

Wolfram Language



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

$$\frac{10}{9}$$

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

```
Out[121]= 1.11111111111111111111111
```

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[122]:= 1/2 x * Sqrt[x]/1
```

$$\frac{x^{3/2}}{2}$$

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

```
In[117]:= N[π, 100]
```

```
Out[117]= 3.1415926535897932384626433832795028841971693993751058209749445923078164062862089`.  
98628034825342117068
```

```
In[19]:= Plus[10, 3]
```

```
Out[19]= 13
```

Vamos ver algumas funções:

```
In[20]:= Range[10]
```

```
Out[20]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
Table[x, {x, -10, 10}]
```

```
Out[13]= {-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
In[125]:= Solve[x^2 + 3 x + 2 == 0, x]
```

```
Out[125]= {{x → -2}, {x → -1}}
```

```
In[126]:= Table[Solve[x^2 + 3 x + 2 == z, x], {z, 0, 10}]
```

```
Out[126]= {{x → -2}, {x → -1}}, {{x →  $\frac{1}{2}(-3 - \sqrt{5})$ }, {x →  $\frac{1}{2}(-3 + \sqrt{5})$ }}, {{x → -3}, {x → 0}},  

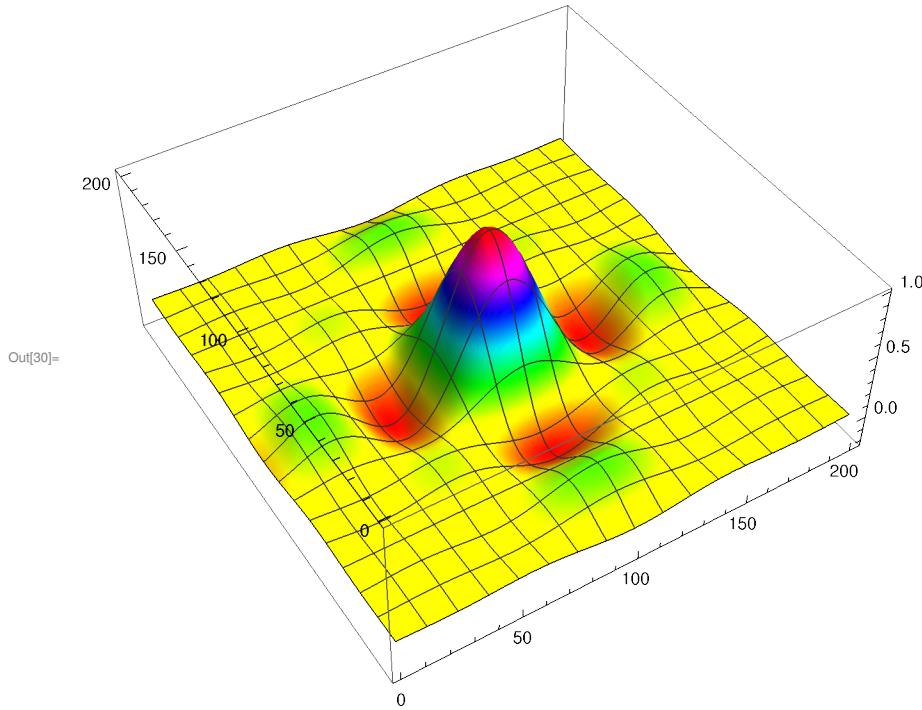
{{x →  $\frac{1}{2}(-3 - \sqrt{13})$ }, {x →  $\frac{1}{2}(-3 + \sqrt{13})$ }}, {{x →  $\frac{1}{2}(-3 - \sqrt{17})$ }, {x →  $\frac{1}{2}(-3 + \sqrt{17})$ }},  

{{x →  $\frac{1}{2}(-3 - \sqrt{21})$ }, {x →  $\frac{1}{2}(-3 + \sqrt{21})$ }}, {{x → -4}, {x → 1}},  

{{x →  $\frac{1}{2}(-3 - \sqrt{29})$ }, {x →  $\frac{1}{2}(-3 + \sqrt{29})$ }}, {{x →  $\frac{1}{2}(-3 - \sqrt{33})$ }, {x →  $\frac{1}{2}(-3 + \sqrt{33})$ }},  

{{x →  $\frac{1}{2}(-3 - \sqrt{37})$ }, {x →  $\frac{1}{2}(-3 + \sqrt{37})$ }}, {{x →  $\frac{1}{2}(-3 - \sqrt{41})$ }, {x →  $\frac{1}{2}(-3 + \sqrt{41})$ }}}
```

```
In[30]:= ListPlot3D[Table[Sinc[x]*Sinc[y], {x, -10, 10, .1}, {y, -10, 10, .1}],
ColorFunction → Hue, PlotRange → All]
```



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

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

Out[22]/TraditionalForm=

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

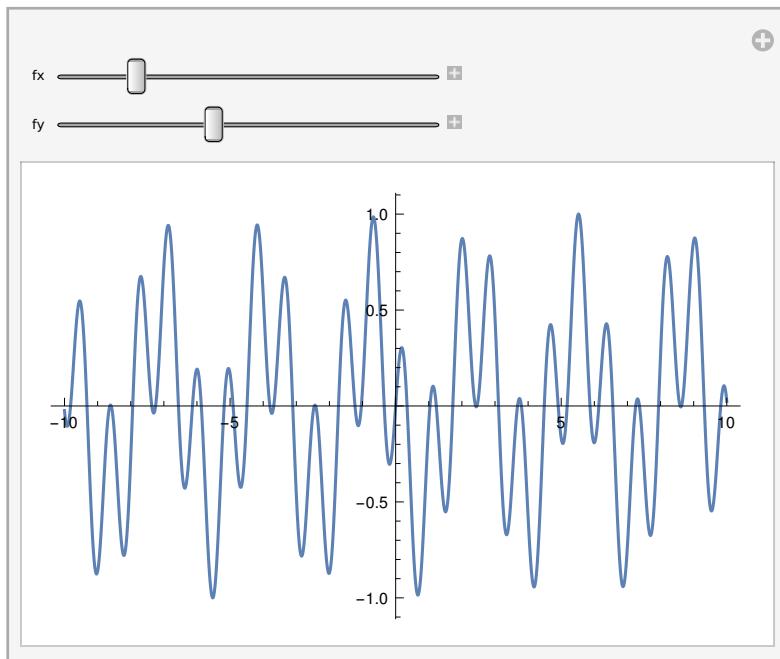
```
D[x^2 + 3 x + Cos[x], x] // TraditionalForm
```

Out[127]/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}]
```



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

Pressione F1 para o help

In[129]:= ?Four*

Out[129]=

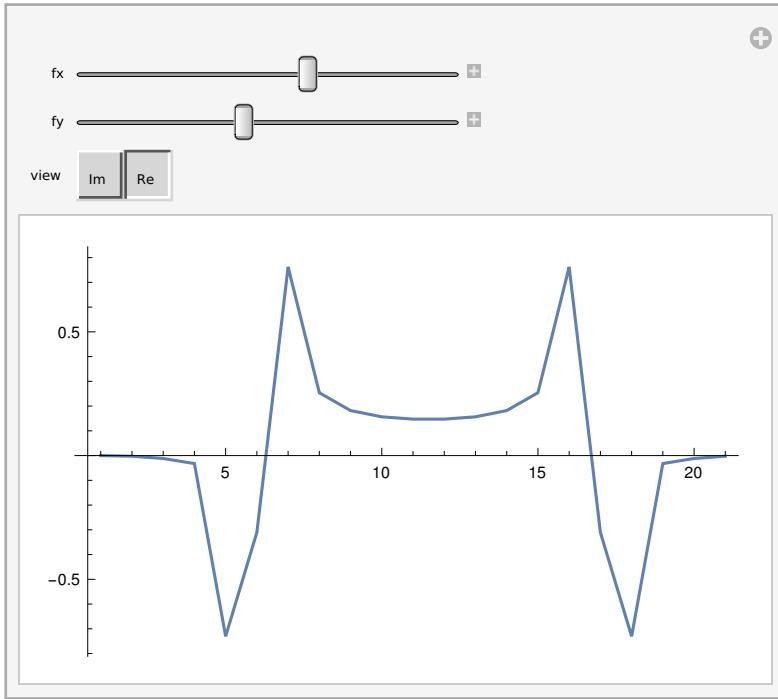
System` Fourier FourierCoefficient FourierCosCoefficient FourierCosSeries	FourierCosTransfo`rm FourierDCT FourierDCTFilter FourierDCTMatrix	FourierDST FourierDSTMatrix FourierMatrix FourierParameters	FourierSequenceTr`ansform FourierSeries FourierSinCoefficient FourierSinSeries	FourierSinTransfor`m FourierTransform FourierTrigSeries
--	--	--	---	--

In[128]:= ?Fourier

Out[128]=

Symbol	<i>i</i>
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[52]:= 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}}]
```



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[6]= 69 yr 9 mo 28 days

= How many days the Brazil president is in charge?

Out[7]= Jair Bolsonaro (from January 1, 2019 to present)

= Distance from NYC to Paris

Out[8]= 3618.85 mi

☰ Distance from Diadema to Registro in kilometers

Out[1]:= 153.273 km

In[158]:= **☰ Whather in São Paulo now**

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

Out[158]:= 21 °C

☰ What is the boiling point of water?

Out[1]:= 99.9839 °C

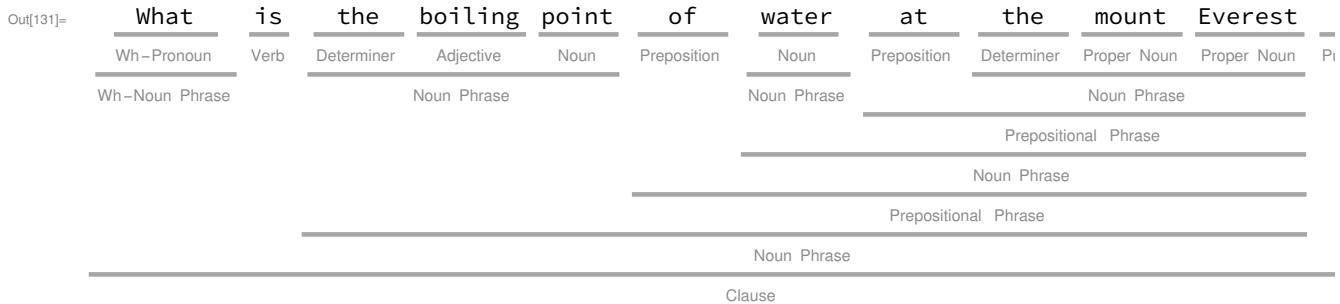
In[1]:= **☰ What is the boiling point of water at the mount Everest ?**

↳ Result

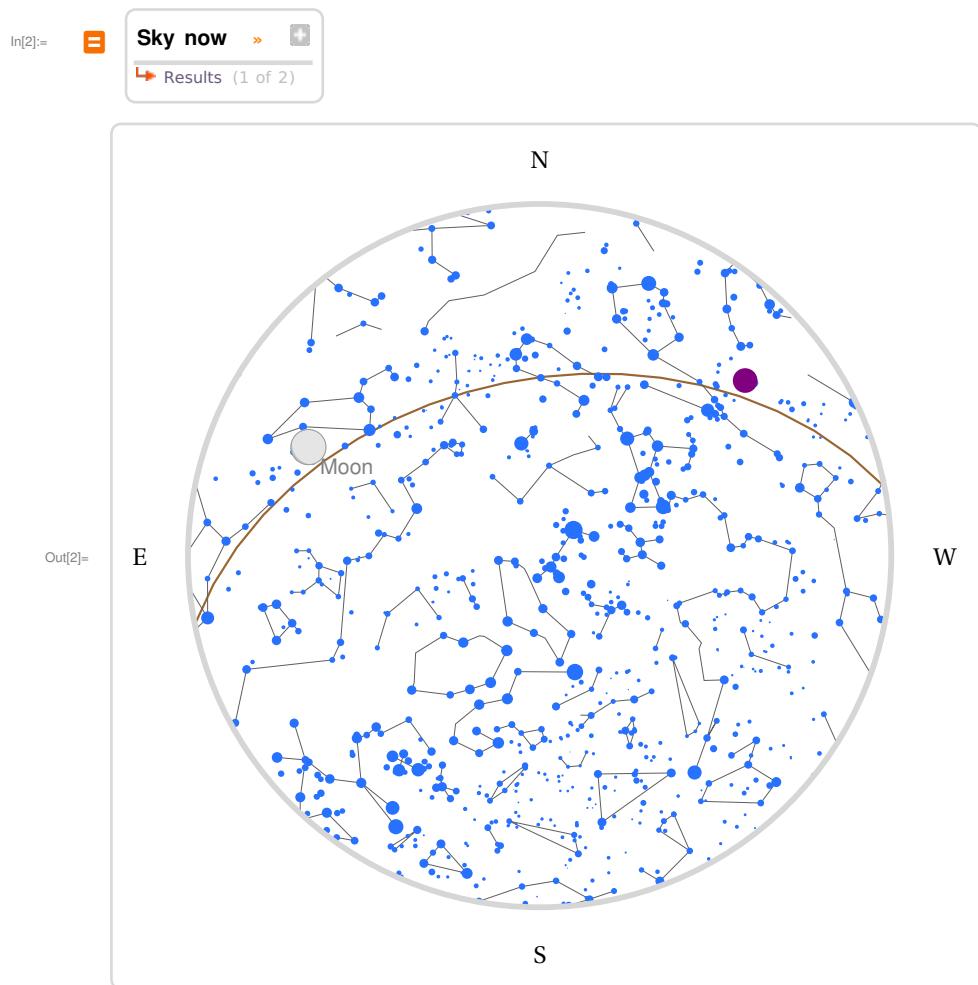
70.41 °C

Out[1]:= 70.41 °C

In[131]:= **TextStructure ["What is the boiling point of water at the mount Everest ?"]**

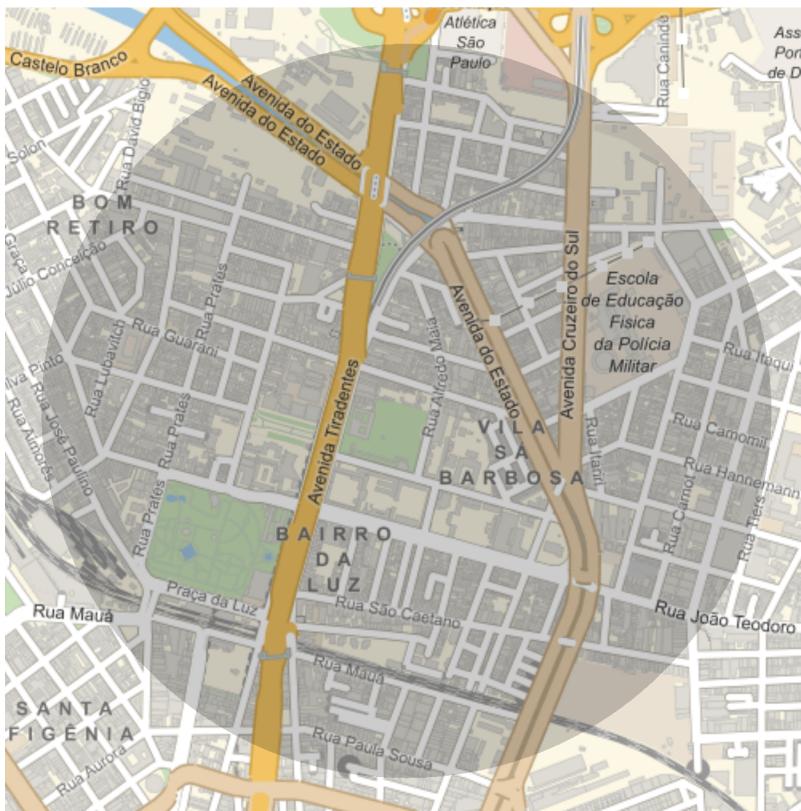


Como está o céu agora, onde estou?



In[113]:= **GeoGraphics [GeoDisk [São Paulo CITY , 1 km]]**

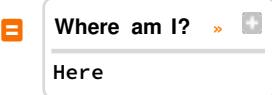
Out[113]=



In[114]:= **Here**

Out[114]= **GeoPosition [{-23.63 , -46.63}]**

In[115]:=

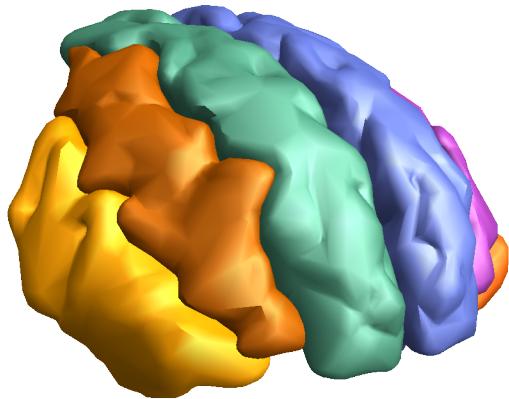


Out[115]= **GeoPosition [{-23.63 , -46.63}]**

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

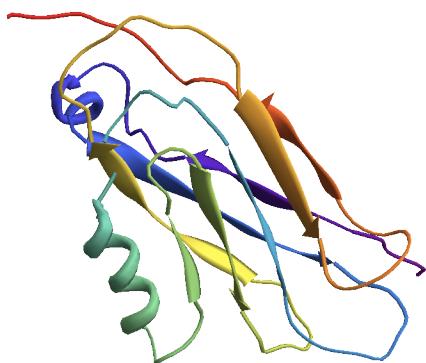
```
In[155]:= AnatomyPlot3D [Entity["AnatomicalStructure ", "PrefrontalCortex "],  
"PlotTheme " → "Scientific "]
```

Out[155]=



```
In[154]:= ProteinData ["A2M", "MoleculePlot "]
```

Out[154]=

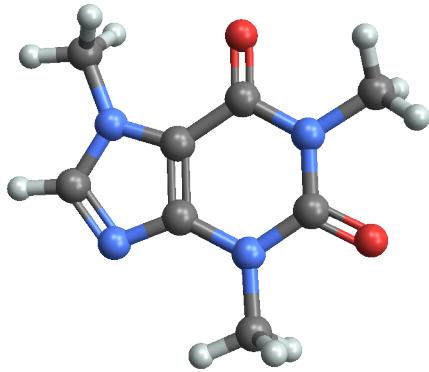


In[146]:=

3D caffeine molecule

caffeine CHEMICAL [molecule plot]

Out[146]=

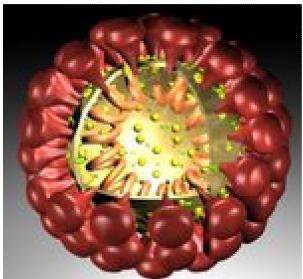


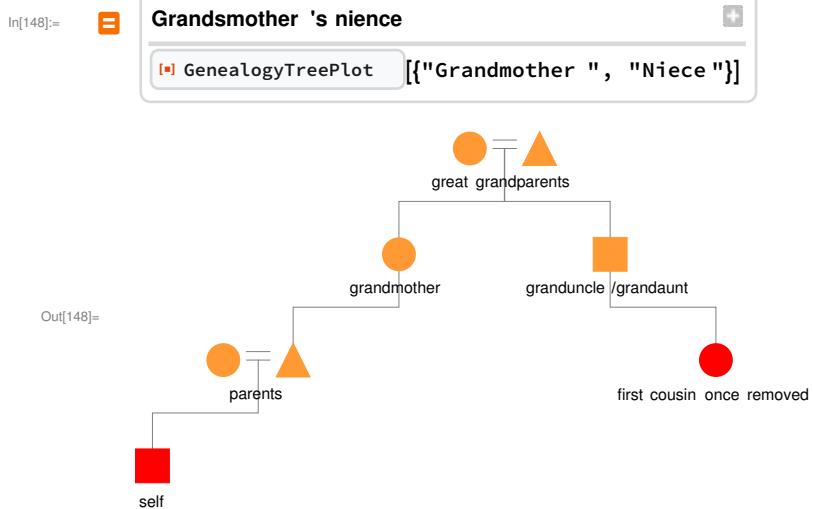
In[147]:=

Coronavirus image

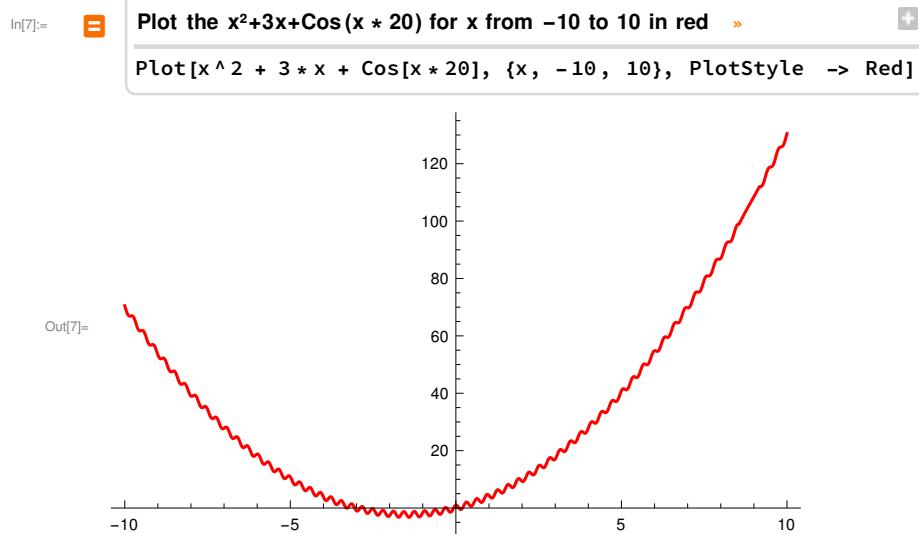
Coronavirus SPECIES SPECIFICATION [image]

Out[147]=





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



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

```
In[10]:= FacialFeatures [  ] // Dataset
```

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

```
In[9]:= ImageIdentify [
```

Out[9]= **terrier**

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

In[12]:= WordCloud [%]

animal

Out[12]=



Yorkshire terrier

In[34]:= i =



;

In[35]:= ImageBoundingBoxes [i]

Out[35]= <| domestic cat → {Rectangle[{50.0503, 48.6728}, {381.29, 504.017}]} |>

In[36]:= **HighlightImage** [i, %]

Out[36]=



Ops! Cade a maçã! Ver o help da função `ImageBoundingBoxes`...

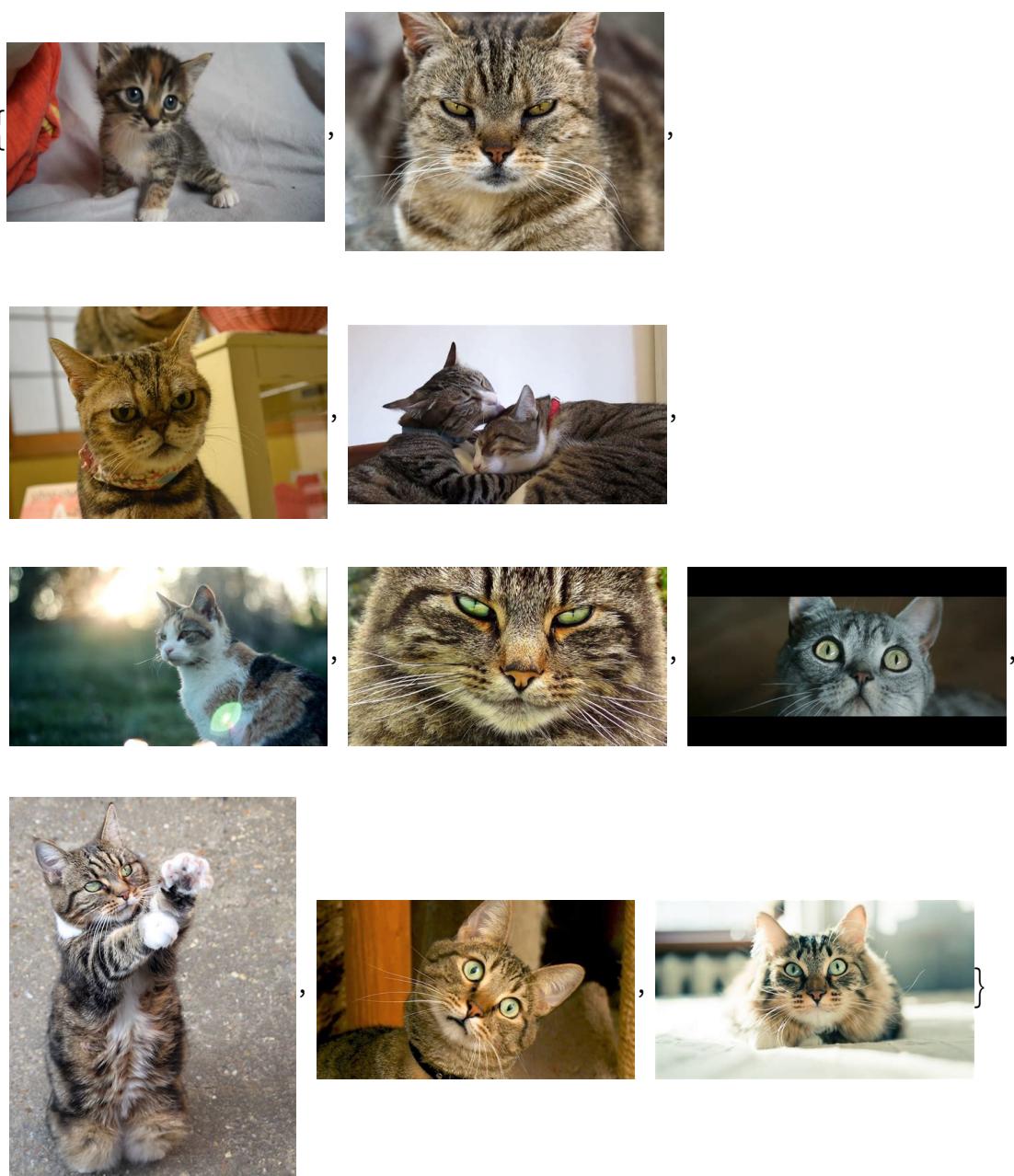
Vamos buscar imagens na Internet para treinamento:

```
In[14]:= dog = WebImageSearch ["dog", "Thumbnails"]
```

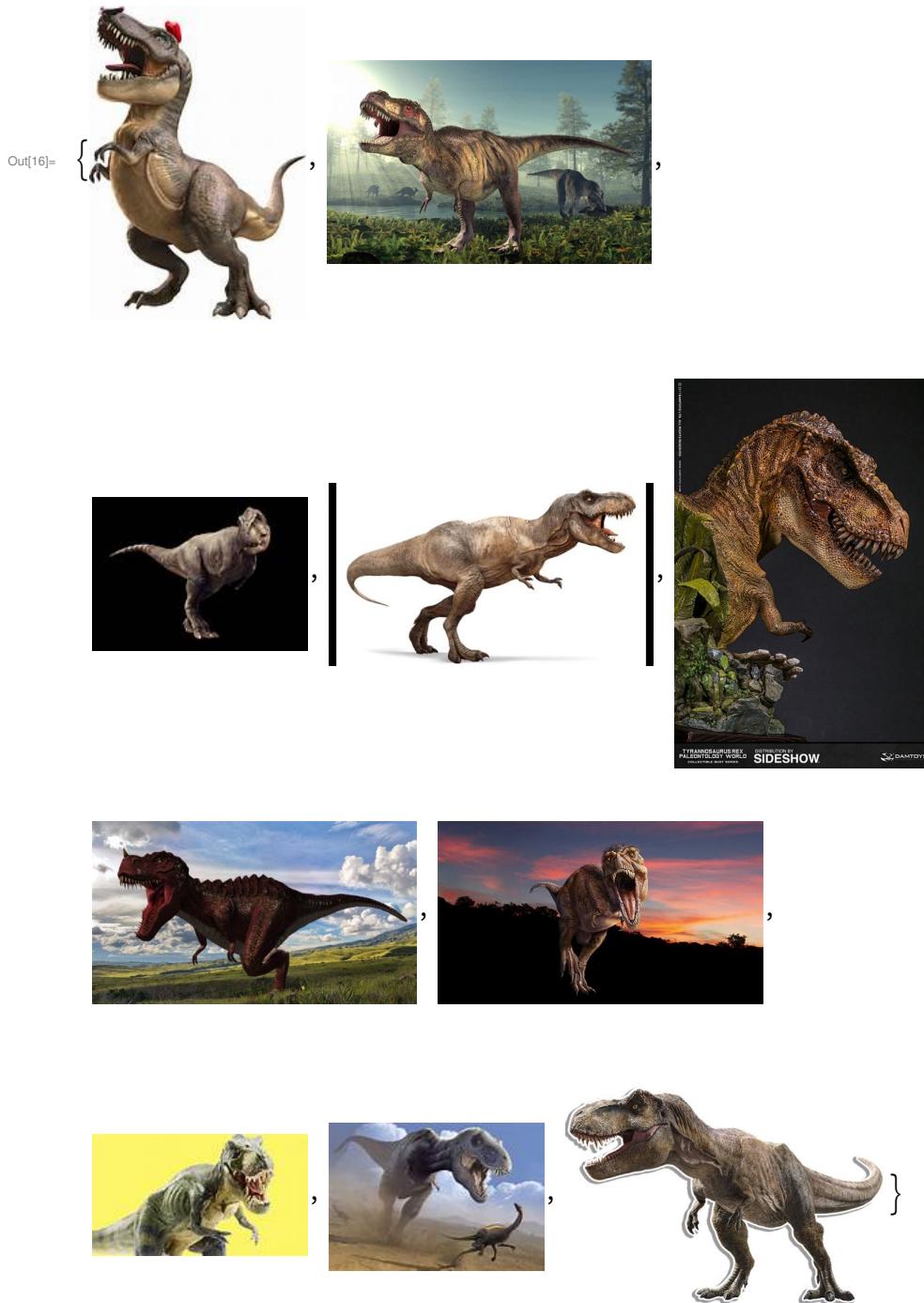


```
In[15]:= cat = WebImageSearch ["cat", "Thumbnails "]
```

Out[15]=



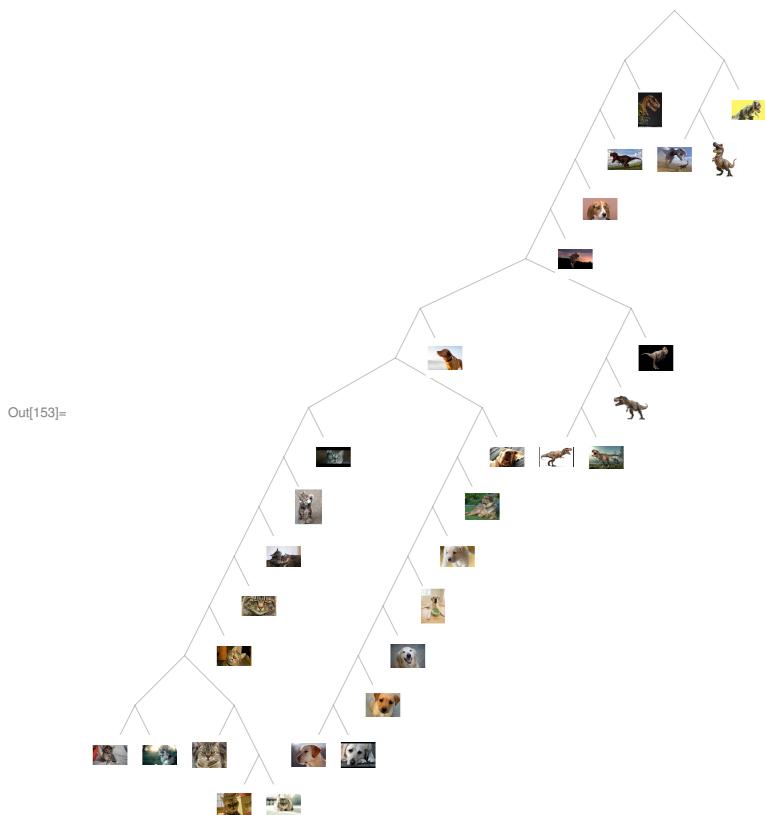
```
In[16]:= t rex = WebImageSearch ["t-rex", "Thumbnails "]
```



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



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

```
In[17]:= animal = Classify[<
  "cão" → dog,
  "gato" → cat,
  "t-rex" → trex
  >]
```

```
Out[17]= ClassifierFunction [
  Input type: Image
  Classes : cão, gato, t-rex
  Method : LogisticRegression
  Number of training examples : 30
  Data not in notebook ; Store now » ]
```

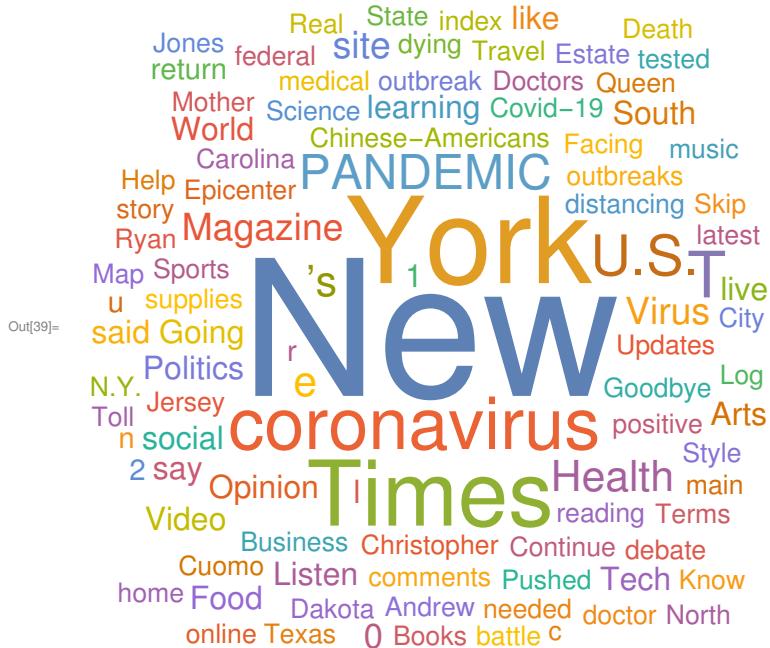
Teste com imagem de um lagarto...

```
In[18]:= animal[]
```

```
Out[18]= t-rex
```

O que está pegando no NYT hoje?

In[39]:= WordCloud[DeleteStopwords[Import["http://www.nytimes.com"]]]



Função em Wolfram Language que implementei e foi publicada:

In[41]:= ResourceFunction["SequenceGraph"]

Out[41]= SequenceGraph

In[156]:= Range[10]

Out[156]= SequenceGraph

Out[156]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

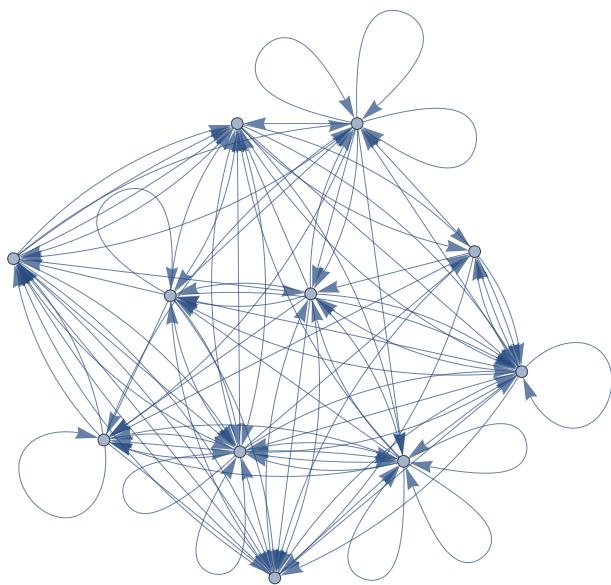
Out[157]= A sequence graph consisting of ten nodes arranged horizontally, connected by arrows pointing from left to right.

```
In[44]:= RandomInteger [10, 100]
```

[✖] SequenceGraph [✖]

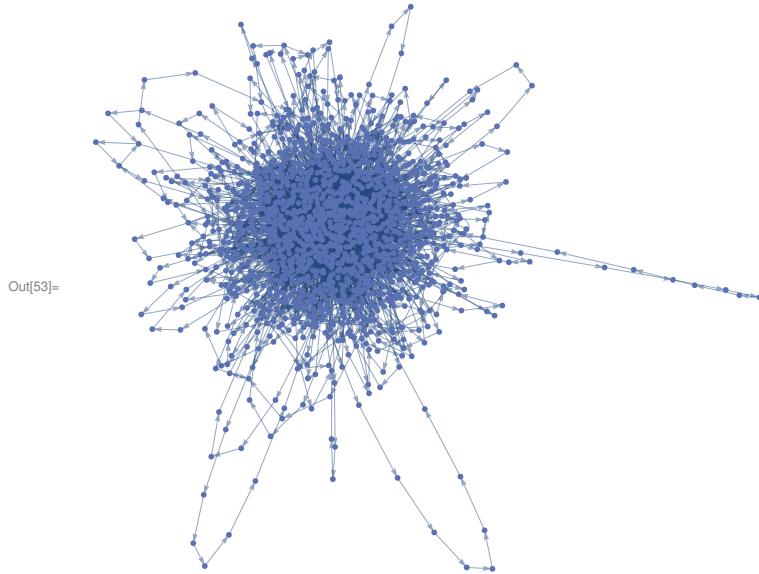
```
Out[44]= {2, 3, 6, 9, 1, 6, 0, 2, 2, 10, 6, 4, 7, 6, 2, 4, 3, 10, 0, 7, 0, 5, 2, 0,
3, 8, 7, 10, 5, 5, 9, 8, 0, 3, 2, 7, 0, 1, 4, 6, 2, 9, 7, 0, 3, 4, 5, 6, 10,
9, 7, 4, 7, 4, 8, 1, 4, 2, 8, 5, 6, 3, 2, 8, 8, 5, 5, 8, 3, 1, 3, 3, 5, 5, 0,
8, 8, 0, 1, 3, 8, 9, 4, 3, 1, 10, 10, 10, 10, 6, 4, 7, 8, 5, 9, 10, 7, 8, 9, 9}
```

Out[45]=



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

```
In[53]:= ResourceFunction["SequenceGraph"][[StringSplit[
DeleteStopwords[ResourceData["Alice in Wonderland"]], RegularExpression ["\\W+"]]]]
```



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

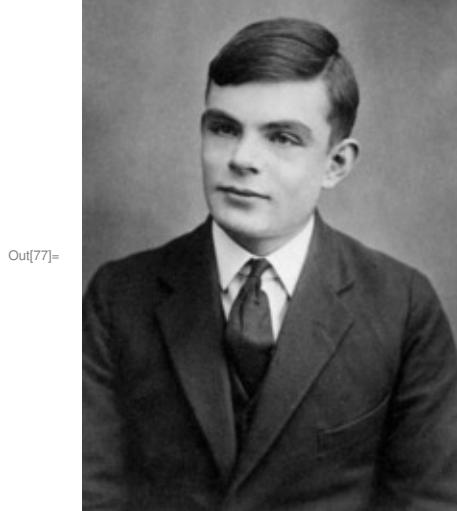
```
In[56]:= ResourceFunction["DuckDuckGoQuery"]
```

Out[56]= +

```
In[76]:=  + ["Alan Turing"]["Image"]
```

Out[76]= <https://duckduckgo.com/i/a377e75e.jpg>

```
In[77]:= atImage = Import[%]
```



In[78]:= **Classify**["NotablePerson ", atImage]

Out[78]= **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[103]:= **Alan Turing** ["Dataset"]

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	{  ,  ,  ,  }
mother	{Ethel Sara Stoney }
  rows 1–20 of 42   	

In[79]:= **FindFaces**[atImage, "Image"]



Processamento de imagens:

In[80]:= **ColorNegate**[**EdgeDetect**[**Blur**[atImage, 5], 5]]



In[97]:= **ImageIdentify**[atImage]

Out[97]=

person

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

Brazil COUNTRY

In[98]:=

Brazil COUNTRY ["Flag"]

Out[98]=



In[99]:=

Brazil COUNTRY ["GDP"]

Out[99]= $\$1.86863 \times 10^{12}$ per year

In[100]:= **China** COUNTRY [*GDP* + nominal] / **Brazil** COUNTRY [*GDP* + nominal]

Out[100]= 7.28244

In[101]:= **United States** COUNTRY [*dependencies*]

Out[101]= { **American Samoa** , **Guam** , **Northern Mariana Islands** ,
Puerto Rico , **United States Minor Outlying Islands** , **United States Virgin Islands** }

In[102]:= **Brazil** COUNTRY [*dependencies*]

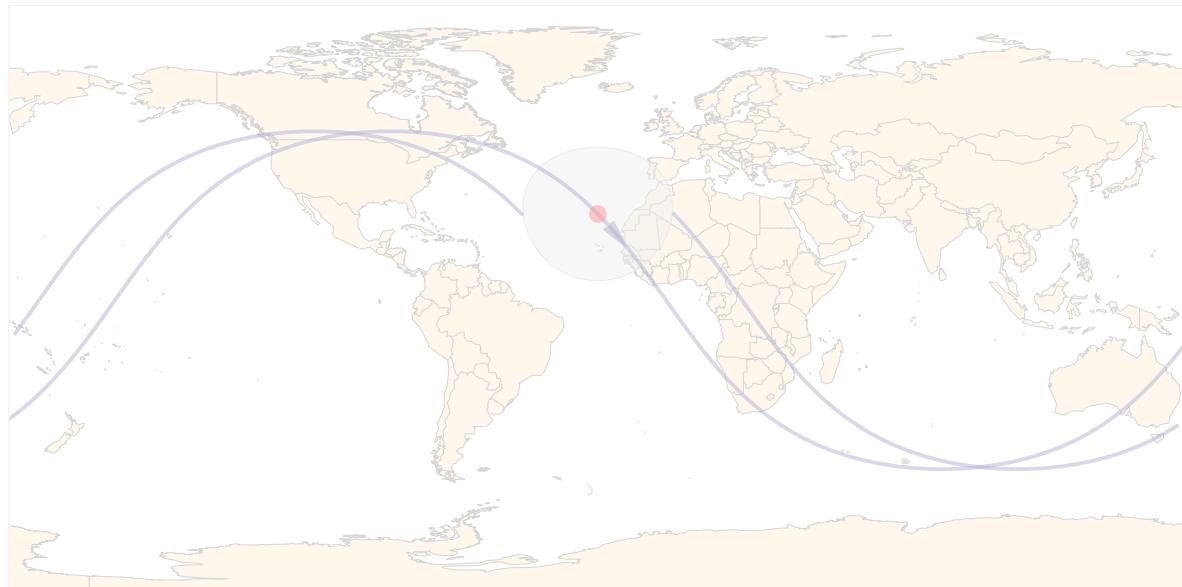
Out[102]= {}

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:

Where is the ISS now? +
↳ Results (1 of 3)

Out[55]=



In[54]:= **What is the meaning of life?** » +

↳ Result

Out[54]= 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[108]:= **CloudPublish []**

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

In[111]:= **URLShorten ["www.wolframcloud.com/obj/dcarvalho/Published/Insper-Dica-010B.nb"]**

Out[111]= <https://wolfr.am/LAR7Rrxt>

In[112]:= **BarcodeImage ["https://wolfr.am/LAR7Rrxt", "QR"]**



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/>