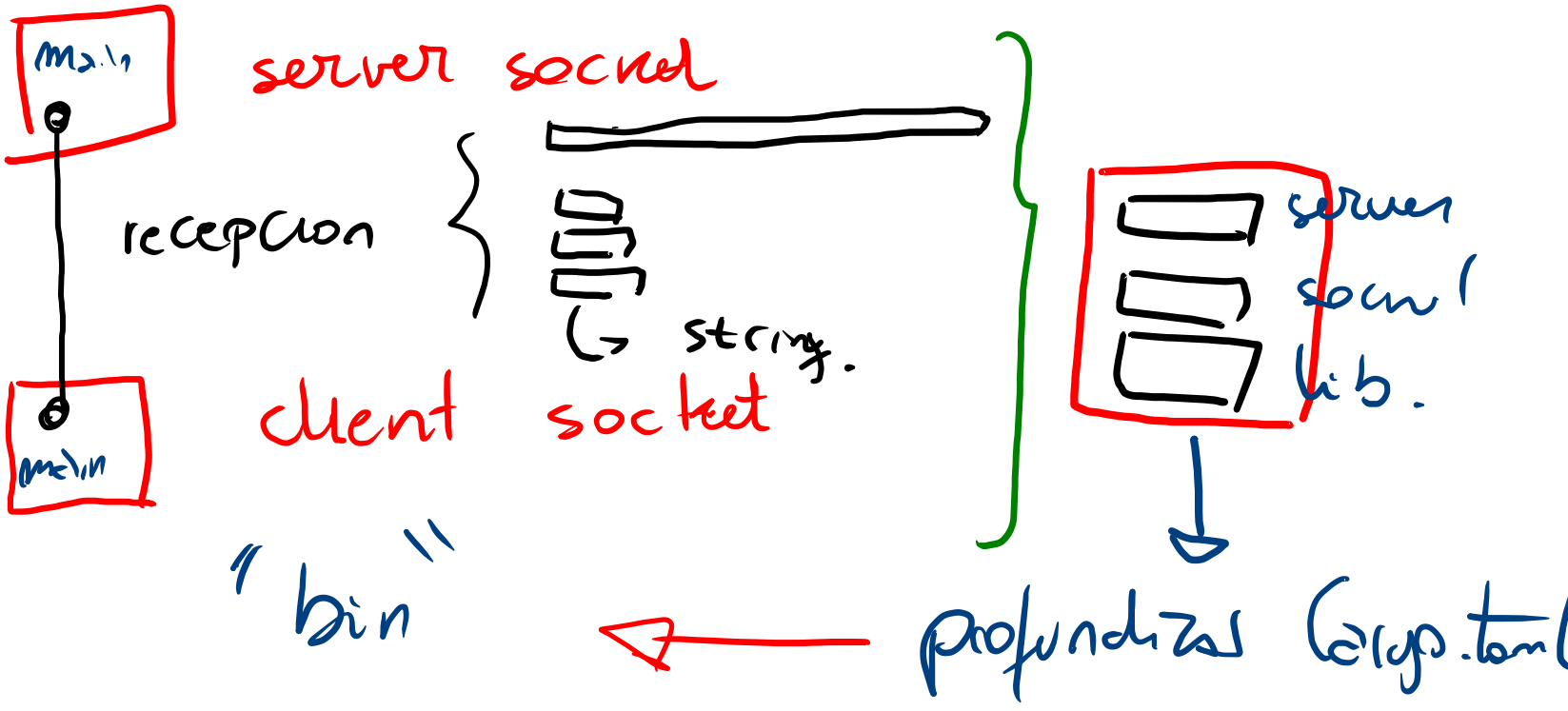
11/03/23

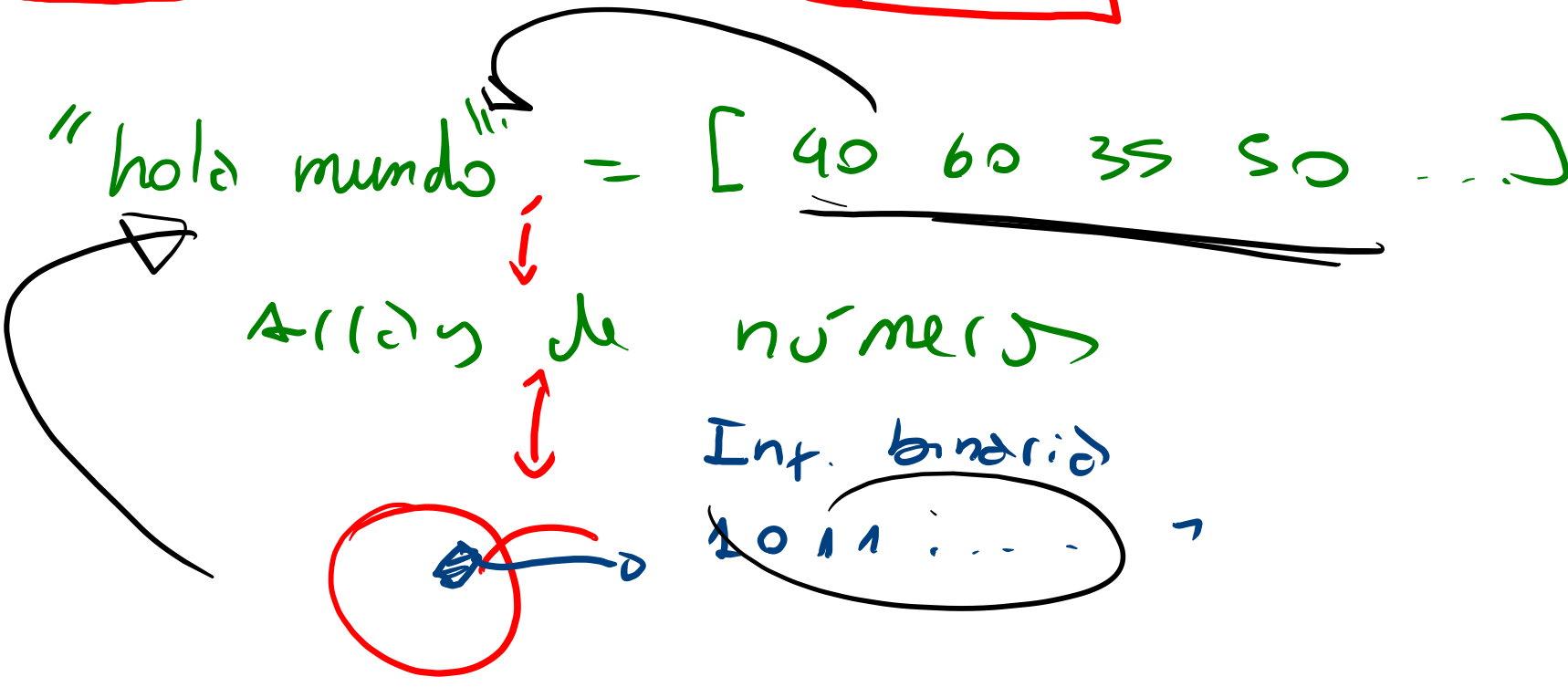
Taller de Rust. 12

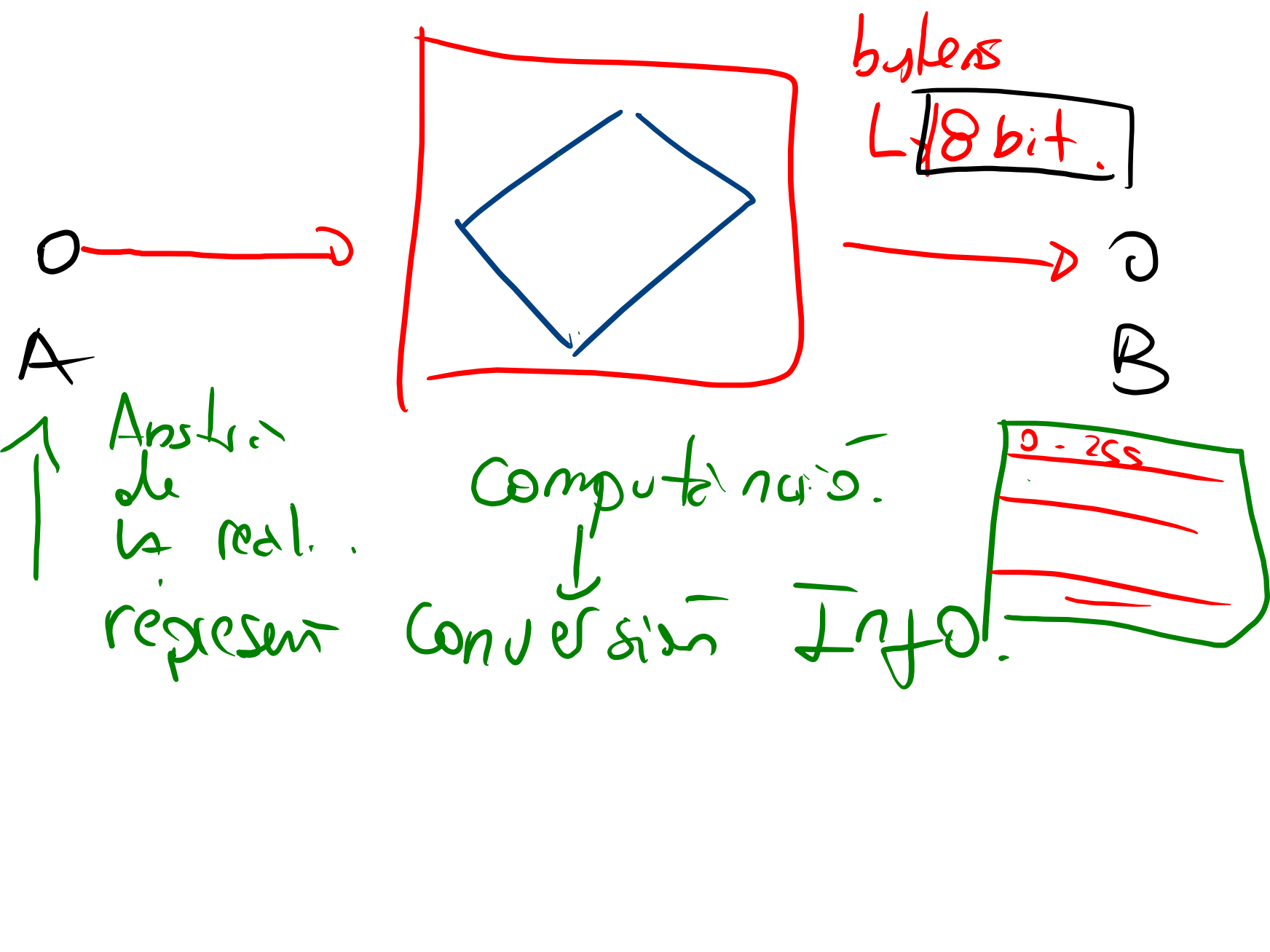
- [ ] conversión bytes → tipos (?)
- [ ] control del 'await' (?)
- [ ] transmisión de mensajes 50%
  + complejos que un 'string'
- [x] reestructuración proyecto.

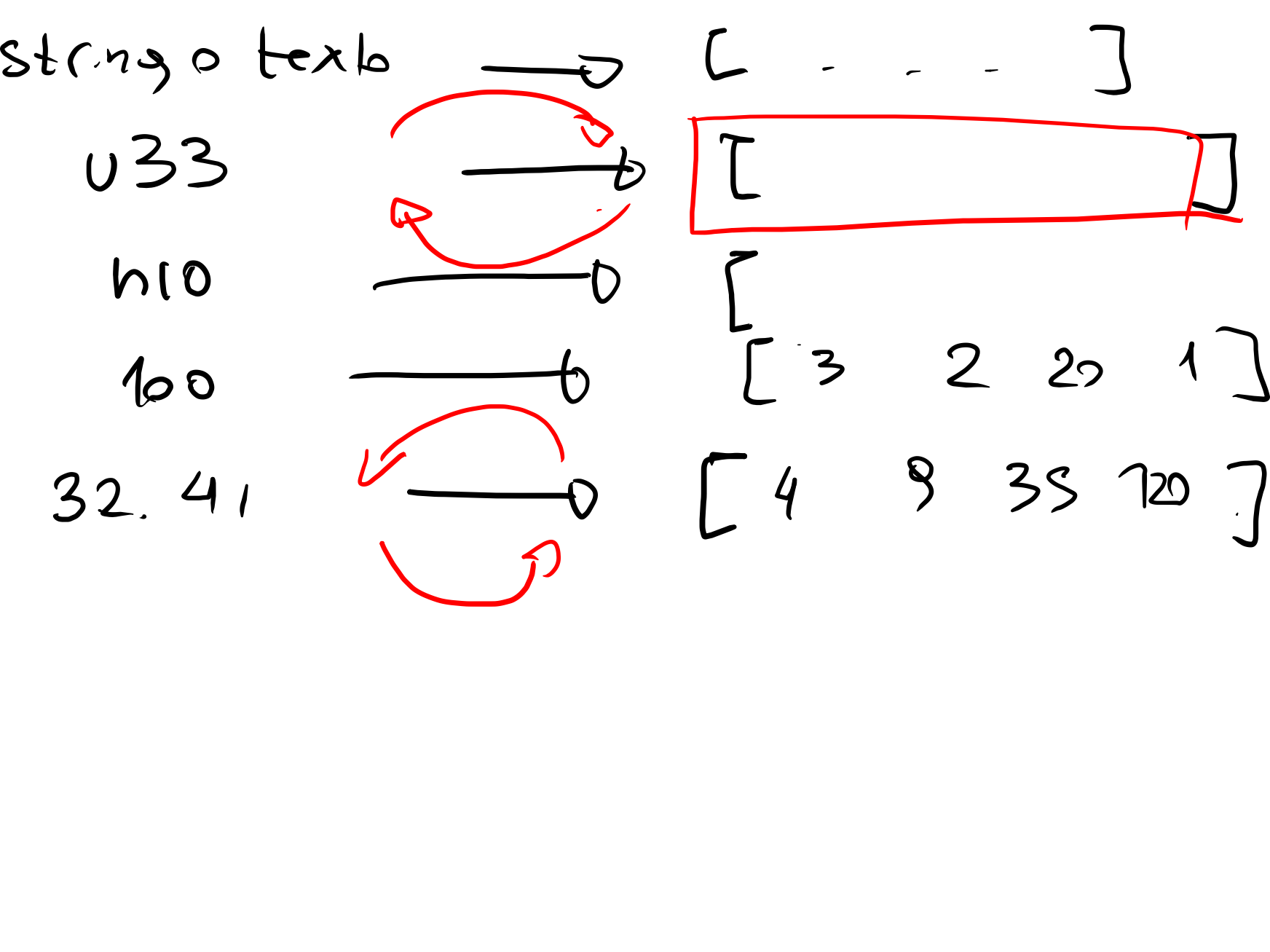
Reestructuración Proyecto



Conversión a tipo.




bytes
L 8 bit.
A
B
Abstracción de la real. representación
Computacional.
Conversión Info.
0 - 255

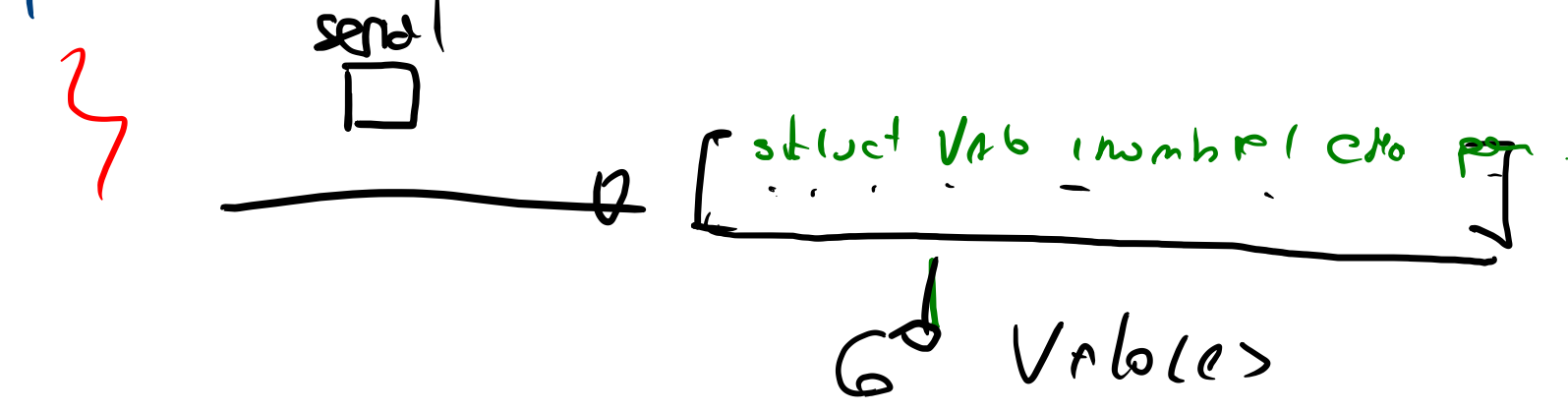

string texto
[- - -]
u33
[
h10
[
100
[3 2 20 1]
32.41
[4 9 35 720]

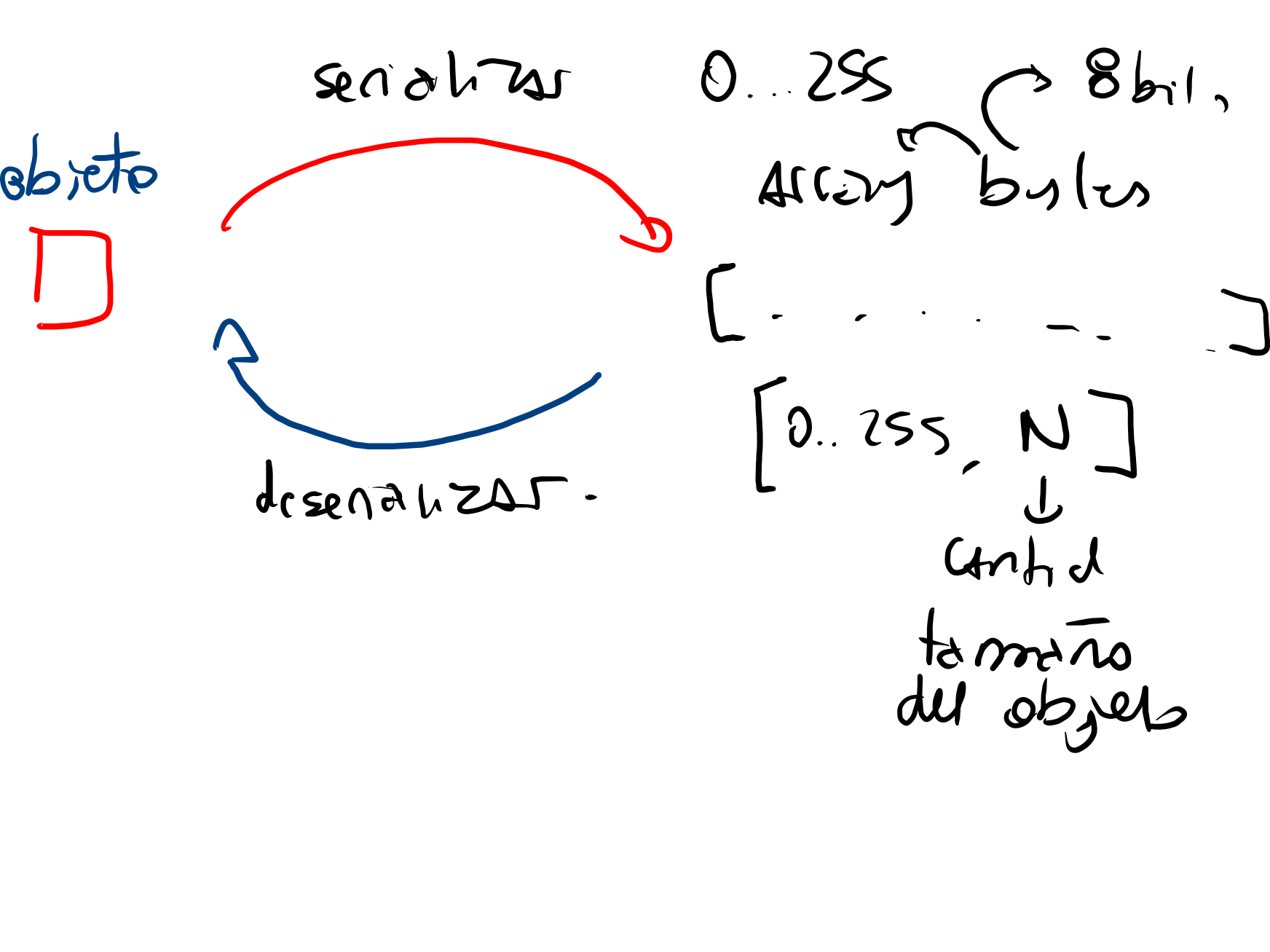
struct Valores }

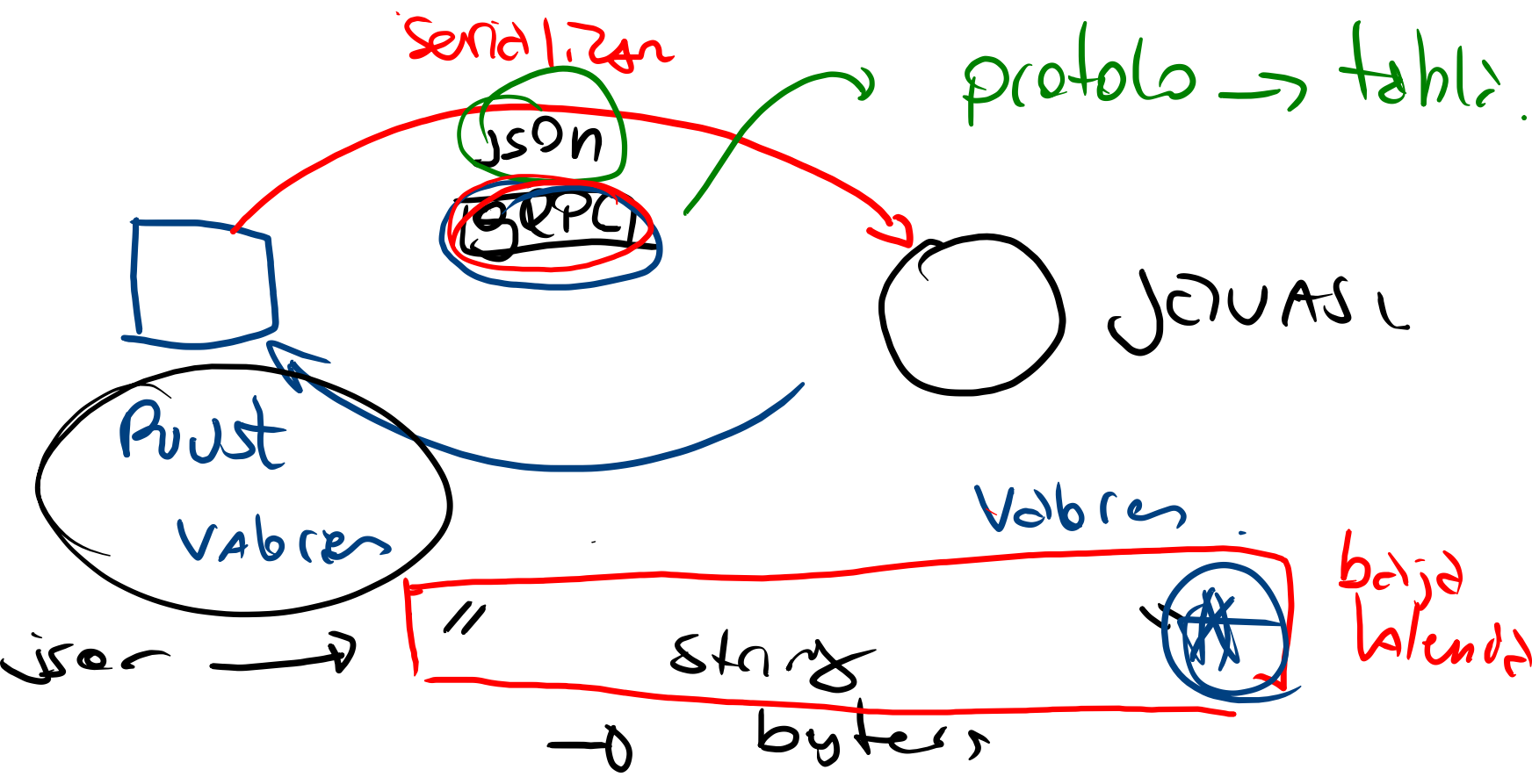
nombre : Str.

edad : u8 (0 . . . 255)
(0 - - - 80)

peso : f64




serializar
0...255
8bit,
objeto
Array bytes
[. . . . .]
[0..255, N]
Cantidad
tamaño
del objeto
desserializar.


Serializar
json
gRPC
protolo → tabla
JAVA
Rust
Valores
Valores
bajo
String
bytes

i 32 → [4b]

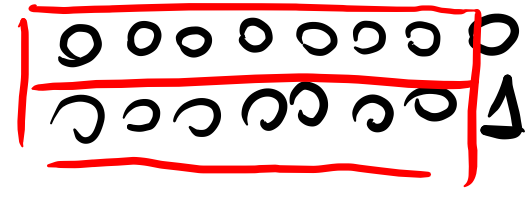
u 32 → [4b]

(con signo)

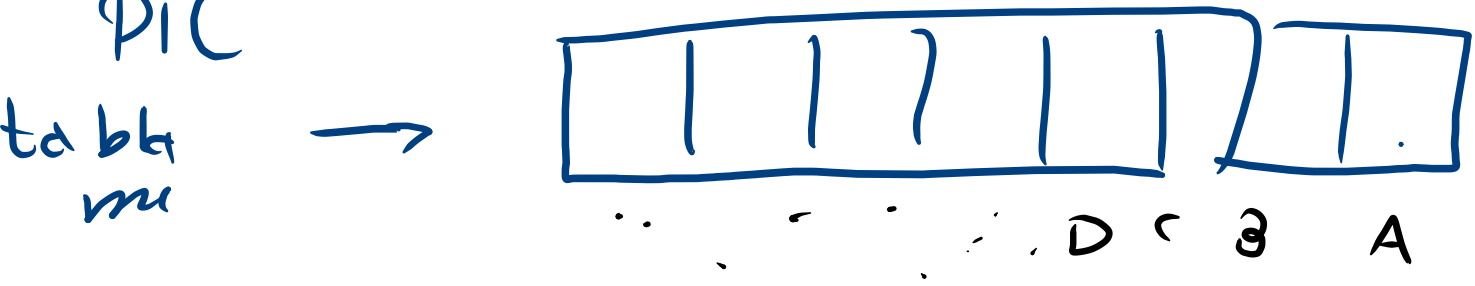
f64 → [8 bytes]

bool → [1 byte (0 , 1)]

1 bit →



PIC



bytes → bit

Protocols

↳ empaq. msg → ⇕

Doc.

Protocolo Documento

tablas

→ campos → tipos.

class
struct

str
int
float
bool.

i32

L0 [4Bytes]

Endianess.

chip (cpu)

234 -0 [3 5 7 10] bigend

[10 7 5 3] litlee

val << n ← n espacios a la izq

val >> n → n espacios a la derecha.

python → repl.

fórmula conver.

-1 entre sym

1 → [0 0 0 1]

-1 → [255 255 255 255]

[256 256 256 256)

1º valor en positivo

[0 0 0 1]

-1 bu [(256-v)]

2 → [0 0 0 2]

NATIVO [2 0 0 0]

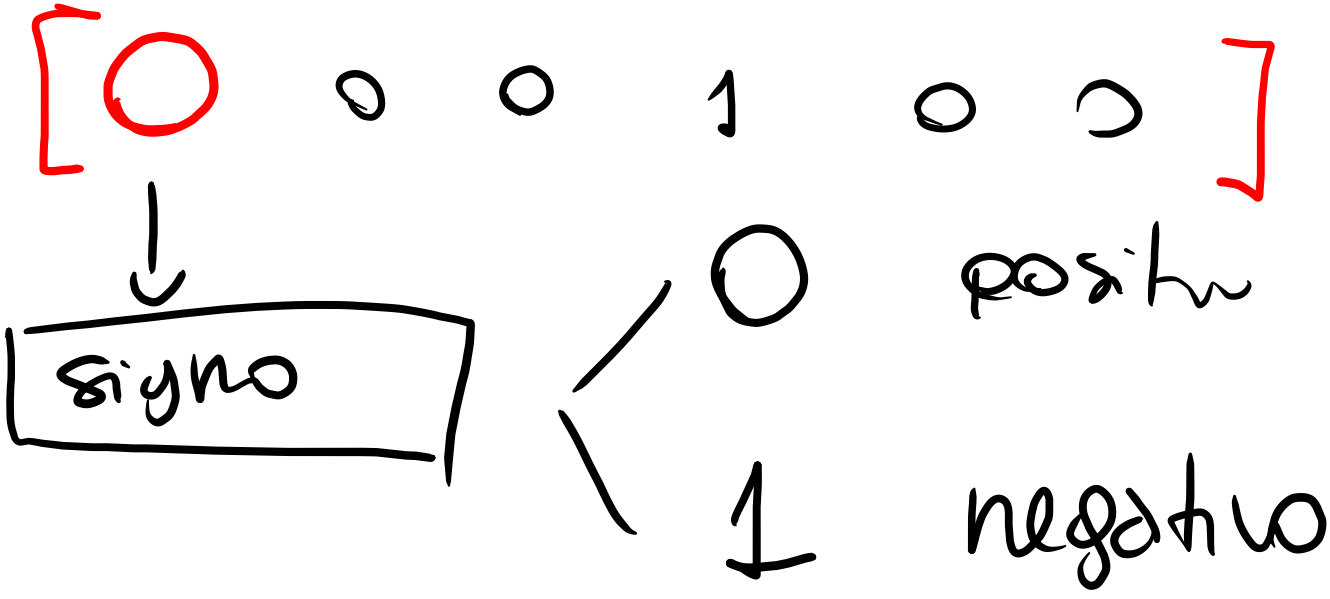
-2 → [255 255 255 254]

[254 255 255 255]

positivo [255 255

bin < 0 → 256 − (bin pos) − 1

Enteros signed



en un array que sea un int<0

[a, b, c, d]

↳ bits → [1 ...] →

7 bits.

[0 7 bits)