

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de
Telecomunicación

Plataforma para la caracterización experimental de
dispositivos de RF

Autor: Ignacio Pérez Lupiáñez

Tutor: María José Madero Ayora

Dep. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2016



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Plataforma para la caracterización experimental de dispositivos de RF

Autor:

Ignacio Pérez Lupiáñez

Tutor:

María José Madero Ayora

Profesor titular

Dep. de Teoría de la Señal y Comunicaciones

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2016

Trabajo fin de grado: Plataforma para la caracterización experimental de dispositivos de RF

Autor: Ignacio Pérez Lupiáñez

Tutor: María José Madero Ayora

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2016

El Secretario del Tribunal

A mi familia

A mis maestros

Agradecimientos

En primer lugar, agradecer a mi tutora y profesora María José Madero Ayora por el excelente trato que ha tenido conmigo durante el transcurso de este proyecto, por el interés que ha mostrado en cada momento y por la ayuda recibida tanto en tutoría como en el laboratorio a la hora de comprobar resultados.

Al Departamento de Teoría de la Señal por permitirme el acceso al laboratorio de radiocomunicación para el uso de la instrumentación que he necesitado en la realización de las medidas.

A mis amigos de toda la vida y a los que he conocido durante la carrera, en especial a Pepe y María por animarme y compartir estos últimos años juntos.

A Elena, por todo el apoyo y ayuda que me ha dado durante todos estos años y, cómo no, en el tiempo que hemos dedicado juntos a hacer nuestros proyectos. Contigo he aprendido a creer en uno mismo y a esforzarme a pesar de las adversidades.

A mis padres y a mi hermana por haber confiado siempre en mí y transmitirme toda la fuerza y compañía que me han permitido llegar al final de esta carrera. Sin vuestra ayuda no hubiera sido posible.

Ignacio Pérez Lupiáñez

Sevilla, 2016

Resumen

Los sistemas de radiocomunicación permiten el intercambio de información utilizando el espectro radioeléctrico como medio de transmisión. Debido a la expansión de este tipo de tecnología, sobre todo en comunicaciones móviles, resulta de interés realizar medidas, usando instrumentación de laboratorio, con el objeto de observar el comportamiento que tienen los dispositivos de radiofrecuencia. La obtención de algunos datos necesarios para caracterizar este tipo de dispositivos requiere la realización de miles de medidas, pudiéndose tardar horas e incluso días en completarse manualmente.

Mediante la herramienta de programación gráfica LabVIEW, es posible diseñar programas que automaticen una medida en concreto de forma remota. El desarrollo de estas aplicaciones permite controlar y adquirir datos de los equipos fácilmente en cuestión de minutos, de este modo, se reduce la tarea de configuración de los equipos de forma manual y el tiempo que ello conlleva.

En este proyecto se diseñarán dos programas principales, uno para caracterizar efectos no lineales de un amplificador de RF y otro capaz de realizar una modulación ARB y estudiar su salida.

Abstract

Radio systems allow the exchange of information using the radio spectrum as a transmission method. Due to the expansion of this technology, especially in mobile communications, it is interesting to take measurements using laboratory instrumentation in order to observe the behavior of the RF devices. Characterizing these devices could require thousands of measures that can take hours or even days to complete manually.

By using the LabVIEW graphical programming tool, however, it is possible to design programs that automate a specific measure remotely. Developing these applications allows controlling equipment and acquiring data from it easily in few minutes.

This project will include two main applications, one designed to characterize nonlinear effects of an RF amplifier and another capable of performing an arbitrary modulation in a signal generator.

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xv
Índice de Tablas	xvii
Índice de Figuras	xix
Glosario	xxiii
1 Introducción	1
1.1 <i>Introducción al tema</i>	1
1.2 <i>Objetivo y alcance</i>	1
1.3 <i>Estructura de la memoria</i>	1
2 Material y método	3
2.1 <i>Generador de señal Rohde & Schwarz SMIQ02B</i>	3
2.1.1 <i>Modulaciones</i>	4
2.1.2 <i>Generador de forma de onda arbitraria</i>	6
2.1.3 <i>Control remoto</i>	7
2.2 <i>Analizador de señal Agilent N9010A EXA</i>	7
2.2.1 <i>Modos de funcionamiento</i>	8
2.3 <i>Bus GPIB (IEEE 488)</i>	9
2.4 <i>Comandos SCPI</i>	9
3 LabVIEW	11
3.1 <i>Descripción del programa y sus posibilidades</i>	11
3.2 <i>Entorno de edición de LabVIEW</i>	11
3.2.1 <i>Panel frontal y diagrama de bloques</i>	11
3.2.2 <i>Barra de herramientas</i>	13
3.2.3 <i>Herramientas comunes</i>	14
3.2.4 <i>Controles e Indicadores</i>	15
3.2.5 <i>Funciones</i>	17
3.3 <i>Creación de un VI como ejemplo</i>	20
4 Control de instrumentos	27
4.1 <i>Controladores Plug & Play para los equipos</i>	29
4.1.1 <i>Controlador del Analizador Agilent EXA n9010a</i>	31

4.1.2	Controlador del Generador Rohde & Schwarz SMIQ02b	32
4.2	<i>Configuración para la interfaz USB/GPIB Agilent 82357B</i>	33
4.2.1	Keysight IO Libraries Suite	33
4.3	<i>Ejemplo práctico del uso de controladores</i>	35
5	Introducción teórica	39
5.1	<i>Introducción al problema de la no linealidad</i>	39
5.1.1	Distorsión armónica	40
5.1.2	Productos de intermodulación	41
5.1.3	Punto de compresión de 1dB	42
5.1.4	Punto de intercepto de tercer orden	43
6	Desarrollo	45
6.1	<i>Ejercicio 1: Comportamiento del amplificador de potencia, prueba de un tono</i>	45
6.1.1	Esquema de montaje	46
6.1.2	Mediciones previas	47
6.1.3	Descripción funcional	48
6.1.4	Diseño del panel frontal	51
6.1.5	Diseño del diagrama de bloques	52
6.1.6	Ejecución y resultados	62
6.2	<i>Ejercicio 2: Modulación ARB</i>	64
6.2.1	Esquema de montaje	65
6.2.2	Descripción funcional	65
6.2.3	Diseño del panel frontal	68
6.2.4	Diseño del diagrama de bloques	69
6.2.5	Ejecución y resultados	76
7	Conclusiones y líneas futuras de trabajo	79
8	Bibliografía	81
9	Anexos	83
9.1	<i>Datasheet del amplificador ZIL-6G+</i>	83
9.2	<i>Código para la generación de la señal WCDMA</i>	85
9.2.1	Función gold2047	86

ÍNDICE DE TABLAS

Tabla 2-1 Modulaciones posibles del SMIQ02B	6
Tabla 4-1 Controladores Plug and Play de los equipos	29
Tabla 6-1 Resultados de pérdidas en los cables	48
Tabla 6-2 Reference Level para los armónicos	57
Tabla 6-3 Frecuencia de los armónicos	62

ÍNDICE DE FIGURAS

Figura 2-1	Generador de señal SMIQ02B	3
Figura 2-2	Panel frontal del generador	4
Figura 2-3	Ejemplo de un tono	4
Figura 2-4	Esquema del modulador	5
Figura 2-5	Modulación vectorial	5
Figura 2-6	Menú de modulaciones analógicas	5
Figura 2-7	Esquema del generador ARB	6
Figura 2-8	Menú ARB MOD del SMIQ02B	6
Figura 2-9	Puerto GPIB en la parte trasera del generador	7
Figura 2-10	Analizador de señal Agilent N9010A EXA	7
Figura 2-11	Captura de pantalla del analizador	8
Figura 2-12	Ejemplo de medida del modo VSA	8
Figura 2-13	Agilent 82357B	9
Figura 3-1	Panel Frontal (Front Panel)	12
Figura 3-2	Diagrama de bloques (Block Diagram)	12
Figura 3-3	Opción mostrar ventanas	13
Figura 3-4	Barra de herramientas del diagrama de bloques	13
Figura 3-5	Paleta de herramientas	14
Figura 3-6	Paleta de controles	15
Figura 3-7	Ejemplo de un control del panel frontal	16
Figura 3-8	Ejemplo de un control modificado	16
Figura 3-9	Ejemplo de un control en el diagrama de bloques	16
Figura 3-10	Controles de ejemplo para el panel frontal	17
Figura 3-11	Paleta de funciones	18
Figura 3-12	Bucle For	19
Figura 3-13	Bucle While	19
Figura 3-14	Estructura Case	20
Figura 3-15	Estructura de secuencia	20

Figura 3-16 Pantalla de carga de LabVIEW	21
Figura 3-17 Menú principal de LabVIEW	21
Figura 3-18 Ventana de creación de proyectos	21
Figura 3-19 Propiedades del menú Ring	23
Figura 3-20 Panel frontal de la calculadora	23
Figura 3-21 Configuración de estructura Case	24
Figura 3-22 Diagrama de bloques de la calculadora	25
Figura 3-23 Ejecución de la calculadora desde el panel frontal	25
Figura 4-1 Uso de controladores de instrumentos	27
Figura 4-2 Esquema de capas para el control de instrumentos	28
Figura 4-3 Carpeta de drivers de LabVIEW	29
Figura 4-4 Controladores de instrumentos en la paleta de funciones	30
Figura 4-5 Árbol de bloques del controlador para el Agilent EXA n9010a	31
Figura 4-6 Árbol de bloques del controlador para el R&S SMIQ02b	32
Figura 4-7 Contenido de rssmiqu.llb	33
Figura 4-8 Breakpoint en controlador del SMIQ02b	33
Figura 4-9 Keysight Connection Expert	34
Figura 4-10 Configuración 488 en Connection Expert	34
Figura 4-11 Activación del fichero NiVisaTulip.dll en MAX	35
Figura 4-12 Bloque de inicialización	36
Figura 4-13 Diagrama de bloques de ejemplo para el control de instrumentos	36
Figura 4-14 Panel Frontal de ejemplo para el control de instrumentos	37
Figura 5-1 Esquema de un sistema lineal	39
Figura 5-2 Gráfica de un sistema lineal	39
Figura 5-3 Comportamiento en frecuencia de un sistema lineal	40
Figura 5-4 Distorsión armónica a la salida de un amplificador	41
Figura 5-5 Productos de intermodulación	41
Figura 5-6 Punto de compresión de 1 dB	42
Figura 5-7 Punto de intercepto de tercer orden	43
Figura 6-1 Amplificador Mini-Circuits ZJL-6G+	45
Figura 6-2 Esquema de montaje para el ejercicio 1	46
Figura 6-3 Conectores SMA macho (izquierda) y hembra (derecha)	47
Figura 6-4 Montaje para el cálculo de pérdidas en los cables	47

Figura 6-5 Diagrama de flujo para el ejercicio 1	50
Figura 6-6 Configuración del menú de unidades para el generador (izquierda) y el analizador (derecha)	51
Figura 6-7 Panel frontal del ejercicio 1	52
Figura 6-8 Inicialización de equipos	53
Figura 6-9 Adaptación de unidades para la frecuencia central	53
Figura 6-10 Cálculo del número de niveles de potencia del tono	53
Figura 6-11 Configuración de la potencia de entrada al generador	54
Figura 6-12 Bucle for perteneciente al generador de señal	55
Figura 6-13 Configuración de la frecuencia en el analizador	56
Figura 6-14 Configuración del Reference Level en el analizador	58
Figura 6-15 Calculo de la potencia a la entrada y a la salida del amplificador	59
Figura 6-16 Bucle del analizador	59
Figura 6-17 Configuración de la gráfica	60
Figura 6-18 Diagrama de bloques completo del primer ejercicio	61
Figura 6-19 Resultados obtenidos para el primer ejercicio	63
Figura 6-20 Componentes en fase y cuadratura	64
Figura 6-21 Esquema de montaje para el ejercicio 2	65
Figura 6-22 Diagrama de flujo para el ejercicio 2	67
Figura 6-23 Panel frontal del ejercicio 2	68
Figura 6-24 Estructura para la conversión de extensión	69
Figura 6-25 Configuración del generador para el ejercicio 2	70
Figura 6-26 Concatenar valores numéricos con instrucción SCPI	71
Figura 6-27 Configuración de la separación decimal en LabVIEW	72
Figura 6-28 Obtención de medidas para el ejercicio 2	73
Figura 6-29 Obtención de los valores del tiempo para el ejercicio 2	73
Figura 6-30 Diagrama de bloques del ejercicio 2 (parte 1)	74
Figura 6-31 Diagrama de bloques del ejercicio 2 (parte 2)	75
Figura 6-32 Resultados obtenidos para el segundo ejercicio	76
Figura 6-33 Resultados obtenidos en el analizador para el segundo ejercicio	77

3GPP	3rd Generation Partnership Project
AM	Amplitude Modulation
ARB	Arbitrary
BJT	Bipolar junction transistor
FET	Field-Effect Transistor
FM	Frequency Modulation
FSK	Frequency Shift Keying
GPIO	General Purpose Interface Bus
GPS	Global Positioning System
GSM	Global System for Mobile Communications
IEE	Institute of Electrical and Electronics Engineers
IVI	Interchangeable Virtual Instruments
PM	Phase Modulation
PnP	Plug and Play
QAM	Quadrature Amplitude Modulation
RAM	Random Access Memory
RAW	En bruto
RF	Radiofrecuencia
SCPI	Standard Commands for Programmable Instruments
SMA	SubMiniature version A
VI	Virtual Instrument
VISA	Virtual Instrument Software Architecture
VSA	Vector Signal Analysis
VXA	Vector Signal Analysis Measurement Application
WCDMA	Wideband Code Division Multiple Access

1 INTRODUCCIÓN

1.1 Introducción al tema

Los sistemas de radiocomunicación permiten el intercambio de información utilizando el espectro radioeléctrico como medio de transmisión. Debido a la expansión de este tipo de tecnología, sobre todo en comunicaciones móviles, resulta de interés realizar medidas, usando instrumentación de laboratorio, con el objeto de observar el comportamiento que tienen los dispositivos de radiofrecuencia. La obtención de algunos datos necesarios para caracterizar este tipo de dispositivos requiere la realización de miles de medidas, pudiéndose tardar horas e incluso días en completarse manualmente.

Mediante la herramienta de programación gráfica LabVIEW, es posible diseñar programas que automaticen una medida en concreto de forma remota. El desarrollo de estas aplicaciones permite controlar y adquirir datos de los equipos fácilmente en cuestión de minutos, de este modo, se reduce la tarea de configuración de los instrumentos de forma manual y el tiempo que ello conlleva.

1.2 Objetivo y alcance

En ese apartado se detallan cada uno de los objetivos que han motivado la realización de este documento:

- Conocer la plataforma y entorno de desarrollo de programación gráfica LabVIEW, estudiando su terminología y métodos básicos de utilización del programa.
- Introducir todas las configuraciones necesarias que permiten el control remoto de los equipos desde un ordenador que tenga instalado el software LabVIEW.
- Diseño de un programa en LabVIEW capaz de medir algunos de los efectos no lineales de un amplificador de RF.
- Diseño de un programa en LabVIEW que realice la modulación arbitraria de una portadora con una forma de onda introducida.

En definitiva, lo que se pretende con este proyecto es diseñar programas capaces de automatizar ciertas medidas que requieren una tarea costosa en el caso de que se hagan de manera manual. De la misma forma, puede servir como una guía para usuarios que deseen realizar cualquier aplicación para el control remoto de equipos de medida en LabVIEW.

1.3 Estructura de la memoria

Para explicar el desarrollo de este proyecto se ha decidido utilizar la estructura organizada por capítulos que se describe a continuación:

- Capítulo 2: “Material y método”. Se lleva a cabo una descripción de todo el material de

laboratorio que va a ser utilizados durante la realización del proyecto.

- Capítulo 3: “LabVIEW”. El programa que se va a manejar para el control de instrumentos es LabVIEW. En este capítulo se hará un recorrido para introducir los conceptos básicos que permiten desarrollar un programa de manera gráfica. Se incluyen ejemplos prácticos con el fin de utilizar todos estos conceptos diseñando un programa real.
- Capítulo 4: “Control de instrumentos”. Este capítulo contiene todo lo relacionado con el control remoto de los instrumentos a usar. En él se incluyen las configuraciones y controladores necesarios para poder manejar los instrumentos a través de LabVIEW. Para comprobar que toda la configuración se ha realizado correctamente se propone un ejemplo práctico que consiste en la realización de una medida básica.
- Capítulo 5: “Introducción teórica”. Se realiza un breve repaso teórico de algunos conceptos necesarios para entender el diseño de los programas a diseñar.
- Capítulo 6: “Desarrollo”. Aquí se explica, de forma detallada y ordenada, el diseño de dos programas en LabVIEW para realizar dos medidas diferentes de manera automática con los instrumentos que se han descrito. Al final de cada programa se comentan los resultados obtenidos tras su ejecución.
- Capítulo 7: “Conclusiones y líneas futuras”. Por último, en este capítulo se describen las conclusiones a las que se ha llegado tras la finalización del proyecto indicándose las futuras líneas de investigación.

2 MATERIAL Y MÉTODO

En este capítulo inicial se llevará a cabo una descripción de la instrumentación utilizada para la realización de medidas de radiofrecuencia y algunos de los elementos necesarios para el control de estos equipos. Para ello se ha hecho uso de los manuales proporcionados por los distintos fabricantes.

Una de las características del generador de señal Rohde & Schwarz SMIQ02B y el analizador de señal Agilent EXA es que pueden ser controlados remotamente por un PC mediante el bus GPIB por el cual se transmiten las diferentes órdenes que son procesadas por estos equipos.

Como bien se adelantaba anteriormente, en este proyecto se va a hacer uso de los equipos comentados de forma remota y es por ello que puede resultar interesante comentar algunos aspectos característicos sobre su control, así como de las instrucciones que manejan ambos dispositivos.

2.1 Generador de señal Rohde & Schwarz SMIQ02B

El generador de señal SMIQ02B permite generar diferentes tipos de señales arbitrarias desde 300 kHz hasta 2.2 GHz ofreciendo una gran variedad de opciones de modulación que pueden ser configuradas por el usuario.

La selección de estos parámetros se puede realizar directamente usando el control manual desde el panel frontal que dispone el equipo o bien mediante control remoto desde un PC mediante el bus GPIB.



Figura 2-1 Generador de señal SMIQ02B

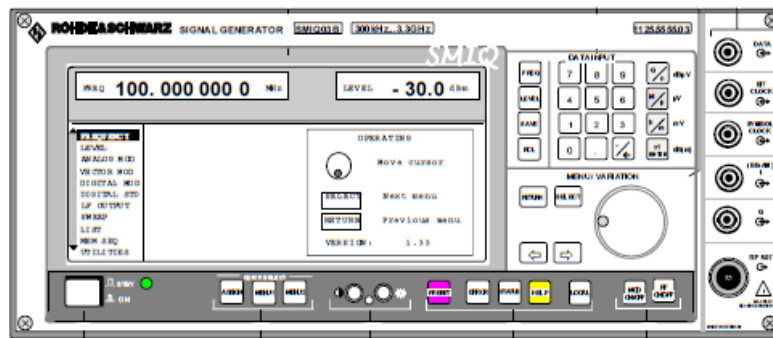


Figura 2-2 Panel frontal del generador

En la pantalla principal del panel frontal que aparece en la Figura 2-2, se puede observar la frecuencia y potencia de la señal de salida por el puerto RF OUTPUT que es posible modificar con la ruleta de variación, o bien haciendo uso del teclado numérico que aparece en la parte derecha del panel. En la parte central, ocupando la mayor parte del espacio, se encuentra el display del generador donde se muestra el menú navegable que servirá para realizar configuraciones o para consultar datos del equipo.

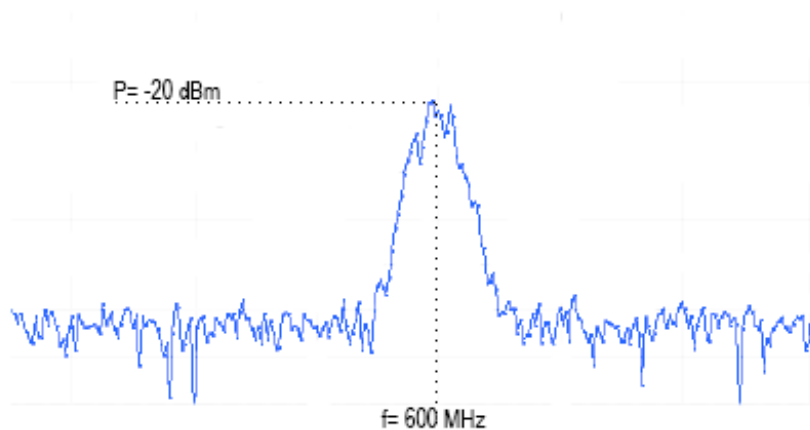


Figura 2-3 Ejemplo de un tono

Si se quisiera generar un tono como el de la Figura 2-3 se pulsaría el botón **FREQ** del panel frontal y se seleccionaría una potencia de 600 MHz con la ruleta o panel numérico. De igual forma, presionando sobre **LEVEL** se configuraría la potencia a -20 dBm.

2.1.1 Modulaciones

Este generador de señal lleva incorporado en su interior el modulador SMIQ-B20 que soporta una gran cantidad de modulaciones digitales, desde FSK hasta 256QAM, además de muchos de los estándares de comunicaciones móviles más populares.

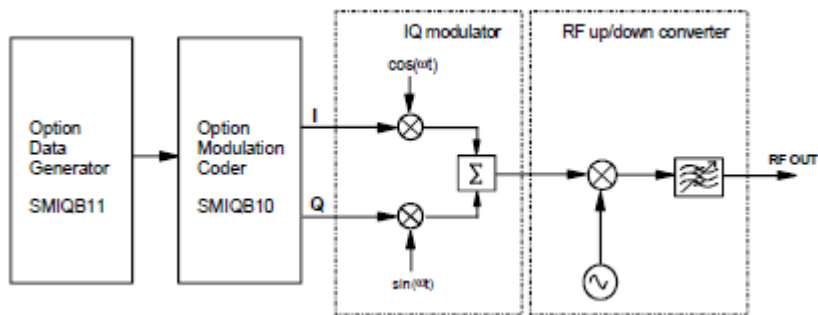


Figura 2-4 Esquema del modulador

Dentro de las modulaciones digitales se incluye la modulación vectorial. En este tipo de modulación es posible introducir las componentes I (*In-Phase*) y Q (*Quadrature*) de manera externa para una modulación compleja de la portadora tal y como aparece en la Figura 2-5

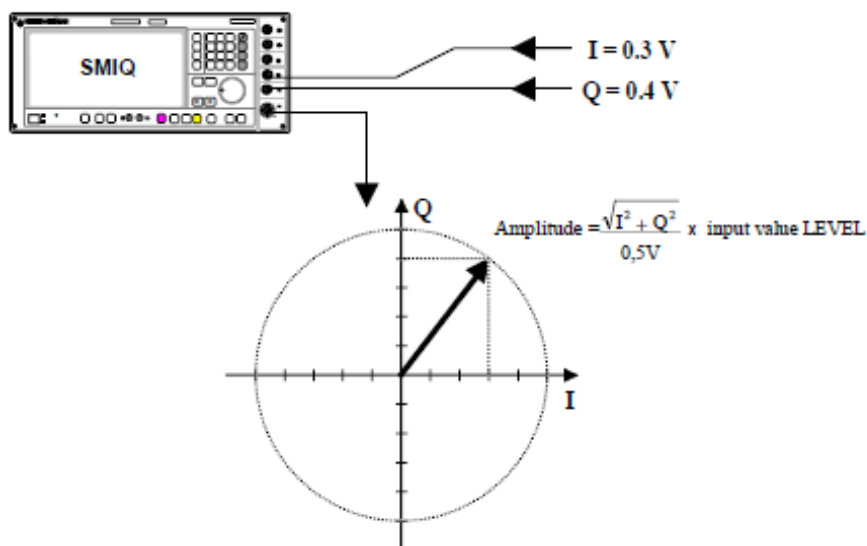


Figura 2-5 Modulación vectorial

A parte de las digitales también permite configurar las modulaciones analógicas en amplitud, fase, frecuencia y pulso más conocidas.

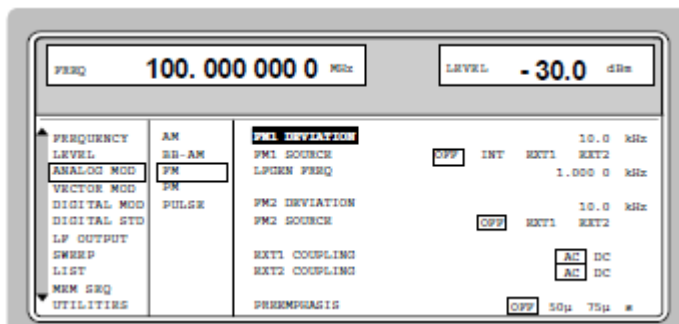


Figura 2-6 Menú de modulaciones analógicas

En la Tabla 2-1 se recogen las distintas modulaciones soportadas por el generador y los estándares que se han comentado anteriormente.

Modulaciones Analógicas: Modulaciones Digitales: Estándares:

- AM
- BB-AM
- FM
- PM
- Pulse modulation
- PSK
- QAM
- $\pi/4$ DPSK
- FSK
- Vectorial
- PHS
- IS-95
- W-CDMA
- 3GPP W-CDMA
- NADC
- PDC
- GSM/EDGE
- DECT
- GPS

Tabla 2-1 Modulaciones posibles del SMIQ02B

2.1.2 Generador de forma de onda arbitraria

Este generador lleva incorporado internamente la opción SMIQB60, un generador de formas de onda arbitrarias de dos canales. Las formas de onda se pueden cargar directamente en el equipo SMIQ y almacenarlas en la memoria RAM interna. El SMIQB60 proporciona señales I/Q arbitrarias al modulador I/Q del SMIQ02B.

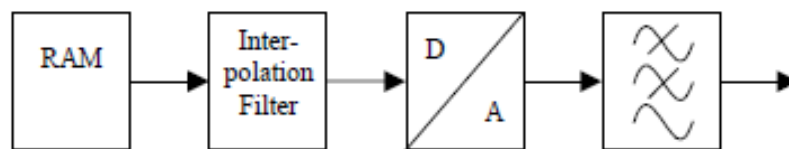


Figura 2-7 Esquema del generador ARB

La señal moduladora se genera de forma externa al generador con MATLAB u otro software y se debe transferir al SMIQ con la extensión *.vw. Este fichero incluye las muestras de la componente en fase y cuadratura de la señal y su frecuencia de muestreo. Tras seleccionar la señal moduladora (fichero .vw en el generador) se procede a modular la portadora previamente configurada con la señal arbitraria.



Figura 2-8 Menú ARB MOD del SMIQ02B

Mediante el menú del equipo se pueden configurar las diferentes características de la modulación (Trigger, Reloj...), y gestionar los ficheros con formas de onda almacenados en la RAM.

distintas ventanas correspondientes a cada tipo de medida.

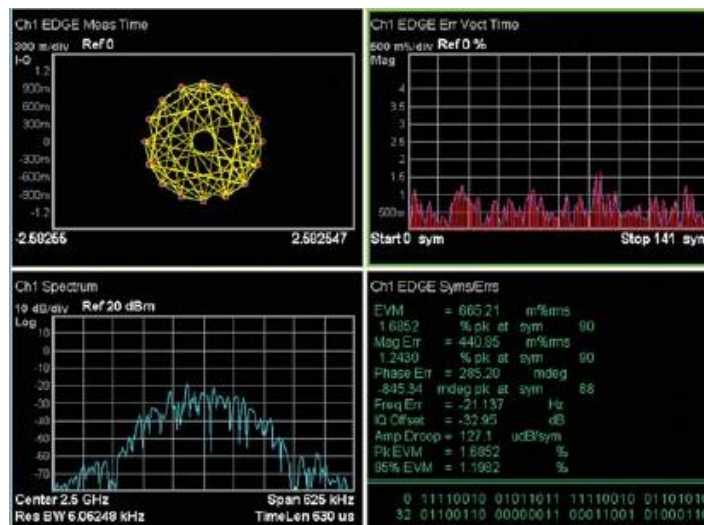


Figura 2-11 Captura de pantalla del analizador

2.2.1 Modos de funcionamiento

Es posible configurar el analizador de espectro para que funcione en diferentes modos, estos son:

- **Analizador de espectros:** Sirve para realizar las medidas típicas del analizador de espectros corriente con barrido de frecuencia y análisis FFT.
- **Analizador I/Q:** Mediante este modo se puede obtener una medida tanto en el dominio de la frecuencia como en el del tiempo.
- **VXA:** Se incluyen herramientas para la demodulación de una señal y su análisis vectorial. Es posible verificar el funcionamiento de una señal en tiempo real con varias vistas simultaneas en el dominio de la frecuencia, tiempo y modulación.
- **VSA:** Es similar al modo VXA pero ofreciendo mayor flexibilidad en cuanto a la configuración de los parámetros de las medidas a realizar. El software que incluye para este modo es el 89600.

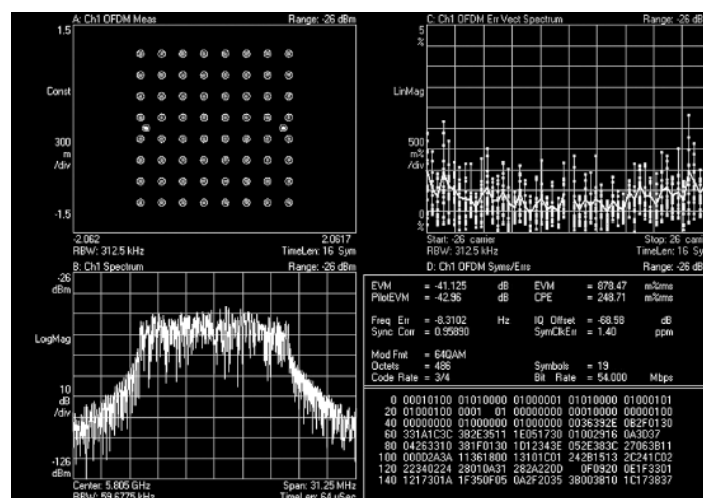


Figura 2-12 Ejemplo de medida del modo VSA

2.3 Bus GPIB (IEEE 488)

El bus GPIB (*General Purpose Interface Bus*) es un estándar de transmisión de datos capaz de conectar instrumentos de medición programables con dispositivos que los controlen mediante ordenes como puede ser un ordenador.

Inicialmente, Hewlett Packard diseñó a finales de los 60 el estándar HP-IB para conectar y controlar equipos de su propia marca. Más adelante surgió la necesidad de crear una interfaz capaz de interconectar instrumentos de distintos fabricantes, de esta forma surge en 1975 el estándar IEE 488 publicado por el Institute of Electrical Electronic Engineers que contiene las características eléctricas, mecánicas y funcionales del sistema de interfaz.

El GPIB es un bus en paralelo de 8 bits y permite una velocidad de transferencia de hasta 8 MB/s dependiendo de la versión. Es posible usar diversas topologías de conexión debido a sus formas de conectores macho y hembra. Tiene una estructura de 24 pines donde 8 de ellos son para datos, 3 para *handshake*, 5 para la gestión del bus y 8 de masa.

Para la realización de este proyecto se hará uso de la interfaz USB/GPIB Agilent 82357B para la conexión del controlador (PC) con los equipos y de cables con conectores macho y hembra del estándar IEE 488.



Figura 2-13 Agilent 82357B

Para el correcto funcionamiento de esta interfaz es necesario instalar los drivers del fabricante **Keysight IO Libraries Suite** que se pueden descargar de manera gratuita desde la web oficial de Keysight.

2.4 Comandos SCPI

Los comandos SCPI (*Standard Commands for Programmable Instruments*) definen un estándar para la sintaxis de comandos a usar en dispositivos controlables como los vistos anteriormente.

Estos comandos fueron incluidos en 1960 dentro del estándar IEEE 488.2 en la *sección Standard Codes, Formats, Protocols, and Common Commands*. En esta sección se especifica la estructura de comandos y el formato de datos que pueden ser usados con todos los instrumentos sin dependencia del fabricante del equipo. Se agrupan en subsistemas o familias que pertenecen a un instrumento, por lo cual no todos los instrumentos tienen los mismos bloques.

Algunos ejemplos básicos de comandos SCPI genéricos serían:

- *RST - Resetea el instrumento a los valores iniciales
- *IDN? - Identificación del instrumento
- *CLS – Limpia el estado

La estructura de los comandos va por niveles separados por dos puntos, al nivel principal se le conoce como raíz y horizontalmente le siguen los siguientes niveles o subfamilia de la raíz.

Un ejemplo de comando para el control de un instrumento sería:

```
SENSe:FREQuency:STARt 2GHZ
```

En este caso, la raíz o nivel principal sería SENSe mientras que FREQuency y STARt serían subfamilias de la raíz. Con este comando se establecería en 2 GHz la frecuencia central de un barrido. El uso de las letras que aparecen minúsculas no es necesario, pero se utilizan para reconocer la instrucción con más facilidad y solo sería obligatorio utilizar las mayúsculas, por lo tanto, el mismo comando podría escribirse también de la siguiente forma:

```
SENS:FREQ:STAR 2GHZ
```

3 LABVIEW

3.1 Descripción del programa y sus posibilidades

El programa LabVIEW, creado por la compañía National Instruments, es un entorno de desarrollo que utiliza un lenguaje de programación gráfico y está destinado al diseño de aplicaciones de diversos propósitos, como por ejemplo la adquisición de datos de instrumentos, procesado, análisis de datos y control remoto de instrumentos y equipos.

Básicamente LabVIEW nos permite obtener datos del mundo real en un ordenador, tomar decisiones en base a los resultados recogidos y mandar instrucciones a los equipos para modificar o automatizar cualquier operación. Un ejemplo práctico del uso de LabVIEW sería la adquisición de los valores medidos en un acelerómetro y la visualización gráfica de los movimientos en los 3 ejes, estos datos se representarían en el panel frontal de programa a través de un *display*.

El uso de un sistema de programación gráfica como LabVIEW tiene muchas **ventajas** que facilitarán en gran medida el diseño de un programa.

- Sistema muy intuitivo fácil de aprender y manejar, sin necesidad de usar cientos de comandos.
- Tiempo de desarrollo más bajo que con cualquier otro lenguaje de programación.
- No es necesaria la instalación de compiladores o entornos de programación, LabVIEW viene preparado para comenzar a programar inmediatamente después de su instalación.
- Posibilidad de incorporar código escrito en otro lenguaje de programación.
- Trabajar con diferentes plataformas de hardware desde un mismo software, existiendo una gran compatibilidad con los equipos y dispositivos de cada marca.

3.2 Entorno de edición de LabVIEW

A los programas diseñados en LabVIEW se les llama VIs que son las siglas de Virtual Instruments, es decir, instrumentos virtuales. Como su propio nombre indica, cada programa simula un instrumento con su panel frontal cargado de botones, indicadores, leds y pantallas, todo ello conectado a una circuitería interna que gestiona el funcionamiento de los elementos en conjunto.

3.2.1 Panel frontal y diagrama de bloques

Cada VI está estructurado en dos partes que aparecen como ventanas:

- **Panel Frontal (Front Panel):** Es la interfaz gráfica para el usuario, simula la carcasa rígida de un instrumento. En esta ventana se configuran los elementos de entrada que proporcionan datos al diagrama de bloques y los elementos de salida que permiten visualizar información generada en la ejecución. Se puede ver este panel en la Figura 3-1.

- **Diagrama de bloques (Block Diagram):** Contiene toda la estructura de programación compuesta por bloques que se unen entre sí para formar el programa final o instrumento virtual. Los elementos de entrada y salida también aparecen en el diagrama de bloques como terminales. Un ejemplo de esta ventana se muestra en la Figura 3-2.

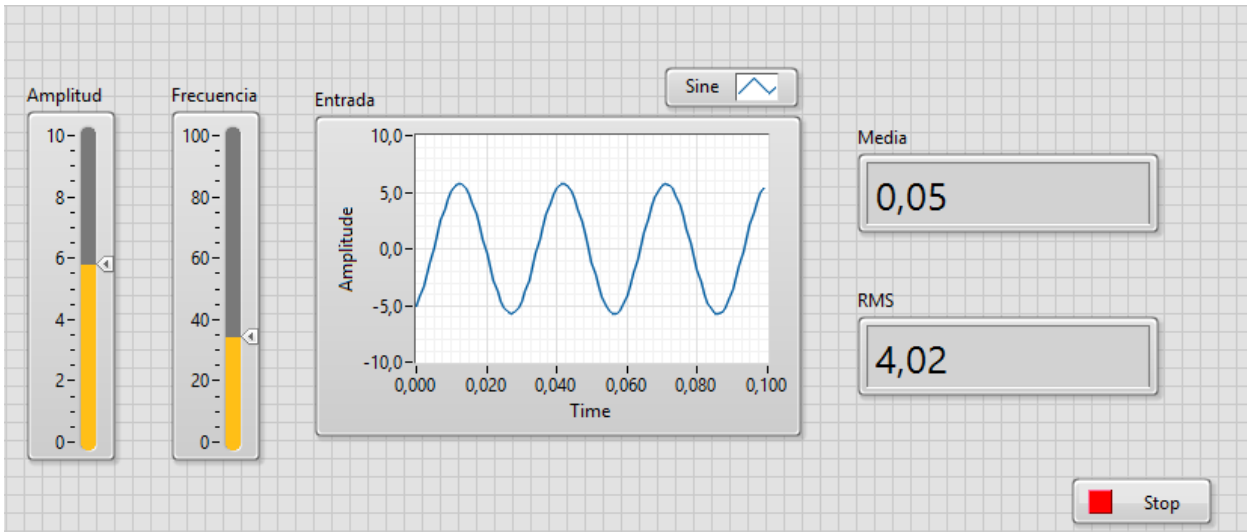


Figura 3-1 Panel Frontal (Front Panel)

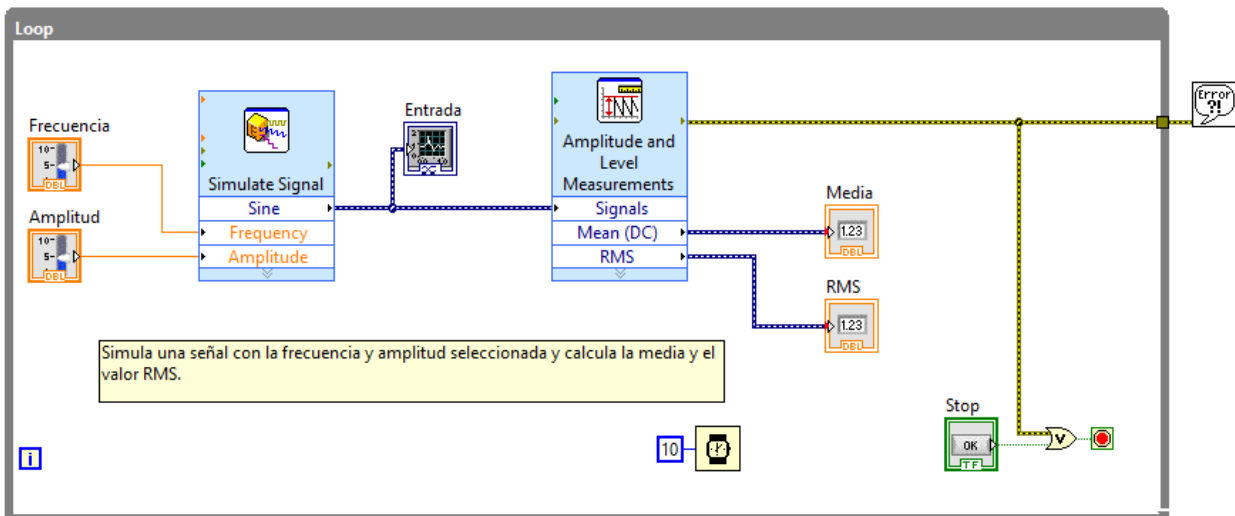


Figura 3-2 Diagrama de bloques (Block Diagram)

La conmutación entre estas dos ventanas se realiza desde el menú **Windows** haciendo click en **Show Block Diagram/Front panel** o usando el comando de teclado **Ctrl + E**.

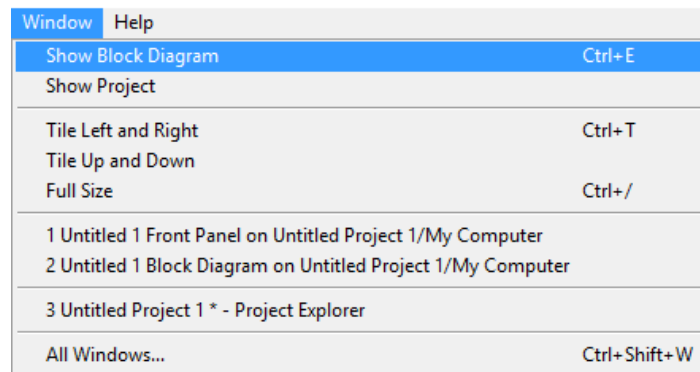


Figura 3-3 Opción mostrar ventanas

Cada bloque del diagrama representa un nodo, cuando un bloque se ejecuta puede producir datos que se pasan al nodo que está conectado inmediatamente después. La ejecución del programa se realiza de izquierda a derecha ejecutando todos los bloques y entrando en las estructuras o bucles del tipo while, for, etc.

En el ejemplo de la Figura 3-1 y la Figura 3-2 se presenta un instrumento virtual que genera una señal sinusoidal con amplitud y frecuencia seleccionada en el panel frontal y calcula su valor medio y eficaz (RMS).







3.2.2 Barra de herramientas








El programa puede iniciarse o interrumpirse en cualquier momento desde el *Front Panel* o el *Block Diagram* usando la barra de herramientas que ofrece diferentes opciones posibles. Esta barra se encuentra en la parte superior.



Figura 3-4 Barra de herramientas del diagrama de bloques

Los botones que forman la barra de herramienta son los siguientes:

- **Run** . Inicia la ejecución del programa VI codificado en el diagrama de bloques. Si el botón Run aparece con el aspecto  durante el diseño del programa, indica que existe un error en la estructura del diagrama de bloques que impide la ejecución del VI. Si se pulsa sobre este icono se muestra la ventana **Error List** que incluye los errores y advertencias que producen esta situación.
- **Run Continuously** . Ejecuta el programa en bucle continuo, es decir, una vez que se ha llegado al final de la ejecución vuelve a iniciarse.
- **Abort Execution** . Aborta la ejecución del programa de forma inmediata.
- **Pause** . Pausa la ejecución. Durante el tiempo que está pausado, el icono aparece coloreado en rojo . Pulsando el botón de nuevo se continúa con la ejecución normal.

- **Herramienta de Etiquetado** . Para crear y editar etiquetas de texto. Con ella también es posible editar los valores de algunos controles y constantes.
- **Herramienta de Cableado** . Con esta herramienta se unen los objetos del diagrama de bloques. Para crear un cable basta con hacer click en ambos bloques.
- **Herramienta de Menú de Acceso Directo de Objetos** . Despliega un menú con las opciones disponibles según el objeto.
- **Herramienta de Desplazamiento** . Permite moverse en las ventanas sin tener que utilizar las barras de desplazamiento.
- **Herramienta de Breakpoint** . Con ella se crean puntos de parada dentro del diagrama de bloques. La colocación de estos puntos hace que la ejecución se interrumpa al pasar por ellos.
- **Herramienta de Probe** . Sirve para colocar puntos de prueba en el diagrama de bloques. Con estos puntos es posible capturar resultados durante la ejecución.
- **Herramienta de Copiar Color** . Copia el color de un determinado objeto

3.2.4 Controles e Indicadores

Los controles e indicadores son los diferentes elementos que se van a incluir para formar el panel frontal del instrumento virtual. La selección de cualquiera de estos objetos en concreto se hace a través de la ventana Controls que se abre haciendo click derecho en cualquier lugar del panel frontal.

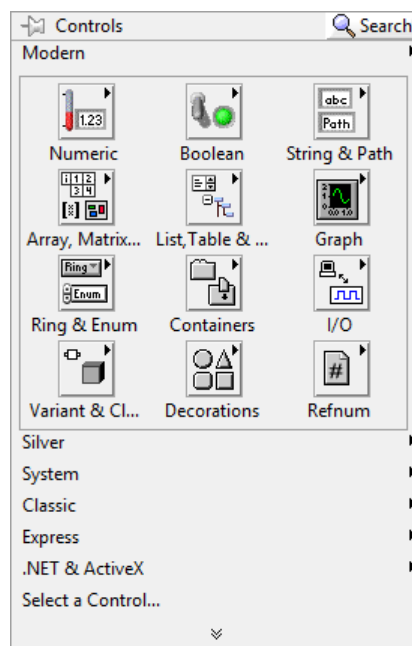


Figura 3-6 Paleta de controles

Existe una gran variedad de objetos que pueden ser incluidos en el panel frontal, todos ellos están ordenados por secciones para ser localizados fácilmente. Además, existe un buscador en la esquina superior derecha que permite realizar la búsqueda de un objeto del que se desconozca su localización.

Haciendo click sobre cualquier control o indicador se colocan en el panel frontal. Con la herramienta de etiquetado vista anteriormente se puede cambiar el título de los controles e indicadores y los valores que aparecen.

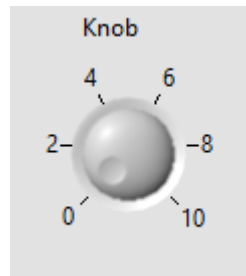


Figura 3-7 Ejemplo de un control del panel frontal

En la Figura 3-7 puede observarse un control del tipo *knob* que simula la típica ruleta incluida en la carcasa de un instrumento. Su valor se puede modificar girando la ruleta con la herramienta de operación de la paleta de herramientas.

Existe la posibilidad de modificar las propiedades de cada objeto pulsando con el botón derecho en el elemento y seleccionando la opción **Properties** del menú desplegable. Desde aquí se configura el aspecto con el que va a ser presentado el objeto en el panel y el tipo de datos que va a manejar entre otras cosas.

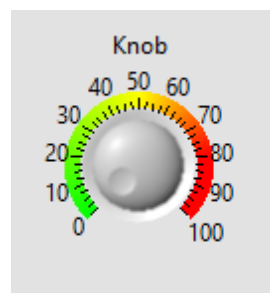


Figura 3-8 Ejemplo de un control modificado

Al colocar un elemento en el panel frontal también se añade al diagrama de bloques con otro icono para poder cablearlo con los demás objetos, el control se convierte en un terminal al pasar a la ventana del diagrama. Los terminales son puertos de entrada y salida que intercambian información entre el panel frontal y el diagrama de bloques. De esta forma el *knob* del ejemplo anterior presentaría el siguiente aspecto en el diagrama de bloques:

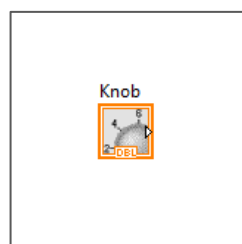


Figura 3-9 Ejemplo de un control en el diagrama de bloques

La Figura 3-10 muestra un ejemplo de los controles disponibles para añadirse al panel frontal.

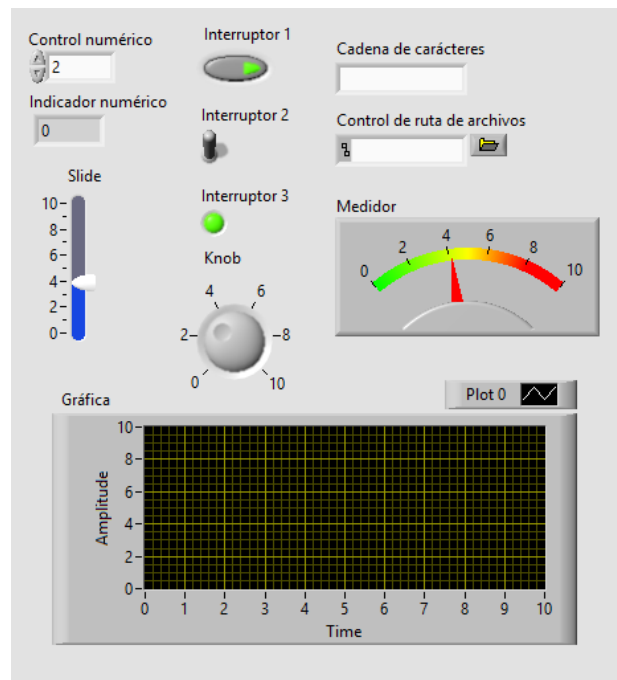


Figura 3-10 Controles de ejemplo para el panel frontal

3.2.5 Funciones

En el diagrama de bloques, a los elementos fundamentales de operación que forman la estructura del programa se les llama funciones. El acceso a las diferentes funciones se hace mediante una paleta destinada a ello que se muestra pulsando en **View/ Functions Palette** o haciendo click derecho en cualquier zona del diagrama de bloques.

Las funciones están divididas en diferentes secciones dependiendo del propósito de cada una de ellas. Por si resultara costoso encontrar alguna, se puede buscar una función en concreto acudiendo al buscador de la parte superior derecha de esta paleta.

La paleta de funciones se presenta de la siguiente forma:

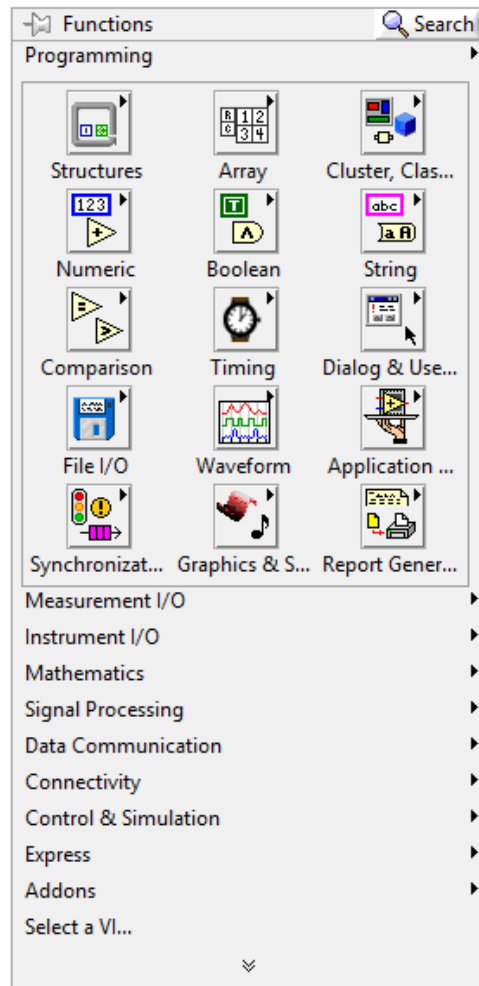


Figura 3-11 Paleta de funciones

Existe una gran variedad de funciones posibles en LabVIEW, algunas son típicas de los lenguajes de programación habituales (Operadores matemáticos, sentencias, arrays, manejo de ficheros) y otras destinadas a tareas más específicas como pueden ser la conectividad y el control de instrumentos externos.

3.2.5.1 Estructuras

Como en cualquier lenguaje de programación, en LabVIEW se incluyen las sentencias más conocidas que permiten ejecutar un bloque un número determinado de veces, repetirlo hasta que se alcance un requisito o ejecutarlo o no dependiendo de unas condiciones. Todas estas estructuras están incluidas como funciones dentro de la paleta de la Figura 3-11.

Las estructuras más importantes que se incluyen en la carpeta **Structures** son:


- **Bucle For** . Ejecuta el conjunto incluido dentro de esta estructura un número n de veces, donde n es el valor conectado al terminal N de la esquina superior izquierda. En la esquina inferior izquierda, el terminal i proporciona el valor que indica en que repetición se encuentra el bucle, siendo el valor inicial 0 en la primera iteración, 1 en la segunda y así sucesivamente.



Figura 3-12 Bucle For




- **Bucle While** . Repite lo que se encuentra dentro hasta que se cumpla una condición. El terminal  evalúa un valor booleano que, dependiendo de si es TRUE o FALSE, seguirá o no ejecutando el bloque. El terminal **i**, como en el bucle for, indica el número de la iteración del bucle. Este bucle siempre se ejecuta al menos una vez.



Figura 3-13 Bucle While

- **Estructura Case** . Dentro de esta estructura se pueden incluir varios subdiagramas que se ejecutarán dependiendo del caso en el que se encuentre. El terminal **?** será el que reciba el valor del caso que se va a arrancar, puede ser un entero, un carácter booleano o incluso una cadena de caracteres. Con el selector de la parte superior se configuran los distintos subdiagramas que corresponden a un caso, para modificar el valor de cada caso hay que utilizar la herramienta de etiquetado de la paleta de herramientas en dicho selector.

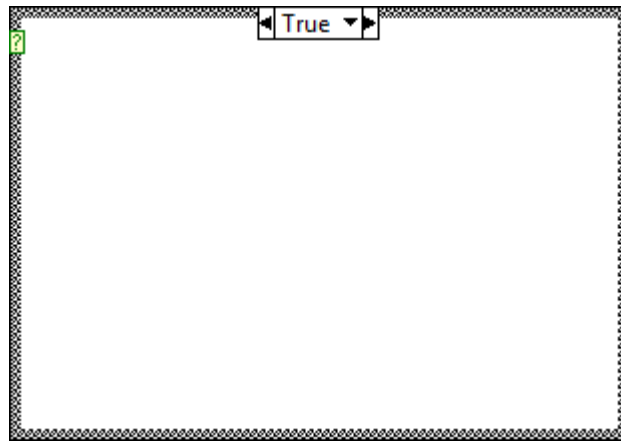



Figura 3-14 Estructura Case

- Estructura de Secuencia** . Consiste en varios cuadros donde se incluyen subdiagramas. Cada cuadro se ejecutará de forma secuencial de izquierda a derecha. Es importante usar esta estructura si se requiere una ejecución ordenada del programa, aunque no es aconsejable abusar de ella.

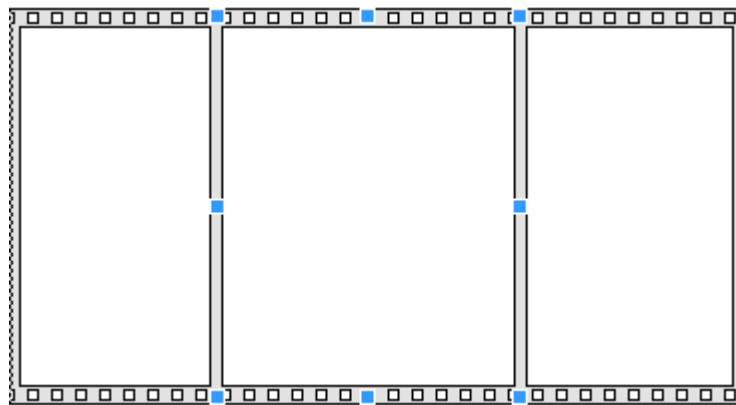


Figura 3-15 Estructura de secuencia

Todas estas estructuras tienen en común que no se ejecutarán o se entrará en ellas hasta que las variables que involucren su ejecución no estén preparadas.

3.3 Creación de un VI como ejemplo

En esta sección se realizará un instrumento virtual en LabVIEW desde cero para entender de una forma más práctica lo expuesto en este capítulo.

El programa en cuestión consistirá en una calculadora con cinco operaciones posibles (suma, resta, multiplicación, división, potencia) entre dos números que se introducirán en el panel frontal.

Lo primero es abrir el programa LabVIEW haciendo click en su acceso directo y esperando a que se cargue correctamente. La carga de LabVIEW puede tardar algunos minutos dependiendo del ordenador desde el cual se esté ejecutando.

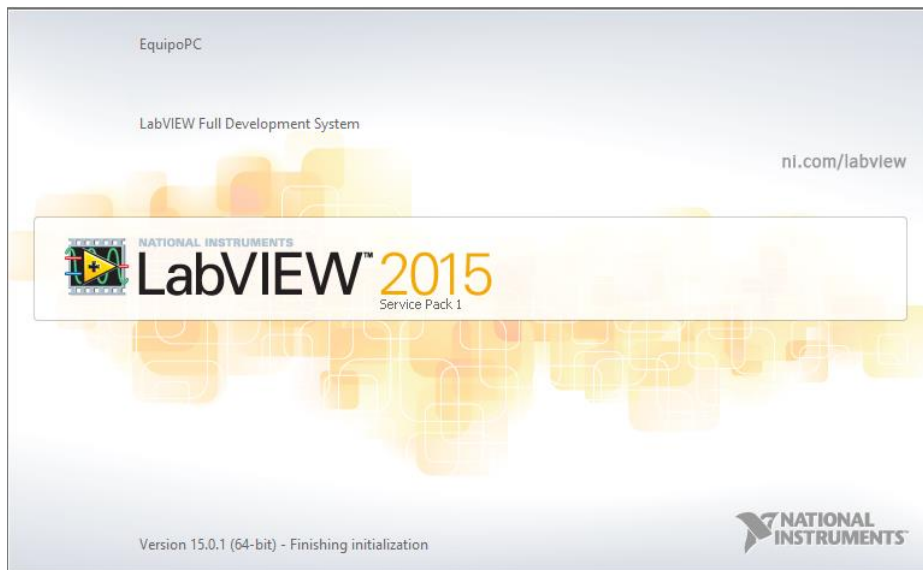


Figura 3-16 Pantalla de carga de LabVIEW

Cuando se complete la carga, aparecerá el menú principal del programa. Desde esta ventana se puede crear un nuevo proyecto o abrir un proyecto ya existente en el ordenador.

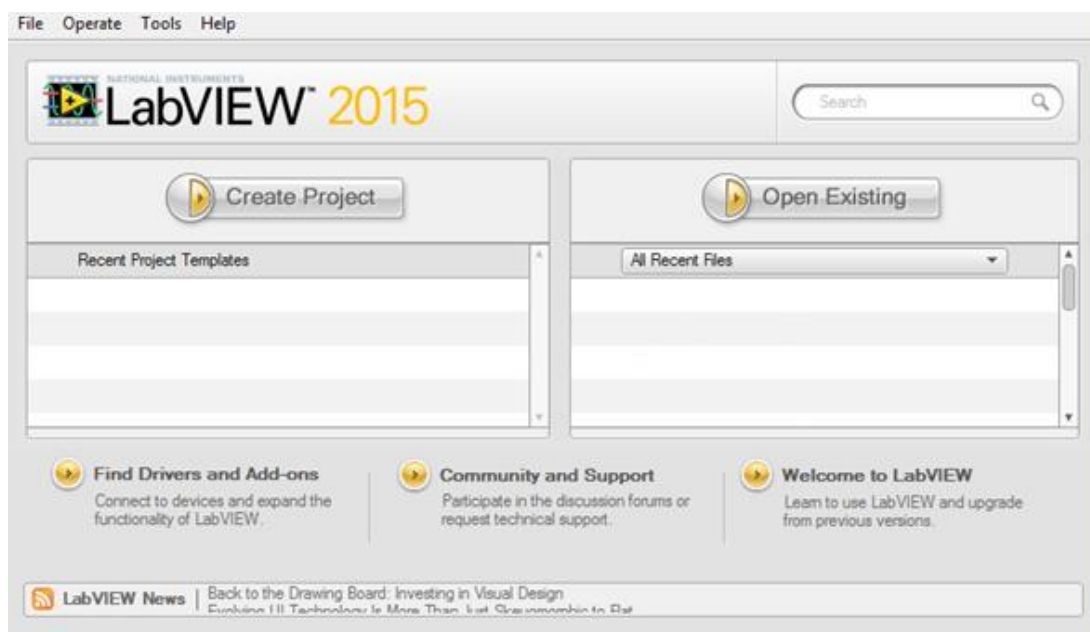


Figura 3-17 Menú principal de LabVIEW

Para crear un nuevo proyecto lo más sencillo es pulsar sobre **Create Project** y seleccionar **Blank VI** dentro de las posibles plantillas.

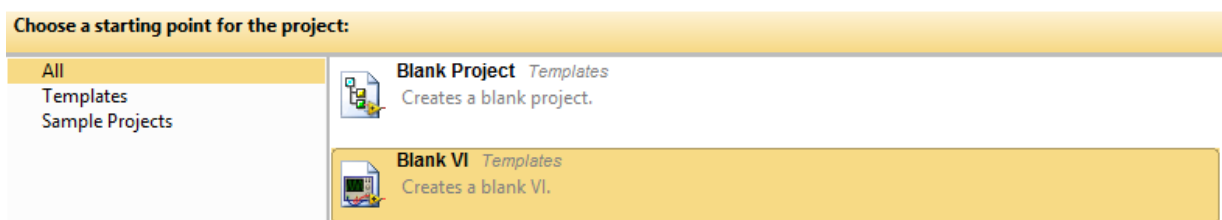






Figura 3-18 Ventana de creación de proyectos

Tras seleccionar un instrumento virtual vacío, se abrirán las ventanas del panel frontal y diagrama de bloques descritas anteriormente. Es posible que la primera vez que se inicie el programa solo se muestre la ventana del panel frontal, desde **Window/ Show Block Diagram** se abre el diagrama de bloques en caso de que no aparezca.

Lo más lógico para este tipo de programas será diseñar el panel frontal en primer lugar para así tener una primera idea de la estructura y de los elementos que van a componer el programa. Cabe recordar que para añadir un elemento se hace desde la paleta de controles, abriéndola pulsando el botón derecho del ratón en cualquier zona del panel frontal.

En este ejemplo los elementos que van a componer el panel frontal son los siguientes:

- **Controles numéricos** . En ellos se incluirán los números que van a realizar la operación entre sí. Se colocarán dos elementos como estos por ser una calculadora de dos números, se pueden poner más controles si se desea para hacer una calculadora de más operaciones. Este control se encuentra en la sección **Numeric** dentro de la paleta de controles.
- **Indicador numérico** . Aquí se mostrará el valor resultante de la operación que se lleva a cabo. Se encuentra también dentro de **Numeric** en la paleta de controles.
- **Menú** . Este control permitirá seleccionar la operación que se va a aplicar entre los dos números introducidos en los controles numéricos. En la paleta de controles aparece dentro de **Ring & Enum**.
- **Botón Stop** . Con este botón se parará el funcionamiento de la calculadora. Se encuentra en la sección **Boolean**.

Mediante la herramienta de texto se cambian las etiquetas de cada elemento para que todo quede más claro a simple vista.

Cada operación posible del menú se identificará con un número entero, este número servirá como parámetro de entrada en una estructura del tipo Case que se incluirá en el diagrama de bloques. Accediendo a las propiedades del elemento Ring haciendo click derecho sobre el elemento y pulsando en **Properties** es posible cambiar la configuración para que quede de la siguiente forma:

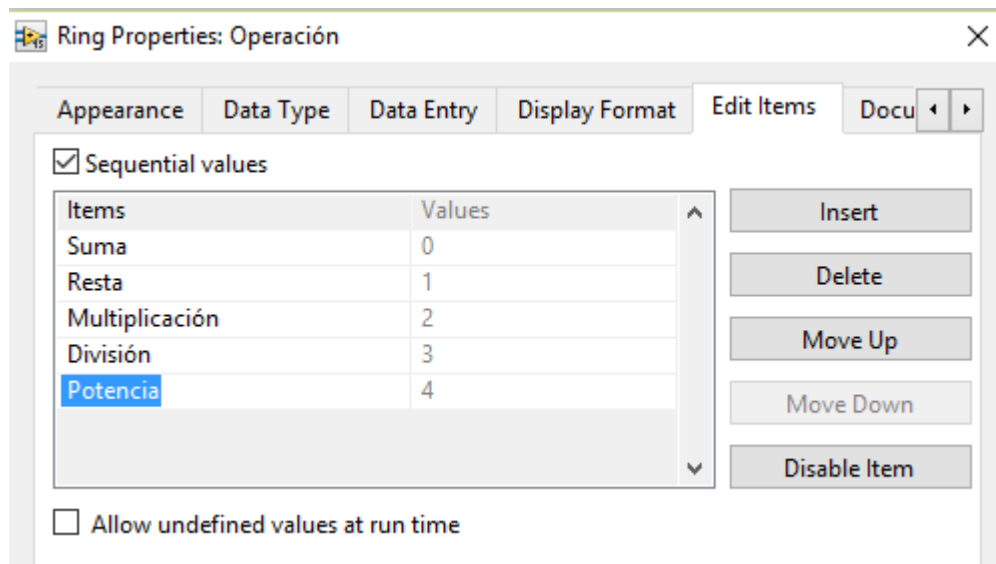


Figura 3-19 Propiedades del menú Ring

En la pestaña Edit Items se organizan los elementos que se han añadido secuencialmente pulsando el botón Insert y renombrando cada uno de ellos.

De esta forma el panel frontal de la calculadora estaría completado.

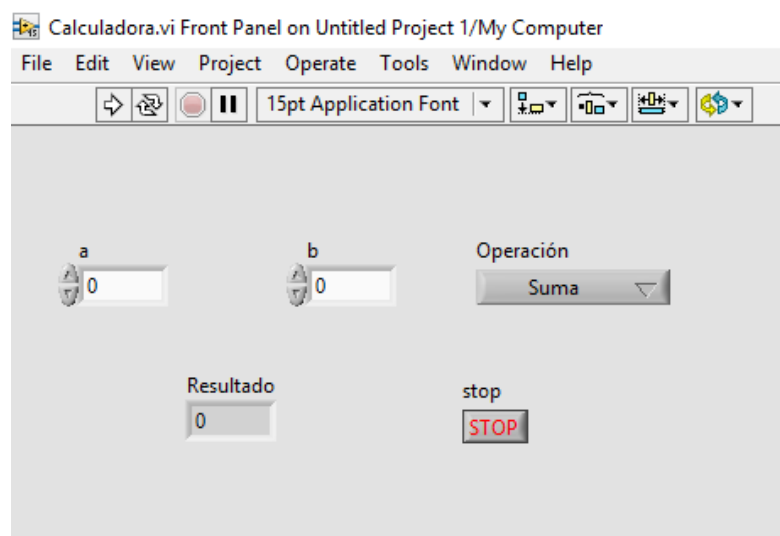





Figura 3-20 Panel frontal de la calculadora

Todos los elementos creados en el panel frontal también se añaden automáticamente en el diagrama de bloques para su futura conexión. Además de estos habría que incluir los siguientes:

- **Estructura Case** . Contendrá varias ventanas, cada una de ellas albergará una función diferente dependiendo de la operación matemática que se quiera llevar a cabo. Este diagrama se encuentra dentro de **Structures** en la paleta de funciones.
- **Bucle While** . Estructura que se incluye para que el programa se ejecute en bucle hasta que se presione el botón de Stop desde el panel frontal que parará la ejecución. Se encuentra dentro de **Structures** en la paleta de funciones.

- **Operaciones numéricas** . Realizaran las operaciones entre los dos números. Habría que colocar cuatro para cada caso (suma, resta, multiplicación, división, potencia), cada una tiene un símbolo diferente. Se encuentra dentro de **Numeric** en la paleta de funciones.

Con el botón derecho y pulsando sobre **Add Case** se añaden las ventanas que representarán cada caso, para este ejemplo hay que añadir cinco casos enumerados del 0 al 4 con la herramienta de texto.

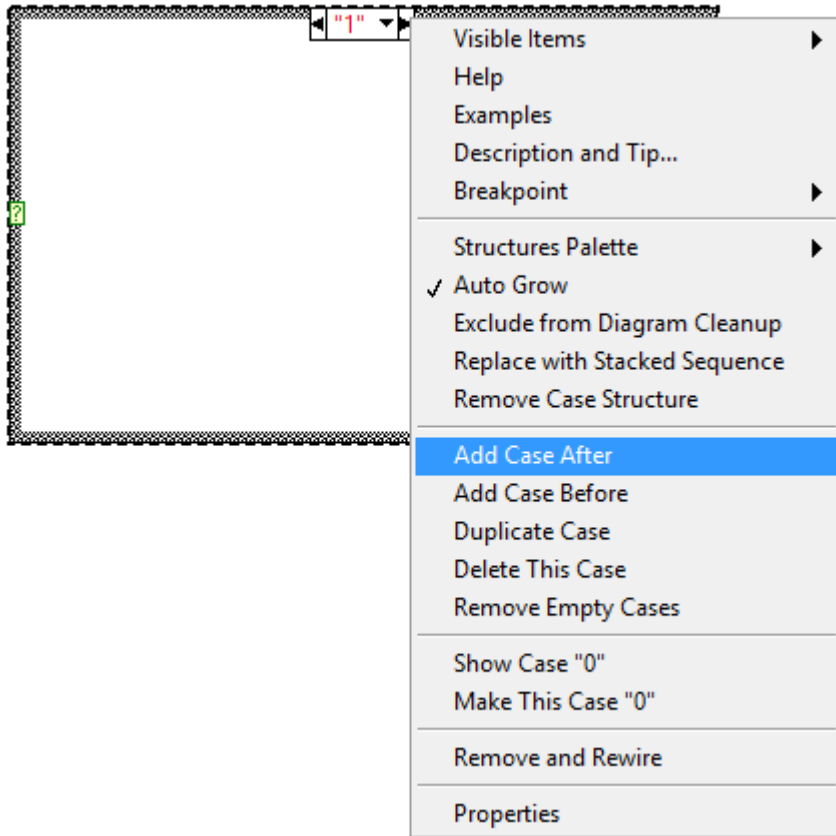


Figura 3-21 Configuración de estructura Case

A continuación, se procede a conectar los elementos entre sí para formar el programa tal y como muestra la Figura 3-22.

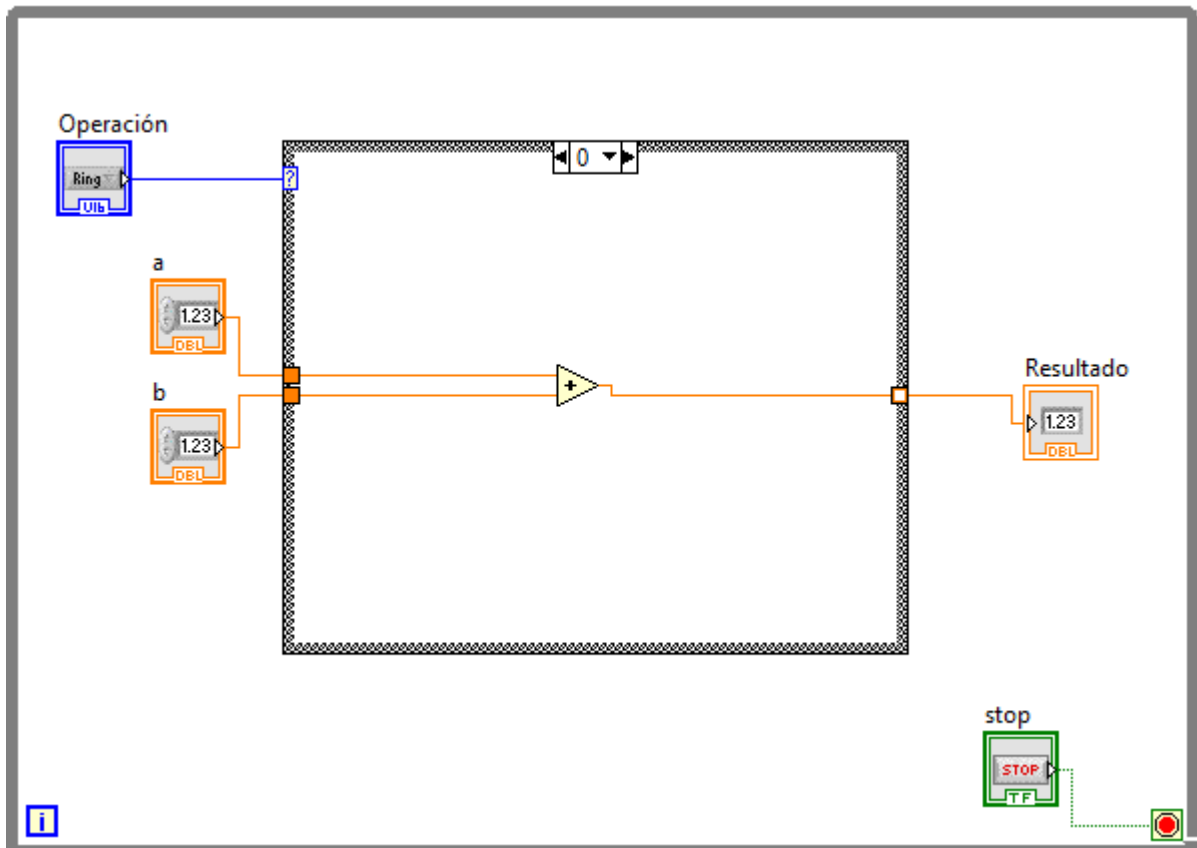


Figura 3-22 Diagrama de bloques de la calculadora

Cabe destacar que en la imagen se representa solo el caso 0 que se correspondería con la suma. Los demás casos tendrán una estructura similar cambiando solo el elemento de la operación numérica que será diferente en cada ventana.

Ejecutando el programa con el botón **Run** desde el panel frontal y cambiando los valores de a y b obtendremos un resultado similar al siguiente:

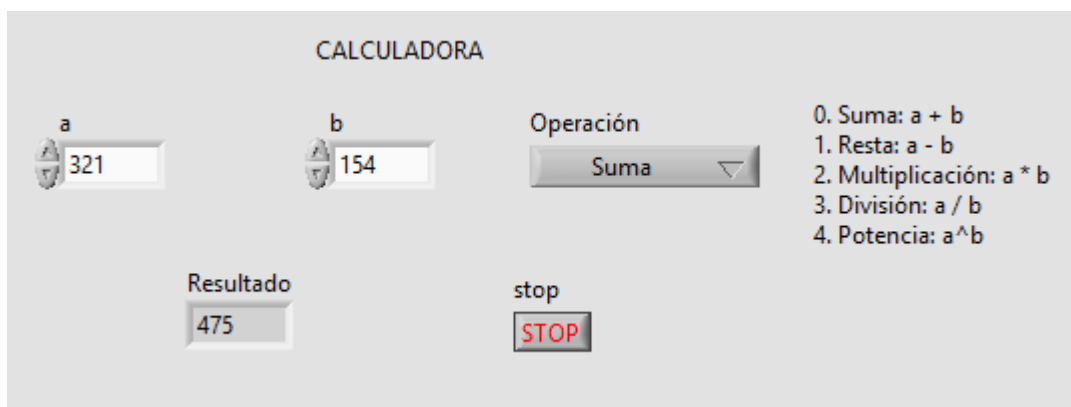


Figura 3-23 Ejecución de la calculadora desde el panel frontal

4 CONTROL DE INSTRUMENTOS

Como se ha comentado al inicio de la memoria, una de las ventajas más importantes que tiene LabVIEW es la posibilidad de diseñar programas que sirvan para controlar instrumentos de manera remota sin necesidad de acceder a los paneles de cada equipo.

En general, cuando es necesario comunicarse con cualquier instrumento, se hace por medio de comandos SCPI. Esto implica tener que estar acompañado en cada momento del manual del instrumento en cuestión para buscar el comando asociado a cada función. Además, puede requerir una tarea de programación intensa repleta de código que difícilmente pueda ser entendida por otra persona que no esté familiarizada con este sistema de programación.

La solución que LabVIEW da para este tipo de tareas es el uso de Instruments Drivers (controladores de instrumentos). Mediante estos controladores el usuario puede olvidarse completamente de los comandos SCPI y de líneas de código para controlar un instrumento.

Normalmente, cada controlador incluye una biblioteca con ejemplos prácticos asociados a ese instrumento en cuestión, que pueden ser muy útiles para entender el funcionamiento de los bloques incluidos.

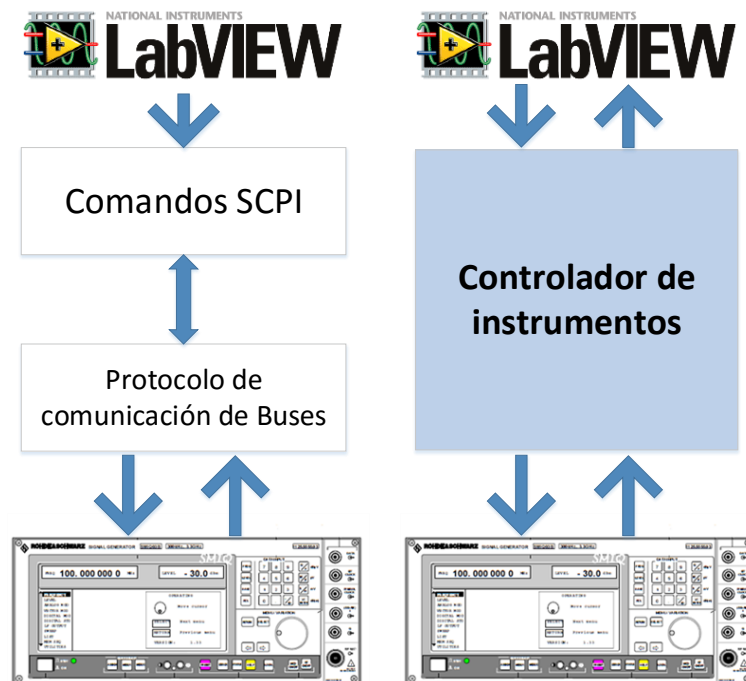


Figura 4-1 Uso de controladores de instrumentos

Existen tres tipos de controladores en LabVIEW:

- **Controladores Plug and Play:** Es un conjunto de funciones que se utilizan para el control y la comunicación con el instrumento. Cada función corresponde a una operación posible, como puede ser: la configuración, lectura, escritura o activación del instrumento. Estas funciones

aparecen dentro de la paleta de funciones de LabVIEW una vez instalado el controlador en la carpeta correspondiente.

Están desarrollados en LabVIEW y normalmente se puede descargar su código fuente en la web oficial del fabricante del instrumento.

- **Instrumentos virtuales intercambiables (IVI Drivers):** Son controladores más sofisticados y con más características que cuentan con un mayor rendimiento. Están diseñados para aplicaciones de prueba más complejas que requieran capacidad de intercambio, almacenamiento en caché o simulación de instrumentos.
- **Arquitectura de Software del instrumento virtual (VISA):** Es un paquete estándar para la configuración, programación y solución de problemas para sistemas que usen interfaces *GPIB*, *VXI*, *PXI*, *Serie*, *Ethernet* o *USB*.

NI-VISA es la implementación del fabricante National Instruments del estándar VISA I/O. Éste incluye algunas herramientas de interés, como el VISA Interactive Control, que sirven para monitorizar y comprobar el funcionamiento de la interfaz que se esté usando.

Para este proyecto se necesitan los controladores Plug and Play de cada instrumento y el NI-VISA que se puede obtener desde la página oficial de National Instruments.

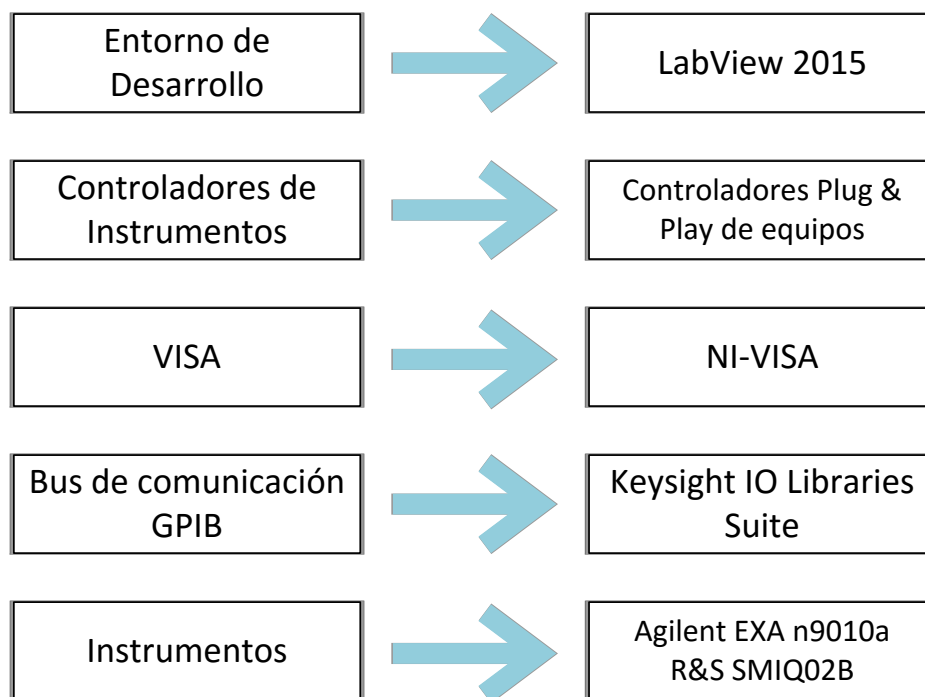


Figura 4-2 Esquema de capas para el control de instrumentos

4.1 Controladores Plug & Play para los equipos

Los controladores que se han utilizado para los instrumentos descritos anteriormente son los siguientes:

Agilent EXA n9010a	http://sine.ni.com/apps/utf8/niid_web_display.download_page?p_id_guid=3E3412EF63953891E0440003BA7CCD71
R&S SMIQ02b	https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_driver/rssmiq/rssmiq_lv60_3_1.zip

Tabla 4-1 Controladores Plug and Play de los equipos

La instalación de estos controladores se hace extrayendo el contenido dentro de la carpeta de *drivers* de LabVIEW que normalmente se encuentra en: *C:\Archivos de Programa\National Instruments\LabVIEW 2015\instr.lib*.

El aspecto de la carpeta *instr.lib* después de haber instalado estos dos drivers tiene que ser similar al de la Figura 4-3.

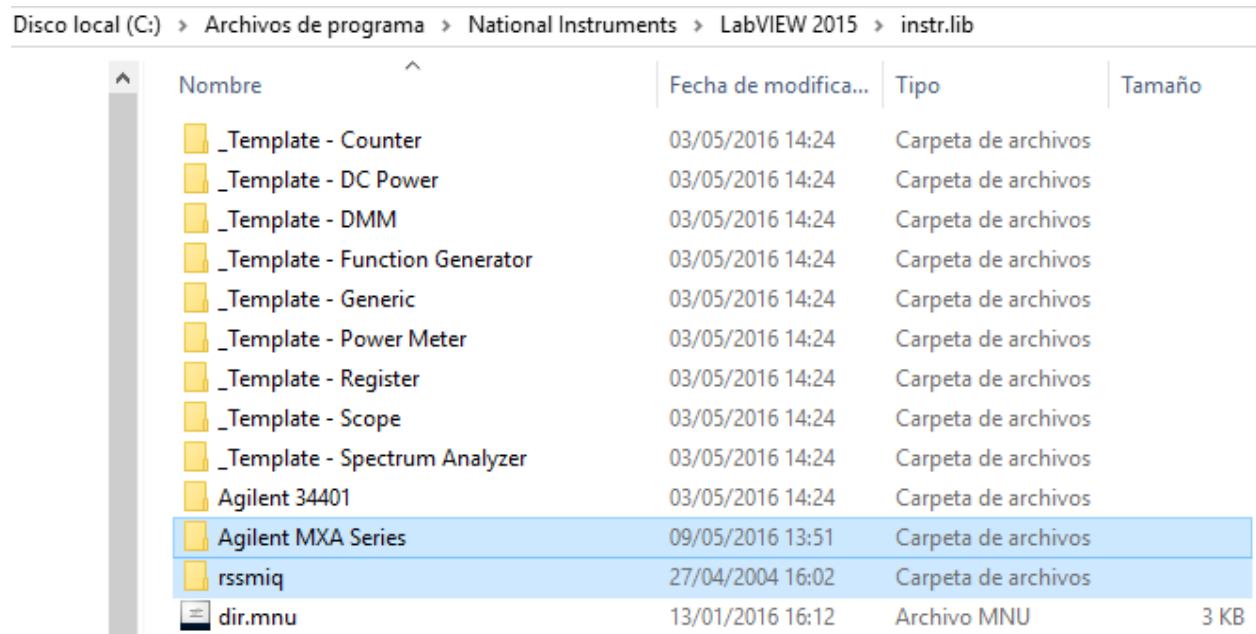


Figura 4-3 Carpeta de drivers de LabVIEW

Una vez que instalados los drivers en su carpeta, estarán disponibles para su uso en LabVIEW dentro de la paleta de funciones en el apartado **Instruments I/O > Instr Drivers**.

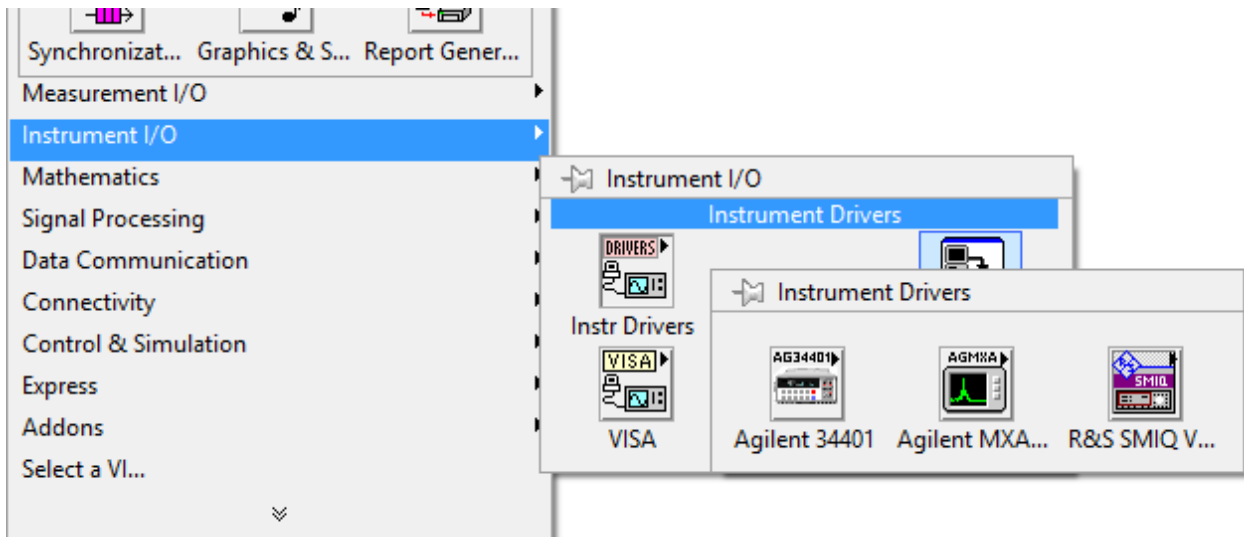


Figura 4-4 Controladores de instrumentos en la paleta de funciones

Se puede observar en la Figura 4-4 cómo aparecen los dos *drivers* instalados anteriormente, el del analizador Agilent EXA n9010a aparece bajo el nombre de Agilent MXA Series y el del generador de señales Rohde & Schwarz nombrado como R&S SMIQ Vector Signal Generator.

4.1.1 Controlador del Analizador Agilent EXA n9010a

A continuación, se presenta el árbol de bloques incluido en el controlador del instrumento Agilent EXA n90100a.

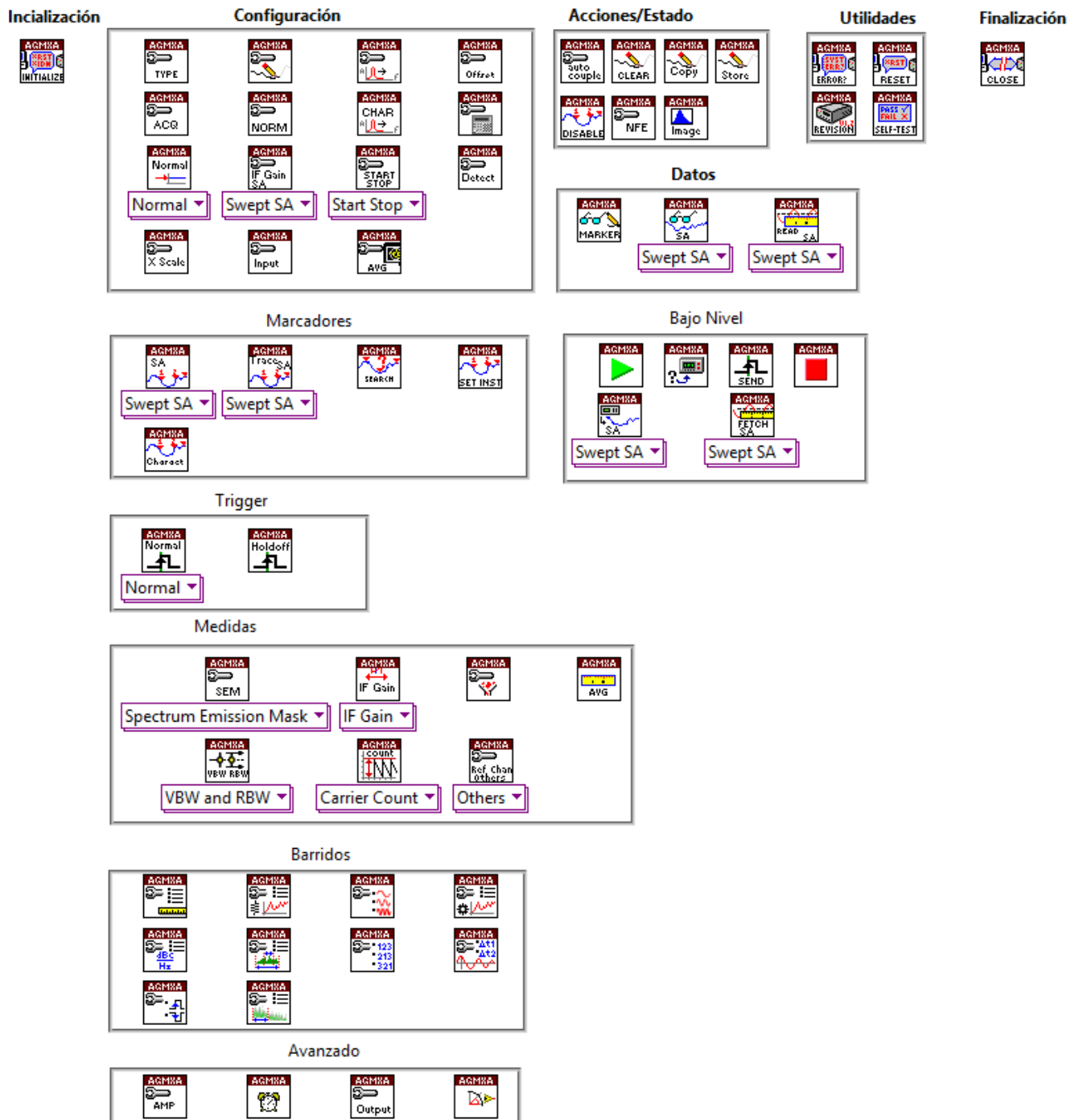


Figura 4-5 Árbol de bloques del controlador para el Agilent EXA n9010a

Se puede usar el buscador de ejemplos de LabVIEW para encontrar los ejemplos asociados a este equipo. Para lanzar el buscador, seleccionar **Help/Find Examples** en la barra de opciones.

4.1.2 Controlador del Generador Rohde & Schwarz SMIQ02b

El árbol de funciones que presenta este controlador es el siguiente:

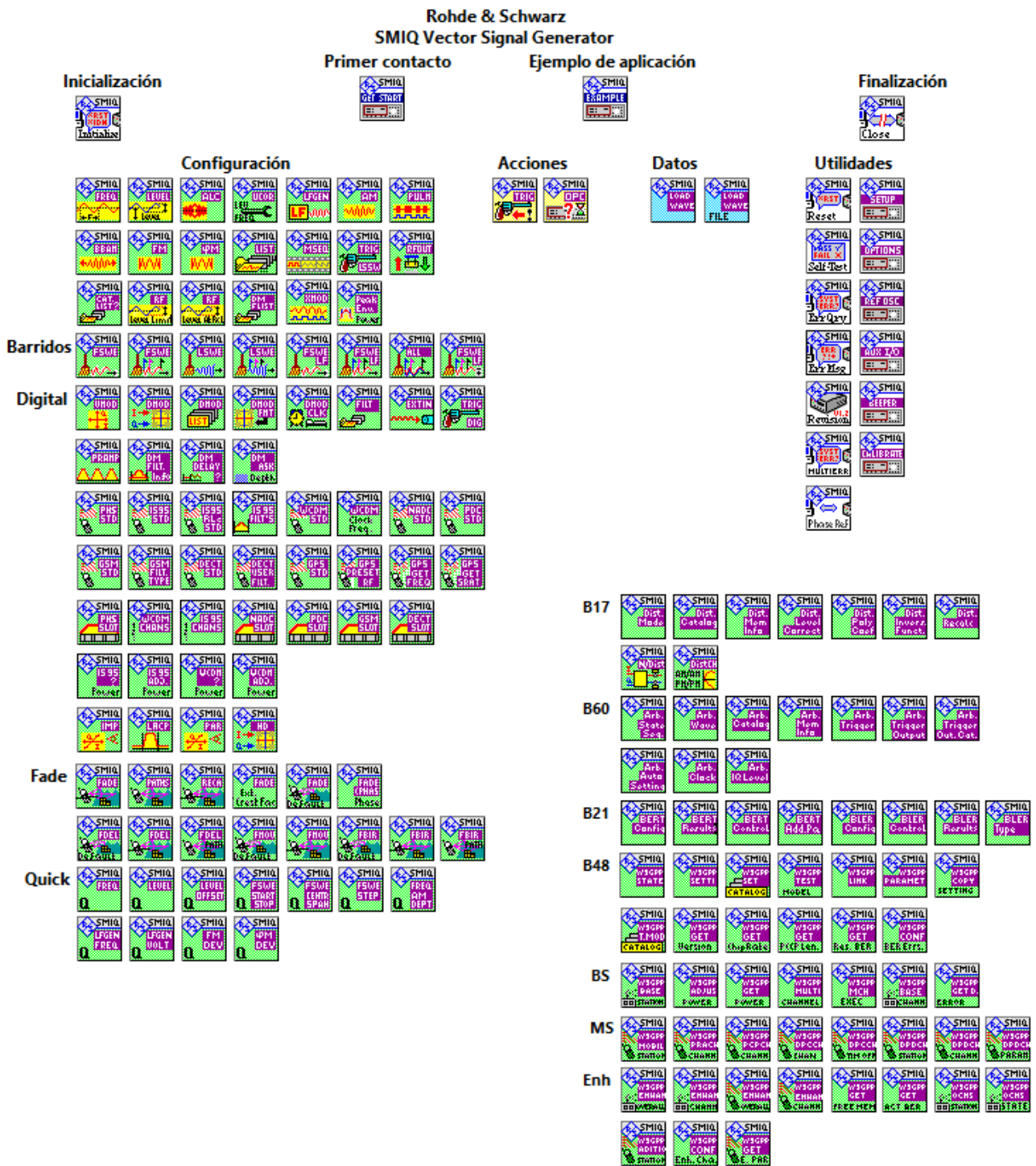


Figura 4-6 Árbol de bloques del controlador para el R&S SMIQ02b

Se pueden encontrar los ejemplos de este equipo en el buscador de ejemplos de LabVIEW.

Existe un error de diseño dentro de los *drivers* proporcionados por Rohde & Schwarz. Uno de los bloques internos del controlador contiene un *breakpoint* que detiene la ejecución y no permite que se

ejecute el programa completo cuando se está utilizando este controlador. Este punto de parada se encuentra dentro del instrumento virtual con nombre *RSSMIQ Utility Default Instrument Setup.vi*. El acceso a este VI se hace abriendo con LabVIEW el archivo *rssmiqu.llb* localizado dentro de la carpeta del controlador del SMIQ02b.

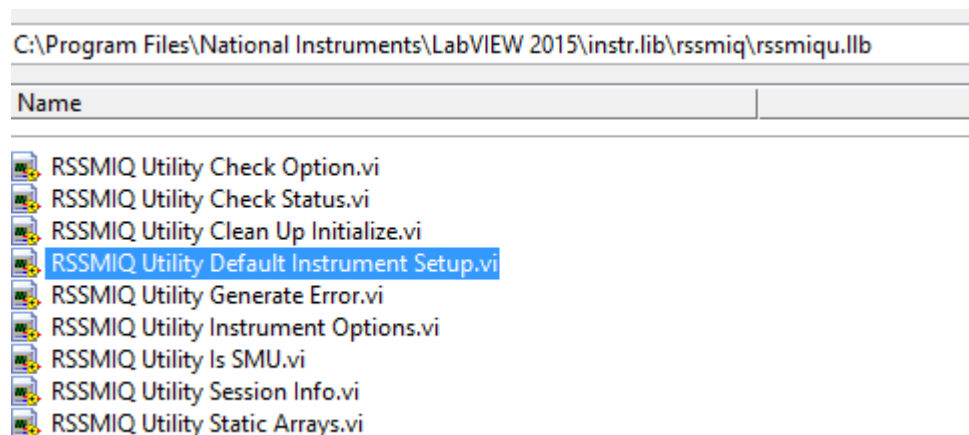


Figura 4-7 Contenido de rssmiqu.llb

Abriendo el Utility Default Instrument Setup y mostrando su panel frontal se puede observar cómo está colocado el breakpoint.



Figura 4-8 Breakpoint en controlador del SMIQ02b

Se elimina haciendo click en el punto coloreado en rojo con la herramienta de breakpoint. Después de eliminarlo es necesario guardar el instrumento virtual desde **File/Save**.

4.2 Configuración para la interfaz USB/GPIB Agilent 82357B

Para que LabVIEW pueda trabajar correctamente no bastaría solo con conectar el USB al ordenador, sino que habría que instalar los controladores del fabricante y hacer algunas configuraciones.

4.2.1 Keysight IO Libraries Suite

Lo primero sería instalar los *drivers* de la interfaz que se pueden descargar desde la propia web del fabricante (ahora con el nombre de Keysight) en esta URL: <http://www.keysight.com/en/pd-1985909/io-libraries-suite?cc=ES&lc=eng>. El paquete recibe el nombre de **IO Libraries Suite** e incluye, además de los controladores necesarios, un conjunto de aplicaciones para la monitorización de los dispositivos conectados a esta interfaz.

Una de estas aplicaciones es **Connection Expert**, que mediante una interfaz intuitiva proporciona

información detallada de cada uno de los equipos conectados a la interfaz en tiempo real. Es de utilidad para descubrir instrumentos y ver qué puertos están utilizando, además de poder enviar comandos SCPI entre los equipos y recibir respuesta de ellos.

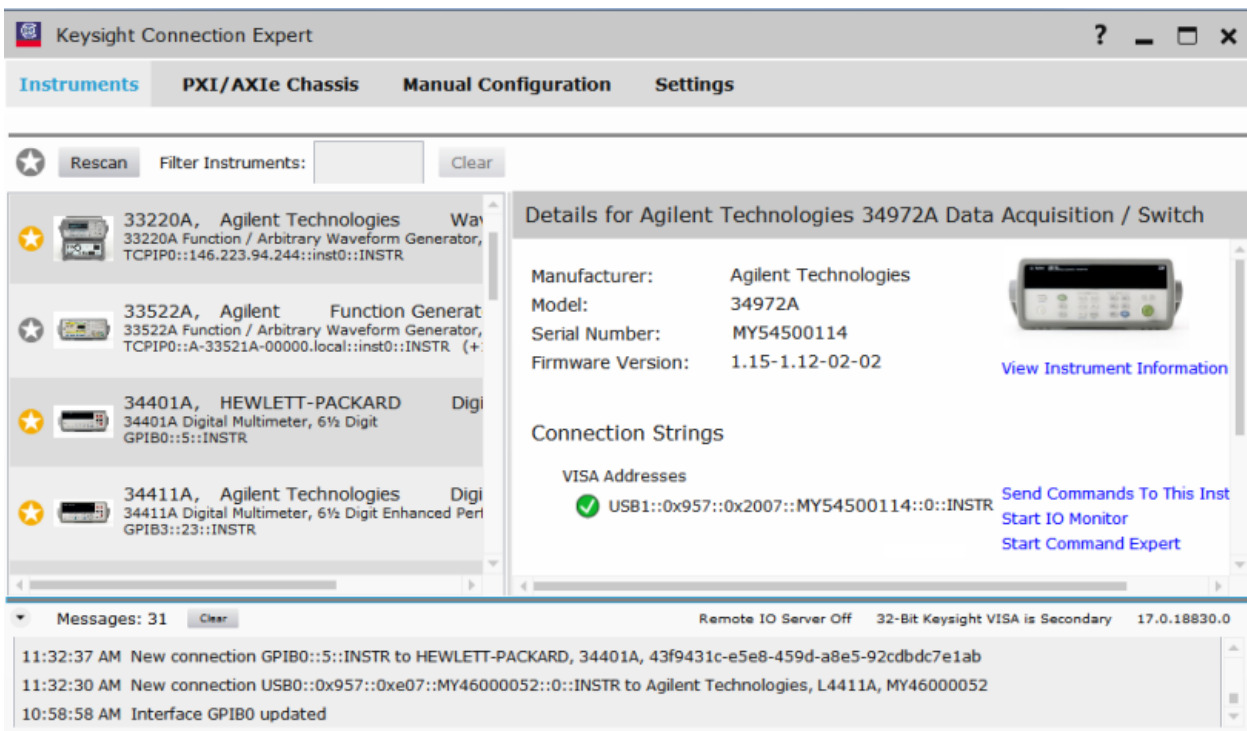


Figura 4-9 Keysight Connection Expert

Desde Connection Expert es necesario realizar una configuración para habilitar el uso del protocolo 488.2 desde programas externos, en este caso LabVIEW.

1. Desde la barra **Settings** seleccionar **Options/Keysight 488 Options**.
2. Marcar la opción **Enable Keysight GPIB cards for 488 programs**.

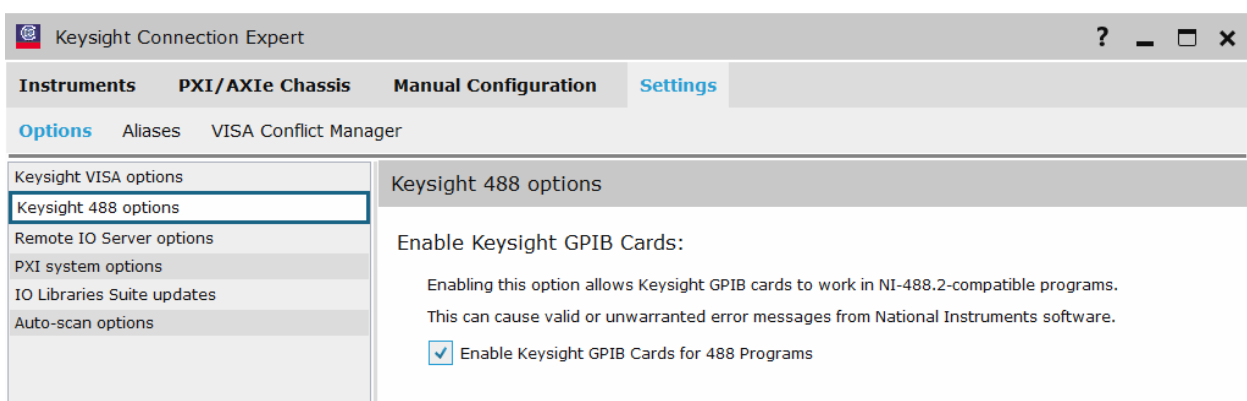


Figura 4-10 Configuración 488 en Connection Expert

4.2.1.1 Activación de NiVisaTulip.dll

Para que el controlador GPIB incluido en el paquete NI-VISA y el controlador de Keysight puedan trabajar juntos en el mismo sistema es necesario activar este archivo .dll.

1. A partir de LabVIEW abrir el Measurement & Automation Explorer (MAX) desde la barra **Tools/ Measurement & Automation Explorer**.
2. Dentro de MAX seleccionar **Tools/NI VISA > Visa Options**.
3. En el apartado **Passports** marcar la opción **NiVisaTulip.dll – NI-VISA Passport for Tulip**.
4. Guardar la configuración con el botón **Save**.

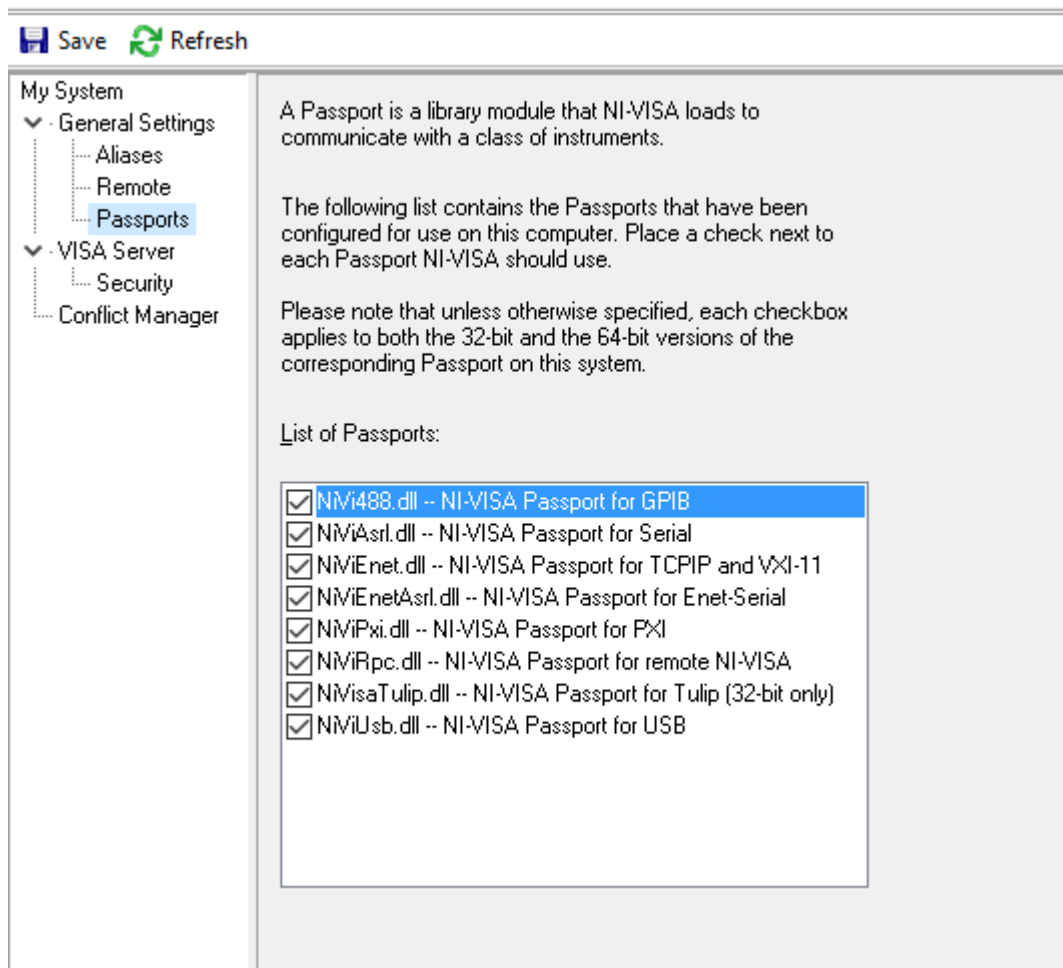


Figura 4-11 Activación del fichero NiVisaTulip.dll en MAX

4.3 Ejemplo práctico del uso de controladores

Para conocer de primera mano cómo funcionan los controladores dentro de LabVIEW se propone el siguiente ejemplo sencillo que consiste en la lectura de un barrido en frecuencia en el analizador n9010a.

Para la conexión con cualquier instrumento es necesario indicarle al programa que número de puerto GPIB está utilizando ese equipo. Para ello, los controladores incluyen un bloque de inicialización que recibe como parámetro de entrada la dirección del puerto perteneciente al equipo.



Figura 4-12 Bloque de inicialización

Para añadir el puerto bastaría con hacer click derecho con la herramienta de cableado sobre la entrada VISA resource name y elegir **Create/Control**. De esta forma se añadirá un bloque de control en el panel frontal donde se puede seleccionar uno de los puertos GPIB que se hayan detectado en ese momento.

Existe también un bloque de cierre para cerrar la sesión de ese puerto GPIB. El uso de estos dos bloques es obligatorio para la comunicación de LabVIEW con los equipos.

La línea VISA resource name va conectada a todos los bloques de ese instrumento desde la inicialización hasta el cierre, ya que todos necesitan recibir el puerto GPIB donde se va a realizar la acción.

Para la notificación de errores se dispone de una línea error que va recogiendo los posibles errores que se producen en cada bloque mostrándolos al final. El uso de esta línea no es necesario, pero resulta de interés para depurar el programa sin tener que echar un vistazo a los equipos.

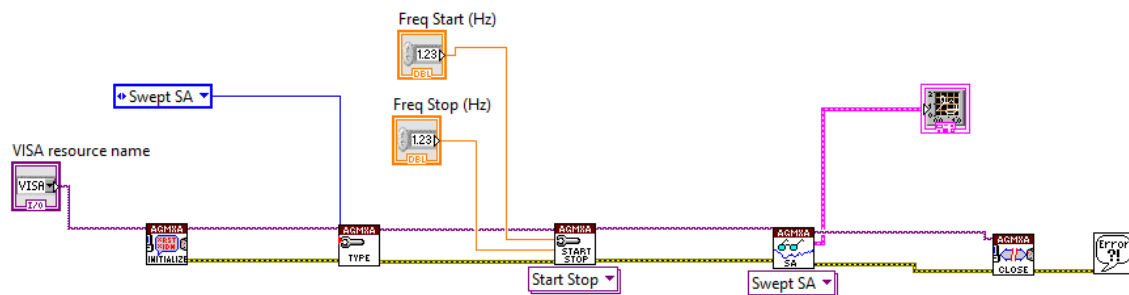


Figura 4-13 Diagrama de bloques de ejemplo para el control de instrumentos

La Figura 4-13 muestra el diagrama de bloques de un instrumento virtual que consiste en hacer un barrido en el analizador de espectro y mostrar los datos gráficos en el panel frontal de LabVIEW.

Entre los bloques de inicialización y cierre están los elementos que configuran el analizador de espectro y la recepción de los datos. Sobre estos bloques se entrará en más profundidad en los apartados siguientes.

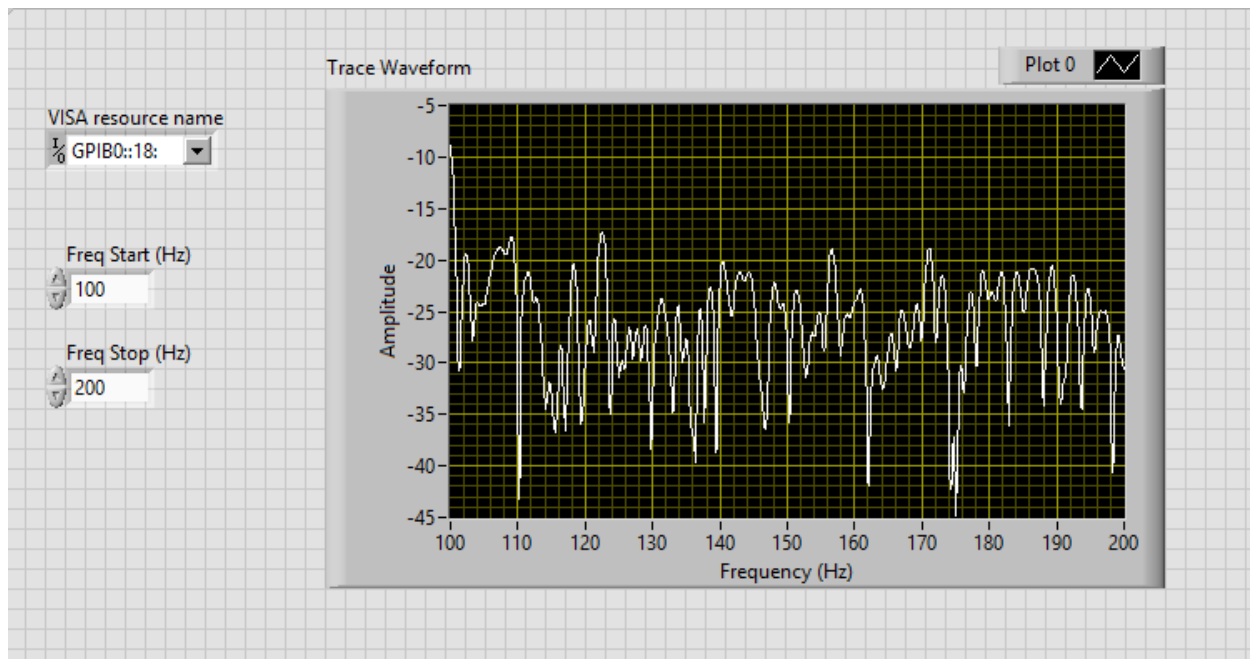


Figura 4-14 Panel Frontal de ejemplo para el control de instrumentos

En el panel frontal de la Figura 4-14 se configura el puerto al que pertenece el analizador y se introducen las frecuencias de inicio y fin para el barrido. El *display* muestra los datos obtenidos después de finalizar la ejecución si no ha habido ningún error.

Para una primera ejecución es interesante usar la opción de Highlight Execution para ver en tiempo real cómo se van ejecutando los bloques en los instrumentos en tiempo real. De esta forma la ejecución es más lenta pero puede servir para encontrar posibles fallos en el diseño del programa que se pueden pasar por alto con la opción de ejecución normal.

5 INTRODUCCIÓN TEÓRICA

5.1 Introducción al problema de la no linealidad

Se le llama sistema lineal a aquel que satisface el principio de proporcionalidad. Esto quiere decir, que si se le aplica una entrada x , el sistema obtendrá una salida proporcional igual a ax siendo a un factor constante real o imaginario que se multiplica a la señal. Es definitiva, la salida depende linealmente de la entrada. Esto implica que la salida de un sistema lineal contenga únicamente las frecuencias que forman parte de la señal de entrada.

El amplificador es uno de los elementos más importantes que componen los sistemas de radiocomunicación. Su diseño basado en transistores BJT o FET entre otros, hace que sea un dispositivo no lineal y presente los efectos característicos de estos sistemas. El comportamiento no lineal se hace notar a medida que se aumenta la potencia de trabajo de los equipos, a mayor potencia, mayor es el efecto de la no linealidad en el sistema.

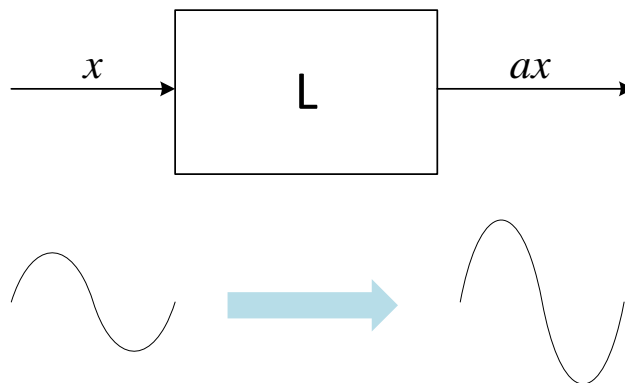


Figura 5-1 Esquema de un sistema lineal

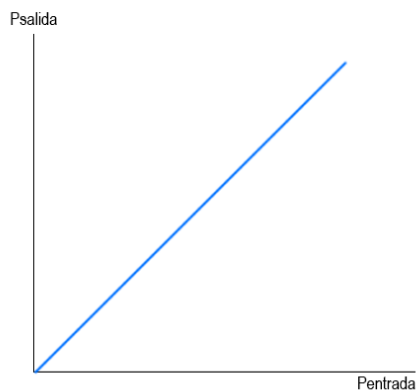


Figura 5-2 Gráfica de un sistema lineal

En el caso de un amplificador lineal, la ganancia del sistema es constante para cualquier valor de la potencia de entrada. En el dominio de la frecuencia ocurre algo similar, ante un tono de entrada se obtiene un tono de salida a la misma frecuencia, pero de potencia proporcional a la de la señal introducida en el sistema.

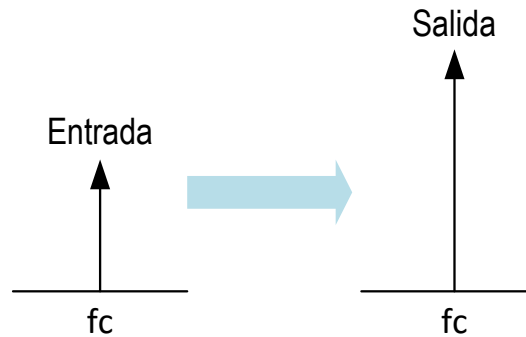


Figura 5-3 Comportamiento en frecuencia de un sistema lineal

De la misma forma, al introducir dos tonos con diferente frecuencia, a la salida se obtienen los mismos tonos con mayor potencia.

La realidad es bien distinta, los sistemas reales son no lineales, es decir, el resultado a la salida no es proporcional a la entrada. Además de eso, la señal obtenida a la salida contendrá un conjunto de componentes no deseadas en diferentes frecuencias que deben estudiarse para caracterizar el sistema. Si se introduce un tono, a la salida se añadirían otros tonos en frecuencias distintas. La relación entre la entrada x y las componentes que se añaden a la salida y tienen la siguiente expresión polinómica:

$$y = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots$$

Observando la ecuación, se puede ver cómo aparecen nuevos componentes de continua que se multiplican a unos términos de distorsión que son a su vez potencias consecutivas de la señal de entrada.

A este efecto se le conoce como distorsión y se manifiesta con la aparición de **distorsión armónica** y **productos de intermodulación**.

5.1.1 Distorsión armónica

Si a la entrada de un sistema no lineal, como por ejemplo un amplificador, se introduce un tono con frecuencia central f_0 , a la salida se obtiene, además del tono amplificado, otros tonos de distinta frecuencia en $2f_0, 3f_0, 4f_0 \dots$. Estas componentes reciben el nombre de **armónicos**.

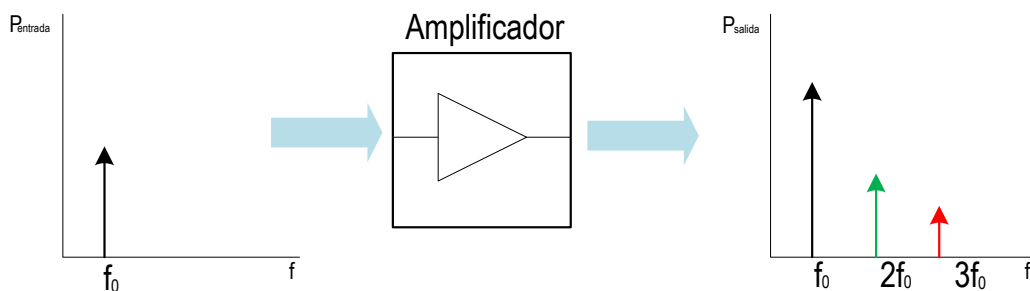


Figura 5-4 Distorsión armónica a la salida de un amplificador

La aparición de armónicos modifica la señal deseada y posiblemente estorben en el propósito del problema que se quiere solucionar. La eliminación de estas componentes puede realizarse mediante el uso de filtros para que solo dejen pasar el tono fundamental que se desee obtener a la salida.

5.1.2 Productos de intermodulación

Los productos de intermodulación aparecen al introducir dos o más señales en un dispositivo no lineal, como es el caso de los amplificadores de RF. Además de las componentes de entrada y los armónicos, aparecen también unas componentes de mezcla.

Si se introducen dos tonos de frecuencia f_1 y f_2 en el sistema no lineal, a su salida se producirán los productos de intermodulación en las frecuencias suma y diferencia de múltiplos de las frecuencias originales:

$$f_{int} = af_1 \pm bf_2$$

Siendo a y b números enteros mayores que uno, ya que en los casos donde cualquiera de estos factores es igual a cero se produce la distorsión armónica comentada anteriormente.

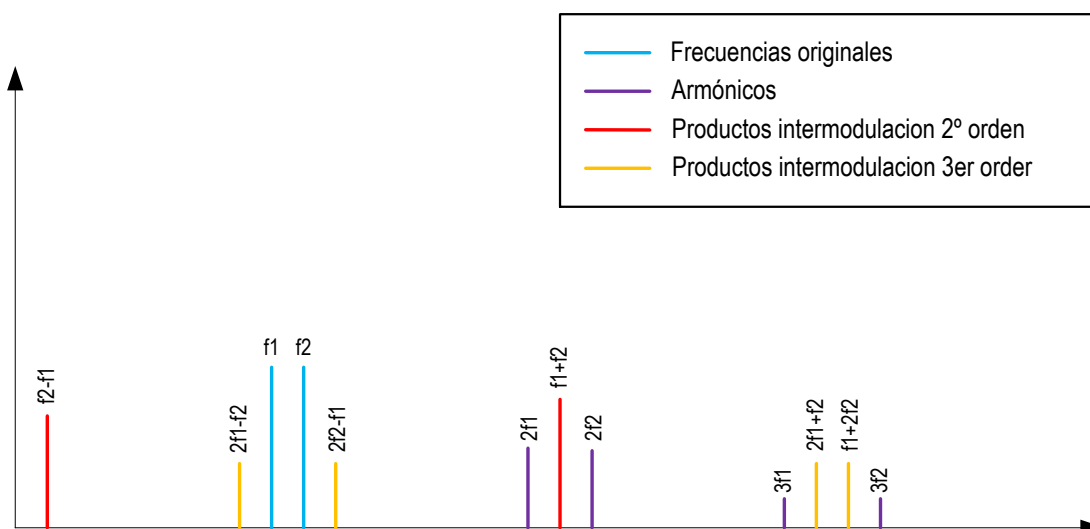


Figura 5-5 Productos de intermodulación

Como se puede observar en la Figura 5-5 los productos de intermodulación de segundo orden y los armónicos aparecen alejados de las frecuencias de interés, por lo tanto, no suelen causar muchos problemas ya que pueden ser filtrados en el caso de que hiciera falta. Por otro lado, los productos de intermodulación de tercer orden $2f_1 - f_2$ y $2f_2 - f_1$ sí que pueden acarrear problemas por estar muy cercanos a las frecuencias de interés f_1 y f_2 y su filtrado puede llegar a ser complicado.

En una señal multicanal donde existan muchas más señales con diferentes frecuencias, los productos de intermodulación pueden caer encima de las frecuencias de interés resultando imposible su filtrado. Este tipo de problemas suele causar dificultades en la práctica a la hora de obtener un resultado deseado.

5.1.3 Punto de compresión de 1dB

Los amplificadores presentan una zona de comportamiento lineal en el que la salida es proporcional a la entrada multiplicada por la ganancia G del amplificador en unidades naturales. A medida que aumenta la potencia, el amplificador empieza a saturar y llega un momento en el que la potencia de salida no aumenta proporcionalmente como lo hacía al principio, presentando así un comportamiento no lineal.

Para caracterizar los amplificadores uno de los parámetros que se suele determinar es el llamado punto de compresión de 1 dB. Este parámetro se define como aquel punto en el que la potencia de entrada hace que la potencia de salida que se obtenga sea de 1 dB inferior a la que se obtendría en el caso de que el amplificador tuviera un comportamiento lineal.

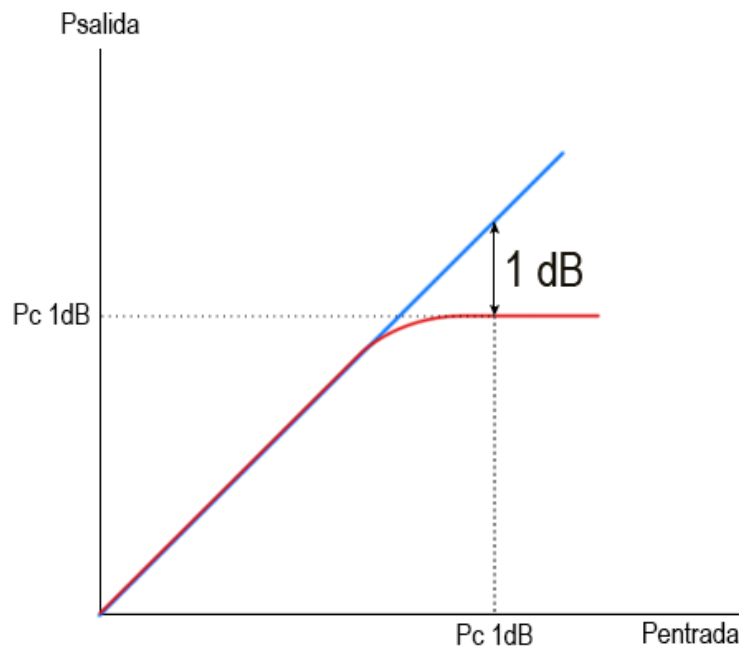


Figura 5-6 Punto de compresión de 1 dB

Este parámetro puede definirse tanto a la entrada como a la salida, existiendo el punto de compresión de 1 dB a la entrada y punto de compresión de 1 dB a la salida.

5.1.4 Punto de intercepto de tercer orden

Otro parámetro importante que sirve para caracterizar los amplificadores es el punto de intercepto de tercer orden. Se define como el punto donde el producto de intermodulación de tercer orden corta con la recta de potencia que tendría el amplificador si fuera lineal.

Normalmente este punto se localiza fuera del rango de trabajo del amplificador, así que debe obtenerse extrapolando el producto de intermodulación de tercer orden y la recta del amplificador ideal.

El punto de intercepto de tercer orden se encuentra habitualmente por encima del punto de compresión de 1 dB y muy alejado del punto de saturación del amplificador.

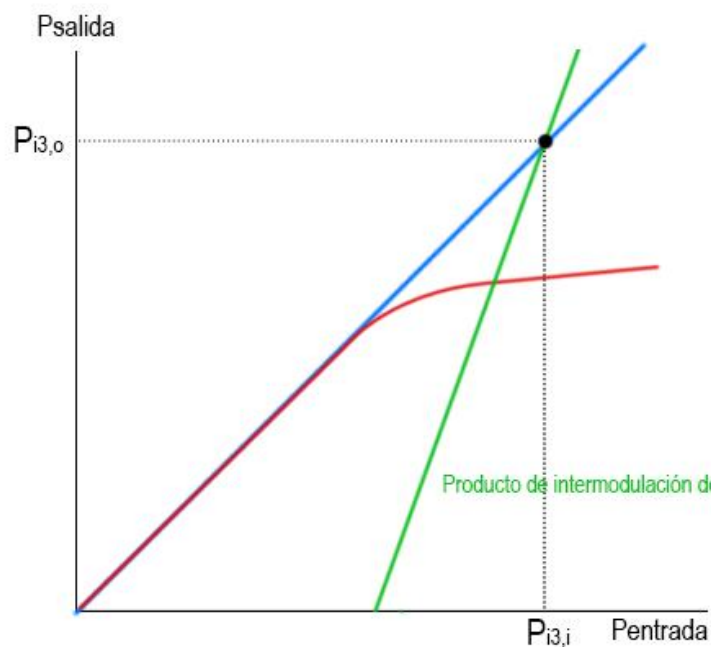


Figura 5-7 Punto de intercepto de tercer orden

Al igual que en el punto de compresión de 1 dB existe una componente de entrada y otra de salida para el punto de intercepto de tercer orden.

6 DESARROLLO

6.1 Ejercicio 1: Comportamiento del amplificador de potencia, prueba de un tono

En este apartado se realizará un instrumento virtual en LabVIEW con el fin de caracterizar a un amplificador real en el laboratorio.

Para ello, se van a estudiar los armónicos de hasta quinto orden que aparecen a la salida del amplificador. A su entrada, se introducirá un tono que puede tomar diferentes valores de potencia y para el que se usará una frecuencia fija a configurar por el usuario. Tras la ejecución, el programa generará una gráfica donde se observará el comportamiento de cada armónico a la salida del amplificador a medida que aumenta la potencia en la entrada de este.

El amplificador que se va utilizar para caracterizar sus propiedades no lineales es el modelo ZJL-6G+ del fabricante Mini-Circuits. Sus principales características son:

- Amplificador RF de baja potencia.
- Admite frecuencias que van desde 20 MHz hasta 6000 MHz.
- Valor típico de ganancia de 13 dB con un margen de ± 1.6 dB.
- 12 V de alimentación

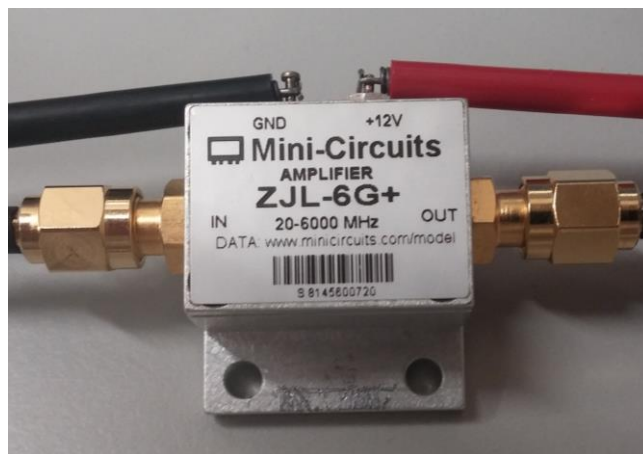


Figura 6-1 Amplificador Mini-Circuits ZJL-6G+

6.1.1 Esquema de montaje

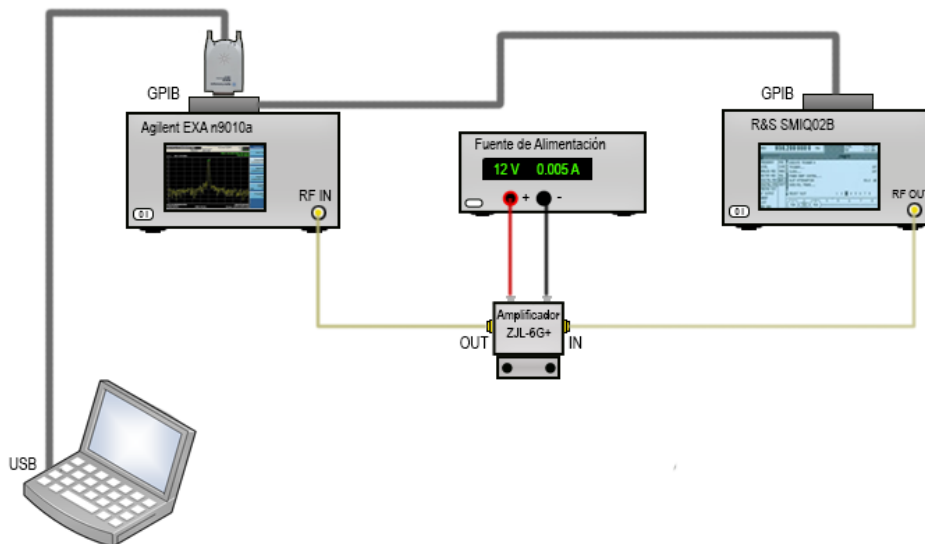


Figura 6-2 Esquema de montaje para el ejercicio 1

Las conexiones del montaje son las siguientes:

- **Ordenador – Analizador Agilent EXA n9010a:** Para esta conexión se utiliza la interfaz USB-GPIB Agilent 82357B descrita al principio de la memoria. La elección de conectar el ordenador al analizador es debida a la cercanía del ordenador al analizador en el laboratorio, por lo tanto, el ordenador puede ir conectado al analizador o al generador indistintamente ya que ambos irán conectados entre sí.
- **Analizador Agilent EXA n9010a – Generador R&S SMIQ02B:** Ambos equipos se conectan entre sí mediante un cable GPIB estándar.
- **Amplificador ZJL-6G+ - Fuente de alimentación:** Para que el amplificador funcione necesita estar alimentado por ser un dispositivo activo. Este amplificador se alimenta con un nivel de tensión de 12 V.
- **Generador R&S SMIQ02B - Amplificador ZJL-6G+:** La salida del generador, indicada como RF OUT (conector SMA hembra), se conecta a la entrada IN del amplificador (conector SMA hembra). Esta conexión se realiza a través de un cable coaxial con conectores tipo SMA macho en ambos extremos.
- **Amplificador ZJL-6G+ - Analizador Agilent EXA n9010a:** Mediante conectores SMA se conecta la salida del amplificador OUT con la entrada RF INPUT del analizador de espectro. De la misma forma, para esta conexión se usa un cable coaxial con conectores SMA macho en ambos extremos.

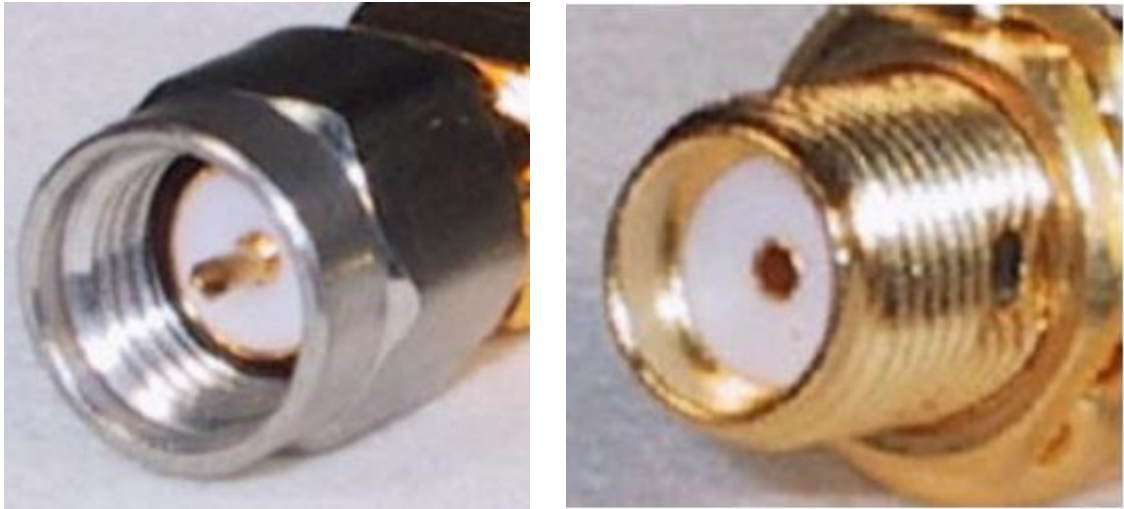


Figura 6-3 Conectores SMA macho (izquierda) y hembra (derecha)

6.1.2 Mediciones previas

Antes de comenzar con el desarrollo del problema, es interesante estudiar los valores de pérdidas en los cables y conectores utilizados para obtener unos resultados finales mucho más precisos a la hora de caracterizar al amplificador.

Tan solo es necesario realizar un montaje simple conectando únicamente el generador al analizador con el cable coaxial SMA como se muestra en la imagen:



Figura 6-4 Montaje para el cálculo de pérdidas en los cables

1. Generar un tono en el generador de señal con una frecuencia central de 600 MHz y una potencia igual a -25 dBm .
2. Configurar la frecuencia central del analizador de espectro a 600 MHz con un SPAN de 10 MHz para ver el tono.
3. El valor de las pérdidas en los cables para la componente fundamental vendrá determinado por la diferencia de potencia entre la señal generada en el generador y la señal obtenida en el analizador. Las pérdidas vienen dadas en decibelios dB.

$$L_{\text{cable}} = P_{\text{generador}} - P_{\text{analizador}}$$

4. Repetir los pasos 1,2 y 3 para los distintos armónicos, es decir, para frecuencias iguales a $2 \cdot 600 \text{ MHz}$, $3 \cdot 600 \text{ MHz}$, $4 \cdot 600 \text{ MHz}$ y apuntar sus valores.

Tras realizar esta medida en el laboratorio se han obtenido los siguientes resultados para las pérdidas en los cables:

Componente Fundamental	$L_{cable} = 0.6 \text{ dB}$
Segundo Armónico	$L_{cable} = 1.3 \text{ dB}$
Tercer Armónico	$L_{cable} = 1.6 \text{ dB}$
Cuarto Armónico	$L_{cable} = 1.8 \text{ dB}$
Quinto Armónico	$L_{cable} = 1.9 \text{ dB}$

Tabla 6-1 Resultados de pérdidas en los cables

6.1.3 Descripción funcional

El objetivo de este ejercicio es obtener los valores de pico (máximo valor de potencia) de un número determinado de armónicos para un rango de potencias variables a la entrada del amplificador y con frecuencia fundamental fija.

Se generará una señal de un tono en el generador de señal que tendrá una frecuencia fija y variará su nivel de potencia en un barrido, abarcando un rango de potencia que irá desde un valor inicial hasta un valor final con un incremento de paso, todo ello configurable por el usuario en el panel frontal. Para cada tono y nivel de potencia, se buscarán los armónicos en las frecuencias correspondientes en el analizador de espectro, que dependiendo del número de armónicos que se quiera buscar estarán en $f_{central}$, $2f_{central}$, $3f_{central}$, $4f_{central}$, $5f_{central}$, y se guardarán sus valores de potencia en una tabla. Una vez se hayan obtenido todos los resultados, se creará una gráfica donde el eje horizontal representará la potencia de entrada al amplificador y el eje vertical el valor de la potencia de salida del amplificador.

Para la realización de esta tarea, el programa en LabVIEW necesitará tener una estructura de la siguiente forma:

- **Bucle for** para el generador de señal. Este bucle se encargará de ir generando el tono a las diferentes potencias en el generador. Comenzará con un valor de potencia inicial e incrementará su valor en cada iteración hasta alcanzar el valor de potencia final.
- **Bucle for** dentro del bucle del generador para controlar el analizador de espectro. Su función es la de configurar los parámetros del analizador necesarios para buscar los armónicos del tono.

En cada iteración se buscará un armónico hasta llegar al número de armónicos a buscar que se recibe como parámetro de entrada.

Los parámetros configurables por el usuario que podrán ser introducidos desde el panel frontal del instrumento virtual son los siguientes:

Generador de señal

- **Frecuencia central del generador:** Es la frecuencia a la que se va a configurar en el generador de señal para realizar la medida.
- **Potencia Inicial:** Primera potencia que tomará el tono a generar en el generador de señal. Medida en dBm.
- **Potencia Final:** Último valor de potencia a generar. Medido en dBm.
- **Incremento de potencia:** Cantidad de potencia que se incrementa en el generador en cada repetición hasta llegar a la potencia final. Medida en dB.

Analizador de espectro

- **Frecuencia central del analizador:** Frecuencia de trabajo del analizador de espectro. En el caso que nos ocupa de la medida para un amplificador de potencia, su valor es igual al de la frecuencia central del generador, aunque se pone en otra variable independiente por si en otra ocasión se desea caracterizar un dispositivo que realice conversión de frecuencia, como es el caso de un mezclador.
- **Armónicos a buscar:** Numero de armónicos que se deseen estudiar. El valor máximo es de cinco armónicos.

Pérdidas en los cables

- **Pérdidas a la entrada:** Pérdidas en los cables en el trayecto que va del generador al amplificador
- **Pérdidas a la salida:** Pérdidas en los cables para la frecuencia fundamental en el trayecto que va del amplificador al analizador
- **Pérdidas en los armónicos:** Pérdidas en los cables para cada armónico en el trayecto del amplificador al analizador.

6.1.4 Diseño del panel frontal

En el panel frontal del instrumento virtual se deben incluir los bloques que permitan introducir valores de entrada para la configuración de la medida que se desea realizar. Además de los valores de entrada, se incluirá un *display* donde se mostrará la gráfica con los datos obtenidos en la medida.

El panel frontal se compone de los siguientes elementos:

- **Controles numéricos:** En estos controles se incluirán los valores de entrada para la frecuencia central del generador, potencia inicial, incremento de potencia, potencia final, frecuencia central del analizador, armónicos a buscar y pérdidas a la entrada.
- **Array de controles numéricos:** Aquí se incluirán los valores de las pérdidas a la salida para cada armónico que se calcularon previamente. Se ha decidido poner en forma de *array* para poder leer los valores de las pérdidas en los armónicos de manera independiente usando un índice desde el diagrama de bloques.
- **Menús seleccionables:** Se incluirán estos menús para seleccionar las unidades de la frecuencia central en el analizador y el generador.
- **Gráfica XY:** En ella se representará la gráfica final con los valores obtenidos tras la ejecución del programa.

Los bloques de frecuencia incluidos en los *drivers* del analizador de espectro reciben los datos a configurar en *Hz* mientras que los del generador de señal los recibe en *MHz*. Para hacer más sencilla la selección de las frecuencias centrales en ambos instrumentos, se incluyen los menús seleccionables comentados en este apartado. Estos menús representan varios valores constantes que pueden configurarse en las propiedades del elemento. De este modo, si por ejemplo se desea configurar en 1 *MHz* el analizador de espectro, al introducir un 1 en el control para la frecuencia central del analizador y seleccionar *MHz* en el menú, el valor del control se multiplicará por 10^6 en el diagrama de bloques haciendo que a la entrada del bloque el valor sea igual a 1000000 *Hz*. La configuración de estos menús se realiza en la pestaña **Edit Items** dentro de las propiedades del elemento (Click derecho > **Properties**)

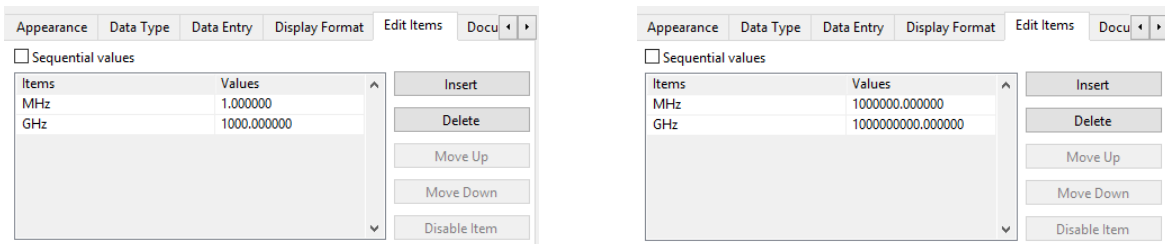


Figura 6-6 Configuración del menú de unidades para el generador (izquierda) y el analizador (derecha)

Para el control correspondiente al número de armónicos a buscar se puede restringir su rango de valores de 1 a 5 en la pestaña **Data Type** dentro de las propiedades del elemento. Así se evitaría la introducción de valores erróneos por parte del usuario.

El aspecto del panel frontal del instrumento una vez se han incluido y configurado todos los elementos nombrados es el siguiente:

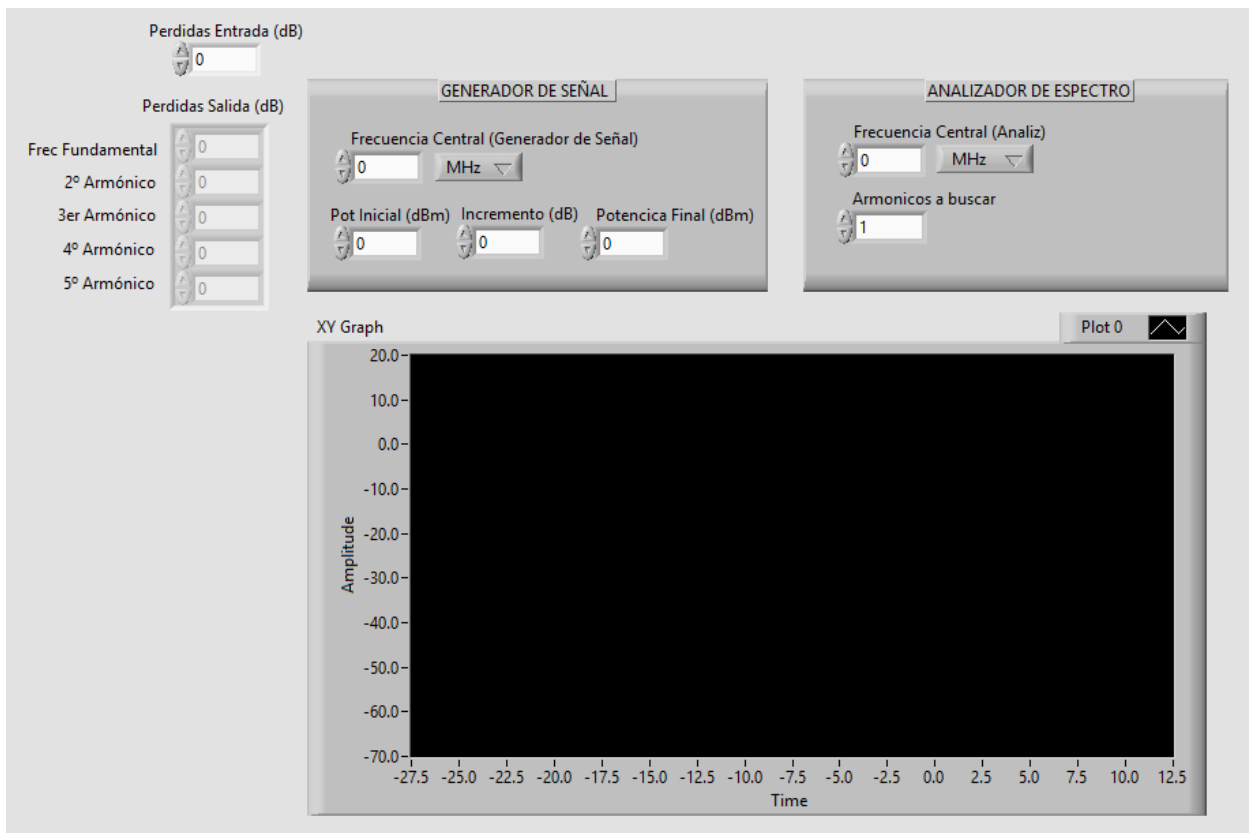


Figura 6-7 Panel frontal del ejercicio 1

Para crear un panel más vistoso e intuitivo se han incluido elementos de decoración y etiquetas para separar los controles que corresponden al analizador y al generador.

6.1.5 Diseño del diagrama de bloques

Como se venía adelantando, la estructura principal del código fuente del programa principal, que en este caso se corresponde con el diagrama de bloques, estará formado por dos bucles for anidados, uno para realizar la configuración del generador de señal y otro para el analizador de espectro.

Al estar usando dos instrumentos por control remoto mediante el bus GPIB, es necesario comenzar el programa con la inicialización de ambos equipos mediante los bloques que sus controladores incluyen para ello. Los dos bloques de inicialización deben recibir como entrada el puerto GPIB que identifica a cada equipo, en este caso los puertos por defecto son el **GPIB::28** para el generador de señal y el **GPIB::18** para el analizador de espectro. Estos puertos pueden variar y presentar otro número debido a que pueden ser modificados en cualquier momento, por lo tanto, es importante que el usuario revise el puerto asignado a cada equipo mediante el software **Keysight Connection Expert**.

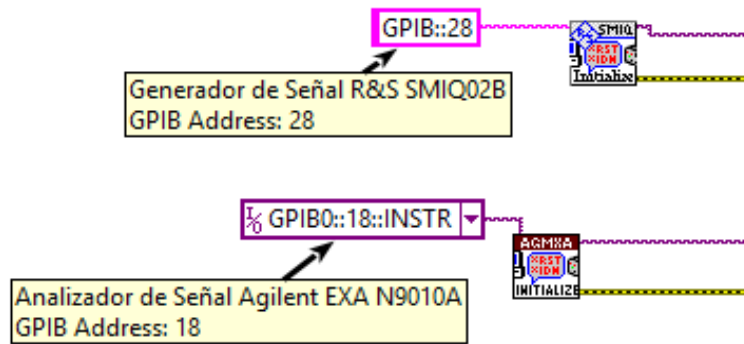


Figura 6-8 Inicialización de equipos

Antes de introducir los valores para las frecuencias centrales en los bloques correspondientes, es necesario multiplicarlas por los valores que se incluyeron en el menú seleccionable del panel frontal. Para ello habría que usar un bloque de multiplicación para aplicárselo al control de frecuencia central y al control que ha generado en el diagrama de bloques el Ring Menu creado en el panel frontal. De este modo quedaría así:

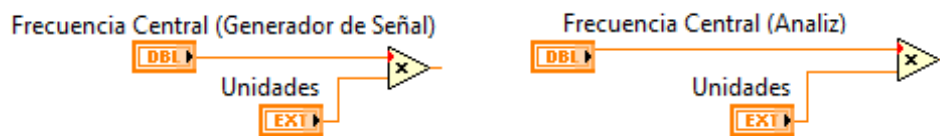


Figura 6-9 Adaptación de unidades para la frecuencia central

En el bucle correspondiente al generador, se va a generar un tono de diferentes potencias a la salida. De todos los parámetros de entrada que componen el panel frontal, no existe ninguno que indique directamente el número total de niveles de potencia que tomará el tono, sin embargo, se puede calcular mediante una sencilla operación matemática usando la potencia de entrada, potencia de salida e incremento de potencia.

$$\frac{P_{final} - P_{inicial}}{Incremento} + 1$$

Esta ecuación se puede implementar en LabVIEW de la siguiente manera:

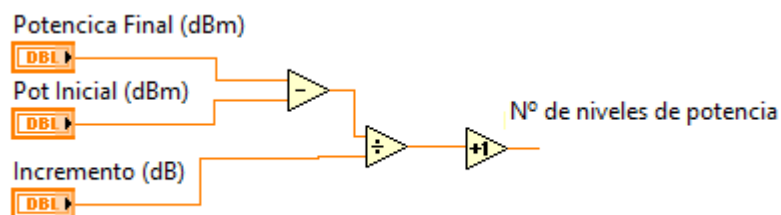


Figura 6-10 Cálculo del número de niveles de potencia del tono




El resultado de esta operación será el valor que se introduzca en el terminal N del bucle for asociado al generador. Este terminal indica el número de veces que se repite el bucle que en este caso coincide con el número de niveles de potencia a generar.

Tras estas configuraciones iniciales lo siguiente será el diseño de los bucles anidados de ambos equipos.

6.1.5.1 Bucle for del generador de señal

Este bucle es el encargado de generar los tonos con diferente potencia. El parámetro N indica el número de iteraciones o repeticiones que se dan y el parámetro i representa un conteo con el número de la iteración que se está realizando en cada momento.

Los bloques pertenecientes al controlador del R&S SMIQ02B que se van a incluir en este bucle son los siguientes:

- **Bloque de frecuencia** . Configura el valor de la frecuencia central para la señal RF de salida.
- **Bloque de nivel de potencia** . Configura el nivel de potencia de la señal RF de salida. Recibe como parámetro de entrada el control asociado a la frecuencia central del generador.
- **Bloque de salida de RF** . Activa o desactiva la salida del generador. Con un valor booleano de tipo True se activa y para un valor False se desactiva. En este caso se pone con valor True, de modo que se genere la señal de salida una vez se hayan configurado los valores de frecuencia y potencia previamente.

Para cada iteración se configura una potencia diferente con el fin de generar diferentes niveles para el tono, debido a esto, el valor de potencia a configurar debe seguir la siguiente ecuación que tiene que implementarse en el diagrama de bloques dentro de este bucle:

$$P_{entrada_i} = P_{inicial} + i \cdot Incremento$$

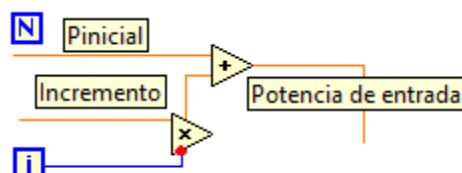


Figura 6-11 Configuración de la potencia de entrada al generador

Nótese que los cables que entran en los bloques de suma y multiplicación proceden de una ramificación de los controles potencia inicial e incremento que también aparecen en la Figura 6-10.

Sin incluir todavía la configuración del analizador, el aspecto que debe presentar el bucle del generador es este:

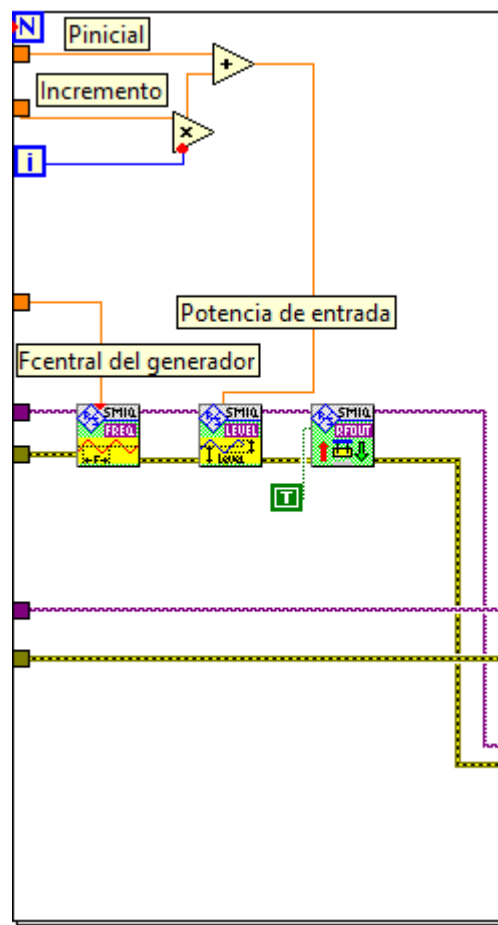











Figura 6-12 Bucle for perteneciente al generador de señal

6.1.5.2 Bucle for del analizador de espectro

Esta parte es la encargada de buscar los armónicos para cada tono generado en el SMIQ02B. Este bucle de tipo for se encuentra dentro del bucle asociado al generador de señal, de modo que cuando se ejecute el bloque de RF OUT del generador, se entrará en este bucle del analizador de espectro y se repetirá N de veces para buscar los N armónicos del pulso. El contador i del bucle será de gran ayuda a la hora de obtener valores necesarios para la configuración.

Para la configuración del analizador de espectro se incluirán estos bloques en el diagrama:

- **Frecuencia y span** . Configura la frecuencia donde se desea centrar el analizador y su valor del span. El valor de frecuencia a recibir es el perteneciente a la del armónico que se esté buscando en ese momento.
- **Configuración de nivel** . Este bloque es el que sirve para introducir el valor de referencia (Reference Level) con el que configurar el analizador en cada medida.

- **Configuración del barrido** . Con este bloque se pueden configurar las propiedades del barrido. En este caso el único parámetro de entrada que se va a introducir es el valor del ancho de banda de resolución (Resolution Bandwidth)
- **Configuración de la adquisición** . Bloque para configurar las propiedades de adquisición de la medida en el analizador. Se dejarán los valores por defecto que trae, por lo que no es necesario cambiar ninguna propiedad.
- **Inicialización de la medida** . Al ejecutarse este bloque se inicia el barrido en el analizador, es decir, con la configuración previamente introducida se comienza la medida.
- **Espera de adquisición** . Espera a que se complete el barrido en curso. Se incluye para asegurar que los valores que se tomarán después no se adquieran hasta que el barrido esté totalmente finalizado. Si no se incluye es posible que los valores que se obtengan de los armónicos sean incorrectos.
- **Configuración de marcador** . Con este bloque se crea el nuevo marcador del que se van a leer los valores de potencia.
- **Buscador de pico (Peak Search)** . Al ejecutarlo, el marcador se coloca automáticamente en el pico más alto de potencia que aparece en la pantalla.
- **Lectura del marcador** . Su salida entrega el valor numérico en el eje X e Y del marcador utilizado, donde el eje X representa a la frecuencia y el eje Y la potencia. En este caso lo que se leerá será el valor de pico del marcador creado anteriormente, este valor se corresponde con el valor de potencia del armónico (eje Y) que se esté buscando en la iteración del bucle for del analizador.

El parámetro que recibirá el bloque de frecuencia será igual al valor de la frecuencia de cada armónico.

$$f_{\text{armónico}_n} = n \cdot f_{\text{fundamental}}$$

donde $n = i + 1$, ya que el contador i del bucle comienza con el valor 0.

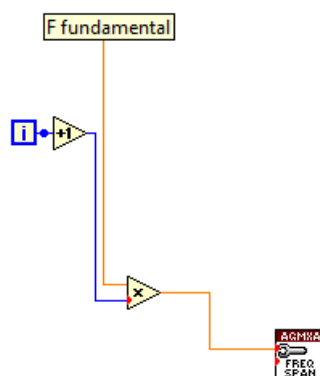


Figura 6-13 Configuración de la frecuencia en el analizador

Tras un estudio del comportamiento de los armónicos en el laboratorio, se ha determinado cómo deben ser los valores a configurar en el analizador para conseguir los valores correctos. Para la configuración del *Span* y el *Ancho de banda de resolución*, se ha observado que el valor que hace que la medida sea fiable es el resultado de dividir la frecuencia del tono fundamental entre cien.

$$\text{Span} = \text{Ancho de banda de resolución} = \frac{f_{\text{fundamental}}}{100}$$

El caso del *Reference Level* es algo más complejo ya que variará dependiendo del armónico. Inicialmente se determina un valor inicial resultado de sumar 25 dB a la potencia del tono que se está evaluando.

$$\text{RefLevel}_{\text{inicial}} = P_{\text{tono}_i} + 25 \text{ dB}$$

Partiendo de este nivel, los valores de *Reference Level* para cada armónico serán:

Primer armónico o tono fundamental	$\text{RefLevel} = \text{RefLevel}_{\text{inicial}}$
Segundo armónico	$\text{RefLevel} = \text{RefLevel}_{\text{inicial}} - 10 \text{ dB}$
Tercer armónico	$\text{RefLevel} = \text{RefLevel}_{\text{inicial}} - 15 \text{ dB}$
Cuarto armónico	$\text{RefLevel} = \text{RefLevel}_{\text{inicial}} - 20 \text{ dB}$
Quinto armónico	$\text{RefLevel} = \text{RefLevel}_{\text{inicial}} - 25 \text{ dB}$

Tabla 6-2 Reference Level para los armónicos

Para implementar esto en el programa, se creará una tabla con los valores 0, 10, 15, 20, 25. Cada elemento de esta tabla representa la diferencia que hay que restarle al valor inicial de *Reference Level* dependiendo del armónico. La variable de conteo *i* del bucle se encargará de realizar la elección de cada elemento de la tabla, así, se seleccionará el primer elemento cuando *i* valga 0, el segundo cuando *i* valga 1 y así sucesivamente. Por ejemplo, cuando *i* vale 1 se está buscando el segundo armónico, por lo que se cogería el segundo valor de la tabla que es igual a 10 y se sumaría al valor inicial de *Reference Level*. Por otro lado, el valor máximo que permite configurar el EXA n9010a para el *Reference Level* es igual a 23 dBm, por lo tanto, se incluye un bloque en el diagrama que limita este valor en caso de que se exceda.

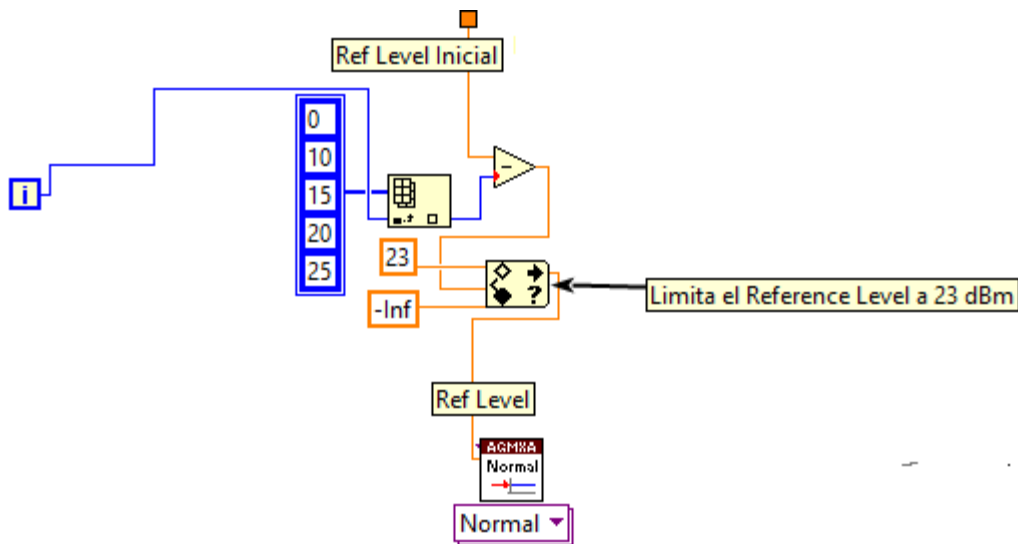


Figura 6-14 Configuración del Reference Level en el analizador

El bloque es el encargado de devolver el elemento i de la tabla, tal y como se ha comentado, y se encuentra bajo el nombre **Index Array** en la carpeta **Array** del menú de funciones. Para limitar los valores finales de *Reference Level* se usa el bloque que recibirá el límite superior, establecido en 23, y el límite inferior, que como en este caso no genera problema se ha configurado en menos infinito. Este bloque recibe el nombre de **In Range and Coerce** dentro de la carpeta **Comparison** de la paleta de funciones.

Los valores que se necesitan para caracterizar al amplificador son los de potencia de entrada y potencia de salida al amplificador. Hasta ahora en el generador se han generado tonos con diferente potencia y se han medido los valores de potencia de sus armónicos en el analizador. Estos valores generados y leídos no son los referidos a la entrada y la salida del amplificador, ya que sufren el efecto de las pérdidas en los cables. Con el fin de calcular los valores reales a la entrada y a la salida del amplificador, se van a realizar las siguientes operaciones dentro de este bucle.

$$P_{in_{amp}} = P_{out_{generador}} - Perdid_{entrada}$$

$$P_{out_{amp}} = P_{in_{analizador}} + Perdid_{salida}$$

Los valores de pérdidas a la salida se encuentran guardados en una tabla, el acceso a cada uno de sus elementos se haría de igual manera que en la tabla del *Reference Level* usando el bloque **Index Array** como se muestra en la siguiente imagen:

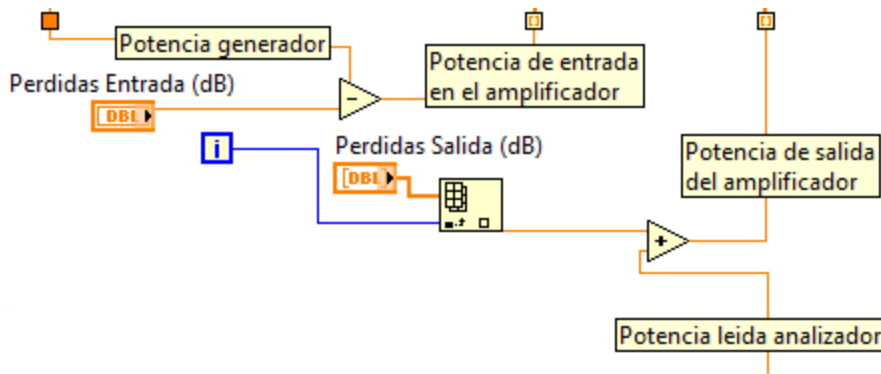


Figura 6-15 Calculo de la potencia a la entrada y a la salida del amplificador

Tras incluir las configuraciones mencionadas y todos los bloques necesarios para el analizador, el resultado del bucle quedaría como en la siguiente imagen:

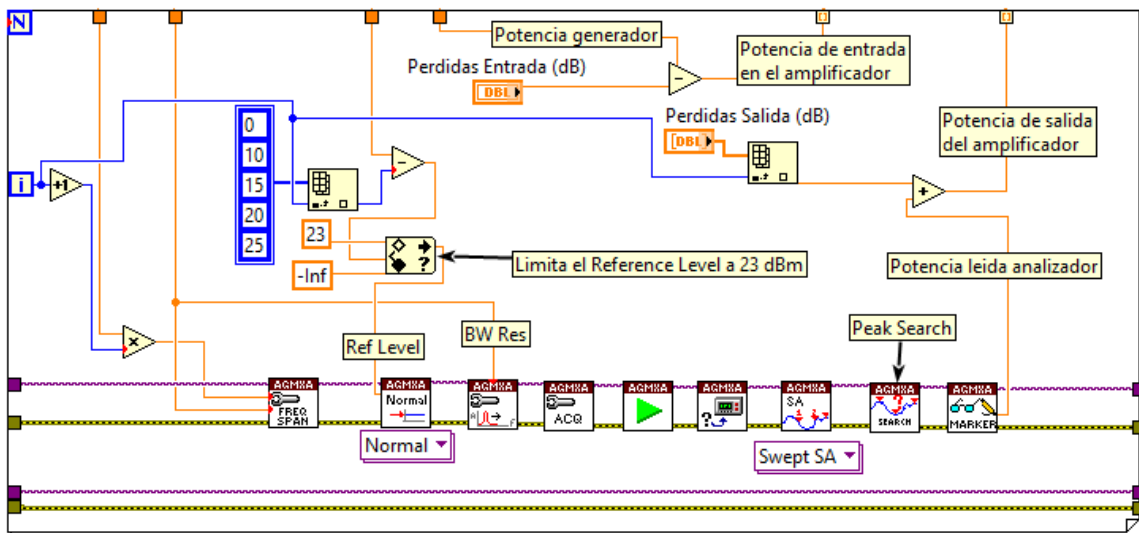


Figura 6-16 Bucle del analizador

6.1.5.3 Formato de los datos

En LabVIEW, siempre que se incluye un bucle de cualquier tipo, los datos internos pueden extraerse a la salida del mismo como una tabla o *array* con los diferentes valores que toma en cada repetición. En este caso, los datos etiquetados como potencia de entrada en el amplificador y potencia de salida del amplificador están saliendo de dos bucles (analizador y generador), por lo tanto, se obtiene una matriz en ambos casos.

Para el caso de la potencia de salida del amplificador, la matriz sería de esta forma:

$$\begin{pmatrix} PA_{11} & PA_{21} & PA_{31} & \dots & PA_{n1} \\ PA_{12} & PA_{22} & PA_{32} & \dots & PA_{n2} \\ PA_{13} & PA_{23} & PA_{33} & \dots & PA_{n3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ PA_{1m} & PA_{2m} & PA_{3m} & \dots & PA_{nm} \end{pmatrix}$$


donde PA_{nm} representa el valor de potencia del armónico n para el tono m .

Y para la potencia de entrada en el amplificador:

$$\begin{pmatrix} P_{in_1} & P_{in_1} & P_{in_1} & \dots & P_{in_1} \\ P_{in_2} & P_{in_2} & P_{in_2} & \dots & P_{in_2} \\ P_{in_3} & P_{in_3} & P_{in_3} & \dots & P_{in_3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{in_m} & P_{in_m} & P_{in_m} & \dots & P_{in_m} \end{pmatrix}$$

donde P_{in_m} es la potencia de entrada al amplificador del tono m .

En ambos casos, los valores están ordenados por columnas, sin embargo, el bloque de gráfica XY de LabVIEW necesita que los datos introducidos estén ordenados por fila. Es por ello que se aplicará el

bloque  **Transpose Matrix** para obtener ambas matrices transpuestas y así queden preparadas para introducirlas en la gráfica XY.

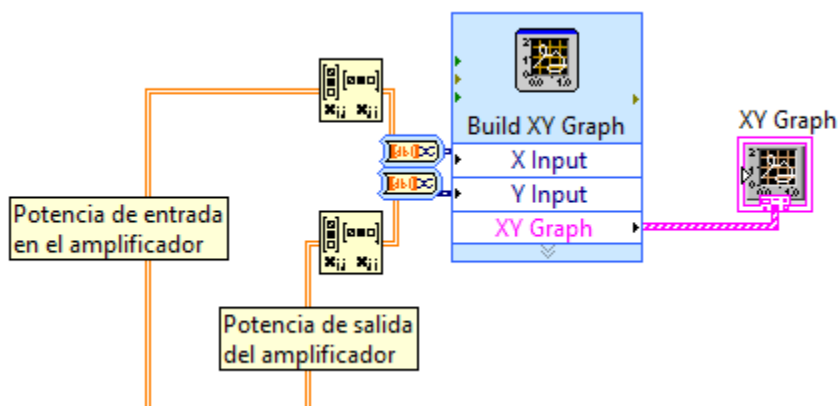


Figura 6-17 Configuración de la gráfica

6.1.5.4 Diagrama de bloques completo

A continuación, se muestra el resultado final del diagrama de bloques donde se incluye todo lo explicado en los subapartados anteriores.

Nótese que al final de los bucles se han introducido los bloques **Close** que cierran la conexión GPIB con ambos equipos, además de un bloque **RF OUT** del generador para parar la salida RF de este instrumento.

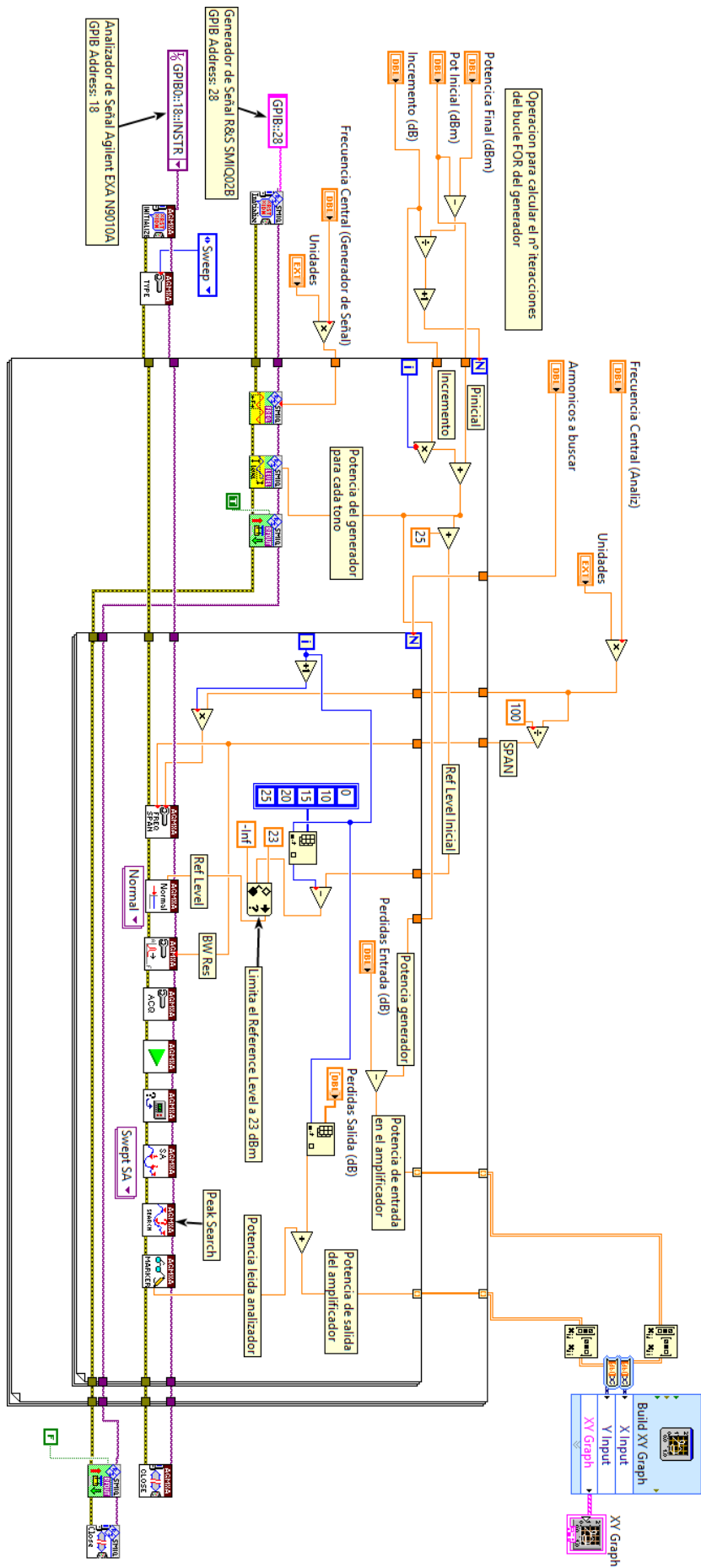


Figura 6-18 Diagrama de bloques completo del primer ejercicio

6.1.6 Ejecución y resultados

Para elegir los datos a introducir para configurar la medida, se han tenido en cuenta varios aspectos. La frecuencia máxima con la que se vaya a configurar el analizador no puede estar fuera del rango de frecuencia de trabajo del analizador (26,9 GHz). Esta frecuencia máxima se corresponderá con la del último armónico que se quiera estudiar, por lo tanto, se debe cumplir que:

$$9 \text{ kHz} < f_{\text{central analizador}} < \frac{26,9 \text{ GHz}}{n^{\circ}_{\text{armonicos}}}$$

Para la potencia final se elegirá un valor cercano a la potencia máxima que el amplificador acepta a su entrada (15 dBm).

- Frecuencia Central del generador: 600 MHz
- Potencia Inicial: -25 dBm
- Incremento: 0,1 dB
- Potencia Final: 12 dBm
- Frecuencia Central del analizador: 600 MHz
- Armónicos a buscar: 5

Con estos valores, en el generador se irán generando tonos de frecuencia igual a 600 MHz con potencias que irán desde los -25 dBm hasta los 12 dBm (cerca del máximo admitido por el amplificador) con incremento de 0,1 dB. Para cada tono se calculará la potencia de salida del amplificador de cada armónico, que estarán situados en las siguientes frecuencias:

Primer armónico	600 MHz
Segundo armónico	1200 MHz
Tercer armónico	1800 MHz
Cuarto armónico	2400 MHz
Quinto armónico	3000 MHz

Tabla 6-3 Frecuencia de los armónicos

Tras presionar el botón de ejecución en el panel frontal, el programa comenzará a funcionar y los equipos empezarán a configurarse secuencialmente. El proceso de ejecución durará varios minutos ya que el rango de potencia que se ha configurado es muy amplio.

Al finalizar se obtendrá el siguiente resultado en el panel frontal:

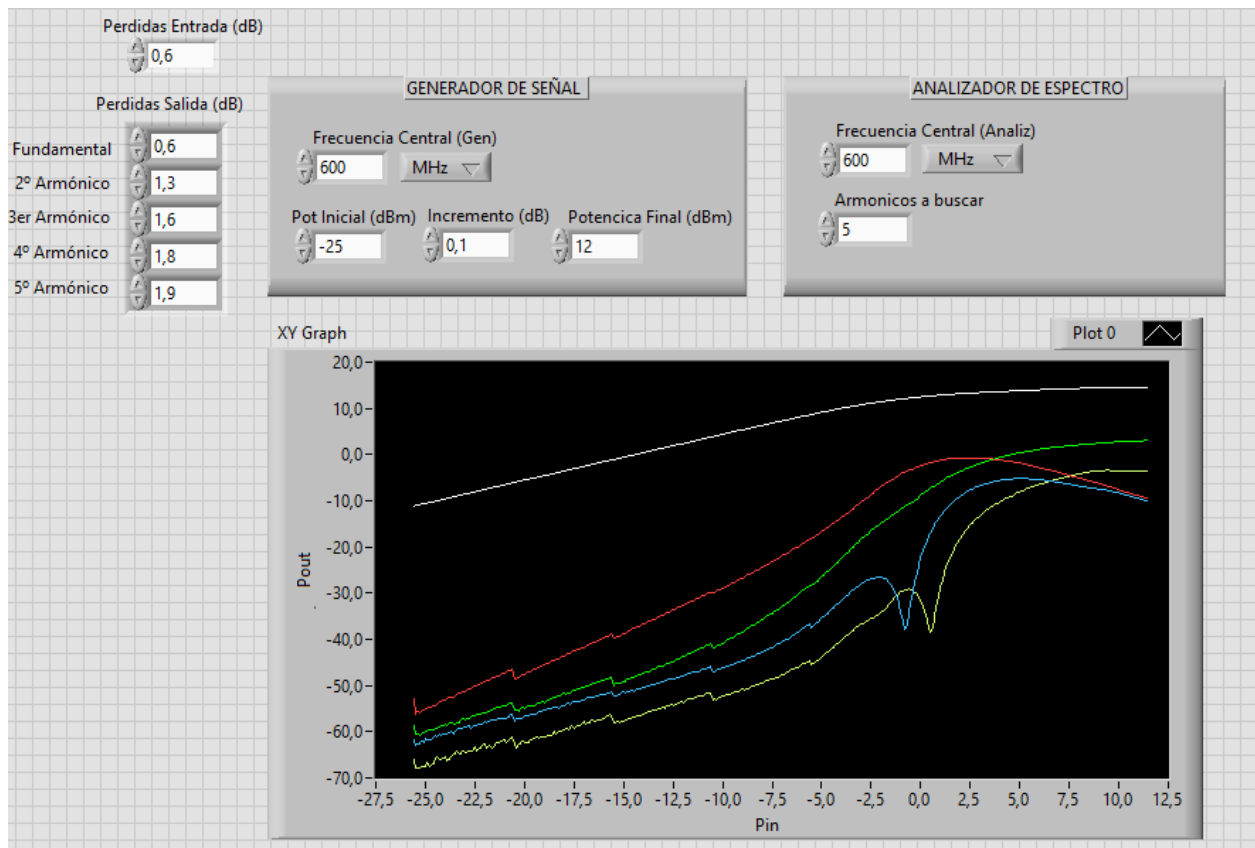


Figura 6-19 Resultados obtenidos para el primer ejercicio

Se obtienen los valores de la componente fundamental o primer armónico (trazo blanco), segundo armónico (trazo rojo), tercer armónico (trazo verde), cuarto armónico (trazo azul) y quinto armónico (trazo amarillo).

Observando la gráfica se ve cómo la componente fundamental queda amplificada a la salida del amplificador teniendo un comportamiento prácticamente lineal hasta un punto de potencia de entrada en el que la salida comienza a saturar tomando valores en torno a los 13 dBm que coinciden con lo especificado en la hoja de características del ZJL-6G+ (ver Anexo).

Los armónicos siguen un comportamiento similar, pero con potencias de salida menores. Para el cuarto y quinto armónico (trazo azul y amarillo), existe un punto en el que la potencia cae bruscamente para unos valores a la entrada cercanos a 0 dBm.

6.2 Ejercicio 2: Modulación ARB

Este ejercicio consiste en la realización de una modulación arbitraria de una portadora con una forma de onda definida previamente en MATLAB, que se llevará a cabo mediante el generador de señal R&S SMIQU02B. Al finalizar la modulación se representará su resultado en el dominio del tiempo en el analizador Agilent n9010a.

Para realizar una modulación ARB, el generador necesita cargar en su memoria RAM un archivo en formato .wv que contenga las componentes en fase y cuadratura de la forma de onda con la que se quiera modular la portadora. Normalmente, estas formas de onda se definen en MATLAB con la extensión .mat, donde se incluyen las muestras de la componente en fase en la parte real y las de la componente en cuadratura en la parte imaginaria. Por lo tanto, de alguna forma habría que convertir el archivo con extensión .mat a la extensión .wv aceptada por el generador.

$$x = x_I + jx_Q$$

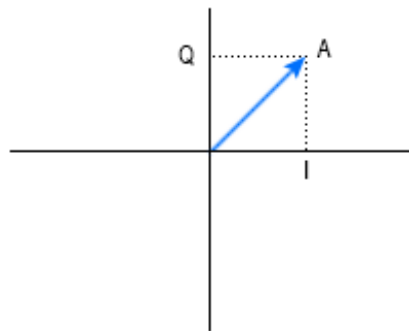


Figura 6-20 Componentes en fase y cuadratura

Tras la lectura de los datos de la forma de onda por parte del generador, se pasan por el DAC (convertidor analógico-digital) y se modulará a la frecuencia central y potencia de la portadora. La frecuencia de muestreo a la que se interpretan las muestras del fichero .mat corresponde al producto de la tasa de símbolos que se quiera conseguir y el número de muestras por símbolo que se generan al crear la forma de onda en MATLAB.

En el analizador se pretende medir los resultados de esta modulación en el dominio del tiempo. Para obtener esta medida es necesario usar otro modo de funcionamiento diferente al de analizador de espectro básico, ya que este solo permite realizar medidas en el dominio de la frecuencia y no en el del tiempo. El modo que se va a utilizar es el VXA por ser más rápido y sencillo a la hora de controlar remotamente.

Los controladores de LabVIEW que Agilent proporciona para el n9010a, que se han usado en el ejercicio anterior, no incluyen bloques para configurar el equipo en el modo VXA ni VSA. Debido a esto, para este ejercicio se van a utilizar comandos SCPI en el diagrama de bloques del instrumento virtual LabVIEW para configurar el modo VXA del analizador de espectro.

6.2.1 Esquema de montaje

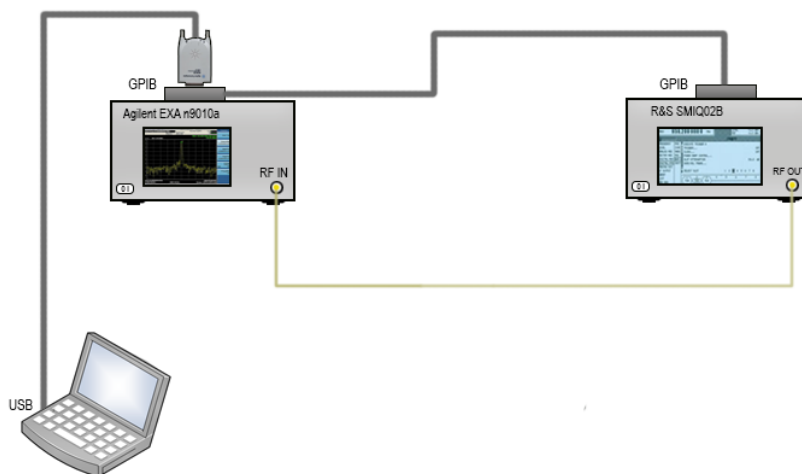


Figura 6-21 Esquema de montaje para el ejercicio 2

En este ejercicio el montaje es más simple que en el anterior, ya que solo se va a hacer uso del analizador de espectro, del generador de señal y de un ordenador configurado con LabVIEW que también tenga instalado MATLAB.

6.2.2 Descripción funcional

Para la realización de la modulación ARB, el generador recibirá de manera externa el archivo con la forma de onda en el formato correcto, además de la frecuencia de muestreo, frecuencia central y potencia que el usuario desee configurar para la portadora. Una vez generada la señal modulada, el analizador de espectro se configurará para hacer la medida en el tiempo correctamente y para enviar los datos a LabVIEW y que queden representados en gráficas. Los resultados se representarán en dos *displays*, uno mostrando los valores en el tiempo de la componente I de la señal modulada y la otra los valores de la componente Q.

En este ejercicio se utilizarán las estructuras de secuencia para cada instrumento, de esta forma se garantizará que el analizador comienza a realizar la medida después de que el generador haya acabado y enviado por su salida RF OUTPUT la señal modulada.

Los parámetros de entrada de este instrumento virtual son los siguientes:

Conversión de .mat a .wv

- **Ruta:** Aquí se indicará la dirección de la carpeta donde se localizará el archivo .mat a convertir. Dentro de esta carpeta se guardará el fichero resultante de la conversión.
- **Nombre del fichero .mat:** Nombre que el fichero .mat tiene dentro de la ruta.
- **Nombre del fichero .wv a convertir:** Nombre con el que se quiere guardar el nuevo fichero

convertido.

- **Tasa de muestreo:** Valor de la tasa de muestreo de la forma de onda incluida en el fichero de MATLAB. Es el producto de la tasa de símbolos que se desea conseguir para la modulación multiplicada por el número de muestras por símbolo o factor de sobremuestreo que se haya definido al crear la forma de onda en MATLAB. Este valor es necesario para la conversión.

Generador de señal

- **Frecuencia de la portadora:** Frecuencia a configurar en el generador para la portadora.
- **Potencia de la portadora:** Potencia a configurar en el generador para la portadora.
- **Ruta del archivo .wv:** Ruta donde se encuentra el archivo previamente convertido que se introducirá en el generador.

Analizador de espectro

- **Frecuencia central:** Frecuencia central a configurar en el analizador para realizar la medida. Se pone independientemente de la frecuencia configurada en el generador por si en el montaje hubiera un dispositivo conversor de frecuencia, como un mezclador, que modificara la frecuencia a la salida.
- **SPAN:** Valor correcto del SPAN para que pueda observarse correctamente la señal obtenida.
- **Tiempo a capturar:** Rango de tiempo que se desee capturar de la señal.
- **Rango de potencia:** Es el máximo valor de potencia que se espera de la señal.
- **Número de puntos:** Número de puntos a configurar para el barrido.

El funcionamiento del programa se resume en el siguiente diagrama de flujo:

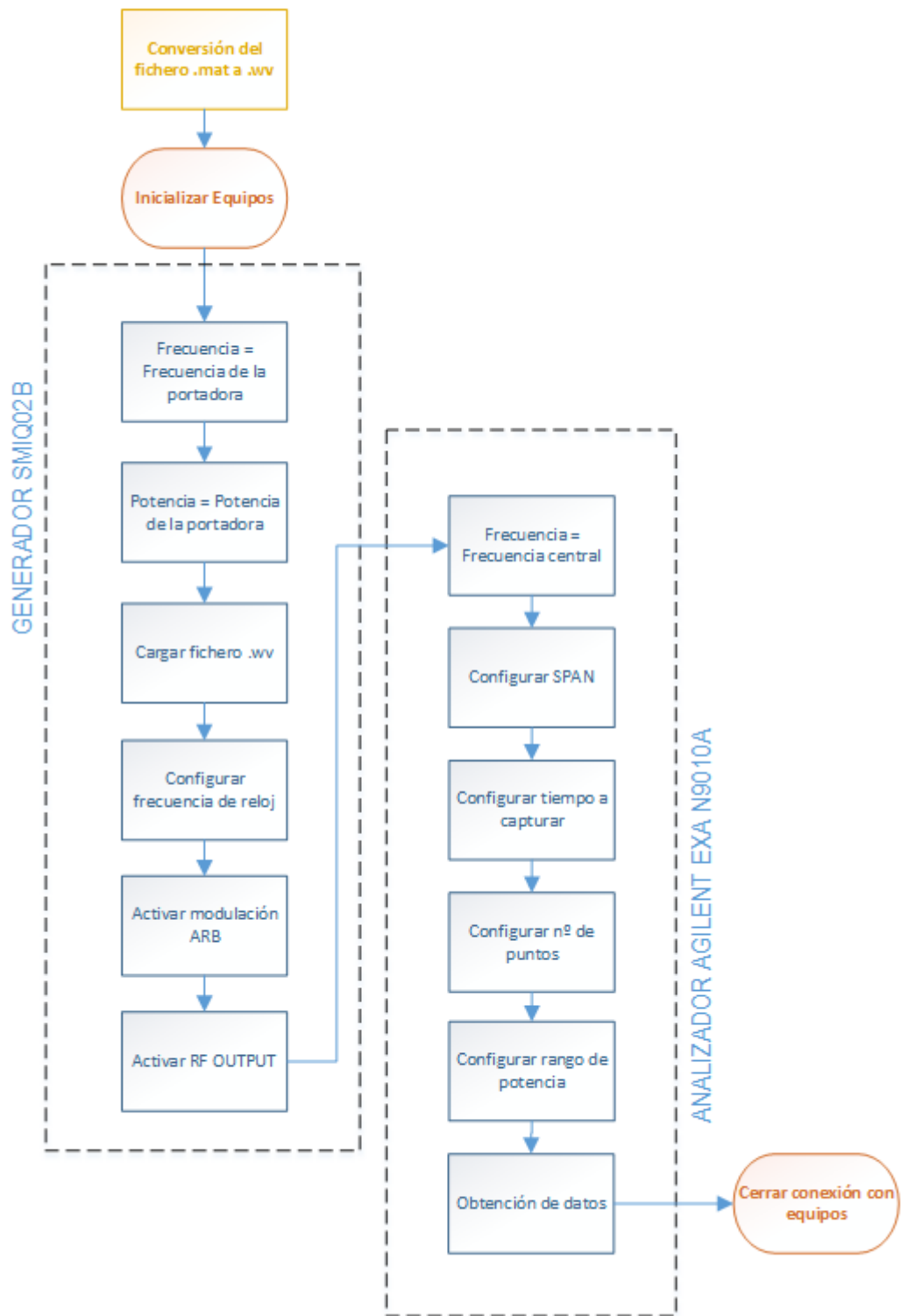


Figura 6-22 Diagrama de flujo para el ejercicio 2

6.2.3 Diseño del panel frontal

Para diseñar el panel frontal de este instrumento se harán falta los siguientes elementos:

- **Controles numéricos:** En estos controles se incluirán los valores de entrada para la tasa de muestreo, frecuencia central de la portadora, potencia de la portadora, frecuencia central del analizador, SPAN, tiempo a capturar, rango de potencia y el número de puntos.
- **Controles tipo string:** Son los encargados de recoger una cadena de caracteres con las rutas y nombres de los ficheros a utilizar.
- **Menús seleccionables:** Se incluirán estos menús para seleccionar las unidades de la frecuencia en el analizador y el generador. Su configuración es exactamente igual que en el ejercicio anterior.
- **Gráficas XY:** En ellas se representarán los resultados de las componentes en fase y cuadratura en el dominio del tiempo.

Además de esto, se vuelven a incluir elementos decorativos con el fin de dejar el panel más claro.

El resultado final debe asemejarse al de la siguiente imagen:

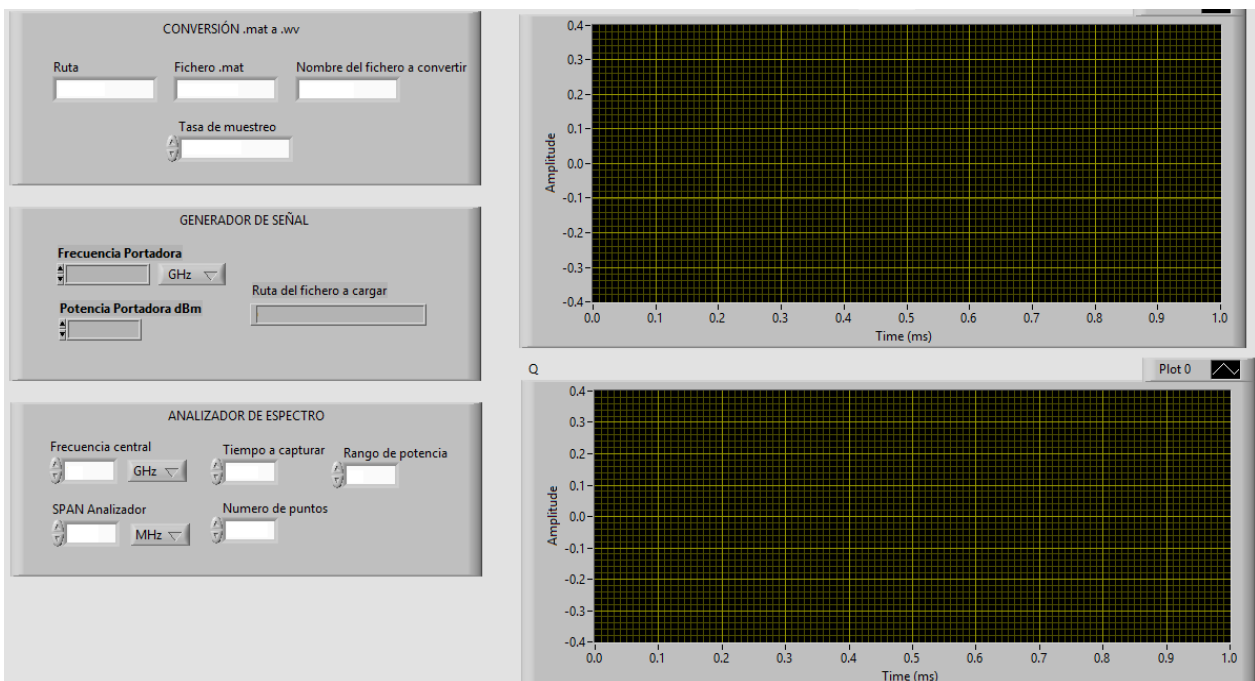


Figura 6-23 Panel frontal del ejercicio 2

6.2.4 Diseño del diagrama de bloques

El diseño del diagrama de bloques estará estructurado con bloques secuenciales de modo que la parte del generador y el analizador se ejecuten de forma consecutiva.

6.2.4.1 Conversión de extensión .mat a .wv

La primera estructura secuencial de este instrumento virtual incluirá la programación necesaria para convertir el fichero .mat de MATLAB al fichero .wv que necesita el generador. Para realizar esta tarea, es necesario acudir a un código escrito en MATLAB que proporciona el fabricante Rohde & Schwarz y que ha sido modificado para cumplir los objetivos de esta memoria.

Para incluir un código de MATLAB en LabVIEW existe un bloque específico. Este bloque recibe el nombre de **MATLAB Script** y se ubica dentro de la carpeta **Mathematics > Scripts & Formulas > Script Nodes** dentro de la paleta de funciones.

Se pueden configurar los parámetros de entrada y salida de este código pulsando sobre **Add Input/Output** tras hacer click derecho sobre el elemento en el diagrama de bloques.

En este ejercicio, los parámetros que va a recibir como entradas serán los controles de tipo string y numéricos que indican la ruta, nombre del fichero .mat, nombre del fichero a convertir y tasa de muestreo, quedando como resultado el de la siguiente imagen:

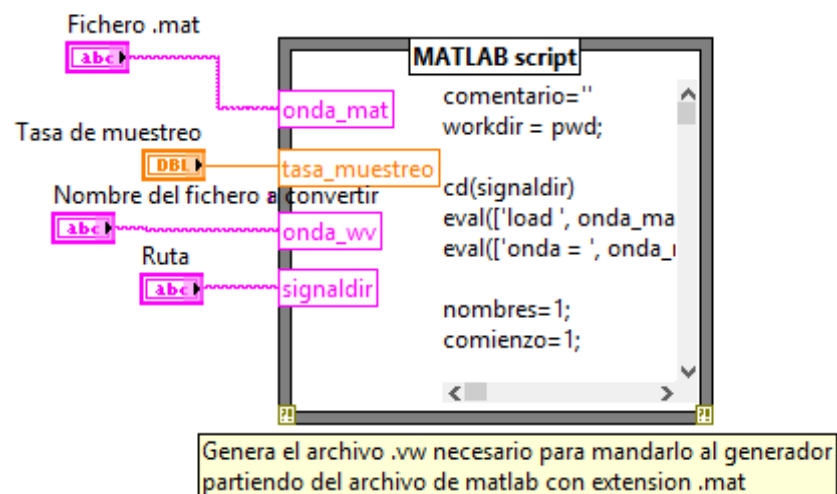






Figura 6-24 Estructura para la conversión de extensión

Al ejecutarse este bloque se genera un fichero con la extensión .wv en la misma carpeta que se indicó en la entrada ruta.

El código incluido en este bloque se puede encontrar en la sección anexos del final de la memoria.

6.2.4.2 Configuración del generador

Dentro de esta estructura secuencial se configuran los parámetros del generador para realizar la modulación, siendo necesarios, además de los bloques de frecuencia, potencia y nivel de RF descritos en el ejercicio 1, los siguientes elementos:

- **Carga del fichero con forma de onda** . Con este bloque es posible cargar el fichero .wv en la memoria del generador. Como parámetro recibirá la ruta donde se encuentra el archivo en el ordenador incluyendo el .wv de la extensión, por ejemplo: *C:/señales/wcdma.wv*
- **Selección de forma de onda** . Selecciona una forma de onda a utilizar de todas las que están cargadas en la memoria del generador. Como parámetro de entrada recibe el nombre de la forma de onda incluyendo el .wv.
- **Configuración del reloj** . Configura la frecuencia de reloj para la generación de la señal ARB a la salida. En este caso será igual a la frecuencia de muestreo de la señal que se introduce.
- **Activación/Desactivación ARB** . Activa o desactiva la salida del generador de forma de onda arbitraria. Se activa con un valor TRUE a la entrada y se desactiva con un valor FALSE.

La inclusión de todos los bloques en la estructura secuencial hace que quede de este modo:

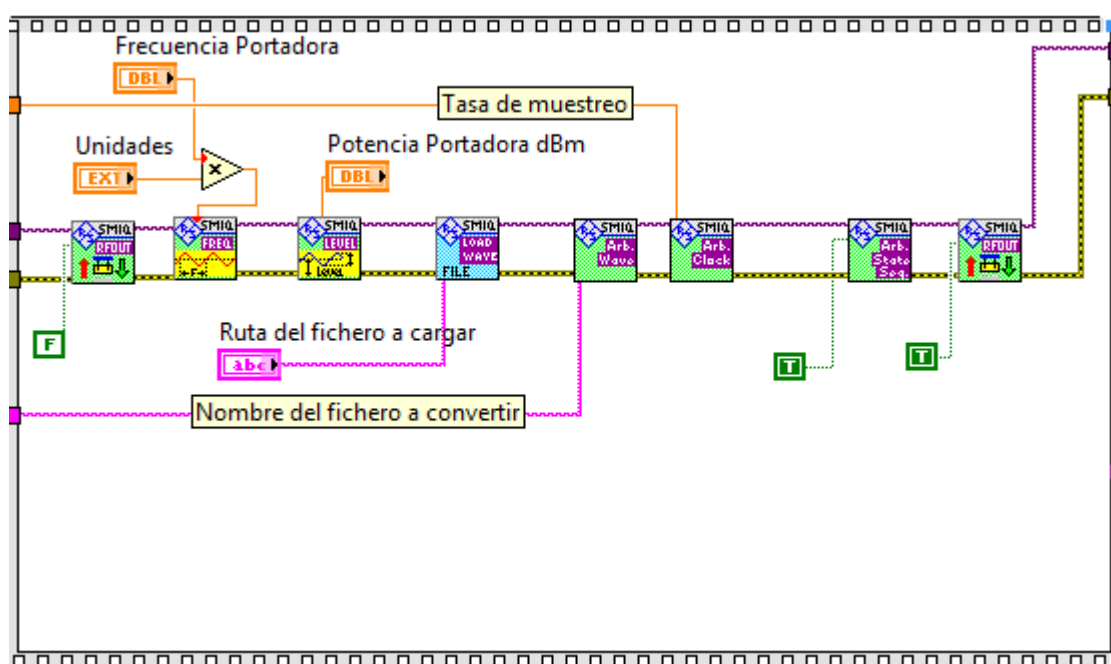




Figura 6-25 Configuración del generador para el ejercicio 2

6.2.4.3 Configuración del analizador

La configuración para el analizador de espectro es bien distinta a la que se ha visto hasta ahora. Para este ejercicio se ha comentado que la configuración del modo VXA no está disponible en los bloques que el fabricante incluye en los controladores del instrumento, por lo tanto, la única opción que queda es la utilización de comandos SCPI.

En LabVIEW es posible incluir comandos SCPI mediante el bloque de escritura  y leer su salida con el bloque de lectura . Para la escritura, se recibe el comando SCPI en formato cadena de caracteres (string). El bloque de lectura solo recibe el valor de bytes a leer, como en este caso se desconocen a priori el número de bytes que se van a recoger, este parámetro se configurará con un valor muy alto e igual a 100000000 para evitar problemas.

Los comandos que se van a usar para la configuración son:

- **INST VSA** - Selecciona el modo VXA del analizador.
- **FREQuency:CENTer** *{frecuencia}* – Configura la frecuencia central.
- **FREQuency:SPAN** *{span}* – Configura el Span.
- **VECT:SWE:TIME** *{tiempo}* – Configura el tiempo a capturar.
- **VECT:SWE:POIN** *{nº de puntos}* – Configura el número de puntos.
- **POW:RANG** *{potencia}* – Configura el rango de potencia.
- **INITiate:CONTinuous ON** – Inicia el modo continuo de barrido.
- **FORM:DATA ASCII** – Selecciona ASCII como formato de los datos a leer.
- **CALC:VECT:DATA2:RAW?; *WAI** – Obtiene los valores en bruto de la medida.
- **CALC:VECT:DATA2? X** – Obtiene los valores del eje X, es decir, del tiempo.

La mayoría de estos comandos necesitan ser completados con el valor de la configuración. Por ejemplo, para el caso de la frecuencia el código completo sería: **FREQuency:CENTer 1000000** para configurar 1 MHz en el analizador.

Los valores con los que se quieren configurar los instrumentos se han recibido mediante controles numéricos desde el panel frontal. Es por ello que para completar el comando SCPI con el valor de la configuración ya incluido, es necesario concatenar la cadena de caracteres con el inicio del código SCPI y el valor numérico convertido a cadena de caracteres como indica la siguiente imagen:

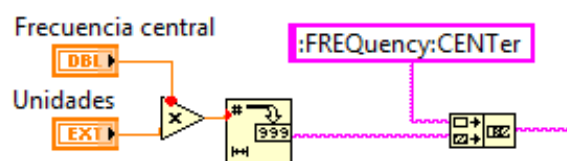




Figura 6-26 Concatenar valores numéricos con instrucción SCPI



El bloque  se encarga de convertir un valor numérico en cadena de caracteres (string) y se puede localizar en la carpeta **String>Number/String Conversion** de la paleta de funciones. Como es lógico, el otro bloque  es el que realiza la concatenación de ambas cadenas de caracteres, se encuentra en la carpeta **String**.

Existe un problema a la hora de usar los valores numéricos del panel frontal como parámetros de los comandos SCPI. Por defecto LabVIEW viene configurado de tal forma que utiliza la coma como separador decimal, esto es así porque se configura automáticamente dependiendo del país que se haya indicado en su instalación. Por otro lado, los equipos solo aceptan el punto como separador decimal a la hora de recibir comandos SCPI. Para evitar este problema, se va a configurar LabVIEW para que utilice el punto como separador en todo momento. Desde el menú **Tools/Options** se debe desmarcar la opción **Use localized decimal point** en la pestaña **Front Panel**.

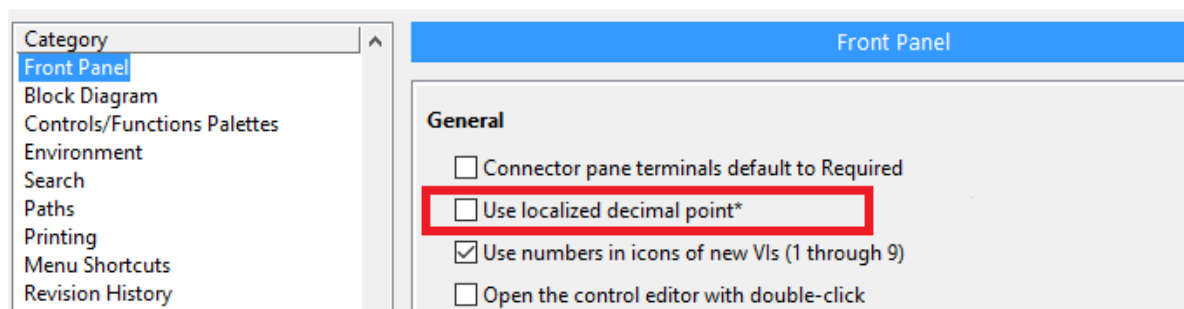


Figura 6-27 Configuración de la separación decimal en LabVIEW

El siguiente paso a realizar es la obtención en bruto (RAW) de los valores medidos en el analizador. Los valores leídos del analizador incluirán el valor de la componente en fase seguido del valor de la componente en cuadratura y así sucesivamente, por lo tanto, las componentes en fase se encuentran en las posiciones impares del vector recibido y las componentes en cuadratura en las posiciones pares. Para separar ambas componentes y tenerlas organizadas independientemente, se incluye un bloque de MATLAB con el siguiente código:

```
stack = str2num(data); % Convierte de cadena de caracteres a valor numérico

ii = stack(1:2:end); % Obtiene las componentes impares del vector (FASE)
qq = stack(2:2:end); % Obtiene las componentes impares del vector
(CUADRATURA)

i=resample(ii,32,25);
q=resample(qq, 32, 25);
```

El analizador utiliza una frecuencia de muestreo igual a $1.28 \cdot SPAN$ para coger las muestras de la señal. Con el objetivo de eliminar ese factor de 1.28, en el código anterior se ha incluido una función de remuestreo con factor $32/25$.

La parte encargada de obtener los datos del analizador tendrá esta forma:

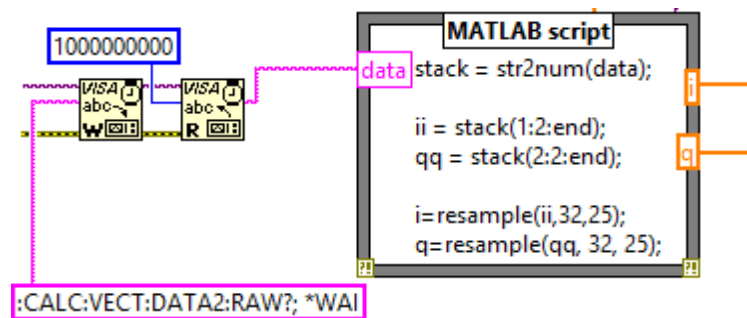


Figura 6-28 Obtención de medidas para el ejercicio 2

A la hora de hacer la gráfica final, se necesitan conocer los valores del eje del tiempo. Para recoger esta información se usa de nuevo un bloque de MATLAB que tiene el siguiente código:

```
tt = str2num(data2); % Convierte de cadena de caracteres a valores numéricos
tt=resample(tt, 32, 25);
t=tt.*1000; % Para representar su resultado en milisegundos
```

De la misma forma se ha usado la función `resample` para que coincidan los valores del tiempo con los de la muestra.

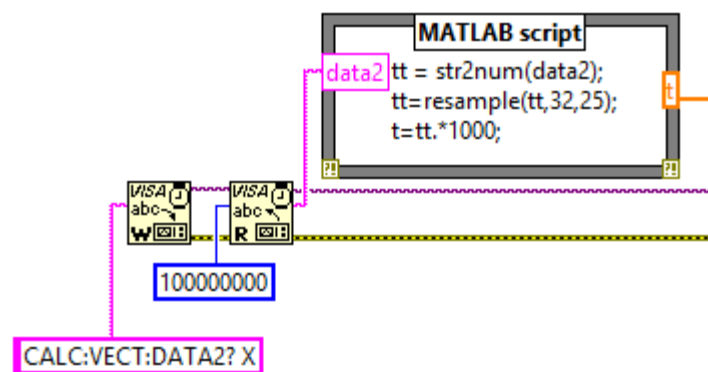


Figura 6-29 Obtención de los valores del tiempo para el ejercicio 2

Las variables de salida de los bloques de MATLAB que toman el nombre **i**, **q** y **t** se conectarán con los bloques de gráficas para dibujar el resultado. El procedimiento es el mismo que en el ejercicio 1 y se puede ver en el diagrama de bloques final.

6.2.4.4 Diagrama de bloques completo

A continuación, se muestra el resultado del diagrama de bloques completo tras añadir todo lo comentado en esta sección.

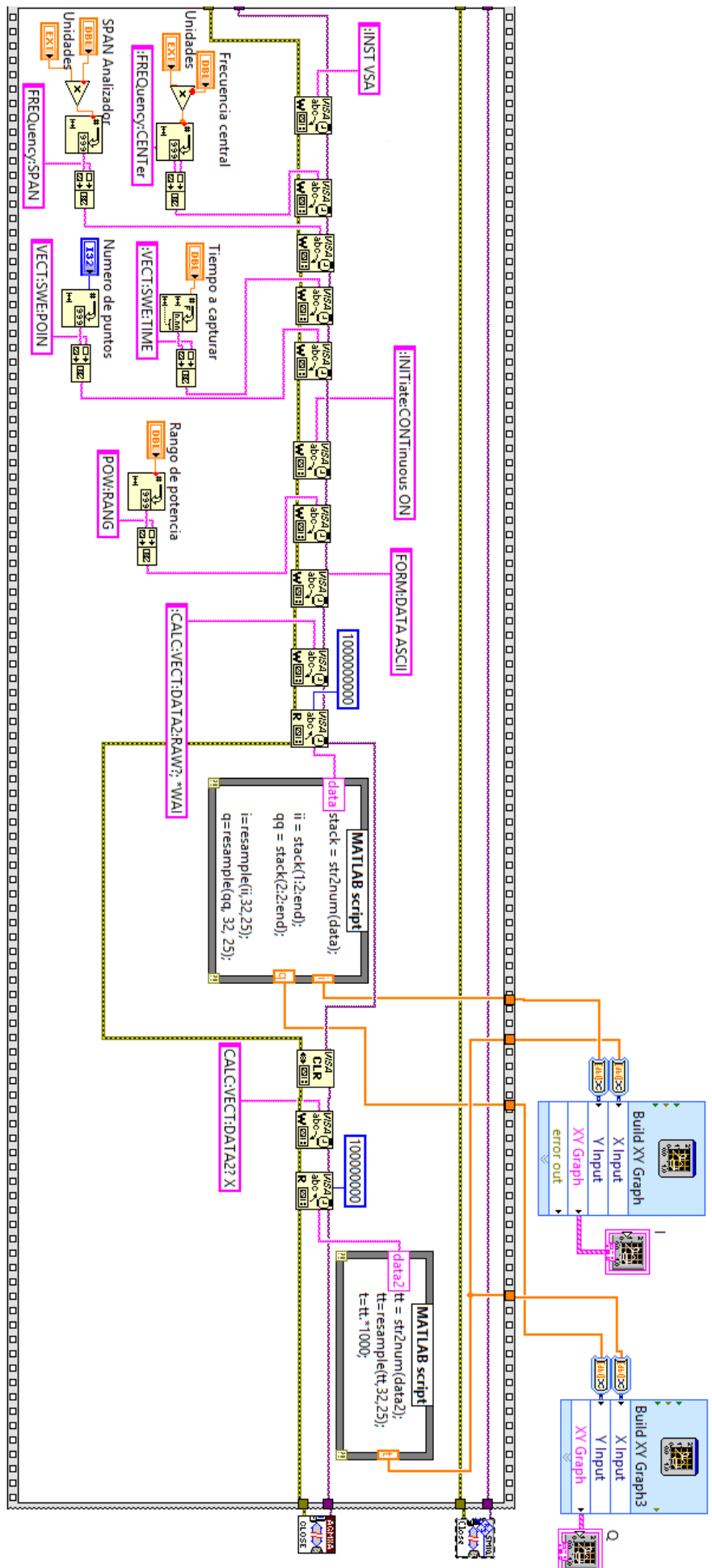


Figura 6-31 Diagrama de bloques del ejercicio 2 (parte 2)

6.2.5 Ejecución y resultados

Para la ejecución de este ejercicio se ha elegido una forma de onda con 8 muestras por símbolo y una tasa de 3.84 Msps, lo que da una tasa de muestreo de 30.72 MHz. Es una señal con modulación QPSK con filtrado tipo coseno alzado con un factor de roll off igual a $\alpha = 0.22$ que se corresponde con el estándar WCDMA del 3GPP de la tercera generación de comunicaciones móviles. El código en MATLAB que genera esta forma de onda con extensión .mat se puede encontrar la sección Anexos.

Los parámetros introducidos para esta medida son:

- Forma de onda: x_wcdma.mat
- Frecuencia de la portadora: 1 GHz
- Potencia de la portadora: -10 dBm
- Frecuencia central del analizador: 1 GHz
- Span: 7.68 MHz
- Tiempo a capturar: 0.001 segundos
- Rango de potencia: 0 dBm
- Número de puntos: 409601

Para el valor de Span se ha elegido el doble de 3.84 MHz que es aproximadamente el ancho de banda de la señal a obtener. El tiempo a capturar con valor de 1 ms es el máximo con el que se ha podido hacer la ejecución sin generar problemas de memoria en LabVIEW. Como rango de frecuencia se ha introducido un valor alto, ya que, si se elige un valor por debajo del máximo de la señal obtenida, se saturará la potencia de la señal en el analizador obteniendo el mensaje OV1. El número de puntos es el máximo permitido por el analizador.

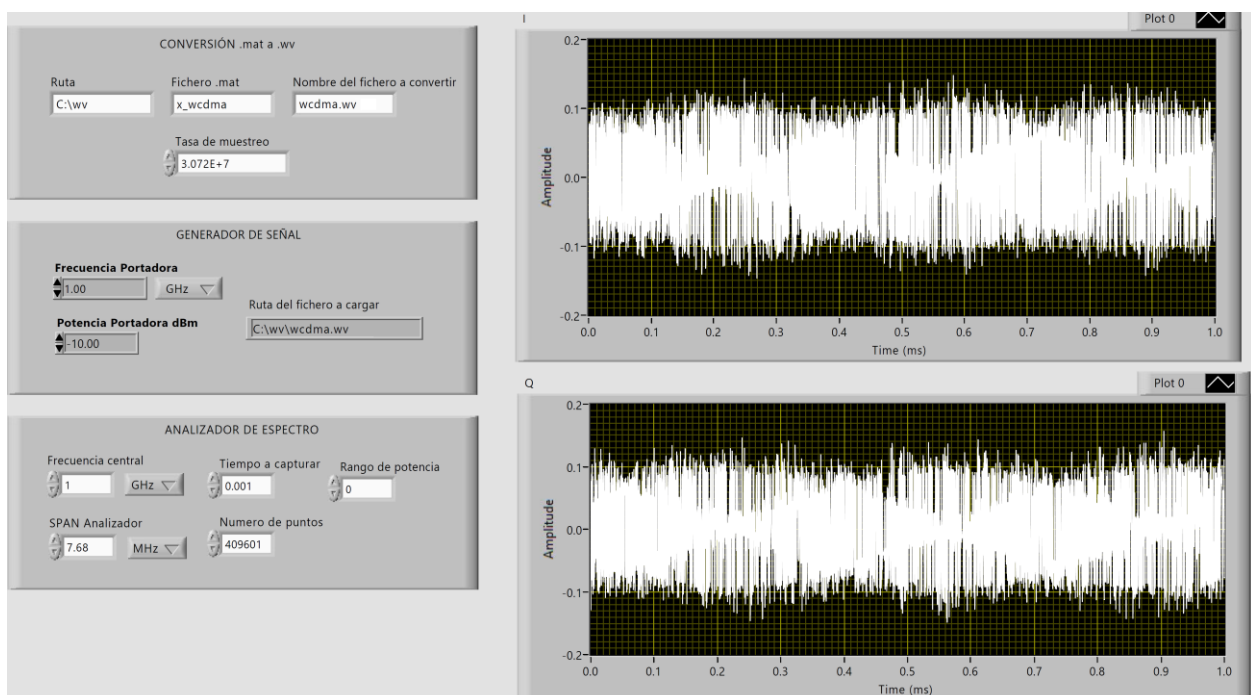


Figura 6-32 Resultados obtenidos para el segundo ejercicio

El resultado obtenido tras la ejecución en el panel frontal se muestra en la Figura 6-32. Por otro lado, en la pantalla del analizador estas son las gráficas obtenidas de la medida:

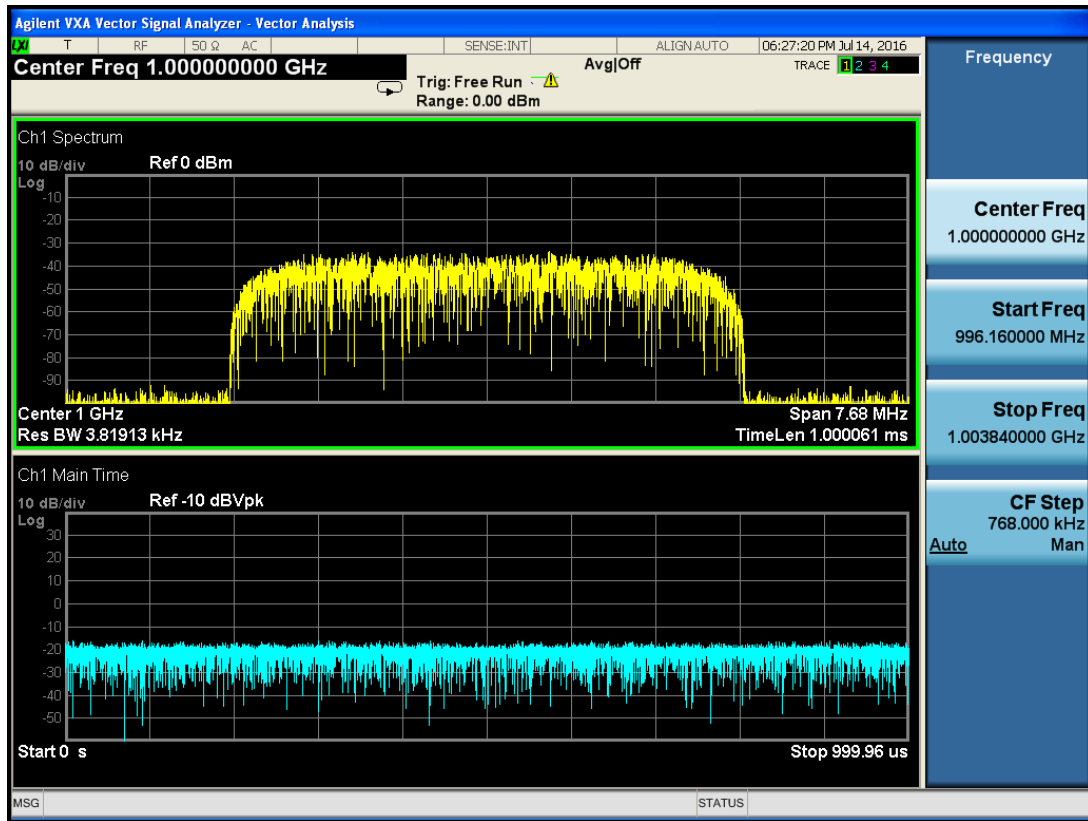


Figura 6-33 Resultados obtenidos en el analizador para el segundo ejercicio

7 CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO

Durante este proyecto se ha demostrado satisfactoriamente cómo el diseño de una plataforma experimental desarrollada en LabVIEW ofrece una herramienta de trabajo muy valiosa a la hora de controlar de forma remota instrumentación de medida. Gracias a esto ha sido posible facilitar la realización de medidas con el fin de caracterizar dispositivos de RF, de modo que el usuario final no tenga que interaccionar directamente con el equipo y pueda configurar sus parámetros desde la ventana del panel frontal. Lo más importante de esta herramienta es el tiempo que puede llegar a ahorrarse en realizar medidas que necesiten cientos o miles de configuraciones de los equipos. Tomando como ejemplo la medida realizada en el primer ejercicio donde se hace un barrido en potencia que va desde -25 dBm hasta 12 dBm con paso de 0.1 dB, harían falta un total de 1900 configuraciones que se tendrían que haber hecho de forma manual si no se contara con una herramienta como ésta. Otra de las ventajas que se ha podido comprobar en el transcurso de la memoria, es la rápida curva de aprendizaje que tiene el lenguaje de programación gráfico, ya que evita tener que usar líneas de código u otras disciplinas de programación y permite que un usuario cualquiera que no esté familiarizado con LabVIEW pueda entender la estructura de un instrumento virtual en pocos minutos.

Aprovechando lo diseñado en este proyecto se pueden afrontar nuevos objetivos como por ejemplo el estudio de la predistorsión u otras técnicas de linealización para el diseño de amplificadores que mejoren la linealidad, algo que sería de gran importancia en los sistemas de radiocomunicación actuales.

Uno de los problemas que han surgido durante la realización ha sido la inexistencia de ciertos bloques en los controladores que permitan configurar ciertos parámetros en los equipos. Puede resultar interesante diseñar este tipo de bloques mediante la creación de subVIs que usen internamente comandos SCPI, de este modo, se evitaría por completo el uso de comandos que puedan resultar confusos.

8 BIBLIOGRAFÍA

- [1] Ronald W. Larsen. *LabVIEW for Engineers, 1st Edition, Pearson, 2010.*
- [2] Joaquín del Río Fernández. *LabVIEW: Programación para Sistemas de Instrumentación, Ed. Garceta, 2011.*
- [3] María José Madero Ayora, Carlos Crespo Cadenas. *Prácticas de la asignatura Fundamentos de Radiocomunicación. 2015.*
- [4] National Instruments. *LabVIEW Getting Started:*<http://www.ni.com/getting-started/labview-basics/esa/>.
- [5] Wikipedia. Wikipedia, la enciclopedia libre:<https://es.wikipedia.org>.
- [6] Agilent N9010A EXA, *Specifications Guide.*
- [7] Rohde & Schwarz Vector Signal Generator SMIQ02B, *Operating Manual.*
- [8] 89601X VXA, *Measurement Application User's and Programmer's Reference.*

9 ANEXOS

En este capítulo se añadirán documentos e información que no se ha incluido en capítulos anteriores para agilizar la lectura.

9.1 Datasheet del amplificador ZJL-6G+

Coaxial Amplifier

50Ω Low Power 20 to 6000 MHz

ZJL-6G+



CASE STYLE: BW459

Connectors	Model	Price	Qty.
SMA	ZJL-6G+	\$114.95 ea.	(1-9)

+ RoHS compliant in accordance with EU Directive (2002/95/EC)

The +Suffix has been added in order to identify RoHS Compliance. See our web site for RoHS Compliance methodologies and qualifications.

Features

- ultra wideband, 20 to 6000 MHz
- compact rugged case, 1.07" x 0.61" (including mounting bracket)
- protected by US Patent, 6,943,629

Applications

- communications systems
- radar
- instrumentation
- laboratory use

Amplifier Electrical Specifications

MODEL NO.	FREQUENCY (MHz)		GAIN (dB)			MAXIMUM POWER (dBm)			DYNAMIC RANGE		VSWR (:1) Typ.		DC POWER	
	f_L	f_U	Typ.	Min.	Flatness ¹ Typ.	Output (1 dB Compr.) L	U	Input (no damage)	NF (dB) Typ.	IP3 (dBm) Typ.	In	Out	Volt (V) Nom.	Current (mA) Max.
ZJL-6G+	20	6000	13	10	±1.6	+9	+10	+15	4.5	+24	1.5	1.4	12	50

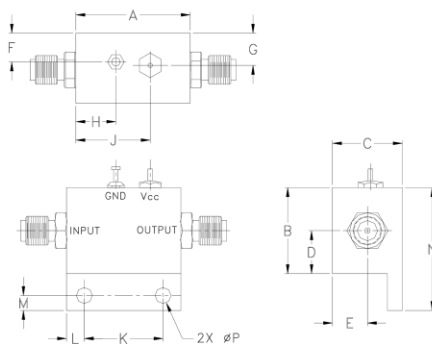
1. Flatness specified to 0.75 fU, dynamic range at 2 GHz.
Open load is not recommended, potentially can cause damage.
With no load derate max input power by 20 dB

L= low range (f_L to $f_U/2$) U= upper range ($f_U/2$ to f_U)

Maximum Ratings

Operating Temperature	-40°C to 75°C
Storage Temperature	-55°C to 100°C
DC Voltage	+13V Max.

Outline Drawing



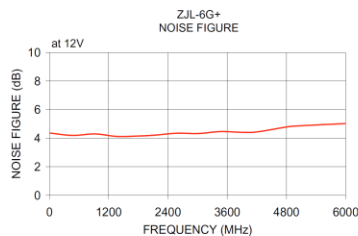
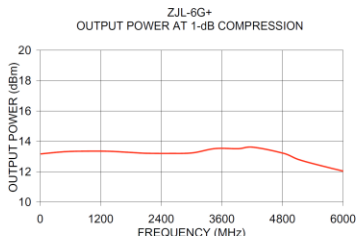
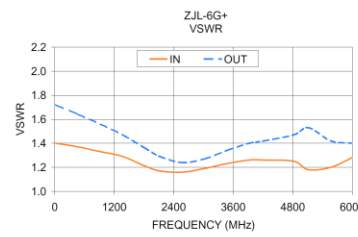
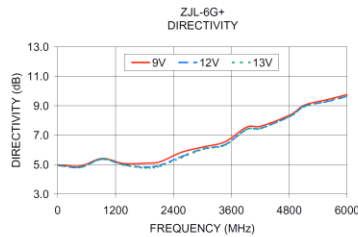
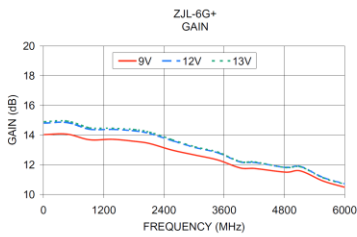
Outline Dimensions (inch)

A	B	C	D	E	F	G	H	J	K	L	M	N	P	wt
1.00	.75	.61	.38	.29	.25	.26	.35	.65	.688	.156	.13	1.07	.140	grams
25.40	19.05	15.49	9.65	7.37	6.35	6.60	8.89	16.51	17.48	3.96	3.30	27.18	3.56	25

Typical Performance Data/Curves

ZJL-6G+

FREQUENCY (MHz)	GAIN (dB)			DIRECTIVITY (dB)			VSWR (:1)		NOISE FIGURE (dB)	POUT at 1 dB COMPR. (dBm)
	9V	12V	13V	9V	12V	13V	IN	OUT		
20.00	14.03	14.80	14.90	4.98	4.94	4.94	1.40	1.72	4.36	13.17
470.00	14.06	14.84	14.94	4.90	4.82	4.83	1.37	1.64	4.20	13.31
920.00	13.68	14.39	14.48	5.42	5.37	5.39	1.33	1.56	4.30	13.35
1370.00	13.71	14.38	14.46	5.08	4.99	4.98	1.29	1.47	4.13	13.34
1820.00	13.57	14.28	14.37	5.10	4.83	4.78	1.21	1.36	4.16	13.26
2120.00	13.42	14.10	14.19	5.19	4.96	4.89	1.17	1.29	4.21	13.21
2570.00	12.98	13.62	13.69	5.84	5.55	5.48	1.16	1.24	4.35	13.20
3020.00	12.65	13.16	13.21	6.20	6.09	6.05	1.19	1.27	4.33	13.24
3470.00	12.32	12.81	12.86	6.55	6.36	6.33	1.23	1.34	4.47	13.52
3920.00	11.78	12.19	12.21	7.55	7.40	7.40	1.26	1.40	4.41	13.51
4220.00	11.74	12.15	12.18	7.62	7.49	7.46	1.26	1.42	4.45	13.61
4820.00	11.50	11.82	11.81	8.38	8.29	8.30	1.25	1.47	4.80	13.21
5120.00	11.58	11.88	11.86	9.05	8.98	8.95	1.18	1.53	4.88	12.80
5570.00	10.90	11.14	11.12	9.40	9.27	9.30	1.20	1.42	4.96	12.39
6000.00	10.48	10.71	10.69	9.76	9.64	9.67	1.28	1.40	5.03	12.04



9.2 Código para la generación de la señal WCDMA

```
% Generacion de una señal WCDMA

Fd = 1; Fs = 8; delay = 96;
Rb = 3.840e6;
N = Fs/Fd;
Nsymb = 2047;

% Secuencia Gold
u = gold2047(-2); u = 2*u - 1;
v = gold2047(-1); v = 2*v - 1;

% Envolvente compleja
x = u+j*v;

% Extensión periodica para evitar efecto borde en el filtrado
x_ex = [x(end-delay+1:end), x, x(1:delay)];

% Se añade el filtro raíz de coseno alzado del transmisor
y_ex = rcosflt(x_ex, Fd, Fs, 'fir/sqrt', 0.22, delay);

% Se recortan las colas que introduce el filtro coseno alzado
y = y_ex(2*delay*N + 1:end-(2*delay*N),:);

% Secuencia de instantes de tiempo asociados a las muestras
t = (0:1/Rb/Fs:Nsymb/Rb-1/Rb/Fs)';

% Espectro de la señal
hs = spectrum.mtm(10);
ypsd = psd(hs, y, 'Fs', Fs*Rb, 'CenterDC', 1);
ys = ypsd.data;
f = ypsd.frequencies;
figure, plot(f/1e6, 10*log10(abs(ys)))
xlabel('Frequency Offset (MHz)'), ylabel('Normalized Power Spectral Density (dB/Hz)')

x_wcdma = y;
save x_wcdma x_wcdma
```

9.2.1 Función gold2047

```

function g = gold2047(PRN_number)

% g = gold(PRN_number) devuelve el codigo Gold (127 bits)
% numero PRN_number. El generador de codigos se compone de
% dos registros de desplazamiento, con 7 bits cada uno.
% Las realimentaciones que se aplican en el registro superior
% son: 4,7; y para el inferior, 4,5,6,7. Ambos casos responden
% a m-secuencias de longitud maxima. La seleccion de los codigos
% dependen de PRN-number.

r1 = [1 1 1 1 1 1 1 1 1 1 1];
r2 = [1 1 1 1 1 1 1 1 1 1 1];

for k = 1:2047
    u(k) = r1(11);
    r1d(1) = xor(r1(9), r1(11));
    r1d(2:11) = r1(1:10);
    r1 = r1d;
    v(k) = r2(11);
    r2d(1) = xor(r2(3), xor(r2(6), xor(r2(9), r2(11))));
    r2d(2:11) = r2(1:10);
    r2 = r2d;
end

if PRN_number == -2
    g = u;
elseif PRN_number == -1
    g = v;
else
    g = xor(u, [v(PRN_number+1:end), v(1:PRN_number)]);
end

%b = 2*g - 1; % Senalizacion polar para el codigo PRN.
%corr = xcorr(b);
%vec_n = (-0.5 * (1 + length(corr)) + 1:0.5*(1 + length(corr)) - 1);
%plot(vec_n, corr), %title('Funcion de autocorrelacion del codigo PRN'), ...
%grid, pause

% Fin de funcion.

```