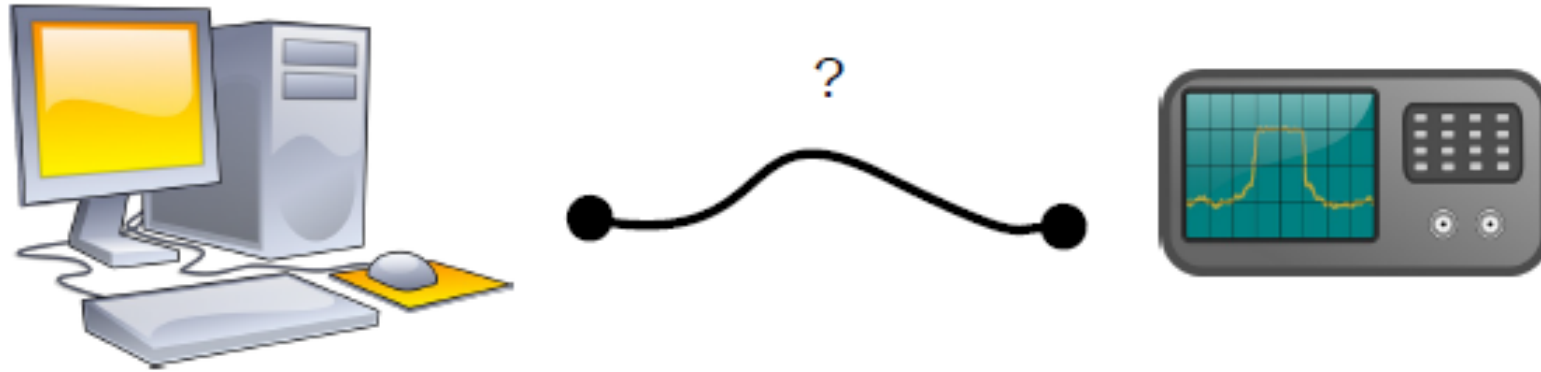


Introducción a la Adquisición de datos

Comunicación con instrumentos



- Qué tipo de interface de hardware tengo? (USB, Puerto Serie o RS232, LAN, GPIB, etc)
- Sobre qué aplicación voy a trabajar? (En la PC: C++, Fortran (?), Python, Matlab, Labview, etc)
- Qué voy a ordenarle o pedirle al instrumento? (algo que el instrumento entienda; por ej SCPI)

Instrumentos de Laboratorio



Generador de funciones



Osciloscopio

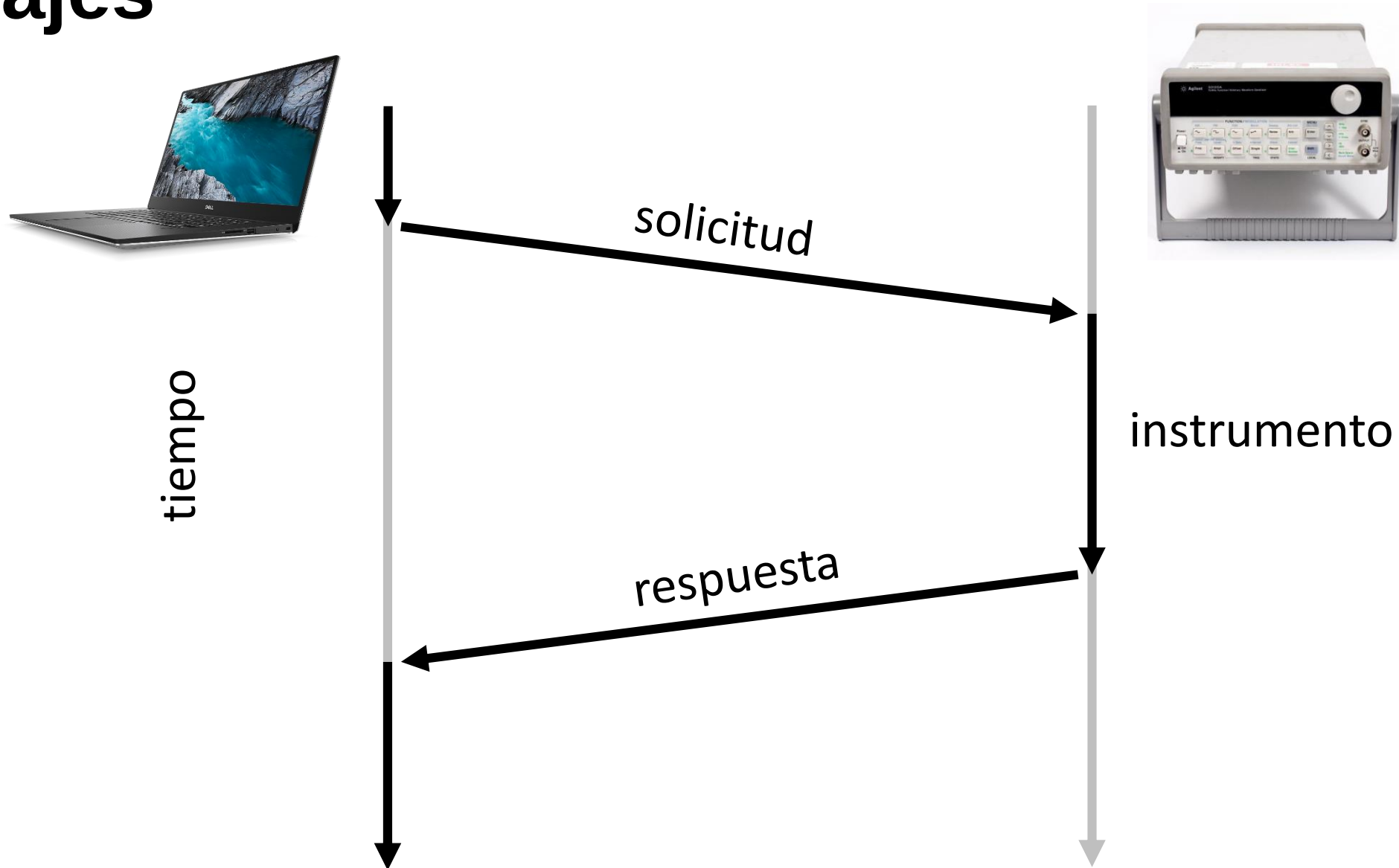


Amplificador



Fuente

Hablando con **instrumentos** intercambiando mensajes



Transmitiendo mensajes



S O S
... - - - ...



Valores digitales

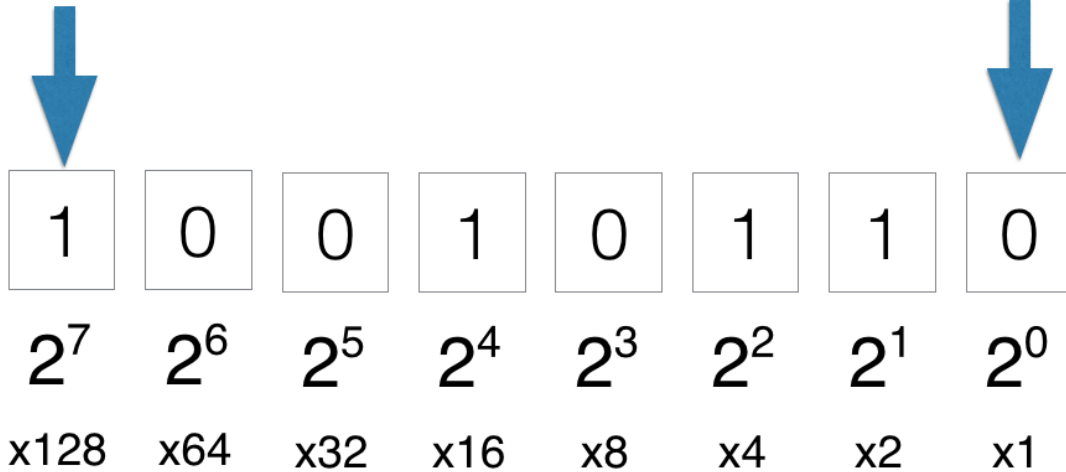
bit

True / 1 / High / On

False / 0 / Low / Off

MSB

LSB



8 Bits = 1 Byte

$$\sum_{i=0}^{N-1} b_i \cdot 2^i$$

Codificando mensajes

ASCII
American Standard Code
for Information Interchange

	ASCII	Decimal	Binary
→	null	0	00000000
	start of header	1	00000001
	start of text	2	00000010
	end of text	3	00000011
	end of transmission	4	00000100
	enquiry		
	acknowledge		
	bell		
	backspace		
	horizontal tab	9	00001001
→	linefeed	10	00001010
	vertical tab	11	00001011
	form feed	12	00001100
→	carriage return	13	00001101

ñ16410100101

ASCII	Decimal	Binary
A	65	01000001
B	66	01000010
C	67	01000011
D	68	01000100
E	69	01000101
F	70	01000110
G	71	01000111
H	72	01001000
I	73	01001001
J	74	01001010
K	75	01001011
L	76	01001100
M	77	01001101
N	78	01001110
O	79	01001111
T	84	01010100
U	85	01010101
V	86	01010110
W	87	01010111
X	88	01011000
Y	89	01011001
Z	90	01011010

Hablando con Instrumentos

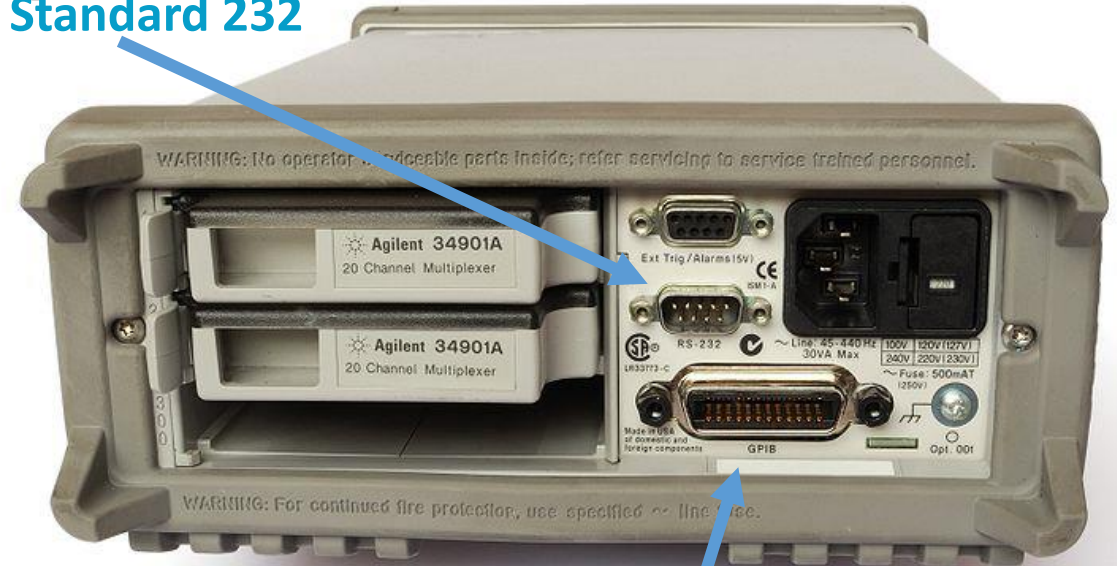
RS-232 (Serie) Recommended Standard 232



Ethernet

USB

Universal Serial Bus



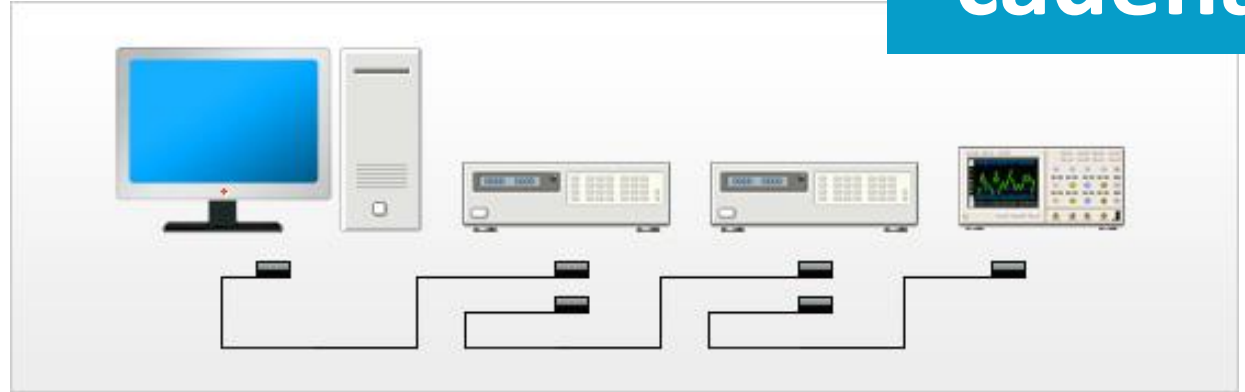
IEEE-488 (GPIB)

General Purpose Interface Bus

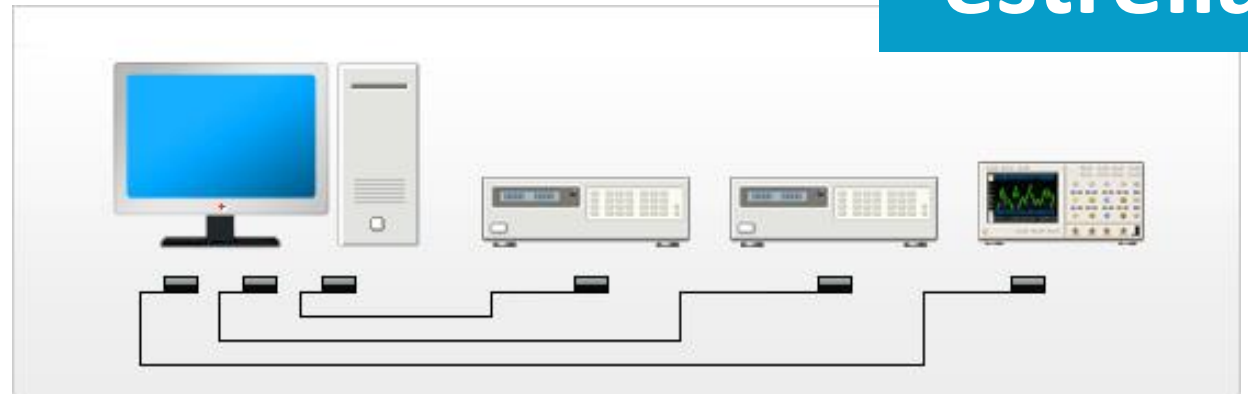
Comunicación vía **GP**IB (General Purpose Instrument Bus)



cadena



estrella



Hasta 15 instrumentos

Ancho de banda y Latencia

Ancho de banda

Capacidad máxima de datos a transmitir (bps)

Latencia

Tiempo que tarda en transmitirse la información entre el origen y el destino



Ancho de banda y Latencia



```
→ ~ ping df.uba.ar
PING df.uba.ar (157.92.11.116): 56 data bytes
64 bytes from 157.92.11.116: icmp_seq=0 ttl=50 time=24.457 ms
64 bytes from 157.92.11.116: icmp_seq=1 ttl=50 time=26.265 ms
64 bytes from 157.92.11.116: icmp_seq=2 ttl=50 time=29.971 ms
64 bytes from 157.92.11.116: icmp_seq=3 ttl=50 time=25.230 ms
64 bytes from 157.92.11.116: icmp_seq=4 ttl=50 time=26.202 ms
```



20 km



Tiempo **mínimo** = $40 \text{ km} / (2/3 c) = \mathbf{0.2 \text{ ms}}$

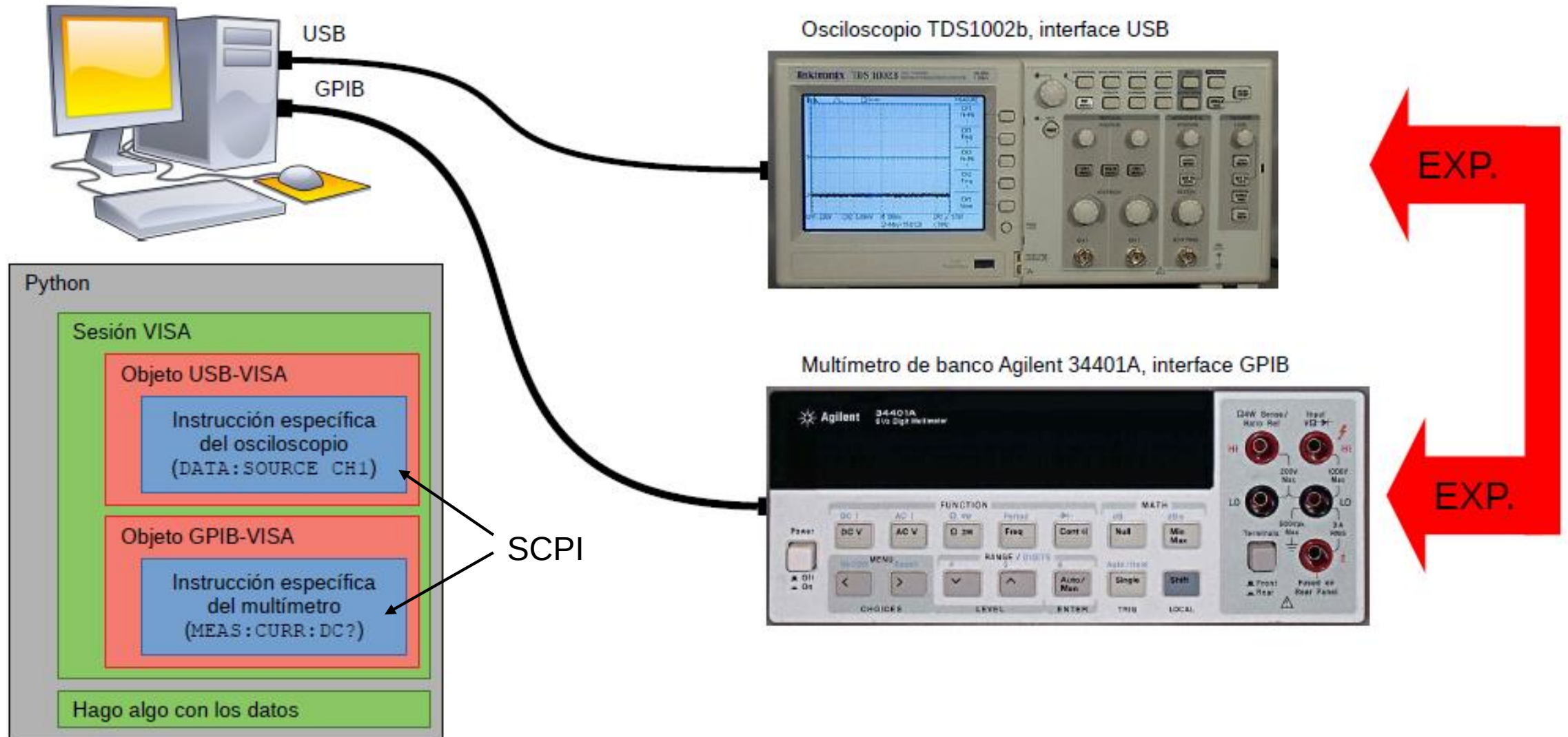
Tiempo **medido**

= **26 ms**

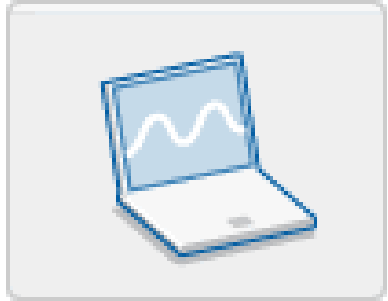
Ancho de banda, Latencia y rango

	Ancho de banda (MB/s)	Latencia (μ s)	Rango (m)
RS-232	20 kbps	100 000	15
GPIB	1.8 (488.1) 8 (HS488)	30	20
USB	60 (Hi-Speed)	1000 (USB) 125 (Hi-Speed)	5
Ethernet LAN/LXI	12.5 (Fast) 125 (Gigabit)	1000	100

Comunicación con instrumentos



Procesador (PC)



Desde el procesador se puede configurar el instrumento y adquirir los datos.
La señal original es reconstruida desde los datos adquiridos por el software
Permite procesar, visualizar y almacenar datos de medida

Componentes del Software

Controlador

permite que el sistema operativo de la PC pueda reconocer el dispositivo y dar así a los programas acceso al instrumento para escritura y lectura.

[VISA](#)

Virtual instrument
software architecture

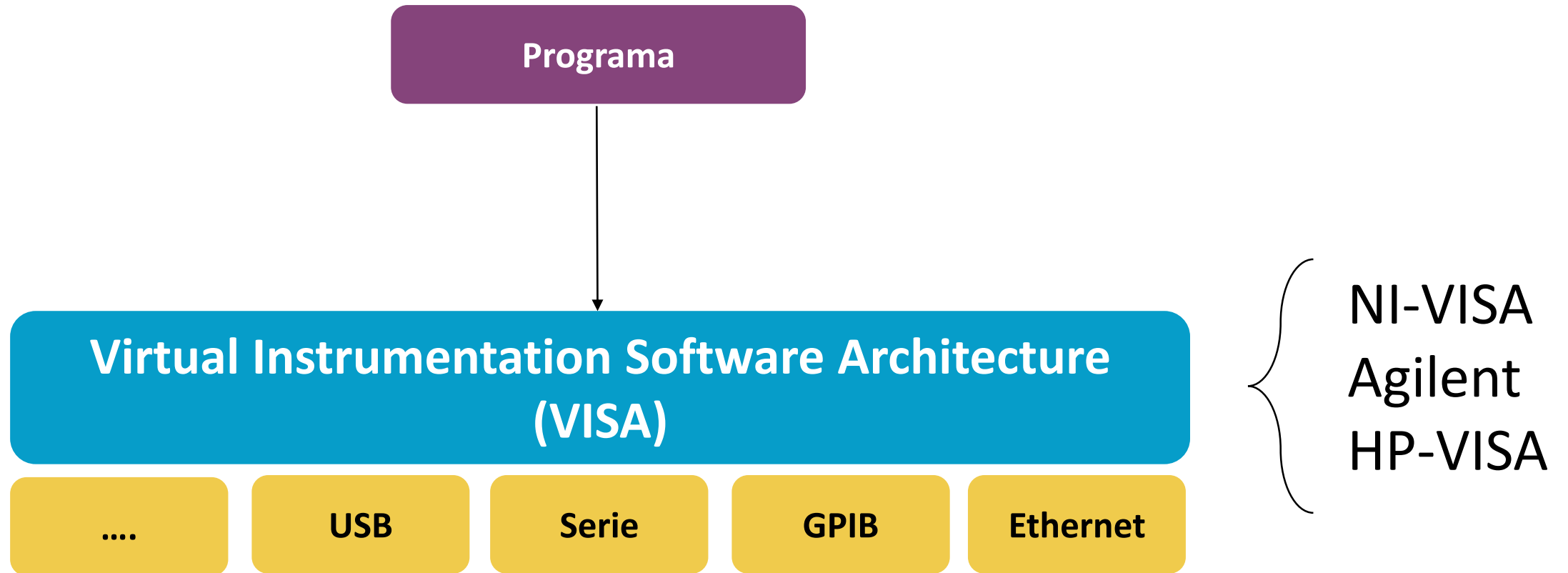
Aplicación

facilita la interacción entre la PC y el usuario
Permite configurar el instrumento, adquirir, analizar y presentar datos de las mediciones
Eje: C++, Fortran (?), Python, Matlab, Labview, etc
En general se SCPI para la sintaxis de los comandos

[SCPI](#)

Standard Commands for
Programming Instruments

Arquitectura VISA



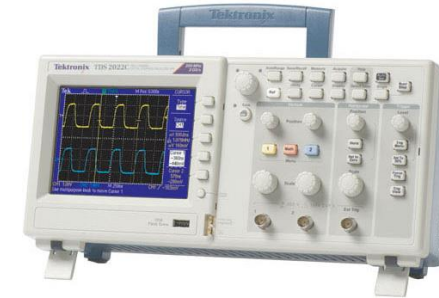
Conceptos VISA

ResourceManager

Session

Session

Resources



Comunicándonos con Python

Usamos el paquete PyVisa
(<https://pyvisa.readthedocs.io/en/latest/>)



Ejemplo de comunicación con un instrumento

```
9 import pyvisa as visa
10
11 rm = visa.ResourceManager()
12
13 instrumentos = rm.list_resources()
14 print(instrumentos)
```

```
In [1]: runfile('C:/Users/User/Google Drive/Laboratorio 4 2dp2021/
Instrumentos/comunicación.py', wdir='C:/Users/User/Google Drive/
Laboratorio 4 2dp2021/Instrumentos')
('USB0::0x0699::0x0363::C065093::INSTR', 'ASRL4::INSTR',
'ASRL5::INSTR')
```

```
In [5]: instrumentos?
Type:      tuple
String form: ('USB0::0x0699::0x0363::C065093::INSTR', 'ASRL4::INSTR',
'ASRL5::INSTR')
Length:    3
```

```
23 #Con ese nombre abro el vínculo con el osciloscopio
24
25 osc=rm.open_resource(instrumentos[0])
26 #osc=rm.open_resource('USB0::0x0699::0x0363::C065093::INSTR')
```

osc es un objeto que representa a la comunicación con el instrumento

¿Qué comandos entiende VISA?

write: le enviamos un mensaje al instrumento

```
In [13]: osc.write('MEASU:MEAS3:SOURCE CH2')
Out[13]: 24

In [14]: osc.write('MEASU:MEAS3:TYPE CRMs')
Out[14]: 23
```

read: si el instrumento mandó un mensaje, lo leemos
(si no mandó nada, obtendremos un timeout)

```
In [16]: osc.read
Out[16]: <bound method MessageBasedResource.read of
<'USBInstrument'('USB0::0x0699::0x0363::C065093::0::INSTR')>>
```

query: enviar mensaje y leer respuesta luego

```
In [17]: osc.query('*IDN?')
Out[17]: 'TEKTRONIX,TDS 1002B,C065093,CF:91.1CT FV:v22.11\n'
```

¿Qué comandos entiende VISA?

LEER EL MANUAL DEL INSTRUMENTO!!!!

MEASUrement:IMMed:SOUrce[1] Set or query the channel for immediate measurement

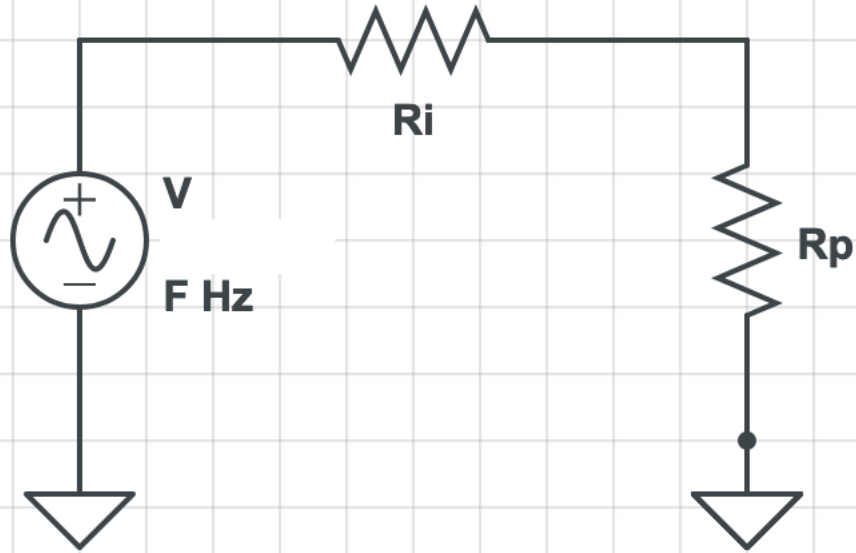
Ej.: osc.write('MEAS:INM:SOU CH1')

MEASUrement:IMMed:TYPe Set or query the immediate measurement to be taken

MEASUrement:IMMed:UNIts? Return the immediate measurement units

MEASUrement:IMMed:VALue? Return the immediate measurement result

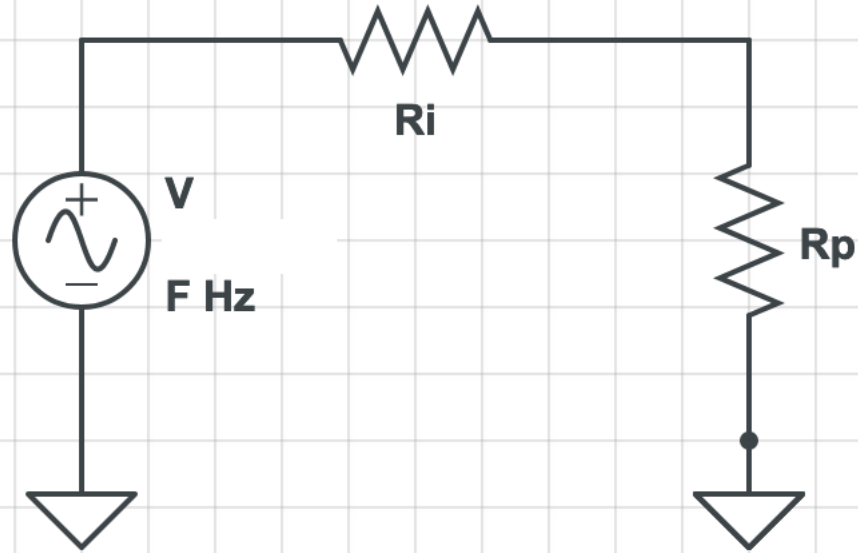
Qué estudiamos hoy?



Experimento 1

1. Armar el circuito
2. Usar una rampa
3. Medir
4. Cambiar la frecuencia
5. Volver al punto 3

Qué estudiamos hoy?



Experimento 2

1. Usar una senoidal
2. Estudiar Ley de Ohm en R_p

¿Cómo lo hacemos?