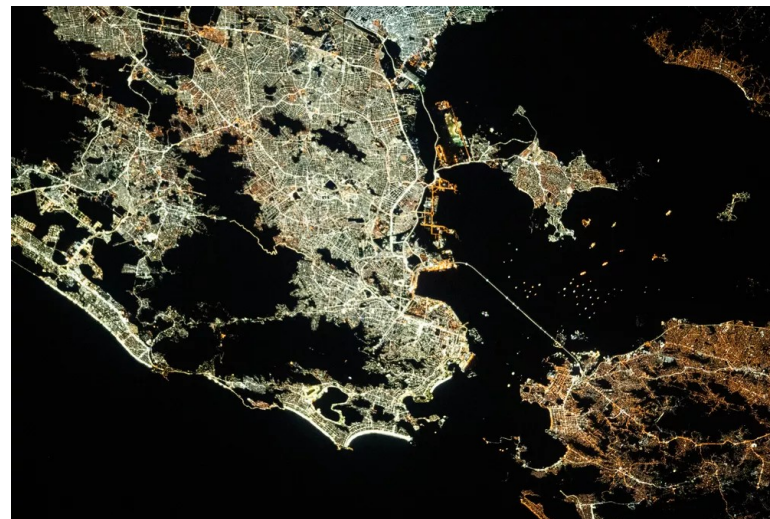


Programação em R para Geoprocessamento



Pedro José Farias Fernandes

pj_fernandes@id.uff.br

Laboratório de Geografia Física

Quem sou eu?

- Pedro José Farias Fernandes
- Geógrafo lotado no Departamento de Geografia (UFF) e membro do LAGEF desde 2012
- Formado pela UFF (2010)
- Mestre em Sensoriamento Remoto (INPE - 2013)
- Esp. em Geoprocessamento (PUC Minas – 2016)
- Esp. em Ciência de Dados (PUC Minas - 2021)
- Doutor em Geografia (UFF - 2019)
- Interesses atuais: R, Python, Data Science, Machine Learning e Desenvolvimento Web e Mobile.
- pj_fernandes@id.uff.br

Quem são vocês?



O papel da técnica na Geografia



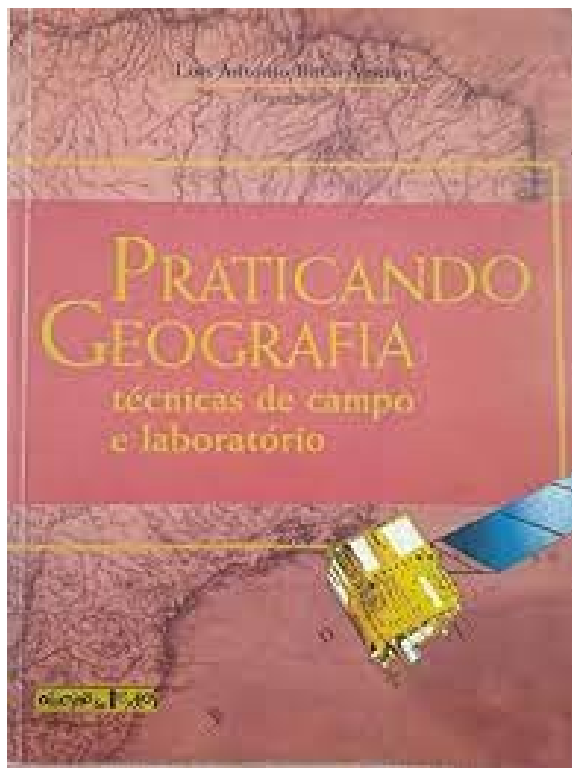
O uso da técnica possibilita a obtenção e o processamento de dados sobre a realidade **“que embasarão os caminhos percorridos pelo método”**

A técnica se desenvolve no plano do fazer.

Trabalho técnico x trabalho científico

Quando o fazer está vinculado a um processo de pesquisa guiado por um método, temos um trabalho científico.

O papel da técnica na Geografia



A obtenção dos dados está sempre vinculada a um objetivo e a um problema. **NADA DE SAIR COLETANDO DADOS ALEATÓRIOS E SEM SENTIDO POR AÍ.**

O pesquisador pode não ter um domínio profundo da técnica, mas é importante que possua uma noção para que, junto com a definição da problemática e dos objetivos da pesquisa, possa pedir ajudar ou terceirizar essa parte.

“É interessante considerar que a falta de tal domínio poderá levá-lo perder espaço no mercado de trabalho fora do meio acadêmico”. (VENTURI, 2005).

AVISO

- Não se deve desprezar técnicas clássicas por não serem “sofisticadas”.
- É importante conhecer o maior número possível e aplicar de acordo com o seu problema.

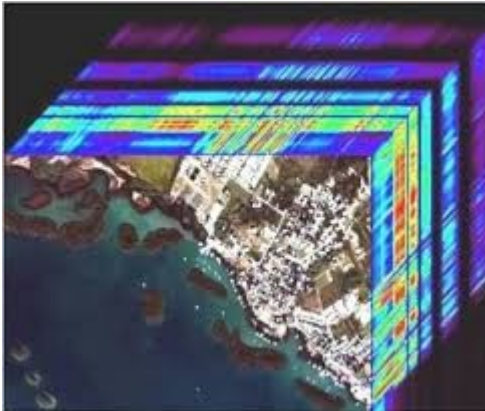


Big Data e Geografia



Avanços tecnológicos: processamento de dados, disponibilidade de dados, softwares, linguagens de programação de fácil uso, redes neurais, etc.

BIG DATA: Como processar e integrar grandes bases de dados de diferentes fontes para realizar uma análise espacial?



Muitas ferramentas criadas a partir de 2015 para lidar com esse cenário.

O que é o R?

- Uma linguagem orientada a objetos criada em 1996. Permite a realização de análises estatísticas, cálculos e criação de gráficos.
- É gratuita
- 7429 pacotes em julho de 2017. 18407 pacotes em <https://cran.r-project.org/web/packages/>(21/07/2022)



EMPREGOS E CARREIRAS

Salários para trabalhar com dados vão de R\$ 7 mil a R\$ 22 mil, diz pesquisa

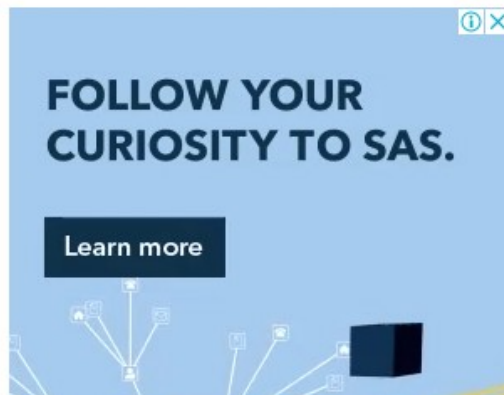
Claudia Varella

Colaboração para o UOL, em São Paulo

17/06/2021 10h00

Os salários de profissionais da área de dados, como analistas, engenheiros e cientistas, vão de R\$ 7.000 a R\$ 22 mil no mercado de trabalho, de acordo com pesquisa feita pela empresa Intera, "rhtech" de recrutamento digital. Foram ouvidos cerca de 4.000 profissionais de 34 corporações —entre [startups](#) e empresas em transformação digital— de todo o país, em fevereiro e marco deste ano.

PUBLICIDADE



**FOLLOW YOUR
CURIOSITY TO SAS.**

[Learn more](#)

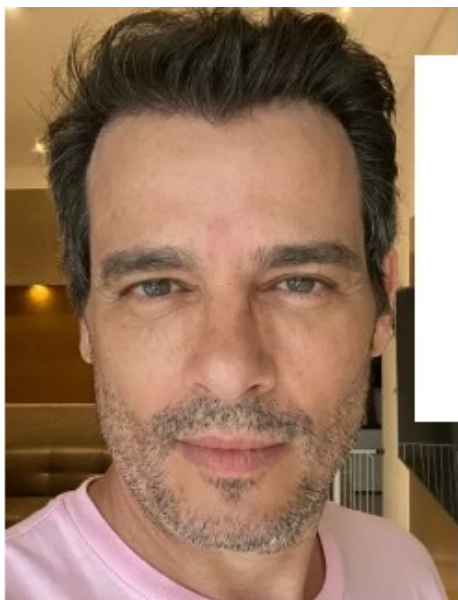
2 Organização e apresentação de variáveis. 3 Métodos para sumarização e análise exploratória de dados. 3.1 Distribuição de frequências: absoluta, relativa, acumulada. 3.2 Medidas de posição: média, moda, mediana e quartis. 3.3 Medidas de dispersão: amplitude, variância, desvio-padrão, coeficiente de variação, amplitude interquartil. 3.4 Correlação. 3.5 Histogramas e curvas de frequência. 3.6 Diagrama de caixa (*boxplot*) e identificação de valores atípicos (*outliers*). 3.7 Diagrama de dispersão. 4 Análise de dados categorizados. 5 Distribuições de probabilidade. Distribuição binomial. Distribuição normal. 6 Inferência estatística. Estimativa de parâmetros por ponto e por intervalo. Intervalo de confiança. Testes de hipóteses. Testes paramétricos: médias e proporções. 7 Análise de regressão linear. 8 Técnicas de Amostragem. 9 Análise multivariada. 10 Séries Temporais: componentes estruturais das séries temporais e médias móveis.

ANÁLISE DE DADOS: 1 Dados estruturados e não estruturados. Dados abertos. Coleta, tratamento, armazenamento, integração e recuperação de dados. Processos de ETL. Formatos e tecnologias: XML, JSON, CSV. Representação de dados numéricos, textuais e estruturados; aritmética computacional. Representação de dados espaciais para georeferenciamento e geosensoriamento. 2 Bancos de dados relacionais: teoria e implementação. Uso do SQL como DDL, DML, DCL. Processamento de transações. 3 Exploração de dados: conceitualização e características. Noções do modelo CRISP-DM. Técnicas para pré-processamento de dados. Técnicas e tarefas de mineração de dados. Classificação. Regras de associação. Análise de agrupamentos (clusterização). Detecção de anomalias. Modelagem preditiva. 4 Conceitos de PLN: semântica vetorial, redução de dimensionalidade, modelagem de tópicos latentes, classificação de textos, análise de sentimentos,



representações com n-gramas. 5 Conceitos de ML: fontes de erro em modelos preditivos, validação e avaliação de modelos preditivos, *underfitting*, *overfitting* e técnicas de regularização, otimização de hiperparâmetros, separabilidade de dados, redução da dimensionalidade. Modelos lineares, árvores de decisão, redes neurais *feed-forward*, classificador Naive Bayes. 6 Linguagem Python: sintaxe, variáveis, tipos de dados e estruturas de controle de fluxo. Estruturas de dados, funções e arquivos. Bibliotecas: NLTK, Tensor Flow, Pandas, Numpy, Arrow, Sklearn, Scipy. 7 Noções da Linguagem R. Sintaxe, tipos de dados, operadores, comandos de repetição, estruturas de dados, gráficos, *Data frames*. Tidyverse. 8 Pareamento de dados (*record linkage*). Processo e etapas. Classificação. Qualidade de dados pareados. Análise de dados pareados. 9 Segurança da informação: Confidencialidade, integridade, disponibilidade, autenticidade e não repúdio. Políticas de segurança. Políticas de classificação da informação. Sistemas de gestão de segurança da informação. Tratamento de incidentes de segurança da informação. 10 Lei de Acesso à Informação (Lei nº 12.527/2011): conceitos e aplicação. Lei 13.709/2018, Lei Geral de Proteção de Dados Pessoais (LGPD).

Cargo	Remuneração Inicial
AUFC-CE	R\$ 21.947,82



Celso Portioli publicou foto no Instagram

Imagem: Reprodução/Instagram @celsoportioli

Celso Portioli encontra novo hobby e faz curso de Python

Colaboração para o UOL, em São Paulo

03/02/2021 15h46

Celso Portioli está expandindo seus conhecimentos no mundo da programação. O apresentador anda compartilhando com os seguidores

algumas de suas novas descobertas com o curso de Python.

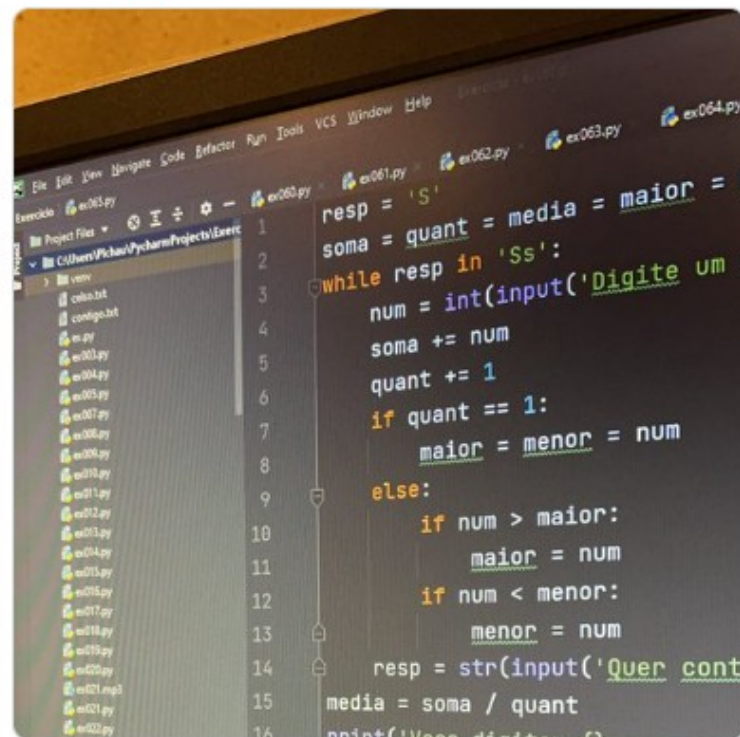


Celso Portioli

@celsoportioli · Seguir



Python ❤️



6:55 PM · 2 de fev de 2021



6 mil



Responder



Copiar link

[Ler 350 respostas](#)

2 Organização e apresentação de variáveis. 3 Métodos para sumarização e análise exploratória de dados. 3.1 Distribuição de frequências: absoluta, relativa, acumulada. 3.2 Medidas de posição: média, moda, mediana e quartis. 3.3 Medidas de dispersão: amplitude, variância, desvio-padrão, coeficiente de variação, amplitude interquartil. 3.4 Correlação. 3.5 Histogramas e curvas de frequência. 3.6 Diagrama de caixa (*boxplot*) e identificação de valores atípicos (*outliers*). 3.7 Diagrama de dispersão. 4 Análise de dados categorizados. 5 Distribuições de probabilidade. Distribuição binomial. Distribuição normal. 6 Inferência estatística. Estimativa de parâmetros por ponto e por intervalo. Intervalo de confiança. Testes de hipóteses. Testes paramétricos: médias e proporções. 7 Análise de regressão linear. 8 Técnicas de Amostragem. 9 Análise multivariada. 10 Séries Temporais: componentes estruturais das séries temporais e médias móveis.

ANÁLISE DE DADOS: 1 Dados estruturados e não estruturados. Dados abertos. Coleta, tratamento, armazenamento, integração e recuperação de dados. Processos de ETL. Formatos e tecnologias: XML, JSON, CSV. Representação de dados numéricos, textuais e estruturados; aritmética computacional. Representação de dados espaciais para georeferenciamento e geosensoriamento. 2 Bancos de dados relacionais: teoria e implementação. Uso do SQL como DDL, DML, DCL. Processamento de transações. 3 Exploração de dados: conceitualização e características. Noções do modelo CRISP-DM. Técnicas para pré-processamento de dados. Técnicas e tarefas de mineração de dados. Classificação. Regras de associação. Análise de agrupamentos (clusterização). Detecção de anomalias. Modelagem preditiva. 4 Conceitos de PLN: semântica vetorial, redução de dimensionalidade, modelagem de tópicos latentes, classificação de textos, análise de sentimentos,



representações com n-gramas. 5 Conceitos de ML: fontes de erro em modelos preditivos, validação e avaliação de modelos preditivos, *underfitting*, *overfitting* e técnicas de regularização, otimização de hiperparâmetros, separabilidade de dados, redução da dimensionalidade. Modelos lineares, árvores de decisão, redes neurais *feed-forward*, classificador Naive Bayes. 6 Linguagem Python: sintaxe, variáveis, tipos de dados e estruturas de controle de fluxo. Estruturas de dados, funções e arquivos. Bibliotecas: NLTK, Tensor Flow, Pandas, Numpy, Arrow, Sklearn, Scipy. 7 Noções da Linguagem R. Sintaxe, tipos de dados, operadores, comandos de repetição, estruturas de dados, gráficos, *Data frames*. Tidyverse. 8 Pareamento de dados (*record linkage*). Processo e etapas. Classificação. Qualidade de dados pareados. Análise de dados pareados. 9 Segurança da informação: Confidencialidade, integridade, disponibilidade, autenticidade e não repúdio. Políticas de segurança. Políticas de classificação da informação. Sistemas de gestão de segurança da informação. Tratamento de incidentes de segurança da informação. 10 Lei de Acesso à Informação (Lei nº 12.527/2011): conceitos e aplicação. Lei 13.709/2018, Lei Geral de Proteção de Dados Pessoais (LGPD).

Cargo	Remuneração Inicial
AUFC-CE	R\$ 21.947,82

2 Organização e apresentação de variáveis. 3 Métodos para sumarização e análise exploratória de dados. 3.1 Distribuição de frequências: absoluta, relativa, acumulada. 3.2 Medidas de posição: média, moda, mediana e quartis. 3.3 Medidas de dispersão: amplitude, variância, desvio-padrão, coeficiente de variação, amplitude interquartil. 3.4 Correlação. 3.5 Histogramas e curvas de frequência. 3.6 Diagrama de caixa (*boxplot*) e identificação de valores atípicos (*outliers*). 3.7 Diagrama de dispersão. 4 Análise de dados categorizados. 5 Distribuições de probabilidade. Distribuição binomial. Distribuição normal. 6 Inferência estatística. Estimação de parâmetros por ponto e por intervalo. Intervalo de confiança. Testes de hipóteses. Testes paramétricos: médias e proporções. 7 Análise de regressão linear. 8 Técnicas de Amostragem. 9 Análise multivariada. 10 Séries Temporais: componentes estruturais das séries temporais e médias móveis.

ANÁLISE DE DADOS: 1 Dados estruturados e não estruturados. Dados abertos. Coleta, tratamento, armazenamento, integração e recuperação de dados. Processos de ETL. Formatos e tecnologias: XML, JSON, CSV. Representação de dados numéricos, textuais e estruturados; aritmética computacional. Representação de dados espaciais para georeferenciamento e geosensoriamento. 2 Bancos de dados relacionais: teoria e implementação. Uso do SQL como DDL, DML, DCL. Processamento de transações. 3 Exploração de dados: conceitualização e características. Noções do modelo CRISP-DM. Técnicas para pré-processamento de dados. Técnicas e tarefas de mineração de dados. Classificação. Regras de associação. Análise de agrupamentos (clusterização). Detecção de anomalias. Modelagem preditiva. 4 Conceitos de PLN: semântica vetorial, redução de dimensionalidade, modelagem de tópicos latentes, classificação de textos, análise de sentimentos,



representações com n-gramas. 5 Conceitos de ML: fontes de erro em modelos preditivos, validação e avaliação de modelos preditivos, *underfitting*, *overfitting* e técnicas de regularização, otimização de hiperparâmetros, separabilidade de dados, redução da dimensionalidade. Modelos lineares, árvores de decisão, redes neurais *feed-forward*, classificador Naive Bayes. 6 Linguagem Python: sintaxe, variáveis, tipos de dados e estruturas de controle de fluxo. Estruturas de dados, funções e arquivos. Bibliotecas: NLTK, Tensor Flow, Pandas, Numpy, Arrow, Sklearn, Scipy. 7 Noções da Linguagem R. Sintaxe, tipos de dados, operadores, comandos de repetição, estruturas de dados, gráficos, *Data frames*. Tidyverse. 8 Pareamento de dados (*record linkage*). Processo e etapas. Classificação. Qualidade de dados pareados. Análise de dados pareados. 9 Segurança da informação: Confidencialidade, integridade, disponibilidade, autenticidade e não repúdio. Políticas de segurança. Políticas de classificação da informação. Sistemas de gestão de segurança da informação. Tratamento de incidentes de segurança da informação. 10 Lei de Acesso à Informação (Lei nº 12.527/2011): conceitos e aplicação. Lei 13.709/2018, Lei Geral de Proteção de Dados Pessoais (LGPD).

Cargo	Remuneração Inicial
AUFC-CE	R\$ 21.947,82

Motivacional

- Programar é importante para realizar tarefas braçais de uma vez só.
- Não depender de softwares pagos para fazer suas análises.
- Resultados são gerados rapidamente.
- Você pode criar suas funções para resolver os seus problemas.
- Data Science está na moda (2548 resultados de vagas no LinkedIn Brasil em 21/07/2022)
- R saiu no New York Times http://www.nytimes.com/2009/01/07/technology/business-computing/07program.html?pagewanted=all&_r=0

“Companies as diverse as Google, Pfizer, Merck, Bank of America, the InterContinental Hotels Group and Shell use it.”

Besteiras que já ouvi sobre R

- “R é caixa preta.”
- “R é coisa de engenheiro”
- “Você não deveria ter feito geografia, e sim ciência da computação”

Onde aprender?

- <http://ecologia.ib.usp.br/bie5782/doku.php?id=start>
- <http://www.cookbook-r.com/>
- <https://cran.r-project.org/manuals.html>
- Coursera: R programming
- <https://pakillo.github.io/R-GIS-tutorial/>
- <https://www.codecademy.com/>
- Google it!!!
- Youtube it!!!



Instalação do R

1) Primeiro passo:

<https://cran.r-project.org/bin/windows/base/>

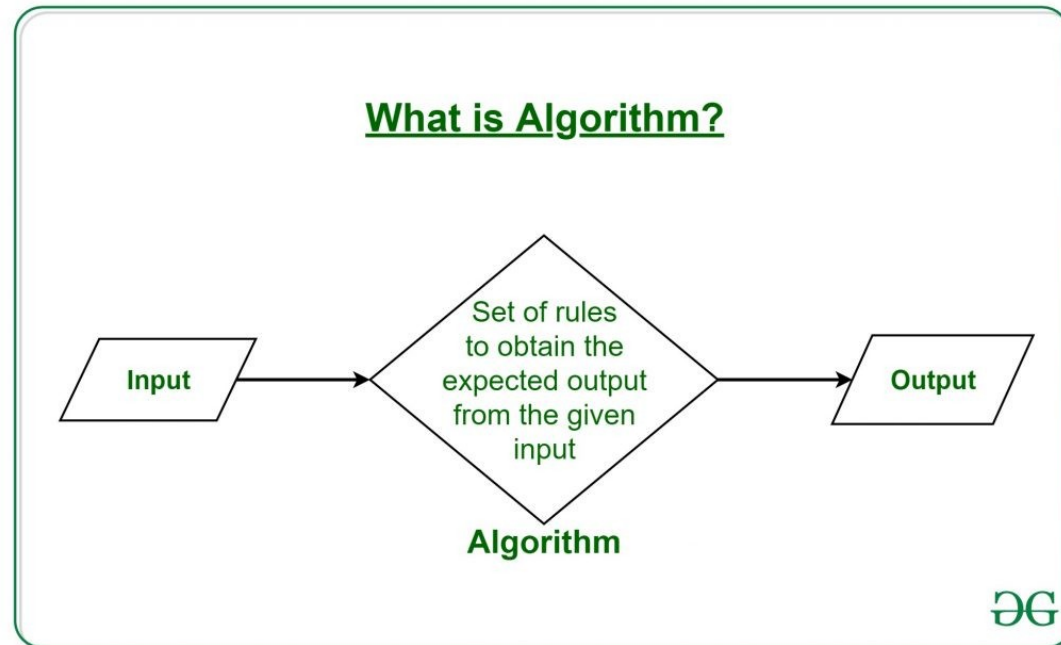
2) Segundo passo:

Baixar o Rstudio (versão free):

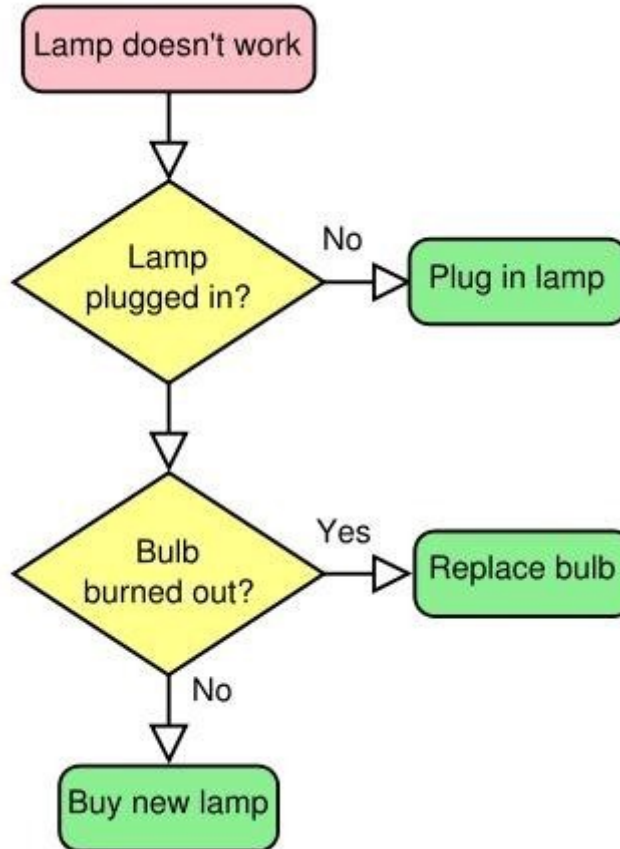
<https://www.rstudio.com/products/rstudio/download/>

Algoritmo

“Um processo sistemático para a resolução de um problema”
(SZWARCFITER; MARKEZON, 2010).



Algoritmo



Algoritmo

- Encontrar a posição do valor 2 na sequência: (1, 6, 7, 2, 9):
- Para cada item do conjunto:
 - Item = 2?
 - Se sim: retorna índice da localização
 - Se não: continue

Algoritmo

```
conjunto<- c(1, 6, 7, 2, 9)
```

```
for (i in 1:length(conjunto)) {  
  if (conjunto[i] == 2) {  
    print(i)  
  } else {  
    next  
  }  
}
```

Estrutura sequencial

- Início

- Ação 1

`x<-2`

- Ação 2

`y<-x+2`

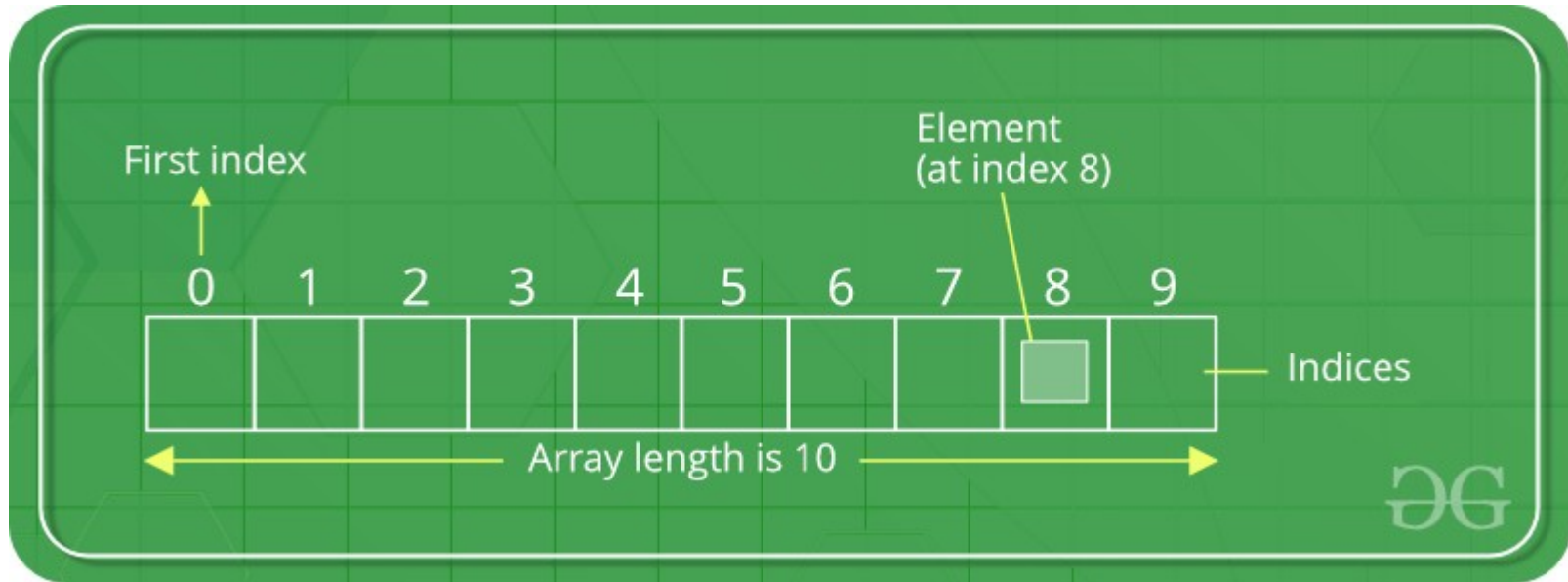
- Ação 3

`print(x*y)`

- Fim

Estrutura de dados composta

- Vetor



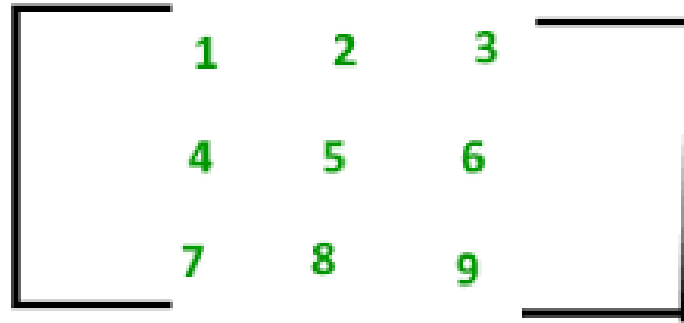
Vetor

```
x<-seq(1,10,2)
c<-seq(5,10,length=30)
y<-c(1,2,3,4,5)
z<-1:10
a<-"blablabla"
b<-c("a","b","c")
d<-rep(c(0,1,2),times=10)
```

```
y[1]
y[c(1,2)]
y[1]<-2
```

```
ls() #comando para listar os objetos
rm(a) #remove o objeto a
rm(list=ls()) #remove tudo
```


Matriz



1	2	3
4	5	6
7	8	9

<https://www.geeksforgeeks.org/matrix/>

Matriz

```
x<-1:9
```

```
m<-matrix(x,ncol=3,nrow=3)
```

```
m2<-matrix(x,ncol=3,nrow=3,byrow=T)
```

```
m+m2
```

```
t(m^m2)
```

```
m[1,1]
```

```
m[,1]
```

```
m[c(1,2),]
```

```
colSums(m)
```

Determinante matriz (Regra de Sarrus)

$$\begin{vmatrix} 4 & 5 & -3 \\ 2 & 1 & 0 \\ 3 & -1 & 1 \end{vmatrix}$$

Products of the first diagonal (blue): -9 , 0 , 10

Products of the second diagonal (red): 4 , 0 , 6

Products of the third diagonal (red): -9 , 1 , 3

Final result: 10

```
matriz<-matrix(c(4,5,-3, 2,1,0, 3,-1,1), ncol=3, nrow=3, byrow=T)  
det(matriz)
```

Matriz inversa

$$A = \begin{bmatrix} 4 & 1 \\ 2 & 0 \end{bmatrix}$$

$$A^{-1} = ?$$

$$A \cdot A^{-1} = I$$

$$\begin{bmatrix} 4 & 1 \\ 2 & 0 \end{bmatrix} \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\begin{cases} 4a + 1c = 1 \\ 4b + 1d = 0 \\ 2a + 0c = 0 \\ 2b + 0d = 1 \end{cases} \Rightarrow \begin{bmatrix} a=0 & b=-\frac{1}{2} \\ c=1 & d=2 \end{bmatrix}$$

```
m<-matrix(c(4,1,2,0),nrow=2,ncol=2,byrow=T)
solve(matriz)
```

	[,1]	[,2]
[1,]	0	0.5
[2,]	1	-2.0

Lista

```
lista<-list(x,m,rnorm(10))
```

```
pes<-list(idade=41,nome="James  
Bond",notas=c(98,95,100))
```

```
lista[1]
```

```
lista[[2]]
```

```
is.list(lista[1])
```

```
is.vector(lista[[2]])
```

Data Frame

```
df<-data.frame(nome=c("João","Bruno","Roberto"), indice=1:3)
```

```
df[,2]
```

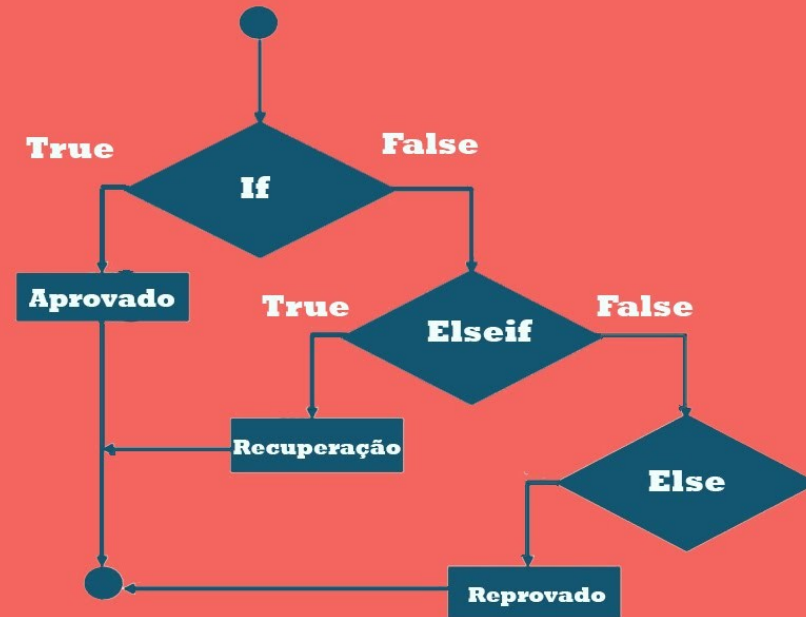
```
df[1,2]
```

```
df$indice
```

```
df$indice[1]
```

IF ELSE

if - elseif - else



IF ELSE

```
idade<-16
```

```
if (idade >= 16) {  
  print("Pode votar")  
} else {  
  print("Não pode votar")  
}
```


For loop



For loop

```
pizza <- c("portuguesa", "quatro queijos", "marguerita")
```

```
for (i in 1:length(pizza)) {  
  print(pizza[i])  
}
```

```
for (fatia in pizza) {  
  print(fatia)  
}
```

For loop (valor máximo)

Qual é o maior valor do conjunto (3, 41, 12, 9, 74, 15)?

- Valor = 0
- Para cada item de conjunto
 - Item > valor ?
 - Se sim, valor := item

```
maior<-0
```

```
x<-c(3,41,12,9,74,15)
```

```
for (i in 1:length(x)) {
```

```
  if (x[i]>maior){
```

```
    maior <-x[i]
```

```
  }
```

```
}
```

```
print(maior)
```

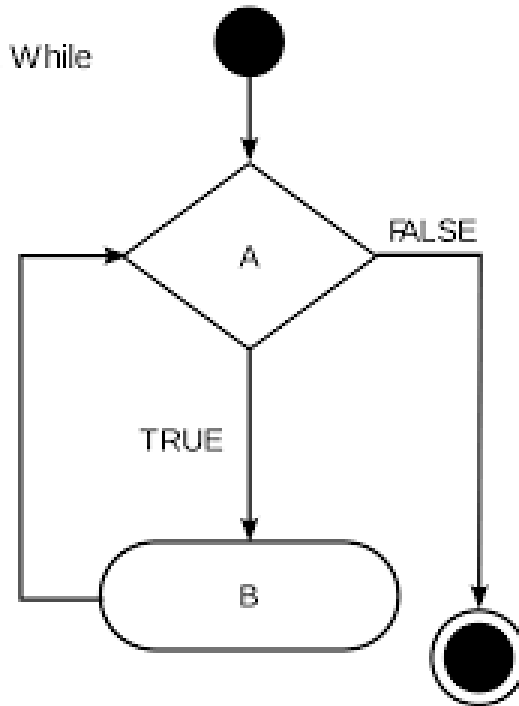
```
identical(maior,max(x))
```

While

Enquanto alguma condição for verdadeira:

- Execute algo

While (A= TRUE) Do
B
End While



https://en.wikipedia.org/wiki/While_loop

While

```
dado<-1:6
```

```
jogada<-sample(dado, 1)
```

```
while(jogada != 5) {
```

```
  print(jogada)
```

```
  jogada<-sample(dado, 1)
```

```
}
```

Jogue o dado até tirar o número 5



While para o cálculo de fatorial

```
#####while fatorial
```

```
x<-1
```

```
fatorial<-1
```

```
while(x <= 5) {
```

```
  fatorial <- fatorial * x
```

```
  x <- x + 1
```

```
}
```

```
print(fatorial)
```

```
fatorial(5)
```

$$0! = 1$$

$$1! = 1$$

$$2! = 2 \times 1$$

$$3! = 3 \times 2 \times 1$$

$$4! = 4 \times 3 \times 2 \times 1$$

$$5! = 5 \times 4 \times 3 \times 2 \times 1$$

$$6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1$$

$$7! = 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$$

$$8! = 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$$

$$9! = 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$$

$$10! = 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$$

Funções

“A function is a **block of organized code that is used to perform a single task**. They provide better modularity for your application and reusability”

https://en.wikiversity.org/wiki/Programming_Fundamentals/Functions

Funções

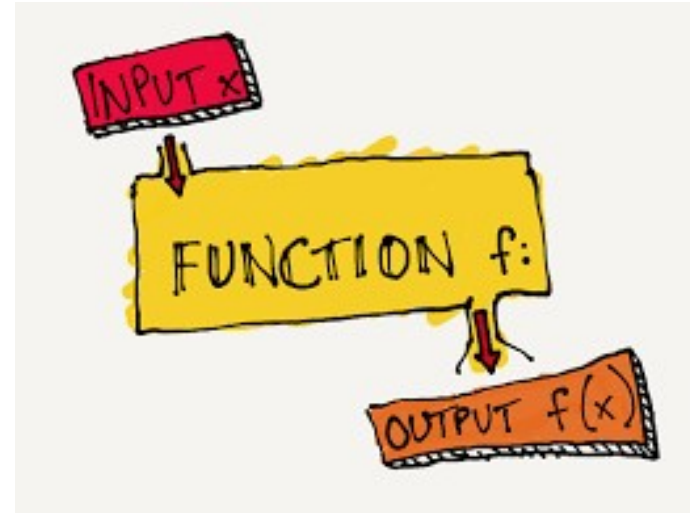
Print function example:

```
print("hello world")
```

Annotations:

- Function Name: `print`
- Parameters separated by commas: `("hello world")`
- Parentheses contain parameters: `("hello world")`

<https://www.futurelearn.com/info/courses/programming-102-think-like-a-computer-scientist/0/steps/53095>



<https://seattlewebsitedevelopers.medium.com/functions-and-why-use-them-74cd32fc2d72>

Funções

parâmetros

/ \

```
function add(x, y) {  
    return x + y;  
}
```

```
sum = add(4, 1);
```

| /

argumentos

[https://
pt.stackoverflow.com/
questions/32448/qual-a-
diferen%C3%A7a-entre-
par%C3%A2metro-e-
argumento](https://pt.stackoverflow.com/questions/32448/qual-a-diferen%C3%A7a-entre-par%C3%A2metro-e-argumento)

Funções

```
##soma
```

```
soma <- function(x, y) {  
  return(x+y)  
}
```

```
resultado <- soma(2, 3)
```

Funções

Gerando números para a mega sena

##mega sena

```
megasena <- function() {  
  sequencia <- 1:60  
  sorteio <- sample(sequencia, 6)  
  return(sorteio)  
}
```



Funções (valor máximo)

```
maximo<-function(vetor) {  
  max <- numeric()  
  for (item in 1:length(vetor)) {  
    if (length(max) == 0 || max < vetor[item]) {  
      max <- vetor[item]  
    }  
  }  
  return(max)  
}
```

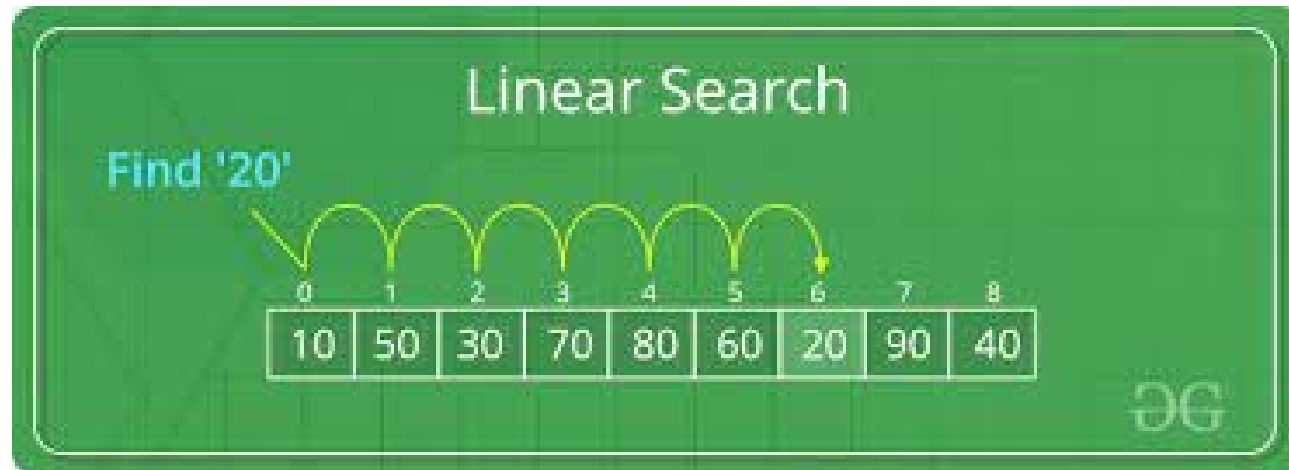
```
x<-c(59, 34, 2, 10, 25, 200, 123, 44)  
maximo(x)
```

Funções (exercícios)

- 1) Implemente função de subtração
- 2) Implemente função para encontrar min
- 3) Implemente função para calcular somatório

Função encontrar_valor()

- Algoritmo de busca linear
- “... pesquisa em vetores ou listas de modo sequencial, i. e., elemento por elemento ...”
(https://pt.wikipedia.org/wiki/Busca_linear)



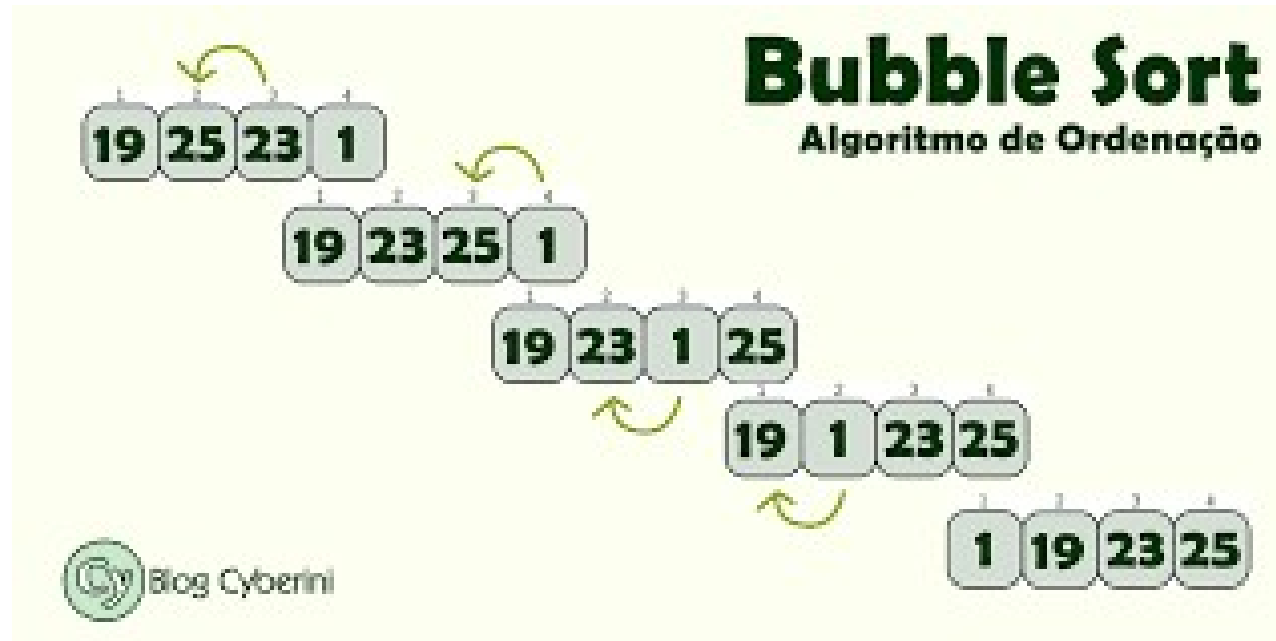
Função encontrar_valor()

```
encontrar_valor <- function(numero, vetor) {  
  resultado <- FALSE  
  
  for (i in 1:length(vetor)) {  
    if (numero == vetor[i]) {  
      resultado <- TRUE  
      #break  
    } else {  
      resultado <- FALSE  
    }  
  }  
  
  return(resultado)  
}
```

```
x<-c(59, 34, 2, 10, 25, 200, 123, 44)
```

```
encontrar_valor(2, x)  
encontrar_valor(200, x)  
encontrar_valor(2, x)  
encontrar_valor(57, x)
```

Desafio: ordenar valores



<https://www.blogcyberini.com/2018/02/bubble-sort.html>

Recursividade

- Um procedimento é recursivo quando contém “uma ou mais chamadas a si mesmo” (SZARCFITER; MARKENZON, 2011)
- Exemplo: fatorial de um inteiro ≥ 0
- Exercício: implemente o fatorial não recursivo



Fatorial

De quantas maneiras podemos organizar 5 alunos em uma fila?

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120 \text{ maneiras}$$

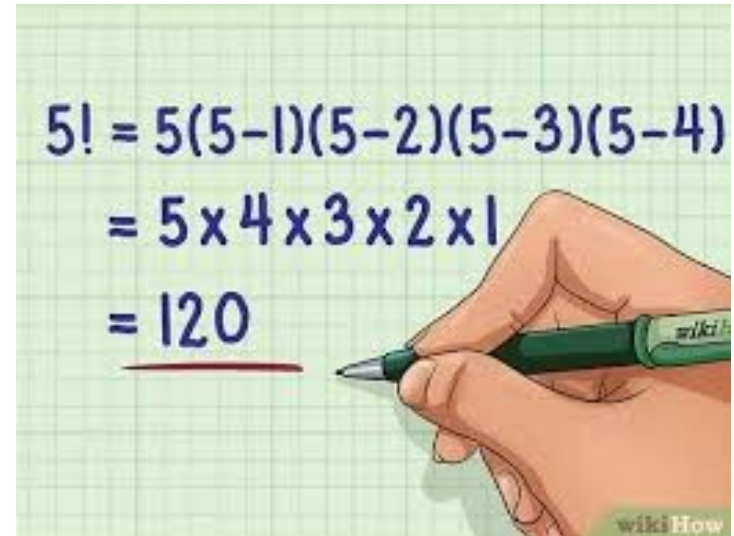
fatorial(n):

Se o n é igual a 0 ou 1

Resultado = 1

Caso contrário

retorne $n \times \text{fatorial}(n - 1)$



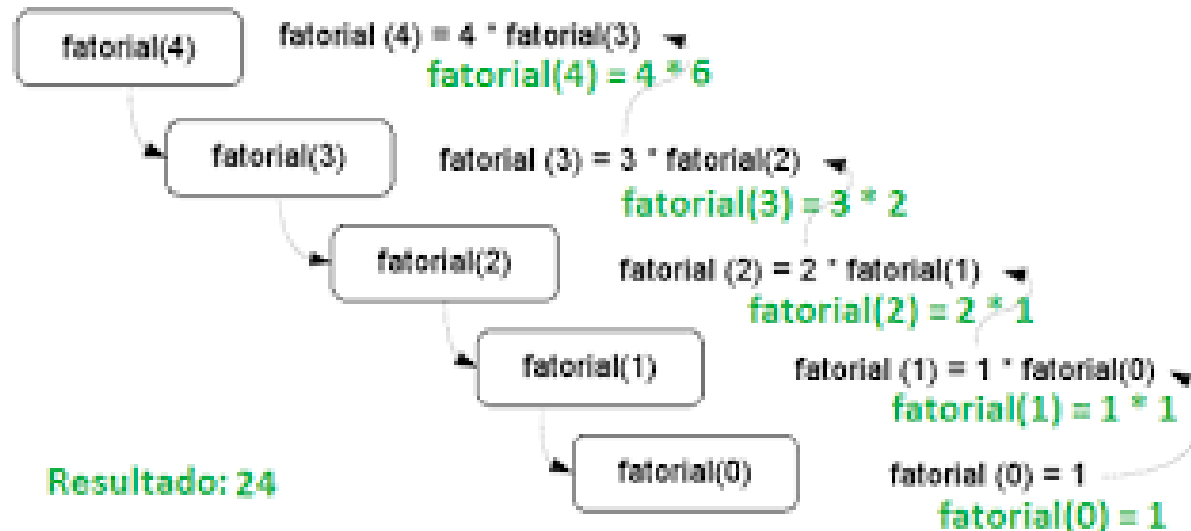
5! = 5(5-1)(5-2)(5-3)(5-4)
= 5 x 4 x 3 x 2 x 1
= 120

A hand holding a green pen is shown writing the final result '120' on a green grid background. The pen is positioned at the end of the underlined '120'.

wikiHow

Recursividade

*Lembrem-se que a leitura é de baixo para cima!



<http://www.linhadecodigo.com.br/artigo/3316/recursividade-em-java.aspx>

Recursividade

- Fatorial de 1 (1!) ou de 0 (0!) são iguais a 1

```
fatorial<-function(numero) {  
  if (numero == 0 || numero==1) {  
    return(1)  
  } else {  
    return(numero*fatorial(numero-1))  
  }  
}
```

```
fatorial(5)
```