



Processamento da Informação

Algoritmos e Linguagens de programação

Prof. Jesús P. Mena-Chalco
CMCC/UFABC

Q2/2018

Construção de programas

Para a construção de programas é necessário um **conjunto de instruções** colocadas em ordem sequencial lógica:
Algoritmo.



Algoritmos

Algumas definições de algoritmo

- **Sequência ordenada de passos** que deve ser seguida para a realização de uma tarefa (ASCENCIO, 1999).
- **Regras formais** para a obtenção de um resultado ou **solução de um problema**, englobando formulas de expressões aritméticas (MANZANO, 1997).
- **Sequência finita de instruções ou operações** cuja execução, em tempo finito, resolve um problema computacional, qualquer que seja sua instância (SALVETTI, 1999)

Algumas definições de algoritmo

Algoritmo:

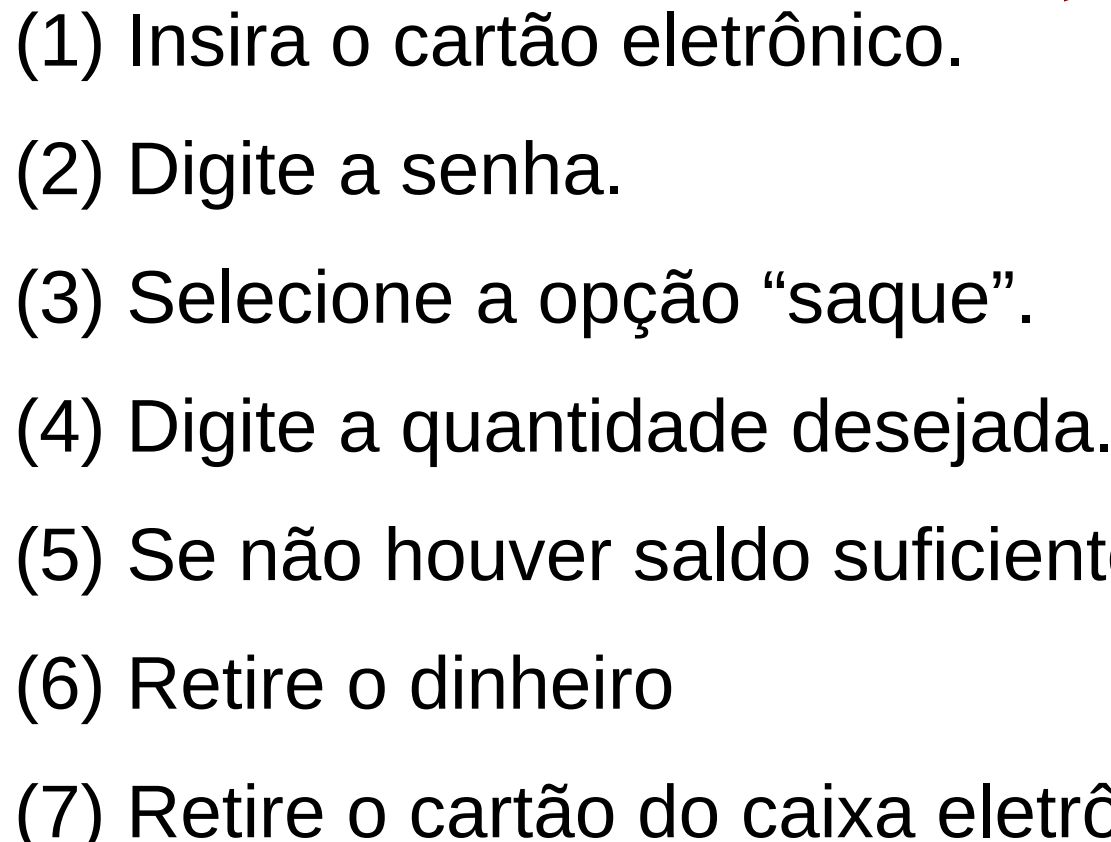
Sequência lógica são passos executados até atingir um objetivo ou solução de um problema.

Exemplos de situações onde uma sequência lógica de passos é necessária:

- Fazer um bolo
- Construir um robô para explorar um local desconhecido
- Trocar uma lâmpada

Exemplo de algoritmo

Sacar dinheiro de um caixa eletrônico (2012):

- 
- (1) Insira o cartão eletrônico.
 - (2) Digite a senha.
 - (3) Selecione a opção “saque”.
 - (4) Digite a quantidade desejada.
 - (5) Se não houver saldo suficiente, continue no passo (7).
 - (6) Retire o dinheiro
 - (7) Retire o cartão do caixa eletrônico

Exemplo de algoritmo

Sacar dinheiro de um caixa eletrônico (2018):

- (1) ~~Insira o cartão.~~ Use o sensor biométrico (iris, face, digitais).
- (2) ~~Digite a senha.~~
- (3) Selecione a opção “saque”.
- (4) Digite a quantidade desejada.
- (5) Se não houver saldo suficiente, continue no passo (7).
- (6) Retire o dinheiro
- (7) ~~Retire o cartão.~~ Transação finalizada.

A relação com a tecnologia?

A sequência de passos **depende do tipo do tecnologia utilizada.**

A sequência de passos **é limitada pela quantidade finita de possíveis operações.**

Exemplo de algoritmo

Verificar se o número x é par ou ímpar:

- Se o resto da divisão de x por 2 for igual a 0, então
O número é par
- Senão, então
O número é ímpar

O resto da divisão:

$$3/2 = 1$$

$$4/2 = 0$$

$$7/2 = 1$$

$$17/2 = 1$$

$$26/2 = 0$$

Al-Khorezmi: Um Matemático pouco conhecido



Matemático, astrônomo, astrólogo, geógrafo e autor Persa (Bagdad, c.850)

Descreveu o sistema numérico atual a um nível entendível.

al-Khwarizmi, Al-Khwarizmi, Al-Khawaritzmi ou al-Khowarizmi

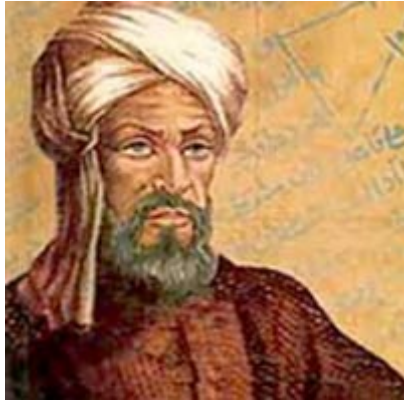
O primeiro pensador algoritmico.



*Selo postal (Rusia, 1983)
Comemorando o aniversário 1200*

Quando em uma subtração nada queda, então escreva um pequeno círculo para que esse lugar não permaneça vacío (Al-Khorezmi explicando o zero, Século IX)

Al-Khorezmi: Um Matemático pouco conhecido



Os termos:

- **Algoritmo**
- **Algarismo** (número/digito)
- **Algol** (linguagem de programação) provém de seu nome.

- Al-Khorezmi = “de Khorezm”
- Algoritmos datam dos gregos (por exemplo, algoritmo de Euclides para **calcular o máximo divisor comum**).
- Al-Khorezmi foi o primeiro em projetar algoritmos pensando na sua eficiência para o calculo raizes de equações.
- Usou um tipo mecânico similar a um ábaco.

Algoritmos e programas

Para que um computador desempenhe uma tarefa é necessário que uma sequência de ações (**algoritmo**) seja especificada de uma forma compreensível pela máquina.

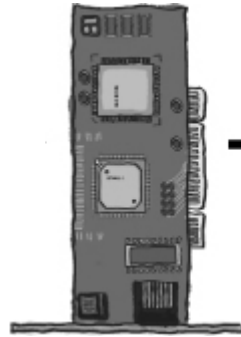
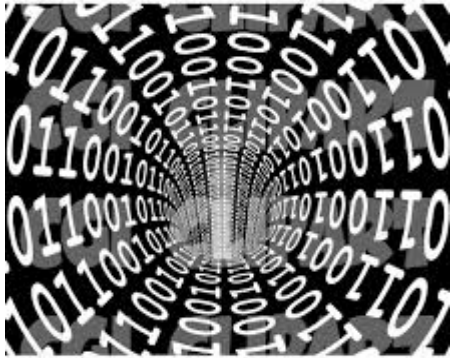
Um programa de computador nada mais é que um algoritmo escrito de forma compreensível pelo computador.

Ações especificadas de maneira “**formal**”.

Algoritmos e programas

Um algoritmo computacional tem que **terminar em tempo finito**.

Se o algoritmo não terminar em tempo finito, alguns pesquisadores classificam essa sequência de passos como **método computacional**.



```
000000: 00111100 00100001 01000100 01001111 01000011 01010100 <!DOCT
0000006: 01011001 01010000 01000101 00100000 01101000 01110100 YPE ht
000000c: 01101101 01101100 00100000 01010000 01010101 01000010 ml PUB
0000012: 01001100 01001001 01000011 00100000 00100010 00101101 LIC "-
0000018: 00101111 00101111 01010111 00110011 01000011 00101111 //W3C/
000001e: 00101111 01000100 01010100 01000100 00100000 01011000 /DTD X
0000024: 01001000 01010100 01001101 01001100 00100000 00110001 HTML 1
000002a: 00101110 00110000 00100000 01010100 01110010 01100001 .0 Tra
0000030: 01101110 01110011 01101001 01101000 01101001 01101111 nsitio
0000036: 01101110 01100001 01101100 00101111 00101111 01000101 nal//E
000003c: 01001110 00100010 00100000 00100010 01101000 01110100 N" "ht
0000042: 01110100 01110000 00111010 00101111 00101111 01110111 tp://w
0000048: 01110111 01110111 00101110 01110111 00110011 00101110 ww.w3.
000004e: 01101111 01110010 01100111 00101111 01010100 01010010 org/TR
0000054: 00101111 01111000 01101000 01110100 01101101 01101100 /xhtml
000005a: 00110001 00101111 01000100 01010100 01000100 00101111 1/DTD/
0000060: 01111000 01101000 01110100 01101101 01101100 00110001 xhtml
0000066: 00101101 01110100 01110010 01100001 01101110 01110011 -trans
000006c: 01101001 01110100 01101001 01101111 01101110 01100001 itiona
0000072: 01101100 00101110 01100100 01110100 01100100 00100010 l.dtd"
0000078: 00111110 00001010 00111100 01101000 01110100 01101101 >.<htm
000007e: 01101100 00100000 01111000 01101101 01101100 01101110 l xmln
0000084: 01110011 00111101 00100010 01101000 01110100 01110100 s="htt
000008a: 01110000 00111010 00101111 00101111 01110111 01110111 p://ww
0000090: 01110111 00101110 01110111 00110011 00101110 01101111 w.w3.o
0000096: 01110010 01100111 00101111 00110001 00110001 00110001 rg/199
000009c: 00111001 00101111 01111000 01101000 01110100 01101101 9/xhtm
00000a2: 01101100 00100010 00100000 01111000 01101101 01101100 l" xml
```

```
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 iHu9dHZlotgmnS0
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
```


ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Unicode Table

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F		
0000																																		Symbols
0020		!	"	#	\$	%	&	'	()	*	+	,	-	.	/	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?		Number
0040	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_		Alphabet
0060	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	{		}	~			
0080	€		,	f	„	…	†	‡	^	%	Š	<	Œ		Ž			'	'	“	”	•	–	—	~	™	š	>	œ		ž	ÿ		
00A0		ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯	°	±	²	³	´	µ	¶	·	,	¹	º	»	¼	½	¾	¿		
00C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß		Latin
00E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ		
0100	Ā	ā	Ă	ă	Ą	ą	Ć	ć	Ĉ	ĉ	Ċ	ċ	Č	č	Ď	ď	Đ	đ	Ē	ē	Ĕ	ĕ	É	é	Ė	ė	Ĝ	ğ	Ğ	ğ				
0120	Ġ	ġ	Ģ	ģ	Ĥ	ĥ	Ħ	ħ	Ĩ	ĩ	İ	ı	Ĳ	ı	Ĵ	ı	Ĳ	ı	Ĵ	ı	Ĵ	ı	Ĵ	ı	Ĵ	ı	Ĵ	ı	Ĵ	ı	Ĵ	ı		
0140	Ł	ł	Ń	ń	Ņ	ņ	Ň	ň	Ŋ	ŋ	Ō	ō	Ŏ	ö	Œ	œ	Ŕ	ŕ	Ŗ	ŗ	Ř	ř	Ś	ś	Ŝ	ŝ	Ş	ş						
0160	Š	š	Ţ	ţ	Ť	ť	Ŧ	ŧ	Ū	ū	Ŭ	ŭ	Ů	ů	Ű	ű	Ų	ų	Ŵ	ŵ	Ŷ	ŷ	Ÿ	Ž	ž	Ż	ż	Ž	ž					
0180	ƀ	Ɓ	Ƃ	ƃ	Ƅ	ƅ	Ɔ	Ƈ	ƈ	Ɖ	Ɗ	Ƌ	ƌ	ƍ	Ǝ	Ə	Ɛ	Ƒ	ƒ	Ɠ	Ɣ	ƕ	Ɩ	Ɨ	Ƙ	ƙ	ƚ	ƛ	Ɯ	Ɲ	ƞ	Ɵ		
01A0	Ɔ	Ƈ	ƈ	Ɖ	Ɗ	Ƌ	ƌ	ƍ	Ǝ	Ə	Ɛ	Ƒ	ƒ	Ɠ	Ɣ	ƕ	Ɩ	Ɨ	Ƙ	ƙ	ƚ	ƛ	Ɯ	Ɲ	ƞ	Ɵ	Ơ	ơ	Ƣ	ƣ	Ƥ	ƥ	Ʀ	
01C0	Ƨ	ƨ	Ʃ	ƪ	ƫ	Ƭ	ƭ	Ʈ	Ư	ư	Ʊ	Ʋ	Ƴ	ƴ	Ƶ	ƶ	Ʒ	Ƹ	ƹ	ƺ	ƻ	Ƽ	ƽ	ƾ	ƿ	ƺ	ƻ	Ƽ	ƽ	ƾ	ƿ			
01E0	Ă	ă	Æ	æ	Ɠ	Ɣ	ƕ	Ɩ	Ɨ	Ƙ	ƙ	ƚ	ƛ	Ɯ	Ɲ	ƞ	Ɵ	Ơ	ơ	Ƣ	ƣ	Ƥ	ƥ	Ʀ	Ƨ	ƨ	Ʃ	ƪ	ƫ	Ƭ	ƭ	Ʈ	Ư	
0200	Ă	ă	Â	â	Ë	ë	Ê	ê	Ï	ï	Î	î	Ï	ö	Ŏ	ö	Œ	œ	Ŕ	ŕ	Ŗ	ŗ	Ů	ů	Ű	ű	Ş	ş	Ţ	ţ	Ț	ț	Ĥ	ĥ
0220	Ŧ	ŧ	Ɔ	Ƈ	ƈ	Ɖ	Ɗ	Ƌ	ƌ	ƍ	Ǝ	Ə	Ɛ	Ƒ	ƒ	Ɠ	Ɣ	ƕ	Ɩ	Ɨ	Ƙ	ƙ	ƚ	ƛ	Ɯ	Ɲ	ƞ	Ɵ	Ơ	ơ	Ƣ	ƣ	Ƥ	ƥ
0240	Ƨ	ƨ	Ʃ	ƪ	ƫ	Ƭ	ƭ	Ʈ	Ư	ư	Ʊ	Ʋ	Ƴ	ƴ	Ƶ	ƶ	Ʒ	Ƹ	ƹ	ƺ	ƻ	Ƽ	ƽ	ƾ	ƿ	ƺ	ƻ	Ƽ	ƽ	ƾ	ƿ			



Linguagem de programação

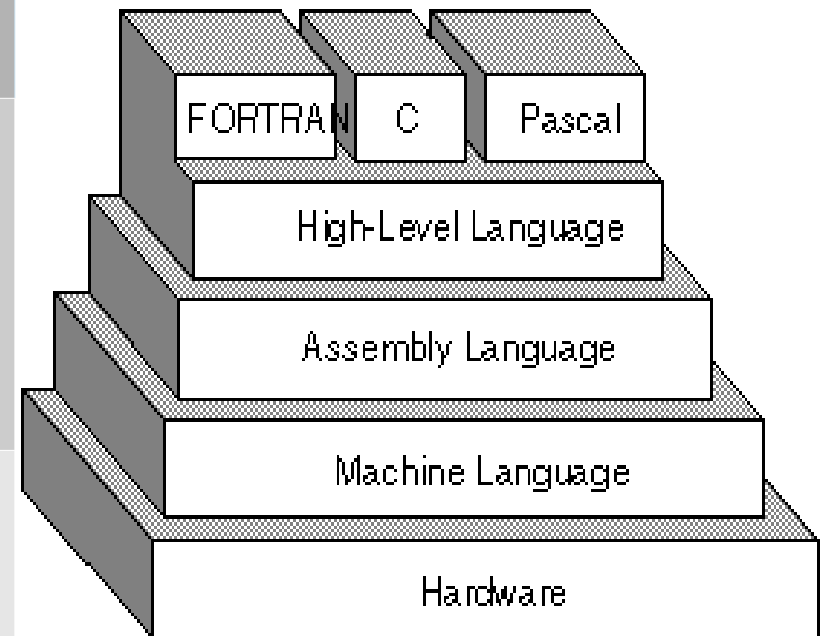
O que é uma linguagem de programação?

É um conjunto limitado de:

- **símbolos** (comandos, identificadores, caracteres, etc)
- **regras de sintaxe** (descrevem de forma precisa ações)

O que é uma linguagem de programação?

Linguagem de máquina	Compreendida pelo computador. Dependente da arquitetura do computador
Linguagem de baixo nível	Utiliza mnemonicos para a representação de ações elementares Ex. Assembler
Linguagem de alto nível	Utiliza instruções próximas da linguagem humana Ex. C, Java, Python, PHP



Programa na linguagem Basic

```
10  REM MOSTRA SEQUENCIA DE NUMEROS de 1 A N
20  INPUT N
30  A = 1
40  PRINT A
50  A = A + 1
60  IF A <= N THEN GOTO 40
```

Programa na linguagem C

```
/* Mostra uma sequencia de numeros de 1 a n */  
Main() {  
    int a, n;  
  
    scanf("%d", &n);  
  
    for(a = 1; a <= n; a++) {  
        printf("%d", a);  
    }  
}
```

Programa na linguagem Python

```
# Mostra uma sequencia de numeros de 1 a n
n = input()

for a in range(1, n+1):
    print(a)
```


Programa na linguagem Java

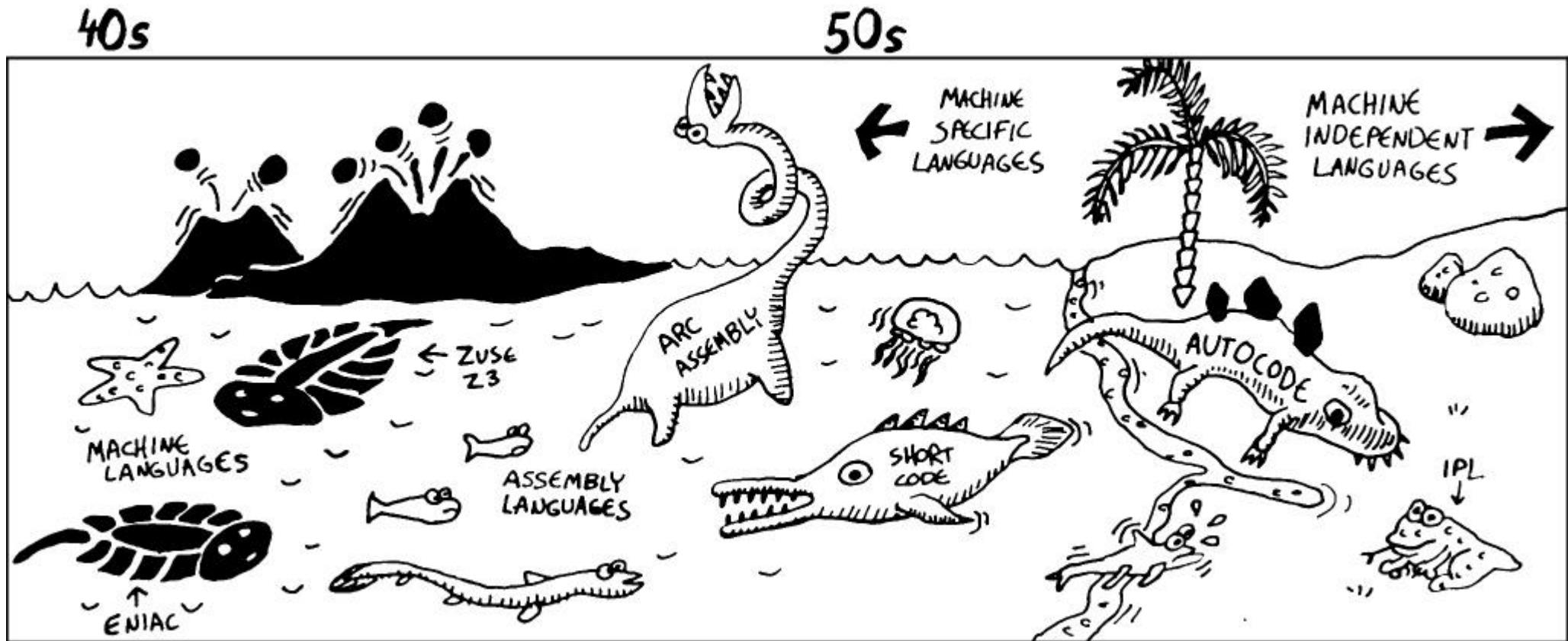
```
import java.util.*;

class MeuPrograma
{
    public static void main (String[] args) {
        Scanner in = new Scanner(System.in);

        int n = in.nextInt();

        for(int a = 1; a <= n; a++) {
            System.out.println(a);
        }
    }
}
```

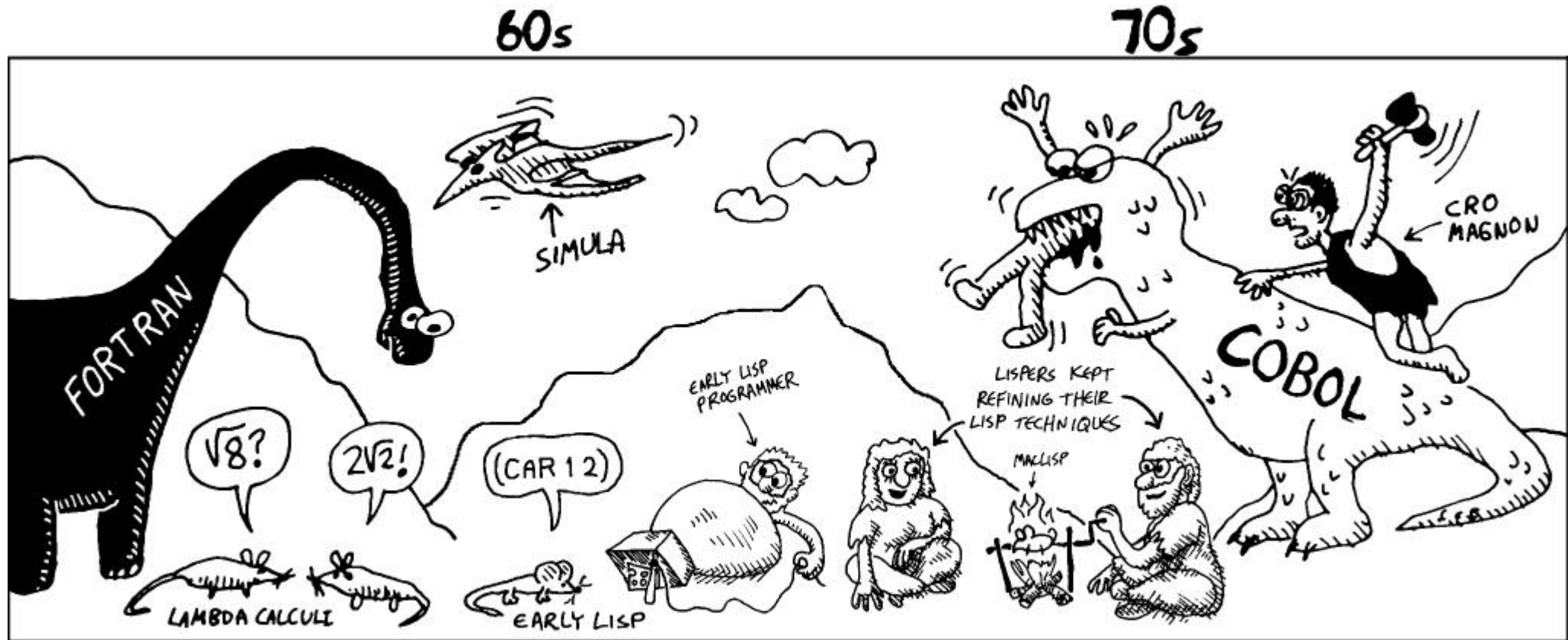
Linguagens de programação



Fonte: "Land of Lisp: Learn to Program in List, One Game at a time". Autor: Conrad Barski.

http://www.amazon.com/gp/product/1593272812/ref=as_li_tf_il?ie=UTF8&camp=1789&creative=9325&creativeASIN=1593272812&linkCode=as2&tag=onionrealit-20

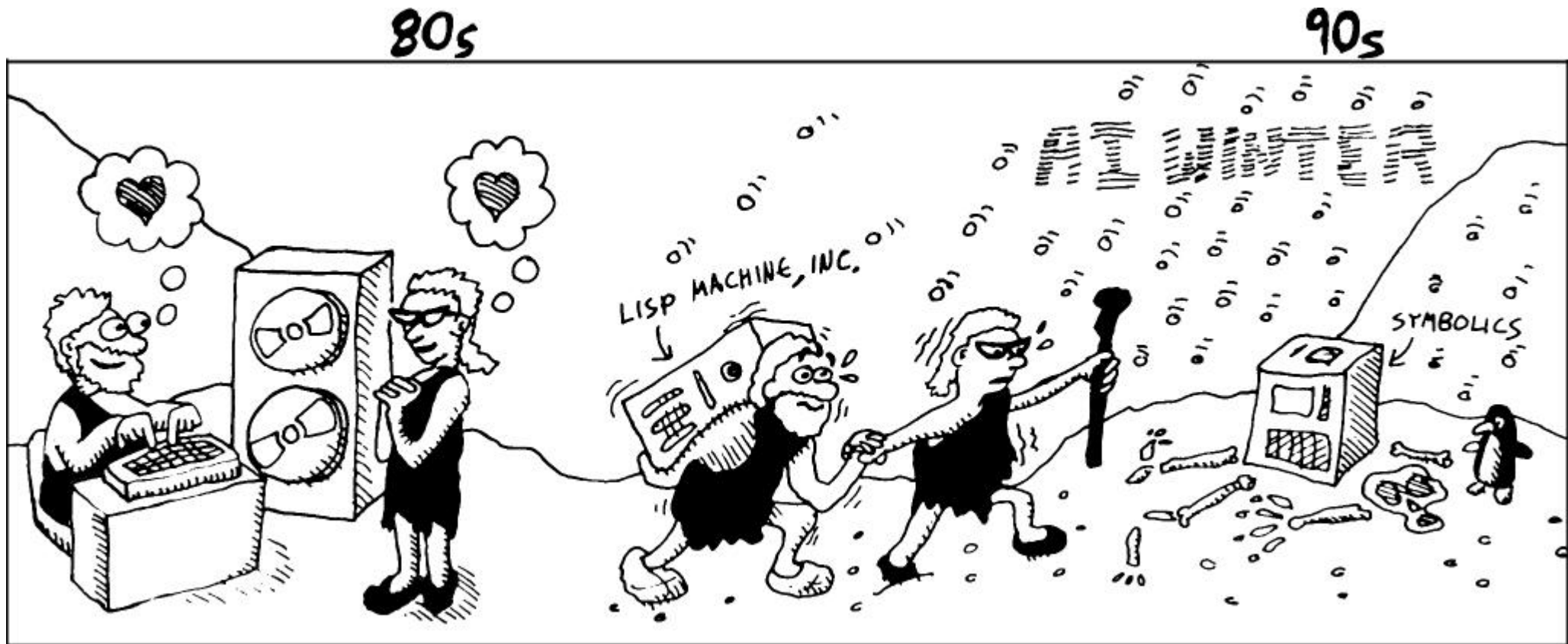
Linguagens de programação



Fonte: "Land of Lisp: Learn to Program in List, One Game at a time". Autor: Conrad Barski.

http://www.amazon.com/gp/product/1593272812/ref=as_li_tf_il?ie=UTF8&camp=1789&creative=9325&creativeASIN=1593272812&linkCode=as2&tag=onionrealit-20

Linguagens de programação

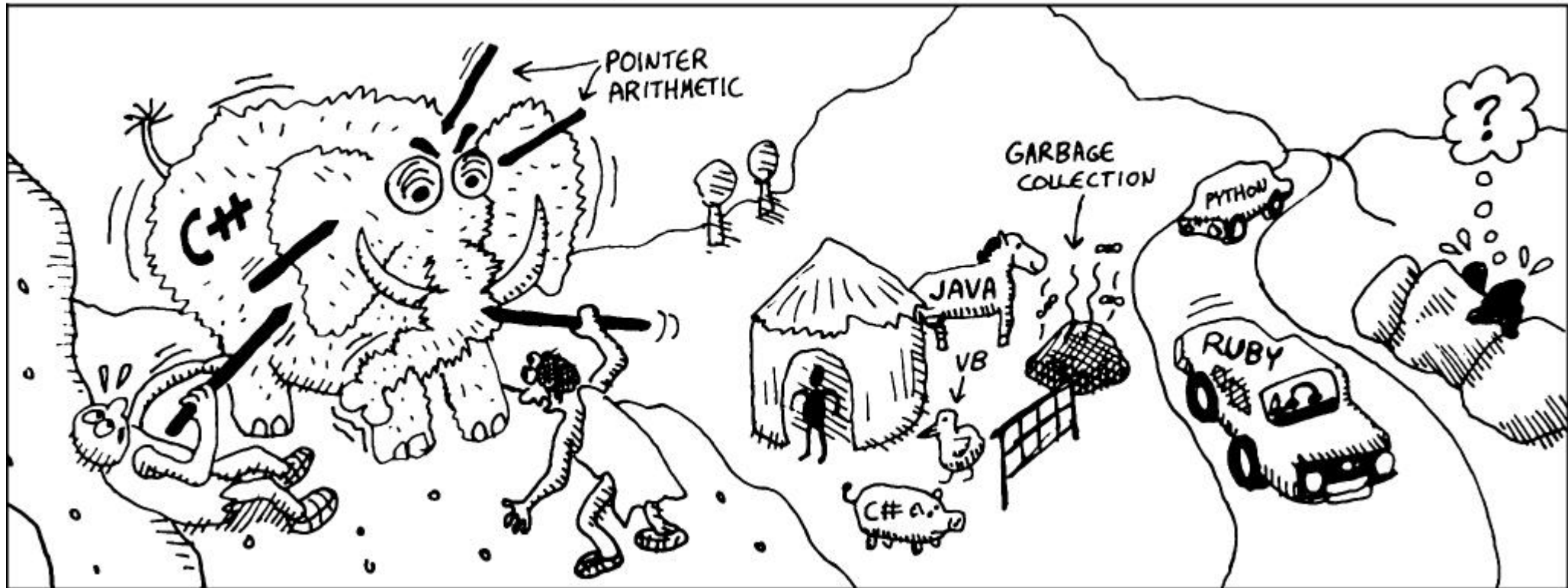


Fonte: "Land of Lisp: Learn to Program in List, One Game at a time". Autor: Conrad Barski.

http://www.amazon.com/gp/product/1593272812/ref=as_li_tf_il?ie=UTF8&camp=1789&creative=9325&creativeASIN=1593272812&linkCode=as2&tag=onionrealit-20

Linguagens de programação

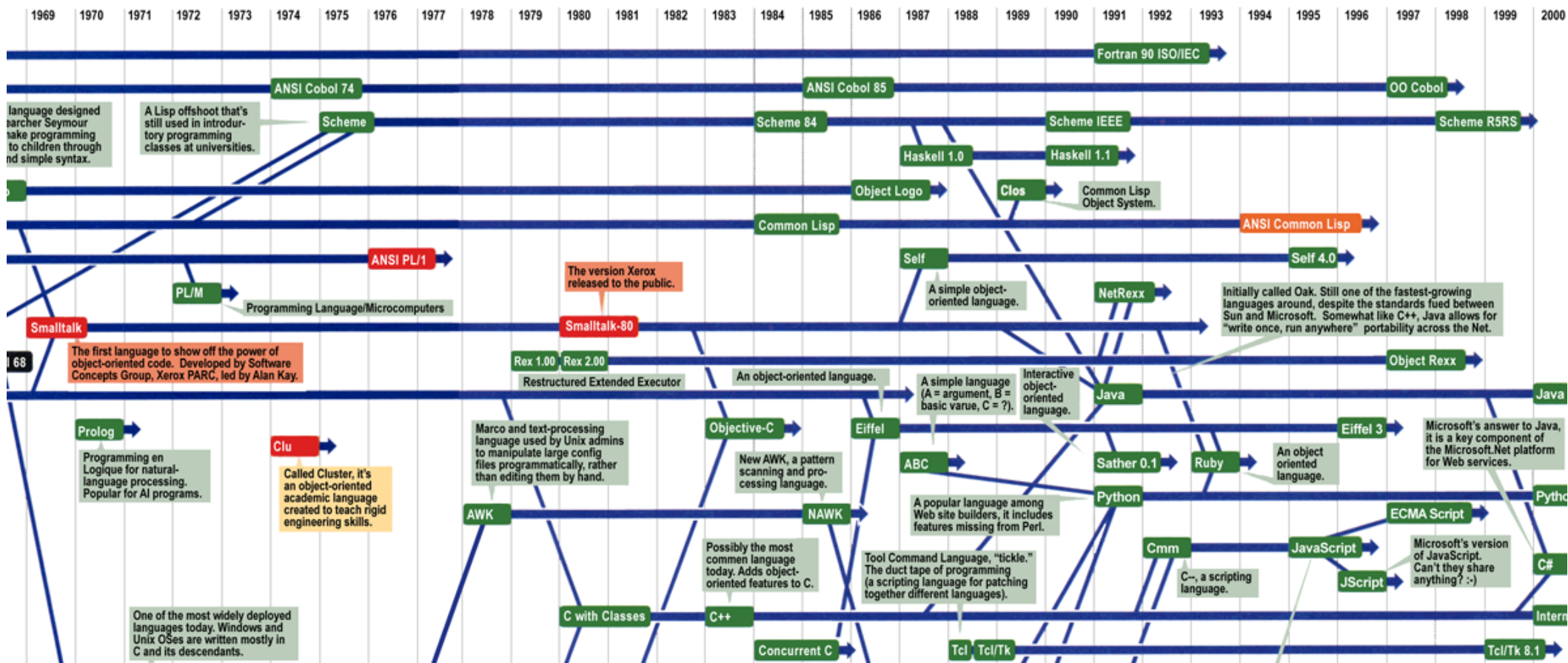
2000



Fonte: "Land of Lisp: Learn to Program in List, One Game at a time". Autor: Conrad Barski.

http://www.amazon.com/gp/product/1593272812/ref=as_li_tf_il?ie=UTF8&camp=1789&creative=9325&creativeASIN=1593272812&linkCode=as2&tag=onionrealit-20

Linguagens de programação

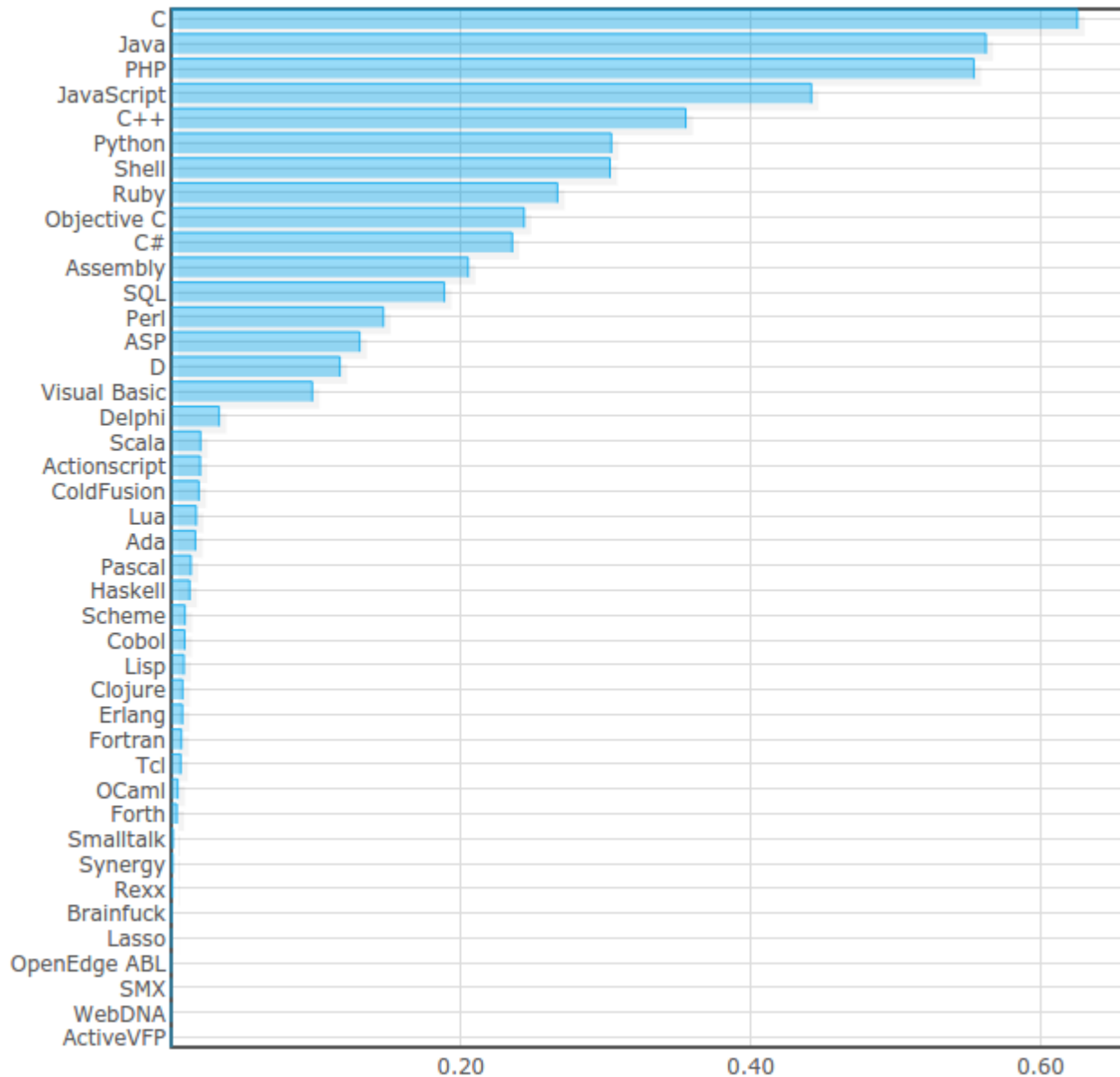


(*) <http://www.digibarn.com/collections/posters/tongues/>

Linguagens de programação



Linguagens de programação



(*) Popularidade das LPs <http://langpop.com/>



Desenvolvimento de programas

Desenvolvendo um programa de computador

Análise:

Estuda-se o enunciado do problema para definir os dados de entrada, o processamento e os dados de saída

Algoritmo (projeto):

Cria-se uma sequência de passos para resolver o problema.

Codificação (implementação):

Traduz-se essa sequência de passos em um programa escrito em uma linguagem que o computador entenda.

Formas de representação de algoritmos

- Descrição narrativa.
- Fluxograma.
- Pseudocódigo (e.g., Portugol).



Tipos de dados

Tipos de dados

Todo o trabalho realizado por um computador **consiste em manipular informações contidas em sua memória.**

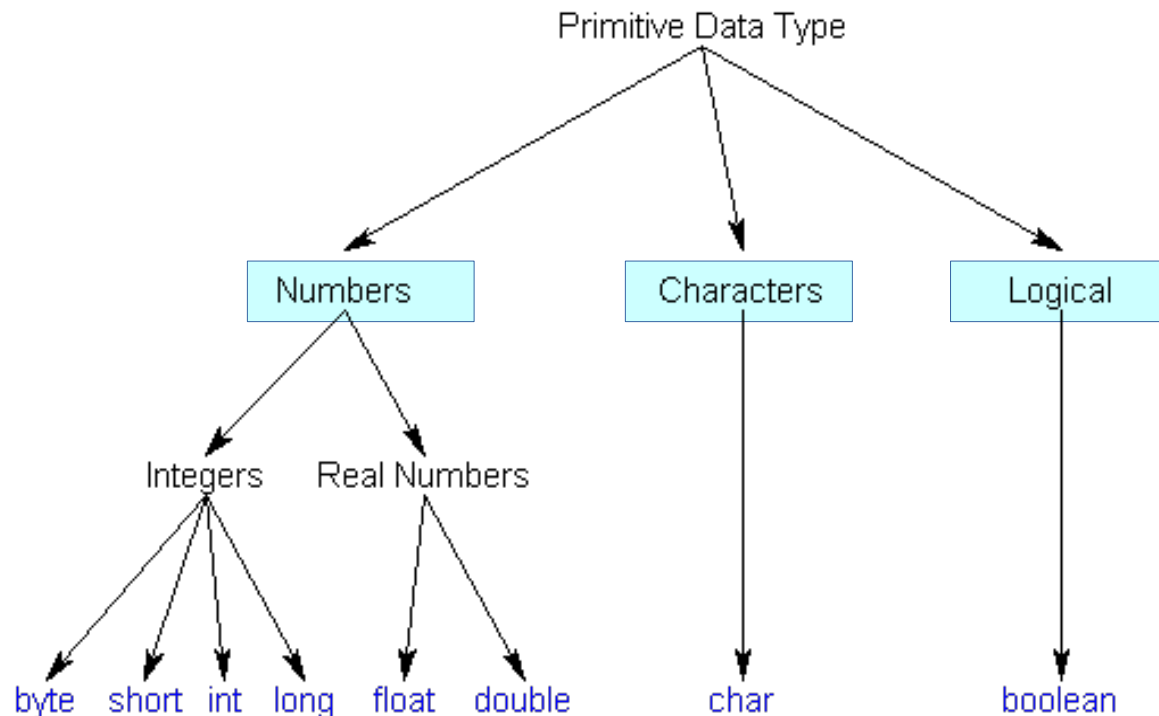
São elas:

- As **instruções**, que comandam o funcionamento da máquina e determinam como os dados são tratados.
- Os **dados**, que correspondem à porção das informações a serem processadas (entradas e saídas).

Tipos de dados em Java

Em geral os dados são classificados em três tipos:

- **numéricos**
- **literais / caracteres**
- **lógicos**



Tipos de dados em Java

	Data Type	Bits	Range
Inteiros	byte	8	-2^7 to 2^7-1 (-128 to 127)
	short	16	-2^{15} to $2^{15}-1$ (-32,768 to 32,767)
	int	32	-2^{31} to $2^{31}-1$ (-2,147,483,648 to 2,147,483,647)
	long	64	-2^{63} to $2^{63}-1$ (-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)
Reais	float	32	1.40129846432481707e-45 to 3.40282346638528860e+38 (positive or negative) (single-precision 32-bit IEEE 754 floating point)
	double	64	4.94065645841246544e-324d to 1.79769313486231570e+308d (positive or negative) (double-precision 64-bit IEEE 754 floating point)
	char	16 (unsigned)	0 to 65,535
	boolean	1	true, false

São os únicos tipos de dados?

Não!

Os tipos de dados que vimos até agora são os chamados **tipos primitivos**.

Podem ser definidos novos tipos de dados.

Inteiros

Reais

Data Type	Bits	Range
byte	8	-2^7 to 2^7-1 (-128 to 127)
short	16	-2^{15} to $2^{15}-1$ (-32,768 to 32,767)
int	32	-2^{31} to $2^{31}-1$ (-2,147,483,648 to 2,147,483,647)
long	64	-2^{63} to $2^{63}-1$ (-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)
float	32	$1.40129846432481707e-45$ to $3.40282346638528860e+38$ (positive or negative) (single-precision 32-bit IEEE 754 floating point)
double	64	$4.94065645841246544e-324d$ to $1.79769313486231570e+308d$ (positive or negative) (double-precision 64-bit IEEE 754 floating point)
char	16 (unsigned)	0 to 65,535
boolean	1	true, false



Prática em sala



share

save

run



```
1 class Main {  
2  
3   public static void main(String[] args) {  
4     System.out.println("hello world");  
5  
6   }  
7  
8 }  
9
```

input

clear

```
java version "1.8.0_31"  
Java(TM) SE Runtime Environment (build 1.8.0_31-  
b13)  
Java HotSpot(TM) 64-Bit Server VM (build 25.31-  
b07, mixed mode)  
hello world
```

Teste



```
class Main {  
    public static void main(String[] args) {  
        int t;  
        t = 1 + 10;  
        System.out.println(t);  
    }  
}
```

input ↵

clear ✕

```
java version "1.8.0_31"  
Java(TM) SE Runtime Environment (build  
1.8.0_31-b13)  
Java HotSpot(TM) 64-Bit Server VM (build  
25.31-b07, mixed mode)
```

```
11
```

Palavras “reservadas” que não podem ser usadas como nomes de variáveis

Java Keywords				
abstract	boolean	break	byte	case
catch	char	class	continue	default
do	double	else	extends	false
final	finally	float	for	if
implements	import	instanceof	int	interface
long	native	new	null	package
private	protected	public	return	short
static	super	switch	synchronized	this
throw	throws	transient	true	try
void	volatile	while		
<i>Keywords that are reserved but not used by Java</i>				
const	goto			