

# Programação em R para Geoprocessamento



Pedro José Farias Fernandes  
[pj\\_fernandes@id.uff.br](mailto:pj_fernandes@id.uff.br)  
Laboratório de Geografia Física

Jairson Alberto Sami  
[jairsonalberto@id.uff.br](mailto:jairsonalberto@id.uff.br)  
Laboratório de Geografia Física

# Quem somos nós?

## **Pedro José Farias Fernandes**

- Geógrafo lotado no Depto. de Geografia (UFF) e membro do LAGEF desde 2012, e do projeto UFF Labs da equipe UFF Mobile.
- Formado pela UFF (2010)
- Msc. em Sensoriamento Remoto (INPE – 2013) e Dr. em Geografia (UFF - 2019)
- Esp. em Geoprocessamento (PUC Minas – 2016) e em Ciência de Dados (PUC Minas - 2021)
- Interesses atuais: R, Python, Data Science, Machine Learning e Desenvolvimento Web e Mobile.
- **`pj_fernandes@id.uff.br`**

## **Jairson Alberto Sami**

- Cientista Ambiental formado pela UFF (2022), membro do LAGEF desde 2022.
- Experiência com uso de Machine Learning na classificação da cobertura da terra
- Mestrando em Dinâmica dos Oceanos e da Terra
- Participante do projeto UFF Labs
- Interesses atuais: Empreendedorismo, R, Python, Data Science, Machine Learning e Desenvolvimento Web e Mobile.
- **`jairsonalberto@id.uff.br`**

# Quem são vocês?

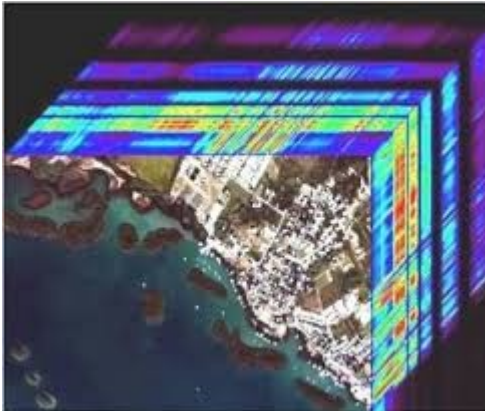


# Big Data e Geografia



**Avanços tecnológicos:** processamento de dados, disponibilidade de dados, softwares, linguagens de programação e frameworks 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.**

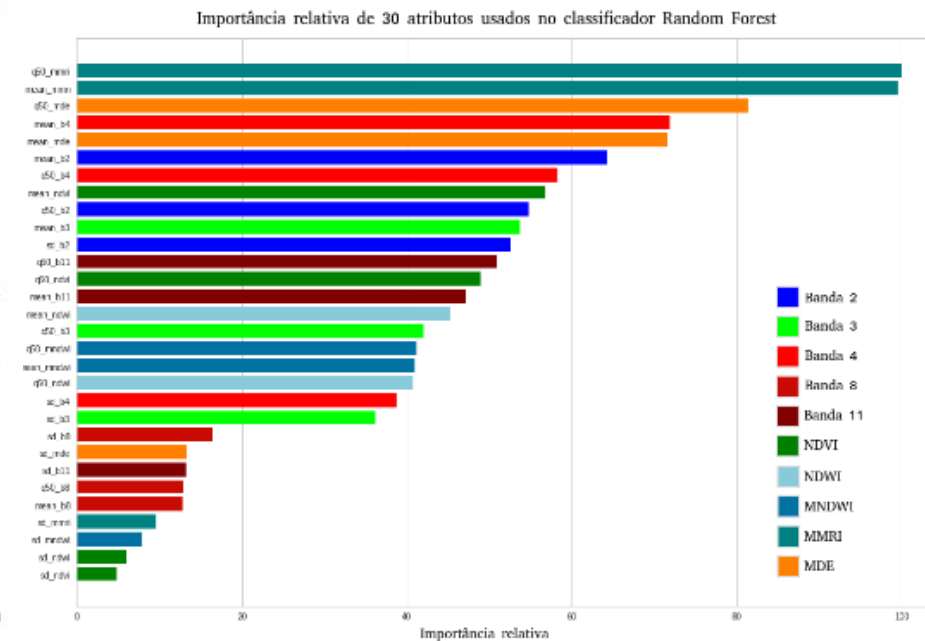
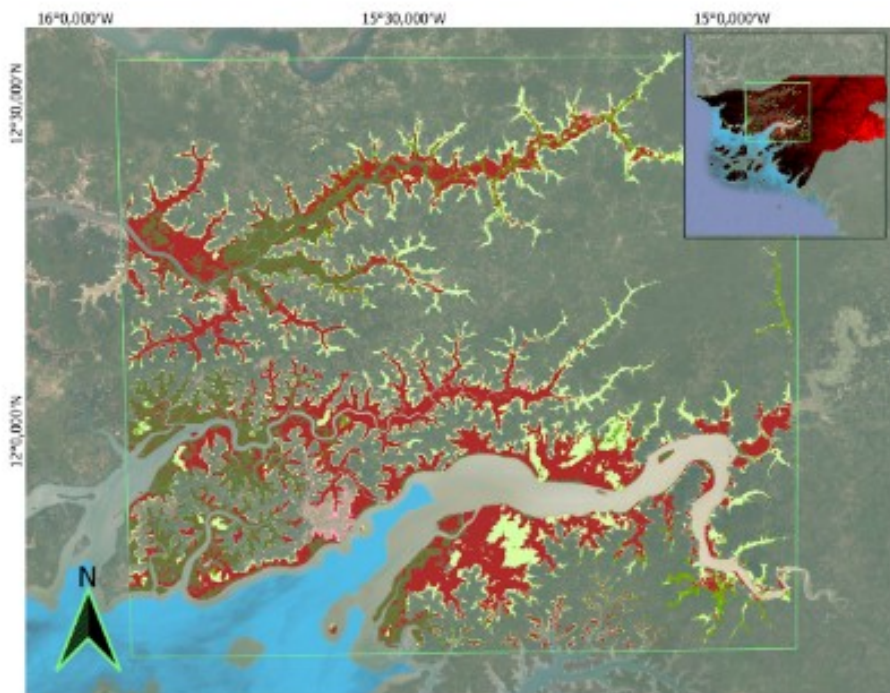


# Os 5 V's do Big Data



<https://site.moki.com.br/5-vs-big-data/>

# Classificação da cobertura da terra





# 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

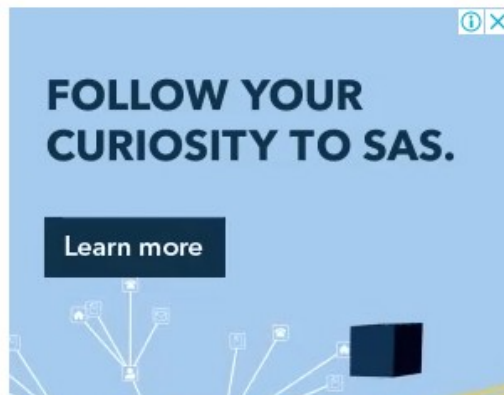
Claudia Varella

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](#)



# Concurso Ministério da Justiça

## Edital 20/2020

### CONHECIMENTOS ESPECÍFICOS

#### Cientista de Dados (Big Data)

**Conhecimentos específicos:** **Estatística:** Estatística descritiva: representação tabular e gráfica, medidas de tendência central e dispersão. Probabilidade: definições e teoremas; variáveis aleatórias e distribuição de probabilidade; variância e covariância; correlação linear simples; distribuição binomial, normal, qui-quadrado, t de Student e F; noções de amostragem e inferência estatística. **Gestão de dados:** DAMA-DMBOK. **Linguagens de programação aplicadas ao tratamento de dados:** Python, linguagem R e Matlab. **Big data:** fundamentos; tecnologia; gerenciamento; big data analítico; implementação; soluções reais. **Técnicas de preparação e apresentação de dados:** ETL (extração, transformação e carga); limpeza; importação; conversão; carga; visualização. **Business Intelligence:** conceitos; data warehouse; OLAP; MOLAP; ROLAP e OLTP; ferramentas e softwares. **Mineração de dados:** conceitos; técnicas; algoritmos; implementação e aplicação. **Inteligência artificial:** conceitos e aplicações. **Aprendizado de Máquina:** fundamentos básicos; algoritmos de aprendizado (supervisionado, não-supervisionado, por reforço); classificação e regressão; sistemas de recomendação; redes neurais artificiais; deep learning (noções); processamento de linguagem natural (noções). **Banco de dados sistemas gerenciadores:** SQL Server; Oracle; Postgre-SQL. **Bancos de dados não relacionais:** MongoDB, Cassandra. **Linguagem SQL:** SQL ANSI. **Variantes de linguagem SQL:** PL-SQL, T-SQL, PL/pqSQL, SQL/PSM. **Legislação específica:** Instrução Normativa nº 01/2019 – SEGES/ME; Lei nº 13.709/2018 (Lei geral de proteção de dados pessoais) e suas alterações.

Remuneração:  
R\$ 8300,00

# Concurso Petrobrás (Edital 1/2021)

33 vagas para Ciência de Dados

Salário básico de R\$ 6.937,43 com garantia de remuneração mínima de R\$ 11.716,82.

**SÍNTESE DAS ATRIBUIÇÕES:** atuar no desenvolvimento, análise e modelagem de dados em diversos segmentos da Companhia; atuar na análise, implementação, implantação e consultoria em soluções baseadas em estatística, aprendizado de máquina, processamento digital de imagens e processamento de linguagem natural (PLN); executar a fiscalização técnica e administrativa dos contratos de bens e serviços.

# Concurso TCU

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: conceituação 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,

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



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).



# Concurso Banco do Brasil (Edital 1/2022)

## Agente de Tecnologia (remuneração inicial R\$ 3622,23)

**PROBABILIDADE E ESTATÍSTICA:** 1 - Representação tabular e gráfica; 2 - Medidas de tendência central (média, mediana, moda, medidas de posição, mínimo e máximo) e de dispersão (amplitude, amplitude interquartil, variância, desvio padrão e coeficiente de variação); 3 - Variáveis aleatórias e distribuição de probabilidade; 4 - Teorema de Bayes; 5 - Probabilidade condicional; 6 - População e amostra; 7 - Variância e covariância; 8 - Correlação linear simples; 9 - Distribuição binomial e distribuição normal; 10 - Noções de amostragem e inferência estatística.

**TECNOLOGIA DA INFORMAÇÃO:** 1. **Aprendizagem de máquina:** Fundamentos básicos; Noções de algoritmos de aprendizado supervisionados e não supervisionados; Noções de processamento de linguagem natural. 2. **Banco de Dados:** Banco de dados NoSQL (conceitos básicos, bancos orientados a grafos, colunas, chave/valor e documentos); MongoDB; linguagem SQL2008; Conceitos de banco de dados e sistemas gerenciadores de bancos de dados (SGBD); Data Warehouse (modelagem conceitual para data warehouses, dados multidimensionais); Modelagem conceitual de dados (a abordagem entidade-relacionamento); Modelo relacional de dados (conceitos básicos, normalização); Postgre-SQL; 3. **Big data:** Fundamentos; Técnicas de preparação e apresentação de dados. 4. **Desenvolvimento Mobile:** linguagens/frameworks: Java/Kotlin e Swift. React Native 0.59; Sistemas Android api 30 e iOS xCode 10. 5. **Estrutura de dados e algoritmos:** Busca sequencial e busca binária sobre arrays; Ordenação (métodos da bolha, ordenação por seleção, ordenação por inserção), lista encadeada, pilha, fila e noções sobre árvore binária. 6. **Ferramentas e Linguagens de Programação para manipulação de dados:** Ansible; Java (SE 11 e EE 8); TypeScript 4.0; Python 3.9.X aplicada para IA/ML e Analytics (bibliotecas Pandas, NumPy, SciPy, Matplotlib e Scikit-learn).



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

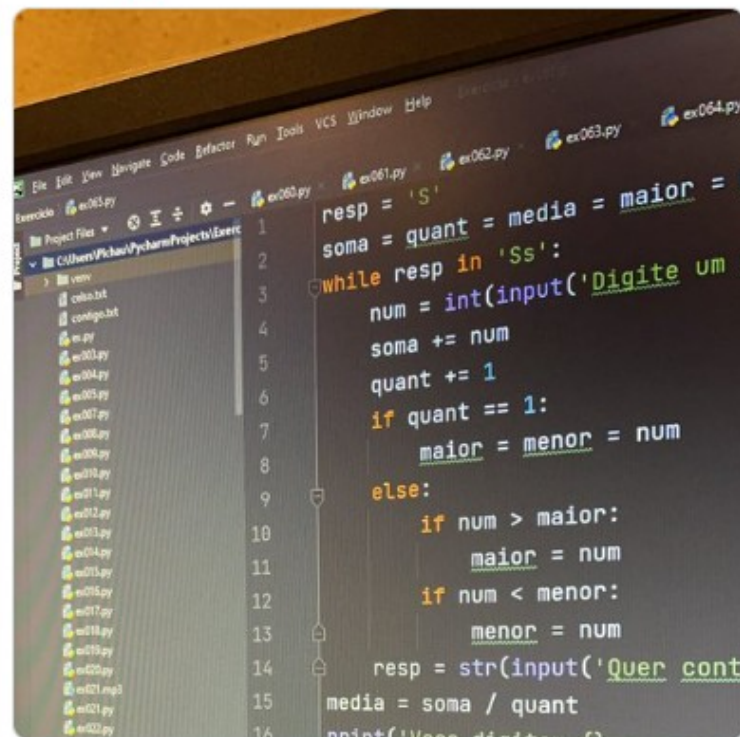
algumas de suas novas descobertas com o curso de Python.



Celso Portioli  
@celsoportioli · Seguir



Python ❤️



6:55 PM · 2 de fev de 2021



6 mil



Responder



Copiar link

[Ler 350 respostas](#)

# 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/>

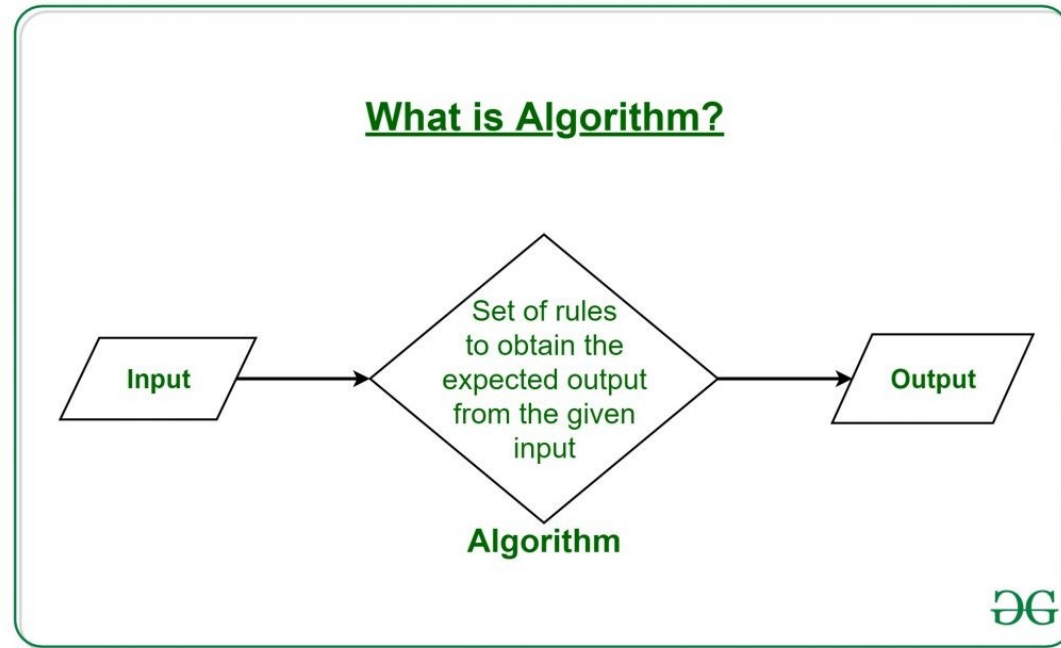
## **2) Segundo passo:**

Baixar o Rstudio (versão free):

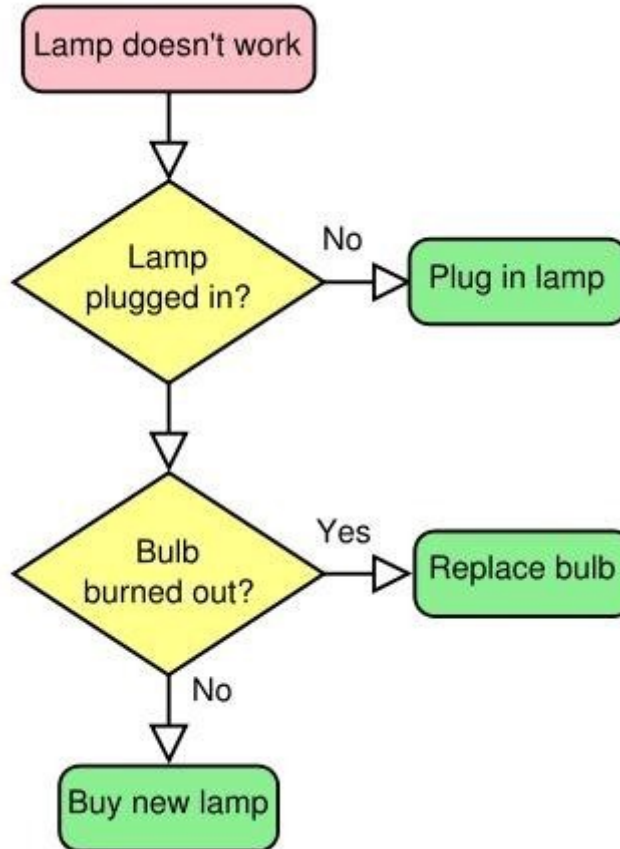
<https://www.rstudio.com/products/rstudio/download/>

# 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  
  }  
}
```

# Classes atômicas



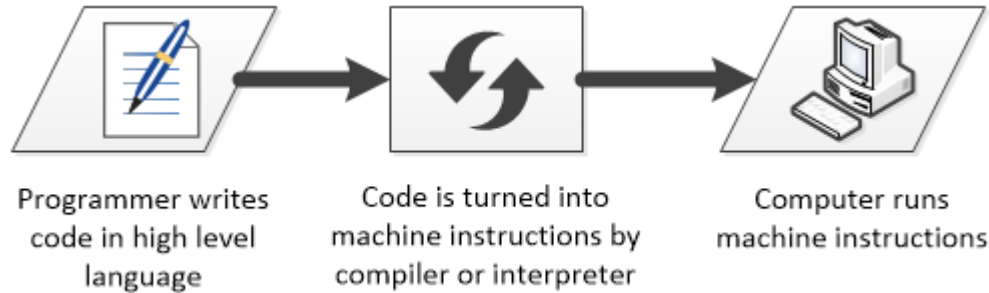
- **Inteiros:** 1, 2, 3
- **Lógico:** TRUE, FALSE
- **Real:** 3.14
- **Complexo:** 4+3i
- **Character (string):** “casa”
  
- **Raw** (comandos charToRaw e intToBits)

Para + detalhes teóricos sobre o raw: <https://blog.betrybe.com/linguagem-de-programacao/sistema-hexadecimal/>



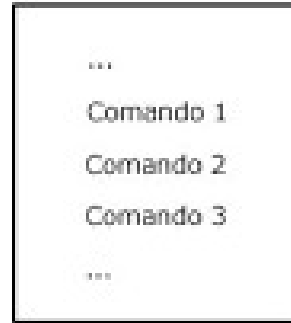
# Curiosidade (números binários)

- “Os computadores digitais trabalham internamente com dois **níveis de tensão**” Fonte: [https://pt.wikipedia.org/wiki/Sistema\\_de\\_numera%C3%A7%C3%A3o\\_bin%C3%A1rio](https://pt.wikipedia.org/wiki/Sistema_de_numera%C3%A7%C3%A3o_bin%C3%A1rio)

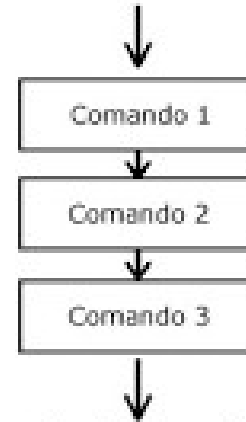


<https://computerscience.chemeketa.edu/cs160Reader/ProgrammingLanguages/HighLevelLanguages.html>

# Estrutura sequencial



Trecho sequencial de um algoritmo



Trecho sequencial de um fluxograma

[http://uab.ifsul.edu.br/tsiad/conteudo/modulo1/lop/lop\\_ud/lop\\_02.html](http://uab.ifsul.edu.br/tsiad/conteudo/modulo1/lop/lop_ud/lop_02.html)

# Estrutura sequencial

- Início

- Ação 1

`x<-2`

- Ação 2

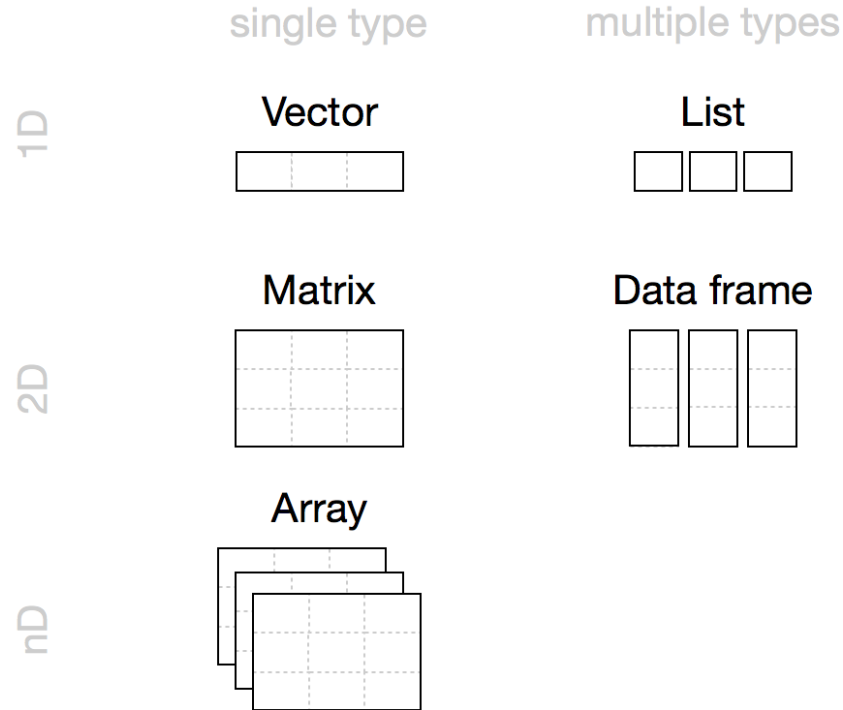
`y<-x+2`

- Ação 3

`print(x*y)`

- Fim

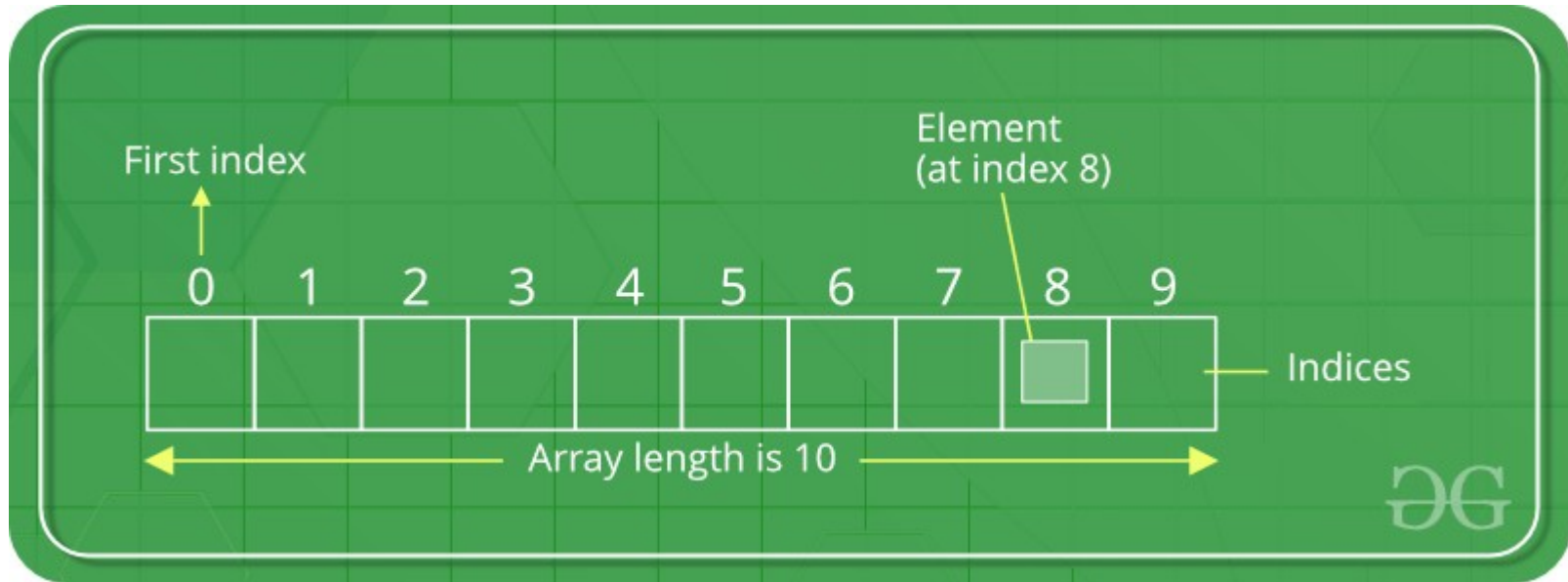
# Estrutura de dados composta



<https://rstudio-education.github.io/hopr/r-objects.html>

# Estrutura de dados composta

- Vetor



# Vetor

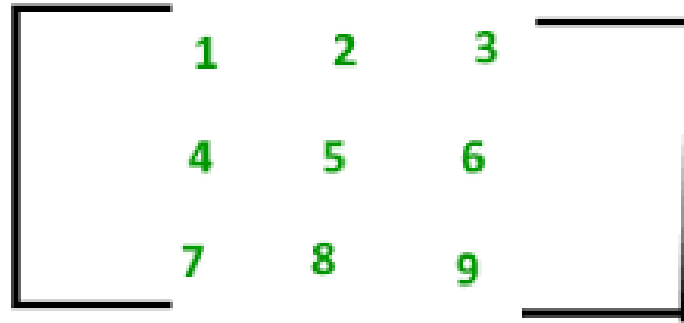
```
x<-seq(1,10,2)
c<-seq(5,10,length=30)
y<-c(1,2,3,4,5)
z<-1:10
a<-"blablabla"
b<-c("a","b","c")
d<-rep(c(0,1,2),times=10)
```

```
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
```



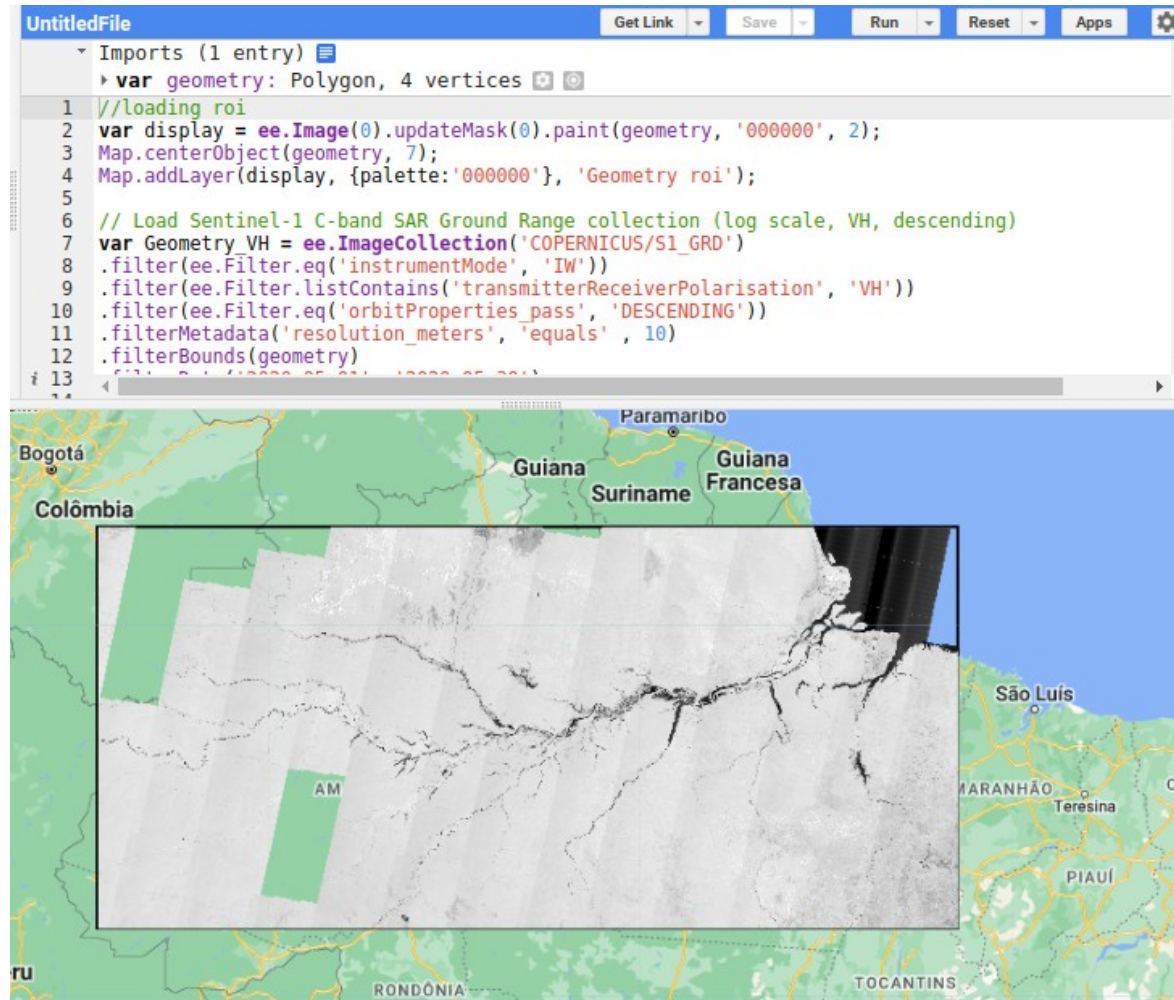
# Matriz



1	2	3
4	5	6
7	8	9

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

# Imagens de SR são matrizes



# 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):  $4$ ,  $0$ ,  $6$

Determinant:  $1$

```
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]
```

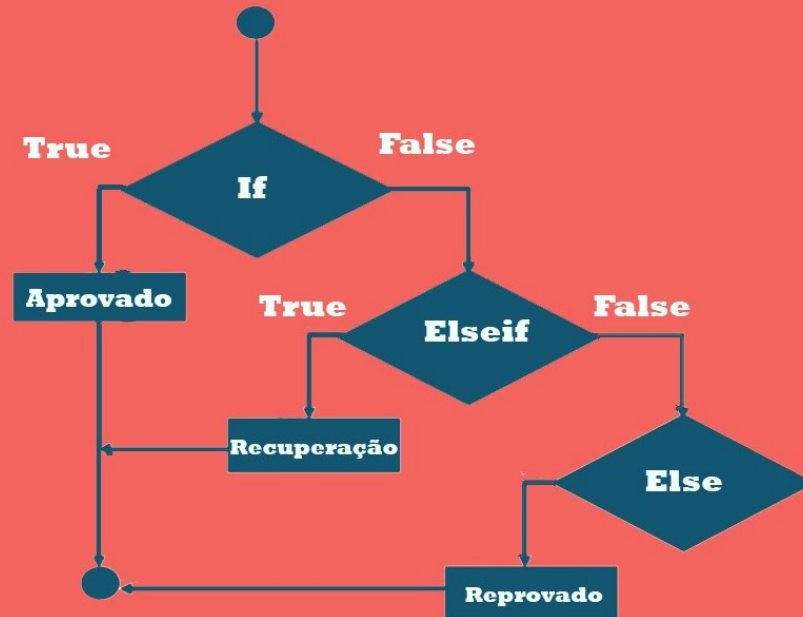
```
df[1,2]
```

```
df$indice
```

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

# IF ELSE

## if - elseif - else

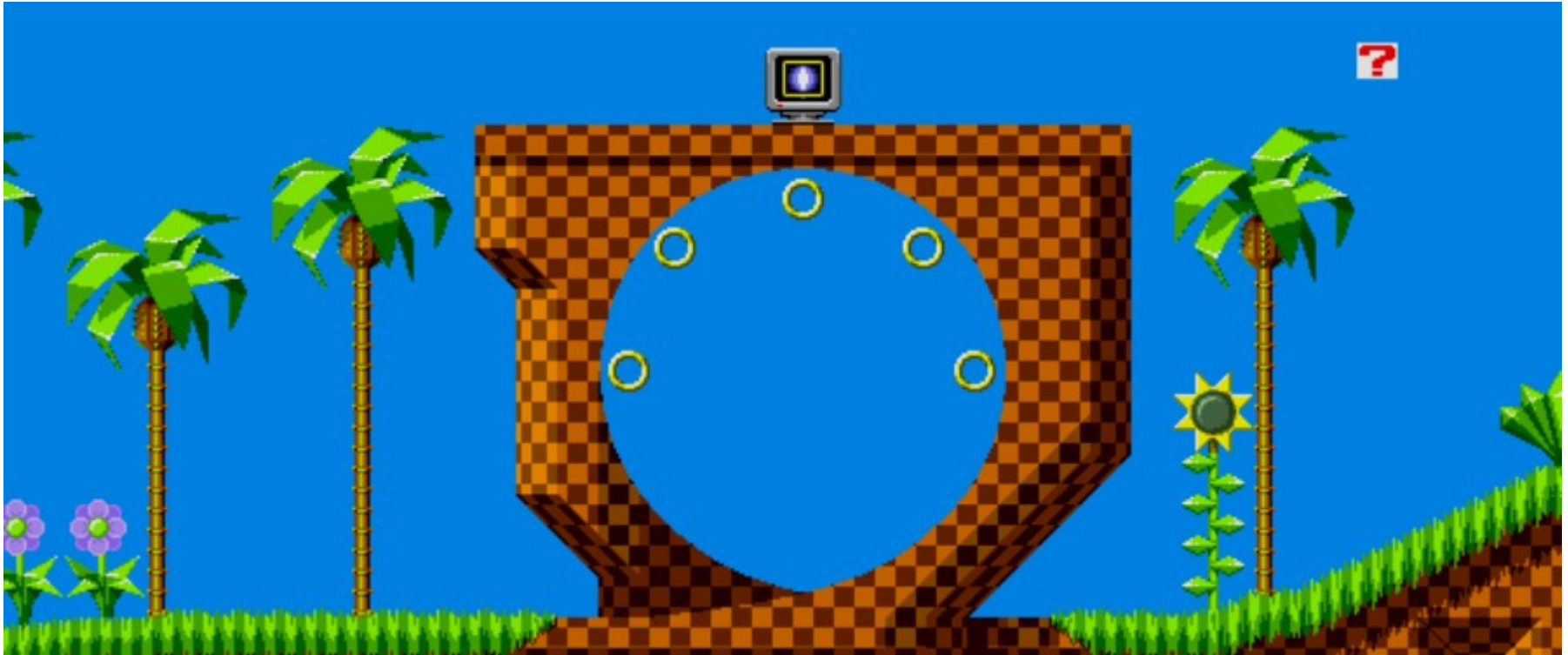


# IF ELSE

```
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]
```

```
  }
```

```
}
```

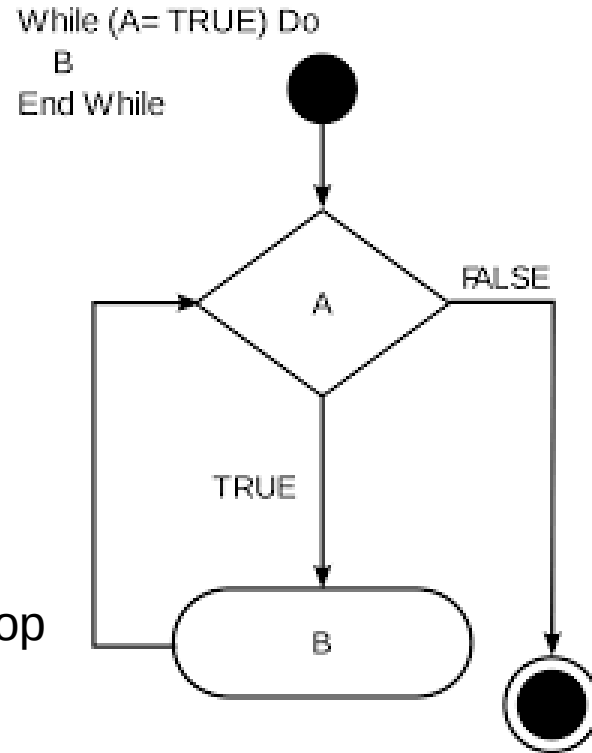
```
print(maior)
```

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

# While

Enquanto alguma condição for verdadeira:

- Execute algo



[https://en.wikipedia.org/wiki/While\\_loop](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](https://en.wikiversity.org/wiki/Programming_Fundamentals/Functions)

# Funções

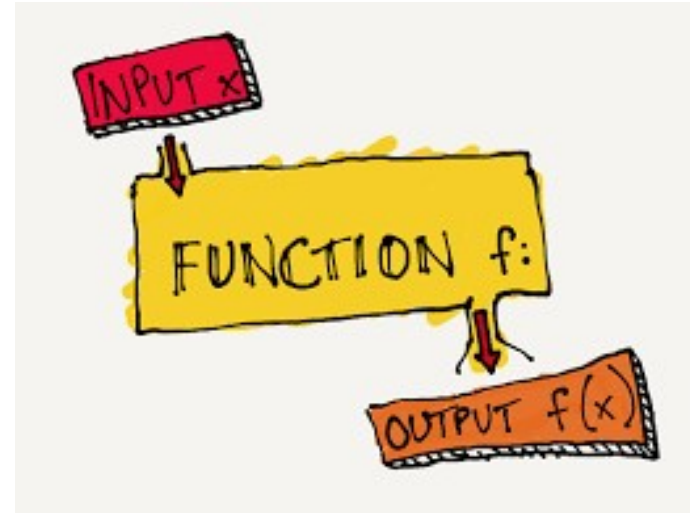
Print function example:

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

Annotations:

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

<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)
```

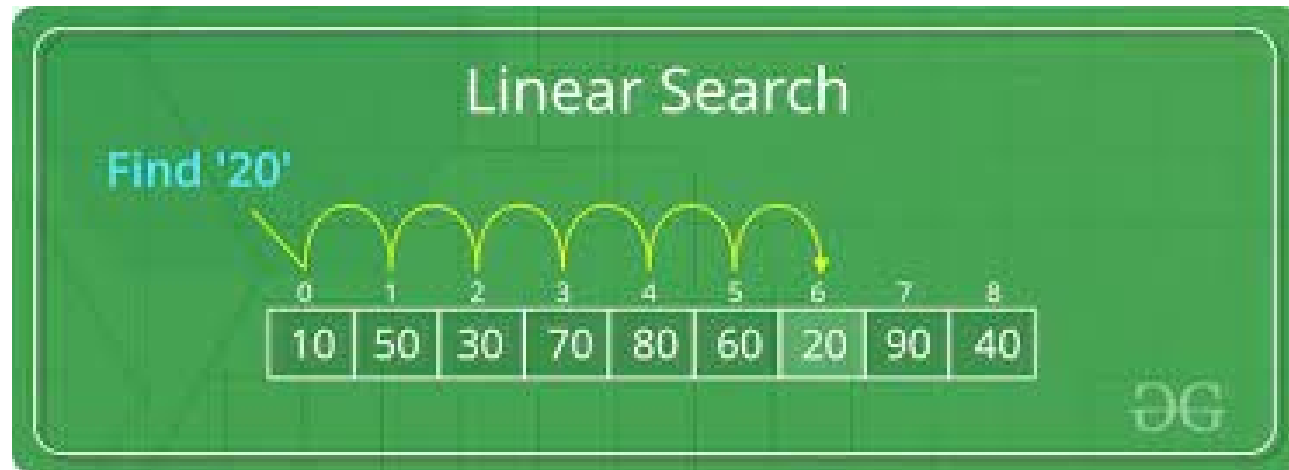
# Funções (exercícios)

- 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](https://pt.wikipedia.org/wiki/Busca_linear))



<https://www.geeksforgeeks.org/linear-search/>

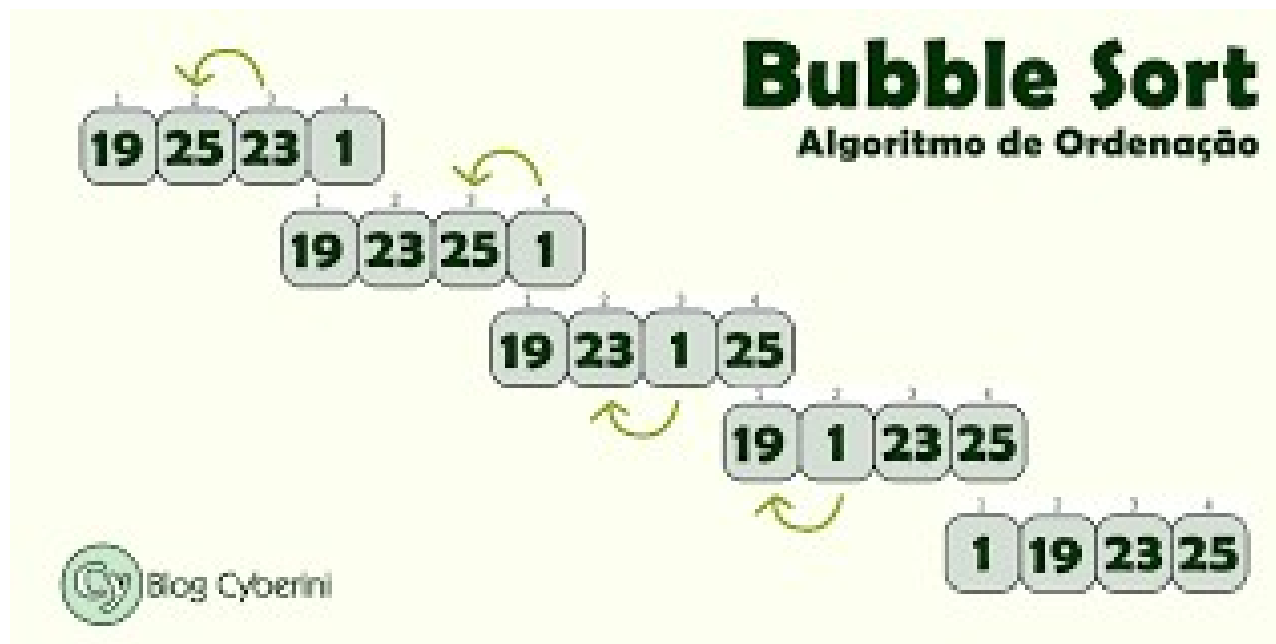
# Função encontrar\_valor()

```
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  $\geq 0$
- Exercício: implemente o fatorial não recursivo



# Fatorial

De quantas maneiras podemos organizar 5 alunos em uma fila?

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

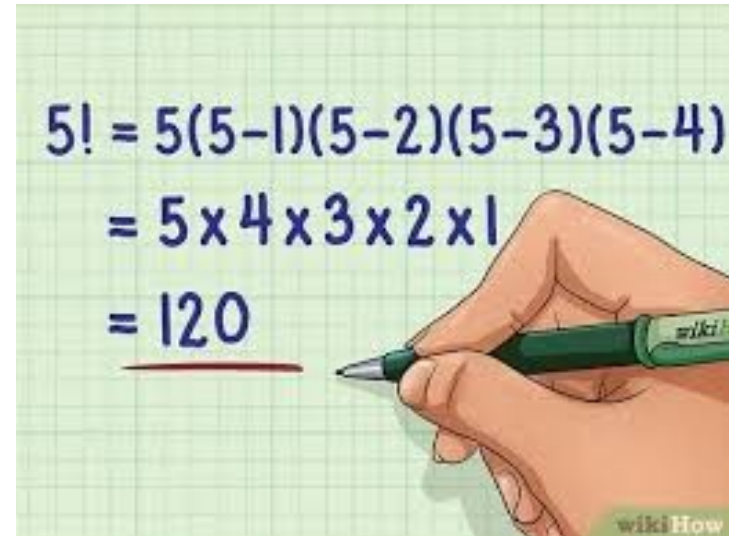
fatorial(n):

Se o n é igual a 0 ou 1

Resultado = 1

Caso contrário

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



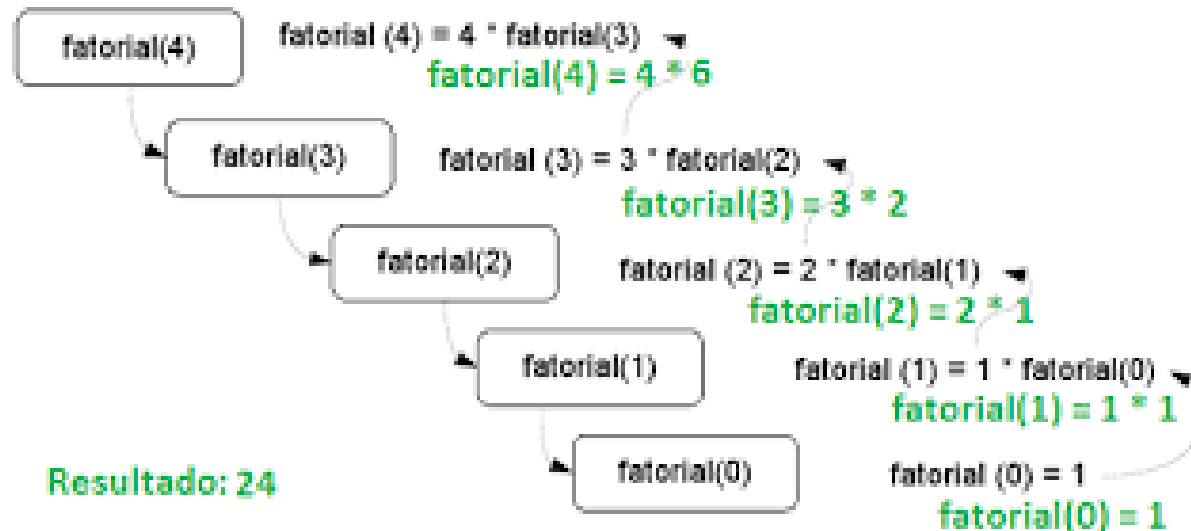
A hand holding a green pen is shown writing the calculation of 5 factorial on a green grid background. The text is written in blue ink and shows the expansion of the factorial formula.

$$\begin{aligned} 5! &= 5(5-1)(5-2)(5-3)(5-4) \\ &= 5 \times 4 \times 3 \times 2 \times 1 \\ &= 120 \end{aligned}$$

The final result, 120, is underlined with a red line. A small 'wikiHow' logo is visible in the bottom right corner of the image.

# 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)
```