

# Tipos de dados

---

ALGORITMOS E ESTRUTURAS DE DADOS III

Prof. Marcos André S. Kutova

# Tipos primitivos de dados

## TIPOS INTEIROS

- byte      Número inteiro de 8 bits com sinal  
-128 a +127
- short     Número inteiro de 16 bits com sinal  
-32.768 a +32.767
- int        Número inteiro de 32 bits com sinal  
-2.147.483.648 a 2.147.483.647
- long       Número inteiro de 64 bits com sinal  
-9.223.372.036.854.775.808 a  
+9.223.372.036.854.775.807

# Tipos primitivos de dados

## TIPOS REAIS

- float      Número de ponto flutuante de 32 bits com sinal  
                  $\pm 1,40129846e-45$  a  
                  $\pm 3,40282347e+38$
- double    Número de ponto flutuante de 64 bits com sinal  
                  $\pm 4,94065645841246544e-324$  a  
                  $\pm 1,79769313486231570e+308$

	Bits	Sinal	Expoente	Fração
Precisão simples (float)		1	8	23
Precisão dupla (double)		1	11	52

# Tipos primitivos de dados

## OUTROS TIPOS

- char            Carácter Unicode de 16 bits, sem sinal, com valor de 0 a 65.535 (em Java) .
- boolean       Variável lógica que indica falso ou verdadeiro

# **Campos numéricos**

# Campos numéricos

- Decimal – 11
  - Formato usado na comunicação humana. Sempre será o formato a ser usado nas interfaces com o usuário. O computador, no entanto, não armazena dados no formato decimal.
- Binário – 0b00001011
  - Formato usado nas operações dos componentes eletrônicos (CPU, memória, etc.).
- Hexadecimal – 0x0B
  - Formato usado nas representações de bytes
- String – “11”
  - Formato usado na representações textuais. A string usada no exemplo é armazenada por meio dos bytes 0x31 0x31. Cada dígito é, na verdade, um caráter com valor 49 (0x31 em hexadecimal).

# Exemplos

- Quantidade de produtos em um cupom fiscal de uma compra de supermercado:
  - short (até 32.767, pois 127 pode ser pouco)
- Contador de registros de operações bancárias
  - long (são milhões de registros por dia...)
- Preço de um livro
  - float (precisão suficiente para a faixa de valores)

**Datas e horas**



# Datas e horas

- Em Java, uma data é um campo numérico que indica a quantidade de milissegundos passadas desde as 00:00 de 01/01/1970.
  - $1.439.500.953.888 = 13/08/2015\ 21:22:33.888$
- No Excel, uma data possui uma parte inteira e uma parte decimal. A parte inteira, indica o número de dias passados desde 01/01/1900. A parte decimal indica a fração de um dia.
  - $42229,89067 = 13/08/2015\ 21:22:33.888$
- A vantagem de se armazenar datas e horas como números é a facilidade de se fazer operações matemáticas (e a economia de bytes)

# **Os “falsos” campos numéricos**

# Os “falsos” campos numéricos

- Exemplos
  - CEP: 30535-901
  - CPF: 072.424.817-08
  - CNPJ: 12.454.755/0002-26
  - Telefone: (31) 3319-4444
- Exercício:
  - Apresente uma alternativa para o armazenamento de placas de carro.

**Identificadores**

# Identificadores

- Identificadores devem ser números sequenciais exclusivos que não carregam significado
  - A alteração de um identificador compromete todas as referências à entidade
  - O formato numérico assegura o melhor aproveitamento da capacidade de armazenamento
- Exercício:
  - A PUC Minas recebe cerca de 5.000 novos alunos por semestre. Durante quanto tempo identificadores de matrículas dos tipos short, int e long serão válidos sem o reaproveitamento de números?

# **Sistemas de codificação**

# Sistemas de codificação

- Representação binária (ou hexa) de cada símbolo
- ASCII (anos 1960)  
*American Standard Code for Information Interchange*
  - Representação de 128 símbolos (7 bits)
    - $100\ 0001 = 65 = 0x41 = 'A'$
  - O 8º bit (mais significativo) era usado como bit de paridade nas comunicações
  - Faltavam os caracteres latinos acentuados, os chineses, os cirílicos (russos), ...

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]	48	30	110000	60	0	96	60	1100000	140	`
1	1	1	1	[START OF HEADING]	49	31	110001	61	1	97	61	1100001	141	a
2	2	10	2	[START OF TEXT]	50	32	110010	62	2	98	62	1100010	142	b
3	3	11	3	[END OF TEXT]	51	33	110011	63	3	99	63	1100011	143	c
4	4	100	4	[END OF TRANSMISSION]	52	34	110100	64	4	100	64	1100100	144	d
5	5	101	5	[ENQUIRY]	53	35	110101	65	5	101	65	1100101	145	e
6	6	110	6	[ACKNOWLEDGE]	54	36	110110	66	6	102	66	1100110	146	f
7	7	111	7	[BELL]	55	37	110111	67	7	103	67	1100111	147	g
8	8	1000	10	[BACKSPACE]	56	38	111000	70	8	104	68	1101000	150	h
9	9	1001	11	[HORIZONTAL TAB]	57	39	111001	71	9	105	69	1101001	151	i
10	A	1010	12	[LINE FEED]	58	3A	111010	72	:	106	6A	1101010	152	j
11	B	1011	13	[VERTICAL TAB]	59	3B	111011	73	;	107	6B	1101011	153	k
12	C	1100	14	[FORM FEED]	60	3C	111100	74	<	108	6C	1101100	154	l
13	D	1101	15	[CARRIAGE RETURN]	61	3D	111101	75	=	109	6D	1101101	155	m
14	E	1110	16	[SHIFT OUT]	62	3E	111110	76	>	110	6E	1101110	156	n
15	F	1111	17	[SHIFT IN]	63	3F	111111	77	?	111	6F	1101111	157	o
16	10	10000	20	[DATA LINK ESCAPE]	64	40	1000000	100	@	112	70	1110000	160	p
17	11	10001	21	[DEVICE CONTROL 1]	65	41	1000001	101	A	113	71	1110001	161	q
18	12	10010	22	[DEVICE CONTROL 2]	66	42	1000010	102	B	114	72	1110010	162	r
19	13	10011	23	[DEVICE CONTROL 3]	67	43	1000011	103	C	115	73	1110011	163	s
20	14	10100	24	[DEVICE CONTROL 4]	68	44	1000100	104	D	116	74	1110100	164	t
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]	69	45	1000101	105	E	117	75	1110101	165	u
22	16	10110	26	[SYNCHRONOUS IDLE]	70	46	1000110	106	F	118	76	1110110	166	v
23	17	10111	27	[ENG OF TRANS. BLOCK]	71	47	1000111	107	G	119	77	1110111	167	w
24	18	11000	30	[CANCEL]	72	48	1001000	110	H	120	78	1111000	170	x
25	19	11001	31	[END OF MEDIUM]	73	49	1001001	111	I	121	79	1111001	171	y
26	1A	11010	32	[SUBSTITUTE]	74	4A	1001010	112	J	122	7A	1111010	172	z
27	1B	11011	33	[ESCAPE]	75	4B	1001011	113	K	123	7B	1111011	173	{
28	1C	11100	34	[FILE SEPARATOR]	76	4C	1001100	114	L	124	7C	1111100	174	
29	1D	11101	35	[GROUP SEPARATOR]	77	4D	1001101	115	M	125	7D	1111101	175	}
30	1E	11110	36	[RECORD SEPARATOR]	78	4E	1001110	116	N	126	7E	1111110	176	~
31	1F	11111	37	[UNIT SEPARATOR]	79	4F	1001111	117	O	127	7F	1111111	177	[DEL]
32	20	100000	40	[SPACE]	80	50	1010000	120	P					
33	21	100001	41	!	81	51	1010001	121	Q					
34	22	100010	42	"	82	52	1010010	122	R					
35	23	100011	43	#	83	53	1010011	123	S					
36	24	100100	44	\$	84	54	1010100	124	T					
37	25	100101	45	%	85	55	1010101	125	U					
38	26	100110	46	&	86	56	1010110	126	V					
39	27	100111	47	'	87	57	1010111	127	W					
40	28	101000	50	(	88	58	1011000	130	X					
41	29	101001	51	)	89	59	1011001	131	Y					
42	2A	101010	52	*	90	5A	1011010	132	Z					
43	2B	101011	53	+	91	5B	1011011	133	[					
44	2C	101100	54	,	92	5C	1011100	134	\					
45	2D	101101	55	-	93	5D	1011101	135	]					
46	2E	101110	56	.	94	5E	1011110	136	^					
47	2F	101111	57	/	95	5F	1011111	137	_					



# Codificações de 8 bits

80	Ä	Å	Ç	É	Ñ	Ö	Ü	á	à	â	ä	ã	â	ç	é	è
90	ê	ë	í	î	ï	ñ	ô	õ	ö	ö	ö	ú	û	ü	ü	ü
A0	†	°	¢	£	§	•	¶	ß	®	©	™	ˆ	˜	˘	˙	˚
B0	∞	±	≤	≥	¥	μ	∂	Σ	Π	Π	∫	≈	≈	Ω	æ	ø
C0	¿	¡	¬	√	ƒ	≈	Δ	«	»	...		À	Á	Ã	œ	ø
D0	-	-	“	”	‘	’	÷	◊	ÿ	ÿ	/	℥	℥	℥	℥	℥
E0	‡	•	,	„	%	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
F0	♥	ò	ú	û	ù	1	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ

A0	ı	¢	£	¤	¥	ı	§	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ	ˆ
B0	°	±	2	3	ˆ	μ	¶	•	1	0	»	¼	½	¾	¿	
C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F0	ä	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Mac Roman

ISO-8859-5 (Cyrillic)

ISO-8859-1 (Latin-1)

A0	Ë	Ђ	Ѓ	€	Š	İ	İ	Ј	Љ	Њ	Ћ	Ќ	Ќ	Ќ	Ќ	Ќ
B0	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
C0	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
D0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
E0	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
F0	№	ë	ђ	ѓ	€	š	ı	ı	ј	љ	њ	ћ	ќ	ќ	ќ	ќ

**Unicode**

# Unicode

- Sistema de codificação capaz de representar texto de qualquer sistema de escrita existente (mais de 100 mil símbolos)
  - Sistemas latino, árabe, cirílico, chinês, hebraico, ...
  - Símbolos matemáticos, geométricos, musicais, setas, ícones, emojis, ...
  - Escrita cuneiforme, Braille, runas, élfico, ...
  - Próximos: hieróglifos egípcios, alfabeto babilônico, ...
- Referências:
  - <http://unicode.org/>
  - <http://www.unicodetables.com/>



# Unicode

- **UTF-8** (*8-bit Unicode Transformation Format*)
  - Codificação Unicode de comprimento variável (1 a 4 bytes), que pode representar qualquer caráter (*code point*) do padrão.
  - Padrão usado na Internet e na Web
- UTF-16
  - Codificação de 2 ou 4 bytes, usada principalmente para escrita em idiomas dos países asiáticos
- UTF-32
  - Codificação de comprimento fixo de 4 bytes

# UTF-8

Bits do <i>code point</i>	Primeiro <i>code point</i>	Último <i>code point</i>	Bytes	Byte 1	Byte 2	Byte 3	Byte 4
<b>7</b>	U+0000	U+007F	1	0xxxxxxx			
<b>11</b>	U+0080	U+07FF	2	110xxxxx	10xxxxxx		
<b>16</b>	U+0800	U+FFFF	3	1110xxxx	10xxxxxx	10xxxxxx	
<b>21</b>	U+10000	U+1FFFFFF	4	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

- 1 byte - Tabela ASCII
- 2 bytes - Caracteres latinos, hebraicos, gregos, ...
- 3 bytes - Caracteres chineses, japoneses, coreanos, ...
- 4 bytes - Alguns outros caracteres e símbolos

# UTF-8

## Representação de strings

C	o	n	c	e	i	ç	ã	o
43	6F	6E	63	65	69	C3A7	C3A3	6F

9 caracteres – 11 bytes

# Strings

# Strings

- Cadeias de caracteres usadas para representar um palavra, uma informação ou um texto
- Valor em bytes de cada caráter depende do sistema de codificação
  - ISO-8859-1: cada caráter usa 1 byte
  - UTF-8: cada caráter por usar de 1 a 4 bytes
- Campos que contém strings podem ser de tamanho fixo ou variável

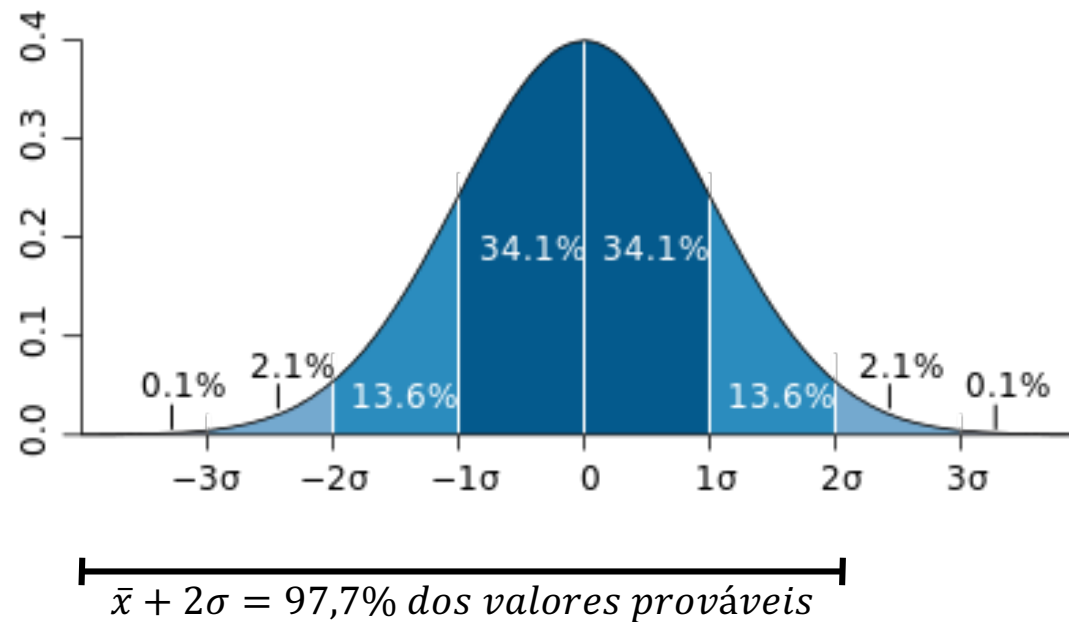


# Strings de tamanho fixo

- Como determinar o tamanho do campo
  - Campos de tamanho naturalmente fixo
    - CPF, CEP, UF, Telefone, Conta Corrente, EAN-13, ...
  - Campos limitados externamente
    - tamanho do relatório, da etiqueta, da tela, ...
  - Probabilidade estatística

# Strings de tamanho fixo

- Distribuição normal



# Exemplo

- Email
  - Especificação ([nome@dominio.com](#))
    - Nome – até 64 caracteres
    - Domínio – até 254 caracteres
    - Total:  $64 + 1 + 254 = 319$  caracteres
  - Estatística
    - Análise de 323 emails
    - Mínimo = 12
    - Máximo = 47
    - Média = 23,0
    - Desvio padrão = 5,2

Faixas:

$$0 \text{ a } (\bar{x} + 1\sigma) = 84,1\% = 28,2$$

$$0 \text{ a } (\bar{x} + 2\sigma) = 97,7\% = 33,4$$

$$0 \text{ a } (\bar{x} + 3\sigma) = 99,8\% = 38,6$$

# Strings de tamanho variável

- Com indicador de tamanho

IND.TAM.		C	o	n	c	e	i	ç		ã		o
00	0B	43	6F	6E	63	65	69	C3	A7	C3	A3	6F

- Com delimitador

C	o	n	c	e	i	ç		ã		o	\n
43	6F	6E	63	65	69	C3	A7	C3	A3	6F	0A

- Delimitadores tradicionais: \n \0 ; | (e outros)

# **Registros de tamanho fixo**

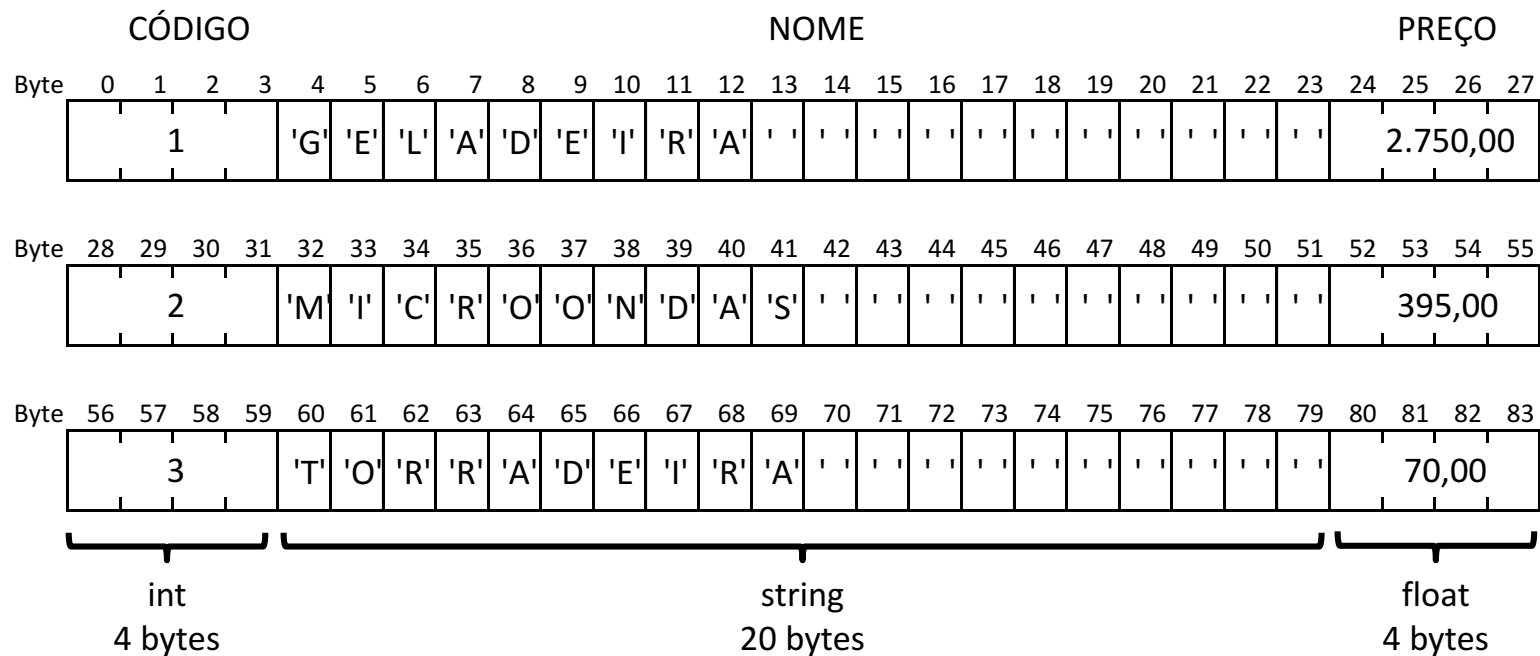
# Registros

- Conjunto de dados (ou campos) sobre uma mesma entidade

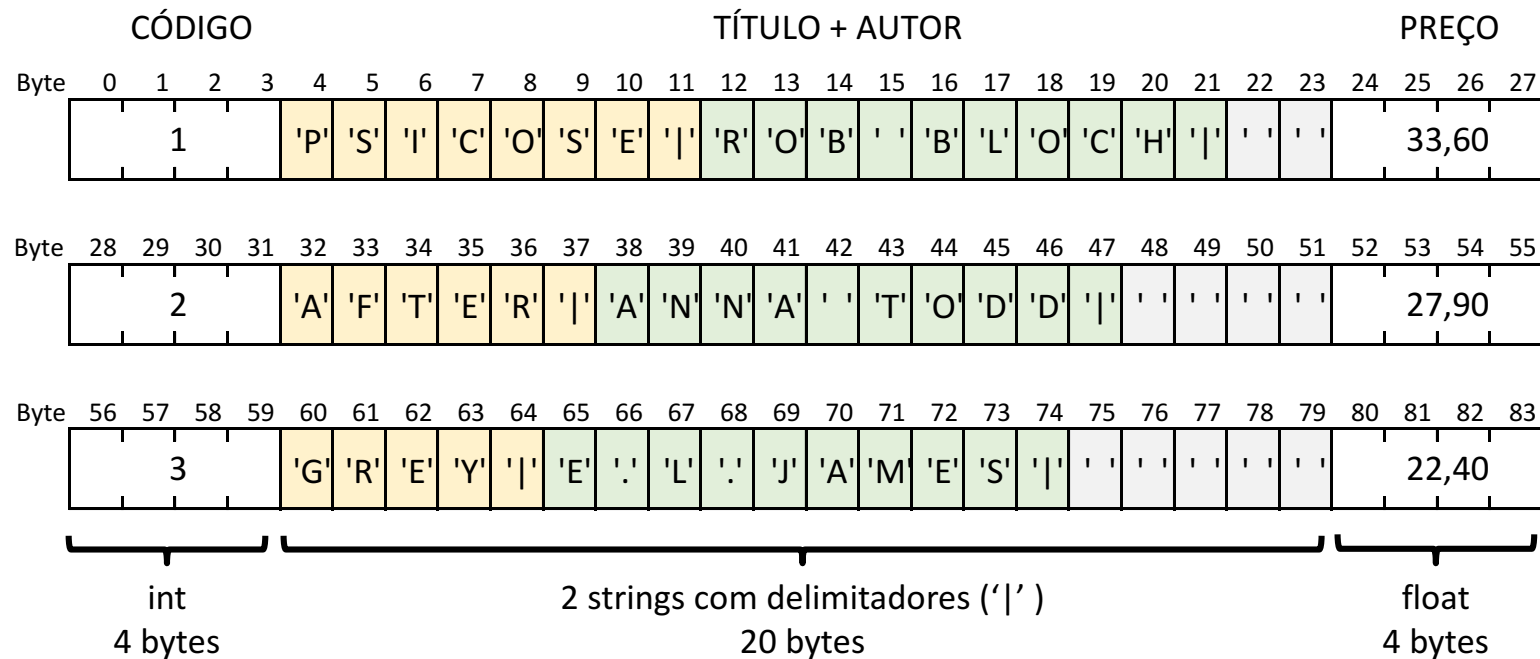
Código	Nome	Capacidade	Altura (cm)	Largura (cm)	Profundidade (cm)	Cor	Preço
1981091	Refrigerador Electrolux Frost Free DW52X com dispenser externo de água 456 L	456 litros	193	71	76	Inox	R\$ 2.744,10
3608237	Forno de Microondas LG Grill Easy Clean MH7053R Espelhado - 30 L	30 litros	29	51	45	Branco	R\$ 395,43
2170868	Torradeira Britânia Tosta Pane	2 torradas	18	17	29	Inox - Preto	R\$ 71,91

# Registros de tamanho fixo

- Registros de tamanho fixo com campos de tamanho fixo



- Registros de tamanho fixo com campos de tamanho variável





# Registros de tamanho fixo

- **Exercício:**
  - Compare os tamanhos de um registro de tamanho fixo, (1) usando campos de tamanho fixo e (2) usando campos de tamanho variável, que descreva um endereço, com os seguintes campos:
    - Rua, número, complemento, bairro, CEP, cidade, estado

# **Registros de tamanho variável**

# Registros de tamanho variável

- Na maioria das situações, a quantidade de campos será fixa e sempre na mesma ordem
  - Alternativas para processamento:
    - Leitura/escrita dos campos na ordem predeterminada
    - Uso de indicadores de tamanho
    - Uso de delimitadores de registro
- Casos especiais
  - Campos com múltiplos valores
  - Campos selecionados: uso de identificadores de campos

# Registros de tamanho variável

- Campos fixos
  - Os campos de cada registro são sempre os mesmos (mesmo se vazios)
  - Os campos de tamanho variável devem adotar indicador de tamanho ou delimitador

CÓDIGO; TÍTULO; AUTOR; PREÇO

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
		1			7	'P'	'S'	'I'	'C'	'O'	'S'	'E'	9	'R'	'O'	'B'	'	'B'	'L'	'O'	'C'	'H'			33,6				2	

Byte	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
	5	'A'	'F'	'T'	'E'	'R'	9	'A'	'N'	'N'	'A'	'	'T'	'O'	'D'	'D'			27,90			3			4	'G'	'R'	'E'	'Y'	9

Byte	60	61	62	63	64	65	66	67	68	69	70	71	72
	'E'	'.'	'L'	'.'	'J'	'A'	'M'	'E'	'S'				22,40

Nesse exemplo, apenas o acesso sequencial aos registros é possível. No entanto, o acesso aleatório pode ser feito por meio do uso de índices.

# Registros de tamanho variável

- Indicadores de tamanho de registro
  - Úteis para evitar o processamento individual dos campos, quando o registro não é o desejado
    - ao se saber o tamanho do registro, pode-se saltar todos os seus bytes

CÓDIGO; TÍTULO; AUTOR; PREÇO																														
Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
	26				1			7	'P'	'S'	'I'	'C'	'O'	'S'	'E'	9	'R'	'O'	'B'	' '	'B'	'L'	'O'	'C'	'H'	33,6				
Byte	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57		
	24				2			5	'A'	'F'	'T'	'E'	'R'	9	'A'	'N'	'N'	'A'	' '	' '	'T'	'O'	'D'	'D'	27,90					
Byte	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84			
	23				3			4	'G'	R'	E'	'Y'	9	'E'	'.'	'L'	'.'	'J'	'A'	'M'	'E'	'S'	22,40							

# Registros de tamanho variável

- Delimitadores de registro
  - Algumas linguagens de programação oferecem o método `readLine()` que facilita a leitura de um registro inteiro
  - Um método como o `split()` também pode ser usado na separação de vários registros

TÍTULO; AUTOR; PREÇO

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
		1			7	'P'	'S'	'I'	'C'	'O'	'S'	'E'	9	'R'	'O'	'B'	' '	'B'	'L'	'O'	'C'	'H'		33,6			'\n'

Byte	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
		2			5	'A'	'F'	'T'	'E'	'R'	9	'A'	'N'	'N'	'A'	' '	'T'	'O'	'D'	'D'		27,90			'\n'

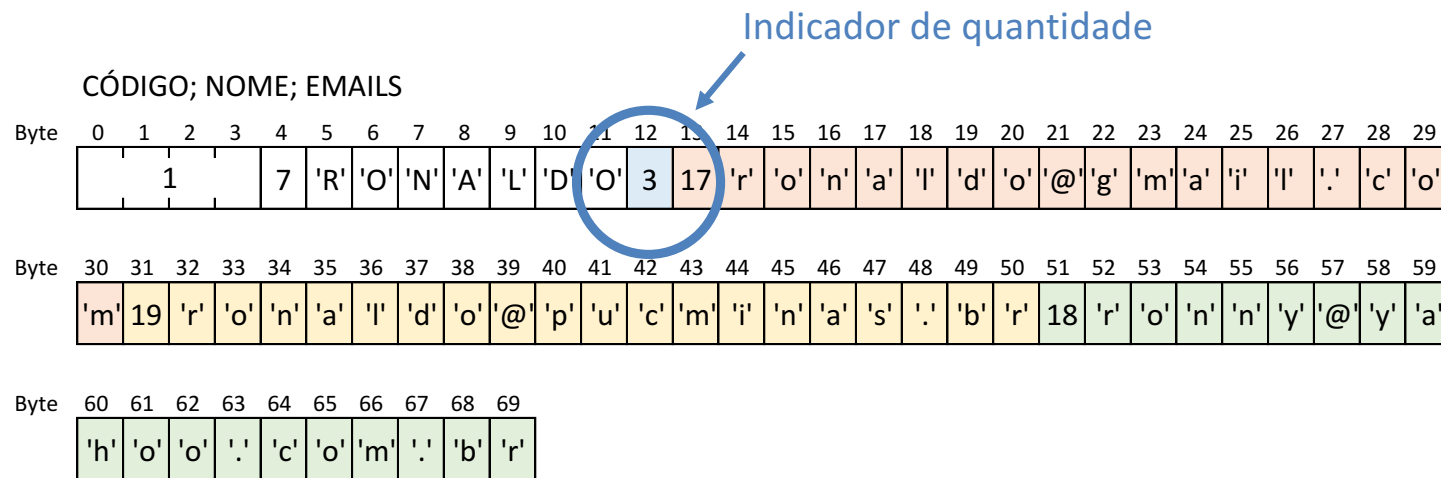
  

Byte	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
		3			4	'G'	'R'	'E'	'Y'	9	'E'	'.'	'L'	'.'	'J'	'A'	'M'	'E'	'S'		22,40			'\n'

## **Outros tipos de campos**

# Campos com múltiplos valores

- Exemplo – emails de um cliente:





# Campos com múltiplos valores


- Alternativa:
  - Usar uma string de tamanho variável e usar um separador entre os valores.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
33	'j'	'o'	's'	'é'	'@'	'g'	'm'	'a'	'i'	'l'	'.'	'c'	'o'	'm'	;	'j'	'o'		
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34					
's'	'é'	'@'	'h'	'o'	't'	'm'	'a'	'i'	'l'	'.'	'c'	'o'	'm'						

# **Campos com identificadores**

- Normalmente, os campos sempre os mesmos e aparecem na mesma ordem
- Em algumas situações, porém, apenas alguns campos (de um conjunto muito grande) são usados
- Em outras situações, um campo pode receber informações diferentes:
  - Contato pode ser email, telefone, pager, etc.

# Campos com identificadores

 <b>MINISTÉRIO DA PREVIDÊNCIA E ASSISTÊNCIA SOCIAL-MPAS</b> <b>INSTITUTO NACIONAL DO SEGURO SOCIAL - INSS</b> <b>GUIA DA PREVIDÊNCIA SOCIAL - GPS</b>		3. CÓDIGO DE PAGAMENTO	1600
		4. COMPETÊNCIA	11/2002
		5. IDENTIFICADOR	4561098732
1. NOME OU RAZÃO SOCIAL/FONE/ENDEREÇO:		6. VALOR DO INSS	45,20
Vera Lúcia da Silva R. Palmital, 114 - Cambu - Curitiba - PR		7.	
		8.	
2. VENCIMENTO (Uso exclusivo INSS)		9. VALOR DE OUTRAS ENTIDADES	
<b>ATENÇÃO:</b> É vedada a utilização de GPS para recolhimento de receita de valor inferior ao estipulado em Resolução publicada pelo INSS. A receita que resultar valor inferior deverá ser adicionada à contribuição ou importância correspondente nos meses subsequentes, até que o total seja igual ou superior ao valor mínimo fixado		10. ATM/MULTA E JUROS	
		11. TOTAL	45,20
12. AUTENTICAÇÃO BANCÁRIA			
<small>Instruções para preenchimento no verso. SÃO DOMINGOS S.A. - INDÚSTRIA GRÁFICA - AV. MIGUEL ESTEFHO, 384 - CATANDUVA - SP - CNPJ 47.064.730/0001-88</small>			

Byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
	1	53	'V'	'e'	'r'	'a'	' '	'L'	'ú'	'c'	'i'	'a'	' '	'd'	'a'	' '	'S'	'i'	'l'	'v'	'a'	'\n'	'R'	'.'	'P'	'a'	'l'	'm'	'i'	't'	'a'	'"

Byte	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
	','	'1'	'1'	'4'	'-'	'C'	'a'	'm'	'b'	'u'	'i'	'-'	'C'	'u'	'r'	'i'	't'	'i'	'b'	'a'	'-'	'P'	'R'	3	4	'1'	'6'	'0'	'0'	4	7	'1'

Byte	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	
	'1'	'/'	'2'	'0'	'0'	'2'	5	10	'4'	'5'	'6'	'1'	'0'	'9'	'8'	'7'	'3'	'2'	6			45,20		11			45,20		

# **Arquivos CSV**

# Comma Separated Values (CSV)

- Formato de arquivo (RFC 4180) comumente usado no intercâmbio de dados entre sistemas



ID;Produto;Preço

1;Geladeira Branca;1.300,00

2;"TV LED 60"" Full HD";3.300,00



ID,Product,Price

1,Refrigerator,1300.00

2,"60"" Full HD LED TV",3300.00