Wolfram Language

Insper | Instituto de Ensino e Pesquisa

Computação para Ciências dos Dados

PÓS-GRADUAÇÃO EM DATA SCIENCE E DECISÃO

Prof. Daniel Carvalho @danielscarvalho

Dica do Dia: 010

Este é o ambiente notebook (UI) da Wolfram, criado em 1988 no Mathematica 1.0 para realização de computação científica exploratória, versão atual 12.1.

Use a combinação das teclas Shift + Enter para processar a célula de input.

Treinamento básico:

```
In[ ]:= u = 10 / 2 + 1 / 9 - 4
```

$$\text{Out[ ]=}\ \frac{10}{9}$$

```
N[u, 20]
```

```
Out[ ]= 1.1111111111111111111
```

O kernel do sistema é simbólico, a maioria das linguagens de programação são numéricas.
Note abaixo que o valor de x não foi definido, mas o sistema é capaz de processar!

```
In[2]:= 1 / (2 x) * 1 / Sqrt[x]
```

$$\text{Out[2]=}\ \frac{1}{2\,x^{3/2}}$$

A Wolfram Language trabalha com precisão arbitrária não depende do tamanho da palavra do SO (32, 64 bits...) ou capacidade da variável numérica.

```
In[ ]:= N[π, 100]
```

```
Out[ ]= 3.14159265358979323846264338327950288419716939937510582097494459230781640628620899
 86280348253421170 68
```

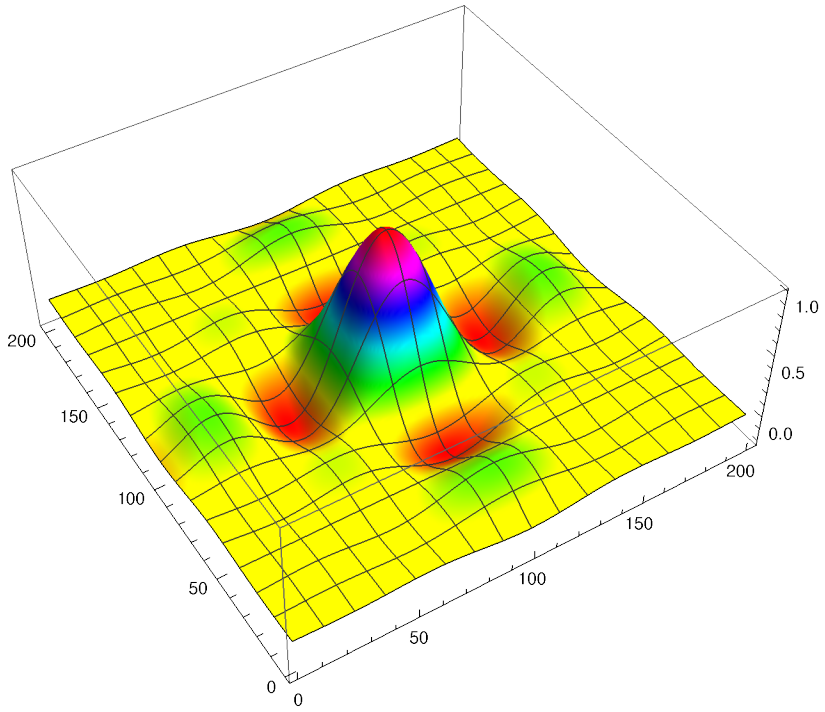
Vamos ver algumas funções:

```
In[ ]:= Plus[10, 3]
Out[ ]= 13

In[3]:= Plus[x, 5]
Out[3]= 5 + x

In[ ]:= Range[10]
Out[ ]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

        Table[x, {x, -10, 10}]
Out[ ]= {-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In[ ]:= Solve[x^2 + 3 x + 2 == 0, x]
Out[ ]= {{x → -2}, {x → -1}}

In[ ]:= Table[Solve[x^2 + 3 x + 2 == z, x], {z, 0, 10}]
```

$$\left\{\{\{x \to -2\}, \{x \to -1\}\}, \left\{\left\{x \to \frac{1}{2}\left(-3-\sqrt{5}\right)\right\}\right\}, \left\{\left\{x \to \frac{1}{2}\left(-3+\sqrt{5}\right)\right\}\right\}, \{\{x \to -3\}, \{x \to 0\}\},\right.$$
$$\left\{\left\{x \to \frac{1}{2}\left(-3-\sqrt{13}\right)\right\}\right\}, \left\{\left\{x \to \frac{1}{2}\left(-3+\sqrt{13}\right)\right\}\right\}, \left\{\left\{x \to \frac{1}{2}\left(-3-\sqrt{17}\right)\right\}\right\}, \left\{\left\{x \to \frac{1}{2}\left(-3+\sqrt{17}\right)\right\}\right\},$$
$$\left\{\left\{x \to \frac{1}{2}\left(-3-\sqrt{21}\right)\right\}\right\}, \left\{\left\{x \to \frac{1}{2}\left(-3+\sqrt{21}\right)\right\}\right\}, \{\{x \to -4\}, \{x \to 1\}\},$$
$$\left\{\left\{x \to \frac{1}{2}\left(-3-\sqrt{29}\right)\right\}\right\}, \left\{\left\{x \to \frac{1}{2}\left(-3+\sqrt{29}\right)\right\}\right\}, \left\{\left\{x \to \frac{1}{2}\left(-3-\sqrt{33}\right)\right\}\right\}, \left\{\left\{x \to \frac{1}{2}\left(-3+\sqrt{33}\right)\right\}\right\},$$
$$\left.\left\{\left\{x \to \frac{1}{2}\left(-3-\sqrt{37}\right)\right\}\right\}, \left\{\left\{x \to \frac{1}{2}\left(-3+\sqrt{37}\right)\right\}\right\}, \left\{\left\{x \to \frac{1}{2}\left(-3-\sqrt{41}\right)\right\}\right\}, \left\{\left\{x \to \frac{1}{2}\left(-3+\sqrt{41}\right)\right\}\right\}\right\}$$

In[ ]:=
```
ListPlot3D[Table[Sinc[x] * Sinc[y], {x, -10, 10, .1}, {y, -10, 10, .1}],
 ColorFunction → Hue, PlotRange → All]
```

Out[ ]=



É a plataforma mais poderosa para resolver cálculos, recentente todos os problemas dos principais livros acadêmicos de cálculos foram testados com sucesso:

```
Integrate[x^2 + 3 x + Cos[x], x] // TraditionalForm
```

Out[ ]//TraditionalForm=

$$\frac{3\,x^2}{2} + x\,\mathrm{x}^2 + \sin(x)$$

```
D[x^2 + 3 x + Cos[x], x] // TraditionalForm
```

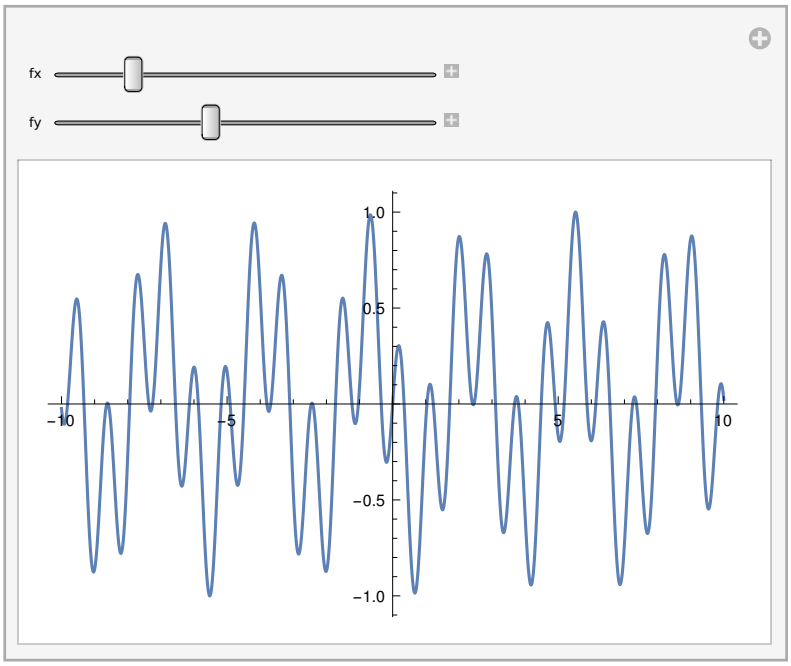
Out[ ]//TraditionalForm=

$$3 - \sin(x)$$

Em uma linha de comando podemos criar uma interface:

```
Manipulate[Plot[Sin[a fx] * Cos[a fy], {a, -10, 10}],
 {fx, 1, 10},
 {fy, 1, 10}]
```

Out[ ]=



Como realizar uma transformada de fourier? Use a função Fourier, simples assim, a função tem o nome relacionado!

Pressione F1 para o help

In[ ]:= `? Four*`

Out[ ]=

System`

Fourier	FourierCosTransform	FourierDST	FourierSequenceTransform	FourierSinTransform
FourierCoefficient	FourierDCT	FourierDSTMatrix	FourierSeries	FourierTransform
FourierCosCoefficient	FourierDCTFilter	FourierMatrix	FourierSinCoefficient	FourierTrigSeries
FourierCosSeries	FourierDCTMatrix	FourierParameters	FourierSinSeries	

In[ ]:= `? Fourier`

Out[ ]=

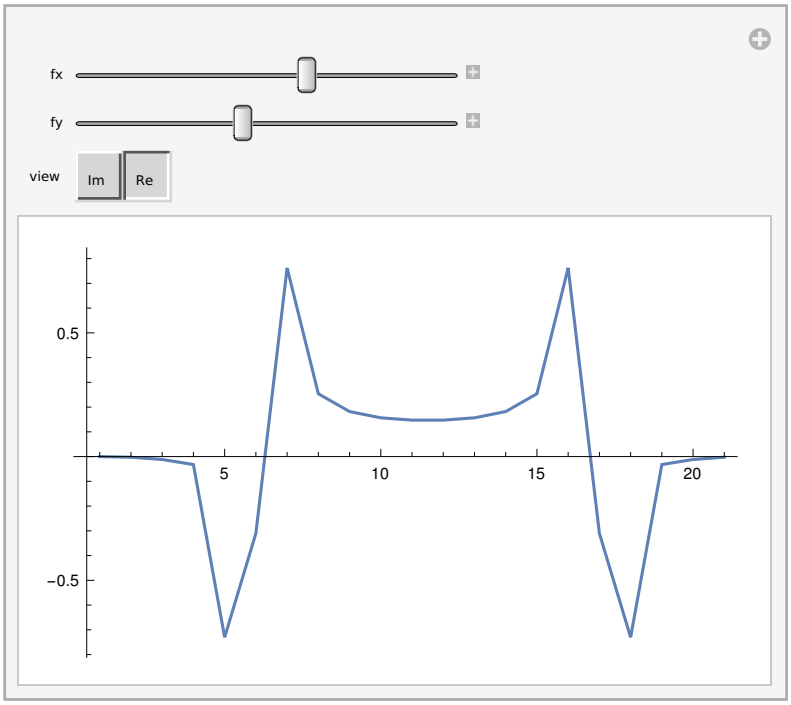
Symbol

Fourier[*list*] finds the discrete Fourier transform of a list of complex numbers.

Fourier[*list*, {p_1, p_2, ...}] returns the specified positions of the discrete Fourier transform.

In[ ]:=
```
Manipulate[
 ListLinePlot[view@Fourier[Table[Sin[a fx]*Cos[a fy], {a, -10, 10}]], PlotRange → All],
 {fx, 1, 10},
 {fy, 1, 10},
 {view, {Im, Re}}]
```

Out[ ]=



Pronto, treinamento acabou, agora vamos lá:

É possível realizar computação avançada neste ambiente, vamos iniciar com NLP, a plataforma tem dados acurados e computáveis.
Diversos sistemas populares utilizam estes recursos da Wolfram, mas você nem imagina!! :-)

Pressionando = no início de uma nova célula para realizar consultas em inglês, em linguagem natural (Shift + Enter para processar)

What is Sonia Braga age?

Out[ ]= 69 yr 9 mo 28 days

How many days the Brazil president is in charge?

Out[ ]= Jair Bolsonaro (from January 1, 2019 to present)

Distance from NYC to Paris

Out[ ]= 3618.85 mi

Distance from Diadema to Registro in kilometers

Out[ ]= 153.273 km

In[ ]:=
Whather in São Paulo now

```
WeatherData [ São Paulo CITY , "Temperature "]
```

Out[ ]= 21 °C

What is the boiling point of water?

Out[ ]= 99.9839 °C

In[ ]:=
What is the boiling point of water at the mount Everest ?

Result

70.41 °C

Out[ ]= 70.41 °C

In[4]:=
Apple stock price

Apple FINANCIAL ENTITY [price]

Out[4]= { Day : Fri 3 Apr 2020 , $241.41 }

In[6]:=
IBM Employees »

IBM COMPANY [total employees]

Out[6]= 381 100 people

In[7]:=
Number of students in Brazil

Brazil COUNTRY [students]

Out[7]= 4.68293×10^7 people

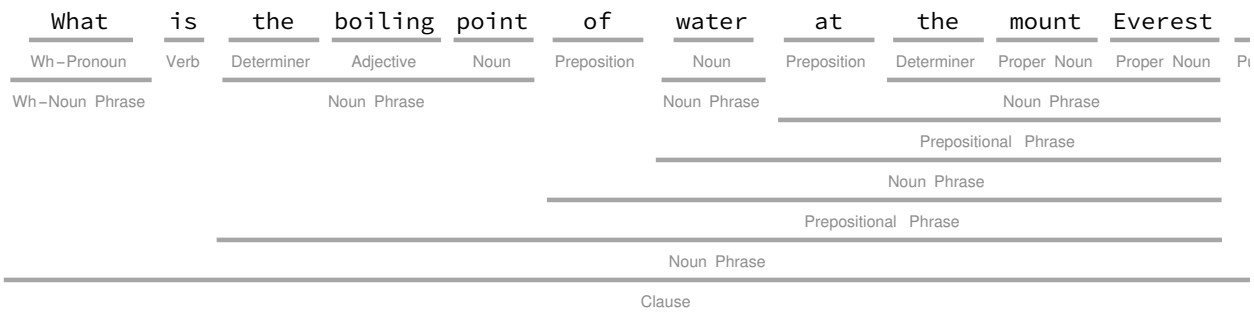
In[8]:=
Number of college students in Brazil

Brazil COUNTRY [students post-secondary education]

Out[8]= 6.92932×10^6 people

In[ ]:=
```
TextStructure["What is the boiling point of water at the mount Everest?"]
```

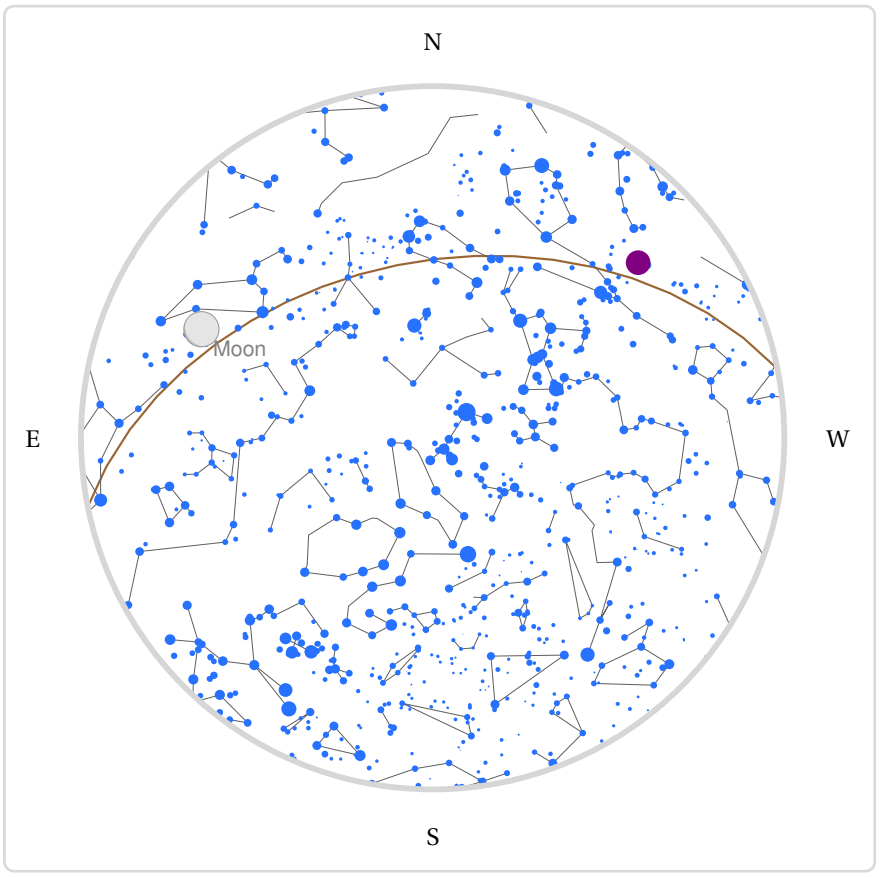
Out[ ]=



Como está o céu agora, onde estou?

In[ ]:=

Out[ ]=



In[ ]:= `GeoGraphics[GeoDisk[São Paulo CITY, 1 km]]`

Out[ ]=

In[ ]:= `Here`

Out[ ]= `GeoPosition[{-23.63, -46.63}]`

In[ ]:=

Where am I? »

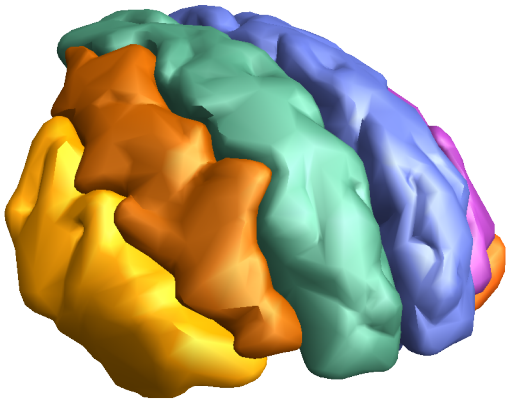
`Here`

Out[ ]= `GeoPosition[{-23.63, -46.63}]`

Há dados computáveis anatômicos, de química, humano, cachorro, gato, cavalo, genoma, medicamentos, doenças etc...

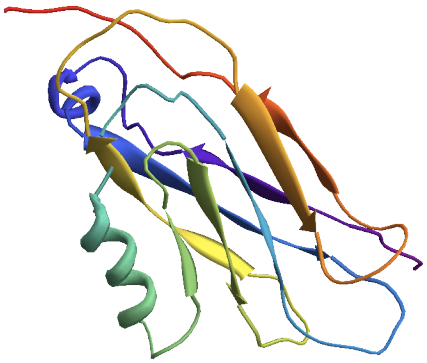
In[ ]:=
```
AnatomyPlot3D[Entity["AnatomicalStructure", "PrefrontalCortex"],
 "PlotTheme" → "Scientific"]
```

Out[ ]=



In[ ]:=
```
ProteinData["A2M", "MoleculePlot"]
```

Out[ ]=

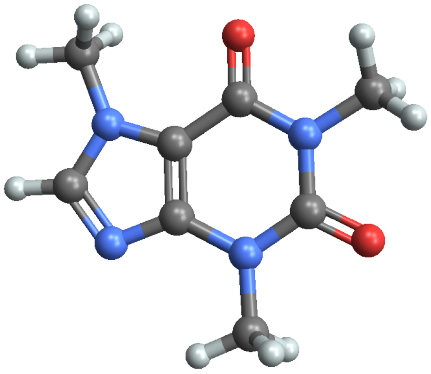


In[ ]:=

3D caffeine molecule

caffeine CHEMICAL [*molecule plot*]

Out[ ]=

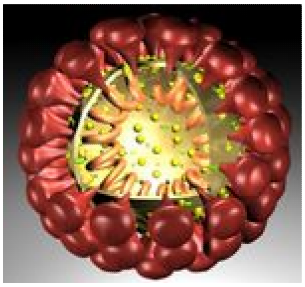


In[ ]:=

Coronavirus image

Coronavirus SPECIES SPECIFICATION [*image*]

Out[ ]=

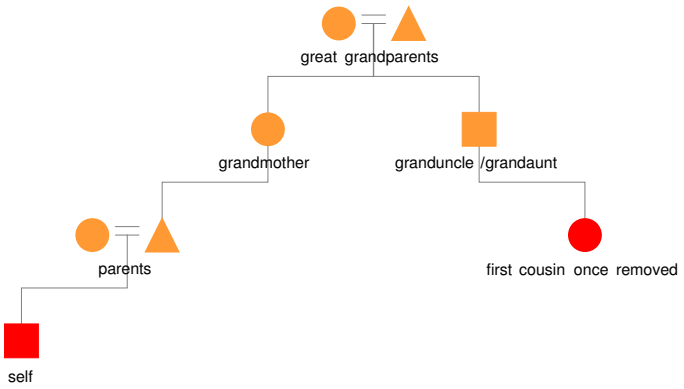


In[ ]:=

Grandsmother 's nience

```
GenealogyTreePlot [{"Grandmother ", "Niece "}]
```

Out[ ]=



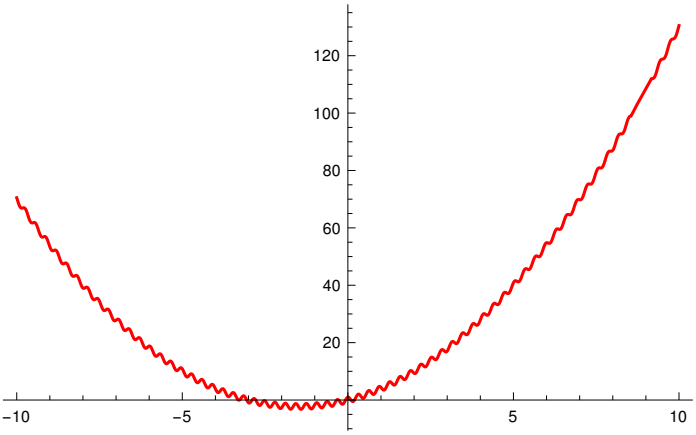
Faça um pedido bem definido e o sistema transforma em código:

In[ ]:=

Plot the $x^2+3x+\text{Cos}(x * 20)$ for x from −10 to 10 in red »

```
Plot[x ^ 2 + 3 * x + Cos[x * 20], {x, -10, 10}, PlotStyle -> Red]
```

Out[ ]=



Há dados acurados computáveis como parte da plataforma, isso aliado a capacidade de processa - mento simbólico, e programação funcional, torna o ambiente ideal para realizar IA e ML:

In[ ]:=
```
FacialFeatures[ ] // Dataset
```


Out[ ]=

Image	Age	Gender	Emotion
	29	Female	happiness
	12	Male	happiness
	52	Male	happiness
	22	Male	happiness

In[ ]:=
```
ImageIdentify[ ]
```


Out[ ]=
terrier

Quanto mais tempo a WL tem para trabalhar, melhor a resposta:

In[ ]:=
```
Table[ImageIdentify[ , SpecificityGoal → g], {g, 0, 1, .1}]
```


Out[ ]=
{ animal , terrier , terrier , terrier , terrier ,
terrier , terrier , terrier , terrier , terrier , Yorkshire terrier }

In[ ]:=
```
WordCloud[%]
```

Out[ ]=

In[ ]:=
```
i =
```

```
;
```

In[ ]:=
```
ImageBoundingBoxes[i]
```

Out[ ]=
```
⟨| domestic cat → {Rectangle[{50.0503, 48.6728}, {381.29, 504.017}]} |⟩
```

In[ ]:=
```
HighlightImage[i, %]
```

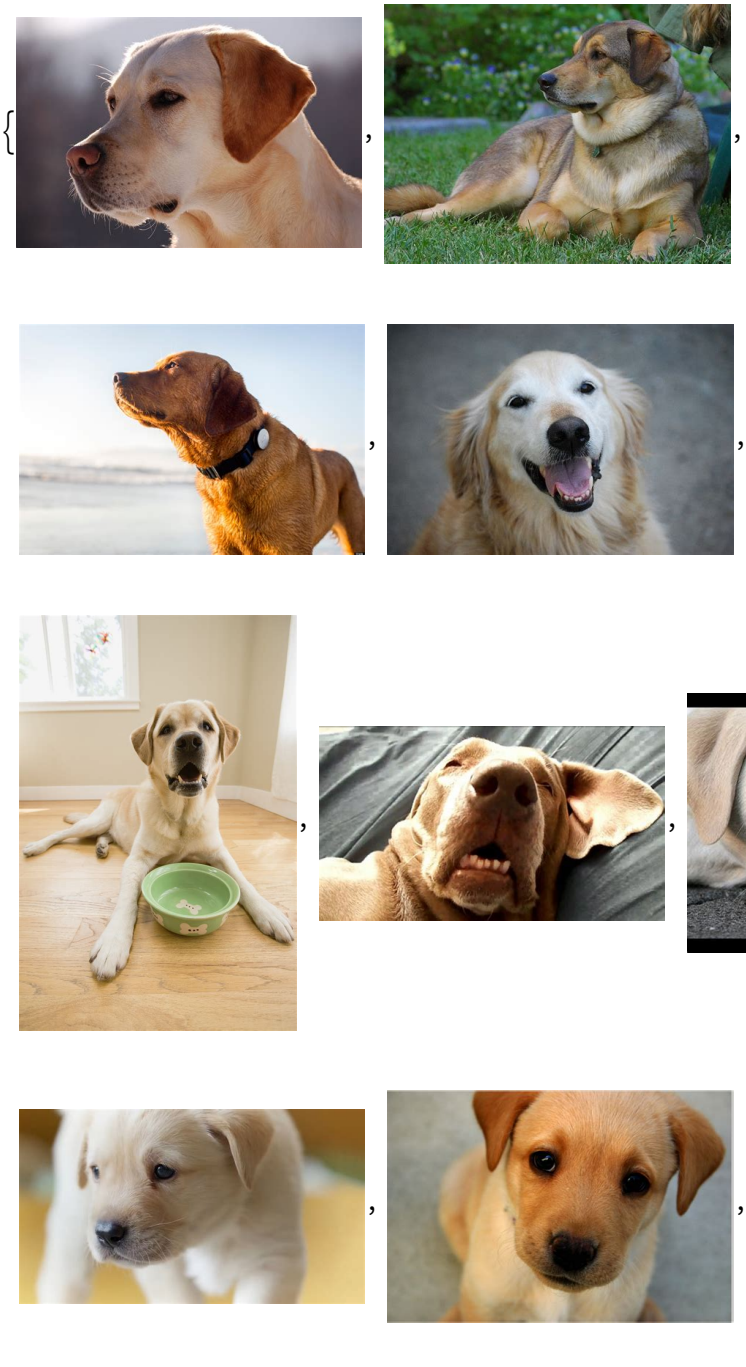
Out[ ]=

Ops! Cade a maça! Ver o help da função ImageBoundingBoxes...

Vamos buscar imagens na Inernet para treinamento:

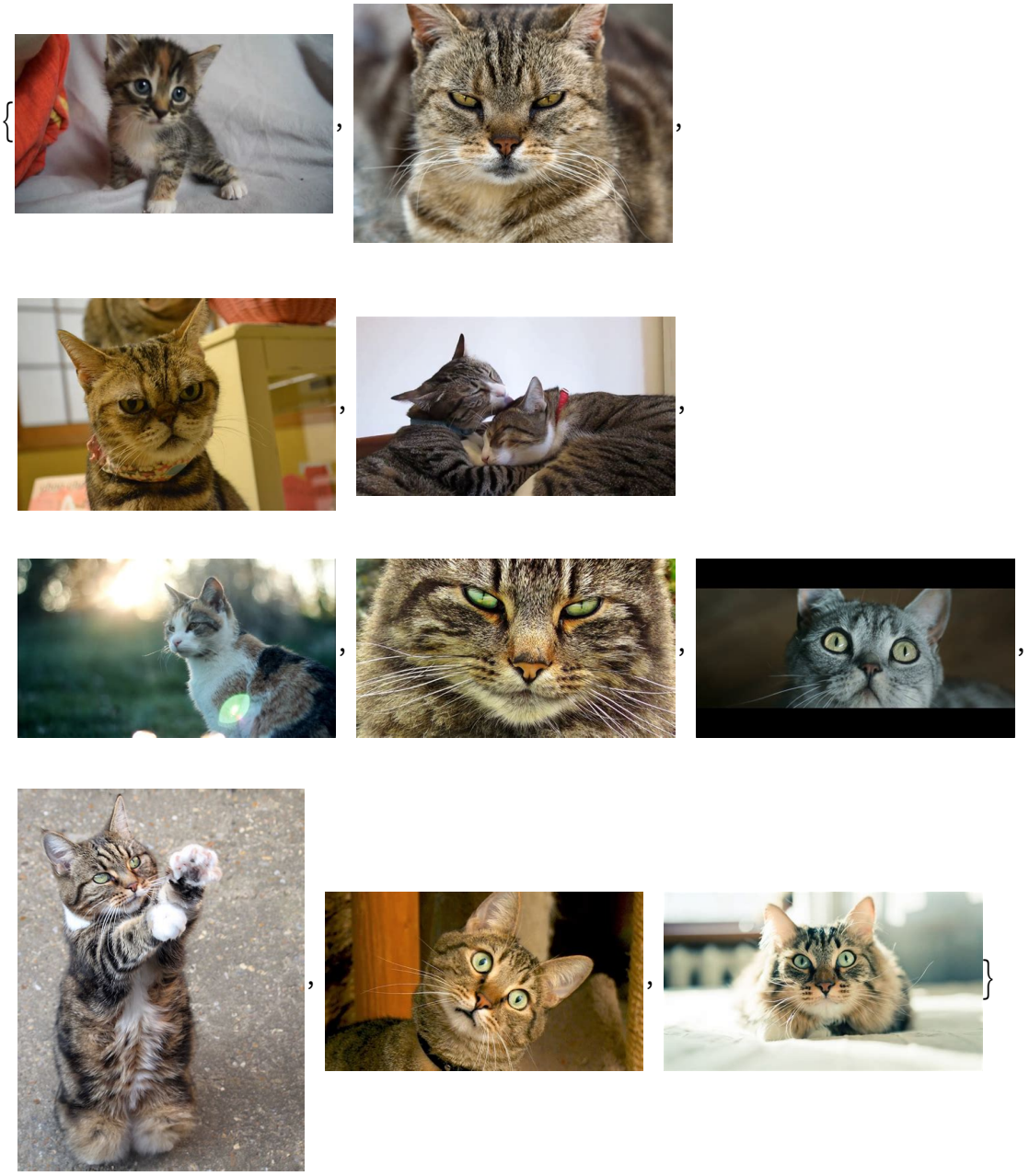
In[ ]:=
```
dog = WebImageSearch["dog", "Thumbnails"]
```

Out[ ]=



In[ ]:=
```
cat = WebImageSearch["cat", "Thumbnails"]
```

Out[ ]=



In[ ]:=
```
trex = WebImageSearch["t-rex", "Thumbnails"]
```

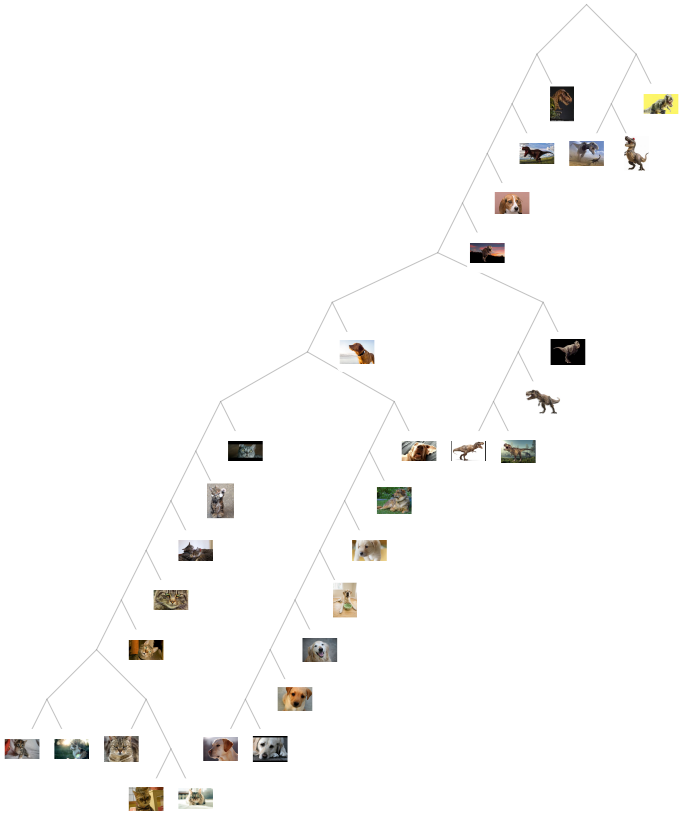
Out[ ]=

Vamos observar as imagens em forma de cluster, note que as imagens são organizadas na arvore não o conteúdo (animais)...

Há diversas funções para realizar classificação, regressão, clusterização entre outros algoritmos que estão na moda.

In[ ]:=
```
ClusteringTree [Flatten[{cat, dog, trex}]]
```

Out[ ]=



Agora vamos criar e treinar um classificador, em uma linha de comando:

In[ ]:=
```
animal = Classify[<|
    "cão" → dog,
    "gato" → cat,
    "t-rex" → trex
    |>]
```

Out[ ]=
```
ClassifierFunction [ ]
```
Input type : Image
Classes : cão, gato, t-rex
Method : LogisticRegression
Number of training examples : 30

Data not in notebook ; Store now »

Teste com imagem de um lagarto...

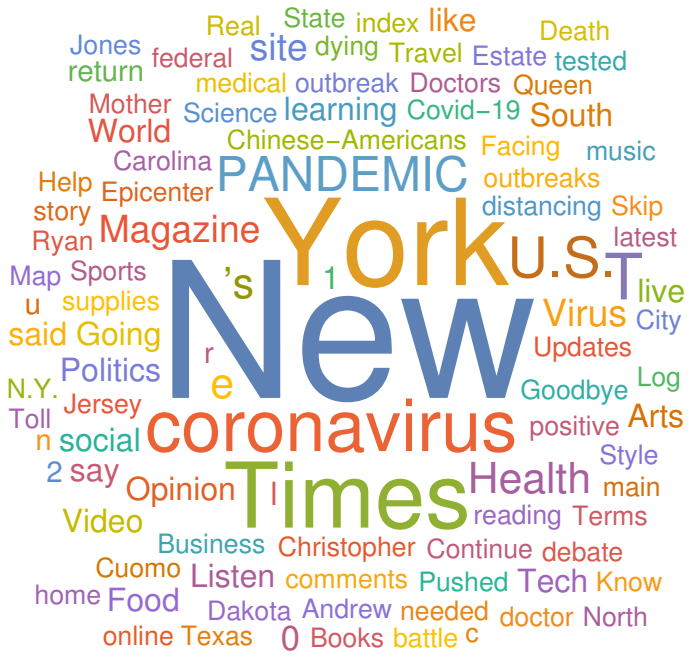
In[ ]:=
```
animal[ ]
```


Out[ ]=
```
t-rex
```

O que está pegando no NYT hoje?

In[ ]:=
```
WordCloud[DeleteStopwords[Import["http://www.nyt.com"]]]
```

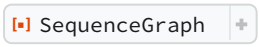
Out[ ]=



Função em Wolfram Language que implementei e foi publicada:

In[ ]:=
```
ResourceFunction["SequenceGraph"]
```

Out[ ]=



In[ ]:=
```
Range[10]
```

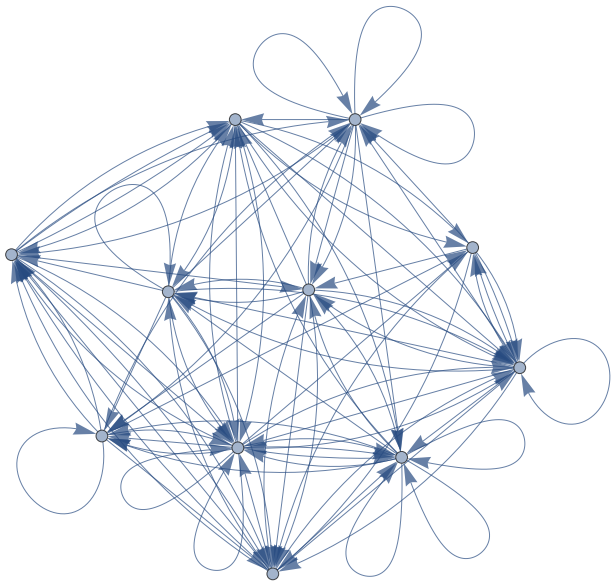
Out[ ]=
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

Out[ ]=

In[ ]:=
```
RandomInteger[10, 100]
SequenceGraph[%]
```

Out[ ]=
```
{2, 3, 6, 9, 1, 6, 0, 2, 2, 10, 6, 4, 7, 6, 2, 4, 3, 10, 0, 7, 0, 5, 2, 0,
 3, 8, 7, 10, 5, 5, 9, 8, 0, 3, 2, 7, 0, 1, 4, 6, 2, 9, 7, 0, 3, 4, 5, 6, 10,
 9, 7, 4, 7, 4, 8, 1, 4, 2, 8, 5, 6, 3, 2, 8, 8, 5, 5, 8, 3, 1, 3, 3, 5, 5, 0,
 8, 8, 0, 1, 3, 8, 9, 4, 3, 1, 10, 10, 10, 10, 6, 4, 7, 8, 5, 9, 10, 7, 8, 9, 9}
```

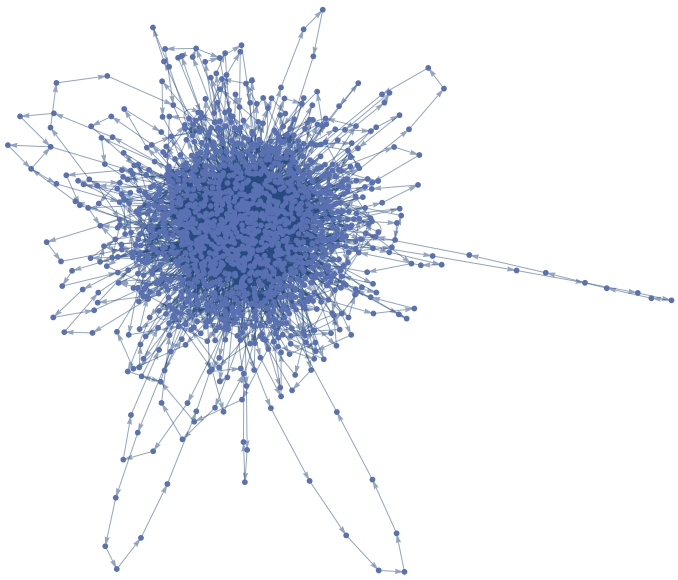
Out[ ]=



Agora vamos gerar um grafo do livro “Alice no pais das maravilhas”, em uma única linha de código, cada nó é uma palavra na ordem que aparece no livro.

In[ ]:=
```
ResourceFunction ["SequenceGraph "][StringSplit [
  DeleteStopwords [ResourceData ["Alice  in  Wonderland "]], RegularExpression ["\\W+"]]]
```

Out[ ]=



Usando outra função que publiquei que consome a API do DuckDuckGo:

In[ ]:=
```
ResourceFunction ["DuckDuckGoQuery "]
```

Out[ ]=
```
DuckDuckGoQuery
```

In[ ]:=
```
DuckDuckGoQuery ["Alan  Turing"]["Image"]
```

Out[ ]=
```
https://duckduckgo.com/i/a377e75e.jpg
```

In[ ]:=
```
atImage = Import[%]
```

Out[ ]=

In[ ]:=
```
Classify["NotablePerson", atImage]
```

Out[ ]=
Alan Turing

Alan Turing é uma Entity com semântica, ou seja, o sistema sabe o que é “Alan Turing” assim como “Brazil”, “São Paulo”, “Moon”, etc...:

In[ ]:=
```
Alan Turing PERSON ["Dataset"]
```

Out[ ]=

other names	{Alan Mathison Turing , Turing , Alan M. Turing , Alan Mathieson Turing }
astrological sign	–
date of birth	Sun 23 Jun 1912
place of birth	London , Greater London , United Kingdom
brothers	{John Turing }
children	–
Chinese zodiac sign	–
daughters	–
date of death	Mon 7 Jun 1954
place of death	Wilmslow , Cheshire , United Kingdom
entity classes	{ {birthplace :metroarea , londonunitedkingdommetro } }
father	{Julius Mathison Turing }
full name	Alan Mathison Turing
gender	Male
height	–
husbands	–
image	
space missions	–
mathematical contributions	{ halting problem , Church -Turing thesis , nondeterministic Turing machine , Tu
mother	{Ethel Sara Stoney }

rows 1–20 of 42

In[ ]:=
```
FindFaces[atImage, "Image"]
```

Out[ ]=

Processamento de imagens:

In[ ]:=
```
ColorNegate[EdgeDetect[Blur[atImage, 5], 5]]
```

Out[ ]=

In[ ]:=
```
ImageIdentify[atImage]
```

Out[ ]=

person

Entity, um elemento que tem semântica... Ctrl = para definir uma entity...

Brazil COUNTRY

In[ ]:=
Brazil COUNTRY `["Flag"]`

Out[ ]=

In[ ]:=
Brazil COUNTRY `["GDP"]`

Out[ ]= 1.86863×10^{12} per year

In[ ]:= China COUNTRY [GDP nominal] / Brazil COUNTRY [GDP nominal]

Out[ ]= 7.28244

In[ ]:= United States COUNTRY [dependencies]

Out[ ]= { American Samoa , Guam , Northern Mariana Islands ,
Puerto Rico , United States Minor Outlying Islands , United States Virgin Islands }

In[ ]:= Brazil COUNTRY [dependencies]

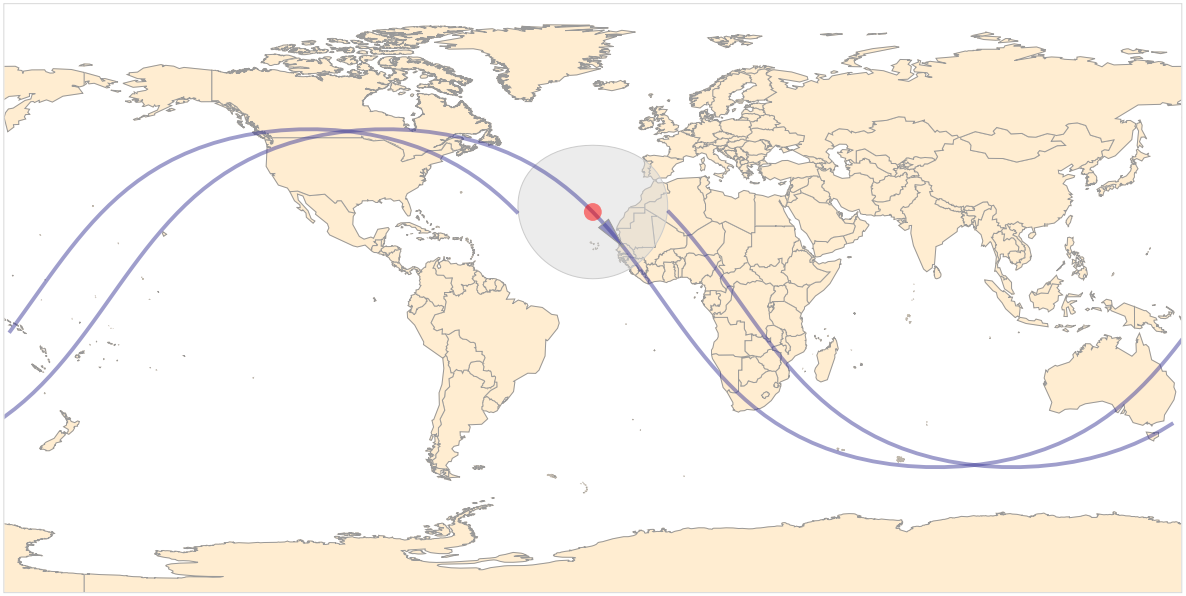
Out[ ]= {}

É possível definir "entities" junto a banco de dados da instituição. Desta forma as colunas do seu banco de dados relacional passam a ter propriedades semânticas e se integram ao contexto da Wolfram Language.

Para finalizar por hoje:

In[ ]:= Where is the ISS now?
Results (1 of 3)

Out[ ]=



In[ ]:= What is the meaning of life? »

Result

Out[ ]= 42

(according to the book *The Hitchhiker's Guide to the Galaxy*, by Douglas Adams)

Agora vamos publicar este notebook na nuvem, gerar um short URL e um QRCode para acesso fácil:

In[ ]:=
```
CloudPublish []
```

Ou no client (Mathematica) File, Publish to Cloud...

In[9]:=
```
URLShorten["www.wolframcloud.com/obj/dcarvalho/Published/Insper-Dica-010.nb"]
```

Out[9]= https://wolfr.am/LBbZEwAP

In[10]:=
```
BarcodeImage["https://wolfr.am/LBbZEwAP", "QR"]
```

Out[10]=

Vou também exportar como PDF...

Notou que não foram importadas bibliotecas externas...

Para saber mais:

- www.wolfram.com
- www.wolframalpha.com
- https://www.wolfram.com/language/fast-introduction-for-programmers/en/
- https://www.wolfram.com/language/elementary-introduction/2nd-ed/
- https://www.wolfram.com/wolfram-u/