Serialização

- Conversão de um objeto complexo em uma sequência de tipos primitivos
  - As linguagens de programação já oferecem métodos para escrever os tipos primitivos (+ string) como uma sequência de bytes

Entidade:

```
Cliente {
  short    ID;
  String   nome;
  String[] emails;
}
```

Registro:

```
ID             → short de 2 bytes
nome           → string UTF-8 com indicador de tamanho
emails.length  → byte para indicar quantidade de emails
emails[0]      → string UTF-8 com indicador de tamanho
emails[1]      → string UTF-8 com indicador de tamanho
...
emails[n-1]    → string UTF-8 com indicador de tamanho
```

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
  ±1,40129846e-45 a ±3,40282347e+38
- double Número de ponto flutuante de 64 bits com sinal
  ±4,94065645841246544e-324 a ±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

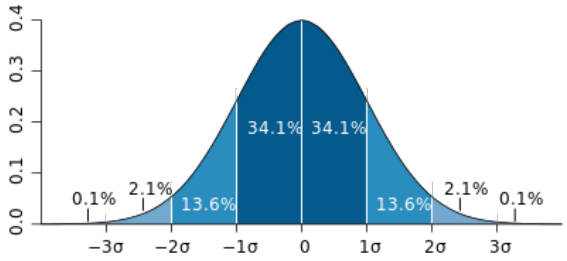
- char Caractere ASCII (1 byte) ou Unicode de 16 bits (2 bytes)
- boolean Variável lógica que indica falso ou verdadeiro (1 byte)

ALÉM DOS TIPOS PRIMITIVOS

- string Sequência de caracteres de tamanho variável em UTF-8, usando indicador de tamanho ou caractere delimitador

Strings de tamanho fixo

- Devem ser completadas com espaços em branco
- Como determinar o tamanho do campo
  - Campos de tamanho naturalmente fixo (CPF, CEP, UF, Telefone, ...)
  - Campos limitados externamente (relatório, etiqueta, tela, ...)
  - Probabilidade estatística



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

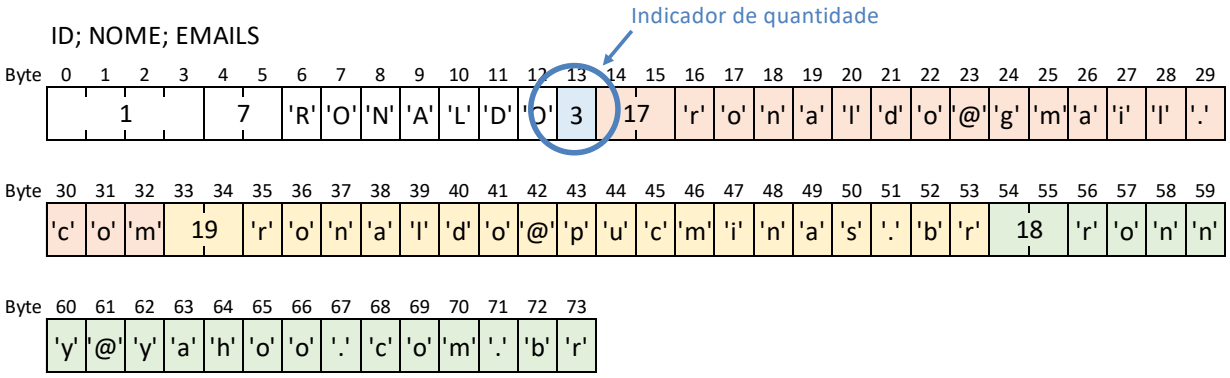
- Com delimitador

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)

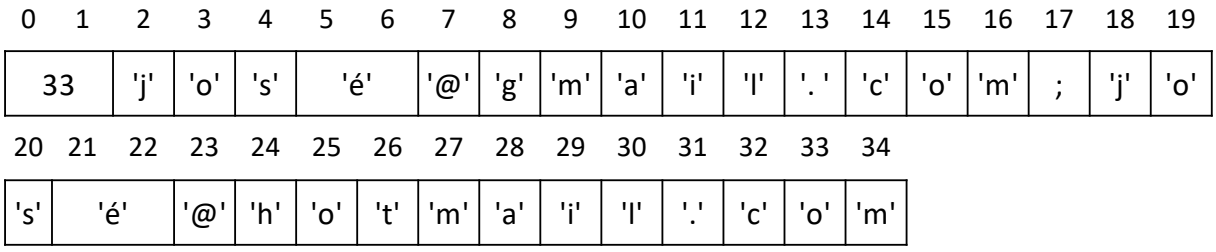
Campos com múltiplos valores

- Exemplo – emails de um cliente:



Campos com múltiplos valores

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



Algumas observações sobre os tipos

- Valores numéricos não devem ser armazenados como strings (ex.: “11”), mas como tipos primitivos: *byte*, *short*, *int*, *long*, *float* e *double*.
- Nos tipos numéricos, não use mais bytes do que o necessário (ex.: byte para número de dependentes, float para preços de livros)
- Veja o padrão usado pela linguagem de programação e pela comunidade de desenvolvedores (ex.: milissegundos para data/hora)
- Cuidado com os falsos números (CPF, CNPJ, CEP, Telefone, ...)