

UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

---

## Medidas Electrónicas II

Proyecto:

“Control Remoto de Instrumentos Virtuales”

2019

D'Angelo Pablo

Koremblum Nicolás Mariano

Ortiz Agustín



# Objetivo

*Poder controlar un set de medición, compuesto por tres instrumentos, de forma virtual y remota.*

## Introducción

En la materia “Medidas Electrónicas II” es común requerir de múltiples instrumentos de medición simultáneamente para poder realizar una cierta medición, esto puede resultar a veces poco práctico debido a que se debe estar cambiando la operación manual de un instrumento a otro, pudiendo producir errores y pérdida de tiempo. También es bien sabido que una vez calibrados los instrumentos, los mismos, deberían permanecer físicamente en el estado en que fueron calibrados, por lo que trasladarlos no es una opción. Por estos motivos es de gran utilidad contar con la posibilidad de controlar a los instrumentos de manera virtual.

Para controlar los instrumentos de forma remota existe un protocolo, ampliamente difundido, llamado VISA por sus siglas: “Virtual instrument software architecture” [1]. Este protocolo busca unificar el control de los instrumentos para facilitarle la labor a los múltiples usuarios y pudiendo reutilizar ciertos scripts en distintos instrumentos, independientemente del fabricante, o bien realizando la mínima cantidad de modificaciones posibles.

Dados los requisitos del proyecto y valiéndonos de las facilidades que brinda el protocolo VISA, desarrollamos un sistema que permite a múltiples clientes acceder simultáneamente a la información recaudada por un servidor. Dicha información es provista a través de una página web que el servidor administra. Para esclarecer esta información, se esquematiza el siguiente el sistema en la Figura 1.

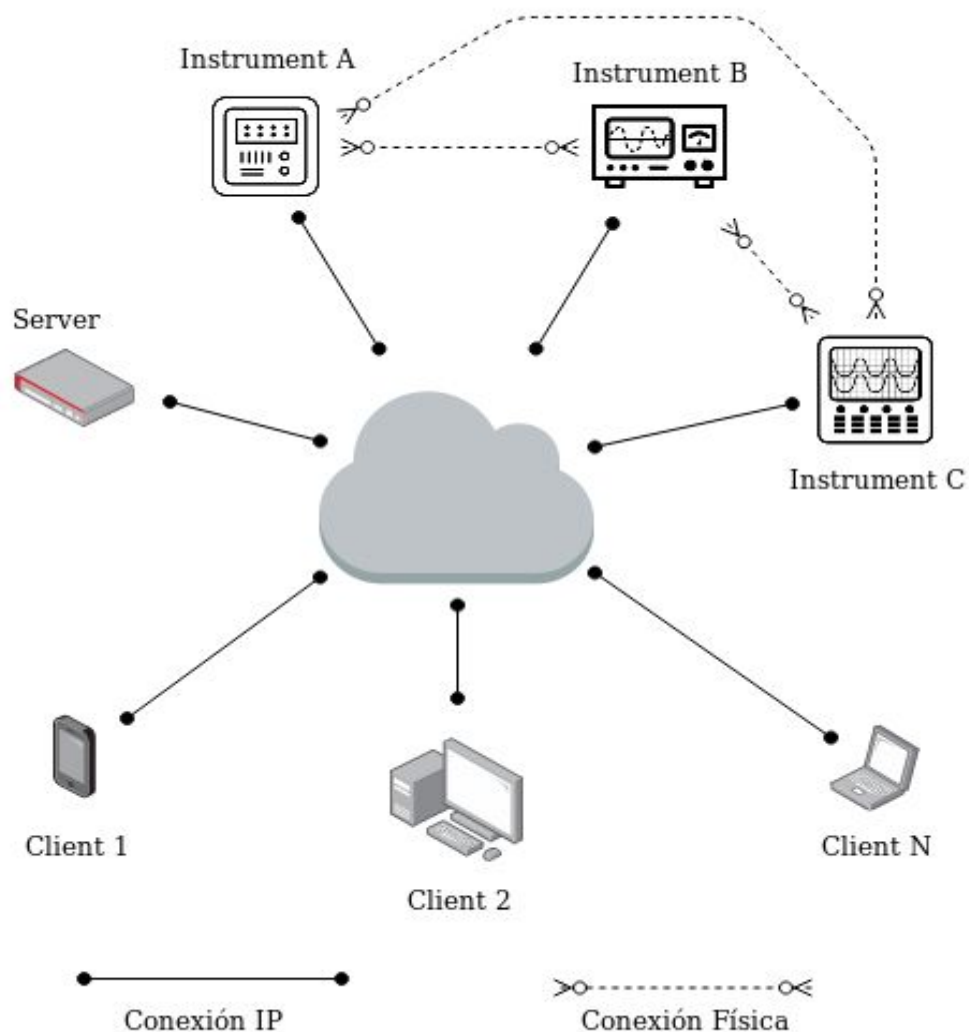


Figura 1: Esquema de interconexión del sistema

Se puede observar que, independientemente de donde se encuentren el servidor, los instrumentos y los clientes, todos los actores se pueden vincular a través de internet (o cualquier esquema de red, LAN, WAN, etc.). Para lograr esto, el servidor debe conocer las IP's de los instrumentos, y los clientes la IP (o bien la "dirección web") del servidor. A su vez todos los actores deben tener acceso a dichas IPs y en el caso del servidor y los instrumentos deben tener los permisos de seguridad de red para que sus recursos puedan ser accedidos desde el exterior.

# Materiales y Métodos

## Hardware

El Hardware se compone principalmente por el **Set de Medición**, que cuenta con los **Instrumentos** y demás componentes anexos para la medición (como conectores, cables, etc.), y el **Servidor** en sí, aunque no necesariamente debe encontrarse on-premises, sino que puede estar hospedado en “la nube”.

En este proyecto, el **Set de Medición**, consiste en un **Generador de Señales RF** (Rigol DSG815) [2], un **Analizador de Espectro** (Rigol DSA815) [3] y un **Powermeter** (Anritsu ML2487B) [4], interconectados entre sí y, a su vez, todos y cada uno de ellos conectados por IP con el servidor, para el desarrollo, utilizando una red LAN. El **Set de Medición** del proyecto se esquematiza en la figura 2.

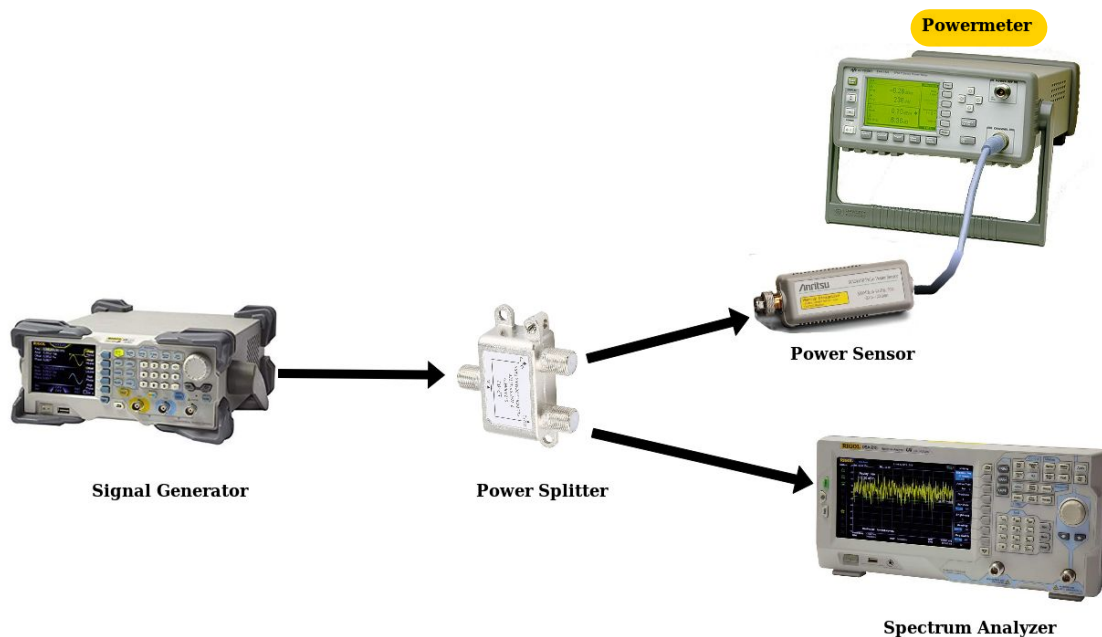


Figura 2: Set de Medición.

## Software

Para entender la división de funcionalidades del software, se debe comprender primero la diferencia entre un **Servidor** y un **Cliente**.

En el caso de este proyecto, el **Servidor**, se encarga de administrar la información. Por un lado, se comunica con los instrumentos para obtener información de ellos o bien para configurarles ciertos parámetros de medición. Por otro lado, el **Servidor**, a través de la interfaz web que provee, envía a los **Clientes** la información recaudada, que será la misma para todos ellos, y recibe sus solicitudes de configuración. Cabe destacar que el **Servidor**, debido a su complejidad y su necesaria segmentación, cuenta con múltiples módulos que se ejecutan en paralelo para simplificar el diseño y testing.

El **Cliente** solo tendrá acceso a la página web que provee el **Servidor**, en una primera instancia, al conectarse a dicha página, el **Cliente** recibe los **HTML's** procesados y los scripts que permiten interactuar con algunos elementos gráficos, los cuales se manifestarán como una página web interactiva. Los códigos en principio son anecdóticos y transparentes para el usuario, ya que para éste sólo importará la interfaz web. A través de dicha interfaz, el usuario podrá observar todos los parámetros de configuración actual de los instrumentos, como así también los valores correspondientes a las mediciones. Por otro lado, cuenta con un panel de configuración para realizar los seteos que considere pertinentes para la correcta medición.

Es importante conocer dicha división para entender los distintos módulos de código que intervienen en el sistema. Del lado del **Servidor**, se utilizan scripts de **Python** que son ejecutados directamente por un intérprete del sistema host (servidor). Dentro de todas las librerías con las que cuenta **Python** se utilizaron, principalmente, tres librerías: **Pyvisa** [5], para la intercomunicación con los instrumentos, **Bokeh** [6], para construir los elementos gráficos (que verá el cliente desde el portal) y **Flask** [7] que permite implementar el servidor web. A su vez el

**Servidor** almacena y procesa las plantillas **HTML**, que se nutren de “estilos” de los archivos **CSS**, y los scripts de **Javascript** que serán consumidos por los **Clientes**. Cuando un **Ciente** se conecta, a través de un navegador web, a la dirección, y puerto, del **Servidor**, el mismo recibe dichas plantillas y scripts, que serán interpretados y ejecutados por el navegador web. El hecho de realizar ciertas configuraciones localmente (utilizando scripts web), permite que la configuración impacte solamente en el **Ciente** interesado sin afectar al resto, además de permitir algunas funcionalidades web que el **HTML** puro no permite.

El **Portal Web** consta de tres secciones principales:

- *Display del Analizador de Espectro*

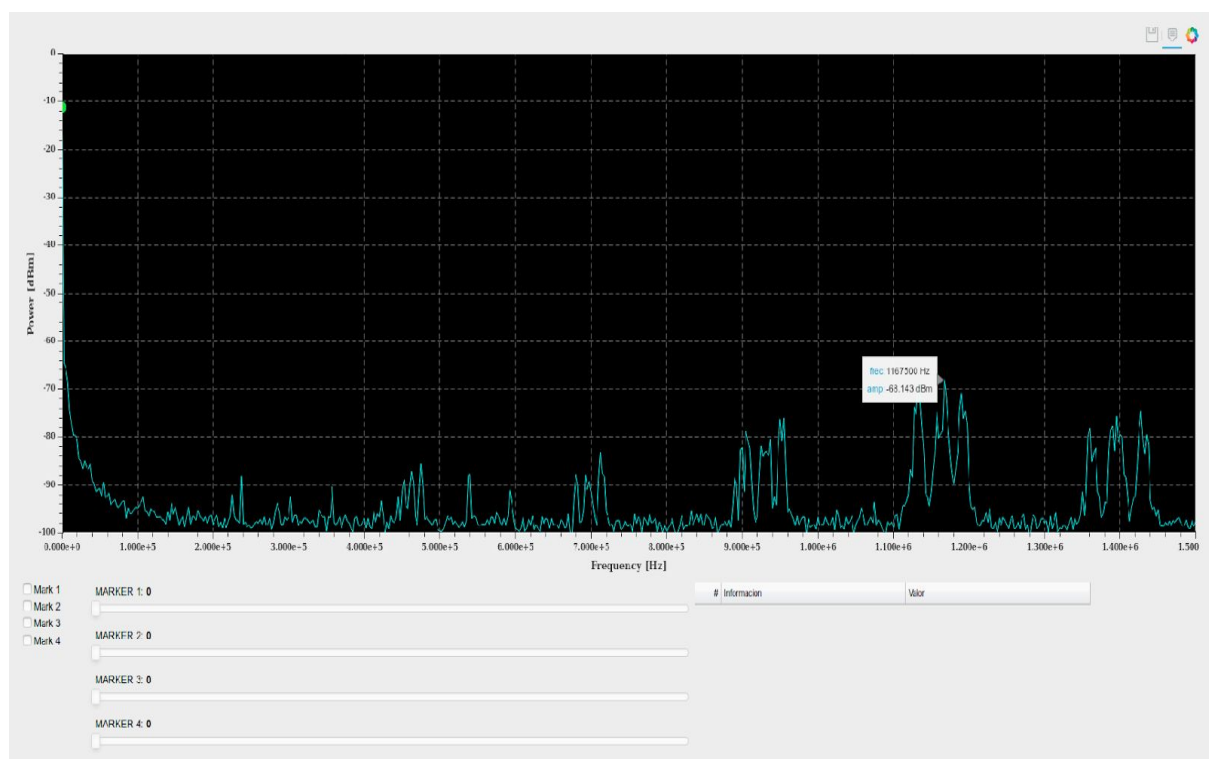


Figura 3: Sección del Display del Analizador de Espectro

En esta sección se puede visualizar la señal del Analizador de Espectro tal cual se vería en el display de dicho instrumento. También cuenta con la posibilidad de utilizar markers para seleccionar distintos puntos de la señal visualizada y algunos valores relacionados con estos markers son

mostrados en una tabla situada debajo del gráfico a la derecha. A su vez, si se mueve el cursor sobre el gráfico, el mismo mostrará información relacionada con la ubicación del puntero. Esto se visualiza en la Figura 3.

### - *Tablas de Configuración Actual*

#	Analyzer	value	#	Generator	value	#	Power meter	value
0	X scale	LIN	0	LF state	OFF	0	Duty cycle	50
1	Y scale	LOG	1	LF freq	1.000 00kHz	1	Average	1
2	Start freq	0	2	LF level	0.50			
3	Stop freq	1500000000	3	LF waveform	SINE			
4	Span	1E6	4	RF state	OFF			
5	Center freq	1E9	5	RF freq	1500000000.00			

Figura 4: Sección de Tablas de Configuración Actual

Estas tablas permiten al usuario conocer la configuración actual de cada dispositivo de medición, indicando instrumento por instrumento y parámetro por parámetro el estado actual del mismo. A modo de ejemplo se muestran las mismas en la Figura 4.

### - *Formularios de Configuración*

Analyzer Settings
Generator Settings
Powermeter Settings

X-Scale

LIN
LOG

Start Frequency [Hz]
Reference Level
Resolution BW [Hz]
Averaging [Times] (1)
Send an arbitrary VISA command:

SET CONFIG

Y-Scale

LIN
LOG

Stop Frequency [Hz]
Scale/Division
Demodulation Time [s]

Range Mode

FREQ
SPAN
FULL

Input Attenuator (10)
Demodulation Type

OFF
AM
FM

Filter Mode

OFF
VIDEO
POWER

Figura 5: Sección de Formularios de Configuración.

Esta sección cuenta con los campos necesarios para configurar los instrumentos a gusto del usuario. Dichas configuraciones son limitadas para facilitar la experiencia del usuario, aunque se da la opción de enviar comandos “crudos” de VISA para extender la funcionalidad. Los seteos se dividen en solapas según el instrumento que se desea configurar. Ver Figura 5.

## Guía para el Desarrollador o Administrador del Sistema

El repositorio de código [8] se encuentra naturalmente dividido según el lenguaje de programación/descripción.

En la raíz del repositorio se encuentran los scripts de **Python (3)** que corre el servidor. Si se desea ejecutar dichos scripts manualmente para hacer funcionar el sistema, primero se deberán setear las respectivas IPs en los archivos “.conf” según corresponda, a su vez es posible setear el puerto de salida del servidor en el archivo “server\_port.conf”. Luego se deberán ejecutar los siguientes scripts: “analyzer\_com.py”, “generator\_com.py”, “powermeter\_com.py” y “web.py”. En los archivos terminados en “\_com.py” se encuentra la lógica que intermedia entre el proceso principal del servidor web (“web.py”) y el instrumento correspondiente. El resto de los “.py” son complementos que no necesitan ser ejecutados manualmente.

Uno de los subdirectorios del repositorio es “templates” en este se encuentran dos archivos **HTML** “impuros” que serán procesados por el servidor al momento de transferirlos a los clientes. Esto se debe principalmente a que la librería gráfica **Bokeh** requiere que así se haga.

Otro subdirectorio es “static”, en este se encuentran los códigos “estáticos”, osea que no requieren ningún tipo de procesamiento por



parte del servidor y son enviados a los Clientes tal cual se encuentran. Dentro de este subdirectorio se separan las carpetas de **CSS**, **HTML** y **Javascript** conteniendo respectivamente los archivos que recibirá el navegador web del cliente para poder presentar la interfaz web tal cual se desea.

## Conclusiones

Con la experimentación surgieron algunas limitaciones y problemas que pueden interferir con el funcionamiento esperado del sistema. Estos son:

- **Delay:** Los instrumentos se toman su tiempo en responder utilizando el protocolo visa, siendo este tiempo variable y aleatorio. Esto se nota principalmente en el gráfico del display del analizador de espectro que demora un cierto tiempo en refrescar, notablemente superior al tiempo de refresco de la propia pantalla del instrumento, y también en el tiempo de seteo de las configuraciones, que uno esperaría sean inmediatas.
- **Conectividad:** La conectividad, tanto Cliente-Red como Server-Red e Instrumentos-Red, debe ser lo suficientemente buena para que los datos (que son de un volumen considerable) se transmitan de forma fluida y sin producir desconexiones. Por obvias razones, el nodo que mayor volumen de tráfico maneja es el Servidor, ya que obtiene los datos del Analizador de Espectro (el mayor volumen) y los distribuye a todos los Clientes que se encuentren conectados.
- **Seguridad de Red:** Las políticas de seguridad de las redes donde se encuentren el servidor y/o los instrumentos pueden limitar el funcionamiento de los mismos.

A pesar de estos problemas/limitaciones, el sistema demuestra ser altamente funcional y práctico, permitiendo mejorar la experiencia multiusuario frente a un mismo set de medición. Sin duda la funcionalidad del sistema irá mejorando con el uso del mismo ya que el feedback generado por los usuarios permitirán a los desarrolladores solucionar los múltiples problemas que aún no hayan sido detectados. A su vez con la mejora de las tecnologías los delays se reducirán permitiendo un comportamiento más fluido.

## Bibliografía

1. <http://www.ivifoundation.org/specifications/default.aspx>, "VISA Specification". IVI Foundation.
2. <https://www.rigolna.com/products/rf-signal-generators/dsg800/>, Rigol DSG815 (RF Signals Generator).
3. <https://www.rigolna.com/products/spectrum-analyzers/dsa800/>, Rigol DSA815 (Spectrum Analyzer).
4. <https://www.anritsu.com/en-us/test-measurement/products/ml2487b>, Anritsu MLB2487B (Power Meter).
5. <https://pyvisa.readthedocs.io/en/latest/>, Python's Library "PyVisa".
6. <https://docs.bokeh.org/en/latest/index.html>, Python's Library "Bokeh".
7. <https://flask.palletsprojects.com/en/1.1.x/>, Python's Library "Flask".
8. <https://github.com/ingenieriaam/me2-control-virtual>, Repositorio de Código del Proyecto.