

**Tiago Mendes de Oliveira**

*Universidade Federal do Rio Grande do Norte, Brasil*

**Vanessa de Sales Marruche**

*Universidade Federal do Amazonas, Brasil*

**Vanessa Elisabete Raue Rodrigues**

*Universidade Estadual do Centro Oeste, Brasil*

**Vania Ribas Ulbricht**

*Universidade Federal de Santa Catarina, Brasil*

**Vinicius da Silva Freitas**  
*Centro Universitário Vale do Cricaré, Brasil*

**Wellington Furtado Ramos**  
*Universidade Federal de Mato Grosso do Sul, Brasil*

**Wellton da Silva de Fatima**  
*Instituto Federal de Alagoas, Brasil*

**Wenis Vargas de Carvalho**  
*Universidade Federal da Grande Dourados, Brasil*

**Yan Masetto Nicolai**  
*Universidade Federal de São Carlos, Brasil*

## PARECERISTAS E REVISORES(AS) POR PARES

### Avaliadores e avaliadoras Ad-Hoc

**Alcidinei Dias Alves**  
*Logos University International, Estados Unidos*

**Alessandra Figueiró Thornton**  
*Universidade Luterana do Brasil, Brasil*

**Alexandre João Appio**  
*Universidade do Vale do Rio dos Sinos, Brasil*

**Artur Pires de Camargos Júnior**  
*Universidade do Vale do Sapucaí, Brasil*

**Bianka de Abreu Severo**  
*Universidade Federal de Santa Maria, Brasil*

**Carlos Eduardo B. Alves**  
*Universidade Federal do Agreste de Pernambuco, Brasil*

**Carlos Eduardo Damian Leite**  
*Universidade de São Paulo, Brasil*

**Catarina Prestes de Carvalho**  
*Instituto Federal Sul-Rio-Grandense, Brasil*

**Davi Fernandes Costa**  
*Secretaria da Educação do Estado de São Paulo, Brasil*

**Denilson Marques dos Santos**  
*Universidade do Estado do Pará, Brasil*

**Domingos Aparecido dos Reis**  
*Must University, Estados Unidos*

**Edson Vieira da Silva de Camargos**  
*Logos University International, Estados Unidos*

**Edwins de Moura Ramires**  
*Serviço Nacional de Aprendizagem Comercial, Brasil*

**Elisiene Borges Leal**  
*Universidade Federal do Piauí, Brasil*

**Elizabeth de Paula Pacheco**  
*Universidade Federal de Uberlândia, Brasil*

**Elton Simomukay**  
*Universidade Estadual de Ponta Grossa, Brasil*

**Francisco Geová Goveia Silva Júnior**  
*Universidade Potiguar, Brasil*

**Indiamaris Pereira**  
*Universidade do Vale do Itajaí, Brasil*

**Jacqueline de Castro Rimá**  
*Universidade Federal da Paraíba, Brasil*

**Jonas Lacchini**  
*Pontifícia Universidade Católica de Minas Gerais, Brasil*

**Lucimar Romeu Fernandes**  
*Instituto Politécnico de Bragança, Brasil*

**Marcos de Souza Machado**  
*Universidade Federal da Bahia, Brasil*

**Michele de Oliveira Sampaio**  
*Universidade Federal do Espírito Santo, Brasil*

**Nívea Consuêlo Carvalho dos Santos**  
*Serviço Nacional de Aprendizagem Comercial, Brasil*

**Pedro Augusto Paula do Carmo**  
*Universidade Paulista, Brasil*

**Rayner do Nascimento Souza**  
*Serviço Nacional de Aprendizagem Comercial, Brasil*

**Samara Castro da Silva**  
*Universidade de Caxias do Sul, Brasil*

**Sidney Pereira Da Silva**  
*Stockholm University, Suécia*

**Suélen Rodrigues de Freitas Costa**  
*Universidade Federal do Espírito Santo, Brasil*

**Thais Karina Souza do Nascimento**  
*Instituto de Ciências das Artes, Brasil*

**Viviane Gil da Silva Oliveira**  
*Universidade Federal do Amazonas, Brasil*

**Walmir Fernandes Pereira**  
*Miami University of Science and Technology, Estados Unidos*

**Weyber Rodrigues de Souza**  
*Pontifícia Universidade Católica de Goiás, Brasil*

**William Roslindo Paranhos**  
*Universidade Federal de Santa Catarina, Brasil*

#### Parecer e revisão por pares

Os textos que compõem esta obra foram submetidos para avaliação do Conselho Editorial da Pimenta Cultural, bem como revisados por pares, sendo indicados para a publicação.



## PREFÁCIO

O *Ensino Reprodutível* (ER) define uma metodologia ativa em ensino e aprendizagem que combina linguagem de programação com conteúdos curriculares.

Em uma frase: **códigos para conteúdos.**

Na prática, o ER envolve aplicar códigos escritos em texto para que um programa gere algum *produto de interesse ao Ensino*, como um gráfico, uma tabela, um texto em vários formatos (pdf, docx, epub), uma simulação, uma animação, e objetos interativos. Dessa forma, é possível imaginar que um educando possa repetir a execução do código em um programa específico produzindo um objeto didático pretendido. E, mais interessante ainda, alterar algum trecho do código para observar um resultado diferente, ou até mesmo criar um novo código buscando outro resultado final.

O *Ensino Reprodutível* deve ser praticado de forma aberta, envolver programas ou aplicativos de distribuição livre e, preferencialmente, por uso de um simples navegador de internet, para possibilitar seu uso pelo maior número de pessoas, sem restrições de configuração de máquinas, e permitindo acesso tanto por computador de mesa e notebooks como por dispositivos móveis, como *tablets* e *smartphones*.

### ERÆÆ...

O acrônimo ERÆÆ sugere as iniciais para *Ensino Reprodutível*, com disposição das fontes em palíndromo, e com a clara intenção de sugerir o exercício ao *erro libertário*. Além disso, trata-se de um arranjo lúdico à analogia dos princípios de desenvolvimento responsável e sustentável dos 3 Rs. Aqui, por tratar-se de códigos para conteúdos em ER, com o significado de *reutilizar* (cópia), *reciclar* (alteração), e *reinventar* (recriação).



## ESTE MATERIAL

Esta obra pretende estimular de modo germinal para a reprodução, modificação e criação de objetos didáticos interativos voltados ao ensino básico, valendo-se de linguagem de programação, e sem a necessidade de conhecimento próprio desta!

Dessa forma, segue um aprendizado suave sobre os tópicos contidos, fartamente ilustrados com situações reais de livros-texto, e com códigos e produtos finais disponíveis de forma irrestrita, tanto de forma estática contida neste material, como de forma interativa junto ao website Bioquanti. O site foi idealizado para contemplar objetos interativos ao *Ensino Reprodutível*, bem como materiais didáticos diversos, ao ensino superior e básico.

Neste material, optou-se por utilizar como referência as ilustrações contidas nos cadernos do Material de Apoio Pedagógico para Aprendizagens – MAPA de Minas Gerais, em sua edição 2024.

A ideia é oferecer ferramentas que possibilitem agregar valor à imagens de moléculas, gráficos e mapas observados nos livros textos e na internet, mas com *interatividade e animação*. Essa *interatividade* permitirá que você recrie e altere figuras que se apresentam estáticas em livros-texto para um **objeto didático mais vivo**. Daí a escolha do termo “vivificando”, que etiqueta o título deste material.

Os códigos e seus produtos interativos contidos neste texto são melhor observados no website de origem em que se encontram, Bioquanti, uma iniciativa seminal para *Ensino Reprodutível* em solo pátrio!

O website contém textos, *ebooks*, objetos interativos, e uma gama diversificada de materiais para o estudo de temas voltados ao **Ensino Reprodutível**. Entre os materiais de interesse plausível ao nível básico de ensino e aprendizagem, a reprodução de todos os materiais contidos neste texto e seus códigos para compilação de alvos interativos, *ebooks* correlatos, um **Jardim de Moléculas** para visualização e estudo de modelos atômicos em 3D, e diversos **objetos interativos em Ensino Reprodutível (OIER)**, os dois últimos dedicados ao ensino básico.

## DIVIRTA-SE!

Ainda que visualizadores moleculares e programas para gráficos interativos com linguagens de programação possam parecer chatos à primeira vista, o material contido neste texto pretende envolver uma curva suave e divertida para o aprendizado dos programas e sua aplicação direta no ensino e aprendizagem de conteúdos curriculares do ensino básico.

Enfim, desejo que você possa aproveitar o potencial técnico, científico, e extremamente lúdico contido aqui, para *moléculas voadoras*, *gráficos* e *mapas interativos*, bem como *animações*, propiciando um jeito diferente, divertido, e necessário, para ensinar e aprender!

**José Maurício**

Alfenas, 17 de abril de 2025.

# SUMÁRIO

## CAPÍTULO 1

<b>Por Quê “Vivificando”?</b> .....	<b>17</b>
1.1 As Ferramentas .....	19
1.2 Um Pouco Mais Sobre as Ferramentas .....	20
1.2.1 Visualização de Moléculas em 3D e Animações - <i>Jmol</i> .....	20
1.2.2 Gráficos Animados, Simulações e Visualização Cartográfica .....	21
1.3 Algumas Políticas Públicas que vão de Encontro a este Material .....	22

## ***JMOL:***

UM VISUALIZADOR INTERATIVO PARA MOLÉCULAS EM 3D .....	<b>24</b>
---	-----------

## CAPÍTULO 2

<b>Usando O <i>Jmol</i> para Observar Moléculas em 3D</b> .....	<b>25</b>
2.1 Onde Começar? .....	26
2.2 Como Carregar uma Molécula <i>Online</i> ? .....	27
2.3 Como Carregar uma Molécula <i>Online</i> , mas em 2D .....	28

## CAPÍTULO 3

<b>Cliques de Mouse <i>versus</i> Texto de Comando</b> .....	<b>29</b>
3.1 Cliques de <i>Mouse</i> .....	30
3.2 Linhas de Comando .....	30

3.3 Cliques de Mouse <i>versus</i> Linhas de Comando .....	31
3.4 Vantagens do Uso de Linhas de Comando sobre o uso de Cliques de <i>Mouse</i> .....	32
3.5 <i>Scripts</i> .....	34
3.5.1 Vantagens do uso de Bloco de Notas ou Editor de Texto para Comandos em Série .....	34

#### CAPÍTULO 4

<b>Alguns Comandos para se Aventurar nas Moléculas Voadoras .....</b>	<b>36</b>
4.1 Como Carregar uma Molécula no <i>JSmol</i> .....	37
4.1.1 Carregando a Molécula com o <i>Mouse</i> .....	37
4.1.2 Carregando a Molécula por Linha de Comando .....	40
4.1.3 Carregando Biopolímeros (Proteínas, Enzimas, Ácidos Nucleicos) por Linha de Comando .....	41
4.2 Agora que a Molécula está na Página do Navegador, o que Posso Fazer com Ela? .....	43
4.3 Salvamento do Modelo no Computador ou Dispositivo Móvel .....	44
4.3.1 Alguns Movimentos no <i>Jmol</i> .....	45
4.4 Movimentos com <i>Mouse</i> .....	45
4.5 Representações do Modelo .....	46
4.6 Cores .....	48
4.7 Medidas .....	48
4.7.1 Para Distâncias .....	49
4.7.2 Para Ângulos .....	49



4.8 Características Moleculares .....	51
4.8.1 Cargas .....	51
4.9 <i>Scripts</i> e <i>Ensino Reprodutível</i> .....	53
4.10 Características Moleculares .....	55
4.10.1 Nuvens de <i>van der Waals</i> .....	56
4.10.2 Ligações de Hidrogênio .....	57
4.11 Superfícies .....	58

#### CAPÍTULO 5

<b>Algumas Animações com <i>Jmol</i></b> .....	<b>59</b>
5.1 Animar as Moléculas .....	60
5.1.1 <i>Spin</i> .....	60
5.1.2 <i>ZoomTo</i> (Redimensionamento do Tamanho) .....	61
5.1.3 <i>Delay</i> .....	61

#### ***R* e *RStudio*:**

GRÁFICOS E MAPAS INTERATIVOS. ....	63
------------------------------------	----

#### CAPÍTULO 6

##### **Como Usar o *R* e *RStudio*:**

Instalação e Nuvem .....	64
6.1 Instalando o <i>R</i> e <i>RStudio</i> no Computador .....	65
6.2 Acessando o <i>R</i> e <i>RStudio</i> pelas Nuvens .....	67

#### CAPÍTULO 7

<b>Comandos Básicos e <i>Scripts</i> no <i>R</i></b> .....	<b>71</b>
7.1 Uma Visão da Interface <i>RStudio</i> .....	73
7.2 Como Funcionam os Comandos no <i>R</i> .....	75

7.3 Elaborando um <i>Script</i> no <i>R</i> .....	75
7.4 Executando um <i>Script</i> no <i>R</i> .....	76
7.5 Algumas Recomendações sobre a Digitação num <i>Script</i> do <i>R</i> .....	76

#### CAPÍTULO 8

<b>Instalando Pacotes no <i>RStudio</i> .....</b>	<b>78</b>
8.1 Pra Quê Instalar Pacotes (e que viram Bibliotecas)?.....	79
8.2 Instalando o Pacote <i>Plotly</i> no <i>R</i> .....	79
8.3 Carregando o Pacote Instalado .....	81

#### CAPÍTULO 9

<b>Construindo Gráficos Interativos com <i>Plotly</i> .....</b>	<b>82</b>
9.1 Criando um Gráfico Interativo.....	84
9.2 Salvando o Gráfico .....	88
9.3 Trabalhando com Relações Matemáticas nas Variáveis .....	88
9.4 Outros Tipos de Gráficos .....	92
9.5 Gráficos Tridimensionais (3D) .....	98
9.6 <i>Plotly</i> por Comandos de Mouse .....	102
9.7 Referência do pacote .....	102

#### CAPÍTULO 10

<b>Mais Interatividade aos Gráficos.....</b>	<b>103</b>
10.1 Adicionando um Controle Deslizante por Intervalo.....	104
10.2. Emissão de CO <sub>2</sub> e o Efeito Estufa .....	104
10.2.1. Adicionando um Menu Suspenso.....	107

CAPÍTULO 11

**Animação em Gráficos Interativos.....110**

11.1 Elevação da Temperatura Média da Terra..... 111

11.2 Expectativa de Vida  
e Produto Interno Bruto..... 114

CAPÍTULO 12

**Mapas Interativos com *Plotly*.....120**

12.1 Produção Mundial de Óleos ..... 123

**Uma palavrinha final..... 127**

Financiamento ..... 128

A Elaboração desta obra..... 128

**Sobre o autor ..... 129**

**Índice remissivo..... 130**

The background is a solid blue color with a subtle, abstract pattern of overlapping organic shapes, resembling cells or fluid motion. A faint grid of small white dots is visible in the lower right quadrant.

1

POR QUÊ  
“VIVIFICANDO”?

Vivificar moléculas, gráficos e mapas permite a visualização 3D e animações para modelos atômicos, bem como diversos graus de interatividade por cliques de mouse em gráficos e mapas: animações, seleção de pontos, *zoom*, simulações, *mouse over* (informações sobre o ponto gráfico por simples passagem do *mouse*), deslizadores, menus e outros tantos. Como um todo, essas funcionalidades oportunizam elevar o ensino e a aprendizagem das temáticas envolvidas a outro nível!

O material contido aqui pretende ser *autoinstrucional* e com uma curva suave de aprendizado. Nossa proposta é focar no potencial que as ferramentas podem oferecer ao ensino e à aprendizagem como uma *metodologia ativa*, ilustrando seu emprego diretamente nos conteúdos do ensino básico. Dessa forma, não se pretende explorar “*a fundo*” as capacidades das ferramentas propostas, pois isso exigiria um tempo e esforços mais significativos. Assim, você pode pensar neste material, metaforicamente, como um mapa panorâmico de pontos turísticos a visitar, mas sem a pretensão de conhecer cada um em detalhes.

O exemplo abaixo, “por exemplo”, permite construir um gráfico interativo da América do Sul... em 4 linhas curtas de código!

```
library(leaflet)
leaflet() %>% # carrega-se o pacote
addTiles() %>% # adiciona-se as “peças” (tiles)
setView(lng= -56.0949, lat= -15.5989, zoom = 4)
```

Um mapa da América do Sul produzido com apenas 4 pequenas linhas de código. Se essas linhas forem reproduzidas no programa específico (RStudio em nuvem, por exemplo), elas geram um gráfico interativo.



Figura 1 - Mapa da América do Sul



Fonte: produzido pelo autor.

## 1.1 AS FERRAMENTAS

Resumindo rapidamente, a *caixa de ferramentas* para este treinamento possui somente *dois utensílios*: um programa para visualização tridimensional de moléculas e um programa para criação de gráficos e mapas interativos. Ambos são de distribuição livre "*digrátis*" e rodam em navegador de internet, chamado *browser*, sem necessidade de instalação em computador.

Qualquer computador, *notebook*, *tablet* ou *smartphone* com navegador pode acessar os materiais didáticos. Imagine usar até um celular mais antigo, e dos antigos mesmo, para observar objetos interativos como em livros! Bom...antigões, *ma non troppo*, pelo menos um Androidezinho ou iOSsezinho, por favor!

## 1.2 UM POUCO MAIS SOBRE AS FERRAMENTAS

### 1.2.1 VISUALIZAÇÃO DE MOLÉCULAS EM 3D E ANIMAÇÕES - *JMOL*

Existem diversos programas para observar e estudar modelos atômicos em 3D, acessíveis por computador, dispositivos móveis e internet. Alguns são gratuitos, outros oferecem demonstração gratuita e outros são pagos.<sup>1</sup>

O *Jmol*, por sua vez, é um programa multiplataforma (funciona em *Windows*, *Mac OS X*, *Linux* e *Unix*) desenvolvido em *Java*. Ele pode ser usado tanto instalado no computador quanto diretamente no navegador. Uma vantagem da versão para computador é que não precisa de instalação, basta executar o arquivo *Java*.

Assim, o *Jmol* pode ser executado tanto de uma pasta com seus arquivos quanto de um disco rígido ou mídia removível (*pendrive*). Além disso, é um programa de código aberto e gratuito para representar moléculas em 3D, oferecendo diversas opções de visualização e cores, movimentos de translação e rotação, zoom, cálculos de distâncias e ângulos, análise de estruturas e superfícies, otimizações moleculares e animações. Muitos sites na internet utilizam o *Jmol*, como a lista disponível na página da comunidade Wiki.

Complementarmente, é possível acessar o *Jmol* online, sem precisar de arquivos no computador. Entre os vários sites com o *applet JSmol* que permitem esse acesso, sugerimos clicar na imagem abaixo, que leva ao *applet* do site da St. Olaf College, mostrando uma *molécula de água*. <https://chemapps.stolaf.edu/jmol/jmol.php?model=water>

## SUMÁRIO

1 Numa lista pequena, pode-se mencionar Pymol, Maestro, iMolView, VMD, MolView, ChemSketch.

**Figura 2** - Logomarca do programa *Jmol/Jsmol*



Fonte: Jmol/Jsmol.

### 1.2.2 GRÁFICOS ANIMADOS, SIMULAÇÕES E VISUALIZAÇÃO CARTOGRÁFICA

O R é um *software* gratuito e de código aberto, criado para computação estatística e gráficos. Ele pode ser executado em várias interfaces gráficas (GUI - *Graphical User Interface*), sendo o *RStudio* um ambiente de desenvolvimento integrado (IDE) gratuito bastante popular. O *RStudio* também oferece uma versão *online* gratuita, acessível pelo site *RStudio Cloud*.

**Figura 3** - Logomarca da linguagem de programação R



Fonte: R.

<https://cran.r-project.org/>

Com o *RStudio*, é possível realizar desde tarefas básicas, como cálculos, edição de texto e criação de gráficos e tabelas, até atividades mais complexas, graças à sua capacidade de expansão por meio de *pacotes instaláveis*. Esses pacotes podem ser obtidos no site oficial do projeto (R CRAN) ou desenvolvidos e disponibilizados por outras pessoas em seus próprios *sites*.

Para ilustrar essa variedade, alguns pacotes (*library*) permitem análise e criação de textos, interfaces para auxiliar pessoas com deficiência visual, animações, gráficos e tabelas com qualidade de publicação técnico-científica, manipulação e análise de dados e imagens, música, arte, Ciência de Dados, internet das coisas (*IoT*), aprendizado profundo de máquina e até inteligência artificial.

Na pesquisa científica, o *R* oferece diversos *pacotes* voltados para temas específicos e variados, como Ecologia, Estatística, Matemática, Física, Biologia, Química, Artes, Geografia, Geologia e História, entre muitos outros.

**Figura 4** - Logomarca do programa *RStudio*



*Fonte: RStudio.*

Entre os inúmeros pacotes oficiais disponíveis (quase 22 mil atualmente!), alguns possibilitam a criação de objetos didáticos como mencionado anteriormente. Neste material, utilizaremos principalmente **um pacote**: o *'plot\_ly'*, para a produção de gráficos e mapas interativos.

## SUMÁRIO

### 1.3 ALGUMAS POLÍTICAS PÚBLICAS QUE VÃO DE ENCONTRO A ESTE MATERIAL

Em outras palavras, este material busca se alinhar com as atuais iniciativas do *Governo Federal* relacionadas ao Plano Nacional de Educação (PNE) para o período de 2024 a 2034, que está sendo analisado na Câmara dos Deputados (PL 5665/23), foi defendido na Conferência Nacional de Educação - CONAE 2024, e aborda a capacitação digital de professores e alunos (*literacia digital*).

Mais especificamente, este material promove a educação digital para o uso crítico, reflexivo e ético das tecnologias da informação e da comunicação, além de buscar garantir a qualidade e a adequação da formação profissional e tecnológica às demandas da sociedade, do mundo do trabalho e das diversas populações.

Nesse sentido, vale a pena mencionar alguns projetos embasados no desenvolvimento de competências previstas na BNCC Computação de 2022, da *Base Nacional Comum Curricular (BNCC)*, tais como: Escola em Tempo Integral do MEC, a Estratégia Nacional de Escolas Conectadas (Enec), o Programa Mais Ciência Na Escola, e a Chamada para a Apresentação de Propostas de Disseminação de Produtos de Inovação Tecnológica voltados a todos os níveis de educação da CAPES, esse último do qual se originou este projeto.

## SUMÁRIO



The background is a solid orange color with a subtle, abstract pattern of overlapping, rounded shapes that resemble molecular structures or cells. These shapes are in various shades of orange, creating a sense of depth and movement. The overall aesthetic is clean and modern, typical of scientific or educational branding.

***JMOL:***

UM VISUALIZADOR  
INTERATIVO PARA  
MOLÉCULAS EM 3D

# 2

**USANDO O *JMOL*  
PARA OBSERVAR  
MOLÉCULAS EM 3D**

#### Objetivos:

1. Acessar a versão *online* do *Jmol*;
2. Carregar uma molécula no *Jmol*.

## 2.1 ONDE COMEÇAR?

Para começar a usar o *Jmol*, existem algumas opções. Se você pretende utilizá-lo no seu computador, notebook ou diretamente de um *pendrive*, basta acessar o site do *Jmol*, baixar, descompactar e executar o arquivo chamado "*Jmol.jar*" presente na pasta principal no site do *Jmol*.

Se preferir não instalar nada, você também pode acessar o *Jmol online* através de diversos sites. Neste treinamento, utilizaremos um site bastante conhecido, adaptado por um dos próprios desenvolvedores do programa. Basta abrir uma nova aba no seu navegador e clicar neste *link*.

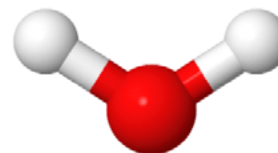
Alternativamente, e ao **longo de todo este material, você poderá copiar o código de qualquer exemplo clicando no ícone de "pastinha" à direita de cada área sombreada. Pronto! O código será copiado para a sua área de transferência. Agora é só colar onde desejar (bloco de notas ou *Jmol*)**. No caso do *Jmol*, copie o *link* abaixo e cole-o em uma nova aba do seu navegador, pressionando *Enter* em seguida.

<https://chemapps.stolaf.edu/Jmol/Jmol.php?model=water>

Agora, clique na molécula com o botão esquerdo do *mouse* ou com o *touchpad* (para *notebooks*) e faça movimentos aleatórios. Ou então, gire o botão do meio do *mouse* ou utilize gestos de pinça (afastar e aproximar com dois dedos) no *touchpad* ou na tela do seu *smartphone* ou *tablet*.

Agora, clique na molécula com o botão esquerdo do *mouse* ou com o *touchpad* (para *notebooks*), e faça movimentos aleatórios. Ou então gire o botão do meio do *mouse*, ou realize gestos de afastamento e proximidade com dois dedos no *touchpad* ou tela de seu *smartphone* ou *tablet*. A figura que segue ilustra o resultado.

**Figura 5** - Tela referente ao modelo da molécula de água renderizado em navegador pelo site do *St. Olaf College*. Para acessar uma "água mais interativa", acesse <https://chemapps.stolaf.edu/jmol/jmol.php?model=water>



Fonte: o Autor (2024).

Essa é a essência principal ao referenciarmos a ideia de **moléculas voadoras**.

## 2.2 COMO CARREGAR UMA MOLÉCULA *ONLINE*?

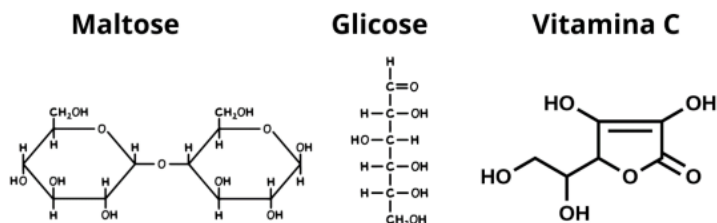
Para *brincar* um pouco com outra molécula, experimente mudar o modelo na própria página da internet, ao final da linha. E agora, seguindo o Material de Apoio Pedagógico para Aprendizagens – MAPA:

**Figura 6** - Algumas representações de moléculas

### ATIVIDADE: IDENTIFICANDO FUNÇÕES OXIGENADAS

**Atividade:** Identificando funções oxigenadas

Examine as estruturas, identifique os grupos funcionais e, em seguida, destaque ou circule os grupos funcionais identificados nas moléculas.



Fonte: MAPA: Ensino Médio - 1º Bim. 3º Ano, p. 51. Ciências da Natureza e suas Tecnologias.



Vamos ilustrar isso com a estrutura da *vitamina C (ácido ascórbico)*.

<https://chemapps.stolaf.edu/Jmol/Jmol.php?model=ascorbicacid>

Você pode experimentar isso com outras moléculas, digitando seus nomes em inglês, já que se trata de um *site* estrangeiro. Ninguém é perfeito! Mas é claro que o banco de dados dessa busca não é ilimitado e, às vezes, o sistema não encontrará a molécula desejada.

Mas existem alternativas. Uma delas é procurar o nome da molécula em um site utilizado como banco de dados, o *PubChem*. Por exemplo, para a vitamina C (ácido ascórbico).

#### Agora é com você:

Entre no site do PubChem;

Procure por Tylenol;

Se existir, digite esse mesmo termo ao final da linha do *JSmol online*, como segue, e veja se deu certo:

<https://chemapps.stolaf.edu/Jmol/Jmol.php?model=tylenol>

## SUMÁRIO

### 2.3 COMO CARREGAR UMA MOLÉCULA *ONLINE*, MAS EM 2D

Às vezes, pode ser interessante visualizar um modelo molecular de forma estática e bidimensional. Para isso, basta acrescentar "*image2d*" à linha de código, como segue:

<https://chemapps.stolaf.edu/Jmol/Jmol.php?model=tylenol&image2d>

3

**CLIQUES DE MOUSE**  
***VERSUS* TEXTO**  
**DE COMANDO**

#### Objetivos:

1. Observar que há 2 formas de conduzir ações em alguns programas: por *mouse* ou por comandos em texto;
2. Observar as características do uso de cada;
3. Conhecer alguns princípios para um "*Ensino Reprodutível*" e as vantagens do uso de linhas de comando ao invés de movimentos de *mouse*.

## 3.1 CLIQUES DE *MOUSE*

Qualquer programa de computador ou aplicativo de dispositivos móveis que você já utilizou provavelmente prioriza a usabilidade centrada na facilidade do emprego de cliques e arrastes com o *mouse*, *touchpad* e até mesmo os dedos (em telas capacitivas). Isso agiliza bastante as ações desejadas. Por exemplo, em editores de texto, é comum clicar em um ícone de formatação (itálico, negrito, por exemplo) ou digitar um atalho para formatar o texto.

Simples, prático e rápido. Dessa mesma forma, pode-se utilizar o *Jmol*, tanto na versão baixada no computador, como na versão *online*. Para a versão baixada, basta observar a gama de itens de menus e submenus. Já para versão de navegador, veja que não há menu!

Não obstante, a versão *online* permite visualizar a mesma informação, embora com outra formatação. Para isso, basta clicar com o *botão direito do mouse* em qualquer área do ecrã (nome clique para a tela contendo alguma informação ou área gráfica, a molécula, no caso).

## 3.2 LINHAS DE COMANDO

Assim como para os cliques do mouse, também é possível acessar um campo de texto para digitar comandos do *Jmol*, tanto na versão baixada

(*standalone*) quanto na versão de navegador (*applet JSmol*). Para a primeira, clique em *File* → *Console*, e uma janela para inserção de texto aparecerá. Na versão *online*, clique com o botão direito do *mouse* em qualquer ponto da tela e escolha *Console*.

### 3.3 CLIQUES DE MOUSE *VERSUS* LINHAS DE COMANDO

Embora o *Jmol* possa ser utilizado tanto por cliques do mouse quanto por comandos de texto, *qual é o melhor método?*

Para auxiliar na resposta, exemplifiquemos com o uso de uma planilha eletrônica, como o Excel do pacote MS-Office, o *Calc* do pacote *LibreOffice* ou o *Planilhas* da *suite* Google. Suponha que você deseje fazer um gráfico simples, pegando duas colunas, cada qual para uma variável (independente ou *x*, e dependente, ou *y*). O usual seria clicar em um item de menu para gráficos, selecionar as colunas desejadas em campos específicos da janela que se abre, selecionar o tipo de gráfico, clicar em *avançar* ou algum termo similar, selecionar outras características (etiquetas ou nomes nos eixos *x* e *y*, por ex) e, finalmente, clicar em *concluir* (ou *OK*, ou termo de significado similar). Simples, rápido e prático.

Mas... (sempre tem um “mas”!), e se você precisasse, além de construir o gráfico, realizar ações adicionais como obter o ajuste linear dos dados, apresentar a reta resultante com determinada cor e estilo, inserir a equação da reta em um ponto específico do gráfico, colocar um título e alterar o símbolo dos pontos (tanto o tipo quanto o tamanho e a cor)? Ufa!

Sem problema, também... desde que você tenha um bom tutorial por perto, claro, ou um estatístico de plantão à sua disposição! Ou que já esteja familiarizado com o programa da planilha, seus menus e as ações necessárias para os vários cliques de mouse que serão precisos para obter um belo gráfico de regressão linear no final.



Agora... mais uma pequena variável a inserir no exemplo: suponha que não seja você a construir o gráfico, mas um aluno(a) da sua disciplina ou um colega de trabalho ou estudo, e que não foi treinado nem no uso da planilha, nem nos cálculos desejados!

Perceba que agora haverá um certo desconforto, posto que:

1. O **aluno(a)** não possui conhecimento prévio no uso da planilha;
2. O **aluno(a)** não possui conhecimento prévio nos cálculos pretendidos;
3. Você terá que treinar o **aluno(a)**, ou oferecer-lhe um *\*guia\** de treinamento correlato;
4. Caso já tenha ocorrido o treinamento, mas não se tenha o *\*guia\** em mãos, tanto você como **aluno(a)** dependerão da *\*capacidade de retenção de memória\** para efetivar com sucesso a empreitada.

Se as instruções para criar o resultado final não estivessem em um *guia* de repetição de cliques do *mouse*, mas sim em um pequeno texto contendo tanto os comandos em sequência quanto os comentários explicando cada ação individual. E se, ao inserir esse texto no programa, ele gerasse o gráfico já formatado, colorido e visualmente atraente, incluindo ainda o ajuste linear e os parâmetros da análise?

## SUMÁRIO

### 3.4 VANTAGENS DO USO DE LINHAS DE COMANDO SOBRE O USO DE CLIQUES DE *MOUSE*

Pelo exemplo hipotético acima, perceba que um pequeno texto contendo as linhas de comando em sequência e os comentários referentes a elas permitem:



## SUMÁRIO

- \* que o produto final seja **\*\*reproduzível\*\*** e não contenha erros; que o produto final seja elaborado sem prévio conhecimento do **aluno(a)**; basta executar o código no programa;
- \* que o produto final seja elaborado independentemente da memória dos **envolvidos** (sequência de cliques, por exemplo);
- \* uma quantidade virtualmente infinita de ações sequenciais, sem necessidade de se decorar a ordem dos cliques de **\*mouse\***;
- \* o aprendizado de cada comando utilizado em linguagem humana, posto que existem comentários do autor para cada linha;
- \* que o produto possa ser modificado para gerar um objeto diferente ao **final** (alteração de cor, etiquetas de eixos, outro título, por exemplo)
- \* que se **\*\*reproduza\*\*** o mesmo gráfico, só que com outro conjunto de **dados** (**\*x\*** e **\*y\***);
- \* que o aprendiz experimente outros comandos para agregar formatações e/ou cálculos distintos ao produto;
- \* que você ou o **aluno(a)** consigam **\*\*reproduzir\*\*** o produto sem recorrer à memória e até por séculos depois, se as previsões de extinção em massa não vingarem;
- \* que qualquer pessoa consiga **\*reproduzir\*** o objeto, independentemente de seu grau de instrução técnica ou de operabilidade do programa;
- \* enfim, que se consiga ensinar determinado conteúdo de modo reproduzível...ou... *Ensino Reprodutível*.

Dessa forma, pretende-se neste curso utilizar somente *linhas de comando*, para que se materializem as vantagens descritas acima, relacionadas a uma metodologia ativa voltada, ainda que de forma inicial, ao *Ensino Reprodutível*, tanto para a ferramenta *Jmol* quanto para as ferramentas *R* e *RStudio*.

Em relação ao *Jmol*, portanto, as ações sequenciais para visualização tridimensional de modelos moleculares serão realizadas pelo *Console*, acessível conforme o item anterior.

## 3.5 SCRIPTS

Dessa forma, quando se possui um conjunto de linhas de comando sequenciais que permitem interagir com um objeto, como um modelo molecular, para inúmeras finalidades, tem-se um *script*. Tecnicamente, um *script* constitui um bloco de instruções sequenciais em texto para compilação em um programa.

No *Jmol* em *browser*, *scripts* podem ser elaborados de duas maneiras:

1. Separando os comandos por “;” - exemplo: “cpk only; color blue”;
2. Separando os comandos por linhas, conforme abaixo:  
“cpk only  
color blue”

Se você deseja que o modelo execute alguns comandos, a melhor opção é separá-los por ponto e vírgula (“;”). Mas se a intenção for algo mais “sofisticado”, sugere-se separá-los por linhas. E mais: linhas comentadas e escritas em um bloco de notas ou em qualquer editor de texto!

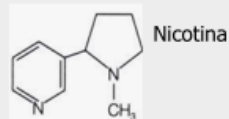
### 3.5.1 VANTAGENS DO USO DE BLOCO DE NOTAS OU EDITOR DE TEXTO PARA COMANDOS EM SÉRIE

Imaginando uma transformação mais significativa na molécula original carregada, como efeitos de ampliação, coloração, representação e movimento, é fácil perceber que um conjunto de linhas comentadas dispostas em sequência facilita tanto a observação do que se pretende com o modelo quanto a identificação de erros e ajustes.

Isso também se reflete nos conceitos de *Ensino Reprodutível*, uma vez que facilita a visualização do código (*human readable format*) e sua depuração (*code debug*). Veja o exemplo a seguir, reflita sobre sua interpretação, copie para um bloco de notas e teste-o no *Console* do *Jmol*.

**Figura 7** - Estrutura da nicotina

P6. Substância do grupo amina presente nas folhas de tabaco.



Fonte: MAPA: Ensino Médio - 2º Bim. 3º Ano, p. 61. Ciências da Natureza e suas Tecnologias.

```
load $nicotine
background black # cor preta do plano de fundo
spin 80 # gira a molécula
delay 3 # aguarda 1 segundo
spin off # interrompe a rotação
cpk # renderiza como modelo de preenchimento
```

### **Agora é com você:**

Copie o trecho de código acima para um bloco de notas;

Altere as linhas de código de forma aleatória, copie e execute novamente no *Console*;

Sugestões para alteração (uma ou outra...ou todas!):

spin 300

delay 1

background magenta

# 4

**ALGUNS COMANDOS  
PARA SE AVENTURAR  
NAS MOLÉCULAS  
VOADORAS**

Objetivos:

1. Carregar uma molécula no *Jmol* de forma alternativa;
2. Utilizar o *Console* para alguns comandos.

## 4.1 COMO CARREGAR UMA MOLÉCULA NO *JSMOL*

O *JSmol* nada mais é do que o próprio *Jmol*, só que desenvolvido para ser utilizado em navegador de internet e que usa, entre suas linguagens, o *JavaScript* (daí o "S" do *JSmol*).

Supondo que você já tenha aberto em seu navegador a janela para o *applet* do *JSmol*, mas que, ao contrário do que foi feito antes (nome da molécula ao final do site *PubChem*), você queira:

- carregar uma molécula a partir de outro banco de dados;
- carregar uma molécula cujo arquivo já esteja em seu computador.

Bom, nesse caso você pode usar o *mouse* ou uma *linha de comando*, como preferir.

### 4.1.1 CARREGANDO A MOLÉCULA COM O *MOUSE*

Para isso, basta clicar com o botão direito do mouse na tela, como anteriormente, e selecionar File -> Load. As opções que se apresentam são:

- \* **Open local file** # abre janela para buscar o arquivo do modelo no computador;
- \* **Open URL** # abre janela para buscar o endereço de *internet* que possui o arquivo



\* **Get PDB file** # abre janela para inserir um código de macromolécula do site homônimo (proteínas, ácidos nucleicos, principalmente)  
\* **Get MOL file** # abre janela para buscar um arquivo \*.mol  
\* **Open script** # abre janela para buscar um trecho de código no computador

A primeira opção é autoexplicativa (*Open local file*). A segunda opção (*Open URL*) depende do endereço correto para um determinado modelo molecular. A terceira (*Get PDB file*) refere-se ao banco de dados Protein Data Bank Brookhaven para biopolímeros. A quarta (*Get MOL file*) envolve a busca *online* em um banco de dados específico para pequenas moléculas. E a última (*Open script*) permite buscar um arquivo que contenha linhas de código do *Jmol* para um conjunto de ações.

Como os livros didáticos geralmente abordam estruturas moleculares pequenas, normalmente associadas aos grupos funcionais da *Química Inorgânica e Orgânica*, bem como exemplos específicos em áreas como *Saúde, Biotecnologia e Indústria*, incluindo também *alguns modelos de macromoléculas*, pode-se concluir que é mais provável que você utilize a busca remota de pequenas moléculas (*Get MOL file*), moléculas contidas em seu computador (*Open local file*) e/ou biomacromoléculas (*Get PDB file*).

O carregamento de pequenas moléculas é idêntico ao que foi experimentado adicionando-se o nome do modelo ao final do endereço do *JSmol*. Já o carregamento remoto para modelos de proteínas, enzimas e ácidos nucleicos requer o conhecimento do *código PDB* desses modelos ou a busca por palavras-chave no site *Protein Data Brookhaven*.

Já o carregamento de moléculas salvas no computador envolve algumas poucas etapas, a saber (... PC?! Coisa de gente velha, quis dizer *desktop* ou *notebook*!):

## SUMÁRIO

- primeiro, obtém-se o modelo da molécula pela *internet* ou o constrói;
- baixa-se o arquivo correspondente ao modelo (geralmente com um atributo \*.mol, \*.cif, \*.cml, \*.sdf, entre mais de 60 formatos);
- carrega-se na página do *JSmol* por dois meios alternativos:

1. Por clique de *mouse*: File --> Load --> *Open local file*;
2. Por arraste do arquivo da pasta onde se encontra para a aba do *JSmol* no navegador.

Por exemplo, digamos que você queira visualizar a estrutura da *aspirina* que você baixou no seu computador.

### Agora é com você:

Acesse o site do *Pubchem*: <https://pubchem.ncbi.nlm.nih.gov/>;

Digite no campo *aspirin* e clique na imagem 3D que aparece;

Baixe o modelo clicando em *Download Coordinates* e seguindo-se com a opção SDF;

Clique no primeiro *link* pra abrir as informações da *aspirina*.  
Baixe o modelo estrutural no *PubChem*;

Abra o *Console* do *JSmol* no navegador (clique no ecrã com o botão direito do *mouse* e selecione *Console*);

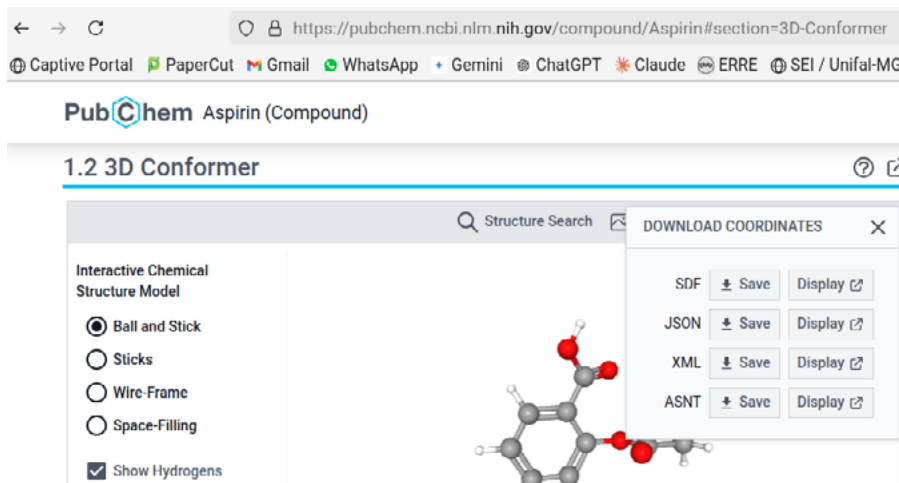
Alternativamente:

Localize o arquivo no PC por "File->Load->Open local file",  
clikando depois em "*Load*" para o carregamento;

Clique no arquivo baixado ("*aspirin.sdf*", por exemplo) e arraste-o diretamente para a janela do *Jsmol*.



Figura 8 - Exemplo do modelo da *aspirina* para \*download\* no site *PubChem*



Fonte: site PubChem.

## 4.1.2 CARREGANDO A MOLÉCULA POR LINHA DE COMANDO

O carregamento de um modelo específico por linha de comando se limita à sua busca pela internet, em bancos de dados ou páginas da *web*. Para isso, abra o *Console* como já explicado. A parte superior exibe os resultados dos comandos, e a parte inferior é para a digitação. Nesse caso, clique no quadro inferior do *Console* e digite o comando de carregamento, exemplificado aqui para um *alcano*:

```
load $alkane
```

O *Console* do *Jmol*, ainda que constitua uma linguagem própria de programação de comandos, possui uma vantagem interessante sobre demais linguagens de programação: é possível efetuar o comando pelo Console tanto com letras maiúsculas como *minúsculas*, e *tanto no singular como no plural*.

Você pode tentar com outras moléculas, como *aspirin*, *cholesterol*, *phenol* etc. (nomes em inglês, por conta do banco de dados). Para recuperar uma linha de comando que foi escrita antes, basta *navegar entre os comandos que foram utilizados com as setas para cima e para baixo do teclado (histórico de comandos)*.

Os modelos moleculares são carregados a partir do banco de dados *Cactus - CADD Group Chemoinformatics Tools and User Services*.

### 4.1.3 CARREGANDO BIOPOLÍMEROS (PROTEÍNAS, ENZIMAS, ÁCIDOS NUCLEICOS) POR LINHA DE COMANDO

Como mencionado acima, o carregamento de macromoléculas biológicas dá-se por meio da identificação de um código alfanumérico correspondente no banco de dados *PDB-Protein Data Bank*. Após obter esse código, você poderá carregar o biopolímero pelo *link online* ou pelo *Console*. Mas saiba que as instruções são diferentes (e não me pergunte por quê!).

#### Pelo Console:

`load=XXXX` # onde XXXX é o código da macromolécula

# Obs: Perceba que o sinal de “\$” é trocado por “=” para o PDB

#### Pelo link online:

`pdbid=XXXX`

# Obs: Como para o link é mais “truquento”, segue um exemplo completo para a bungarotoxina, um veneno proteico de serpentes:

# `https://chemapps.stolaf.edu/Jmol/Jmol.php?&pdbid=1ik8`

Isso pode ser ilustrado pelo carregamento remoto da proteína espícula (*spike*) do vírus SARS-CoV-2, conforme segue:

1. Entre no site do PDB-Protein Data Bank - <https://www.rcsb.org/>;

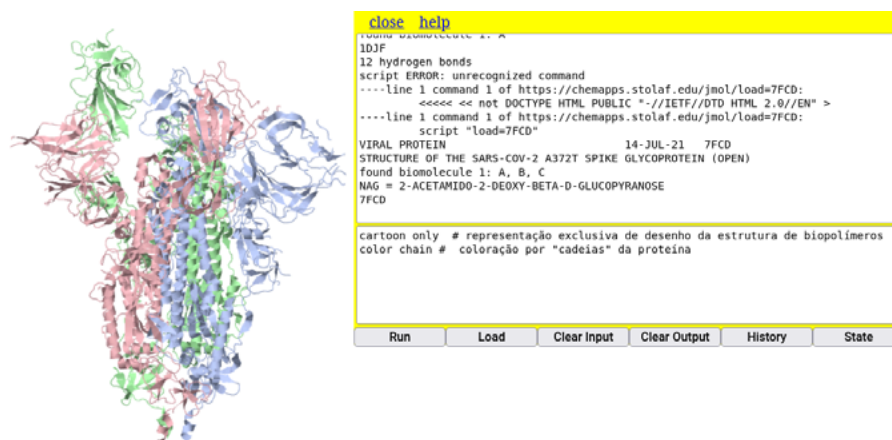
2. No campo de busca, digite “*spike sars-cov-2*”;

3. Selecione a 1ª opção (o site vai direcionar para várias estruturas da proteína espícula);
4. Memorize o código da 1ª. opção (embora qualquer uma também sirva), ou seja, “7FCD”;
5. Digite a linha para carregar a proteína: “load=7FCD” (tanto faz se maiúsculas ou minúsculas).

A representação padrão para proteínas no *Jmol* é a de arame (“wire-frame”). Para visualizar a proteína do vírus de modo mais “amigável” e semelhante ao que aparece em textos ou na internet, digite os comandos abaixo — sua **primeira sequência em linguagem de programação**.

`cartoon only` # representação exclusiva de desenho da estrutura de biopolímeros  
`color chain` # coloração por “cadeias” da proteína

**Figura 9** - Representação da proteína \*spike\* do vírus de Sars-Cov-2 (coloração por número de cadeias)



Fonte: o Autor (2024).

Proteínas, enzimas, ácidos nucleicos e associações macromoleculares são mais pertinentes ao estudo da *Bioquímica* Estrutural. Nesse sentido, convido você a visitar uma parte do website autoral que apresenta descrições e representações detalhadas de estruturas bioquímicas com o auxílio do *Jmol* — o *site* Bioquanti— e que inclui também uma forma “*internetesável*” desse mesmo material.

**Figura 10** - Bioquanti, um *website* autoral para estudos quantitativos em Bioquímica e em Ensino Básico, e que inclui diversos modelos moleculares para o *Jmol*.  
Acesse o *website* em <https://bioquanti.netlify.app/>



Fonte: do autor.

## SUMÁRIO

### 4.2 AGORA QUE A MOLÉCULA ESTÁ NA PÁGINA DO NAVEGADOR, O QUE POSSO FAZER COM ELA?

Muuuuuuuista coisa!

O *Jmol* possui um menu com diversas operações, centenas de comandos e talvez mais uma centena de tutoriais pela *internet*. Para observações estruturais mais diretas e imediatas, contudo, pode-se resumir as operações em:

Movimentos com *mouse* (rotação, translação, *\*zoom\**);  
Representações do *modelo* (bola e varetas, espaço preenchido, arame);  
*Cores* (modelo e plano de fundo);  
*Medidas* (distâncias e ângulos);  
Características *moleculares* (ligações de *H*, nuvem de *van der Waals*, carga parcial e efetiva);  
Superfícies (molecular, eletrostática);  
Seleção de átomos e visualização (água, hidrogênio);  
Animações (*zoom*, rotação automática), cortes.

## 4.3 SALVAMENTO DO MODELO NO COMPUTADOR OU DISPOSITIVO MÓVEL

Todas as ações realizadas com a molécula produzem um novo modelo que pode ser baixado para o computador. Isso é bem legal porque a molécula modificada (com alterações de cores, representações, animações, por exemplo) pode ser carregada no *Jmol* ou no *JSmol* (internet), como já mencionado. Para tanto, pode-se usar cliques do *mouse* ou linhas de comando, como segue:

1. Botão direito no ecrã -> File -> Save -> Save as PNG/JMOL  
# por mouse  
ou...
2. Write nome\_da\_molecula.pngj # por linha de comando no Console

**Uma das características impressionantes do *Jmol* é que, ao salvar a molécula como PNG/JMOL, você poderá abrir o arquivo como uma imagem estática com um duplo clique, apenas para mostrar a molécula, ou mesmo arrastar o arquivo até a janela do *Jmol* no navegador, quando será carregada a estrutura tridimensional e interativa do modelo!**

### **Agora é com você:**

Carregue um modelo para “fenol” digitando no *Console*: `load $phenol`;

Salve o modelo digitando no *Console*: ``write fenol.pngj`;

Mude a orientação do modelo aleatoriamente com o mouse (só dar uma mexidinha);

Localize o arquivo `fenol.pngj` em seu computador;

Abra-o como uma figura normal, só pra testar;

Agora arraste o arquivo para a janela do *Jmol online*, e veja se o modelo substitui o anterior (basta conferir a orientação).

## 4.3.1 ALGUNS MOVIMENTOS NO *JMOL*

Para exemplificar algumas ações, usaremos inicialmente o modelo da *vitamina C*, carregando-o com o comando abaixo no *Console*.

```
load $ascorbate
```

## 4.4 MOVIMENTOS COM *MOUSE*

Para *rotação* e *translação* do modelo, bem como *ampliação*:

*zoom* - botão do meio do *mouse*; se não houver o botão, Shift+botão esquerdo;  
*rotação* - botão esquerdo do *mouse*;  
*translação* - Ctrl+botão direito;  
*rotação no eixo* - Shift+botão direito.



## 4.5 REPRESENTAÇÕES DO MODELO

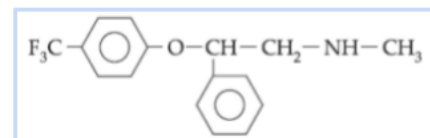
As representações referem-se ao aspecto visual do modelo (renderização) ou ao seu estilo. Assim, o *Jmol* pode renderizar o modelo como *vareta*, *arame*, *espaço preenchido*, *bola e vareta*, *traço* e *desenho* (tudo em inglês, claro — *cpk*, *wireframe*, *trace*, *cartoon*).

Experimente esses estilos, incluindo a opção *only*. Essa opção permite que a ação não seja sobreposta às anteriores (no caso, a sobreposição das representações). Para isso, copie, cole e execute **em cada linha separada** o trecho de código que segue no *Console* do *Jmol*, cuja renderização ilustra a molécula de fluoxetina.

**Figura 11** - Estrutura da *fluoxetina*

### ATIVIDADES DE FIXAÇÃO

1 – (PUC-RS) - A fluoxetina, presente na composição química do Prozac, apresenta fórmula estrutural:



Fonte: MAPA: Ensino Médio - 3º Bim. 2º Ano, p. 79. Ciências da Natureza e suas Tecnologias.

```
load $fluoxetine # carrega a molécula
wireframe only # estilo em arame
cpk only # espaço preenchido
cpk 20 only, bond 1 # bolas e varetas
cartoon only # representação de cartoon, mas que não ocorre com
moléculas pequenas, apenas com biomacromoléculas (proteínas,
ácidos nucleicos, por exemplo)
```



Observe também que a representação em *cartoon* não resulta em uma renderização para o modelo da fluoxetina. Isso ocorre porque a representação em *cartoon* é restrita a *biopolímeros somente, ou seja, proteínas, ácidos nucleicos e algumas associações supramoleculares*.

Contudo, se quiser experimentar o *cartoon*, será necessário conhecer o código alfanumérico de uma proteína ou ácido nucleico. Exemplificando, para a mioglobina — proteína transportadora de oxigênio em mamíferos (código: *1mcy*):

```
load=1mcy # carregando a mioglobina
```

# Obs: veja que a linha de comando pra biopolímeros é ligeiramente diferente da usada pra moléculas pequenas

Note que a renderização padrão para grandes moléculas é a de *bolas e varetas*, pouco didática para o aprendiz. Nesse caso, pode-se representá-la como desenho exclusivo, digitando-se:

```
cartoon only # renderizando em desenho
```

Para obter esse e outros códigos de proteínas e ácidos nucleicos, deve-se acessar o banco de dados do PDB - Protein Data Bank, RCSB, e digitar o nome no campo de busca (no caso, *myoglobin*). O sistema retorna diversos modelos estruturais e seus códigos, bastando transcrever um desses códigos no *Console* do *JSmol*.

Se desejar observar um pouco mais sobre biomoléculas e o uso do *Jmol*, deixo-lhe um convite para visitar nosso portal de Bioquímica Quantitativa. Lá, você encontrará também outras aplicações para o *Jmol*, *R* e *RStudio*, bem como um programa que simula transformações de elementos presentes em esquemas (diagramas, fluxogramas, heredogramas, redes e teias) por diferenças de luminância entre o elemento e seu predecessor/sucessor (SISMA - Sistema de Mapas Autocatalíticos).

## 4.6 CORES

Existe grande flexibilidade de cores para o *Jmol* (e, por consequência, para o *JSmol*), tanto para os modelos inteiros, quanto para partes do modelo (átomos específicos ou um conjunto) e para o plano de fundo. A visualização padrão de cores segue a convenção CPK (Corey–Pauling–Koltun). Exemplificando para o modelo anterior de vitamina C (*load \$ascorbate*), experimente separadamente a variação que segue:

```
color pink
color blue
color lighthgreen
background yellow # cor do plano de fundo
```

O último comando da lista acima permite variar a coloração do plano de fundo.

Adicionalmente, também é possível a coloração das ligações entre os átomos, como segue:

```
color bonds LightSeaGreen
```

Para um grande espectro de cores, você pode consultar a referência do *Jmol Colors* ou um *link* mais "mastigado", de nossa autoria, junto ao material de aprendizado do programa no ensino superior: o portal *Bioquanti* e, mais especificamente, o tópico de cores para o *Jmol*.

## 4.7 MEDIDAS

O *Jmol* permite calcular distâncias e ângulos em um modelo molecular. Para exemplificar isso, talvez seja interessante carregar um *modelo de água* ("load \$water"), cujas distâncias e ângulos estão presentes em alguns livros de Química.

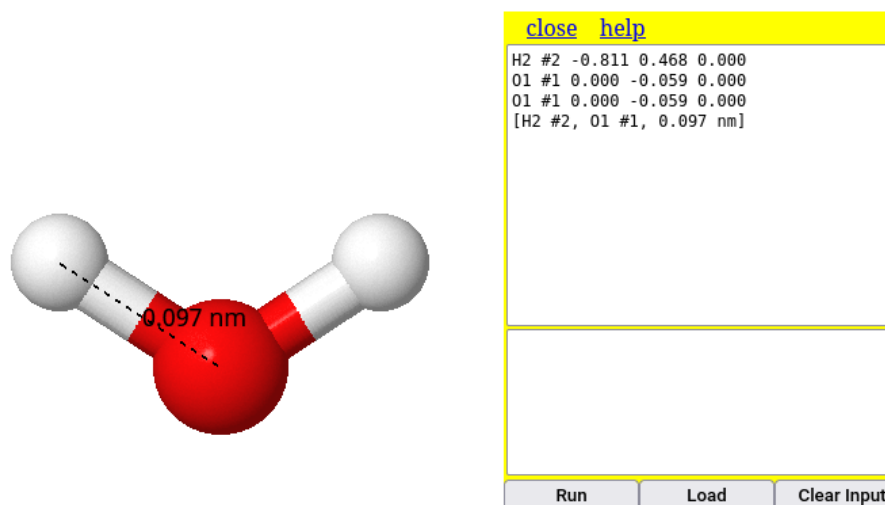
## 4.7.1 PARA DISTÂNCIAS

No exemplo da molécula de água, para se determinar a distância de uma ligação O-H, por exemplo execute:

1. Duplo-clique do mouse no 1º átomo;
2. Arraste do mouse ao 2º átomo;
3. Clique do mouse no º átomo.

Experimentando para a distância da ligação O-H, o programa retorna o valor de 0,097 nm, ou 0,97 Ångströms, o valor convencional para esse tipo de ligação covalente.

**Figura 12** - Medindo distâncias dentro da molécula



Fonte: o Autor (2024).

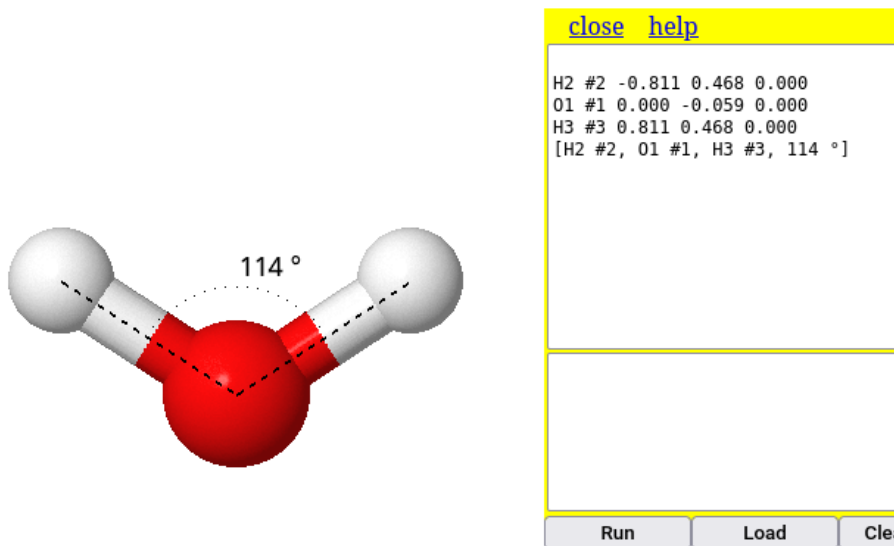
## 4.7.2 PARA ÂNGULOS

Para a mesma molécula de água, experimente determinar o ângulo de ligação:

1. Duplo clique no 1º átomo (exemplo: H);
2. Arrasta ao 2º átomo (exemplo: O);
3. Clique no 2º átomo;
4. Arraste ao 3º átomo (exemplo: o outro H);
5. Clique no 3º átomo.

Perceba que o sistema retorna o valor de 114 graus, um valor próximo do previsto para a molécula (109,5 graus) ou medido (104,5 graus). Essa aproximação decorre da construção do modelo de água.

**Figura 13 - Medindo ângulos dentro da molécula**



Fonte: o Autor (2024).

Para limpar as medidas, use o comando:

measure off

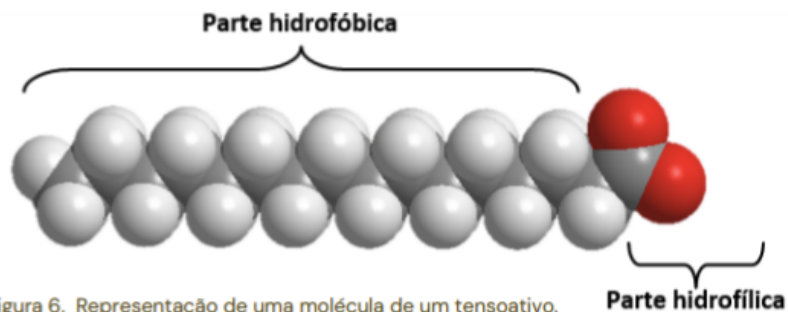
## 4.8 CARACTERÍSTICAS MOLECULARES

São diversas as informações tangíveis a um modelo molecular no *Jmol*. Exemplificando as mais básicas para a molécula de um componente do molho *shoyo*, o *glutamato*.

### 4.8.1 CARGAS

Por vezes, pode ser interessante apresentar a polaridade das moléculas em função de sua distribuição de cargas. No *Jmol*, há dois tipos de cargas: *carga efetiva* (*formaCharge*) e *carga parcial* (*partialcharge*). Podemos ilustrar a distribuição de cargas em uma molécula de tensoativo, como o *hexadecanoato*.

**Figura 14** - Exemplo de tensoativo de 16 carbonos, o ácido hexadecaenoico



**Figura 6.** Representação de uma molécula de um tensoativo.

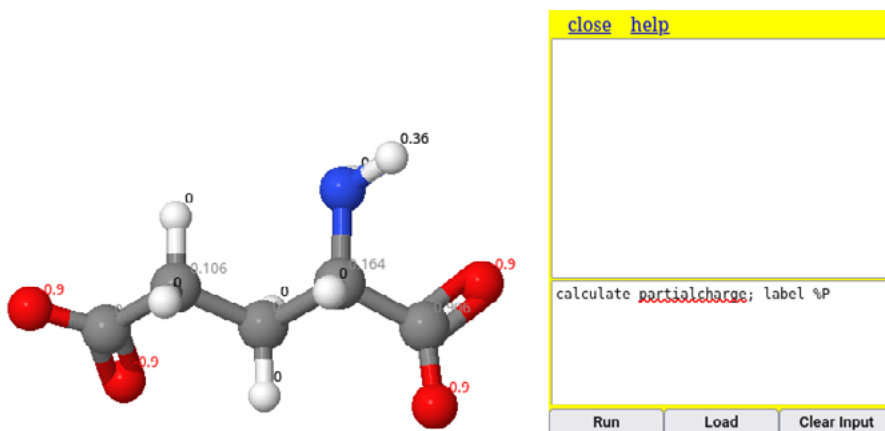
Fonte: MAPA: Ensino Médio - 3º Bim. 3º Ano, p. 63. Ciências da Natureza e suas Tecnologias.

Para isso, digite no *Console* os comandos abaixo:

```
load $hexadecanoate
calculate partialCharge # cálculo de cargas parciais do modelo
label %P # apresentação das cargas (etiquetagem)
```

Uma característica do *Jmol* que torna a execução de suas ações mais eficiente é a disposição sequencial dos comandos. Dessa forma, não é necessário clicar em *Enter* para cada comando, bastando separar os comandos por ponto e vírgula (;) — como ilustrado abaixo, para o cálculo de cargas parciais da molécula de *glutamato*:

**Figura 15** - Apresentação de cargas parciais no modelo de glutamato, um componente do molho *Shoyo*, também ilustrando uma sequência de ações no *Jmol*

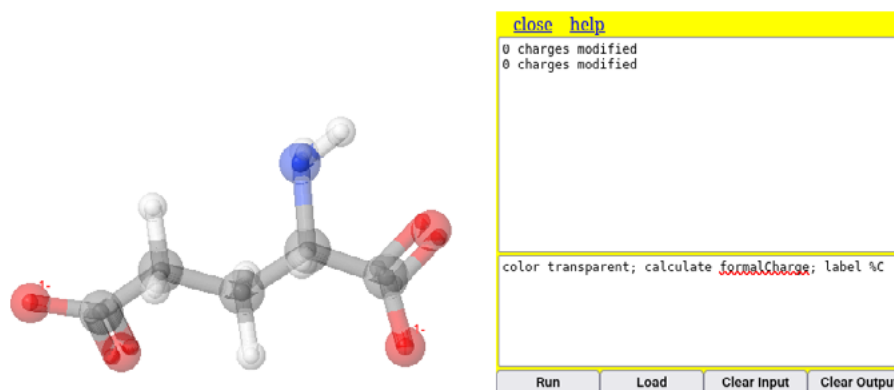


Fonte: o Autor (2024).

Da mesma forma, pode-se ilustrar a obtenção de *cargas formais* no modelo. Nesta, adicionou-se a coloração transparente para melhor visualização da carga unitária negativa do *ácido carboxílico*:

```
calculate formalcharges # cálculo de cargas parciais do modelo
label %C # apresentação das cargas (etiquetagem)
```

**Figura 16** - Ilustração dos comandos em sequência para visualização de cargas formais na molécula de *glutamato*



Fonte:

Perceba que os comandos da figura misturam maiúsculas e minúsculas, de modo diferente da linha de comando que a antecede. Essa é uma **característica bem legal do Jmol, que não se importa com a capitalização da fonte**. Ou seja, tanto faz se minúsculo, maiúsculo ou uma combinação de ambos: o Jmol executa a ação do mesmo modo.

## 4.9 SCRIPTS E ENSINO REPRODUTÍVEL

O exemplo acima apresenta uma maneira simples de concatenar comandos, facilitando a execução automática e sequencial de um conjunto deles. No entanto, a visualização da linha de comando fica um pouco prejudicada com a separação por “;”, o que pode acarretar uma poluição visual quando houver vários comandos.

A situação de contorno envolve a disposição dos comandos no formato de um *script*. Esse nada mais é do que um trecho de código contendo um comando por linha, o que melhora a visualização do código como um todo. Além disso, o *script* possui a vantagem adicional de permitir a inserção



de comentários entre as linhas de comando, facilitando uma melhor apropriação do código e do seu aprendizado.

Essas características de um *comando por linha*, com comentários explicativos, conferem ao *Jmol* seu aspecto para *programação* de ações sequenciais e enraizam, por consequência, uma das premissas básicas para um *Ensino Reprodutível*: a redação de trechos de código em comandos unitários por linha, escritos como num bloco de notas, com comentários sobre as ações do programa em cada linha.

Exemplificando para um *script* envolvendo as ações para o glutamato acima, apenas copie o trecho abaixo e cole-o no *Console* do *JSmol*, executando-o.

```
load $glu # carregamento de micromolécula
wireframe only # renderização exclusiva de varetas
calculate partialCharge # carga parcial
label %P
```

Outro aspecto inerente à iniciativa de *Ensino Reprodutível* reside na *possibilidade de avaliar o código com alguma alteração*, objetivando um produto final ligeiramente modificado. Tente repetir o trecho acima, mas para cargas efetivas, ou seja:

```
load $glu # carregamento de micromolécula
cpk only # renderização exclusiva por espaço preenchido
calculate formalCharge # carga efetiva
label %C
```


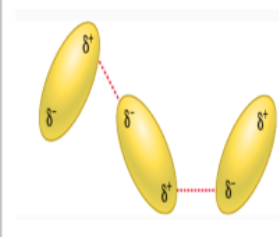

Complementarmente, pode-se atuar alterando mais comandos do código, de modo a criar um resultado completamente diferente do original. Isso define outra característica do *Ensino Reprodutível*, qual seja, a *criação de trechos de código*. Ilustrando, segue um trecho baseado no anterior, mas para minimização de energia e reestruturação dos orbitais da molécula.

load \$glu # carregamento de micromolécula  
cpk only # renderização exclusiva por espaço preenchido  
minimize # comando para minimização de energia da estrutura

## 4.10 CARACTERÍSTICAS MOLECULARES

Por vezes, também é interessante apresentar à turma o conceito de *forças fracas* que permeiam as interações moleculares, tal como ilustrado abaixo.

Figura 17 - Algumas forças fracas em interação molecular

Forças de London ou dipolo induzido dipolo induzido	ou dipolo -dipolo	Ligação de Hidrogênio
Únicas existentes entre moléculas apolares e de gases nobres. Exemplos de substâncias apolares: $\text{Cl}_2$ , $\text{O}_2$ , $\text{CH}_4$ , $\text{F}_2$ .	Esse tipo de força intermolecular é característico de moléculas polares. Exemplos de substâncias polares: $\text{HBr}$ , $\text{HCl}$ , $\text{CO}$ .	Ocorre mais comumente em moléculas que apresentam átomos de hidrogênio ligados a átomos de flúor, oxigênio e nitrogênio. Ex.: $\text{H}_2\text{O}$ , $\text{NH}_3$ .
		

Fonte: MAPA: Ensino Médio - 2º Bim. 1º Ano, p. 75. Ciências da Natureza e suas Tecnologias.

Além da previsão estrutural para *carga parcial* e *carga formal*, o *Jmol* também permite evidenciar *forças fracas* no modelo, tais como nuvem de *van der Waals* e ligações de hidrogênio, como segue.

#### 4.10.1 NUENS DE *VAN DER WAALS*

Para a representação de delocalização eletrônica de moléculas, digite:

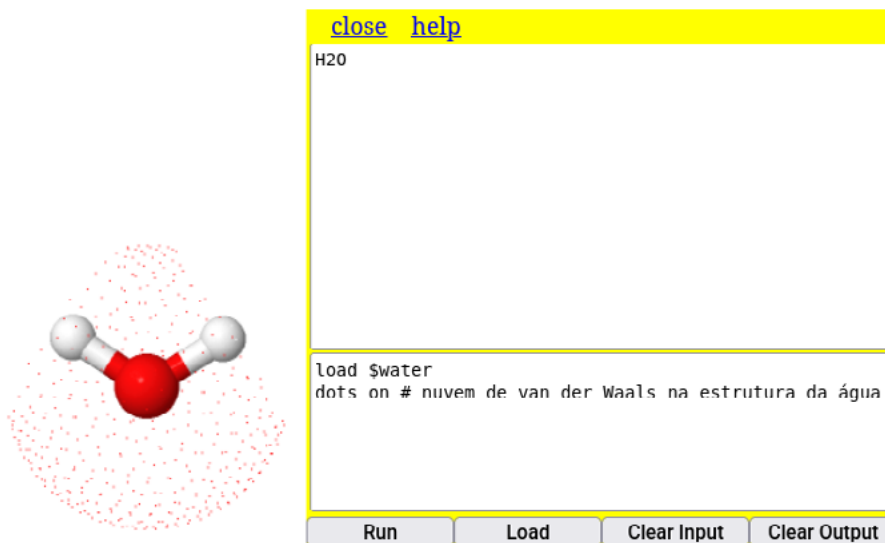
```
dots on # nuvem de van der Waals nos átomos do modelo (retira-se com “dots off”)
calculate hbonds # identifica ligações de hidrogênio no modelo
```

Ilustrando, copie e cole o trecho que segue no *Console*:

```
load $water
dots on # nuvem de van der Waals na estrutura da água
dots ionic # nuvem iônica sobre o modelo
```

A primeira linha renderiza a nuvem de *van der Waals*:

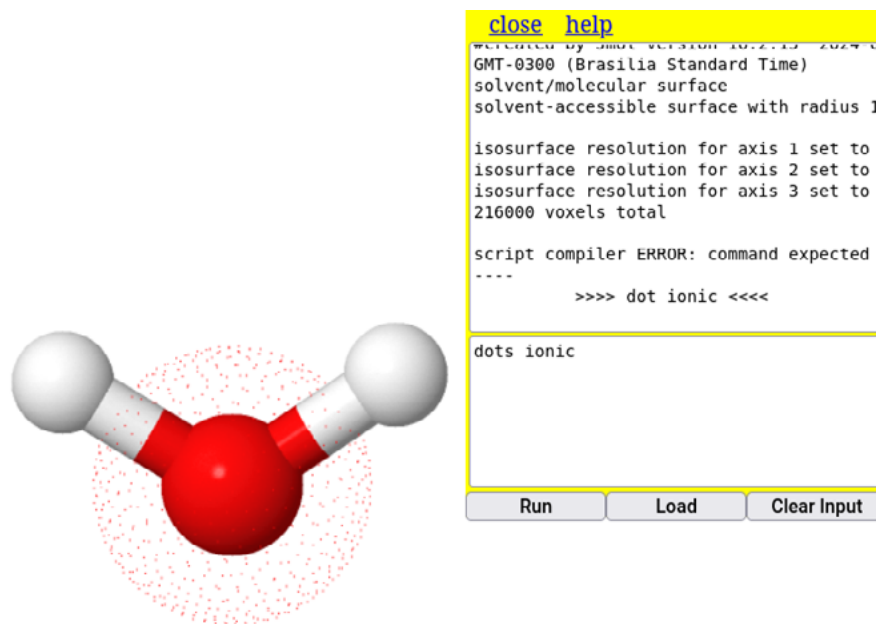
**Figura 18** - Exemplificando a sobreposição de nuvens de *van der Waals* nos átomos da molécula de água



Fonte: o Autor (2024).

A segunda linha de comando, por sua vez, resulta numa nuvem iônica para o modelo:

**Figura 19** - Ilustrando a representação de nuvem iônica para o modelo da água



Fonte: o Autor (2024).

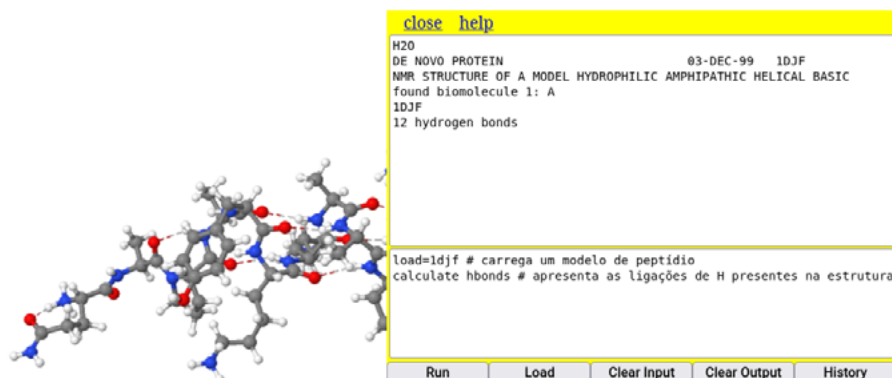
## SUMÁRIO

### 4.10.2 LIGAÇÕES DE HIDROGÊNIO

Há comandos variados no *Jmol* para a configuração de ligações de *H* (antigamente chamadas de pontes de hidrogênio). Mas segue comandos diretos para sua visualização.

```
load=1djf # carrega um modelo de peptídio
calculate hbonds # apresenta as ligações de H presentes na
estrutura
```

Figura 20 - Exemplificando ligações de hidrogênio num modelo de peptídeo



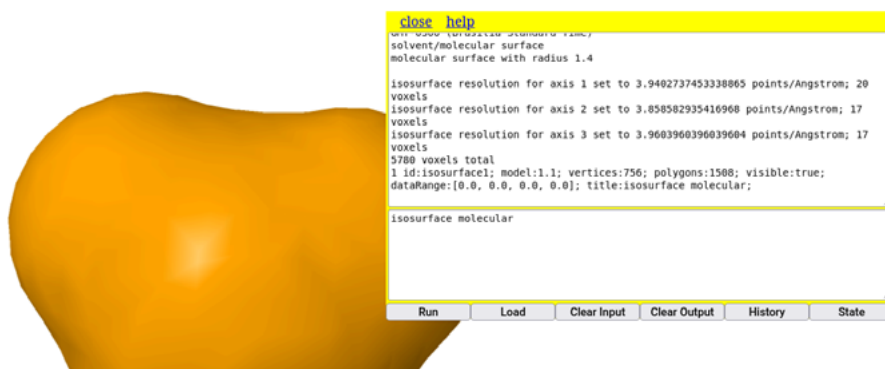
Fonte: o Autor (2024).

## 4.11 SUPERFÍCIES

Além da superfície de *van der Waals* (*dots on*) vista acima, o *Jmol* é capaz de representar algumas superfícies para modelos moleculares. Quanto maior a molécula, maior o cálculo interno necessário para gerar a superfície, o que pode dificultar sua visualização. Assim, ilustrando um comando simples para a superfície da molécula de água:

`isosurface molecular # superfície molecular que inclui o solvente`

Figura 21 - Superfície molecular para o modelo da água



Fonte: o Autor (2024).



# 5

**ALGUMAS  
ANIMAÇÕES  
COM *JMOL***



Objetivos:

1. Compreender uma animação de molécula como um *script*;
2. Utilizar o *Console* animação de moléculas.

## 5.1 ANIMAR AS MOLÉCULAS

E chegamos à última parte do treinamento com o *Jmol*, para apresentar um efeito didático-pedagógico bem interessante do programa: *animação de moléculas*.

Pode ser que você queira mostrar uma molécula com estilos distintos durante um determinado período, fazer uma transição lenta entre cores em átomos específicos ou ainda aplicar efeitos de ampliação, redução ou rotação controlados. E outros tantos que a imaginação e o rigor técnico permitirem ao estudo.

Nesses casos, é interessante conhecer e aplicar alguns poucos comandos para animação, explicados a seguir. Os exemplos foram selecionados de modo a simplificar cada comando. Mas não se iluda quanto à capacidade do *Jmol*, porque para cada comando sempre há várias opções complementares. E, se você quiser saber algo sobre qualquer comando do *Jmol*, faça uma visita ao *site* de referência.

### 5.1.1 *SPIN*

Comando simples de rotação da molécula.

```
load $glucose; spin 30 # o valor refere se à velocidade de rotação
```

## 5.1.2 *zoomto* (REDIMENSIONAMENTO DO TAMANHO)

Esse comando impressiona, pois permite uma ampliação ou redução controlada no tempo. Sua sintaxe é:

*\*zoomTo\** : tempo (expressão opcional) tamanho

Exemplos:

Aumentar em 3x, meio segundo por vez: `zoomto 0.5 *3`

Aumentar em 4x, meio segundo por vez: `zoomto 0.5 400`

Focar num ligante de biopolímero com ampliação de 2x: `zoomto 2(ligand) 0`

Focar num ligante de biopolímero com ampliação de 4x, a meio segundo por vez: `zoomto 0.5(ligand)* 4`

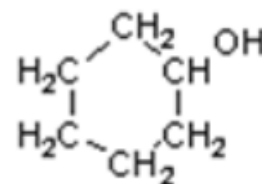
## 5.1.3 *delay*

Esse é um **comando-chave** para qualquer animação, pois permite que a imagem da molécula no *Jmol* realize uma pausa (em segundos) antes da próxima ação. É utilizado, normalmente, na sequência de comandos pelo *Console*, ao longo de um *script*.

`delay 3` # aguarda 3 segundos antes da próxima ação

Podemos experimentar uma animação com os comandos acima junto ao modelo do *ciclohexanol*.

Figura 22 - Estrutura do *ciclohexanol*



ciclohexanol

Fonte: MAPA: Ensino Médio - 2º Bim. 3º Ano, p. 57. Ciências da Natureza e suas Tecnologias.

Segue um trecho de código para você experimentar animações com o *Jmol*. Basta copiá-lo, colar no *Console* e executá-lo. Se preferir alguma modificação, é melhor copiar o código para um bloco de notas, realizar as alterações desejadas e testar o *script* no *Console*.

#### Agora é com você:

```
load $cyclohexanol
delay 2
spin Y 70
delay 2
spin off
zoomTo 2 *2
cpk
delay 2
color transparent
zoomTo 2 *0.5
spin 50
```

The background is a solid orange color with a complex pattern of overlapping, organic, wavy shapes in various shades of orange and brown. In the lower right quadrant, there is a faint grid of small white dots.

# **R E RSTUDIO:**

GRÁFICOS E MAPAS  
INTERATIVOS

# 6

**COMO USAR  
O *R E R* STUDIO:  
INSTALAÇÃO E NUVEM**



Objetivos:

1. Conhecer o *R* e o *RStudio*;
2. Compreender os passos para instalação em computador;
3. Acessar o *RStudio* pelas nuvens.

Existem dois ambientes alternativos para se utilizar o *R* (programa) e o *RStudio* (interface do usuário): instalando no computador ou pelas nuvens. Há poucas diferenças entre ambas as maneiras de trabalhar, mas a principal é que a *instalação* permite o uso dos programas *offline*, sem necessidade de internet, porém, pelas nuvens... bom, você já sabe. Por outro lado, na versão em nuvem não é necessário instalar nem o *R* nem o *RStudio*, enquanto que, na versão instalada, existem alguns passos para isso, descritos abaixo.

Contudo, você poderá utilizar a versão em nuvem durante todo o *Curso*, não sendo necessário preocupar-se com as instalações do *R* ou do *RStudio*.

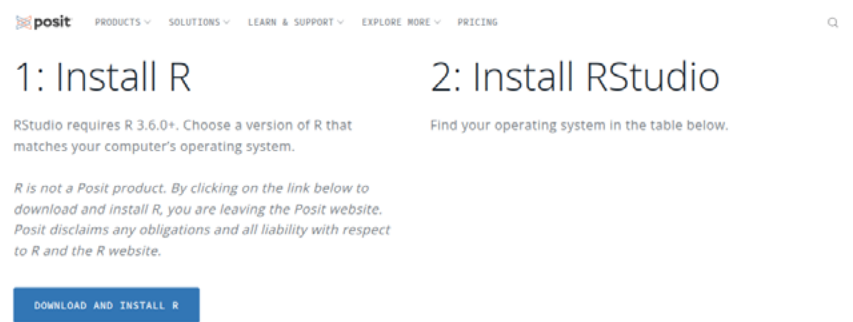
## 6.1 INSTALANDO O *R* E *RSTUDIO* NO COMPUTADOR

Você precisa seguir apenas alguns passos para instalar o *R* e o *RStudio* no computador. Na prática, ambos os programas são baixados e instalados como qualquer outro *software*, seja para *Windows*, *Linux* ou *Mac*. A seguir, apresentamos os passos:

1. Acesse o site do *RStudio* e faça o *download* do programa *R*;



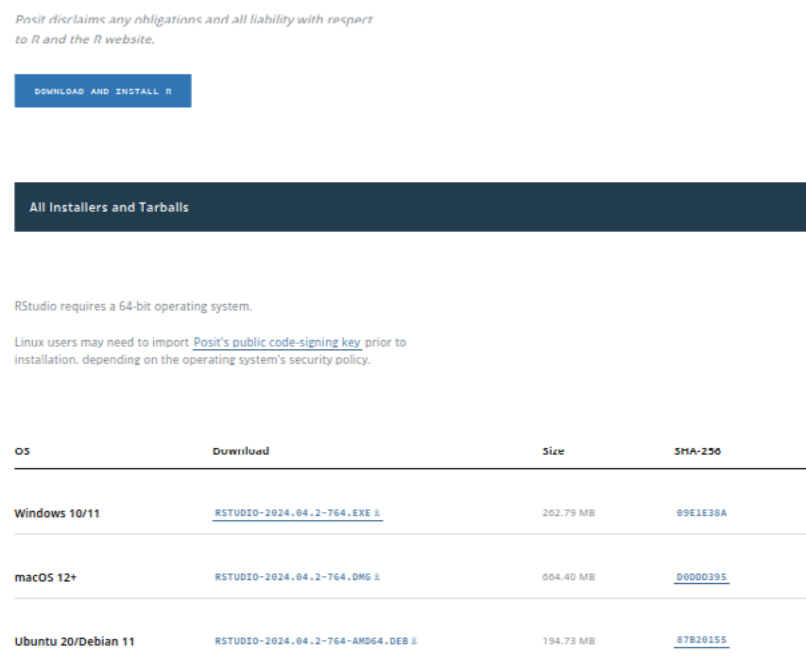
Figura 23 - Site da Posit para baixar o RStudio



Fonte: o Autor (2024).

2. Instale o programa com as opções padrão;
3. No mesmo site do RStudio procure um pouco mais abaixo pelo instalador mais apropriado a seu sistema operacional;

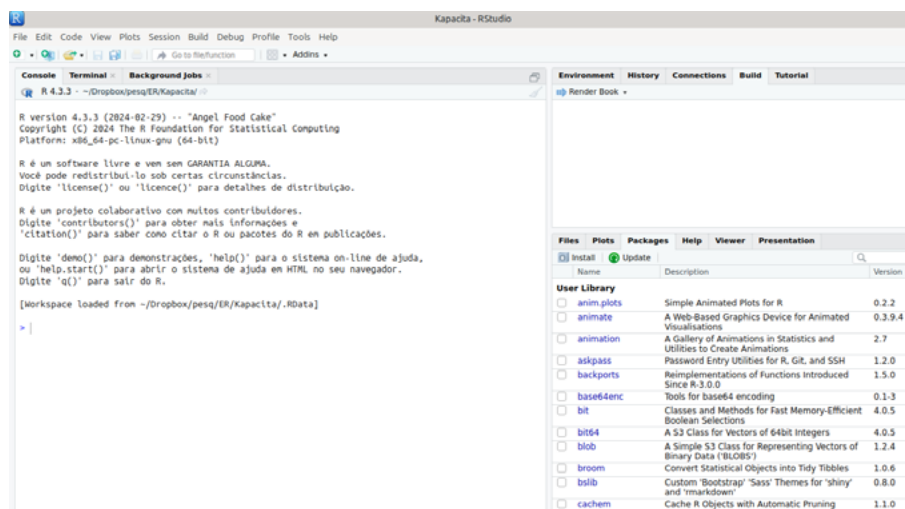
Figura 24 - Página de instalação para o RStudio



Fonte: o Autor (2024).

4. Baixe o arquivo e instale-o como qualquer outro programa;
5. Abra o programa *RStudio*, e cuja interface será parecida com a que segue;

**Figura 25** - Visão de abertura do *RStudio*, apresentando janela de comandos, manipulação de dados e arquivos, e de pacotes e visualizações



Fonte: o Autor.

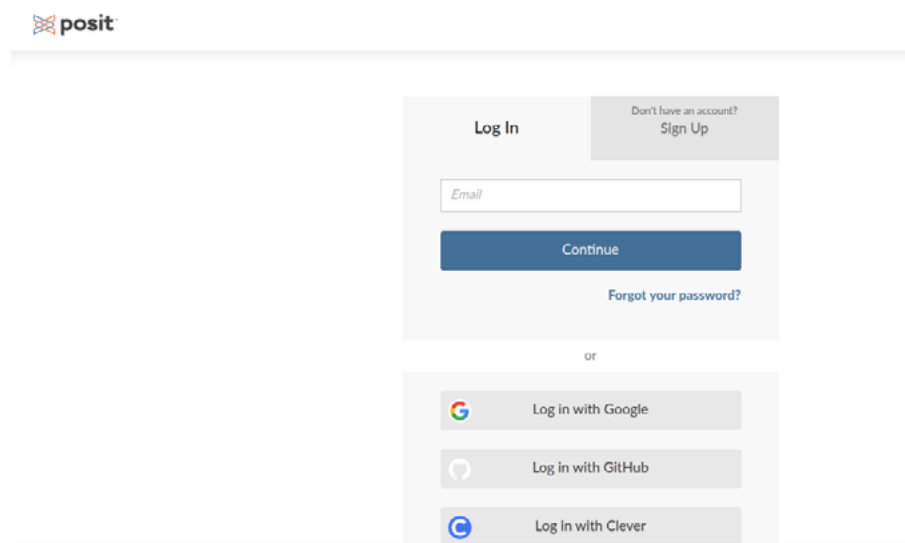
## SUMÁRIO

## 6.2 ACESSANDO O R E RSTUDIO PELAS NUVENS

Essa é uma opção simples e que não requer qualquer instalação. A interface acessada é praticamente igual à da instalação no computador. Entre as vantagens, destaca-se a velocidade normalmente superior para rodar e instalar pacotes, já que o servidor está na nuvem. Por ser um acesso online, é necessário realizar uma inscrição inicial, com *login* e *senha*. A seguir, apresentamos os passos:

1. Acesse o *site* do *RStudio Cloud*

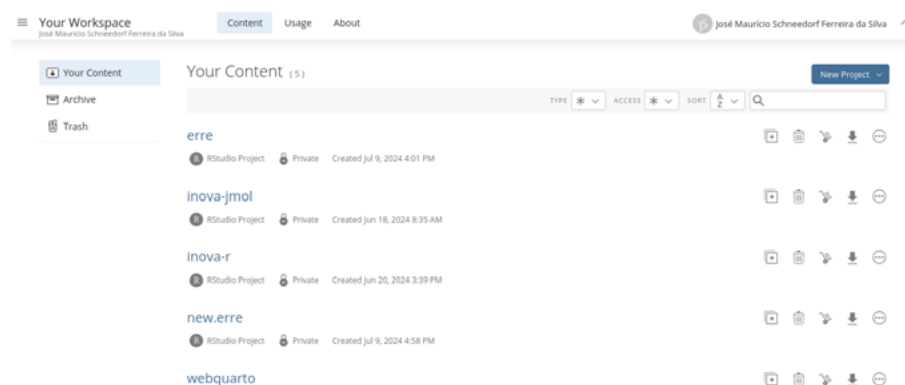
Figura 26 - Site RStudio Cloud



Fonte: o Autor (2024).

2. Realize a inscrição (*sign up*) ou acesse pelo *Google* (mais simples);
3. A janela deverá parecer-se com a que segue, embora sem os projetos listados;

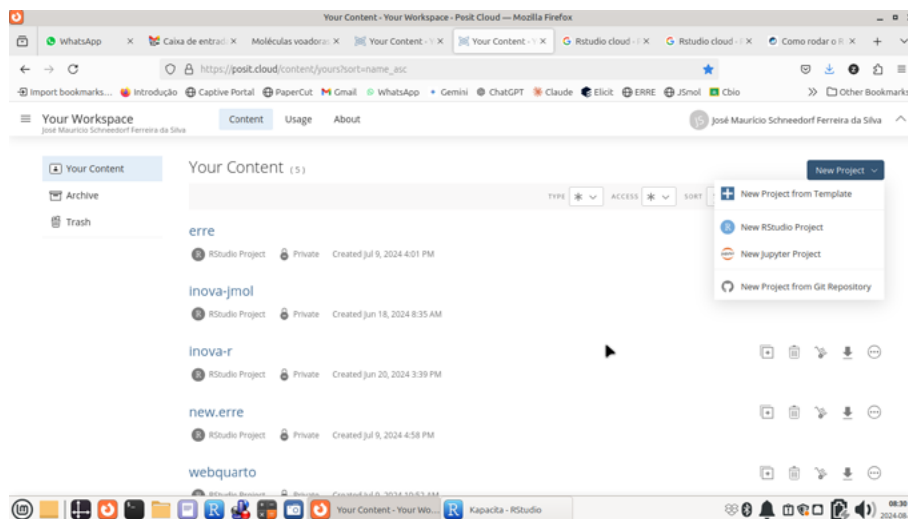
Figura 27 - Página de acesso da Posit para o RStudio em nuvem



Fonte: o Autor (2024).

- Agora a parte interessante. Clique em *New Project* e selecione *New RStudio Project*.

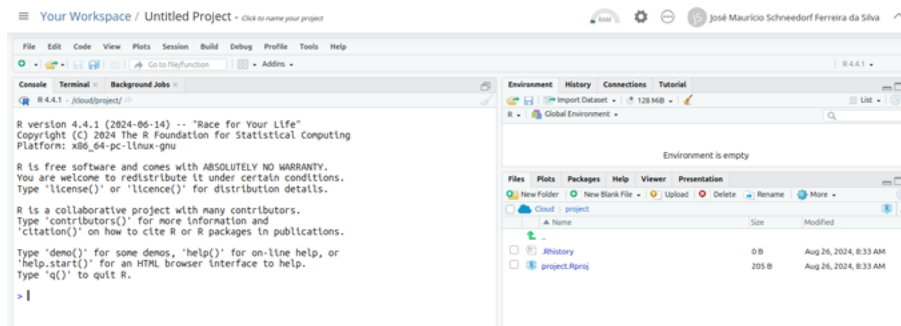
**Figura 28** - Página principal de projetos do *RStudio* em nuvem



Fonte: o Autor (2024).

A imagem final será bem parecida com a apresentada pela versão instalada, veja:

**Figura 29** - Página do *RStudio* em nuvem, aberto e funcional, tal como a versão instalada em computador



Fonte: o Autor (2024).

Pronto! Você pode utilizar uma ou outra forma para acessar as atividades propostas. A seguir, serão apresentadas algumas ações e comandos rápidos para trabalhar com a interface do *RStudio* e com o *R* — apenas o necessário para criar, executar e modificar alguns *scripts* desenvolvidos para *interatividade por meio de animações, simulações e visualização cartográfica*.

Mãos à obra, agora!

# 7

## COMANDOS BÁSICOS E *SCRIPTS* NO *R*



#### Objetivos:

1. Entender para que servem as janelas e abas do *RStudio*;
2. Compreender a lógica de comandos e atributos do *R*;
3. Utilizar comandos por *script*.

O *R* é um programa que opera por linha de comando. Isso é um pouco chato, como já vimos, pois qualquer erro na digitação de um comando resulta na interrupção da leitura do código. Por outro lado — e também como já vimos — *linhas de comando encadeadas e comentadas permitem a reprodução e modificação de trechos de código voltados à criação de um produto* qualquer, neste caso, objetos didáticos para o Ensino Médio, e *sem a necessidade de memorizar cliques de mouse ou operações técnicas*.

Diferente do *Jmol*, contudo, o *R* é bem chatinho quanto à sintaxe, não sendo possível mesclar letras maiúsculas e minúsculas, nem usar singular e plural de forma intercambiável. Para que o código funcione, é necessária a digitação correta. Por outro lado, pode-se *tranquilamente aumentar ou reduzir o espaço entre os comandos*, o que não faz diferença para o compilador do *R*.

Algumas operações são realizadas alternativamente por mouse, linha de comando ou ambos, dependendo da ação. A seguir, serão apresentadas algumas funcionalidades básicas para a reprodução de códigos para objetos didáticos, sem descrições detalhadas das operações próprias do *R* e do *RStudio*, para simplificar e tornar este trabalho mais objetivo.

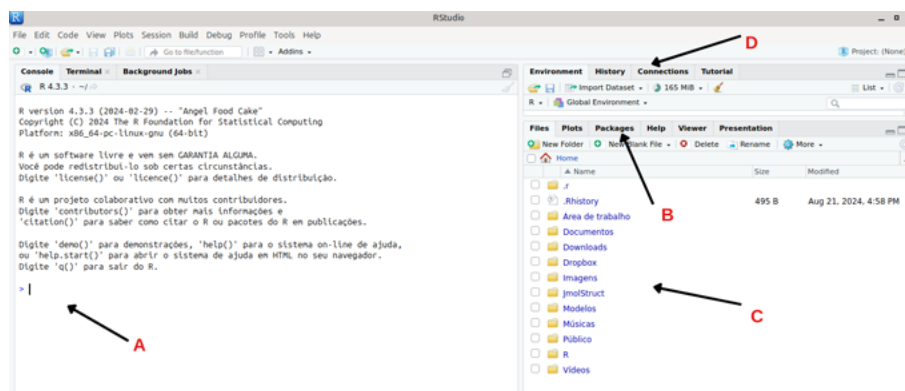
Se você desejar saber mais a respeito de ambos os programas — versão instalada ou em nuvem — sugerimos os inúmeros sites e tutoriais disponíveis na internet, além de centenas de livros já escritos sobre o assunto e cursos *online* em várias plataformas de ensino.

## SUMÁRIO

## 7.1 UMA VISÃO DA INTERFACE *RSTUDIO*

O *RStudio* nada mais faz do que oferecer uma *interface gráfica para o usuário* do R (ou *GUI*, do inglês *Graphical User Interface*), sendo este um programa executado por linguagem própria de códigos, assim como o *Jmol*. Diversas operações também podem ser realizadas sem comandos ou códigos, como abrir e salvar um arquivo, ou visualizar e salvar um gráfico, por exemplo. Vejamos a divisão da janela principal do *RStudio*.

**Figura 30** – Janela básica do *RStudio*



A - área de digitação de comandos ("prompt");

B - área de abas de trabalho (diretório, gráficos, pacotes, etc.);

C - arquivos que aparecem na aba homônima;

D - área de abas de administração (ambiente, história de comandos, etc.).

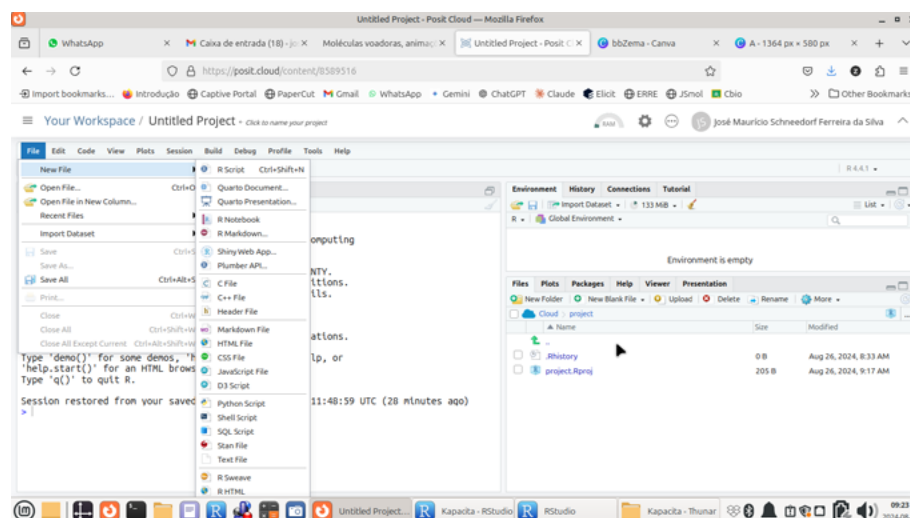
Fonte: o Autor (2024).

Para o treinamento aqui, contudo, será interessante uma área adicional, a área de *scripts*, a qual se acessa como segue:

File --> New File --> *RScript*

... ou por atalho: **Ctrl + Shift + N**

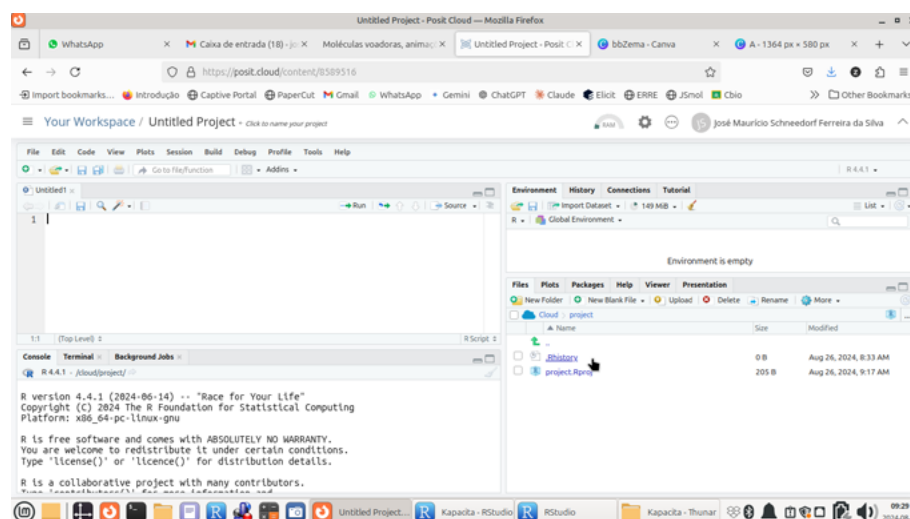
Figura 31 - Ação para criar um "script" no RStudio



Fonte: o Autor (2024).

Veja que agora a janela principal se dividiu em mais uma parte, a que incluiu a aba nova para *scripts*.

Figura 32 - Janela principal do RStudio contendo a aba para elaboração e execução de scripts



Fonte: o Autor (2024).

## 7.2 COMO FUNCIONAM OS COMANDOS NO R

Todos os comandos do *R* são compostos por um *nome* seguido de *argumentos*, esses entre parênteses. Seguem exemplos.

```
comando(argumento 1, argumento 2, argumento 3, ...)
```

Exemplos:

```
plot(x,y)
```

```
mean(z)
```

```
read.csv(file = "meus.dados.csv")
```

## 7.3 ELABORANDO UM SCRIPT NO R

Para produzir um *script* no *R*, a melhor forma é redigir as linhas de comando de modo similar ao que foi realizado com o visualizador molecular 3D *Jmol* em seu *Console*, separando-as em linhas individuais. Veja um exemplo de cálculo simples:

```
x = 5  
x^2 + 7  
[1] 32
```

Para executar o *script* acima, basta copiá-lo e colá-lo na área de *script* aberta. E aqui vai uma **dica de ouro**: no canto superior direito do *script* existe um ícone para copiar o texto do *script*. Basta clicar nesse ícone, e o texto será copiado.

Agora é só colar na aba do *script* aberto (em nuvem, por exemplo) e executá-lo como segue.

## 7.4 EXECUTANDO UM *SCRIPT* NO *R*

Existem algumas formas de se executar um *script*, como no exemplo acima, no *R*. Seguem as mais comuns:

1. Se deseja executar algumas linhas de um *\*script\**, pode-se selecionar as linhas e clicar Ctrl + Enter ;
2. Se desejar executar todo o *\*script\**, seleciona-se todo o texto (Ctrl + A) seguido da ação acima, Ctrl + Enter ;  
Opcionalmente, pode-se clicar no ícone “-->Source”;
3. Se desejar executar apenas uma linha, basta clicar na linha seguido de Ctrl + Enter ;  
Opcionalmente, pode-se clicar no ícone “-->Run”;

### Agora é com você:

A partir do *script* rodado, e transcrito abaixo:

$x = 5$

$x^2 + 7$

Modifique a segunda linha de comando para o cálculo em “x” utilizando outras operações. Sugestões:

$\sqrt{x}$ ; raiz quadrada

$\log_{10}(x)$ ; logaritmo de base 10

$\sin(x)$ ; seno

## 7.5 ALGUMAS RECOMENDAÇÕES SOBRE A DIGITAÇÃO NUM *SCRIPT* DO *R*

Existem algumas premissas básicas para que um *script* do *R* seja lido de forma clara por seu elaborador, bem como compilado corretamente pelo programa:

## SUMÁRIO

- **Digitação:** sempre que houver um erro no *script* no *RStudio*, surgirá uma cruz vermelha ao lado esquerdo da linha de comando; contornado o erro, o sinal desaparecerá;
- **Comentários:** para que o *script* seja lido também por "um ser humano", é aconselhável incluir comentários nas linhas de comando (iniciados por #) — uma das bases do *Ensino Reprodutível*;
- **Identação:** permita "identação" quando a linha estiver um pouco longa, clicando na tecla *Enter* após uma separação de argumentos por "vírgula". Dessa forma, a linha continua logo abaixo, mas com um pequeno deslocamento à direita. Isso facilita a legibilidade do código.
- **Nomes:** os comandos do *R* são em língua inglesa. Dessa forma, deve-se evitar o uso de variáveis e nomes de arquivos com acentuação ou sinais gráficos em português (exemplo: ç, ~). Além disso, o *R* é um compilador de códigos. Se você definir um nome composto para um arquivo ou variável, ou seja, com espaço entre os termos (como é normal no cotidiano, ex.: meu arquivo), o *R* tentará executar os termos separadamente (ex.: "meu" e depois "arquivo"), o que incorrerá na interrupção da leitura e numa mensagem de erro. Assim, para nomes de variáveis e arquivos, dê preferência a um dos três tipos de **convenções comuns usadas em programação**, a saber:
  1. Separação por *underline*, "\_" ou hífen; exemplo: minha\_variável, minha-variável (convenção *Snake Case*);
  2. Separação por maiúscula; exemplo: minhaVariável (convenção *camelCase*);
  3. Separação por pontos; exemplo: minha.variável;
  4. Sem separação; exemplo: MinhaVariável (convenção *PascalCase*).
- **Observação importante:** no *R*, deve-se evitar a separação por hífen (exemplo: minha-variável; convenção *kebab-case*), pois a linguagem utiliza o hífen como operador de subtração, o que pode causar erros na execução do código.



# 8

**INSTALANDO  
PACOTES  
NO *RSTUDIO***

Objetivos:

1. Entender o que são e para que servem os pacotes do *R*;
2. Saber instalar e carregar um pacote do *R*, exemplificado para *plotly*.

## 8.1 PRA QUÊ INSTALAR PACOTES (E QUE VIRAM BIBLIOTECAS)?

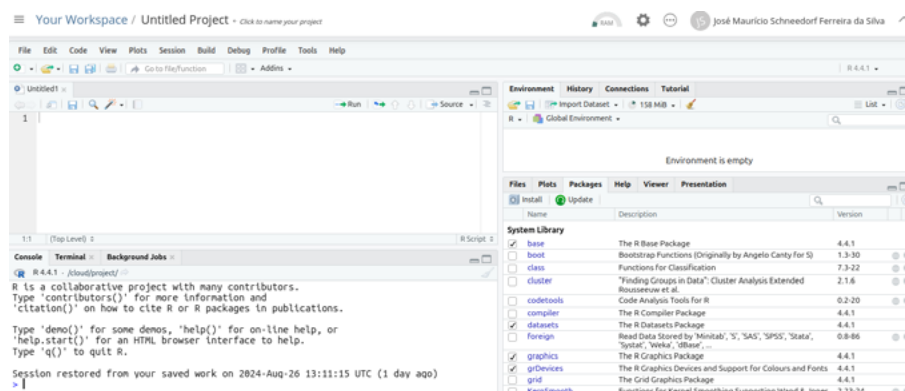
O *R* possui quase 22 mil pacotes atualmente, cada um oferecendo funcionalidades que o *R* original não tem. Ou seja, eles ampliam o programa para atender a necessidades específicas. *Neste material, nosso objetivo com o R é reproduzir e modificar códigos que gerem objetos didáticos interativos para o Ensino Médio.*

Para isso, usaremos principalmente um pacote chamado *plotly* e, raramente, outro chamado *leaflet*, que é voltado para mapas. Esses pacotes de interatividade precisam ser instalados antes de usar os códigos, mas não se preocupe: essa instalação é simples e tranquila.

## 8.2 INSTALANDO O PACOTE *PLOTLY* NO *R*

Acesse a aba *Packages* do *RStudio*.

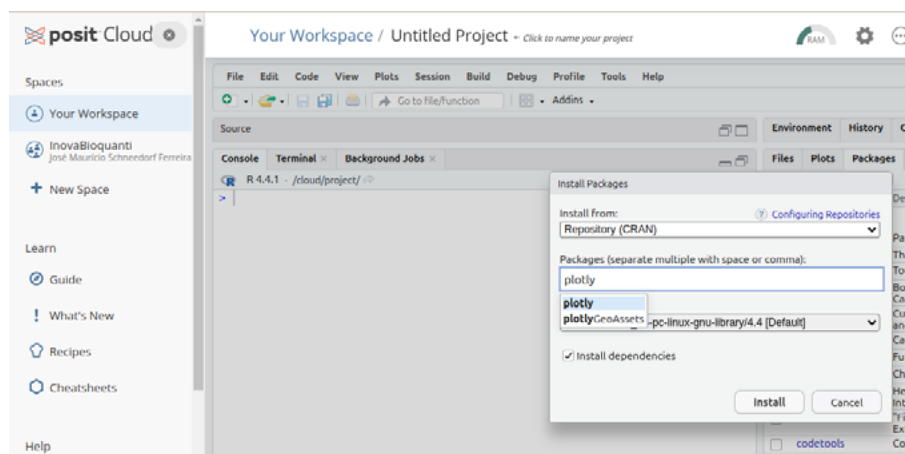
**Figura 33** – Janela e aba do *RStudio* para instalação de pacotes.  
Uma vez instalados, ou seja, descompactados e ativos, convertem-se em bibliotecas do *R*



Fonte: o Autor (2024).

Digite o nome do pacote no campo ("*plotly*"), e clique em *Install*.

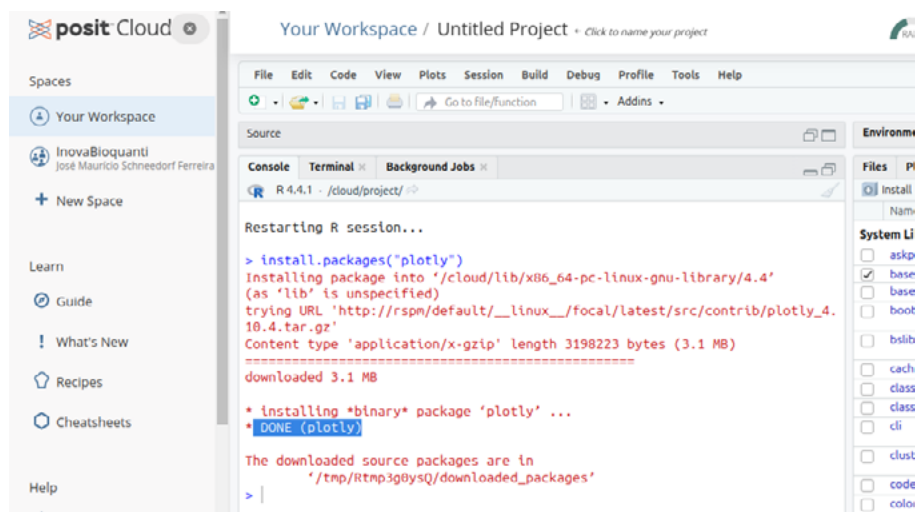
**Figura 34** - Janela do *RStudio* para instalação de pacotes



Fonte: o Autor (2024).

Pronto! O pacote será instalado a partir do servidor de nuvem do *RStudio* (tanto faz se no seu computador ou em nuvem), com algumas mensagens intermediárias em vermelho, como abaixo.

Figura 35 – Mensagens associadas à instalação de pacotes no R



Apesar da coloração em vermelho, essas mensagens são apenas notificações de andamento da instalação. Quando ocorre um erro de fato, a mensagem é bem mais clara, geralmente contendo o aviso: *"Warning... had non-zero exit status"*. Embora essa frase não seja muito intuitiva, ela indica de forma direta que houve um problema na instalação.

Fonte: o Autor (2024).

## 8.3 CARREGANDO O PACOTE INSTALADO

O R permite a instalação de um número enorme de pacotes. No entanto, é importante lembrar que você não utilizará todos eles simultaneamente. Por isso, para indicar qual pacote deseja usar em determinado momento, é necessário carregá-lo com o comando *library*. A seguir, um exemplo para carregar o pacote *plotly*:

```
library(plotly)
```

Pronto! Pacote instalado e carregado!

Agora é possível executar alguns trechos de códigos para objetos didáticos.



# 9

**CONSTRUINDO  
GRÁFICOS  
INTERATIVOS  
COM *PLOTLY***

## SUMÁRIO

### Objetivos:

1. Compreender para que serve o "*plotly*" e seu potencial interativo para ensino e aprendizagem;
2. Utilizar o "*plotly*" por *scripts* para a construção de gráficos interativos;
3. Verificar a interatividade dos gráficos criados.

A biblioteca *plotly* é uma das mais ricas do *R* para gráficos interativos. Permite, entre outros, efeitos de *zoom* no gráfico, bem como de *mouse over*, em que a simples passagem do *mouse* sobre um elemento do gráfico abre as informações daquele ponto. Além disso, permite animações controladas pelo usuário, a inserção de seletores, de controles deslizantes, menus e botões.

Complementarmente, por permitir integração com a linguagem *JavaScript* — muito utilizada atualmente —, a biblioteca *plotly* também é amplamente empregada em painéis de dados, como no *Power Bi* da *Microsoft*, uma suíte de aplicativos conectados para visualização de dados. Na verdade, é a própria biblioteca *plotly.js*, escrita em *JavaScript* e externa ao *R* e ao *RStudio*, que é incorporada dentro do ambiente *R* para permitir essa interatividade avançada.

A elaboração de gráficos com o *plotly* exige apenas alguns comandos simples. A boa notícia é que o gráfico gerado já vem com interatividade embutida, como ampliação/redução (*zoom*), deslocamento dos dados nos eixos, exibição de informações ao passar o *mouse* sobre os pontos, salvamento da imagem em formato *PNG*, entre outros recursos. Para construir qualquer gráfico, é necessário ter *dados*. Basicamente, existem três formas principais de obter esses dados:



- \* Criando se os dados;
- \* Criando se uma equação que vai gerar os dados;
- \* Importando se os dados (de uma planilha, por exemplo)

## 9.1 CRIANDO UM GRÁFICO INTERATIVO

Vamos começar criando os dados a partir de uma equação aplicada a um *vetor*. Para isso, precisamos... do *vetor*! Pense em um vetor como uma coluna (ou linha) do *Excel*. No *R*, os vetores são criados pela *concatenação* de valores separados por *vírgula*, tal como segue.

Mas antes, uma pequena lembrança:

Para acessar todas as imagens interativas deste material envolvendo o *R* e *RStudio*, clique no endereço pertinente ao *website Bioquanti*.

```
x = c(1,2,3,4,5) # um vetor; o "c" indica "concatenação"
# atribui valores de 1 a 5 à variável "x"

# Alternativamente,

x = 1:5 # também atribui valores de 1 a 5 à variável "x"
```

Para elaborar o gráfico interativo, ilustremos a equação de *lançamento vertical* abaixo.

**Figura 36** – Equações de lançamento vertical, para ascensão e queda de um projétil, e exercício correlato

A seguir temos as fórmulas que determinam a altura (H) de um projétil, no movimento de lançamento vertical, em função de sua velocidade inicial ( $V_0$ ) e do tempo (t).

$$H = V_0 t - \frac{1}{2}gt^2 \text{ (Quando a bola sobe, aceleração no sentido contrário ao movimento).}$$

$$H = V_0 t + \frac{1}{2}gt^2 \text{ (Quando a bola desce, aceleração no mesmo sentido do movimento).}$$

Tabela: H x t

t (s)	H (m)
0	
1	
2	20
3	
4	

Fonte: (Pechava, 2014)

Determine os pontos e trace o gráfico da altura em função do tempo.

- e) O gráfico obtido é uma curva denominada?
- f) Quais são as coordenadas do vértice dessa parábola?
- g) Essa função admite valor máximo ou mínimo? O que esse valor representa?

Fonte: MAPA: Ensino Médio - 2º Bim. 1º Ano, p. 27, Matemática e suas Tecnologias.

Agora, vamos criar um gráfico interativo a partir desses dados. Mas antes, claro, é necessário instalar o pacote *plotly* no *R*. Você pode fazer isso pela aba *Packages* do *RStudio*, caso ainda não tenha feito, conforme explicado anteriormente na seção sobre Pacotes.

Etapa final... construir um *gráfico de dispersão de pontos* da função de ascensão vertical.

Para fazer isso, basta copiar o trecho abaixo e colá-lo num novo *script R*. E executá-lo de qualquer das formas mencionadas em seção anterior.

# Dados:

$t = 1:20$  # define o vetor de tempo

$V_0 = 100$  # velocidade inicial, 100 m/s

$g = 9.8$  # aceleração da gravidade,  $m/s^2$

# Equação (ascensão vertical):

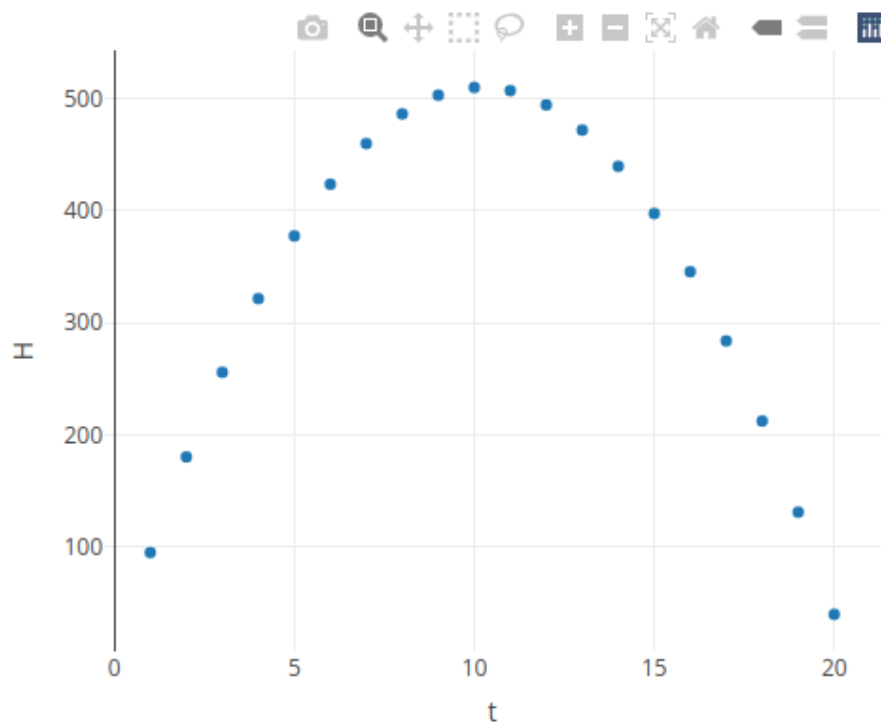
$$H = V_0 * t - 1/2 * g * t^2$$

# Gráfico interativo:

`library(plotly)`

`plot_ly(x = ~t, y = ~H)`

**Gráfico 1** - Gráfico de dispersão de pontos para a equação de lançamento vertical



Fonte: o Autor (2024).

```
# Observação:
# Sintaxe do plotly: ~variável, para atribuir uma variável (x ou y)
# type: para atribuir um tipo de gráfico
```

O *R* costuma apresentar algumas mensagens (*Warnings*) após rodar os comandos. Não são erros, mas informações adicionais, tais como na reprodução do gráfico anterior. Nesse caso, a informação é que está faltando caracterizar o tipo de gráfico, um espalhamento de pontos (*scatter*):

```
plot_ly(x = t, y = H, type = 'scatter')
```

Agora observe quanta interatividade surgiu com o simples comando acima, passando o *mouse* pelos pontos do gráfico, ou clicando-se nos ícones que apareceram acima do gráfico. Teste essa **interatividade**:

1. Passando o mouse sobre os pontos do gráfico (*hover*), você obterá as coordenadas de cada ponto;
2. Usando o botão de rolagem do mouse, você amplia ou reduz o gráfico;
3. Clicando com o botão esquerdo do mouse em qualquer parte do gráfico e desenhando um retângulo, você obterá uma ampliação da área;
4. Se der dois cliques após a ampliação, você retornará ao gráfico original;
5. Posicionando o ponteiro do mouse entre os valores de um eixo e arrastando o mouse, você verá um deslocamento do eixo selecionado;
6. Selecionando um ícone no canto superior direito do gráfico, você poderá, na sequência da esquerda para a direita: baixar o *plot* como imagem, realizar uma ampliação, deslocar os eixos, selecionar os pontos dentro de uma caixa ou dentro de um laço, ampliar, reduzir, escalonar ao tamanho original, realinhar os eixos aos do *plot* original, observar as coordenadas (*x* e *y*), observar somente a coordenada *y* e retornar ao início.

## 9.2 SALVANDO O GRÁFICO

Agora, uma **característica bem interessante do plotly: você pode salvar o gráfico mantendo toda a sua interatividade em um arquivo HTML**. Dessa forma, qualquer pessoa poderá abrir seu gráfico em um *browser* de internet (*Firefox, Chrome, Edge*, por exemplo), o que lhe permitirá observar os detalhes e a ação interativa em qualquer computador, *notebook, tablet* ou *smartphone*!

E para salvar seu **primeiro gráfico interativo**, é muito simples:

1. Após feito o gráfico, clique em “Export”, logo acima do gráfico na aba `Plots`;
2. Clique em “Save As Web Page”;
3. Escolher um nome para o gráfico e salvá-lo.

Agora, basta localizar o arquivo no seu computador, abrir o arquivo automaticamente em um *browser* e verificar que sua interatividade foi mantida. E se desejar compartilhá-lo, basta enviar o arquivo do gráfico interativo para alguém ou exibi-lo em um projetor multimídia.

## 9.3 TRABALHANDO COM RELAÇÕES MATEMÁTICAS NAS VARIÁVEIS

Às vezes, na construção de um gráfico, é interessante poder executar um cálculo em uma variável sem precisar criar um novo vetor. Vamos exemplificar isso para uma transformação isotérmica no estudo de gases (*lei de Boyle-Mariotte*), como segue:

**Figura 37** - Relação de pressão e volume numa transformação isotérmica

**Experimento 1:**

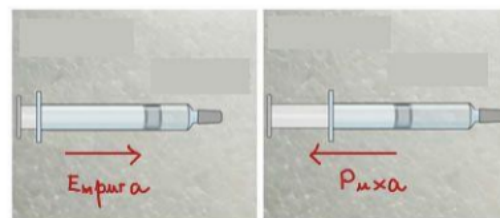
Na transformação isotérmica - observar que à temperatura constante, a pressão varia inversamente com o volume conforme previsto pela lei de Boyle-Mariotte ( $P \cdot V = \text{constante}$ ).

A seringa com o êmbolo sendo empurrado e sendo puxado respectivamente. Primeiramente, a seringa com a tampa e o êmbolo aberto ao máximo, empurra-se o êmbolo devagar. Observe o que

28

irá acontecer e anote nos resultados e discussões. Depois faz-se o oposto, com o êmbolo fechado e com a seringa tampada puxa-se o êmbolo tentando abrí-lo e depois solta. Observe o que aconteceu e anote nos resultados e discussões.

Imagem 01 - Transformação Isotérmica



Fonte: MAPA: Ensino Médio, 2º Bim. 3º Ano, p. 28, Ciências da Natureza e suas Tecnologias.

Exemplificando, se você está trabalhando em uma planilha eletrônica (exemplo: Excel) e deseja construir um gráfico da relação acima, digamos  $V$  versus  $p$ , terá que montar uma coluna com essa operação. No Plotly, assim como no R como um todo, não precisa, já que  $p$  pode ser considerado como  $1/V$  constante. De fato, essa constante é representada pela constante geral dos gases ideais,  $R$ , de valor conhecido.

Resolvendo para a situação acima:

# Dados:

$R = 8.314$  # J/(mol\*K), constante geral dos gases ideais

$V = \text{seq}(0,22,4, \text{length.out}=50)$  # vetor de "Volume" (em litros), com 50 pontos

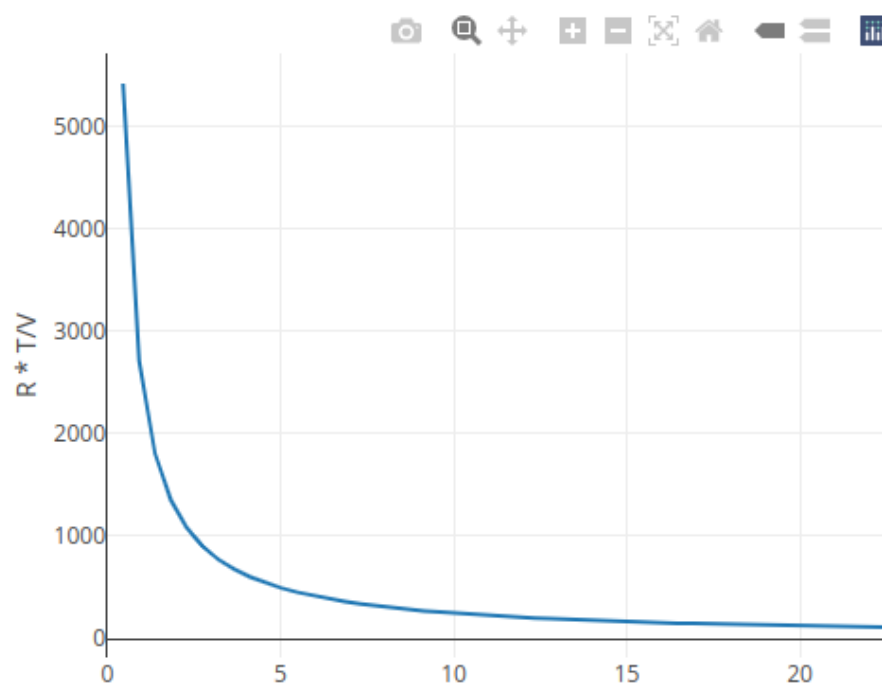
$T = 298$  # K, temperatura absoluta



```
# Equação
# pV = RT; p = RT/V
p = R*T/V

# Gráfico:
library(plotly)
plot_ly(x = V, y = ~R*T/V, type = 'scatter', mode='lines')
```

**Gráfico 2** - Relação de pressão e temperatura pela Lei dos gases ideais



Fonte: o Autor (2024).

Agora, se quiser nomear as *etiquetas dos eixos* e fornecer um *título* ao gráfico para apresentar melhor o significado físico das quantidades envolvidas, basta acrescentar o comando `layout`, como segue:

```
library(plotly)
library(magrittr) # biblioteca para o operador pipe “%>%”
plot_ly(x = V, y = ~R*T/V, type = 'scatter', mode='lines') %>%
layout(
  title = “Transformação Isotérmica de um Gás”,
  xaxis = list(title = “Volume V, L”),
  yaxis = list(title = “Pressão p, bar”)
)
```

**Gráfico 3** - Aplicação de *layout* às etiquetas dos eixos, para a relação de gases ideais



Fonte: o Autor (2024).

### Agora é com você:

Abra um novo *script* e construa um gráfico que apresente uma relação qualquer entre variáveis, tal como sugerido abaixo:

Crie os valores da variável independente (exemplo:  $x = 1:10$ );

Carregue a biblioteca *plotly* - `library(plotly)`;

Digite uma linha geral de comando para o gráfico:  
`plot_ly(x = ~x, y = ~sqrt(x), type = "scatter")`

Selecione essas linhas, dê um Ctrl+Enter, e observe a saída (ou seja, o gráfico, na aba *plots*);

Modifique a variável "y", substituindo o valor de "x" por alguma outra relação, tal como:  $\sim \exp(x)$  - exponencial,  $\sim \sin(10 \cdot x)$  - seno,  $\sim \sqrt{x}$  - raiz quadrada.

## 9.4 OUTROS TIPOS DE GRÁFICOS

Também é possível elaborar outros gráficos, como de *linhas*, *barras*, *histograma* ou *box-plot* ("caixa e bigodes"). Algumas dessas possibilidades são ilustradas abaixo.

### Gráfico 4 - Alguns tipos de gráficos apresentados no Manual de Apoio



Figura 3: Gráfico de barras

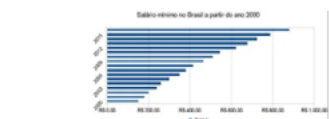


Figura 4: Gráfico de setores



Figura 6: Gráfico de linhas



É importante ressaltar que o gráfico de linhas, algumas das vezes e dependendo da relação entre os dados, as linhas podem se cruzar ou até mesmo se estarem no mesmo ponto ao mesmo tempo.

Figura 7: Gráfico de linhas que se cruzam

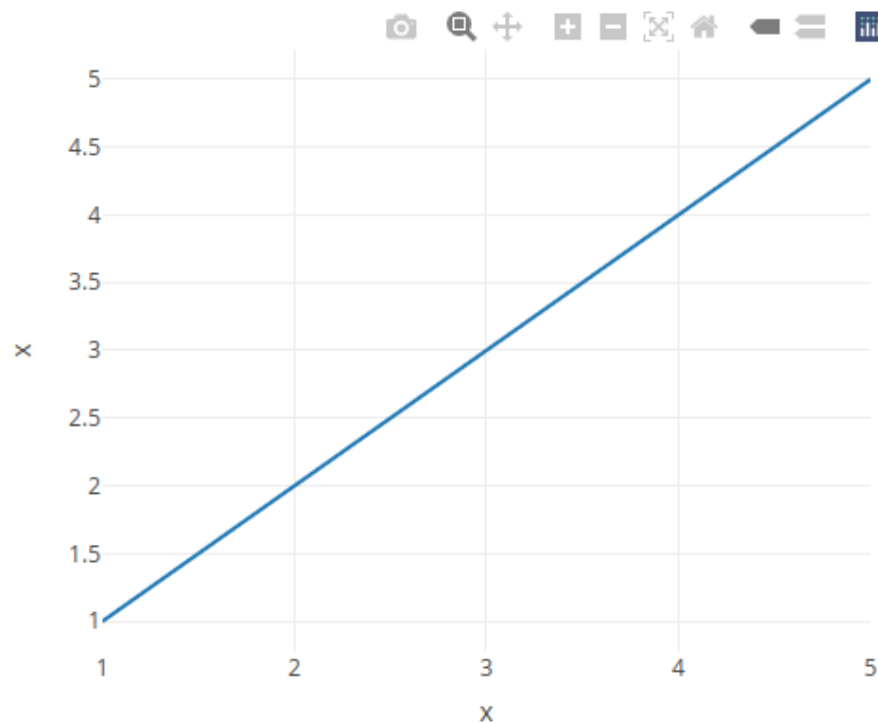


Fonte: MAPA: Ensino Médio. 2º Bim. 3º Ano, p. 9, 10, 22 e 23, Matemática e suas Tecnologias.

No *plotly* há uma gama de gráficos possíveis. Experimente alguns tipos:

```
# Linhas
library(plotly)
plot_ly(x = ~x, y = ~x, type = 'scatter', mode = 'lines')
```

Gráfico 5 - Um gráfico de linha



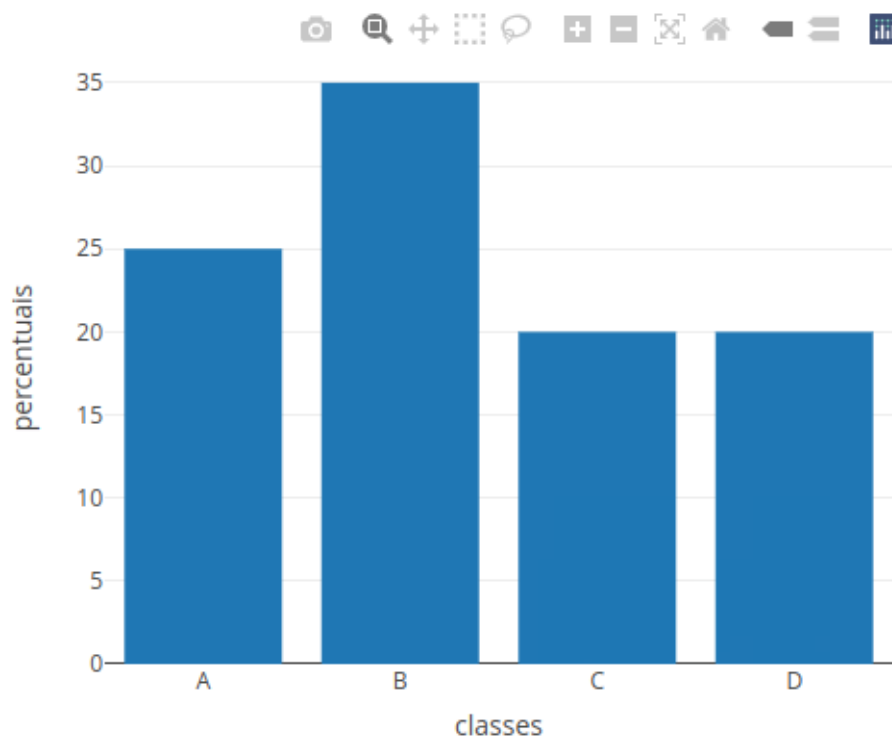
Fonte: o Autor (2024).

```
# Barras
library(plotly)

classes <- c("A", "B", "C", "D") # dados para o gráfico de barras
percentuais <- c(25, 35, 20, 20)

plot_ly(x = ~classes, y = ~percentuais, type = 'bar')
```

Gráfico 6 - Um gráfico de barras



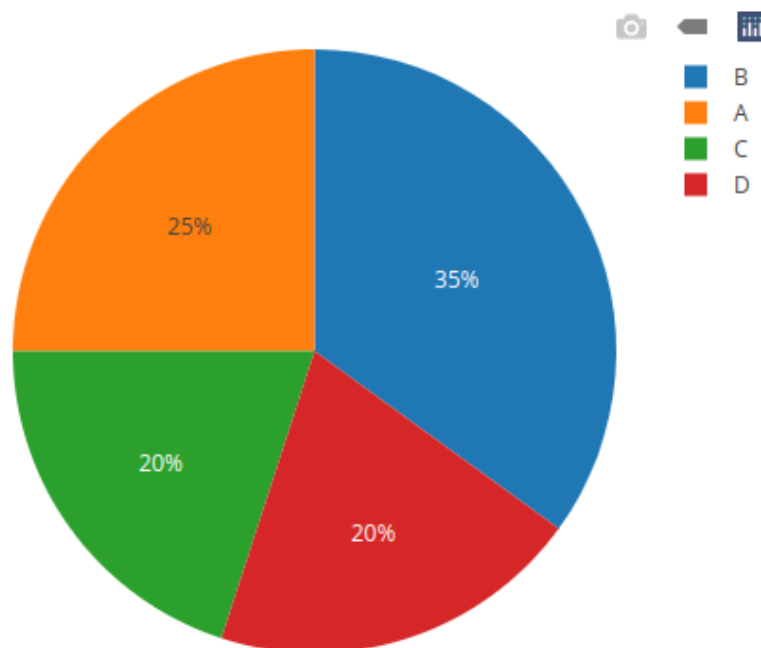
Fonte: o Autor (2024).

## SUMÁRIO

```
library(plotly)
classes <- c("A", "B", "C", "D") # dados para o gráfico de barras
percentuais <- c(25, 35, 20, 20)

# Gráfico de torta
plot_ly(labels = classes, values = percentuais, type = 'pie')
```

Gráfico 7 - Um gráfico de torta



Fonte: o Autor (2024).

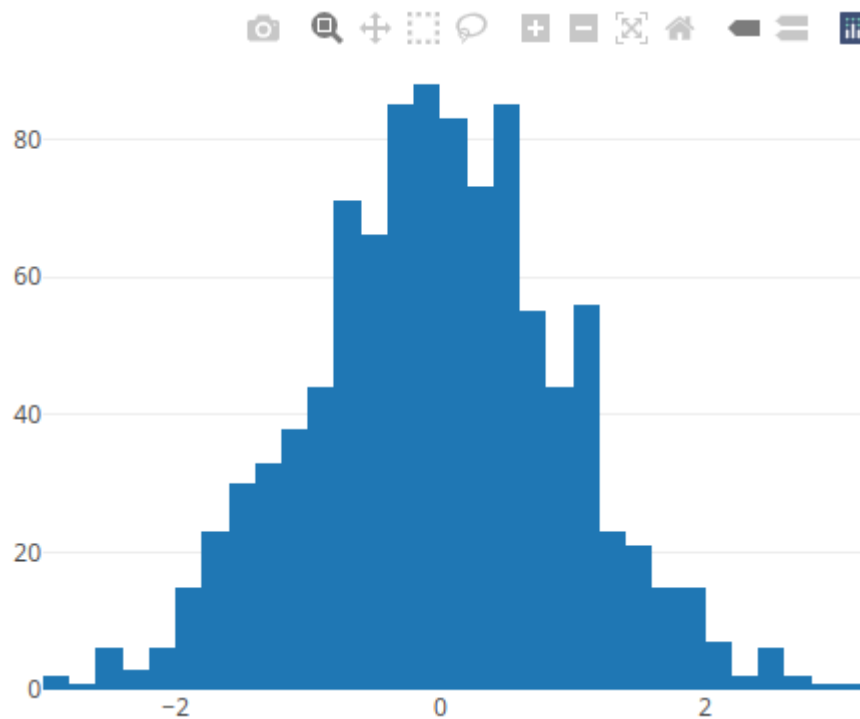
## SUMÁRIO

```
# Histograma  
library(plotly)
```

```
x <- rnorm(1000) # comando pra gerar dados aleatórios no `R`  
plot_ly(x = ~x, type = "histogram")
```



Gráfico 8 - Um gráfico de torta



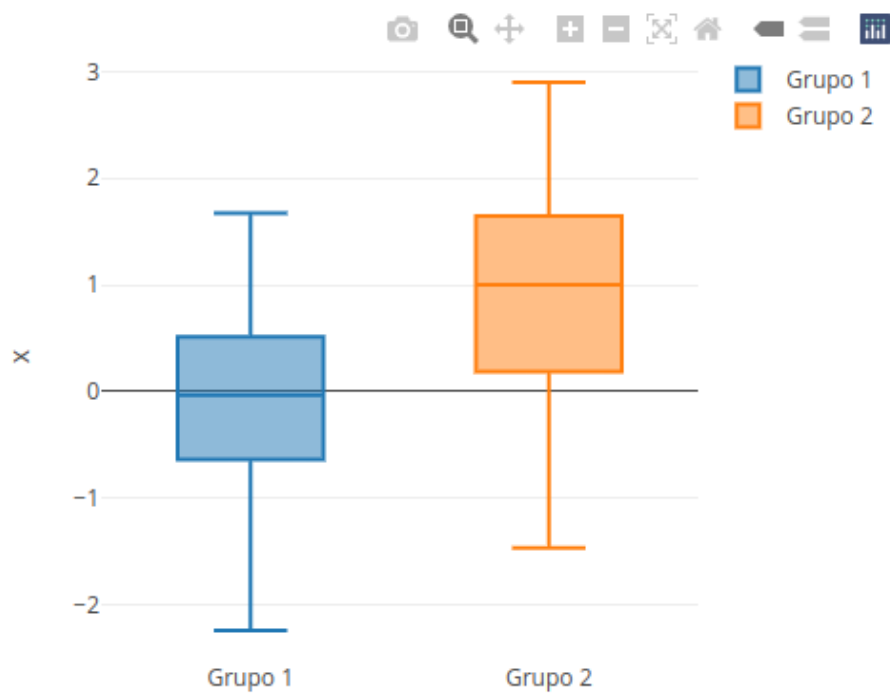
Fonte: o Autor (2024).

## SUMÁRIO

```
# Boxplot
library(plotly)
x <- rnorm(50) # gera dados aleatórios
y <- rnorm(50, mean = 1) # gera a variação estatística nos dados

plot_ly(y = ~x, type = "box", name = 'Grupo 1') %>% # adiciona
os dois boxes para os dados
add_trace(y = ~y, type = "box", name = 'Grupo 2')
```

**Gráfico 9** - Um gráfico de caixa e bigodes (Box-Whisker)

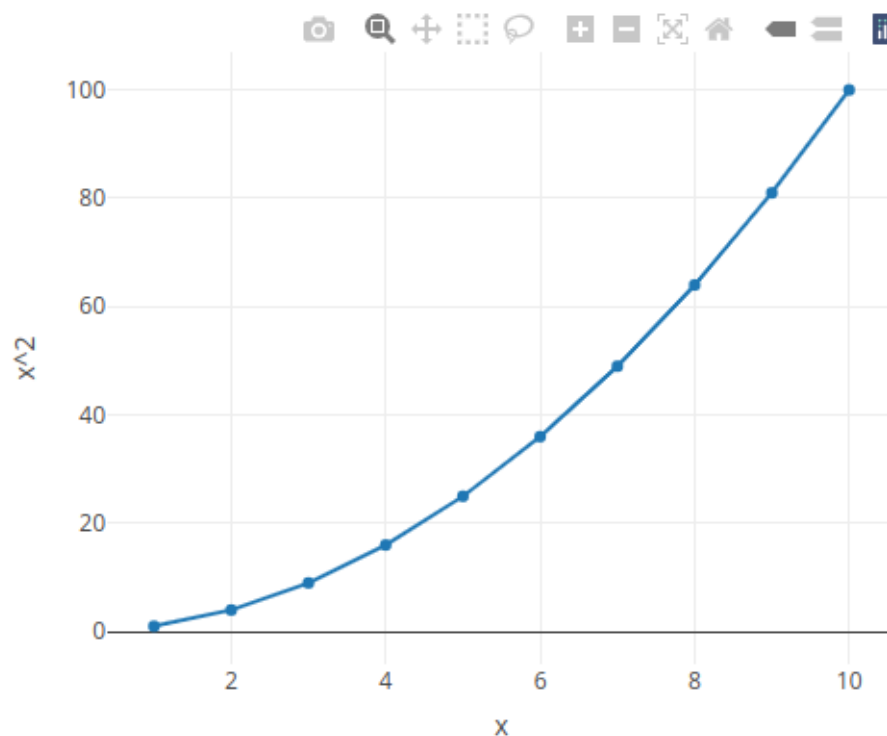


Fonte: o Autor (2024).

Também é possível combinar alguns tipos, como um gráfico de pontos e linhas:

```
# Pontos e linhas
library(plotly)
x <- 1:10
plot_ly(x = ~x, y = ~x^2, type = 'scatter', mode = 'markers,
lines') # também dá se 'markers+lines'
```

Gráfico 10 - Um gráfico de pontos e linhas



Fonte: o Autor (2024).

## SUMÁRIO

### 9.5 GRÁFICOS TRIDIMENSIONAIS (3D)

Para encerrar esta parte, *gráficos tridimensionais*! A versão básica de um gráfico 3D é bem simples de se executar no *plotly*, e seu efeito visual e de interatividade são bem expressivos!

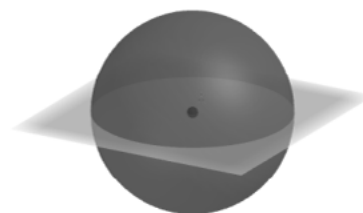
Um exemplo de gráfico 3D pode ser obtido das relações de *raio*, *área superficial* e *volume* de uma esfera, tal como segue.

**Figura 38** - Relações matemáticas de área superficial e de volume de uma esfera em função de seu raio

#### Esfera

Uma esfera é um sólido geométrico resultante da rotação de um semicírculo.

Imagem 8: Esfera



O volume da esfera é:

$$V = \frac{4}{3} \times \pi \times r^3$$

A superfície da esfera é dada por:

$$V = 4 \times \pi \times r^2$$

*Fonte: MAPA: Ensino Médio, 1º Bim. 3º Ano, p. 15, Matemática e suas Tecnologias.*

Agora, vamos elaborar o gráfico 3D interativo no *plotly*. O mais simples seria utilizar as etiquetas com os nomes padrão *x*, *y* e *z*, mas é melhor acrescentar o comando *layout* para atribuir significado físico aos eixos e melhorar a interpretação do gráfico.

```
library(plotly)
```

```
# Dados:
```

```
r = seq(0,100, length.out = 50)
```

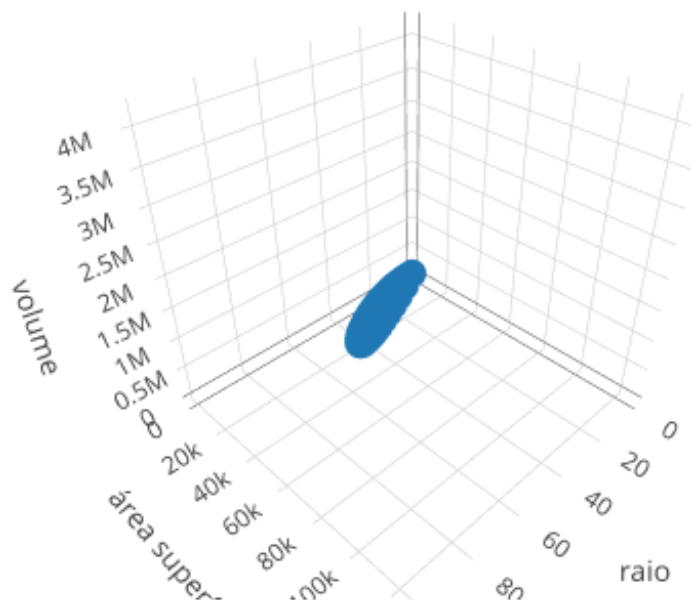
```
AreaSup = 4*pi*r^2 # cálculo de área superficial da esfera
```

```
Volume = 4/3*pi*r^3 # cálculo do volume da esfera
```

```
# Gráfico:
library(plotly)
plot_ly(x = r, y = AreaSup, z = Volume,
        mode = 'markers+lines') %>%
layout(scene = list(
  xaxis = list(title = "raio"),
  yaxis = list(title = "área superficial"),
  zaxis = list(title = "volume")))

```

**Gráfico 11** - Gráfico tridimensional de dispersão de pontos da relação entre área de superfície e volume de uma esfera em função de seu raio



Fonte: o Autor (2024).

Observe que o gráfico é interativo também em outros aspectos, como a possibilidade de *rotação em qualquer um dos três eixos* (note também o ícone de rotação acima do gráfico). Sob o ponto de vista do

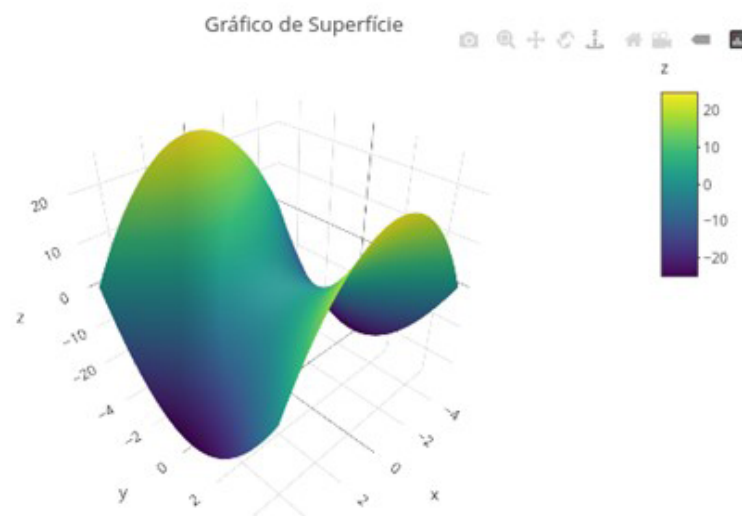
conteúdo, esse gráfico ilustra como os valores de área superficial — e, ainda mais, de volume — variam enormemente em função do raio de uma esfera. Isso também justifica, em parte, o grande sucesso do uso de *nanopartículas* na Ciência e Tecnologia atualmente, devido à alta área de superfície em tamanho reduzido.

Outro gráfico tridimensional interessante é o de *superfície 3D*. Assim como o anterior, poucas linhas de código são necessárias para defini-lo. A diferença é que você pode trabalhar com uma equação para a superfície. Veja:

```
x <- seq(-5, 5, length.out = 50)
y <- seq(-5, 5, length.out = 50)
z <- outer(x, y, function(x, y) x^2 - y^2) # equação x^2+y^2

plot_ly(x = ~x, y = ~y, z = ~z, type = 'surface') %>%
  layout(title = "Gráfico de Superfície")
```

**Gráfico 12** - Um gráfico tridimensional de superfície para relação quadrática



Fonte: o Autor (2024).

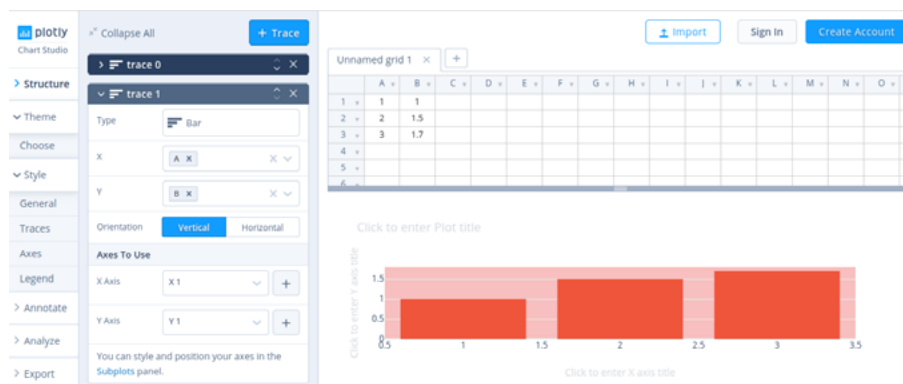
Existem outros tipos de gráficos para o *plotly*, pelo que vale uma visitinha ao website para mais informações.



## 9.6 PLOTLY POR COMANDOS DE MOUSE

Ainda que esse treinamento insista nas vantagens de se utilizar linhas de comando, em vez de cliques de *mouse*, não podemos nos furtar de apresentar uma solução desse tipo para quem prefere o uso do recurso. Entre alguns aplicativos online, destacamos o Plotly Chart Studio, abaixo, que permite a construção de gráficos interativos variados com o pacote.

**Figura 39** - *Plotly Chart Studio*, um recurso para criação de gráficos e mapas interativos com *plotly*, mas com uso de cliques de mouse ao invés de linhas de comando. Para acessar o aplicativo web, acesse <https://chart-studio.plotly.com/create/>



Fonte: Plotly Chart Studio.

## 9.7 REFERÊNCIA DO PACOTE

**Geral** - <https://cran.r-project.org/web/packages/plotly/index.html>

**Manual** - <https://cran.r-project.org/web/packages/plotly/plotly.pdf>

**Tutorial** - <https://plotly.com/r/>

# 10

**MAIS INTERATIVIDADE  
AOS GRÁFICOS**

Objetivos:

1. Observar a capacidade extensiva de interação com o pacote "plotly";
2. Elaborar um gráfico com controle deslizante;
3. Elaborar um gráfico com menu suspenso.

Até o momento só "arranhamos" o potencial de interatividade gráfica do pacote *plotly*. Como já mencionado, essa biblioteca permite um grande conjunto de ações de usuário, como deslizadores (*sliders*), menu de escolha, e botões, entre muitos.

## 10.1 ADICIONANDO UM CONTROLE DESLIZANTE POR INTERVALO

Um *slider* dessa natureza permite que se escolha uma janela de dados para um estudo mais detalhista naquela região. Nesse caso, é possível agregar a um gráfico simples um *controle deslizante de intervalo* (*rangeslider*).

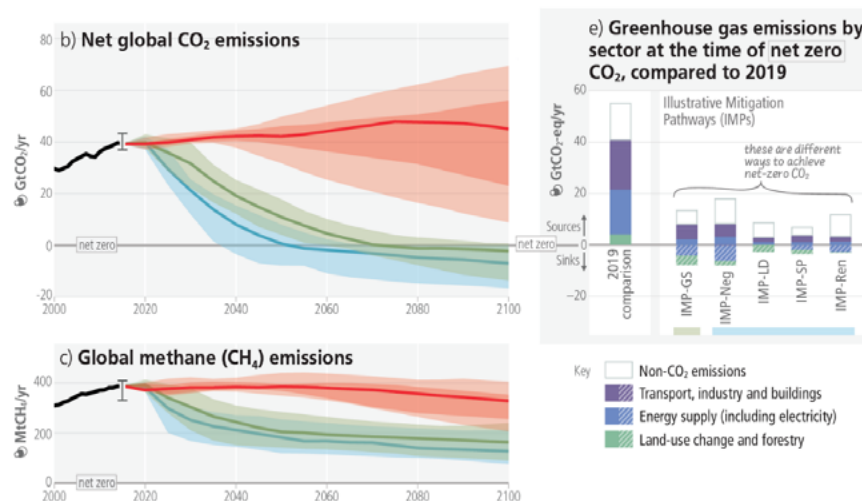
Podemos ilustrar seu emprego por meio da observação de gases de efeito estufa, em especial da *emissão de dióxido de carbono no Brasil*, a partir de uma base de dados da internet. Para isso, você aprenderá a *obter um arquivo a partir dessa base de dados, filtrar para um subconjunto desejado e elaborar o gráfico resultante, com um controle deslizante adicional*.

## 10.2. EMISSÃO DE CO<sub>2</sub> E O EFEITO ESTUFA

As emissões de CO<sub>2</sub> e outros gases pela queima de combustíveis fósseis têm grande responsabilidade sobre o efeito estufa, incidindo diretamente nas alterações climáticas. Para reduzir essas emissões, é necessário transformar a matriz energética atual, a indústria e os sistemas alimentares.

Para compreender a emissão de CO<sub>2</sub> observada no Brasil no período de 1890 a 2022, execute o trecho de código que segue em um *script* do R (ou seja, copie, cole e execute), a partir da fonte Our World in Data.

**Gráfico 13 - Uma abordagem sobre gases do efeito estufa**



Fonte: MAPA: Ensino Médio, 1º Bim. 2º Ano, p. 19, - Ciências Humanas e suas Tecnologias.

## SUMÁRIO

```
library(readr) # biblioteca de importação de dados
library(dplyr) # biblioteca para uso do operador pipe "%>%"
library(plotly)

# Carregamento dos dados da internet
url <- "https://raw.githubusercontent.com/owid/co2-data/master/owid-co2-data.csv"
co2_data <- read_csv(url)

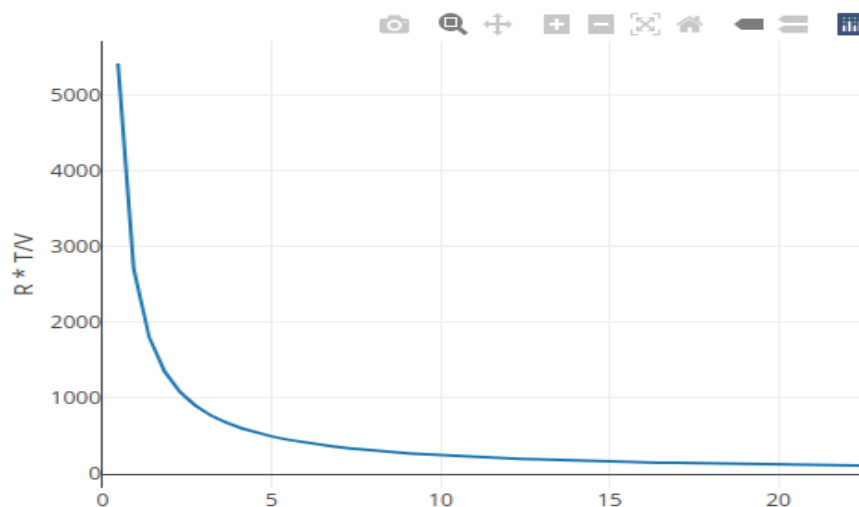
# Filtragem dos dados para o Brasil usando subset()
co2_brasil <- subset(co2_data, country == "Brazil")
```

```
# Criação do gráfico interativo com plot_ly
co2_plot <- plot_ly(data = co2_brasil, x = ~year, y = ~co2, type =
'scatter', mode = 'lines+markers') %>%
layout(title = "Emissões de CO2 no Brasil ao longo dos anos",
xaxis = list(title = "Ano"),
yaxis = list(title = "Emissão de CO2 (milhões de toneladas)"))
```

Agora, a cereja do bolo: a inserção de um controle deslizante para a seleção de faixas, permitindo um estudo mais focado.

```
co2_plot %>%
  rangelsider()
```

**Gráfico 14** - Um gráfico de *plotly* apresentando um controle deslizante (*slider*) para visualização destacada de período para emissões de CO<sub>2</sub> no Brasil ao longo dos anos



Fonte: o Autor (2024).

Você pode copiar e colar os *scripts* na sequência para execução, ou apenas adicionar o comando *rangelsider()* com o operador **pipe** %>% ao final.

Experimente agora posicionar o *mouse* em um dos dois marcadores laterais do gráfico inferior, arrastando-o em seguida, e observe o resultado. O controle deslizante pode ser útil quando se deseja focar em determinada região do gráfico, por exemplo, ajustar a emissão de CO<sub>2</sub> para os últimos anos.

### 10.2.1. ADICIONANDO UM MENU SUSPENSO

Menus suspensos (*dropdown menu*) permitem observar um gráfico diferente a cada opção selecionada. Para exemplificar esse recurso interativo, vamos primeiramente elaborar um conjunto de dados (*dataframe*) que contenha a resposta linear, quadrática e cúbica a uma variável independente, conforme segue:

```
x = 1:10 # vetor da variável independente "x"
yLin = x
yQuad = x^2 # criação da variável dependente quadrática
yCub = x^3 # criação da variável dependente cúbica
datLQC <- data.frame(x,yLin,yQuad,yCub) # criação da planilha de dados
```

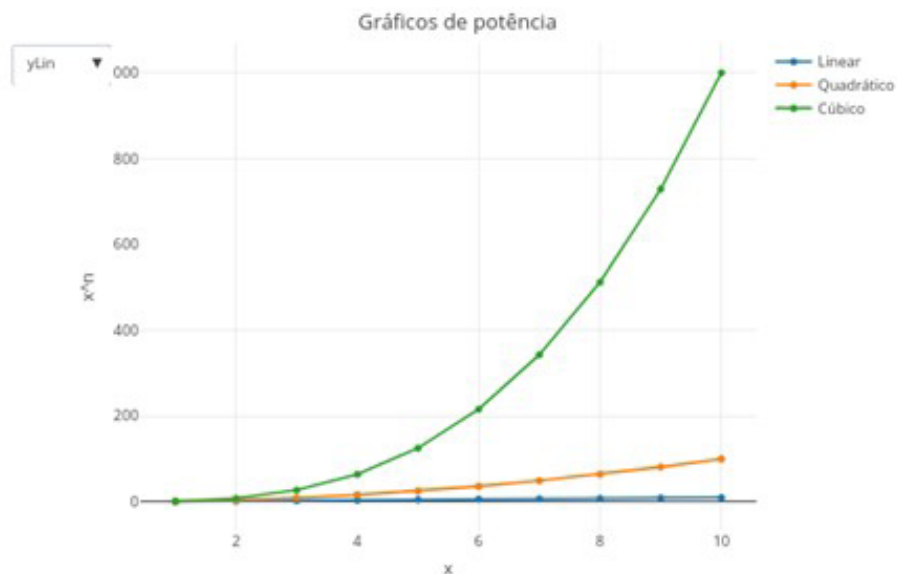
Agora podemos inserir o *menu suspenso* para opção das tendências matemáticas:

```
plot_ly(datLQC, x = ~x, y = ~yLin, type = "scatter", mode =
"line", name = "Linear") %>%
add_trace(x = ~x, y = ~yQuad, mode = "line", name =
"Quadrático") %>%
add_trace(x = ~x, y = ~yCub, mode = "line", name = "Cúbico")
%>%
layout(
title = "Gráficos de potência",
xaxis = list(title = "x"),
yaxis = list(title = "x^n"),
updatemenus = list(
```



```
list(
  buttons = list(
    list(label = "yLin", method = "update", args = list(list(visible =
c(TRUE, FALSE, FALSE)))),
    list(label = "yQuad", method = "update", args = list(list(visible =
c(FALSE, TRUE, FALSE)))),
    list(label = "yCub", method = "update", args = list(list(visible =
c(FALSE, FALSE, TRUE))))
  )
)
```

**Gráfico 15** - Menu suspenso no canto superior direito do gráfico, apresentando as opções para visualização exclusiva de ajustes distintos (linear, quadrático e cúbico)



Fonte: o Autor (2024).

Ainda que você possa achar meio complicado o trecho de código acima, apenas copie-o, cole-o num *script* e execute-o. Isso exemplifica a

*essência inerente ao Ensino Reprodutível, desde a simples reprodução do código até sua alteração e, mesmo, a criação de novos.* Sentindo curiosidade, você pode alterar alguns termos do código acima, como as etiquetas (*label* — substitua um nome, por exemplo) que surgem no menu suspenso, o tipo de gráfico pretendido (substitua *scatter* por *bar*, por exemplo) ou o título do gráfico (*title*).

Em relação à interatividade produzida, adiciona-se, às que já estavam presentes pelo comando *plot\_ly*, a seleção do tipo de potência a ser representada pelo menu suspenso.

Assim como para vários pacotes do *R*, existe um número significativo de comandos e widgets interativos com o *Plotly*, que, nesse caso específico, daria “pano pra manga” para uma obra literária isolada. Mas você pode consultar inúmeros sites sobre o *Plotly* para um aprendizado mais abrangente — os *links* abaixo — e até mesmo um livro online gratuito, com códigos e gráficos correlatos. Para observar a imensa riqueza de gráficos interativos, dê uma olhada no website do *Plotly* para o *R*.

**Geral** - <https://cran.r-project.org/web/packages/plotly/index.html>

**Manual** - <https://cran.r-project.org/web/packages/plotly/plotly.pdf>

**Tutorial** - <https://plotly.com/r/>

# 11

**ANIMAÇÃO  
EM GRÁFICOS  
INTERATIVOS**

Objetivos:

1. Conhecer o potencial do "*plotly*" para criar animações interativas;
2. Elaborar gráficos animados por importação de banco de dados.

Além do aspecto puramente interativo dos gráficos elaborados com o *Plotly* — o que já representa um grande diferencial na preparação de materiais ilustrativos para conteúdos didáticos —, a biblioteca ainda é capaz de rodar animações com os gráficos!

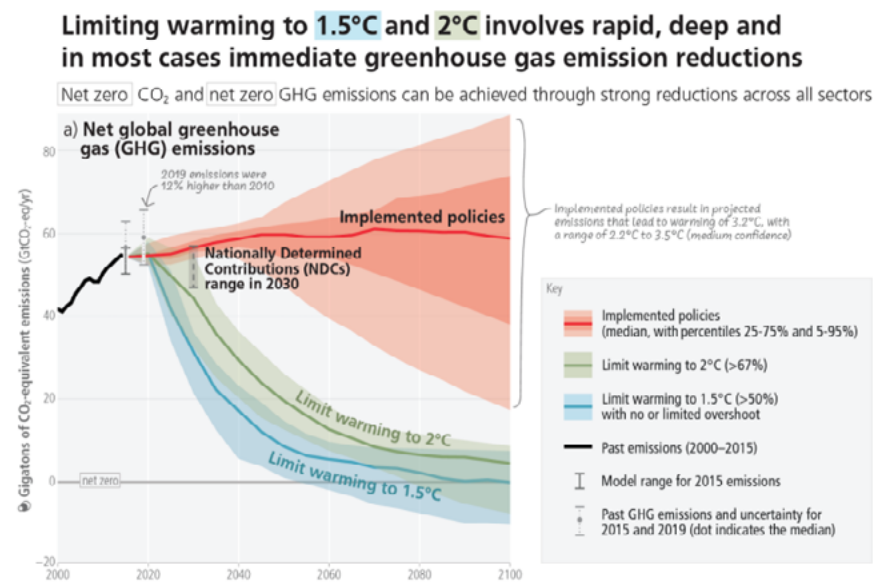
A animação ocorre por meio da transição de uma imagem para outra em um gráfico, quando se deseja observar o que acontece ao alterar uma variável (numérica ou categórica). O comando-chave para isso é *frame* (quadro). A animação no *Plotly* também serve para os três tipos de entrada de dados, ou seja: *equações*, *vetores* e *datasets importados*.

## 11.1 ELEVAÇÃO DA TEMPERATURA MÉDIA DA TERRA

Desnecessário mencionar os impactos recentes das mudanças climáticas no planeta em função da ação humana, incluindo uma elevação média da temperatura superficial do globo terrestre, em decorrência de fatores como o efeito estufa.

**Gráfico 16 - Mudanças climáticas do planeta em função da emissão de gases e da elevação média de temperatura**

Trajetórias de emissões globais consistentes com políticas implementadas e estratégias de mitigação



Fonte: MAPA: Ensino Médio, 1º Bim. 2º Ano, p. 99, Ciências humanas e suas Tecnologias.

## SUMÁRIO

library(plotly)

library(magrittr) # bibliotecas necessárias

# 1) Obtendo os dados da internet

```
url <- "https://raw.githubusercontent.com/datasets/global-temp/refs/heads/main/data/annual.csv" # define o link para os dados
```

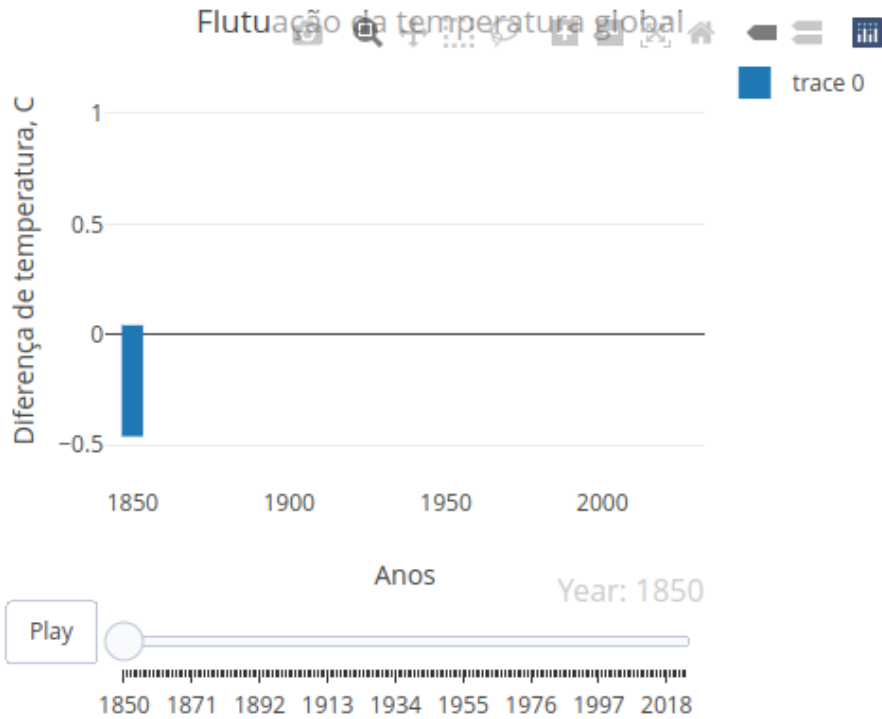
```
dados <- read.csv(url) # lê o arquivo dos dados
```

# 2) Construindo o gráfico com animação

```
plot_ly(dados, x = ~Year, y = ~Mean,
        type = "bar",
        marker = list(line = list(width = 10)),
        frame = ~Year) %>%
```

```
animation_opts(  
  frame = 150, # Velocidade da animação  
  transition = 0,  
  redraw = TRUE  
) %>%  
layout(  
  title = "Flutuação da temperatura global",  
  xaxis = list(title = "Anos"),  
  yaxis = list(title = "Diferença de temperatura, C"))
```

**Gráfico 17** - Flutuação da temperatura global com inserção de controle deslizante animado. Ao clicar-se no botão *PLAY*, o gráfico exibirá a variação da temperatura ao longo dos anos por animação da barra em azul



Fonte: o Autor (2024).



Para observar melhor o gráfico, clique em *Zoom*, que abrirá uma janela maior. Agora, a parte “*chique*”: clique em *PLAY* e veja o que acontece. Você também pode selecionar qualquer período para a emissão, bastando usar a barra de rolagem do gráfico.

## 11.2 EXPECTATIVA DE VIDA E PRODUTO INTERNO BRUTO

Um emprego bem interessante para o uso do *plot\_ly* em animação gráfica ocorre quando precisamos apresentar vários dados sobre determinado tema. A isso dá-se o nome de *dados multivariados*. Ilustrando essa situação, digamos que se deseje oferecer informações variadas em um gráfico que envolva a relação entre o produto interno bruto de um país e a expectativa de vida de seus habitantes ao longo do tempo.

**Tabela 1** - Alguns índices econômicos associados à expectativa de vida em 2001

País	Posição em 2001	IDH-99	População (milhões, 1999)	PIB “per capita” (PPP US\$, 1999)
Noruega	1	0,939	4,4	28.433
EUA	6	0,934	280,4	31.872
Argentina	34	0,842	36,6	12.277
Brasil	69	0,750	168,2	7.037
Armênia	72	0,745	3,8	2.215
Geórgia	76	0,742	5,3	2.431
Bolívia	104	0,648	8,1	2.355
Serra Leoa	162	0,258	4,3	448

Fonte: MAPA: Ensino Médio, 1º Bim. 2º Ano, p. 101, Ciências humanas e suas Tecnologias.

**Tabela 2** - Alguns índices econômicos associados à expectativa de vida em 2001

Expectativa de vida ao nascer (anos, 1999)	Gastos públicos com saúde (% PIB, 1998)	Mortalidade Infantil (por mil nascidos vivos, 1999)
78,4	7,4	4
76,8	5,8	7
73,2	4,9	19
<b>67,5</b>	<b>2,9</b>	<b>34</b>
72,7	3,1	25
73,0	0,5	19
62,0	4,1	64
38,3	0,9	182

Fonte: MAPA: Ensino Médio, 1º Bim. 2º Ano, p. 101, Ciências humanas e suas Tecnologias.

Para ilustrar a riqueza interativa que se pode obter com o *Plotly* sobre a influência do *Produto Interno Bruto (PIB)* na expectativa de vida, podemos importar um conjunto de dados da internet e criar um gráfico sobre essa relação, como segue:

```
library(plotly)
```

```
# Obtendo os dados na internet
```

```
url <- read.csv("https://raw.githubusercontent.com/kirenz/  
datasets/refs/heads/master/gapminder.csv")
```

```
dadosExpVida <- url # atribuindo os dados a um objeto do `R`
```

```
# Criando o gráfico interativo
```

```
plot_ly(
```

```
  dadosExpVida, # dados convertidos da internet
```

```
  x = ~gdpPercap, # nome da coluna de renda per capita nos dados
```

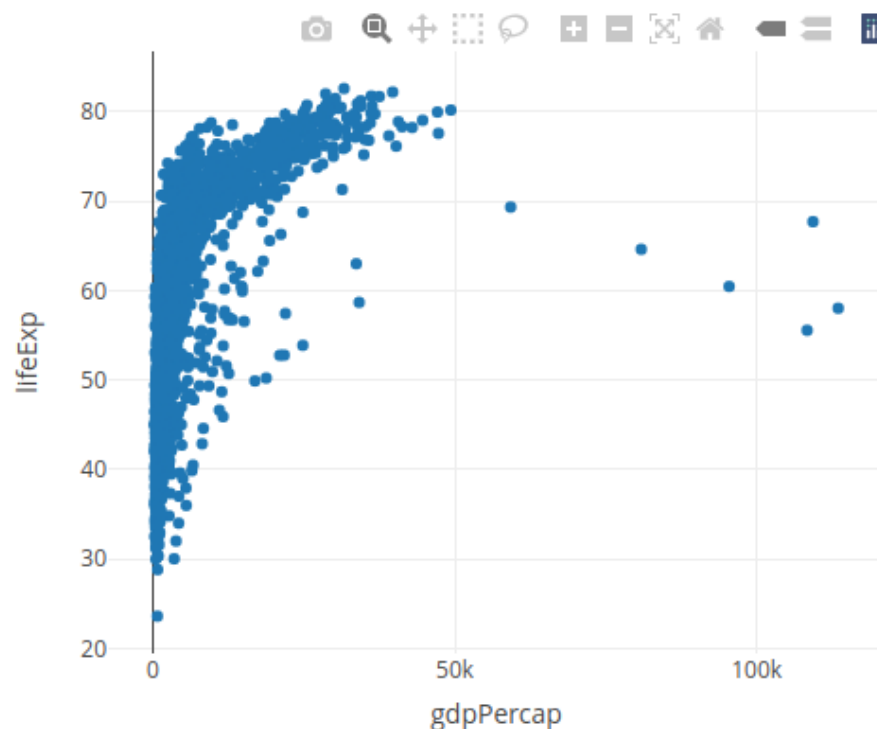
```
  y = ~lifeExp, # nome da coluna de expectativa de vida nos dados
```

```
  type = 'scatter', # tipo de gráfico (espalhamento)
```

```
  mode = 'markers' # tipo de espalhamento (pontos)
```

```
)
```

**Gráfico 18** - Representação da expectativa de vida em relação ao produto interno bruto de vários países entre 1950 e 2012



Fonte: o Autor (2024).

Pronto! Código simples, direto e interativo. Ao reproduzir esse gráfico, basta passar o mouse pelos pontos para observar as coordenadas de PIB e expectativa de vida. Porém, não é possível identificar, por meio deste gráfico, qual país corresponde a cada ponto, nem acessar outras informações que constam na planilha original baixada da internet. Para se ter uma ideia, essas tabelas possuem, além dos valores de PIB e expectativa de vida, o nome do país, sua população, o continente ao qual pertence, bem como o ano em que os dados foram medidos.

Dessa forma, estamos diante de um quadro de *dados multivariados*, muito comum em bases de dados diversas, como IBGE ou DATASUS.

Que tal se pudéssemos apresentar tudo de uma só vez, ou seja: PIB, expectativa de vida, país, tamanho da população, continente e o ano em que esses dados foram medidos — ou seja, **seis** variáveis, entre *numéricas* e *categóricas* (classes)?

Impossível?! Não para o *R*! Segue um trecho de código para isso, junto com o resultado proposto. Não se preocupe com o tamanho ou os detalhes. Se quiser **reproduzir** esse código, já sabe: basta copiar, colar e executar o *script* no *R*.

```
library(plotly)

# Obtendo os dados na internet
url <- read.csv("https://raw.githubusercontent.com/kirenz/
datasets/refs/heads/master/gapminder.csv")

dadosExpVida <- url # atribuindo os dados a um objeto do `R`

# Criando o gráfico interativo com animação
plot_ly(
  dadosExpVida, # dados convertidos da internet
  x = ~gdpPercap, # renda per capita
  y = ~lifeExp, # expectativa de vida
  size = ~pop, # tamanho dos pontos em função da população
  color = ~country, # cor dos pontos em função do país
  frame = ~year, # Frame para a animação por ano de coleta dos
  dados
  text = ~continent, # País como informação ao passar o mouse
  hoverinfo = "text",
  type = "scatter", # tipo de gráfico
  mode = "markers",
  marker = list(sizemode = "diameter", opacity = 0.7)
) %>%
layout( # atribuição de título e etiquetas dos eixos
  title = "Produto interno bruto X Expectativa de vida",
  xaxis = list(title = "PIB (log), US$", type = "log"),
  yaxis = list(title = "Expectativa de Vida, anos"),
  showlegend = TRUE # possibilidade ou não de aparecer a
  legenda
```

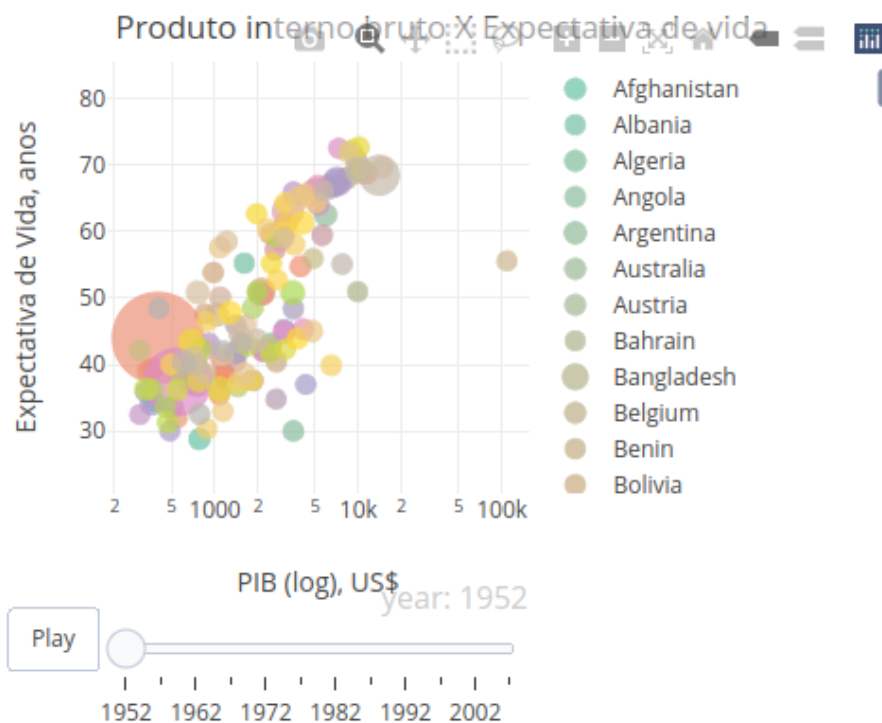
## SUMÁRIO

```
) %>%
  animation_opts(
    frame = 1000, # Velocidade da animação
    transition = 0,
    redraw = TRUE
  )
```

Para ir ao "playground" agora, tecele em *PLAY* e observe a transição temporal da expectativa de vida em função do *PIB* dos países. Veja que todos os demais dados estão lá, representados pelo tamanho dos pontos (população), cor (país), continente (*hover*, passagem do *mouse*) e ano (quadro de animação ou *frame*)!

#### Gráfico 19 - Expectativa de vida em relação a produto interno bruto.

Apesar de utilizar os mesmos dados do Gráfico 18, a representação atual separa a informação por cores (país), texto (continente), tamanho do símbolo (população), e por ano da medida (*slider* de tempo com animação facultativa)



Fonte: o Autor (2024).

Mais um detalhe! Se você observar a legenda, verá que ela também é *deslizante*, podendo ser movimentada com o arraste do mouse, e identifica cada país por uma cor. Quer saber onde está o Brasil nessa relação entre PIB e expectativa de vida do gráfico doidão? Fácil: ache o Brasil na legenda, dê dois cliques e observe que agora somente esse ponto estará destacado.

## SUMÁRIO



# 12

**MAPAS INTERATIVOS  
COM *PLOTLY***

#### Objetivos:

1. Conhecer o potencial do "plotly" para criação de mapas interativos;
2. Elaborar um mapa interativo com dados inseridos;
3. Elaborar um mapa interativo com dados importados.

Por fim, deixamos a última parte deste treinamento com a plataforma *R* e *RStudio* para apresentar outro potencial "pra lá de bacana" do *Plotly* para ensino e aprendizagem: a construção de mapas interativos.

Mapas interativos permitem que se observe informações por meio do *mouse hover* (passagem do *mouse* sobre os dados) ou por clique do *mouse* em um mapa contendo esses dados. Por se tratar de um mapa, as informações são obtidas em coordenadas geográficas específicas. Essa característica torna indispensáveis, portanto, as *coordenadas de longitude e latitude* relacionadas aos pontos geográficos que se deseja apresentar.

Segue um exemplo simples, localizando os três municípios do Sul de Minas Gerais onde ficam os *campi* da Universidade Federal de Alfenas (UNIFAL-MG).

```
library(plotly)
```

```
# Criar dados de exemplo com coordenadas de algumas cidades
cities <- data.frame(
  name = c("Alfenas", "Varginha", "Poços de Caldas"),
  lat = c(-21.42943530, -21.539957, -21.783731), # Latitude
  lon = c(-45.95948212, -45.433960, -46.564178) # Longitude
)
```

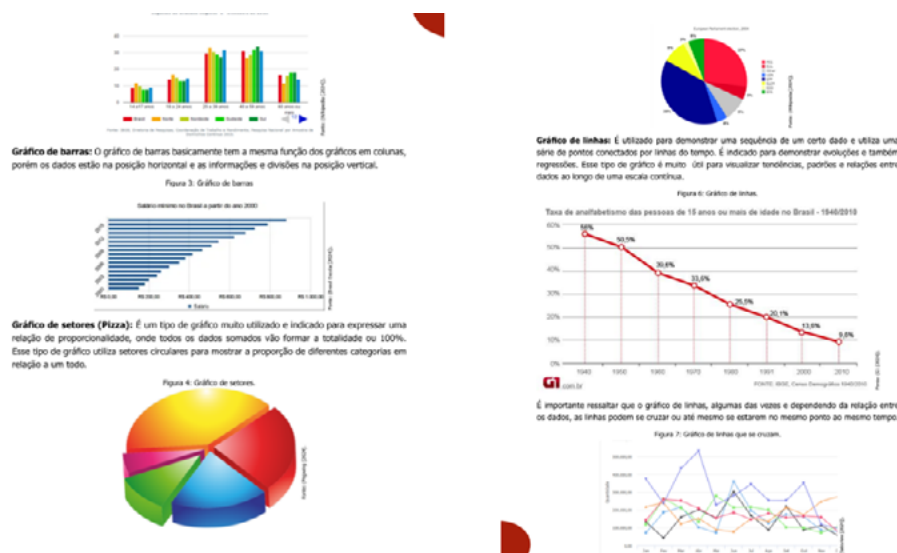
```
# Criar o mapa interativo
```

```
plot_ly(
  data = cities,
  lat = ~lat,
  lon = ~lon,
  type = 'scattergeo',
```

## SUMÁRIO

```
mode = 'markers+text',
text = ~name,
marker = list(size = 10, color = 'blue', opacity = 0.7),
textposition = "top center"
) %>%
layout(
title = "Municípios dos campi da UNIFAL-MG",
geo = list(
scope = 'south america',
showland = TRUE
)
)
```

**Figura 40** - Mapa da América do Sul apresentando as coordenadas dos municípios em que a UNIFAL-MG mantém campus: Alfenas, Varginha e Poços de Caldas, todos ao sul de Minas Gerais. Meio atropelados, diga-se de passagem, em função do nível de ampliação da imagem (vide o site Bioquanti para a versão interativa)



Fonte: o Autor (2024).

Para observar os municípios, vá ampliando a imagem com o *mouse*. Veja que o mapa inicia na América do Sul, uma condição inserida no código para facilitar a localização das cidades. Experimente colocar um comentário (#) à esquerda do trecho `"scope = ..."`, e a visualização passará a partir do *mapa-múndi*. Observe agora que, ao passar o *mouse* sobre os municípios, são exibidas as coordenadas geográficas correspondentes.

### Agora é com você:

Localize as coordenadas geográficas (longitude e latitude) de sua cidade natal ou da de um ente querido. Para isso, faça uma busca na internet;

Copie o código acima e cole-o num novo *script*;

Substitua os atributos `"name"`, `"lat"`, e `"lon"` no `data.frame` do código pelos que você buscou;

Rode o *script* e observe no mapa interativo o município escolhido. Dica: se você selecionou uma cidade fora da América do Sul, colocar um `"#"` antes da linha de `"scope = ..."`.

## SUMÁRIO

### 12.1 PRODUÇÃO MUNDIAL DE ÓLEOS

Agora imagine que você possa, em vez de inserir os dados um a um, *importar informações de algum repositório da internet* para a construção do mapa — como foi feito na etapa anterior. Para ilustrar isso, vamos importar uma planilha referente à produção de óleo no planeta. Essa categoria inclui petróleo bruto, óleo de xisto, areias betuminosas, condensados e líquidos de gás natural (etano, GLP e nafta separados da produção de gás natural).

Ao mesmo tempo, vamos filtrar os dados importados para o ano de 2014, tal como encontrado no banco de dados do *Our World In Data*.

Figura 41 - Produção mundial de petróleo



Fonte: MAPA: Ensino Médio, 3º Bim. 2º Ano, p. 81, Ciências Humanas e suas Tecnologias.

## SUMÁRIO

library(plotly)

# Exemplo de dataframe com valores fictícios

```
df <- read.csv("https://raw.githubusercontent.com/owid/owid-datasets/refs/heads/master/datasets/Oil%20production%20-%20Etemad%20%26%20Luciana/Oil%20production%20-%20Etemad%20%26%20Luciana.csv")
```

```
# Renomeando as colunas para facilitar interpretação e plotagem
names(df) <- c("País", "Ano", "Produção.TeraWatt")
```



```
# Filtrando os dados para o último ano (2014)
```

```
df <- subset(df, Ano == "2014")
```

```
head(df)
```

```
País Ano Produção.TeraWatt
82 Albania 2014 12.10759
165 Algeria 2014 1014.64045
224 Angola 2014 1035.41999
331 Argentina 2014 421.63374
351 Aruba 2014 1.65714
402 Australia 2014 281.78225
```

```
# Criando o mapa choropleth com a escala de cores ajustada
```

```
library(plotly)
```

```
plot_ly(
```

```
  data = df,
```

```
  locations = ~País,
```

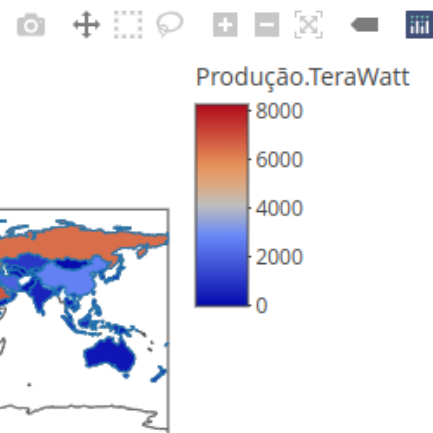
```
  locationmode = "country names",
```

```
  z = ~Produção.TeraWatt, # Variável que determina as cores
```

```
  type = "choropleth",
```

```
  colorscale = "RdBu") # outras escalas: # outras escalas: Viridis,  
Inferno, Blues, Cividis, Greens, ...
```

**Figura 42 - Produção mundial de petróleo**



Fonte: MAPA: Ensino Médio, 3º Bim. 2º Ano, p. 81, Ciências Humanas e suas Tecnologias.



Seguindo a mesma lógica do mapa anterior, ao passar o *mouse* sobre os países, você verá o consumo identificado em cada um deles. Observe que há uma barra lateral com a legenda indicando o quantitativo de produção.

Note também que o *type* do gráfico agora é *choropleth* (e não *scattergeo*). Em sua versão mais simples, esse tipo de gráfico exige apenas os nomes padronizados dos países. No entanto, ele também pode ser construído a partir de um banco de dados que contenha apenas as coordenadas geográficas de latitude e longitude.

Para auxiliar nesse processo, seguem dois *links* práticos para arquivos de coordenadas geográficas, acessíveis pelo *R*:

**Coordenadas de municípios brasileiros** - <https://raw.githubusercontent.com/kelvins/municipios-brasileiros/refs/heads/main/csv/municipios.csv>

**Coordenadas de capitais mundiais** - [https://raw.githubusercontent.com/bahar/WorldCityLocations/refs/heads/master/World\\_Cities\\_Location\\_table.csv](https://raw.githubusercontent.com/bahar/WorldCityLocations/refs/heads/master/World_Cities_Location_table.csv)

E, nesse caso, qualquer banco de dados nesse formato será bem-vindo. Isso significa, na prática, a possibilidade de exemplificar qualquer informação de conteúdo didático de forma interativa em um mapa — como produção ou exportação de insumos, observações clínicas, marcos históricos, entre outros.

## UMA PALAVRINHA FINAL

Chegamos ao final deste material sobre **ferramentas digitais voltadas aos conteúdos de modelos atômicos, moléculas, gráficos e mapas interativos para o Ensino Básico**. Esperamos que você tenha gostado do conteúdo e que possa aproveitá-lo de alguma forma.

Uma sugestão é explorar o *Jardim de Moléculas*, para os conteúdos relacionados a modelos atômicos e moléculas, e a seção *Gráficos & Mapas* para os demais temas, ambos disponíveis no *site Bioquanti*. O *Jardim de Moléculas* ilustra diversos modelos atômicos, classificados em grupos por classes químicas ou por interesse geral. Já a seção *Gráficos & Mapas* apresenta diversos objetos didáticos prontos para uso e/ou adaptação (com códigos e saídas), organizados por área do Ensino Médio, tema e habilidades específicas.

Complementarmente, se desejar se aventurar um pouco mais nessa área, reiteramos o convite para acessar o *site Bioquanti*, cuja denominação é uma contração lexical para *Bioquímica Quantitativa*. O portal oferece outras aplicações e instruções para o uso do *Jmol*, *R* e *RStudio*, além de um programa que simula transformações dinâmicas por diferença de matizes entre elemento precursor e sucessor. Essa funcionalidade permite a construção e visualização dinâmicas de diversos tipos esquemáticos, assim como a simulação do que seria desenhado com um lápis e uma folha de papel em branco — trata-se do SISMA (Sistema de Mapas Autocatalíticos).

Metidez à parte, o website foi agraciado com o 2º lugar na 3ª Edição do Prêmio Nacional de Ensino de Bioquímica e Biologia Molecular “Bayardo Baptista Torres”, promovido pela Sociedade Brasileira de Bioquímica e Biologia Molecular (2023). Seu conteúdo foi publicado na Revista de Ensino de Bioquímica, e disponibilizado como um recurso educacional aberto no portal eduCAPES, do Ministério da Educação do governo brasileiro.

Adicionalmente, enquanto este texto estava em fase de conclusão para envio à Editora, um objeto didático voltado ao estudo de gráficos

### SUMÁRIO

interativos para funções e dados no ensino e na pesquisa, foi concebido para permitir edição, compartilhamento irrestrito e visualização em navegadores comuns. O aplicativo foi contemplado com o 1º lugar na 4ª Edição do mesmo Prêmio Nacional de Ensino (2025), e também disponibilizado no portal eduCAPES.

Por fim, se desejar fazer qualquer comentário, crítica ou sugestão sobre o material construído, entre em contato conosco pelo submenu homônimo deste mesmo *website*: Contato.

**Tenha uma ótima Vida!**

## FINANCIAMENTO

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

## A ELABORAÇÃO DESTA OBRA

A formatação de texto (*RMarkdown*), a inserção e a referenciação cruzada de figuras, os *hiperlinks*, o gerenciamento bibliográfico e a conversão do *documento dinâmico* final para compilação em um modelo de arquivo *DOCX* previamente configurado foram conduzidos exclusivamente com a linguagem de programação *R* (versão 4.3.3, fev/2024) em ambiente de desenvolvimento integrado *RStudio* (versão 2024.09.1 Build 394), utilizando o pacote *quarto* (versão 1.4.4) como sistema de publicação científica.

## SUMÁRIO

## SOBRE O AUTOR

### **José Maurício Schneedorf Ferreira da Silva**

Biólogo com doutorado direto em Bioquímica pela UFMG (Belo Horizonte, 1998; interação experimental ligante-proteína), e estágio pos-doc pela UFV (Viçosa-MG, 1999; termodinâmica de células tumorais). Atualmente é professor titular junto ao Departamento de Bioquímica da UNIFAL-MG, onde atua como docente em disciplina homônima para diversos Cursos de Graduação, e para os Programas de Pós-Graduação de Química (Eletroanálise) e de Biotecnologia (Bioquímica Quantitativa) da instituição. Temas de interesse: Ensino Reprodutível, e Bioeletroquímica.

*Lattes:* <http://lattes.cnpq.br/0436922594542722>

*ORCID:* <https://orcid.org/0000-0002-2031-6315>

# ÍNDICE REMISSIVO

## A

animação 9, 10, 60, 61, 111, 112, 113, 114, 117, 118

aplicativos 9, 83, 102

aprendizagem 9, 10, 11, 18, 83, 121

## C

código 9, 18, 20, 21, 26, 28, 33, 34, 35, 38, 41, 42, 46, 47, 53, 54, 62, 72, 77, 101, 105, 108, 109, 117, 123

comando 30, 32, 33, 34, 37, 40, 41, 44, 45, 47, 48, 50, 52, 53, 54, 55, 57, 58, 60, 61, 72, 75, 76, 77, 81, 87, 90, 92, 95, 99, 102, 106, 109, 111

computador 9, 19, 20, 26, 30, 37, 38, 39, 44, 45, 65, 67, 69, 80, 88

## D

delay 35, 61, 62

## E

ensino básico 10, 11, 18

Ensino Reprodutível 9, 10, 30, 33, 34, 53, 54, 77, 109, 129

ensino superior 10, 48

## F

ferramentas 10, 18, 19, 33, 127

## G

gráfico 9, 18, 31, 32, 33, 73, 83, 84, 85, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 104, 106, 107, 108, 109, 111, 112, 113, 114, 115, 116, 117, 119, 126

## I

interatividade 10, 18, 70, 79, 83, 87, 88, 98, 104, 109

## J

Jmol 20, 21, 24, 25, 26, 28, 30, 31, 33, 34, 37, 38, 40, 41, 42, 43, 44, 45, 46, 47, 48, 51, 52, 53, 54, 55, 57, 58, 59, 60, 61, 62, 72, 73, 75, 127

JSmol 20, 28, 31, 37, 38, 39, 44, 47, 48, 54

## L

linguagem 9, 10, 21, 33, 40, 42, 73, 77, 83, 128

livros 10, 19, 38, 48, 72

## M

metodologia ativa 9, 18, 33

molécula 20, 26, 27, 28, 30, 34, 35, 37, 39, 44, 46, 49, 50, 51, 52, 53, 54, 56, 58, 60, 61

mouse 18, 26, 30, 31, 32, 33, 37, 39, 44, 45, 49, 72, 83, 87, 102, 107, 116, 117, 118, 119, 121, 123, 126

## N

navegador 9, 19, 20, 26, 27, 30, 31, 37, 39, 44

nuvem 18, 44, 55, 56, 57, 65, 67, 68, 69, 72, 75, 80

## O

objetos didáticos 10, 22, 72, 79, 81, 127

online 20, 21, 26, 28, 30, 31, 38, 41, 45, 67, 72, 102, 109

## P

pacote 18, 22, 31, 79, 80, 81, 85, 102, 104, 128

plotly 79, 80, 81, 83, 85, 86, 87, 88, 90, 91, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 104, 105, 106, 109, 111, 112, 115, 117, 121, 124, 125

programa 9, 18, 19, 20, 21, 22, 26, 30, 31, 32, 33, 34, 47, 48, 49, 54, 60, 65, 66, 67, 72, 73, 76, 79, 127

programação 9, 10, 11, 21, 40, 42, 54, 77, 128

## R

RStudio 18, 21, 22, 33, 47, 63, 64, 65, 66, 67, 68, 69, 70, 72, 73, 74, 77, 78, 79, 80, 83, 84, 85, 121, 127, 128

## S

script 34, 38, 53, 54, 60, 61, 62, 72, 74, 75, 76, 77, 85, 91, 105, 108, 117, 123

spin 35, 60, 62

## T

tabelas 21, 22, 116

## Z

zoomTo 61, 62

## SUMÁRIO



A decorative network of icons and chemical structures is overlaid on the left side of the page. It includes a pie chart, a DNA double helix, a chemical structure of a branched alkene, a stylized plant, a chemical structure of a cyclohexene derivative, and a chemical structure of a branched alkane. These are connected by a network of white lines with orange dots at the nodes.

[www.PIMENTACULTURAL.com](http://www.PIMENTACULTURAL.com)

# VIVIFICANDO CONTEÚDOS PARA O ENSINO BÁSICO

Moléculas, Gráficos,  
e Mapas Interativos

