



# Processamento da Informação

## Apresentação

Prof. Jesús P. Mena-Chalco  
CMCC/UFABC

Q2/2018

# Apresentação

- **Professor:**

Jesús P. Mena-Chalco (CMCC)

[jesus.mena@ufabc.edu.br](mailto:jesus.mena@ufabc.edu.br)

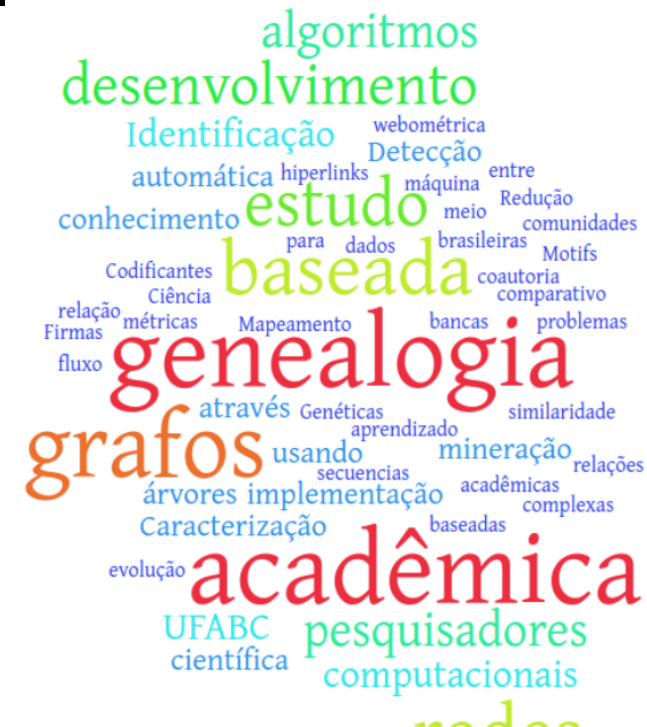
- **Formação:**

- Engenheiro da Computação.
- Mestre e Doutor em Ciência da Computação.  
Instituto de Matemática e Estatística da USP.

- Sala 517-A, torre 2, 5º Andar.

- **Áreas de pesquisa:**

- Pattern recognition
- Graph mining
- Scientometrics/Bibliometrics



<b>Pós-Doutorado</b>			
Laura Cristina Simões Viana	Genealogia acadêmica e sua relação com a constituição e a evolução do conhecimento na Escola Nacional de Saúde Pública Sergio Arouca	2017/Q3	Área: Engenharia do conhecimento - UFABC - ENSP/FIOCRUZ
<b>Doutorado</b>			
Antônio de Abreu Batista Júnior	Identificação automática de pesquisadores especialistas: Uma abordagem baseada em mineração de dados	2017/Q1	Ciência da Computação - UFABC - Bolsa FAPEMA
Julia Ferreira Bernardo (Coorientador)	Mapeamento de pesquisadores difusores da área de prevenção de drogas em contextos educacionais	2017/Q1	Educação e Saúde na Infância e Adolescência - UNIFESP - Bolsa CAPES
Priscilla Labanca	Modelos computacionais preditivos em grafos de genealogia acadêmica	2018/Q1	Ciência da Computação - UFABC
Rafael Jeferson Pezzuto Damaceno	Criação de grafos de genealogia acadêmica: desenvolvimento de metodologias baseadas em mineração de dados	2016/Q1	Ciência da Computação - UFABC - Bolsa CAPES
Luciano Rossi	Os caminhos da ciência por meio do fluxo de conhecimento na perspectiva da genealogia acadêmica	2015/Q3	Ciência da Computação - UFABC - Bolsa CAPES
<b>Mestrado</b>			
Iara Miranda Prates	Algoritmos de simplificação em grafos de genealogia acadêmica	2017/Q1	Ciência da Computação - UFABC - Bolsa CAPES
Rozivaldo Zacarias de Jesus	Caracterização de pesquisadores usando medidas de multidisciplinaridade: Uma abordagem baseada em mineração de publicações acadêmicas	2016/Q3	Ciência da Computação - UFABC
Diogo Fornaziero Segura Ramos	Análise temporal das relações de co-participação em bancas de defesa: uma abordagem baseado em evolução de grafos	2016/Q1	Ciência da Computação - UFABC - Bolsa CAPES
<b>Iniciação científica</b>			
Ana Laura Belotto Claudio (Coorientador)	Identificação de similaridade em disciplinas do catálogo da UFABC	2017-18	Ciência da Computação - UFABC - Bolsa IC
Arthur Veloso Kamienski	Identificando pesquisadores relevantes em grafos de genealogia acadêmica: Uma abordagem baseada em PageRank	2017-18	Ciência da Computação - UFABC - Bolsa FAPESP
Brian Alves Andreossi	Heurísticas computacionais para identificação e redução de potenciais conflitos de interesse em bancas de concursos	2017-18	Ciência da Computação - UFABC
Bruna Pereira Santos	Caracterização de obras literárias usando redes de co-ocorrências	2017-18	Ciência da Computação - UFABC - Bolsa FAPESP
Erick Akio Oti Aoyagui (Coorientador)	Deteção de comunidades em grafos de genealogia acadêmica através de propagação de rótulos	2017-18	Ciência da Computação - UFABC - Bolsa IC
Isabela Maria Biagioni Pedro (Coorientador)	Análise de regiões urbanas brasileiras com base em características da configuração de logradouros	2017-18	Bacharelado em C&T - UFABC - Bolsa IC
<b>Trabalho de conclusão de curso</b>			
Leonardo Nascimento	O impacto da Ciência da Computação nas diferentes áreas do conhecimento a partir da análise de redes brasileiras de coautoria	2018/Q1	Ciência da Computação - UFABC
Lucas Hideki Kimura	Mineração de dados acadêmicos: uma abordagem baseada no catálogo de teses e dissertações da Capes	2018/Q1	Ciência da Computação - UFABC

0000000:	3c21	444f	4354	5950	4520	6874	6d6c	2050	<!DOCTYPE html P
0000010:	5542	4c49	4320	222d	2f2f	5733	432f	2f44	UBLIC "-//W3C//D
0000020:	5444	2058	4854	4d4c	2031	2e30	2054	7261	TD XHTML 1.0 Tra
0000030:	6e73	6974	696f	6e61	6c2f	2f45	4e22	2022	nsitional//EN" "
0000040:	6874	7470	3a2f	2f77	7777	2e77	332e	6f72	http://www.w3.or
0000050:	672f	5452	2f78	6874	6d6c	312f	4454	442f	g/TR/xhtml1/DTD/
0000060:	7868	746d	6c31	2d74	7261	6e73	6974	696f	xhtml1-transitio
0000070:	6e61	6c2e	6474	6422	3e0a	3c68	746d	6c20	nal.dtd">.<html
0000080:	786d	6c6e	733d	2268	7474	703a	2f2f	7777	xmlns="http://ww
0000090:	772e	7733	2e6f	7267	2f31	3939	392f	7868	w.w3.org/1999/xh
00000a0:	746d	6c22	2078	6d6c	3a6c	616e	673d	2270	tml" xml:lang="p
00000b0:	742d	6272	2220	6c61	6e67	3d22	7074	2d62	t-br" lang="pt-b
00000c0:	7222	203e	0a09	3c68	6561	643e	200a	0909	r" >..<head> ...
00000d0:	3c6d	6574	6120	6e61	6d65	3d22	676f	6f67	<meta name="goog
00000e0:	6c65	2d73	6974	652d	7665	7269	6669	6361	le-site-verifica
00000f0:	7469	6f6e	2220	636f	6e74	656e	743d	224e	tion" content="N
0000100:	6948	7539	6448	5a6c	6f74	676e	6d6e	534f	iHu9dHZlotgnmnS0
0000110:	4e76	7a39	7759	6370	4677	7548	7947	6f35	Nvz9wYcpFwuHyGo5
0000120:	7852	4477	6556	6735	3745	2220	2f3e	0a20	xRDweVg57E" />.
0000130:	2020	2020	2020	203c	6d65	7461	2068	7474	<meta htt
0000140:	702d	6571	7569	763d	2263	6f6e	7465	6e74	p-equiv="content
0000150:	2d74	7970	6522	2063	6f6e	7465	6e74	3d22	-type" content="
0000160:	7465	7874	2f78	2d61	7070	6c69	6361	7469	text/x-applicati
0000170:	6f6e	3b20	6368	6172	7365	743d	5554	462d	on; charset=UTF-
0000180:	3822	203e	0a09	093c	6d65	7461	2070	726f	8" >..<meta pro
0000190:	7065	7274	793d	226f	673a	696d	6167	6522	perty="og:image"
00001a0:	2063	6f6e	7465	6e74	3d22	6874	7470	3a2f	content="http:/
00001b0:	2f75	6661	6263	2e65	6475	2e62	722f	7465	/ufabc.edu.br/te
00001c0:	6d70	6c61	7465	732f	5546	4142	432f	696d	mplates/UFABC/im
00001d0:	6167	6573	2f6c	6f67	6f5f	7566	6162	632e	ages/logo_ufabc.
00001e0:	706e	6722	2f3e	0a09	0920	203c	6d65	7461	png"/>... <meta

2018

1400

WHY SHOULD I  
LEARN TO PROGRAM?



WHY SHOULD I  
LEARN TO READ?



stick figures from xkcd.com

# Coding Is the Must-Have Job Skill of the Future

**7.1k**  
SHARES

2985

Share

2372

Tweet

403

Share

1079

in

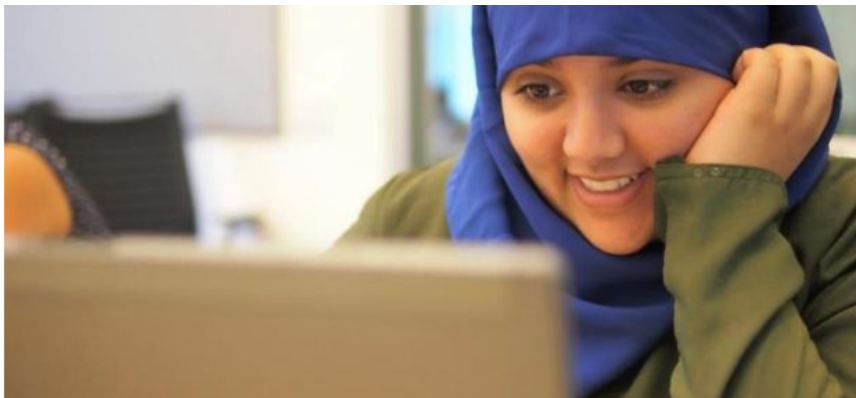
218

41

WHAT'S THIS?

## Why Programming Is The Core Skill Of The 21st Century

*It's never been easier, more accessible, or more essential to learn coding skills.*



**How Coding Can Boost Everyone's Career**

“Everybody in this country should learn how to program a computer... because it teaches you how to think.”

- Steve Jobs

<https://www.youtube.com/watch?v=Dv7gLpW91DM>





Programming languages give you much more freedom than you have with commercial applications.

#### SCIENTIFIC COMPUTING

# Code alert

*Programming tools can speed up and strengthen analyses, but mastering the skills takes time and can be daunting.*

BY MONYA BAKER

**A**uriel Fournier had no choice but to learn programming. The ecology PhD student wanted to use a complex set of

tutorials, mastered the package and now helps other researchers to make sense of R and similar tools. It can be “really intimidating” to learn a programming language, but the long-term benefits are well worth the effort, she says.

methodologist and psychometrician at the University of Amsterdam. “You can only do what the buttons say you can do.” Programming languages are much broader in scope, he says. “In R or other programming languages you can do anything you want.” But to achieve this broad latitude, researchers need to learn how to code — to formulate commands in a programming language that tell a computer what to do. Researchers who know just a bit of code gain access to a wealth of software packages that automate repetitive tasks and increase options for compiling, analysing, sharing and presenting data.

Coders often use R for doing statistics, whereas Python, which is often considered more intuitive, is good for repetitive tasks and for extracting data from web-based applications. Ruby provides more ways of doing the same tasks and is popular with web developers.

Learning to code is empowering and can hugely improve a researcher’s career prospects. “But it does require an investment,” says Fournier. And that includes not just time, but also an ongoing effort to assemble a community of helpful experts and to be willing to make, find and fix mistakes.

#### SMOOTH OPERATORS

Coding is becoming a crucial part of research, says Ethan White, an ecologist at the University of Florida, Gainesville, who has designed courses in quantitative methods. “When you do this well, your life is easier and your results can be regenerated,” he points out. Scientists commonly use languages such as Python and R to conduct and automate analyses, because in this way they can speed data crunching, increase reproducibility, protect data from accidental deletion or alteration and handle data sets that would overwhelm commercial applications. Researchers who use these languages can tackle questions that would be impractical to address if data were manipulated manually, says White. Reconfiguring an analysis or revising graphs becomes quick and straightforward, and researchers can more easily build on their own or others’ work.

Andrew Durso can vouch for those upsides.



Preconceito sobre programador(a)?

Alunos e professores da UFABC

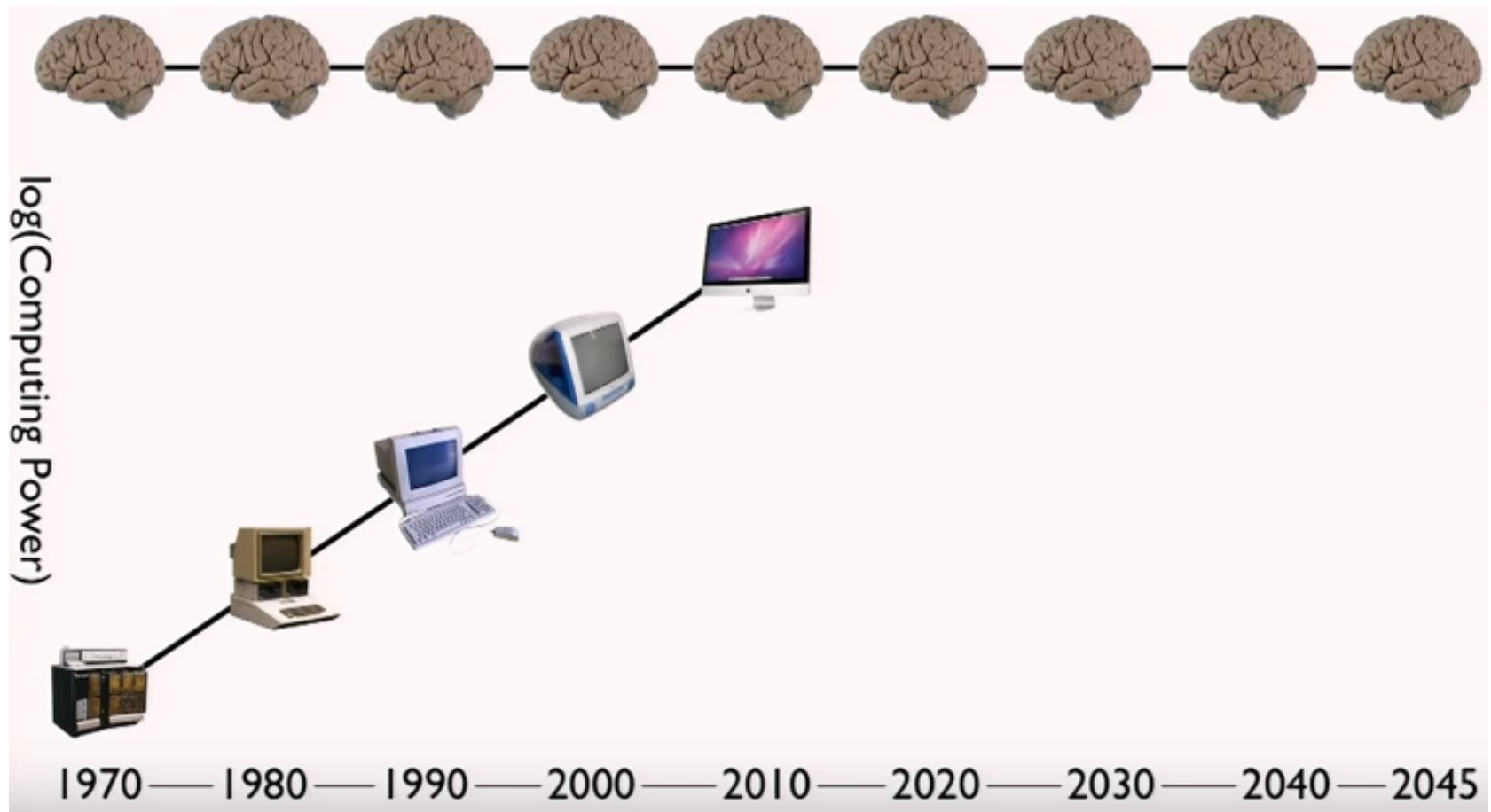


Vamos ler: <https://www.programaria.org/category/debater/>



# Programar?

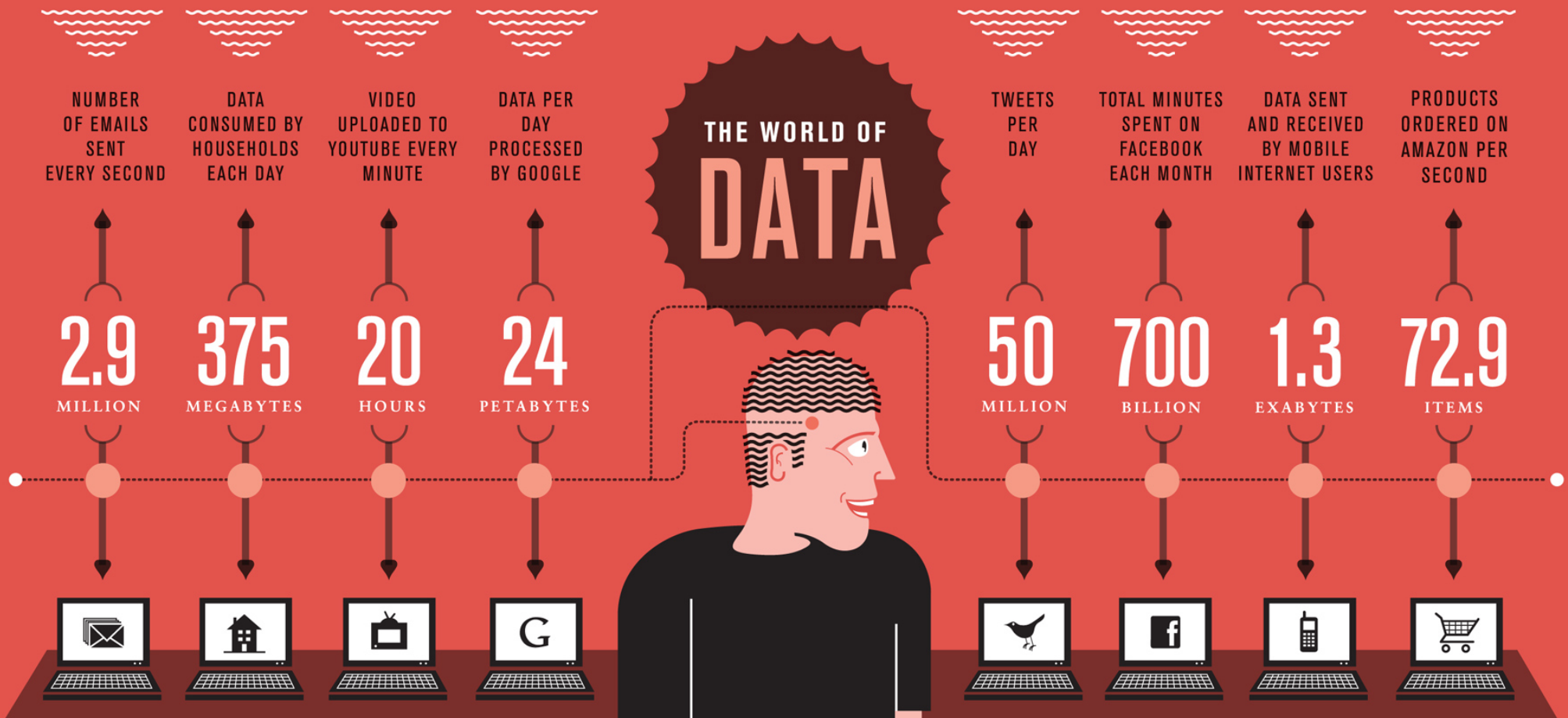
- Programar auxilia a **desenvolver o pensamento lógico.**
- Programar **lhe torna mais independente.**
- ...



You Should Learn to Program: Christian Genco at TEDxSMU  
<https://www.youtube.com/watch?v=xfBWk4nw440>

# Grande escala?

<http://blog.bimeanalytics.com/english/world-of-data-infographic>



Em vez de a ciência não avançar devido à escassez de dados, hoje em dia ela frequentemente encontra **dificuldades em avançar por seu excesso.**

Roberto M. Cesar-Jr (IME/USP)

**“Programming is  
thinking,  
not typing.”**

Casey Patton





# Programar?

**Mas tem um custo:**

- **Tempo**
- **Dedicação**

# Aprender a programar: considerações

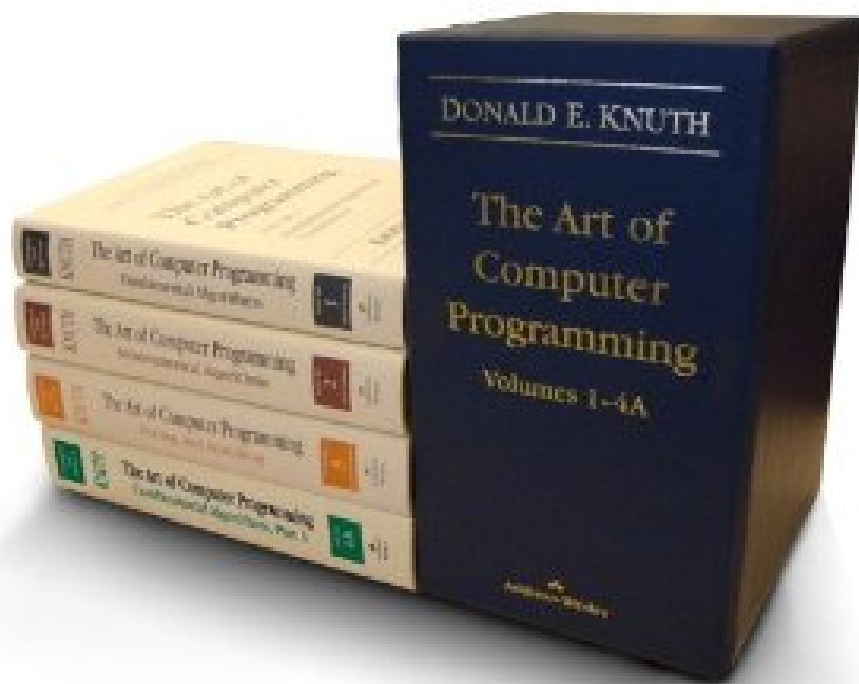
- Fall in love with mathematics (pratique matemática)
- Be self-motivated (trabalhe com pares)
- Never back down (seja persistente)
- Become a master (ensine aos colegas)
- Be a bookworm (seja leitor ávido)

Leia as seguintes sugestões:

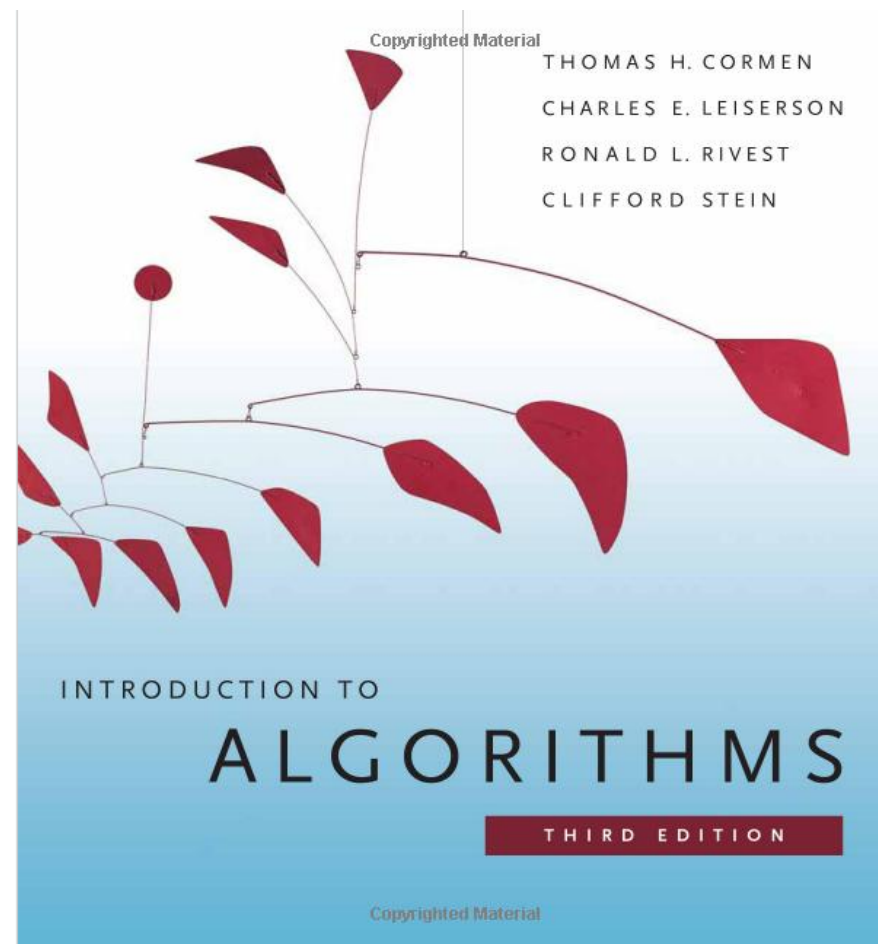
<http://www.wikihow.com/Learn-a-Programming-Language>

*If you think you're a really good programmer... read [Knuth's] Art of Computer Programming... You should definitely send me a résumé if you can read the whole thing.*

—Bill Gates



Graphs, Networks and Algorithms. Second Edition. Dieter Jungnickel.  
An Introduction to the Theory of Numbers. Zuckerman y Montgomery.  
Game Theory. Drew Fudenberg.  
Theory of Games and Economic Behavior. John von Neumann, Oskar Morgenstern.





# **Sobre a disciplina**



# Objetivos

## Objetivos:

- Apresentar os fundamentos sobre manipulação e tratamento da Informação, principalmente por meio da explicação e experimentação dos conceitos e do uso prático da lógica de programação.

## Competências:

- Compreender os conceitos fundamentais a respeito da manipulação e tratamento da Informação.
- Entender a lógica de programação de computadores.
- Desenvolver algoritmos básicos para modelar e solucionar problemas de natureza técnico-científica.

# Estudando programação

**Combinação de teoria e prática de maneira inseparável.**

- Análise de um problema
- Planejar a abordagem
- Implementar uma solução

# Créditos (T-P-I)=(3-2-5)

Cada disciplina na UFABC é representada por três algarismos: T-P-I

- **T:** Número de horas semanais de aulas expositivas presenciais da disciplina (teóricas)
- **P:** Número médio de horas semanais de trabalho de laboratório, aulas práticas ou aulas de exercícios, realizadas em sala de aula (práticas)
- **I:** Estimativa de horas semanais adicionais de trabalhos necessárias para o bom aproveitamento da disciplina (estudos e trabalhos)

# URLs

- **URL:** <http://professor.ufabc.edu.br/~jesus.mena/courses>
- **Cadastre-se no Tidia4:** <http://tidia4.ufabc.edu.br>  
Procurar: “PI-2018-Q2-Jesus”



# Calendário

Teoria

Laboratório

JUNHO						
Dom	Seg	Ter	Qua	Qui	Sex	Sab
					1	2
<b>3</b>	4	5	6	7	8	9
<b>10</b>	11	12	13	14	15	16
<b>17</b>	18	19	20	21	22	23
<b>24</b>	25	26	27	28	29	30

JULHO						
Dom	Seg	Ter	Qua	Qui	Sex	Sab
<b>1</b>	2	3	4	5	6	7
<b>8</b>	9	10	11	12	13	14
<b>15</b>	16	17	18	19	20	21
<b>22</b>	23	24	25	26	27	28
<b>29</b>	30	31				

9 - Revolução Constitucionalista

AGOSTO						
Dom	Seg	Ter	Qua	Qui	Sex	Sab
			1	2	3	4
<b>5</b>	6	7	8	9	10	11
<b>12</b>	13	14	15	16	17	18
<b>19</b>	20	21	22	23	24	25
<b>26</b>	27	28	29	30	31	

20 - Feriado municipal - S B e recesso em S A

# Agenda

Aula	Data	Conteúdo previsto	Lista (prazo de entrega)
1	08/06	Apresentação Introdução, algoritmos e tipos de dados	
2	13/06	Conceitos básicos de modularização [atividade]	Lista 1 (data de entrega 27/06)
3	15/06	Estruturas e seleção simples e composta (if - else) [atividade]	
4	22/06	Repetição enquanto: while [atividade]	
5	27/06	Repetição para até: for [atividade]	Lista 2 (data de entrega 11/07)
6	29/06	Vetores - parte 1 [atividade]	
7	06/07	Vetores - parte 2 [Enunciado]	
8	11/07	Simulado 1 [Gabarito]	
9	13/07	Prova Teoria 1 [ Enunciado   Gabarito ]	
10	20/07	Matrizes - parte 1 [atividade]	Lista 3 (data de entrega 03/08)
11	25/07	Matrizes - parte 2 [atividade]	
12	27/07	Caracteres e Strings - parte 1 [atividade]	
13	03/08	Caracteres e Strings - parte 2 [atividade]	Lista 4 (data de entrega 17/08)
14	08/08	Recursão & Exercícios de programação [atividade]	
15	10/08	Exercícios de programação [atividade]	
16	17/08	Simulado 2 [Gabarito]	Lista 5 (data de entrega 29/08) Lista bônus - opcional
17	22/08	Prova Teoria 2 [ Enunciado   Gabarito ]	
18	24/08	Prova Substitutiva (substitui a menor nota das provas P1, P2). É "aberta": qualquer aluno pode fazer mesmo que não tenha perdido nenhuma prova.	
19	Q3/2018	Prova de Recuperação (Teoria e Laboratório).	

# Avaliação

Uma **nota de Teoria** e uma **nota de Prática**.

O conceito final é uma função dessas duas notas.

**Parte Prática:** Depende do professor de Prática.

**Parte de Teoria:** Duas provas e exercícios.

- Primeira Prova (30%): 17/07/2018
- Segunda Prova (40%): 22/08/2018
- 4 Listas de exercícios (30%)
  
- Prova Substitutiva: 24/08/2018
- Prova de Recuperação: Q3/2018

# Avaliação

## Teoria

- **A:** nota  $\geq 9$
- **B:**  $7,5 \leq \text{nota} < 9$
- **C:**  $6 \leq \text{nota} < 7,5$
- **D:**  $5,0 \leq \text{nota} < 6$
- **F:** nota  $< 5,0$

Nota Teoria	Nota Prática	Conceito
A	A	A
	B	A
	C	B
	D	B
B	A	B
	B	B
	C	B
	D	C
C	A	B
	B	C
	C	C
	D	C
D	A	C
	B	C
	C	D
	D	D

F em teoria ou em prática = F na disciplina  
(Não desista, faça a PRec).

# Bibliografia

## Bibliografia Básica

- Forbellone, A. L. V.; Eberspächer, H. F.; **Lógica de Programação - A Construção de Algoritmos e Estruturas de Dados**; 3ª edição, Editora Pearson Prentice-Hall, 2005.
- Sebesta, R. W.; **Conceitos de Linguagens de Programação**; 5ª edição, Editora Bookman, 2003.

## Bibliografia Complementar

- Ascensio, A.F.; Campos, E.A., **Fundamentos da Programação de Computadores**, Pearson, 3a edição, 2012.
- Puga, S., **Lógica de programação e estruturas de dados com aplicações em Java**, Pearson Prentice-Hall, 2a edição, 2009.