



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

**DESARROLLO DE PLATAFORMA PARA EXPERIMENTOS DE
IONIZACIÓN DE GASES POR RF MEDIANTE EL PROTOCOLO
GPIB**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

**LICENCIADO EN CIENCIAS DE LA
COMPUTACIÓN**

P R E S E N T A:

CÉSAR AUGUSTO VARGAS NOLASCO

DIRECTOR DE TESIS:

DR. JAIME FABIÁN VÁZQUEZ DE LA ROSA

Ciudad Universitaria, Cd. Mx., 2017





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

Son varios, pero puntuales, los nombres que deberían figurar en esta página. Debo tantas cosas. Sin embargo, hay a quien les debo más que a otros el hecho de haber librado este trabajo (no quiere decir que haya cometido la insensatez de ponderar las cosas tan grandes que me han cedido todas las buenas amistades que comparto).

Agradezco...

... siempre primero a mis padres. Una deuda impagable me embarga, aunque por su calidad de santos sé que jamás se me pedirá ser saldada.

... a mi hermano que por sus contrastes he podido darle gravidez a mi vida.

... al Dr. Jaime Fabián Vázquez De la Rosa que en toda su generosidad no solo me ofreció una mano sino también virtudes y astucias.

... a mi maestro y amigo Carlos Ligan, por el acontecimiento de su amistad.

... a mis amigos Alina Rétiz y Manuel López que no solo se distinguieron en los momentos de derrota.

... a todos mis demás grandes amigos.

Finalmente agradezco al programa UNAM-DGAPA-PAPIME proyecto PE105516, Visualización de las ondas electromagnéticas en el intervalo de las Radio Frecuencias, por el apoyo económico brindado.

A Mamá, no en su figura (la madre), sino en su ser, y no en un ser sino en su cualidad (su alma): no en lo indispensable sino lo irremplazable. Que bien podría vivir sin ella, sin la madre (como todos lo hacemos, tarde o temprano), pero la vida que quedaría sería absoluta y enteramente incalificable (sin su bella cualidad).

A Papá, que me ha valido más que cien maestros.

<<Ahí estábamos, por irnos y no.>>

Antonio Di Benedetto

Indice

Introducción	1
Capítulo I: GPIB	3
1.1 Antecedentes	3
1.2 Características Lógicas.....	4
1.2.1 Tipos de Dispositivos.....	4
1.2.2 Secuencia Handshake (lógica de funcionamiento)	5
1.2.3 Direcciones	5
1.2.4 Mensajes de Dispositivos GPIB.....	6
1.2.4.1 Estructura de un comando general.....	7
1.2.5 Comandos SCPI.....	7
1.2.5.1 Descripción de bloques funcionales.....	9
1.2.5.2 Jerarquía.....	10
1.2.5.3 Palabras clave opcionales.....	10
1.2.5.4 Formato largo y corto.....	11
1.2.6.5 Comandos comunes y específicos.....	11
1.3 Líneas de Transmisión	11
1.3.1 Secuencia Handshake (desde las Líneas de Control de Transferencia de Datos)	14
1.4 Topología de conexión	15

1.5 Referencias.....	15
Capítulo 2: Instrumentos	16
2.1 Multi-Device Controller	16
2.1.1 Funcionamiento	16
2.1.2 Descripción del Panel Frontal.....	17
2.1.2.1 Botón Local e Indicadores (1).....	17
2.1.2.2 Configuración (2).....	18
2.1.2.3. Botones de Límite e Indicadores (3).....	18
2.1.2.4 Botones de Posición e Indicadores (4)	19
2.1.2.5 Botones de Paso (5).....	20
2.1.2.6 Botón Scan e Indicador (6)	20
2.1.2.7 Indicadores de Unidad (7)	20
2.1.2.8 Display Principal (8).....	20
2.1.3 Operación remota del Modelo 2090 mediante el uso de comandos GPIB.....	20
2.1.3.1 Operación remota Torre	21
2.1.3.2 Operación remota Mesa Rotatoria	22
2.2 Amplificador de Potencia	23
2.2.1 Funcionamiento	23
2.2.2 Descripción del Panel Frontal.....	23
2.2.2.1 Interruptor de bloqueo bajo llave (Keylock Switch).....	24
2.2.2.2 Interruptor POWER	24
2.2.2.3 Interruptor STANDBY	24
2.2.2.4 Interruptor OPERATE.....	24
2.2.2.5 Interruptor RESET	24
2.2.2.6 Alarma de sobrecarga RF	24
2.2.3 Operaciones del Panel de Control.....	25
2.2.4 Selección del modo de operación del Amplificador.....	25
2.2.4.1 Modo Manual.....	26
2.2.4.2 Modo Pulso	26
2.2.4.3 Modo de Control de Nivel Automático (CNA).....	26
2.2.5 Ajustes del Control de Ganancia RF (RF GAIN Control).....	26
2.2.6 Ajustes de Control del Limite (THRESHOLD Control)	26
2.2.7 Control de Ganancia del Detector (DETECTOR GAIN Control)	26

2.2.8 Control de Respuesta del CNA (ALC RESPONSE Control)	26
2.2.9 Operación Remota	27
2.2.9.1 Dirección IEEE-488 del amplificador	28
2.2.9.2 Formato del conjunto de comandos	29
2.2.9.3 Instrucciones al Amplificador	29
2.3 Generador de señales SML03 (9kHz a 3.3 GHz)	30
2.3.1 Descripción del Panel Frontal.....	30
2.3.2 Frecuencia RF	32
2.3.3 Nivel RF.....	33
2.3.4 RF On/Off.....	33
2.3.5 Manejo remoto del SML03.....	33
2.3.5.1 Dirección GPIB del SML03	33
2.3.5.2 Instrucciones al Generador de Señales	34
2.4 Antena logarítmica de arreglo de dipolos periódicos Modelo 3148.....	35
2.5 Cable USB/GPIB 2.0 de Alta Velocidad Modelo 82357b	35
2.5.1 Características	36
2.5.2 Suite de Bibliotecas IO.....	36
2.5.3 LEDs del 82357B	36
2.6 Referencias.....	37
Capítulo 3: Implementación	38
3.1 Requerimientos	38
3.2 Instrucciones	39
3.2.1 IBCLR.....	39
3.2.2 IBDEV	39
3.2.3 IBLOC	40
3.2.4 IBONL.....	40
3.2.5 IBRD	40
3.2.6 IBWRT	40
3.3 Funciones Globales	41
3.3.1 Palabra de Estado (ibsta).....	41
3.3.2 Función de Error (lberr).....	41
3.4 RFION.....	41

3.5 Base de Datos	42
3.5.1 Usuarios.....	42
2.5.1.1 Usuario Administrador	43
2.5.1.2 Usuario Académico	43
2.5.1.3 Usuario Estudiante	43
2.5.2 Historial	44
2.6 Inicio de Sesión.....	44
3.6 Form Principal	45
3.6.1 Panel Perfil	45
3.6.2 Panel Experimento	47
3.6.2.1 Controles	47
3.6.2.1.1 Inicialización de instrumentos GPIB.....	50
3.6.2.1.2 Configuración del Generador de Señales.....	51
3.6.2.1.3 Configuración del Amplificador de Potencia.....	53
3.6.2.1.4 Actualización y Generación de Reportes.....	54
3.6.3 Form Administración de Usuarios.....	57
3.6.3.1 Historial	57
3.6.3.2 Agregar Usuarios	59
3.6.4 Form Lista de Dispositivos.....	61
3.6.5 Form Ajustes.....	61
3.6.5.1 General	62
3.6.5.2 Propiedades.....	62
3.6.5.2 Test.....	63
3.7 Carga y Guardado de Configuraciones	66
3.8 Referencias.....	66
Capítulo 4: Observaciones y Trabajo Futuro.....	67
4.1 Observaciones del Amplificador Modelo 250W1000AM1.....	67
4.1.1 Cadena de terminación <LF>.....	67
4.1.2 Estado actual	67
4.2 Trabajo Futuro.....	68
4.2.1 Ampliación de RFION.....	68
4.2.1.1 Reportes en línea	68

4.2.2 Software de testeo propio	69
4.3 Referencias.....	69
Capítulo 5: Conclusiones	70
Apéndices	71
Apéndice A: Comandos del Multi-Device Controller	71
Apéndice B: Comandos del Amplificador.....	74
Apéndice C: Comandos del Generador de Señales.....	75

Introducción

No existe lugar alguno donde las ondas electromagnéticas se encuentren ausentes. Específicamente, son las ondas de radiofrecuencia -nombre dado a una parte del espectro electromagnético- con las cuales estamos en constante contacto: señales de radio y televisión, los teléfonos móviles, el Wi-Fi, etc. Sin embargo, a pesar de su permanente presencia no es posible darnos cuenta de ellas; no es posible, al menos, percibir las a través de los sentidos.

A pesar de lo anterior, si es posible lograr la visualización de las ondas electromagnéticas de forma indirecta, en específico las de radiofrecuencia (que son en las que enfocaremos nuestra atención); ósea, desde su manifestación.

Una de las formas para poder visualizar las ondas de radiofrecuencia es realizar experimentos de ionización de gases. Los experimentos de ionización de gases por radiofrecuencia consisten en radiar una onda de RF al aire y observar cómo esta incide en una lámpara de una combinación de diferentes gases. La onda de radiofrecuencia es dada por un generador de señales, la cual pasa por un amplificador de potencia para así ser radiada por una antena del tipo logarítmico. Si la señal radiada logra excitar (ionizar) las partículas de los gases de la lámpara (por ejemplo, argón o neón), está se encenderá sin que exista entre este algún medio físico que conecte la antena con la lámpara.

El objetivo de este trabajo es justo desarrollar una plataforma para realizar Experimentos de Ionización de Gases por RF pero de forma remota. Una plataforma que permita llevar a cabo dichos experimentos dentro de las instalaciones de la Cámara Semianecoica de la Facultad de Ciencias de la UNAM. Esta debe hacer uso de algunos de los instrumentos inscritos dentro de las instalaciones de la Cámara: El Generador de Señales Modelo SML03 del fabricante Rohde & Schwarz, el Amplificador de Potencia Modelo 250W1000AM1 del fabricante Amplifier Research y el Multi-Device Controller Modelo 2090 del fabricante ETS-EMCO.

Los instrumentos utilizados, en conjunto con la restante instrumentación inscrita a la Cámara Semianecoica, se encuentran actualmente conectados en red, mediante el bus de comunicación GPIB. Esta red es manipulada por las aplicaciones instaladas en el dispositivo controlador de la red (la computadora de escritorio del cuarto de mando). Aplicaciones con las cuales actualmente se realizan pruebas de diferente índole, por lo que nos hemos atendido a estas configuraciones.

Por lo anterior, a lo largo del Capítulo 1, se hablará sobre las características lógicas y físicas del bus de comunicación GPIB. El estudio de los pormenores del bus es primordial ya que existe toda una serie de procedimientos para la comunicación con un instrumento; desde una sintaxis hasta un orden del envío de instrucciones, que no depende solo del instrumento a manipular sino de las normas incluidas en las diferentes versiones del estándar IEEE 488. Estándar que tipifica el funcionamiento de los diversos instrumentos que utilizan el bus GPIB para su manejo remoto, sin importar la marca o su funcionamiento.

Sin embargo, debido a la complejidad, especificidad y susceptibilidad de cada instrumento también fue necesario el estudio detenido de cada uno desde sus manuales: su manejo manual y remoto. Cabe señalar que a cada uno de estos manuales solo se puede acceder cuando se realiza la compra del equipo en cuestión, por lo que los manuales son la única fuente de información disponible; estos requirieron de un tiempo considerable para su

estudio detenido debido a susceptibilidad de los instrumentos. En el capítulo 2 se revisará las características físicas y lógicas de operación de los dispositivos involucrados en el desarrollo de la aplicación.

En los capítulos siguientes (3, 4 y 5) se aborda la implementación de la aplicación en sí, las complicaciones de esta, como también su posible futuro.

Los objetivos que este trabajo se ha propuesto lograr son, por un lado, aportar un método de visualización de ondas electromagnéticas atractivo, para que alumnos y profesores puedan entender cómo estas pueden ser transmitidas por el aire; cómo es que son capaces de interactuar con la materia y como pueden ser absorbidas por ésta. Para poder observar esta energía radiada se propone el encendido de lámparas fluorescentes. Esto permitirá determinar de forma experimental la energía necesaria para ionizar dichos gases no habiendo conexiones entre la antena radiante y las lámparas mencionadas.

Al mismo tiempo que también puedan obtener conocimientos y habilidades sobre el funcionamiento de una Cámara Semianecoica y la instrumentación de medición asociada a esta. Cosa que no es menor, ya que la Cámara, instalación de gran envergadura, actualmente no goza del uso y la atención debida de alumnos y académicos.

Sin embargo, también este trabajo, y el software desarrollado, buscan ser un antecedente para el desarrollo de una aplicación de testeo y experimentación propia del laboratorio donde se encuentra la Cámara Semianecoica. Actualmente el software que gestiona los instrumentos para la realización de pruebas no permite la flexibilidad necesaria, respecto a configuraciones, para su uso de forma experimental. Agregar instrumentos de otras marcas como de manufactura reciente, dar nuevas configuraciones de parámetros e instrumentos involucrados en las rutinas de testeo, fuera de las ya definidas por el software (EMC32), son algunas de las cosas que limitan el uso de la instrumentación. Esto se debe a que el software no es de código abierto ya que pertenece a Rohde & Schwarz.

El desarrollo de una aplicación de testeo como de experimentación es un proyecto de gran complejidad y extensión (cabría señalar que para el desarrollo de esta tesis la lectura de los manuales consumió la mayor parte del tiempo que se invirtió). Sin embargo, es una tarea necesaria ya por un lado estas dificultades limitan las posibilidades de la Cámara; pero su desarrollo abriría nuevas posibilidades para la investigación, no solo en el ámbito en el que se inscribe el uso de la Cámara Semianecoica, sino también en lo que corresponde al desarrollo de software dentro de la instrumentación GPIB y científica en general.

Capítulo I: GPIB

1.1 Antecedentes

El GPIB (*General Purpose Interface Bus*), o como fue nombrado por sus creadores: HP-IB, es un canal de comunicación de corto alcance diseñado *por Hewlett-Packard* (HP) a finales de los años 60's con el propósito de comunicar y administrar de manera fiable instrumentos de medición y testeo en una configuración de red fabricados por el mismo HP. La introducción y proliferación de controladores digitales y específicamente de equipos de pruebas programables, ya no solo dentro de HP, originó la necesidad de una interfaz de alta velocidad para establecer comunicación entre instrumentos de diferentes proveedores.

En 1975, el *Institute Electrical and Electronics Engineers* (ó mejor conocido como la IEEE) estandarizó el protocolo, dando especificaciones eléctricas, mecánicas y funcionales del sistema de la interfaz, bajo el nombre de ANSI/IEEE 488-1975¹. Sin embargo, en junio de 1987, se realizó una revisión del estándar original para añadir clasificaciones editoriales y anexos.

Aún, y a pesar de la segunda revisión, el documento original del estándar IEEE 488 no especificaba normas de una sintaxis de preferencia ni de convenciones de formato para la programación de instrumentos, por lo que al estándar se le agregó un apéndice conocido como IEEE 488.2. Éste incluyó formatos, protocolos y comandos comunes para ser utilizados con el IEEE 488 (que fue renombrado como IEEE 488.1).

El IEEE 488.2 incluyó en sus especificaciones de 1990 el documento de Los Comandos Estándares para Instrumentación Programable (SCPI por sus siglas en inglés) donde se definen los comandos que todo instrumento, sin importar quién sea el fabricante, debe obedecer. Por lo que los SCPI garantiza compatibilidad entre diferentes instrumentos volviendo más fácil el remplazarlos, sin la necesidad de realizar cambios significativos en el código fuente, aunque éstos pertenezcan a un proveedor diferente.

Hay que aclarar que IEEE 488.2 no reemplaza el IEEE 488.1, ya que muchos dispositivos que todavía siguen en función solo cumplen el IEEE 488.1. IEEE 488.2 es una expansión del IEEE 488.1, ya que solo define un conjunto común de códigos y formatos para datos, un protocolo de mensajes para dispositivos, un conjunto genérico de comandos comúnmente necesitados por dispositivos y un nuevo modelo para generación de reportes de estado.

En este trabajo nos enfocaremos en ambos estándares IEEE 488.1 y IEEE 488.2, aunque serán tratados indistintamente por lo explicado anteriormente, ya que los instrumentos con lo que cuenta la Cámara Semianecoica de la Facultad de Ciencias-UNAM, presentan estos casos.

¹ American National Standards Institute/Institute Electrical and Electronics Engineers 488-1975

1.2 Características Lógicas

El objetivo principal del bus GPIB consiste en administrar la información que se transfiere entre dos o más dispositivos inscritos en una misma red de comunicación. Antes de que dicha transferencia tenga lugar los dispositivos deben configurarse bajo el mismo protocolo de transmisión de datos, como ser identificados unívocamente a una dirección dentro de la red de comunicación. Esta comunicación puede darse de forma remota o desde un panel de control (comúnmente el de un instrumento o una computadora).

1.2.1 Tipos de Dispositivos.

Dentro de una red de comunicación bajo el protocolo GPIB existen tres tipos diferentes de dispositivos: *hablantes*, *escuchas* y *controladores*. Cada uno desempeña funciones específicas, sin embargo, esto no significa que un dispositivo no pueda desempeñar más de un papel dentro de una red de comunicación.

El dispositivo controlador (*controller*) es un elemento único dentro de la red. Éste se encarga de supervisar todas las operaciones que ocurren en el bus. También de definir qué dispositivo envía información, quien es el que la recibe, en que momento deben hacerlo y hacerles saber con quién debe referirse a cada dispositivo. El controlador de igual manera asegura que no haya más de dos dispositivos hablantes comunicándose al mismo tiempo. Establece roles, indicando qué dispositivos deben mantenerse en estado de habla como también de escucha, excluyendo aquellos dispositivos a los cuales no les interesa escuchar, ya que no siempre todos los dispositivos conectados a una misma red necesitan estar involucrados en una transferencia de datos.

Cuando un dispositivo envía un mensaje a otro, se dice que un dispositivo hablante (*talker*) envía un mensaje a un dispositivo escucha (*listener*). Es aquí donde el dispositivo controlador se encarga de administrar el flujo de información en el bus GPIB. La función del controlador es establecer la comunicación entre dos dispositivos mediante una asignación de direcciones, como de determinar quién habla y quien escucha. En una red puede haber varios escuchas activos (*active listener*) pero solo puede haber un hablante activo (*active talker*).

Una vez finalizada la transmisión y recepción de datos, el controlador de la red, busca asegurarse, mediante un control de transferencia de datos, la secuencia handshake, que todos los dispositivos escuchas hayan recibido la información de forma correcta. El dispositivo más lento es el que limita la velocidad de transferencia de la red, así se asegura la recepción de los datos a todos los dispositivos.

Sin embargo, no siempre es necesario tener un dispositivo controlador dentro de la red de comunicación, que en la mayoría de los casos es una computadora. Cuando un dispositivo hablante envía un mensaje a uno escucha, el controlador, antes de que la comunicación se haga efectiva, habilita a ambos dispositivos y los comunica. Debido a que las funciones de control y la transferencia de datos se encuentran separadas lógicamente, cuando la asignación de roles de cada dispositivo está hecha el dispositivo controlador no participa en la transferencia de información. El controlador solo retoma su función en el bus cuando los dispositivos han dejado de transmitir información; podría, en esta situación, si se requiriera volver a referenciar otros dispositivos.

Existen dispositivos que exclusivamente son hablantes y otros que solamente funcionan como escuchas, por lo tanto, bastaría con que ambos dispositivos estuvieran conectados mediante el bus para poder comunicarse. El controlador solo se vuelve necesario cuando un dispositivo debe cambiar sus funciones –el escucha se vuelve hablante o el hablante, escucha- o cuando el hablante debe comunicarse con más de un dispositivo.

También, en una red GPIB puede haber varios dispositivos controladores, sin embargo, solo uno puede ser el Controller-in-charge (CIC) o el dispositivo controlador a cargo: el dispositivo controlador único que se encargará de gestionar la red. Sin embargo, el controlador a cargo puede ceder sus funciones a otro controlador.

1.2.2 Secuencia Handshake (lógica de funcionamiento)

La fiabilidad y credibilidad de la transmisión de datos del bus GPIB tiene lugar gracias a la secuencia Handshake. Ésta garantiza que la información enviada por un dispositivo hablante sea recibida íntegramente por un dispositivo escucha. Dicha secuencia se encarga de la recepción de cada byte mediante hardware, a través de líneas de control. Sin embargo, la secuencia handshake entre un hablante activo y un escucha puede ser explicada de la siguiente manera:

1. El escucha hace saber que está listo para escuchar.
2. El hablante coloca el byte de información dentro del bus y avisa al escucha que la información está disponible.
3. El escucha, al tanto de que la información está disponible, acepta la información y hace saber al hablante que ha sido aceptada.
4. El hablante, al tanto de que la información ha sido aceptada, detiene el flujo de información y hace saber al escucha que ya no está enviando información.
5. El escucha, al tanto de que el flujo de información ha cesado, hace saber al hablante que está preparado para recibir el siguiente byte.

En la sección de Líneas de Datos se explicará más a detalle su funcionamiento.

1.2.3 Direcciones

Como se mencionó anteriormente el dispositivo controlador es quien se encarga de administrar la red, de asignar los roles y hacerle saber a cada dispositivo con quien debe dirigirse. Sin embargo, esto sería imposible si cada instrumento no tuviera un identificador único con el cual fuera fácil distinguirlo de los demás dispositivos conectados a la red.

Cada dispositivo conectado al bus GPIB debe tener una dirección exclusiva. Es mediante esta dirección que el controlador puede saber qué dispositivo quiere comunicarse y con quien quiere hacerlo. Las direcciones GPIB están constituidas por dos partes: la dirección primaria que es un número identificador entero único que va desde el 0 al 30, es decir que existe 31 direcciones; y la dirección secundaria que configura el estado del dispositivo es decir si está en modo *talker* (TA), *listener* (LA), *unlistener*. Este último estado se refiere a los dispositivos que no están en estado de escucha ni como hablante; siempre un dispositivo al iniciarse comienza en estado *unlistener*.

Posición de Bit	7	6	5	4	3	2	1	0
Significado	0	TA	LA	Dirección primaria GPIB (de 0 a 30)				

Tabla 1.1 Bits, Dirección GPIB

Generalmente todos los instrumentos de medición tienen una dirección dada de fábrica, pero esta puede ser modificada a conveniencia de una configuración personalizada de red, ya sea desde el panel de control del dispositivo o desde el dispositivo controlador de la red. El dispositivo controlador generalmente suele tener asignada la dirección 0.

1.2.4 Mensajes de Dispositivos GPIB

Es mediante mensajes que los dispositivos GPIB se pueden comunicar entre sí. Sin embargo, existen dos tipos de mensajes: **mensajes de dispositivo** y **mensajes de gestión**.

Las **Respuestas de Dispositivos**, también conocidos como mensajes de datos o mensajes de dispositivos, son mensajes que un instrumento envía al controlador después de una petición. Estos mensajes contienen información respecto a un dispositivo específico como pueden ser configuraciones del instrumento, resultados de medición, estado de los instrumentos o archivos de datos.

Los **Mensajes de Comandos**, también conocidos como mensajes de gestión o solo llamados comandos, son mensajes que el controlador envía a los instrumentos. Los instrumentos llevan a cabo las instrucciones que les fueron dadas. Se encargan de administrar el flujo de mensajes en el bus; estos mensajes son comandos que habilitan o deshabilitan dispositivos, los inicializan, preguntan el estado del dispositivo, les hacen peticiones, habilitan el modo remoto o local. Los mensajes de comandos más comunes son los siguientes:

Comando	Nombre de la orden	Función
*CLS	Clear Status Command	Despejar el registro de estado y los registros de incidencia.
*ESE	Event Status Enable Command	Habilita bits del registro de habilitación de incidencias.
*ESE?	Event Status Enable Query	Interroga el registro de habilitación de incidencias estándar.
*ESR?	Event Status Register	Interroga el registro de incidencias estándar.
*IDN?	Identification Query	Identifica el tipo de instrumento y versión de software.
*LRN?	Learn Device Setup	Devuelve el estado actual del equipo.
*OPC	Operation Complete Command	Fija bit de "Operación completa" del registro estándar.
*OPC?	Operation Complete Query	Responde con "1" si se han ejecutado órdenes previas.
*OPT?	Option Identification Query	Devuelve las opciones instaladas en el equipo. (Ejemplos: memoria extendida, Seguridad)
*RCL	Recall Command	Restaura el estado del equipo del registro save/recall

*RST	Reset Command	Sitúa al equipo en el estado básico de referencia.
*SAV	Save Command	Almacena el estado actual en el registro save/recall.
*SRE	Service Request Enable Command	Habilita los bits del registro de habilitación de Byte de estado.
*SRE?	Service Request Enable Query	Devuelve el contenido del registro SER de habilitación del byte de estado.
*STB?	Read Status Byte Query	Devuelve el estado del registro resumido del Byte de estado.
*TRG	Trigger Command	Arranca o dispara la operación del equipo de forma remota.
*TST?	Self-Test Query	Devuelve el resultado del autotest del equipo.
*WAI	Wait-to-Continue Command	Espera a que se realicen todas las operaciones pendientes.

Tabla 1.2 Tabla de Comando Comunes

1.2.4.1 Estructura de un comando general

Los comandos consisten de un encabezado y, en la mayoría de los casos, de uno o más parámetros. Los encabezados y parámetros están separados por un espacio en blanco. Todos los caracteres son código ASCII. El encabezado puede estar compuesto de varias palabras clave. Las peticiones se forman añadiendo directamente un signo de interrogación de cerradura “?” al encabezado.

Si quisiéramos enviar un mensaje de gestión a un dispositivo, debemos agregar un asterisco “*” que anteceda a la cabecera.

Ejemplos:

*RST - RESET, da un reset al dispositivo.

*ESE 255 - IDENTIFICATION QUERY, Establece los bits del registro de habilitación del estado de sucesos.

*IDN? - Preguntamos al dispositivo su identidad.

1.2.5 Comandos SCPI

La revisión de 1990 del estándar IEEE 488.2 trajo consigo uno de los atributos más importantes del bus GPIB: los **Comandos Estándares para Instrumentación Programable**, o mejor conocido como la norma SCPI. Esta norma consigue estandarizar los comandos de control y los formatos de datos de los instrumentos; lo que significa que sin importar el fabricante, los equipos que tienen la misma funcionalidad responden de la misma manera con el mismo conjunto de instrucciones.

La norma SCPI es de gran ayuda en el desarrollo de aplicaciones de control de sistemas de instrumentación para pruebas; ya que vuelve más fácil el aprendizaje y uso de los comandos, facilita el desarrollo y mantenimiento de aplicaciones y lo más importante, vuelve posible la sustitución de instrumentos sin la necesidad de realizar cambios significativos al código. Para lograr esto último la norma SCPI define una sintaxis y el formato de mensajes

general, para que los instrumentos que son funcionalmente iguales respondan a los mismos comandos.

Para que un conjunto de comandos sea compatible para múltiples instrumentos fue necesario definir un modelo general de instrumento; es el que se muestra a continuación.

El modelo general estandariza un conjunto de bloques funcionales, los cuales tiene asignado a su vez, un conjunto de comandos. Un instrumento está compuesto solo por subconjunto de estos bloques funcionales, dependiendo de su funcionalidad. La programación de dicho instrumento será conforme a los bloques funcionales de los que este compuesto. La imagen describe el flujo de datos del modelo general de un instrumento; las flechas continuas representan el flujo de datos, las flechas discontinuas representan el flujo control.

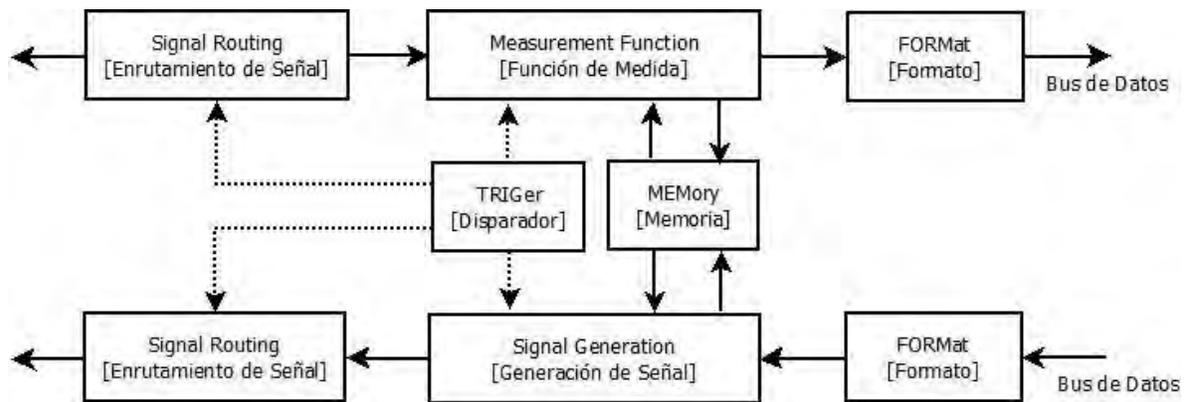


Figura 1.2. Modelo general de un instrumento programable según la norma SCPI. (Diagrama creado a partir de la información obtenida del manual de operación del EMI TEST RECEIVER, Rohde & Schwarz Munich 2001)

La representación de un instrumento está dada por los bloques de control que lo constituyen. Por ejemplo, un multímetro el cual solo tiene una entrada y ninguna memoria direccionable para el usuario puede ser representado de la siguiente manera:

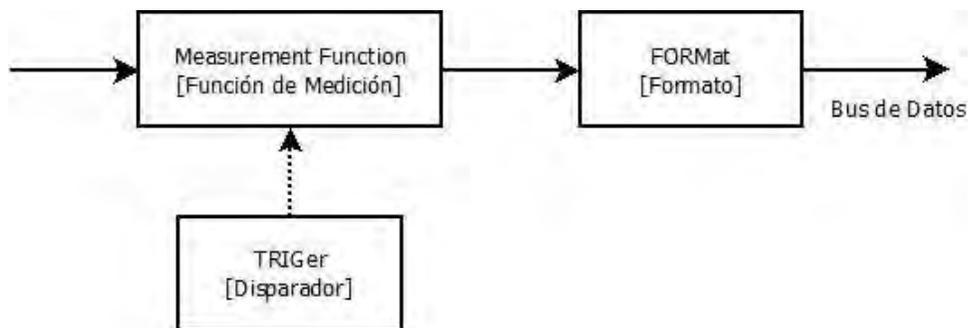


Figura 1.3. Representación de Bloques para SCPI de un multímetro. (Diagrama creado a partir de la información obtenida del manual de operación del EMI TEST RECEIVER, Rohde & Schwarz Munich 2001)

De la misma manera una fuente de alimentación solo necesitaría el de generación de señales (**Signal Generation**) y el bloque **FORMat** para la entrada.

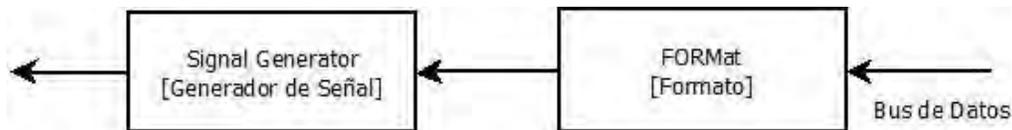


Figura 1.4. Representación de Bloques para SCPI de una fuente de alimentación. (Diagrama creado a partir de la información obtenida del manual de operación del EMI TEST RECEIVER, Rohde & Schwarz Munich 2001)

1.2.5.1 Descripción de bloques funcionales

Signal ROUTing: Este bloque se refiere a la capacidad de ciertos instrumentos de direccionar sus señales de entrada a la entrada de otros circuitos internos, en el caso de que dicho instrumento cumpla con varias funciones. Los comandos se pueden encontrar en la jerarquía ROUTe

MEASurement Function: Este bloque es el que realiza las mediciones; convierte una señal física en datos que pueden ser entendidos dentro del bus. El bloque está dividido en tres sub-bloques: INPut, SENSE y CALCulate. Estos no se muestran explícitamente dentro del modelo general de un instrumento porque si existen instrumentos que realicen la misma medida utilizarían comandos distintos.

INPut: El propósito de este bloque es adaptar la señal recibida antes de que esta sea transformada a datos por el bloque SENSE: es la etapa de adaptación de la señal de entrada.

SENSE: El bloque SENSE se encarga de transformar las señales eléctricas ya adaptadas por INPut de entrada en datos que pueden ser manipulados ya por una computadora.

CALCulate: Este bloque se encarga de transformar los datos adquiridos en información específica para una aplicación concreta. Por ejemplo: frecuencias, tiempos de subida, periodos, etc.

Signal Generation: Este bloque es responsable de la conversión de datos a una señal física. Al igual que el bloque MEASurement, este bloque contiene sub-bloques: CALCulate, SOURce y OUTput. Estos bloques permiten la comunicación con los otros bloques principales como TRIGger y MEMory, dándole la posibilidad de realizar tareas adicionales de cálculos de preconversión y acondicionamiento de señales.

SOURce: Este bloque se encarga de generar una señal con base en las características requeridas y los datos suministrados.

OUTput: Este bloque se encarga de acondicionar la señal de salida ya que ha sido generada. Las funciones de OUTput incluyen filtrado, polarización, conversión de frecuencia y atenuación.

TRIGger: El bloque TRIGger se encarga de dotar al instrumento con la capacidad de poderse sincronizar con eventos externos.

MEMory: El propósito de este bloque es permitir al instrumento almacenar información. Hay memoria para la que el usuario puede ser inaccesible, como los datos de calibración interna.

FORMat: Se encarga de convertir los datos internos del instrumento en otras representaciones de datos, sobre los datos que llegan a transferirse por el bus. Un ejemplo podría ser la conversión de datos internos a ASCII.

1.2.5.2 Jerarquía

Los comandos SCPI tienen estructura jerárquica, por niveles, que van de acuerdo a los bloques funcionales que un instrumento tenga implementado: los comandos de la norma SCPI están agrupados en un árbol. Los diferentes niveles son representados por una combinación de palabras clave separadas por dos puntos ":" en la cabecera. La palabra clave de nivel más alto, conocida como el nivel raíz (*root level*), solo tiene una palabra clave: SOURce (en el caso de los generadores de funciones). Esta sola cabecera (o palabra clave) denota por sí misma un comando del sistema SOURce.

Para comandos de niveles subalternos, la "ruta" o "camino" (árbol jerárquico) tiene que ser especificado, escribiendo por el lado izquierdo el nivel más alto y separando por dos puntos ":" el siguiente nivel.

Ejemplo: [SOURce]:FREQuency:MODE SWEep

Este comando está ubicado en el tercer nivel. Activa el modo de barrido de frecuencia.

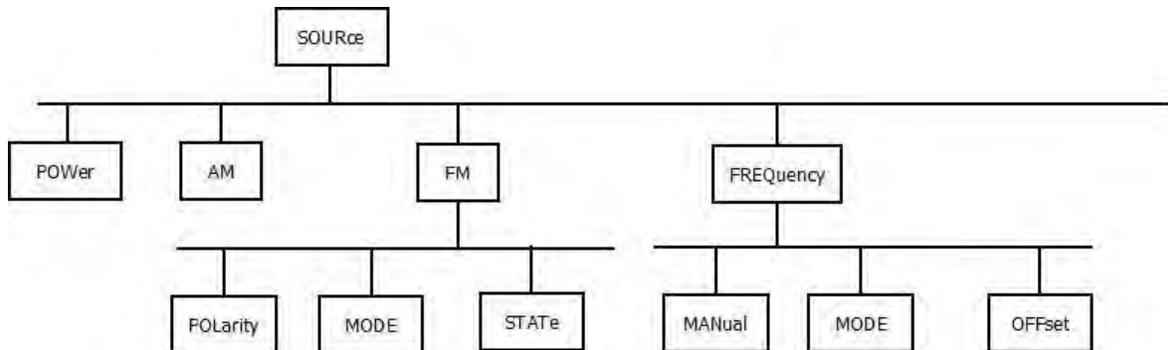


Figura 1.5. Parte de la estructura del árbol de comandos SCPI del bloque SOURce.

Algunas palabras clave se repiten en varios niveles, sin embargo, tienen diferente funcionamiento ya que esta depende del encabezado en el cual sean incluidos.

1.2.5.3 Palabras clave opcionales

Algunas palabras clave pueden ser opcionalmente agregadas a la cabecera de la instrucción. Esta generalmente viene entre corchetes "[]" en el manual de cada instrumento. El instrumento puede interpretar cuando se ha decidido no agregar alguna de estas palabras clave.

Ejemplo: [SOURce]:FREQuency:MODE SWEep sería lo mismo que FREQuency:MODE SWEep

1.2.5.4 Formato largo y corto

Las palabras clave cuentan con un formato largo y uno corto. Tanto el formato largo como el formato corto pueden ser interpretados de manera correcta por un instrumento. Cualquier otra abreviatura es incorrecta.

Ejemplo: **[SOURce]:FREQuency:MODE SWEep** sería lo mismo que **[SOUR]:FREQ:MODE SWE**

El formato corto puede distinguirse ya que se encuentra escrito con letras mayúsculas, el formato largo corresponde a las palabras completas. Las letras mayúsculas son solo la notación con la que se hace referencia a cuales son abreviaturas de las palabras clave, sin embargo, para los instrumentos estas son indistinguibles.

Caracteres	Función
:	Indican que se ha pasado a un nivel jerárquico inferior.
[]	Indica las palabras opcionales.
*	Marca un comando común.
	Separan parámetros opcionales (uno por instrucción).
;	Separa diferentes comandos escritos en una sola línea (la siguiente instrucción puede comenzar desde el ultimo nivel de la anterior).
,	Se utiliza para separar distintos parámetros dentro de una misma línea.
?	Indica que es un comando de consulta. Es necesario que se lea la respuesta del instrumento, ya que podría generar una situación de error.

Tabla 1.3 Tabla de caracteres especiales

1.2.6.5 Comandos comunes y específicos

Hay dos grupos de comando definidos dentro del estándar IEEE 488.2: los comandos comunes son los que se definen obligatoriamente para cualquier instrumento y que ya se han mostrado en la Tabla 1.1. Los comandos específicos son aquellos que son específicos para cierto tipo de instrumentos y que están definidos dentro de la norma SCPI; éstos a su vez están divididos en dos grupos: los obligatorios y los opcionales. Dentro de los obligatorios solo se encuentran los boques SYSTem y STATus.

1.3 Líneas de Transmisión

El bus GPIB está compuesto por 24 líneas de transmisión. Es por estas que el bus gestiona la red de dispositivos conectados, transmite y realiza el control de la transferencia de datos. Sin embargo, 8 de estas líneas son de retorno a tierra. Por lo que la distribución de las líneas restantes sería de la forma siguiente:

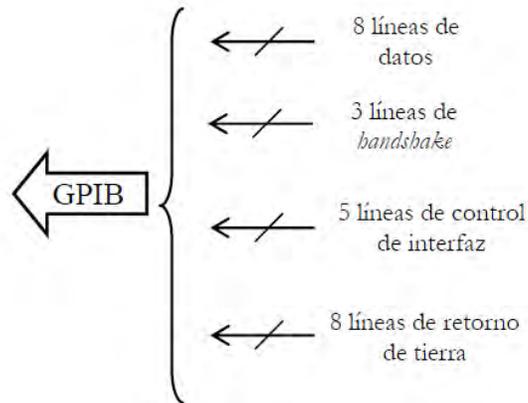


Figura 1.6 Líneas del Bus

Líneas de datos: Están compuestas de 8 líneas bidireccionales, que van del pin DIO1 al DIO8 (Data Input/Output). Transfieren la información de un byte a la vez (1 bit por cada línea); como también comandos. El estado de ATN (una de las 5 líneas de control general) determina si lo que se transmite mediante las líneas de datos son instrucciones o datos.

Líneas de control general de la interfaz o líneas de gestión de bus: Están compuestas por 5 líneas; aseguran el control y la administración del flujo de datos en el bus. Cada línea tiene una función específica como se ve a continuación:

- **ATN (ATtention):** Es usado para la distinción entre datos e instrucciones. Determina el tipo de información en las líneas de datos. Si se encuentra en nivel bajo son instrucciones o direcciones, si está en nivel alto los datos son un comando de instrumento.
- **IFC (InterFace Clear):** Se utiliza para dar un reset a los parámetros y dirección del bus; normalmente usado para inicializar el bus y devolver al estado oyente y escucha inactivo a los instrumentos dentro de la red de comunicación.
- **REN (Remote ENable):** Se utiliza para establecer el control remoto de un instrumento. El control manual, desde el panel del instrumento, ya no será posible para el usuario. Cuando REN es alto los instrumentos bloquean el modo local. Sin embargo, si este cuenta con el botón LOCAL, será la excepción y no estará inhabilitado.
- **EOI (End Or Identify):** Es usado como indicador de que la transferencia total de datos ha terminado.
- **SRQ (Service ReQuest):** Se utiliza para solicitar turno o algún servicio del controlador.

Líneas de control de transferencia de datos (handshake): Estas líneas sirven para garantizar la fiabilidad de los datos al momento de ser transmitidos. Una de las tres líneas es controlada por el hablante, las otras dos por el o los dispositivos escuchas. La manera en la que las líneas de control de transferencia de datos garantizan la fiabilidad de los datos es estableciendo como la velocidad de transmisión la velocidad de transmisión del dispositivo más lento dentro de la red GPIB, logrando que la transmisión sea lo suficientemente lenta para que todos los instrumentos reciban correctamente la información. Están compuestas por las tres líneas siguientes:

- **DAV (Data Valid):** El hablante controla esta línea para hacer saber al o los dispositivos escucha si la información está en condiciones de ser transmitida en el bus.
- **NRFD (Not Ready For Data):** El escucha controla esta línea para hacer saber al dispositivo hablante hasta qué momento debe retener la información para enviarla, se encuentre listo para recibirla.
- **NDAC (Not Data Accepted):** Esta línea está controlada por el dispositivo escucha. El estado de NDAC hace saber al dispositivo hablante si ha recibido correctamente el byte.

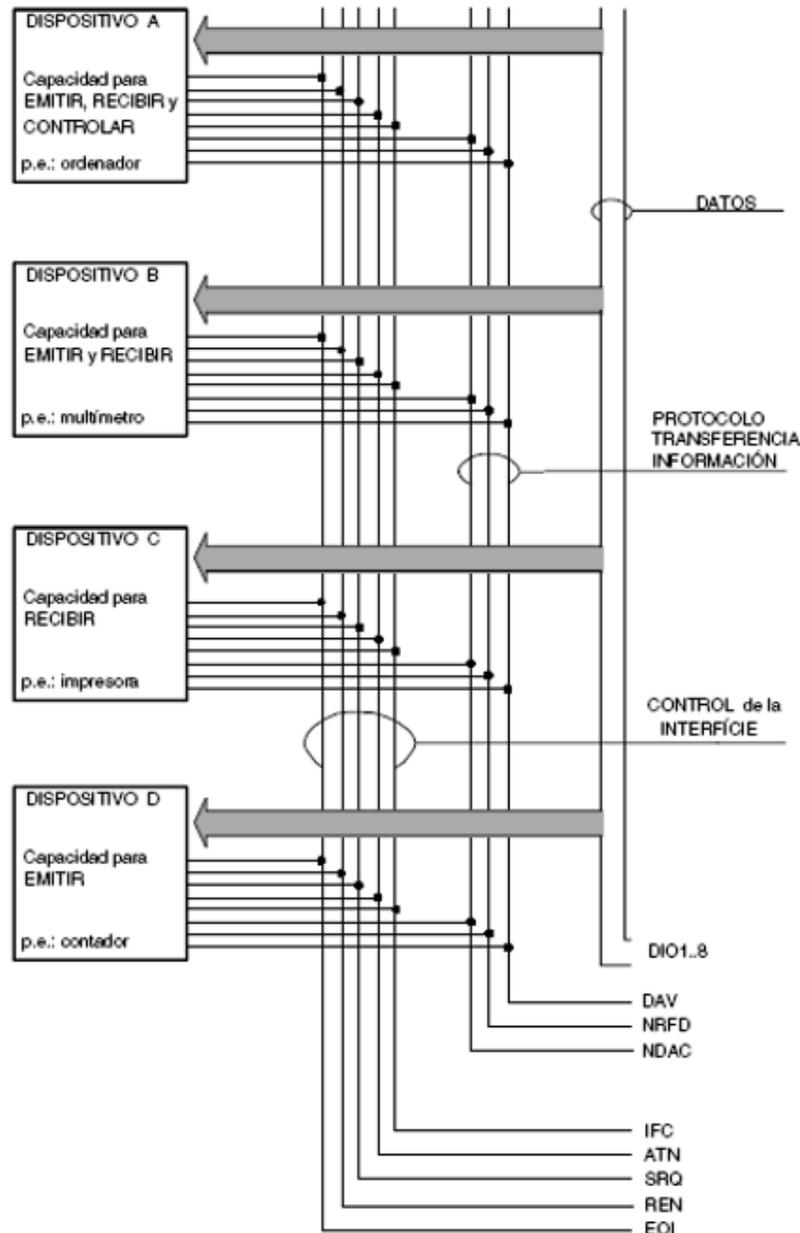


Figura 1.6 Líneas y Señales del bus. (Imagen tomada del trabajo *Aplicación de un software de programación grafica al control de instrumentos vía interfaces GPIB-USB*, Atze Luis Felipe Ruiz Pérez, México 2015)

1.3.1 Secuencia Handshake (desde las Líneas de Control de Transferencia de Datos)

Antes de inicializar la transferencia de datos entre un dispositivo hablante y uno escucha, las Líneas de Control de Transferencia de Datos deben estar en el estado adecuado. Como ya se mencionó la línea DAV es controlada por el dispositivo hablante y las líneas NRFD y NDAC son controladas por los dispositivos escucha. Por lo que el proceso de Handshaking se da de la siguiente manera desde las Líneas de Control de Transferencia de Datos:

1. Inicialmente, el dispositivo hablante mantiene la línea DAV alta, indicando que no hay datos disponibles. Mientras los dispositivos escucha mantienen la línea NRFD alta y la línea NDAC baja, indicando que están listos para recibir datos y que no se aceptan datos, respectivamente.
2. Cuando el dispositivo hablante pone datos en el bus, pone la línea DAV en bajo, lo que indica que los datos son válidos.
3. Los dispositivos escucha establecen la línea NRFD en bajo, lo que indica que no están listos para aceptar nuevos datos.
4. Los dispositivos escucha establecen la línea NDAC en alto, lo que indica que los datos son aceptados.
5. Cuando todos los dispositivos escucha indican que han aceptado los datos, el dispositivo hablante indica mediante la línea DAV que los datos ya no son válidos. Ahora, el siguiente byte de los datos puede ser transmitido.
6. Los dispositivos escucha mantienen la línea NRFD alta indicando que están listos para recibir datos nuevamente, y la línea NDAC se mantiene alto para indicar que son aceptados los datos.

Si la línea ATN es alta durante el proceso handshake, la información es considerada como los datos de un comando de instrumento. Si la línea ATN es baja, la información se considera un mensaje de interfaz GPIB.

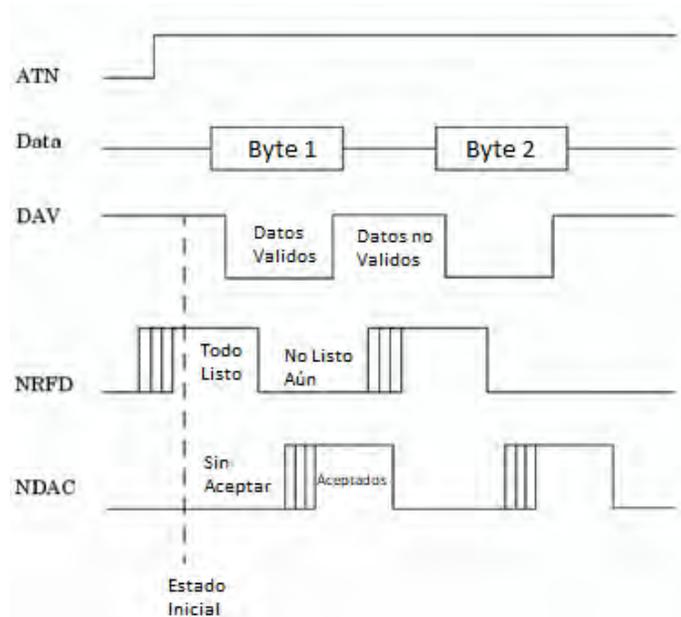


Figura 1.7 Pasos del Handshaking.

1.4 Topología de conexión

Una red de transmisión de datos que utiliza el bus GPIB para que alcance su velocidad óptima (hasta 8 Mbytes/s), debe cumplir con lo siguiente:

- Debe haber un número máximo de 15 dispositivos conectados en el bus, y deben estar encendidos al menos dos tercios de ellos.
- La distancia, entre dos dispositivos deben ser máximo de 4 metros (la longitud de bus entre dos dispositivos).
- La longitud máxima total no debe ser mayor a los 20 metros.

Existen esencialmente dos configuraciones para crear una red basada en el bus GPIB: topología lineal, estrella e híbrida que es la combinación de las dos anteriores.

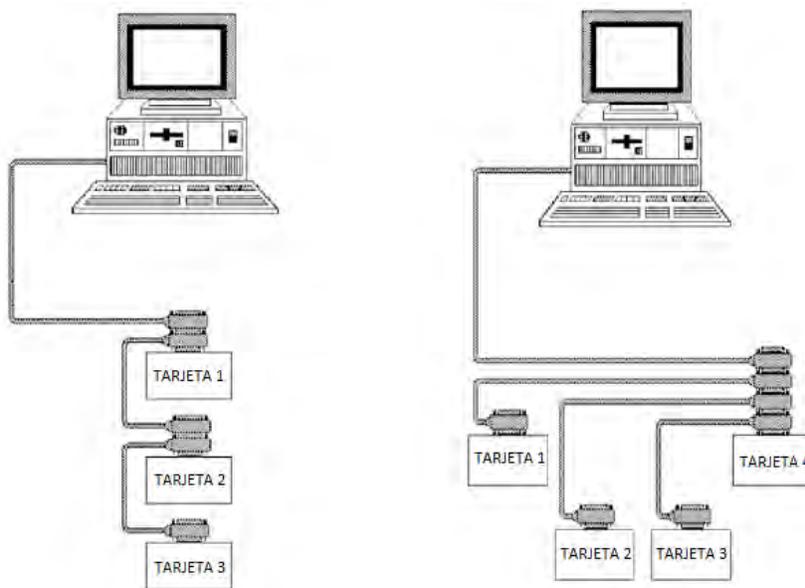


Figura 1.8 Topología lineal (izquierda) y topología de estrella (derecha).

1.5 Referencias

- [1] ¿Qué es GPIB?, National Instruments, 2016, <http://www.ni.com>
[2] *Operating Manual EMI TEST RECEIVER*, Rohde&Schwarz Munich 2001
[3] *Aplicación de un software de programación grafica al control de instrumentos vía interfaces GPIB-USB*, Atze Luis Felipe Ruiz Pérez, México 2015

Capítulo 2: Instrumentos

Para el desarrollo de esta tesis fue necesario el uso de ciertos instrumentos de medición, que serán ampliamente descritos en este capítulo; unos por ser necesarios para llevar a cabo los experimentos de “Ionización de Gases por medio de Radiofrecuencia”; otros para entender el funcionamiento general del bus GPIB y su protocolo, y extender las funcionalidades de la interfaz desarrollada. Todos éstos se encuentran implementados dentro del conjunto de instrumentos que componen la Cámara Semianecoica de la Facultad de Ciencias.

2.1 Multi-Device Controller

El Multi-Device Controller, Modelo 2090, no es como tal un instrumento de medición ni generador de algún tipo de señal. Sin embargo, es parte fundamental en las cámaras del tipo semianecoica. Creado por ETS-EMCO, el Multi-Device Controller es utilizado para el posicionamiento de dispositivos tales como torres con antena, mesas rotatorias, paneles de reverberación, etc. Acopla los dispositivos, cambiándolos de posición, para el ajuste de una variedad de dispositivos.

El controlador permite al usuario sincronizar el movimiento de manera simultánea de dos dispositivos tales como una torre con antena y mesa rotatoria, como es el caso de la Cámara Semianecoica de la Facultad de Ciencias; permitiendo el control tanto en modo manual como en modo remoto vía GPIB.

Cada dispositivo primario está interconectado al controlador (Modelo 2090) a través de una interfaz bidireccional de fibra óptica, utilizando comandos de un protocolo propietario de ETS-EMCO. Las Líneas de comunicación de fibra óptica que conectan los dispositivos a la unidad eliminan cualquier interferencia de RF que normalmente pueden ser conducidas a través de un cable de alambre.

El panel frontal del Modelo 2090 proporciona un panel de control completo para cada dispositivo; cada uno con un display numérico, indicador de estado y botones de función idénticos. Los botones de función permiten al usuario configurar los dispositivos como especificar parámetros y límites, ajustes de posición, y el control de movimiento de los dispositivos. El display y el indicador de estado muestran la posición y la información de la operación del proceso (la posición en la que se encuentra).

2.1.1 Funcionamiento

El Multi-Device Controller es usado para controlar la torre y la mesa rotatoria durante pruebas de emisión de radiación, sea de manera manual o totalmente automatizada mediante rutinas. El controlador es capaz de manejar las funciones de la altura de la torre y la polarización de la antena así como la función de rotación de la mesa desde la cabina de operación de la Cámara Semianecoica. Esta capacidad satisface los requerimientos técnicos de medición para alcanzar en azimut* el rango de altura especificado para detectar el máximo de la fuente de señal del equipo bajo prueba.

Y aunque el Modelo 2090 tenga más aplicaciones ésta es la única necesaria para llevar a cabo los experimentos. Solo se expondrán las capacidades del instrumento de acuerdo a las características de la Cámara Semianecoica de la Facultad de Ciencias, UNAM.

2.1.2 Descripción del Panel Frontal.

En la figura 2.1 se muestra en detalle el panel frontal de Multi-device controller.

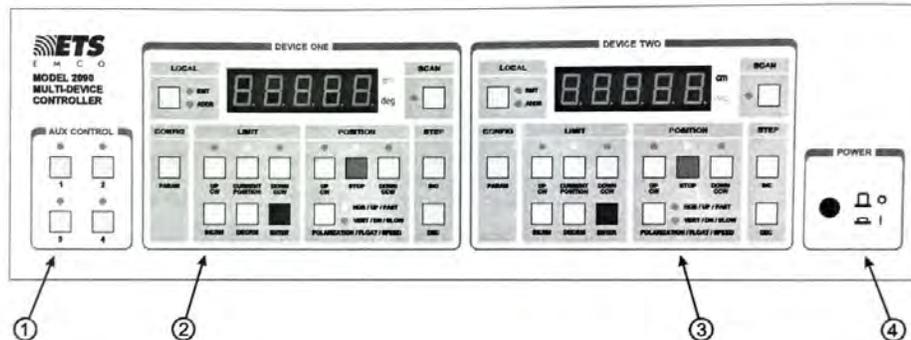


Figura 2.1. Panel frontal del Modelo 2090. (Imagen tomada del manual de operación del *Multi-Device Controller Model 2090*, EMC Test Systems, 2001)

1. **Bloque de control auxiliar:** Pulsando alguno de los cuatro botones permite activar o desactivar la correspondiente salida auxiliar e indicador de uno de los cuatro puertos de control auxiliar.
2. **Bloque de control del dispositivo uno:** El panel de control de dispositivo primario 1.
3. **Bloque de control del dispositivo dos:** El panel de control de dispositivo primario 2.
4. **Bloque de control de Encendido:** El único botón del bloque es el de encendido y apagado del Modelo 2090.

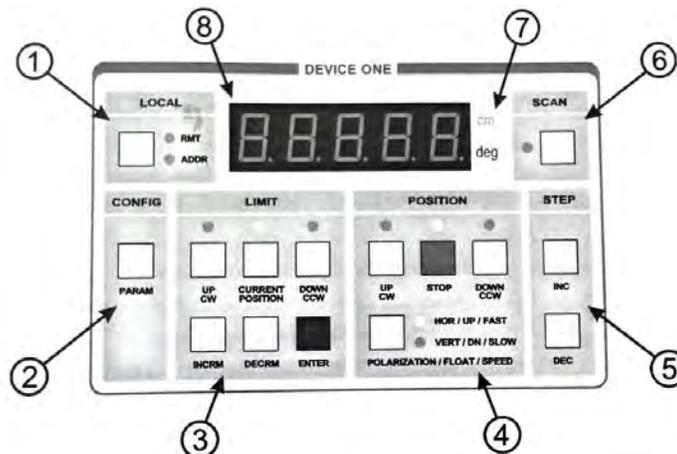


Figura 2.2. Bloque de control del dispositivo del Modelo 2090. (Imagen tomada del manual de operación del *Multi-Device Controller Model 2090*, EMC Test Systems, 2001)

2.1.2.1 Botón Local e Indicadores (1)

Botón LOCAL: El botón LOCAL es multifuncional, ya que dependiendo del estado del controlador (Multi-device controller) su función varía.

- a. Cuando el dispositivo primario se encuentra en modo de control remoto (se enciende el indicador RMT) y si el botón es presionado el controlador pasa a modo de control local.

b. Para un dispositivo de velocidad variable, al presionar el botón LOCAL, mientras este esté en modo local y la posición de muestra en el display, se activará el modo de selección de velocidad la cual podrá ser incrementada o disminuida con los botones INCRM y DECRM.

c. En el modo de edición, al presionar el botón LOCAL incrementara el dígito que se está visualizando en el display al siguiente dígito más grande.

Indicador RMT (Remote): El indicador estará encendido mientras el dispositivo primario correspondiente esté bajo el control remoto GPIB. El panel frontal será desactivado mientras se encuentre en modo remoto.

Indicador ADDR (Addressed): Este indicador se mantendrá encendido mientras el control remoto GPIB este activo.

2.1.2.2 Configuración (2)

Botón PARAM: Detiene el movimiento de todos los dispositivos y muestra la entrada actual de la lista de parámetros de configuración para el dispositivo primario en su correspondiente display. Presionar repetidamente este botón puede avanzar al siguiente parámetro de la lista.

2.1.2.3. Botones de Límite e Indicadores (3)

Botón UP/CW: Al presionar el botón se mostrará el límite superior para la polarización actual de la torre, o en el caso de la mesa rotatoria el límite de los grados que puede rotar en el sentido del reloj.

Indicador UP/CW: El indicador se enciende para indicar que en el display se está mostrando el límite superior de la torre o de los grados que la mesa rotatoria puede rotar en el sentido del reloj.

Botón de CURRENT POSITION: Al presionar este botón se mostrará la posición actual del dispositivo primario correspondiente.

Indicador CURRENT POSITION: El indicador se enciende para indicar que en el display se está mostrando la posición actual del dispositivo primario. Este es el estado por defecto.

Botón DOWN/CCW: Al presionar este botón se mostrará el límite superior para la polarización de la torre, o en el caso de la mesa rotatoria el límite de los grados que puede rotar en contra del sentido del reloj.

Indicar DOWN/CCW: Se enciende para indicar que en el display se muestra el límite de la torre o en el caso de la mesa rotatoria el límite de los grados que puede rotar en contra del sentido del reloj.

Botón ENTER: Al presionar este botón se activa o desactiva el modo de edición. Cuando el modo de edición se encuentra desactivado al presionar ENTER, iniciará el modo de edición para el parámetro que se muestra actualmente. Si el modo de edición se encuentra activado, al presionar ENTER se guardaran las modificaciones hechas y se saldrá del modo de edición.

Botón INCRM: Al presionar este botón incrementará el límite de posición, parámetro o velocidad en el display. En el caso del Modelo 2090 no se encuentre en el modo de edición, al presionar INCRM este automáticamente será activado.

Botón DECRM: Al presionar este botón decrementará el límite de posición, parámetro o velocidad en el display. En el caso del Modelo 2090 no se encuentre en el modo de edición, al presionar INCRM éste automáticamente será activado.

2.1.2.4 Botones de Posición e Indicadores (4)

Botón UP/CW: Al presionar esta tecla ordenará al dispositivo asociado moverse hacia arriba, en el caso de la torre o rotar en el sentido del reloj, en el caso de la mesa rotatoria.

Indicador UP/CW: Se enciende para indicar que el dispositivo se está moviendo hacia arriba, en el caso de la torre o rotando en el sentido del reloj, en el caso de la mesa rotatoria.

Botón STOP: Al presionar este botón se detendrá el movimiento del dispositivo asociado. En el caso de los dispositivos de velocidad variable, se desacelerará hasta la velocidad mínima antes de detenerse.

Indicador STOP: Se enciende para indicar que el dispositivo ha sido detenido. Parpadea para confirmar que la operación está siendo ejecutada mientras el dispositivo desacelera.

Botón DOWN/CCW: Al presionar esta tecla ordenará al dispositivo asociado moverse hacia abajo, en el caso de la torre o rotar en contra del sentido del reloj, en el caso de la mesa rotatoria.

Indicador DOWN/CCW: Se enciende para indicar que el dispositivo se está moviendo hacia abajo en el caso de la torre o rotando en contra del sentido del reloj, en el caso de la mesa rotatoria.

Botón POLARIZATION/FLOAT/SPEED: Este botón puede comportarse de manera diferente dependiendo del tipo dispositivo:

a. Para la torre, cuando el display muestra la posición actual cambia la polarización del brazo alternando entre horizontal y vertical. Cuando se muestra un límite inferior o superior, alterna, mostrando en el display, entre límite, correspondiente, horizontal o vertical.

b. Para la mesa rotatoria estándar de velocidad variable, al presionar este botón avanza al haber seleccionado la configuración de velocidad a la siguiente preajustada seleccionada.

Indicador HOR/UP/FAST: Este indicador puede comportarse de manera diferente dependiendo del tipo dispositivo vinculado:

a. Para la torre, el indicador se enciende para indicar que la torre está polarizada horizontalmente. Este indica solo el estado de la polarización de la bobina solenoide por lo que podría o no reflejar el estado de polarización del brazo de la torre a cualquier instante. El indicador parpadea para señalar que el dispositivo está mostrando el límite superior o inferior de la polarización.

b. Para la mesa rotatoria estándar de velocidad variable el indicador se enciende de forma binaria en conjunto con el indicador VERT/DN/SLOW para representar los bits menos significativos de la velocidad actual seleccionada. Velocidad 1 o 5: Ambos apagados; velocidad 2 o 6: HOR/UP/FAST encendido y VERT/DN/SLOW apagado; velocidad 3 y 7: HOR/UP/FAST apagado y VERT/DN/SLOW encendido; velocidad 4 u 8: ambos encendidos

Indicador VERT/DN/SLOW: Este indicador puede comportarse de manera diferente dependiendo del tipo dispositivo vinculado:

a. Para la torre, el indicador se enciende para indicar que la torre está polarizada verticalmente. Este indica solo el estado de la polarización de la bobina solenoide por lo que podría o no reflejar el estado de polarización del brazo de la torre a cualquier instante. El indicador parpadea para indicar que el dispositivo está mostrando el límite superior o inferior de la polarización vertical mientras esté aún está polarizado horizontalmente.

b. Para la mesa rotatoria estándar de velocidad variable el indicador se enciende de forma binaria en conjunto con el indicador HOR/UP/FAST para representar los bits menos significativos de la velocidad actual seleccionada. Velocidad 1 o 5: Ambos apagados; velocidad 2 o 6: HOR/UP/FAST encendido y VERT/DN/SLOW apagado;

velocidad 3 y 7: HOR/UP/FAST apagado y VERT/DN/SLOW encendido; velocidad 4 u 8: ambos encendidos.

2.1.2.5 Botones de Paso (5)

Botón INC: Tiene dos funciones:

a. Al presionar este botón mientras la posición se visualiza, se ordenará al dispositivo primario asociado moverse hacia arriba, en el caso de la torre, o rotar en el sentido del reloj, en el caso de la mesa rotatoria, hasta que el botón deje de ser presionado.

b. Al presionar esta tecla mientras se visualiza la lista de parámetros se asciende un lugar dentro de la lista de parámetros, permitiendo desplazarse en ella.

Botón DEC: Tiene dos funciones:

a. Al presionar este botón mientras la posición se visualiza, se ordenará al dispositivo asociado moverse hacia abajo, en el caso de la torre, o rotar en contra del sentido del reloj, en el caso de la mesa rotatoria, hasta que el botón deje de ser presionado.

b. Al presionar esta tecla mientras se visualiza la lista de parámetros se desciende un lugar dentro de la lista de parámetros, permitiendo desplazarse en ella.

2.1.2.6 Botón Scan e Indicador (6)

Botón SCAN: Activa y desactiva el modo SCAN del dispositivo. Cuando el modo SCAN está activado el dispositivo primario se moverá entre los límites superior e inferior, respecto a la altura si es la torre, y en sentido del reloj si es la mesa rotatoria; se moverá un límite de ciclos (definidos en la lista de parámetros) o hasta que se le dé la instrucción de detenerse.

Indicador SCAN: Este indicador se mantiene encendido tanto tiempo como el modo SCAN se encuentre activado.

2.1.2.7 Indicadores de Unidad (7)

Indicador cm: Este indicador se enciende para dispositivos configurados como torres, para indicar que la posición y la información sobre los límites está dada en centímetros.

Indicador deg: Este indicador se enciende para dispositivos configurados como mesas rotatorias, para indicar que la posición y la información sobre los límites está dada en grados

2.1.2.8 Display Principal (8)

El display de 5 dígitos (de siete segmentos cada uno) provee información al usuario de la posición actual del instrumento, así como la configuración de límites, parámetros, velocidad, mensajes de error y otros mensajes alfanuméricos.

2.1.3 Operación remota del Modelo 2090 mediante el uso de comandos GPIB

El Multi-Device Controller (Modelo 2090) incorpora un conjunto de comando de dispositivos anteriores, como los Modelos 1050, 1060, y 1090; en conjunto con nuevos comandos.

El dispositivo Modelo 2090 contiene solo un puerto GPIB el cual ha sido configurado para aceptar dos direcciones separadas. Las direcciones válidas, como ya se había abordado en el capítulo anterior, están en el rango de 0 a 30, aunque normalmente el 0 está reservado para el controlador a cargo o la computadora (Controller-In-Charge). No hay un criterio como tal para configurar las direcciones, solo que deben ser diferentes entre ellas, pero es

recomendable que la dirección del Dispositivo 1 sea una dirección abajo que la del dispositivo 2. Las direcciones configuradas por defecto son 8 y 9. Y son con éstas las que se trabajaron.

2.1.3.1 Operación remota Torre

El instrumento referenciado al Dispositivo 1 del Multi-Device Controller es la Torre. Ésta mantiene la dirección GPIB número 8, asignada ya por defecto.

A continuación, se mostrarán una serie de comandos que en el orden dado ponen en funcionamiento la Torre de forma remota, como también una serie de instrucciones que son básicas para su control.

En el capítulo siguiente se explicará con mayor detenimiento las funciones utilizadas para enviar comandos a la Torre. El comando expuesto a continuación para el Multi-Device Controller Modelo 2090 son definidos en el Apéndice A

Conectado el Multi-Device Controller y el dispositivo Controlador a la red de comunicación GPIB:

- Inicializamos el descriptor de la Torre. Con este se identificará en adelante al dispositivo y será utilizado para las llamadas posteriores.
 - `Id = ibdev(0, 8, 0, T10s, 1, 0)`
- Limpiamos las funciones de la Torre.
 - `lbclr(Id)`
- Establecemos el límite inferior a 100 cm, y el límite superior de la Torre a 400cm.
 - `ibwrt(Id, "LL 100", 6)`
 - `ibwrt(Id, "UL 400", 6)`
- Enviamos la Torre a su límite inferior.
 - `ibwrt(Id, "DN", 2)`
- Preguntamos si la Torre ha completado la operación con `*OPC?`. Si el dispositivo responde 1 ha terminado, si responde 0 sigue en movimiento.
 - `ibwrt(Id, "*OPC?", 5)`
 - `ibrd(Id, ValStr, ValStr.Lenght)`

Debe tomarse en cuenta que estas instrucciones, como cualquier otra que cambie de posición la Torre, deben agregarse dentro de iteración que constantemente este verificando el estado de la Torre, como también un *delay* que permita una ventana de 0.1 segundos, esto según las especificaciones del manual de Modelo 2090.

- Una vez terminada la operación, cambiamos a la posición X centímetros la Torre, en este caso a 350 centímetros.

- ibwrt(Id, "SK 350", 6)
- Si se quisiera saber la posición actual de la Torre mientras cambia de posición debe enviársele el comando CP?.
 - ibwrt(Id, "CP?", 3)

2.1.3.2 Operación remota Mesa Rotatoria

El instrumento referenciado al Dispositivo 2 del Multi-Device Controller es la Mesa Rotatoria. Esta mantiene la dirección GPIB número 9, asignada ya por defecto.

A continuación, se mostrará una serie de comandos que en el orden dado muestran como poner en funcionamiento la Mesa Rotatoria de forma remota, como también una serie de instrucciones que son básicas para su control.

En el capítulo siguiente se explicará con mayor detalle funciones utilizadas para enviar comandos a la Mesa. Los comandos expuestos a continuación para el Multi-Device Controller Modelo 2090 son definidos en el Apéndice A.

Conectado el Multi-Device Controller y el dispositivo Controlador a la red de comunicación GPIB:

- Inicializamos el descriptor de la Mesa Rotatoria. Con este se identificará en adelante al dispositivo y será utilizado para las llamadas posteriores.
 - Id = ibdev(0, 9, 0, T10s, 1, 0)
- Limpiamos las funciones de la Mesa.
 - Ibclr(Id)
- Establecemos el límite inferior a 0 grados, y el límite superior de la Mesa, al rotar, a 359 grados.
 - ibwrt(Id, "CL 0", 4)
 - ibwrt(Id, "WL 359", 6)
- Enviamos la Mesa a su límite inferior.
 - ibwrt(Id, "CC", 2)
- Preguntamos si la Mesa ha completado la operación con *OPC?. Si el dispositivo responde 1 ha terminado, si responde 0 sigue en movimiento.
 - ibwrt(Id, "*OPC?", 5)
 - ibrd(Id, ValStr, ValStr.Lenght)

- Debe considerarse el mismo procedimiento que en la Torre para verificar el estado de la Mesa Rotatoria cuando cambia de posición.
- Una vez terminada la operación, cambiamos la Mesa a la posición X grados, en este caso a 180 grados.
 - `ibwrt(Id, "SK 180", 6)`
- Si se quisiera saber la posición actual de la Mesa Rotatoria mientras cambia de posición debe enviársele el comando CP?.
 - `ibwrt(Id, "CP?", 3)`

2.2 Amplificador de Potencia

De los dos amplificadores de potencia con los que cuenta la Cámara Semianecoica se utilizó el Amplificador de Potencia Modelo 250W1000AM1 de la marca *Amplifier Research*. Esto debido a que se encuentra intervalo de frecuencias de ionización en que se investiga.

2.2.1 Funcionamiento

La operación del amplificador Modelo 250W1000AM1 es bastante simple. La señal de entrada del amplificador, ya sea frecuencia de barrido o fija, es alimentado por el puerto de entrada "RF INPUT", y la señal de salida del amplificador es dada por el puerto de salida "RF OUTPUT"; ambos ubicados en el panel trasero. El amplificador se enciende pulsando, desde el panel frontal, el botón POWER. En el caso de que se produzca un error se cuenta con un *switch* para interrumpir el funcionamiento ubicado en la parte trasera de la unidad.

2.2.2 Descripción del Panel Frontal

En la **figura 2.3** se muestra en detalle el panel frontal de Amplificador 250W1000AM1.

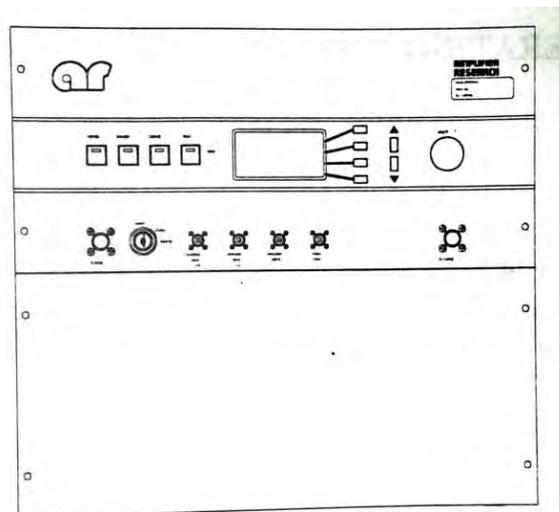


Figura 2.3. Panel frontal de Modelo 250W1000AM1. (Imagen tomada del manual de operación y servicio del Model 250W1000AM1, Amplifier Research, USA 2002)

2.2.2.1 Interruptor de bloqueo bajo llave (Keylock Switch)

El interruptor de bloqueo bajo llave sirve para brindar protección de un uso no autorizado o de un inesperado manejo remoto del amplificador. El amplificador sólo puede ser controlado desde el panel de control frontal cuando el interruptor se encuentra en la posición LOCAL. De la misma manera, el amplificador sólo puede ser controlado remotamente cuando el interruptor se encuentra en la posición REMOTE. Al posicionar el interruptor en INHIBIT el instrumento queda en modo espera y es imposible la operación de este.

2.2.2.2 Interruptor POWER

El interruptor POWER enciende o apaga la alimentación principal del amplificador. Cuando el dispositivo se encuentra encendido el led que del interruptor se ilumina de color verde. Una vez encendido el amplificador el display de 4 líneas es activado.

2.2.2.3 Interruptor STANDBY

El interruptor STANDBY cambia a modo espera el amplificador cuando éste se encuentra en modo operación. Este no es un reinicio. El led amarillo del interruptor se ilumina cuando la unidad se encuentra en modo espera; éste solo se encuentra apagado cuando la alimentación del amplificador está apagada o cuando está en modo operación. Mientras el amplificador se encuentre en este modo no es posible ninguna salida RF.

2.2.2.4 Interruptor OPERATE

El interruptor OPERATE pone el amplificador en modo operación. Mientras el amplificador encuentre en este modo, el led del interruptor OPERATE se encenderá y la salida de RF será posible.

2.2.2.5 Interruptor RESET

El interruptor RESET intenta restaurar el funcionamiento normal del amplificador. Cuando una falla ha ocurrido el led rojo del interruptor RESET se encenderá indicando que ha habido una falla en el amplificador. Al presionar el botón RESET podría corregir la falla, sin embargo, dependiendo del tipo de falla podría no ser así. Si el amplificador se encuentra en una condición de sobre carga, disminución la ganancia del amplificador o el umbral (threshold) ALC o el nivel de entrada RF debería habilitarse el interruptor para restablecer el amplificador (i.e., el dispositivo pasará al modo espera). Si el interruptor RESET es presionado y el led no se apaga. La falla podría ser eléctrica por lo que habrá que remitirse al manual del amplificador Modelo 500A100A.

2.2.2.6 Alarma de sobrecarga RF

Esta Alarma está localizada detrás de la tapa del panel frontal, se activa si la potencia de salida del amplificador (amplifier's forward) excede los 280 watts o si la potencia reflejada del amplificador (amplifier's reflected) excede los 250 watts.

2.2.3 Operaciones del Panel de Control.

Las operaciones descritas a continuación asumen que el usuario realizará dichas operaciones desde el panel frontal del amplificador con el interruptor de bloqueo bajo llave en la posición LOCAL. La computadora accede a los comandos y a las funciones de estatus cuando el interruptor bajo llave se encuentra en la posición REMOTE.

El display de 4 líneas fluorescente se localiza en el centro del panel frontal del amplificador (figura 2.3). Los botones POWER, STANDBY, OPERATE, y RESET están localizados a la izquierda del display. El interruptor de seguridad bajo llave se encuentra abajo a la izquierda de estos botones. Inmediatamente a la derecha se encuentran cuatro botones (conocidos como Soft-Keys) con líneas dirigidas al display. Estos cuatro botones tienen diferentes funciones dependiendo de cada pantalla en la que se encuentre el display. Sirven para navegar en las opciones del display. A la derecha de estos botones existen dos adicionales UP y DOWN para incrementar o decrementar los valores de las variables del amplificador. Y al final a la derecha se encuentra una perilla de ajuste. Esta perilla suele utilizarse para ajustar también la mayoría de las variables. La sensibilidad de la perilla cambia para los diferentes valores que está ajusta.

Cada pantalla está acomodada conforme a un patrón similar. A la derecha de cada pantalla hay etiquetas que cambian dependiendo a la pantalla en la que el usuario se encuentre. Estas etiquetas tienen asociadas uno de los botones que están inmediatamente a la derecha del display (soft-key). La función de cada botón depende de la etiqueta adyacente asociada. En la mayoría de las pantallas se incluye a la izquierda en las dos líneas últimas inferiores (líneas del display) el nivel de potencia de salida (FORWARD) y potencia reflejada (RESERVE).

Las principales pantallas de operación son las siguientes:

- Main - Menú inicial
- Manual - Configuraciones del nivel de salida RF para el modo Manual
- Pulse - Configuraciones del nivel de salida RF de un impulso cerrado para el modo Pulso
- ALC - Configuración del nivel de salida RF para el modo Control de Nivel Automático
- Status - Observar detalles de los fallos.

La navegación de pantallas se hace a través de las soft-keys asociadas a la etiquetas MORE y BACK.

2.2.4 Selección del modo de operación del Amplificador

Antes de que al amplificador le sea dada la instrucción de empezar a operar, el usuario debe seleccionar el modo de operación mediante el cual el amplificador debe funcionar.

Los modos de operación son los siguientes:

- Manual
- Pulse (Entrada de Pulso Externo)
- ALC Interno (Control de nivel automático con detector interno)
- ALC Externo (Control de nivel automático con detector externo)

2.2.4.1 Modo Manual

En el modo manual, el nivel de salida RF es controlado por el nivel de la unidad de entrada (*the input drive level*) y el control de ganancia RF (*the RF Gain Control*). Este modo nos proporciona un control automático del nivel de salida del amplificador distinto al implementado por el circuito de protección del amplificador.

2.2.4.2 Modo Pulso

En el modo Pulso, el nivel de salida de RF del amplificador es controlado por el nivel de la unidad de entrada (*the input drive level*) y el control de ganancia de RF (*the RF Gain Control*). El puerto PULSE INPUT del panel frontal puede ser usado para controlar mediante pulsos la salida RF aplicando una señal TTL pulsada. Si el pulso está en un bajo voltaje, el RF está desactivado.

2.2.4.3 Modo de Control de Nivel Automático (CNA)

En estos modos, el amplificador es operado en un modo de “nivelación automática”, esto significa que es manejado por sus controles de límite (THRESHOLD), ganancia RF (RF GAIN), y ganancia del detector (DETECTOR GAIN). La entrada ALC (retroalimentación) para este modo puede ser Interna (INT) o Externa (EXT).

2.2.5 Ajustes del Control de Ganancia RF (RF GAIN Control)

El control de ganancia RF, el cual tiene como valor mínimo 18 dB, es usado para ajustar el nivel de entrada para el módulo de nivelador-preamplificador (*preamplifier-leveler module*). En el modo Manual o Pulso, simplemente ajusta el nivel de potencia de salida para un nivel de potencia de entrada fijo. En el modo de Control de Nivel Automático, este control debe ajustarse para permitir al amplificador alcanzar la potencia de salida deseada.

2.2.6 Ajustes de Control del Limite (THRESHOLD Control)

Este control es usado para ajustar el nivel de la señal de salida de RF, solo en el modo de Control de Nivel Automático. La nivelación no ocurrirá si la potencia de entrada de RF del amplificador no es suficiente. El Control de Límite no es utilizado cuando el amplificador se encuentra configurado en los modos Manual o Pulso.

2.2.7 Control de Ganancia del Detector (DETECTOR GAIN Control)

Este control es usado para compensar las variaciones de ganancias del ciclo de Control de Nivel Automático, causado por diferencia en los detectores que pueden ser usados para proporcionar retroalimentación de nivel al ciclo de nivelación automático.

2.2.8 Control de Respuesta del CNA (ALC RESPONSE Control)

Es utilizado para ajustar la respuesta en frecuencia del ciclo CNA. Es particularmente útil si la entrada al amplificador es una señal de barrido. Ajustando la velocidad de ciclo CNA, se podría obtener un amortiguamiento crítico, sin oscilación. El nivel 1 es el tiempo constante más rápido y el 6 el más lento.

2.2.9 Operación Remota

La operación remota del Modelo 250W1000AM1 puede ser desde 2 tipos de interfaces: IEEE-488 o RS-232. Para elegir la interfaz IEEE-488, que es el que hemos trabajado, primeramente, se necesita asegurar que el Interruptor de Bloqueo Bajo Llave se encuentre en posición LOCAL. Ir a la pantalla de servicio (Service).

Hacia arriba a la derecha se puede notar la etiqueta que dice 488 COMM, correspondiente a la interfaz IEEE-488, o 232 COMM, correspondiente a la interfaz RS-232. Para que el amplificador pueda ser manejado remotamente desde la interfaz IEEE-488 debe aparecer escrita 488 COMM en el display. Para cambiar de una interfaz a otra solo se necesita presionar la soft-key 1 en la pantalla de servicio.

La interfaz remota del Modelo 250W1000AM1 permite su manejo de todas las funciones del amplificador vía remota, desde una computadora, con excepción del Interruptor de Bloqueo Bajo Llave que debe ser controlada directamente desde el panel frontal. Cuando el amplificador se encuentra en modo remoto una configuración especial de pantalla se muestra en el display. Los controles del panel frontal son desactivados con excepción del Interruptor de Bloqueo Bajo Llave.

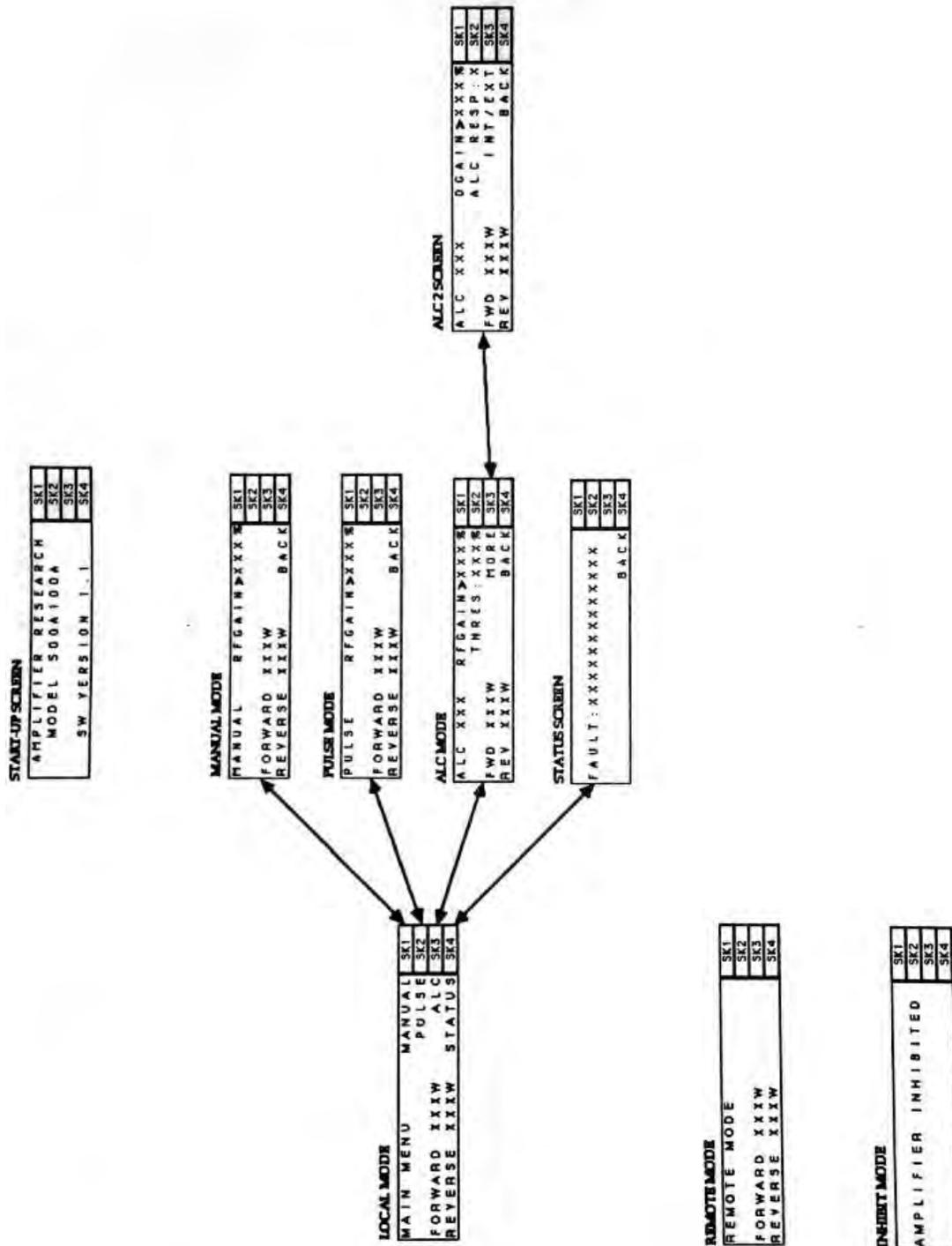


Figura 2.4 Pantallas del display del Amplificador Modelo 250W1000AM1. (Imagen tomada del manual de operación y servicio del Model 250W1000AM1, Amplifier Research, USA 2002)

2.2.9.1 Dirección IEEE-488 del amplificador

Para configurar la dirección vamos a la pantalla SERVICE (servicio). Hay que asegurarse que el bus de comunicación sea 488. Debajo de 488 COMM se podrá ver la dirección

seguida de la palabra ADDRESS. Al presionar SK2 (soft-key 2) se habilitará la edición de esta. Con la perilla puede ser seleccionada una dirección en el rango de 1 a 31. Para confirmar la dirección elegida solo hace falta volver a la pantalla principal.

2.2.9.2 Formato del conjunto de comandos

Cada comando está compuesto de al menos un carácter alfanumérico, hasta cuatro numéricos, y un carácter de terminación de comando. El carácter de terminación de comando es denotado como <LF>. Los comandos deben terminarse con este carácter escrito con mayúsculas y en ese orden estricto para que puedan ser reconocidos. Ningún comando es permitido después de que sea escritor <LF>. El Modelo 250W1000AM1 interpreta los caracteres siguientes a <LF> como el inicio de un nuevo comando.

De la misma manera puede utilizarse el control de línea “*End or Identify*” (EOI), que funciona de la misma manera sin la necesidad de agregarlo explícitamente a la instrucción enviada. Esto es definido al inicializar el descriptor de un instrumento (véase el Capítulo 3). Si se agregan ambos surte el mismo efecto.

2.2.9.3 Instrucciones al Amplificador

A diferencia de los demás instrumentos que en este capítulo se describen, el Amplificador 250W1000AM1 no obedece el estándar 488.2 por lo que no comprende los SCPI como tampoco los comandos comunes. El amplificador se encuentra bajo el estándar 488.1. Y aunque su funcionamiento se limita a ciertas funciones la comunicación con el amplificador no es la deseada. Estos problemas se abordarán en el Capítulo 4.

La dirección GPIB del amplificador es 11. A continuación se mostrarán los comandos utilizados en el Experimento de Ionización de Gases por RF.

En el capítulo siguiente se aborda con mayor detalle las funciones utilizadas para enviar comandos al Amplificador. Los comandos expuestos a continuación para el Modelo 250W1000AM1 son definidos en el Apéndice B

Conectado el Amplificador y el dispositivo Controlador a la red de comunicación GPIB:

- Inicializamos el descriptor de la del Amplificador 250W1000AM1. Con éste se identificará en adelante al dispositivo y será utilizados para las llamadas posteriores.
 - `Id = ibdev(0, 11, 0, T10s, 1, 0)`
- Limpiamos las funciones del Amplificador.
 - `lbclr(Id)`
- Encendemos el Amplificador.
 - `li.ibwrt(Id, "P1<LF>", 6)`
- Cambiamos la Ganancia del Amplificador a 2047: el 50% de la ganancia.
 - `li.ibwrt(Id, "G2047<LF>", 9)`

- Cambiamos a modos OPERATE el Amplificador. Comienza la emisión de la onda amplificada.
 - li.ibwrt(Id, "O<LF>", 5)
- Cambiamos a modo STANDBY para dejar de emitir.
 - li.ibwrt(Id, "S<LF>", 5)
- Apagamos el Amplificador.
 - li.ibwrt(Id, "P0<LF>", 6)

2.3 Generador de señales SML03 (9kHz a 3.3 GHz)

El generador de señales SML03 del fabricante Rohde & Schwarz, con el que se cuenta en la Cámara Semianecoica opera en el rango de frecuencia de 9kHz a 3.3 GHz. La complejidad de este instrumento, por su amplia gama de funciones, fue imposible abordar su descripción completa en este trabajo. Por esta razón solo se describirán ciertas funciones y características del Modelo SML03 que corresponden a su uso dentro del experimento de ionización de gases mediante RF.

2.3.1 Descripción del Panel Frontal

En la figura 2.5 se muestra en detalle el panel frontal de Multi-device controller.

1. Interruptor ON/OFF: Enciende el instrumento cuando se encuentra presionado ("I") o apagado en el caso contrario ("O").

2. Display: Muestra el menú de configuraciones al usuario. En la parte superior del display se muestra la frecuencia y la amplitud con la que se encuentra configurado el generador de señales. En la parte inferior se muestra el menú de navegación para diversas configuraciones.

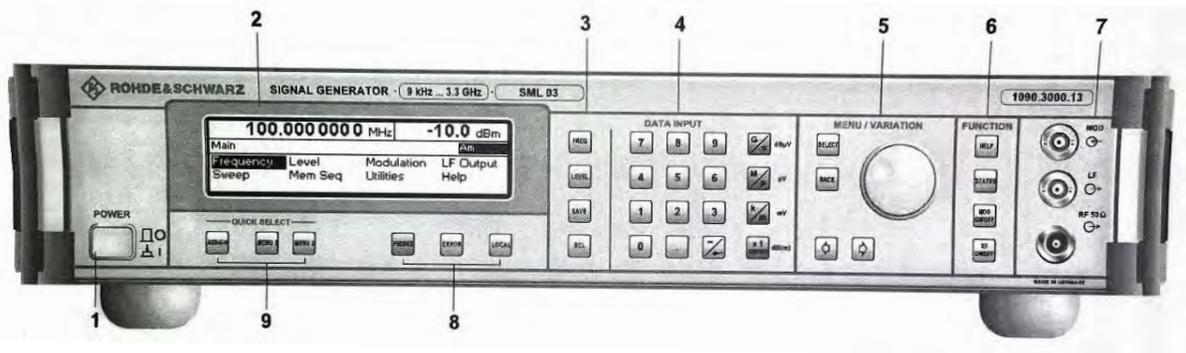


Figura 2.5 Panel frontal Generador de Señales SML03. (Imagen tomada del manual de operación y servicio del *Signal Generator*, Rohde&Schwarz, Munich 2001)

3. Campo de Parámetros: Los parámetros de frecuencia RF y el nivel RF pueden ser introducidos directamente mediante los botones de parámetros, alternativamente desde el menú de operación. Además, las configuraciones del instrumento pueden ser almacenadas y llamadas.

Botón FREQ: Abre la configuración de la frecuencia RF variando los valores desde la perilla de ajustes. Se regresa al menú principal oprimiendo los botones BACK o SELECT

Botón LEVEL: Abre la configuración del nivel RF variando los valores desde la perilla de ajustes. Se regresa al menú principal oprimiendo los botones BACK o SELECT.

Botón SAVE: Abre el almacenamiento de la configuración del instrumentos actual. La selección de memoria se ve afectada al ingresar un número (de 1 a 50) y es terminado por medio de la tecla X1/ENTER.

Botón RCL: Abre la configuración almacenada de un instrumento. La selección de memoria se ve afectada al ingresar un número (de 1 a 50) y es terminado por medio de la tecla X1/ENTER.

4. Sección DATA INPUT: Los valores numéricos, punto decimal, y el signo de menos pueden ser ingresados desde el teclado numérico.



Figura 2.6 Teclado de entrada numérico. (Imagen tomada del manual de operación y servicio del *Signal Generator*, Rohde&Schwarz, Munich 2001)

Botones de Unidades: Los botones de unidad terminan con la entrada de un valor y especifican el factor de multiplicación para la respectiva unidad base.

Las unidades básicas son mostradas junto al campo de entrada mientras los números son ingresados. En caso de los ajustes de nivel, los botones de entrada especifican la unidad.

G/n - dB μ V - Selecciona giga/nano, con nivel RF en dB μ V.

M/u - μ V - Selecciona mega/micro, con nivel μ V

k/m - mV - Selecciona kilo/mili, con nivel mV

x1/Enter - dB(m) - Termina el valor de entrada con la unidad básica y sin unidad. Selecciona con el nivel dBm.

Para cambiar a otras unidades de nivel, simplemente se presiona el botón de la unidad deseada. El parámetro LEVEL tiene que ser activado, por ejemplo, presionado el botón LEVEL.



Figura 2.7 Botones de unidad. (Imagen tomada del manual de operación y servicio del *Signal Generator*, Rohde&Schwarz, Munich 2001)

5. Sección MENU/VARIATION

Botones de menú: Los botones de menú sirven para navegar entre los menús y configuraciones.

SELECT: Reconoce la opción marcada por el menú del cursor ingresando al menú respectivo o a la edición del valor del parámetro seleccionado.

BACK: Regresa el cursor al nivel siguiente más arriba.

← Mueve el cursor una posición a la izquierda. También, si es el caso de la configuración de las opciones, se mueve una posición hacia arriba.

⇒ Mueve el cursor una posición a la derecha. También, si es el caso de la configuración de las opciones, se mueve una posición hacia abajo.



Figura 2.8 Botones de Menú. (Imagen tomada del manual de operación y servicio del *Signal Generator*, Rohde&Schwarz, Munich 2001)

Perilla giratoria: La perilla giratoria mueve el cursor de posición dentro de un nivel de menú o varía el valor de un parámetro. La manera de variación puede ser a de una unidad solamente o de un rango que puede ser definido.

Además, se puede presionar la perilla giratoria cuando el cursor marque una posición en el menú para ingresar a un nivel de menú inferior o desplegar algún ajuste.

6. Sección FUNCTION

Botón HELP: Muestra información auxiliar dependiendo del contexto del menú.

Botón STATUS: Indica el estado del instrumento.

Botón MOD ON/OFF: Enciende y/o apaga la modulación seleccionada en Utilities-ModKey

Botón RF ON/OFF: Enciende o apaga la señal RF.

8.

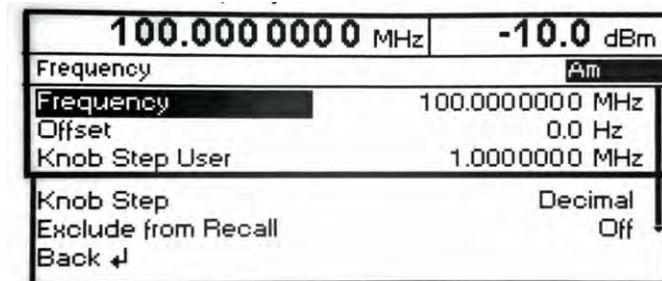
Botón PRESET: Establece un estado definido en el instrumento. Se utiliza el botón SELECT para confirmar.

Botón ERROR: Indica mensajes de error o precaución.

Botón LOCAL: Cambia el instrumento al modo remoto (REMOTO) o local (LOCAL).

2.3.2 Frecuencia RF

La frecuencia RF puede ser ajustada directamente presionando el botón FREQ o por medio del menú de Frecuencia (Frequency). En el menú Frequency la frecuencia de salida de la señal es ingresada e indicada bajo el nombre de la variable Frequency.



100.000 0000 MHz		-10.0 dBm	
Frequency		Am	
Frequency	100.0000000 MHz		
Offset	0.0 Hz		
Knob Step User	1.0000000 MHz		
Knob Step		Decimal	
Exclude from Recall		Off	
Back	←		

Figura 2.9 Menú de Frecuencia (Frequency Menu) . (Imagen tomada del manual de operación y servicio del *Signal Generator*, Rohde&Schwarz, Munich 2001)

2.3.3 Nivel RF

El nivel RF puede ser ajustado directamente presionando el botón LEVEL o por medio del menú de nivel. En los ajustes de Nivel - menú Level, el nivel de salida RF establecido es ingresado e indicado bajo el nombre de variable Amplitude.

dBm, dBuV, mV y uV pueden ser utilizados como unidades de nivel. Los cuatro botones de unidades tienen sobre sí una leyenda con la unidad correspondiente. Para cambiar a otra unidad de nivel, solo es necesario presionar el botón de la correspondiente unidad.

100.000000 MHz		-10.0 dBm
Level/Level		
Amplitude		-10.0 dBm
Offset		0.0 dB
Limit		19.0 dBm
Attenuator Mode		Auto
Atten Fixed Range	0.0 dBm to unlevelled	
Knob Step User		1.0 dB
Knob Step		Decimal
Power Resolution		0.1 dB
Power On State		Previous
Exclude from Recall		Off
Back	←	

Figura 2.10 Menú de Nivel (Level menu) . (Imagen tomada del manual de operación y servicio del *Signal Generator*, Rohde&Schwarz, Munich 2001)

2.3.4 RF On/Off

La señal de salida RF puede ser encendida o apagada con el botón RF ON/OFF. Al cambiar de estado no realiza cambios en el menú actual desplegado en el display. Cuando la señal de salida es apagada, en la línea de cabecera del display aparece "RF Off", cuando es encendida "RF On".

2.3.5 Manejo remoto del SML03

Cuando el instrumento se enciende siempre inicia su operación en modo de manejo local (LOCAL). Sin embargo, el generador de señales puede operar también de manera remota (REMOTE).

El generador de señales puede ser operado remotamente mediante el protocolo GPIB. El instrumento entra en modo remoto cuando mediante el bus le llega una instrucción del controlador.

Durante la operación en modo remoto el panel frontal se deshabilita. El instrumento permanece en modo remoto hasta que se reinicia desde el panel frontal o se le envía la instrucción de regresar al modo Manual. Al cambiar de modo manual a remoto o viceversa no altera la configuración del instrumento.

2.3.5.1 Dirección GPIB del SML03

Para acceder a la dirección del instrumento para su manejo modo remoto hace falta solo ir al sub menú Utilities->System->GPIB-Address. La dirección puede encontrarse en el rango de 1 a 30. La dirección por defecto es 28.

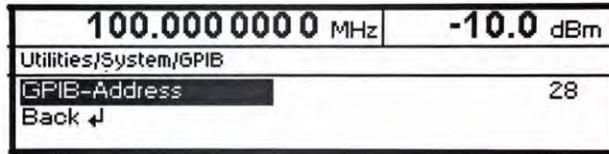


Figura 2.11 Submenú Dirección GPIB. (Imagen tomada del manual de operación y servicio del *Signal Generator*, Rohde&Schwarz, Munich 2001)

También la dirección puede ser cambiada desde el dispositivo controlador de la red de comunicación GPIB mediante el siguiente comando:

```
SYST:COMM:GPIB:ADDR 28
```

2.3.5.2 Instrucciones al Generador de Señales

El Generador de Señales se inscribe al estándar 488.2, como el Multi-Device Controller. Sin embargo, debido a la complejidad del SML03 obedece a los Comandos Estándar para Instrumentos Programables (SCPI). En el Capítulo 1 se ha explicado en qué consisten estos como su empleo. Las instrucciones usadas a continuación pueden consultarse en el Apéndice C. Si se buscara profundizar en la totalidad de las instrucciones del dispositivo, se puede remitir a su manual de uso.

La dirección GPIB del Generador de Señales es 28. A continuación se mostrarán los comandos utilizados en la implementación del Experimento de Ionización de Gases por RF.

En el capítulo siguiente (Capítulo 3) se explicarán con mayor detenimiento las funciones utilizadas para enviar comandos al Generador.

Conectado el Generador y el dispositivo Controlador a la red de comunicación GPIB:

- Inicializamos el descriptor de la del Generador de Señales SML03. Con éste se identificará en adelante al dispositivo y será utilizados para las llamadas posteriores.
 - `Id = ibdev(0, 28, 0, T10s, 1, 0)`
- Limpiamos las funciones del Generador.
 - `lbclr(Id)`
- Cambiamos Frecuencia del Generador a 300 MHz.
 - `li.ibwrt(Id, "FREQ 300 MHz", 12)`
- Cambiamos el Nivel del Generador a 0 dBm.
 - `li.ibwrt(Id, "POW 0 dBm", 9)`
- Iniciamos la emisión de la señal encendiendo la salida RF.
 - `li.ibwrt(Id, "OUT:STAT ON", 11)`

- Una vez que quiera dejarse de radiar se apaga la salida RF.
 - li.ibwrt(ld, "OUT:STAT OFF", 12)

2.4 Antena logarítmica de arreglo de dipolos periódicos Modelo 3148

El Modelo 3148 del fabricante ETS es una antena de banda ancha linealmente polarizada, diseñada para operar sobre el rango de frecuencia de los 200 MHz a los 2 GHz.

Se conecta al amplificador 250W1000AM1 para emitir la onda amplificada. La conexión se realiza mediante un cable coaxial.



Figura 2.12 Antena logarítmica Modelo 3148

2.5 Cable USB/GPIB 2.0 de Alta Velocidad Modelo 82357b

El bus utilizado, fabricado por Keysight Technologies para conectar el dispositivo controlador (una laptop) a la red de comunicación GPIB ya establecida dentro de la Cámara Semianecoica, es el Keysight 82357B USB/GPIB.



Figura 2.13 Bus Keysight 82357B

2.5.1 Características

El bus puede conectarse directamente a un puerto USB de computadora de escritorio o laptop sin la necesidad de instalar a una tarjeta PCI-GPIB sin requerir fuentes de alimentación externa. También es USB 2.0 y la velocidad de transferencias de alta velocidad, superior a los 1.15 MB/s.

2.5.2 Suite de Bibliotecas IO

Es necesario instalar una serie de bibliotecas llamada IO, mediante la cual la computadora reconocer el Bus.



Figura 2.14 Icono que muestra que la Suite de Bibliotecas IO ha sido instalada

Esta incluye, también, un software llamado Keysight Connection Expert, dentro del cual se debe habilitar el envío de instrucciones con programas ajenos a éste, ya que permite enviar mensaje a los dispositivos conectados dentro de la red. Siguiendo la ruta *Setting->Keysight 488 options* se activa el permiso.

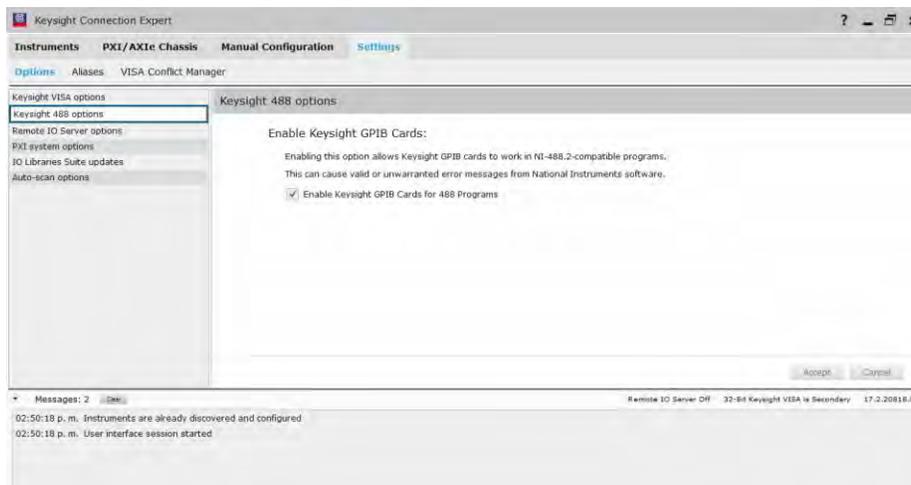


Figura 2.15 Habilitación de permisos para programas externos

2.5.3 LEDs del 82357B

Una vez que el bus 82357B puede ser reconocido por la computadora, se puede conectar. La cabeza del conector GPIB tiene tres LEDs. Inicialmente solo el LED rojo, LED FAIL,

debería estar iluminado. Después de 10 segundos, los 3 LEDs deberían iluminarse mostrando que el 82357B ha sido instalado correctamente y ha sido configurado para su uso con la Bibliotecas IO. Los dos LEDs de color verde, READY y ACCESS, se iluminan cuando la computadora ha reconocido el bus y cuando está listo para comunicarse con la red GPIB, respectivamente. El LED rojo se ilumina cuando ha habido un problema al comunicarse con la computadora o la red.

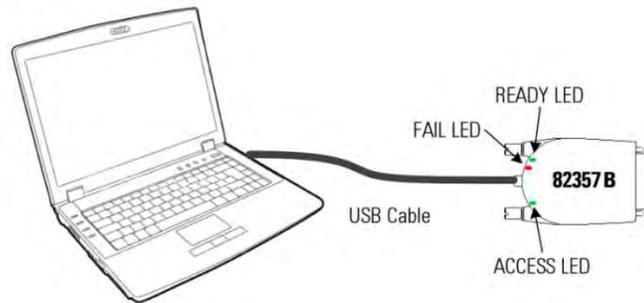


Figura 2.16 LED del bus 82357B

Cabe aclarar que no todas las funciones de los instrumentos son descritas aquí. El resto de las funciones posibles a implementar se encuentran descritas en los respectivos manuales.

2.6 Referencias

- [1] *Manual Multi-Device Controller Model 2090*, EMC Test Systems, 2001
- [2] *Operating and Service Manual Model 250W1000AM1*, Amplifier Research, USA 2002
- [3] *Operating Manual Signal Generator*, Rohde&Schwarz, Munich 2001
- [4] *Keysight 82357B USB/GPIB interface Quick Start Guide*, Keysight Technologies, Malasia 2014
- [5] *Quick Reference Card*, National Instruments, www.ni.com, documento tomando de <http://www.ni.com/pdf/manuals/370497c.pdf>
- [6] *NI-488.2 Function Reference Manual for Windows*, National Instruments, USA 1999
- [7] *Equipment Manual Model 3148 Log-Periodic Dipole Array Antenna*, EMC Test Systems, 1999.

Capítulo 3: Implementación

RFION, nombre dado a la aplicación que este trabajo aborda, fue desarrollado en el lenguaje de programación C#. La compatibilidad del lenguaje con Microsoft Visual Studio, como también la integración y el manejo de Microsoft SQL Server en el mismo panel, vuelven oportuno su uso para el desarrollo de la aplicación. El uso de una IDE es pertinente debido a la extensión del desarrollo de la aplicación por sus herramientas de administración de proyectos y organicidad.

La aplicación establece comunicación con cualquier instrumento GPIB mediante la librería LangInt.dll haciendo referencia a ella con el espacio de nombres National Instruments.NI488.

3.1 Requerimientos

La aplicación descrita a lo largo de este capítulo se desarrolló en base a los requerimientos para el manejo remoto, de uso académico, de los instrumentos de la Cámara Semianecoica involucrados en el Experimento de Ionización de Gases por RF. Esta se opera desde una interfaz gráfica amigable con el usuario, permitiéndole, así, solo realizar las configuraciones de los parámetros necesarios para ejecutar el experimento. Esto se debe a que la complejidad de los instrumentos permite una variada gama de configuraciones que podrían confundir al usuario, el cual suponemos no se encuentra totalmente familiarizado con la singularidad de este tipo de instrumentos.

Sin embargo, la aplicación también ha implementado las medidas necesarias para cumplir con el protocolo de uso de las instalaciones de la Cámara Semianecoica: se debe llevar un registro de uso de las instalaciones. Por lo que la aplicación lleva un historial de uso mediante un sistema de registro de usuarios. El usuario solo puede acceder a la aplicación una vez que este haya sido registrado por el usuario administrador.

Los requerimientos contemplados para la implementación fueron los siguientes:

- Los instrumentos pertenecientes a la Cámara Semianecoica (el Generador de Señales SML03, el Amplificador de Señales 250W1000A, el Multi-Controlador de dispositivos Modelo 2090) deben ser controlados de forma remota mediante el bus de comunicación GPIB.
- Los instrumentos involucrados en el Experimento de Ionización de Gases por RF deben funcionar de forma conjunta. Si uno no se encuentra en condiciones de operar el experimento debe ser imposible de realizar.
- Cada usuario deberá contar con un identificador único (nombre de usuario único) y contraseña para poder hacer uso del programa.
- Solo el usuario administrador podrá registrar usuarios.
- El usuario podrá realizar Experimentos de Ionización de Gases por RF, modificando la amplitud y la frecuencia del generador de señales, la ganancia del amplificador y la duración del experimento.
- Los usuarios con permisos podrán realizar pruebas y configuraciones a los instrumentos de forma individual.
- El usuario administrador podrá consultar un historial de visitas.

- Después de haber realizado una prueba el usuario podrá obtener un reporte en formato PDF con la descripción de las configuraciones últimas del experimento.
- Se deben establecer límites superiores e inferiores para la variación de frecuencia, amplitud y ganancia.
- Los usuarios que no son administradores no pueden consultar su historial de visitas.
- Los usuarios que no son administradores no pueden manejar los instrumentos de manera independiente.

3.2 Instrucciones

Para poder comunicarnos con cualquier dispositivo GPIB conectado a la red se necesita no solo conocer los comandos específicos de éstos, sino también las funciones mediante las cuales uno puede establecer comunicación. Estas funciones principalmente se encargan de administrar la comunicación con los instrumentos: inicializarlos, enviar mensajes, recibir respuestas, terminar la comunicación; son exportados de la librería LangInt.dll.

A continuación, se describen funciones relevantes para la programación de un instrumento GPIB.

3.2.1 IBCLR

Limpia un dispositivo.

```
int ibclr(int di)
```

di – Descriptor del dispositivo a limpiar.
Devuelve el valor de ibsta.

3.2.2 IBDEV

Abre e inicializa el descriptor del dispositivo, configurándolo de acuerdo a los parámetros de entrada.

```
int ibdev (int dt, int di, int dsi, int t, int eot, int eos)
```

dt – Dirección de la Tarjeta. Siempre debe ser 0.
di – Dirección Primaria del dispositivo inicializar.
dsi – Dirección Secundaria del dispositivo a inicializar.
t – Tiempo de espera para respuesta del dispositivo.
eot – Habilita o deshabilita la afirmación de la línea EOI (End-or-Identify) de las operaciones de I/O para el dispositivo. Si es distinto de 0 se afirma cuando el ultimo byte de información se envía. Si es 0 no ocurre nada. Cuando se afirma se dice que ha terminado la transmisión.
eos – Define si habilita el modo EOS (end-of-string), que termina la instrucción automáticamente si la necesidad de indicarlo con una cadena de terminación. (Por ejemplo

<LF> con el amplificador). Si es 1 se activa; 0 desactiva. Esto usualmente se utiliza para dispositivos que obedecen el estándar 488.1
Devuelve el descriptor del dispositivo o -1.

3.2.3 IBLOC

Cambia a modo local el dispositivo.

```
int ibloc (int di)
```

di – Descriptor del dispositivo.
Devuelve el valor de ibsta.

3.2.4 IBONL

Pone el dispositivo en modo en línea o fuera de línea dentro de la red.

```
int ibonl (int di, int v)
```

di – descriptor del dispositivo.
v – indica si el dispositivo entra o sale de línea.
Devuelve el valor de ibsta.

3.2.5 IBRD

Lee información del dispositivo desde un buffer de usuario.

```
int ibrd (int di, void *buf, long c)
```

di- Descriptor del dispositivo.
buf – buffer.
c – tamaño del buffer.
Devuelve el valor de ibsta.

3.2.6 IBWRT

Envía información a un dispositivo a partir de un buffer.

```
int ibwrt (int di, void *buf, long c)
```

di – Descripción del dispositivo.
buf – Información o comando a enviar al dispositivo.
c – Tamaño de bytes a enviar.
Devuelve el valor de ibsta.

3.3 Funciones Globales

Cada que una de las funciones anteriormente descritas es ejecutada actualiza tres funciones globales que reflejan el estado de los dispositivos o la interfaz que se está usando. Esas variables globales son: la palabra de estado (Ibsta), la función de error (Iberr), y la función de conteo (Ibcnt). Estas proveen información útil sobre el estado de la red GPIB. Para la implementación de la aplicación solo se utilizaron las primeras dos funciones. Ibcnt se encarga de contar los bytes escritos y leídos después de cada función de lectura y escritura GPIB por lo que durante el desarrollo no hubo necesidad de utilizarla.

Por otro lado, la aplicación constantemente está revisando el estado de los dispositivos en caso de algún error, por lo que el uso de Ibsta e Iberr, si fue necesario.

3.3.1 Palabra de Estado (ibsta)

Todas las llamadas actualizan la función de estado, ibsta, la cual contiene información acerca del estado de la red GPIB y el hardware. A partir de ibsta se puede consultar si hubo algún error en la ejecución de una instrucción mediante el bit ERR. Existen más bits para verificar el estado, estos se describen con detalle en el Apéndice D.

3.3.2 Función de Error (Iberr)

Si el bit ERR es asignado en ibsta, significa que ha ocurrido un error. Cuando esto sucede, el tipo de error es especificado por Iberr.

3.4 RFION

Para la implementación de la aplicación RFION se decidió dividir de forma lógica su implementación en 4 partes o módulos. Estos módulos están definidos a partir de su funcionalidad dentro del conjunto general del programa. Datos, GPIB, Lógica e Interfaz Gráfica definen la funcionalidad entera del programa. Esta división se realizó con el propósito de facilitar el desarrollo del programa, como también su lectura, mantenimiento y testeo de forma independiente

Módulo Datos: se encarga de establecer la conexión y administrar la conexión con la base de datos. Así como realizar consultas y actualizaciones a ésta.

Módulo GPIB: se encarga de la definición y del manejo de los instrumentos GPIB: El amplificador de potencia 250W1000AM1, el generador de señales SML03, y el controlador de dispositivos *Multi-Device Controller* (Mesa Rotatoria y Torre).

Módulo Lógica: se encarga del manejo de peticiones por parte del módulo Interfaz Gráfica y de respuestas de parte de los módulos de Datos y GPIB. También se encarga de inicializar los instrumentos GPIB como de la generación de reportes del Experimento de Ionización de Gases por RF.

Módulo Interfaz Gráfica: implementa todos los elementos gráficos del programa.

3.5 Base de Datos

Como se mencionaba al inicio del capítulo, la aplicación también debe cumplir con el protocolo de uso de la Cámara Semianecoica. Este señala que debe llevarse un registro de la hora de ocupa como de quienes hacen uso de ella. Este procedimiento se ha venido llevando a cabo mediante el registro del uso de las instalaciones en una bitácora, como también la generación de reportes correspondientes a las pruebas realizadas.

En primer término, se pensó que la aplicación debía de contar con un sistema que pudiera cubrir estas necesidades. Por esta razón se implementó un sistema de registro de usuarios como también un historial de uso. Ambas partes pertenecen a una misma base de datos relacional implementada en Microsoft SQL Server.

La base de datos está compuesta solo de dos tablas: Usuario e Historial. Para los requerimientos de una aplicación de este tipo un modelo relacional de esta extensión es más que suficiente. Si hubiera se hubiera pensado e implementado en un modelo más extenso, este no habría correspondido con el propósito de la aplicación.

A continuación, se muestra el código para la creación de la base de datos, la tabla Usuario y la tabla Historial en el lenguaje SQL.

```
CREATE DATABASE RFIONdb;

CREATE TABLE Usuario{
Nombre_Usuario VARCHAR(40) NOT NULL PRIMARY KEY,
Nombre VARCHAR(30) NOT NULL,
Apellido_Paterno VARCHAR(30) NOT NULL,
Apellido_Materno VARCHAR(30),
Contraseña VARCHAR(40) NOT NULL,
Tipo_Usuario VARCHAR(20) NOT NULL,
Restricciones BOOLEAN NOT NULL,
Imagen_Perfil IMAGE
};

CREATE TABLE Historial{
ID int NOT NULL IDENTITY(1,1) PRIMARY KEY,
Nombre_Usuario VARCHAR (40) NOT NULL,
Fecha DATETIME NOT NULL
};
```

3.5.1 Usuarios

La tabla Usuario de la base de datos define y almacena a los usuarios registrados en la plataforma RFION. Un usuario debe contar con un Nombre de Usuario, Nombre(s) de pila, Apellido Paterno, Apellido Materno, Contraseña, Tipo de Usuario, Restricciones, e Imagen de Perfil.

Varios de estas características del usuario son ingresadas en el momento del registro y no podrán ser cambiadas posteriormente. Solo contraseña e Imagen de perfil son los únicos atributos que un usuario puede editar de su información. Esto se debe a que, por un lado,

el Nombre de usuario corresponde al identificador único de cada usuario. Por el otro, la modificación de Nombre(s) de pila, Apellido Paterno, Apellido Materno podría poner en riesgo la integridad de la información del protocolo de uso de la Cámara Semianecoica. El Tipo de Usuario como las restricciones en la actual versión de RFION no es posible modificarlas, sin embargo, posteriormente podrían editarse en una extensión del programa.

Nombre de Usuario: Es el identificador único (llave primaria) del Usuario. No puede existir dos usuarios con el mismo Nombre de Usuario.

Nombre: Nombre o nombres de pila del usuario.

Apellido Paterno: Apellido paterno del usuario.

Apellido Materno: Apellido materno del usuario, puede estar en blanco este campo.

Tipo de Usuario: Define el tipo de usuario sea Administrador, Académico o Estudiante.

Restricciones: Indica si el usuario tiene restricciones para realizar el Experimento de ionización de Gases por RF: límites establecidos para la operación de los instrumentos. Si no tiene restricciones los límites se atienen a la capacidad física de los instrumentos.

Imagen de Perfil: Fotografía del usuario.

2.5.1.1 Usuario Administrador

El usuario Administrador puede realizar sin restricción de límites el Experimento de Ionización de Gases por RF, agregar y borrar usuarios de la base de datos, puede configurar los límites en los que los instrumentos GPIB operan, como también hacer uso de los instrumentos GPIB de manera independiente. En la configuración actual de la aplicación solo puede existir un administrador de la plataforma, que en este caso es el responsable de la Cámara Semianecoica, ya que el usuario Administrador debe conocer el funcionamiento de los dispositivos porque, aunque se hayan tomado las medidas necesarias para evitar cualquier tipo de error, la configuración de estados incorrectos en los parámetros puede provocar fallas en el sistema e incluso podría dañar los equipos.

2.5.1.2 Usuario Académico

El usuario Académico puede realizar sin restricción de límites el Experimento de Ionización de Gases por RF, sin embargo, esto depende del Administrador a la hora que éste lo registre: este puede aplicarle o no restricciones para el uso de la aplicación. También puede acceder a ajustes de los instrumentos. Este tipo de usuario corresponde para aquellas personas que ya se encuentran familiarizadas con los instrumentos inscritos dentro de las instalaciones de la Cámara Semianecoica.

2.5.1.3 Usuario Estudiante

El usuario Estudiante solo puede llevar a cabo el Experimento de Ionización de Gases por RF con restricciones de límites. Sin embargo, puede establecer los valores de frecuencia, amplitud, ganancia del amplificador y tiempo de duración del experimento, como los otros usuarios. No tiene acceso a la edición de las configuraciones de los instrumentos ya que este tipo de usuario en teoría no conoce el funcionamiento de los instrumentos.

2.5.2 Historial

Una vez que un usuario ha podido iniciar sesión, su visita es registrada. La tabla encargada de almacenar los ingresos de los usuarios dentro de la Base de Datos es Historial.

Cada registro en la tabla Historial está constituido por un Identificador único de visita, el Nombre de Usuario y la Fecha. Ningún registro puede ser modificado, las únicas operaciones que pueden realizarse con esta la tabla Historial es el registro de una nueva visita y el borrado de todo el historial. Esta última operación solo le es posible realizarla al usuario Administrador.

2.6 Inicio de Sesión

Para iniciar la aplicación es necesario cargar el proyecto desde *Microsoft Visual Studio* y ejecutarlo. Al iniciarse la ejecución del programa, se mostrará el *form* o ventana de Inicio de Sesión (Figura 3.1). Ese requiere que el usuario ingrese su nombre de usuario y contraseña para poder acceder a la plataforma.

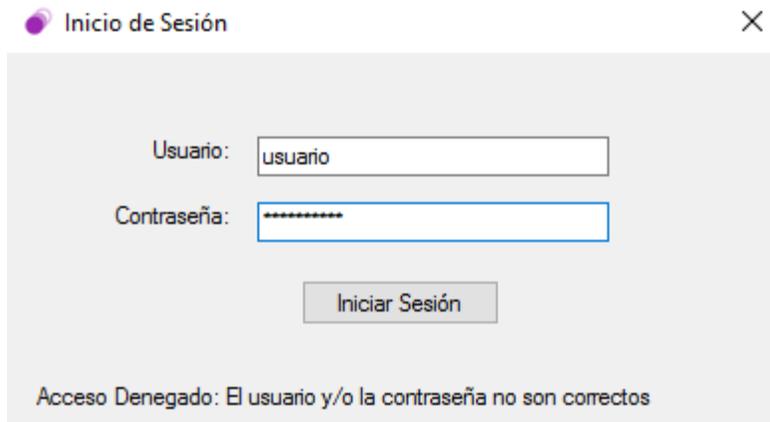


Figura 3.1 Form Inicio de Sesión

Una vez que el usuario ha ingresado los campos y pulsado el botón Iniciar Sesión, se verifica si el nombre de usuario y la contraseña son correctas. Esto se hace realizando una consulta a la tabla Usuario de las Base de Datos, buscando un registro donde coincida el campo Nombre_Usuario con el nombre de usuario ingresado en el *textbox* correspondiente del *form* Iniciar Sesión. Si no se encuentra registro alguno, la base de datos devuelve un objeto nulo, lo que quiere decir que el usuario no existe, por lo que se hace saber al usuario que el acceso le fue denegado. Si existe el registro, la base de datos devuelve una tabla con los datos del registro. Con la tabla se verifica que el campo Contraseña sea igual a lo ingresado en el *textBox* correspondiente de contraseña del *form*. Si son diferentes se hace saber al usuario que el acceso no se le ha sido permitido; en el caso contrario se le pasa un objeto Usuario, con los datos del usuario, al *form* Principal para así poder inicializarse. La programación del inicio de sesión fue la siguiente fue la siguiente:

```
private void buttonIniciarSesion_Click(object sender, EventArgs e)
{
    Usuario usuario = new Usuario();
```

```

        /*Verificamos si existe un usuario que coincida con el nombre de usuario y la
        contraseña ingresadas.
        Si existe devuelve un objeto tipo Usuario, sino devuelve null */
        usuario = usuario.verificarUsuario(textBoxUsuario.Text.Trim(),
        textBoxContrasena.Text);

        if (usuario != null)//Si es diferente a null iniciamos sesión.
        {

            //Damos cuenta al usuario que se le ha permitido el acceso
            labelStatus.Text = "Acceso Permitido";

            //Ocultamos/cerramos el form actual.
            this.Hide();

            //Lanzamos el FormPrincipal
            FormPrincipal f = new FormPrincipal(usuario);
            f.ShowDialog(this);

            //Cuando se haya cerrado la ventana principal volvemos a mostrar ventana de
            Inicio de Sesión
            this.Show();

            //textBoxUsuario.Text = "";
            textBoxContrasena.Text = "";
            labelStatus.Text = "";

        }
        else
        {
            //Damos cuenta al usuario que se le ha negado el acceso
            labelStatus.Text = "Acceso Denegado: El usuario y/o la contraseña no son
            correctos";
        }
    } //buttonIniciarSesion_Click

```

3.6 Form Principal

Una vez que el usuario ha logrado iniciar sesión el *form* de Inicio de Sesión se cerrará y a continuación se desplegará el *form* Principal. En este *form*, como su nombre lo menciona, se pueden acceder a las principales funcionalidades de la aplicación. El panel Perfil, el Panel del Experimento de Ionización de Gases por RF y el Menú Superior constituyen las partes nodales del *form* Principal.

3.6.1 Panel Perfil

Uno de los dos paneles que componen el *form* principal es el panel Perfil. Este está compuesto únicamente por la información general del usuario como su(s) nombre(s) de pila, apellido paterno, apellido materno, nombre de usuario, contraseña e imagen de perfil (Figura 3.2). Cuando el usuario consigue iniciar sesión, se inicializa el *form* Principal pasándole como parámetro al constructor un objeto usuario, que a su vez fue construido a partir de la información obtenida de la base de datos. Es así como se obtiene y presenta la información en el panel Perfil.



Figura 3.2 Panel Perfil del *form* Principal

Como ya se había mencionado, solo dos aspectos de la información del usuario son editables: la contraseña y la imagen de Perfil. Esto obedece al hecho de que el único usuario que puede registrar un nuevo usuario es el Administrador. Si la información básica fuera editable (Nombre(s), Apellidos y Nombre de Usuario) y se hiciera mal uso de los instrumentos bien podría volverse no identificable el usuario responsable.

Para guardar los cambios realizados solo es necesario presiona el botón Guardar Cambios. Este realiza una actualización en la base de datos del Usuario. Las nuevas configuraciones del usuario son tomadas del *CheckBox* de Contraseña y de la ventana emergente invocada por el *LinkLabel* Cambiar Imagen. Antes de que los cambios sean guardados se pregunta al usuario confirmar la operación. Si la actualización fue correcta u ocurrió algún problema, se le hace saber de igual manera al usuario. La rutina fue la siguiente:

```
private void buttonGuardarCambios_Click(object sender, EventArgs e)
{
    DialogResult respuesta = MessageBox.Show("¿Esta seguro de actualizar sus datos?",
    "¿?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

    if (respuesta == DialogResult.Yes)
    {
        //Asignamos los nuevos valores al objeto usuario
        usuario.contrasena = textBoxContrasena.Text;
        usuario.imagen = convertirImageABinary(pictureBoxImagenPerfil.Image);

        //Actualizamos el usuario
        if (usuario.actualizarUsuario(usuario) == 1)
        {
            MessageBox.Show("La actualización fue exitosa", "Actualización Existosa",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        else
        {
            MessageBox.Show("No se puedo actualizar el perfil correctamente",
            "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
//buttonGuardarCambios_Click
```

3.6.2 Panel Experimento

Para poder realizar el Experimento de Ionización de Gases por RF es necesario contar con tres elementos: un generador de señales, un amplificador de potencia y una antena logarítmica. Los instrumentos que se han implementado, en nuestro caso particular, son el Generador de Frecuencias SML03, el Amplificador de Potencia 250W1000AM1 y la Antena Logarítmica Modelo 3148. Estos instrumentos son con los que se cuentan dentro de las instalaciones de la Cámara Semianecoica.

El diseño del panel Experimento muestra una representación gráfica de cada instrumento como de la conexión entre estos (Figura 3.3). Debajo de la figura de cada instrumento se tienen los parámetros para su configuración y manipulación, como también el tiempo de duración del experimento. El panel también integra un botón para actualizar el estado de un instrumento y otro para la generación de un reporte del ultimo experimento ejecutado. El diseño de la interfaz persigue mostrar gráficamente el proceso de la generación de una onda para ser radiada de forma amigable para el usuario.

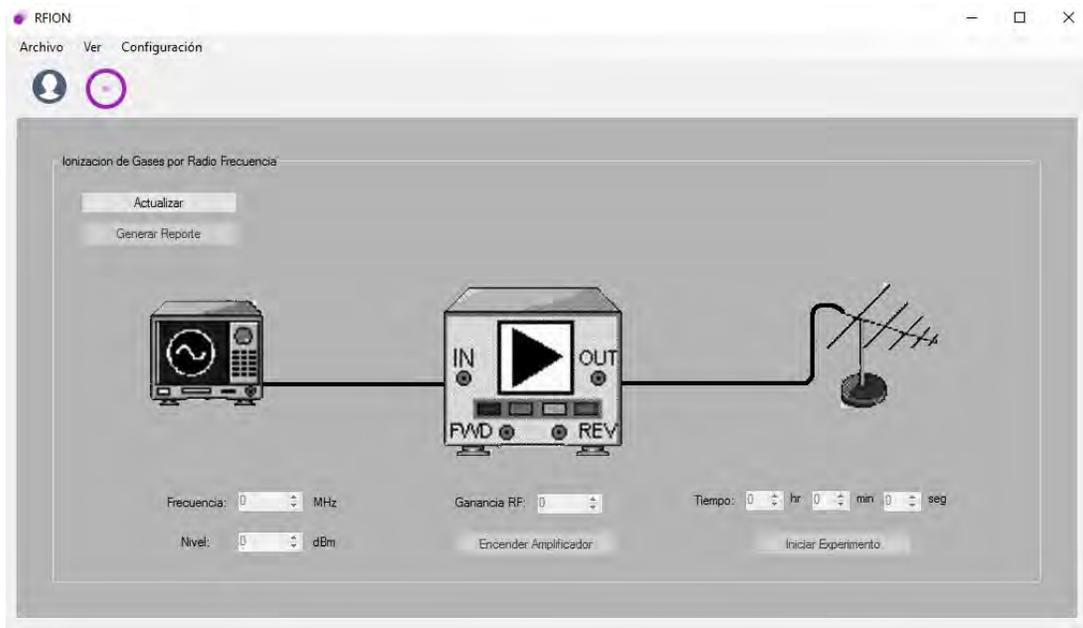


Figura 3.3 Panel Experimento

3.6.2.1 Controles

Las configuraciones para la ejecución del Experimento de Ionización de Gases por RF no corresponden a todas las posibles variables ajustables de cada instrumento. Para la realización del experimento solo es posible editar las siguientes configuraciones:

Parámetro Frecuencia: Configura la frecuencia en la que operara el Generador de Señales.

Parámetro Nivel: Configura el nivel en la que operara el General de Señales.

Parámetro Ganancia RF: Configura la ganancia con la que operara el Amplificador de Potencia.

Parámetros Tiempo: Configura en tiempo que el experimento se estará ejecutando.

Los parámetros del experimento solo se pueden editar cuando todos los instrumentos se encuentran en condiciones de operar: que tanto el Generador de Señales SML03 y el Amplificador de Potencia 250W1000AM se encuentren encendidos y en el caso del Amplificador este se encuentre también con el Interruptor de bloqueo bajo llave (*Keylock Switch*) en modo REMOTE. No se debe olvidar que, aunque esta aplicación maneje de forma remota, estos cuentan con interruptores físicos para habilitarlos y deshabilitarlos como medida de seguridad.

Cuando todos los instrumentos se encuentran en condiciones para operar, para poder habilitar los campos de edición de los parámetros es necesario primero poner el Amplificador en modo encendido presionando el botón Encender/Apagar Amplificador. Por defecto el Amplificador con el *Keylock Switch* en REMOTE se encuentra en modo apagado por seguridad. Una vez que el Amplificador se encuentra en modo encendido los campos de los parámetros del panel Experimento son habilitados para su edición como también el botón Iniciar/Detener Experimento. El experimento puede iniciarse una vez que todos los parámetros son llenados correctamente.

La rutina del experimento consiste en configurar los instrumentos con los parámetros dados y mantenerlos encendidos durante el tiempo establecido. Mientras el experimento se encuentra en ejecución los controles del panel Experimento se deshabilitan con excepción del botón Iniciar/Detener Experimento el cual permite detener el experimento sin que el tiempo se haya agotado. A continuación, se muestra el funcionamiento del botón Iniciar/Detener Experimento.

```
private void buttonIniciarDetenerExperimento_Click(object sender, EventArgs e)
{
    // obtenemos valores de las horas, minutos y segundos que durara el experimento.
    int h = Convert.ToInt32(numericUpDownH.Value);
    int m = Convert.ToInt32(numericUpDownM.Value);
    int s = Convert.ToInt32(numericUpDownS.Value);

    if (h == 0 && m == 0 && s == 0)
        MessageBox.Show("Para iniciar el experimento es necesario definir su
duracion", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    else
    {
        if (experimentoActivo == false)
        {
            //Guardamos el estado del experimento
            guardarUltimoEstadoExperimento();

            // Desabilitamos la edicion de los numericUpDown mientras se ejecuta el
experimento.
            inhabilitarControlesExperimento();

            // Volvemos visible el boton de detener experimento
            buttonIniciarDetenerExperimento.Text = "Detener Experimento";

            // Iniciamos la cuenta regresiva
            timerExperimento.Start();

            //cambiamos la bandera de experimento activo
            experimentoActivo = true;

            // Iniciar experimento
            iniciarExperimento();
        }
    }
}
```



```

gs.cambiarFrecuenciaMHZ(Convert.ToInt32(numericUpDownFrecuencia.Value));
gs.cambiarNivel(Convert.ToInt32(numericUpDownNivel.Value));
gs.encenderRF();

//Amplificador
amp = instrumentos.cargar250W1000AM1();
amp.cambiarGanancia(Convert.ToInt32(numericUpDownGananciaRF.Value));
amp.operate();
} //iniciarExperimento

```

La función *iniciarExperimento()* se encarga de inicializar los instrumentos, el Generador de Señales y Amplificador de Potencia, como también de configurar los parámetros correspondientes de cada instrumento para su operación. Cada que es presionado el botón de Iniciar/Detener Experimento se ejecuta la función *iniciarExperimento()* por lo que se inicializan también, cada vez, los instrumentos. Esto se debe a que durante la ejecución de un experimento a otro el estado de los instrumentos puede cambiar, y alguno dejar de estar disponible.

3.6.2.1.1 Inicialización de instrumentos GPIB

Todos los instrumentos implementados en la aplicación son extendidos de la clase padre InstrumentoGPIB.cs. Esto se debe gracias a la estandarización de los instrumentos GPIB (IEEE 488.2) que permite configurar cualquier instrumento con las mismas instrucciones y parcialmente de la misma forma

Cuando se inicializa un objeto del tipo InstrumentoGPIB es necesario pasarle como parámetro un valor de tipo entero (*int*), que representa la dirección GPIB del instrumento a inicializar. Este parámetro debe encontrarse entre el 0 y el 31. El constructor de la clase Instrumentos es el siguiente:

```

public InstrumentoGPIB(int direccion)
{
    this.PRIM_DIR = direccion;
} //InstrumentoGPIB

```

La creación del objeto InstrumentoGPIB no inicializa como tal el instrumento. La inicialización del instrumento es posible solo a través del método *inicializarInstrumento()*. Esta consiste primeramente en habilitar las funciones y constantes GPIB para su uso, trayéndolas desde la biblioteca LangInt.dll. Después, se da un *reset* a los parámetros y funciones GPIB mediante la línea de gestión IFC, para posteriormente inicializar la red GPIB y el dispositivo controlador. Para finalizar, se inicializa el instrumento mediante la función *ibdev*, ya descrita anteriormente.

Cabe notar que después de cada función ejecutada asociada a las funciones GPIB se revisa el estado de estado *ibsta*, para asegurarnos que la operación se ha realizado correctamente. Cuando las funciones no logran con ejecutarse con éxito la única forma de saber el estado de la operación es a través de las palabras de estado.

```

public void inicializarInstrumento()
{
    try
    {
        /*
        * Inicializa las funciones y constantes GPIB como
        * la direccion de la mesa y el tiempo de espera.
        */
    }
}

```

```

*/
li = new LangInt();
c = new GpibConstants();
TIMEOUT = c.T10s;

li.SendIFC(0); // Limpiamos la interfaz
if ((li.ibsta & c.ERR) != 0) // lanzamos un error
    throw new Exception("No se ha podido inicializar la " + NombreInstrumento);

/*
* ibdev abre e inicializa el descriptor Mesa con el
* que se identificara en adelante al dispositivo y
* se utilizara para llamadas posteriores.
*/
Id = li.ibdev(BDINDEX, PRIM_DIR, SEC_DIR, TIMEOUT, EOTMODE, EOSMODE);
if ((li.ibsta & c.ERR) != 0) // lanzamos un error
    throw new Exception("No se ha podido inicializar la " + NombreInstrumento
);

/*
* Limpia las funciones del dispositivos. Si existe un
* error en el bit ERR se establece es ibsta.
*/
li.ibclr(Id);
if ((li.ibsta & c.ERR) != 0) // throw an error
    throw new Exception("No se ha podido limpiar las funciones del " +
NombreInstrumento );
}
catch (Exception ex)
{
    Console.WriteLine(Environment.NewLine +
        "Error: " + ex.Message + Environment.NewLine + //Nombre del error
        "ibsta = " + li.ibsta + Environment.NewLine + //ibsta
        "iberr = " + li.iberr + Environment.NewLine + //iberr
        NombresError[li.iberr] + Environment.NewLine); //Nombre del error

    deshabilitar();
}
} //Inicializar

```

3.6.2.1.2 Configuración del Generador de Señales

Cuando un instrumento GPIB se ha inicializado correctamente este puede ser configurado y operado. Sin embargo, las funciones que lo permiten no se encuentran contenidas en la clase InstrumentoGPIB.cs sino en las que lo extienden. Ahí es donde se encuentran definidas, específicamente, las funciones para la manipulación del instrumento.

Como se había dicho anteriormente, para la ejecución del Experimento de Ionización de Gases por RF, en el caso del Generado de Señales, es más que suficiente configurar la frecuencia y la amplitud con la que la onda será generada. Estas configuraciones pueden ser dadas a través de los métodos implementados en la clase SML03.cs.

Para la configuración de la frecuencia es necesario ejecutar el método *cambiarFrecuenciaMHZ()*. Este recibe como parámetro un tipo de dato *double* que representa la nueva frecuencia en Megahertz. Sin embargo, no solo es posible configurar la frecuencia en Megahertz sino también en Gigahertz; la funciones *cambiarFrecuenciaGHZ()* se encuentra implementada dentro del código de la clase SML03.cs pero no se ha ocupado para la implementación de la aplicación como tal.

```

public void cambiarFrecuenciaMHZ(double nueva_frecuencia)
{
    try
    {
        String instruccion = "FREQuency " + nueva_frecuencia.ToString() + "MHz";

        li.ibwrt(Id, instruccion, instruccion.Length);
        if ((li.ibsta & c.ERR) != 0)
            throw new Exception("No se enviar la instruccion: " + instruccion);
    }
    catch (Exception ex)
    {
        Console.WriteLine(Environment.NewLine +
            "Error: " + ex.Message + Environment.NewLine + //Nombre del error
            "ibsta = " + li.ibsta + Environment.NewLine + //ibsta
            "iberr = " + li.iberr + Environment.NewLine + //iberr
            NombresError[li.iberr] + Environment.NewLine); //Nombre del error

        deshabilitar();
    }
} //cambiarFrecuencia

```

Para configurar la amplitud o nivel de la onda generada se utiliza la función *cambiarNivel()* que recibe como parámetro un tipo de dato *double*. La amplitud es configurada en dBm (decibel-milivatio).

```

public void cambiarNivel(double nueva_amplitud)
{
    try
    {
        String instruccion = "POWer " + nueva_amplitud.ToString() + "dBm";

        li.ibwrt(Id, instruccion, instruccion.Length);
        if ((li.ibsta & c.ERR) != 0)
            throw new Exception("No se enviar la instruccion: " + instruccion);
    }
    catch (Exception ex)
    {
        Console.WriteLine(Environment.NewLine +
            "Error: " + ex.Message + Environment.NewLine + //Nombre del error
            "ibsta = " + li.ibsta + Environment.NewLine + //ibsta
            "iberr = " + li.iberr + Environment.NewLine + //iberr
            NombresError[li.iberr] + Environment.NewLine); //Nombre del error

        deshabilitar();
    }
} //cambiarModulacion

```

Una vez configurados los parámetros de frecuencia y amplitud, en la ejecución del Experimento de Ionización de Gases por RF, se ejecuta la función *encenderRF()*. Esta se encarga de iniciar el proceso de radiación de la onda. Mientras esta función no se ejecute en el proceso no puede existir emisión alguna.

```

public void encenderRF()
{
    try
    {
        String instruccion = "OUTP:STAT ON";

        li.ibwrt(Id, instruccion, instruccion.Length);
        if ((li.ibsta & c.ERR) != 0)
            throw new Exception("No se enviar la instruccion: " + instruccion);
    }
    catch (Exception ex)
    {

```

```

        Console.WriteLine(Environment.NewLine +
            "Error: " + ex.Message + Environment.NewLine + //Nombre del error
            "ibsta = " + li.ibsta + Environment.NewLine + //ibsta
            "iberr = " + li.iberr + Environment.NewLine + //iberr
            NombresError[li.iberr] + Environment.NewLine); //Nombre del error

        deshabilitar();
    }
} // encenderRF

```

3.6.2.1.3 Configuración del Amplificador de Potencia

En el caso del Amplificador de Potencia la única configuración que debe realizarse es la ganancia. Dentro de la clase correspondiente al Amplificador de Potencia, 250W1000AM1.cs, la función que se encarga de esto es *cambiarGanancia()* que recibe como parámetro un tipo de dato *int*.

```

public void cambiarGanancia(int nueva_ganancia)
{
    try
    {
        string instruccion;
        if (nueva_ganancia < 10)
        {
            instruccion = "G000" + nueva_ganancia.ToString();
        }
        else if (nueva_ganancia < 100)
        {
            instruccion = "G00" + nueva_ganancia.ToString();
        }
        else if (nueva_ganancia < 1000)
        {
            instruccion = "G0" + nueva_ganancia.ToString();
        }
        else
        {
            instruccion = "G" + nueva_ganancia.ToString();
        }

        li.ibwrt(Id, instruccion, instruccion.Length);
        if ((li.ibsta & c.ERR) != 0)
            throw new Exception("No se enviar la instruccion: " + instruccion);
    }
    catch (Exception ex)
    {
        Console.WriteLine(Environment.NewLine +
            "Error: " + ex.Message + Environment.NewLine + //Nombre del error
            "ibsta = " + li.ibsta + Environment.NewLine + //ibsta
            "iberr = " + li.iberr + Environment.NewLine + //iberr
            NombresError[li.iberr] + Environment.NewLine); //Nombre del error

        deshabilitar();
    }
} // cambiarGanancia

```

Una vez que la configuración de la ganancia este dada, de manera similar al Generador de Señales, se debe encender el poner en modo OPERATE el amplificador para que comience la amplificación de la onda que es dada por el generador. Recordemos que si no se cambia el modo del amplificador este se mantendrá en StandBy por lo que la señal jamás será radiada.

```

public void operate()
{
    try

```

```

    {
        string instruccion = "0";
        li.ibwrt(Id, instruccion, instruccion.Length);
        if ((li.ibsta & c.ERR) != 0) // lanzamos un error
            throw new Exception("No se enviar la instruccion: " +
instruccion);
    }
    catch (Exception ex)
    {
        Console.WriteLine(Environment.NewLine +
            "Error: " + ex.Message + Environment.NewLine + //Nombre del
error
            "ibsta = " + li.ibsta + Environment.NewLine + //ibsta
            "iberr = " + li.iberr + Environment.NewLine + //iberr
            NombresError[li.iberr] + Environment.NewLine); //Nombre del
error

        deshabilitar();
    }
} // operate

```

3.6.2.1.4 Actualización y Generación de Reportes

Independientemente de los controles que dan configuración al Experimento de Ionización de Gases por RF, existen dos botones que también se encuentran inscritos al panel Experimento. Estos son: el Botón Actualizar y el Botón Generar Reporte. Ambos botones se encuentran en la parte superior izquierda del panel.

Cuando se inicializa el *form* Principal, por detrás de la aplicación se realiza también el reconocimiento e inicialización de los instrumentos involucrados en el experimento. Si en este proceso los instrumentos se inician correctamente entonces se habilitan los controles del panel. Sin embargo, en el caso contrario, en el que alguno de los instrumentos (Generador de Señales o Amplificador de Potencia) no pueda inicializarse, la aplicación no permitirá la manipulación de ninguno de los controles del panel Experimento, con excepción del botón Actualizar.

El botón Actualizar es el único, dentro del panel Experimento, que a todo momento se encontrara habilitado. La función del botón Actualizar consiste en la verificación del estado de los instrumentos, específicamente cuando estos no se pudieron habilitar en la inicialización de la aplicación. Por lo tanto, si al inicializar la aplicación alguno de los instrumentos no se encuentra en condiciones de ser utilizado, o en medio de la ejecución de la aplicación un instrumento deja de estar disponible para su manejo remoto, y el panel es deshabilitado en consecuencia, el botón podría el panel habilitar sin la necesidad de tener que reiniciar la aplicación, siempre y cuando se hayan solucionado los inconvenientes por los cuales en un inicio no se pudo habilitar.

El botón actualizar está ligado a la función *buttonActualizar_Click_1()* que se encarga a su vez de ejecutar la función *estaDisponibleExperimento()*. Esta función verifica la disponibilidad de los instrumentos involucrados con el Experimento de Ionización de Gases por RF: si se encuentran disponibles se habilitan los controles de panel y se cargan las configuraciones de los instrumentos; en el caso contrario se inhabilitan. Esta función no solo es utilizada por el botón Actualizar, sino también por muchas otras funciones dentro de la aplicación para la constante verificación de la disponibilidad de los instrumentos.

```

public void estaDisponibleExperimento()
{
    //Inicializamos al SML03
    try
    {
        gs = instrumentos.cargarSML03();
        if (gs.estaDisponible() == false)
        {
            inhabilitarControlesExperimento();
            buttonEncenderApagarAmplificador.Enabled = false;
            experimentoDisponible = false;
        }
        else
        {
            habilitarControlesExperimento();
            buttonEncenderApagarAmplificador.Enabled = true;
            experimentoDisponible = true;

            //Establecemos los limites de los controles del generador de señales
            numericUpDownFrecuencia.Minimum = gs.Frecuencia_Minima;
            numericUpDownFrecuencia.Maximum = gs.Frecuencia_Maxima;
            numericUpDownNivel.Minimum = gs.Nivel_Minimo;
            numericUpDownNivel.Maximum = gs.Nivel_Maximo;

            try
            {
                //Imagen blanco y negro - experimento no disponible
                pictureBoxExperimento.Image
Image.FromFile(@"../Resources/BN.png");
            }
            catch (Exception ex)
            {
                MessageBox.Show("Se han alterado archivos fuente del programa. El
funcionamiento puede no ser el adecuado: " + ex.Message, "ERROR", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        //Inicializamos el amplificador 250W1000AM1
        amp = instrumentos.cargar250W1000AM1();
        if (amp.estaDisponible() == false)
        {
            inhabilitarControlesExperimento();
            buttonEncenderApagarAmplificador.Enabled = false;
            experimentoDisponible = false;
        }
        else
        {
            habilitarControlesExperimento();
            buttonEncenderApagarAmplificador.Enabled = true;
            experimentoDisponible = true;

            //Establecemos los limites de los controles de amplificador
            numericUpDownGananciaRF.Minimum = amp.Ganancia_Minima;
            numericUpDownGananciaRF.Maximum = amp.Ganancia_Maxima;

            try
            {
                //Imagen color - experimento disponible
                pictureBoxExperimento.Image
Image.FromFile(@"../Resources/Color.png");
            }
            catch (Exception ex)
            {
                MessageBox.Show("Se han alterado archivos fuente del programa. El
funcionamiento puede no ser el adecuado: " + ex.Message, "ERROR", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
}

```

```

    }
    catch (Exception e)
    {
        inhabilitarControlesExperimento();
        buttonEncenderApagarAmplificador.Enabled = false;
        MessageBox.Show("El archivo de configuración de alguno de los instrumentos
fue modificado: " + e.ToString(),
                        "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

//estaDisponibleExperimento

```

La generación de reportes es otra de las características de la aplicación RFION. Esa función puede ser ejecutada cuando se pulsa el botón Generar Reporte, creando de esta manera un archivo PDF con las especificaciones del ultimo experimento ejecutado en la sesión actual de la aplicación. Cabe destacar que si no se ha ejecutado experimento alguno durante la sesión el botón no se encontrara habilitado.

La información incluida en el reporte, como ya se ha dicho antes, corresponde al último experimento ejecutado de la sesión actual. Una vez se ejecute nuevamente la rutina del experimento la generación de un reporte contendrá la información de este último y ya no será posible generar un reporte con la información del experimento anterior. La información incluida en el reporte corresponde al nombre del operador los instrumentos (el usuario que ha iniciado sesión), el lugar en el que se ha realizado el experimento (por defecto la Facultad de Ciencias de la UNAM), la hora de inicio de la prueba, la duración de la prueba, el estado final de la prueba (si fue interrumpida o termino con éxito) y las configuraciones de los instrumentos involucrados en el experimento (frecuencias, amplitud y ganancia).

```

private void generarReporte()
{
    string nombre_archivo_default = "Reporte_" + usuario.nombre_usuario +
DateTime.Now.ToString("yyyy-MM-dd");

    SaveFileDialog dir = new SaveFileDialog();
    dir.Title = "Guardar reporte en...";
    dir.FileName = nombre_archivo_default;
    dir.Filter = "Pdf Files|*.pdf";
    dir.OverwritePrompt = true;

    DialogResult resultado = dir.ShowDialog();

    if (dir.FileName != "" && resultado == DialogResult.OK)
    {
        FileInfo info = new FileInfo(dir.FileName);

        if (estaEnUso(info))
        {
            GeneradorPDF reporte = new GeneradorPDF(dir.FileName);

            reporte.agregarTitulo("Experimento de ionizacion de gases por RF\n");
            reporte.agregarParrafo(" ");
            reporte.agregarParrafo("Operador: " +
usuario.nombre + " " +
                                usuario.apellido_paterno + " " +
usuario.apellido_materno);
            reporte.agregarParrafo("Lugar del Experimento: FC-UNAM");
            reporte.agregarParrafo("Inicio de la Prueba: " +
DateTime.Now.ToString());
            reporte.agregarParrafo("Duración de la Prueba: " + auxH.ToString()
+ " horas, " +

```

```

                                auxM.ToString() + " minutos," + auxS.ToString()
+ " segundos");
                                reporte.agregarParrafo("Estado final de la Prueba:      " +
estadoFinalExperimento);
                                reporte.agregarParrafo(" ");
                                reporte.agregarSubTitulo("Instrumento:                      SML03");
                                reporte.agregarParrafo("Frecuencia:                          " +
auxFrecuencia.ToString() + " MHz");
                                reporte.agregarParrafo("Nivel:                              " +
auxNivel.ToString() + " dBm");
                                reporte.agregarParrafo(" ");
                                reporte.agregarSubTitulo("Instrumento:                      250W1000A");
                                reporte.agregarParrafo("Ganancia:                          " +
auxGanancia.ToString());

                                reporte.generarPDF();
                                }
                                else
                                {
                                MessageBox.Show("No se puede sobre escribir ya que el archivo esta siendo
ejecutado por otro programa.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                                }
                                }
                                }//generarReporte

```

Para que esta información pueda ser agregada a un archivo PDF fue necesario implementar la librería gratuita *iTextSharp*. Esta es una biblioteca gratuita para la creación de archivos PDF desde .NET. Cuando la función *generarReporte()*, que es la encargada de generar el archivo PDF, es ejecutada cuando el botón Generar Reporte es pulsado, esta abre una ventana (Guardar como) que permite seleccionar el directorio en el que queremos almacenar nuestro reporte. Por defecto la aplicación asigna un nombre al reporte que está compuesto por la palabra *Reporte_* más el nombre de usuario y la fecha.

3.6.3 Form Administración de Usuarios

El *form* de administración de usuarios es una de las secciones de la aplicación a la que solo el usuario Administrador puede acceder ya que esta se encarga de consultar un historial de visitas de la aplicación como de contadores sobre los usuarios y sus visitas; como también de administrar el registro y la eliminación de usuario.

Para acceder al *form* Administración de Usuarios solo hace falta ir a *Menu -> Configuración -> Administración de Usuarios*.

3.6.3.1 Historial

El *form* Administración de Usuario cuenta con dos pestañas, correspondientes a sus funciones. El panel de la primera pestaña, la pestaña Historial (Figura 3.4) se encuentra dividida en dos partes: la lista del historial de visitas, y los contadores de usuarios registrados y visitas realizadas.

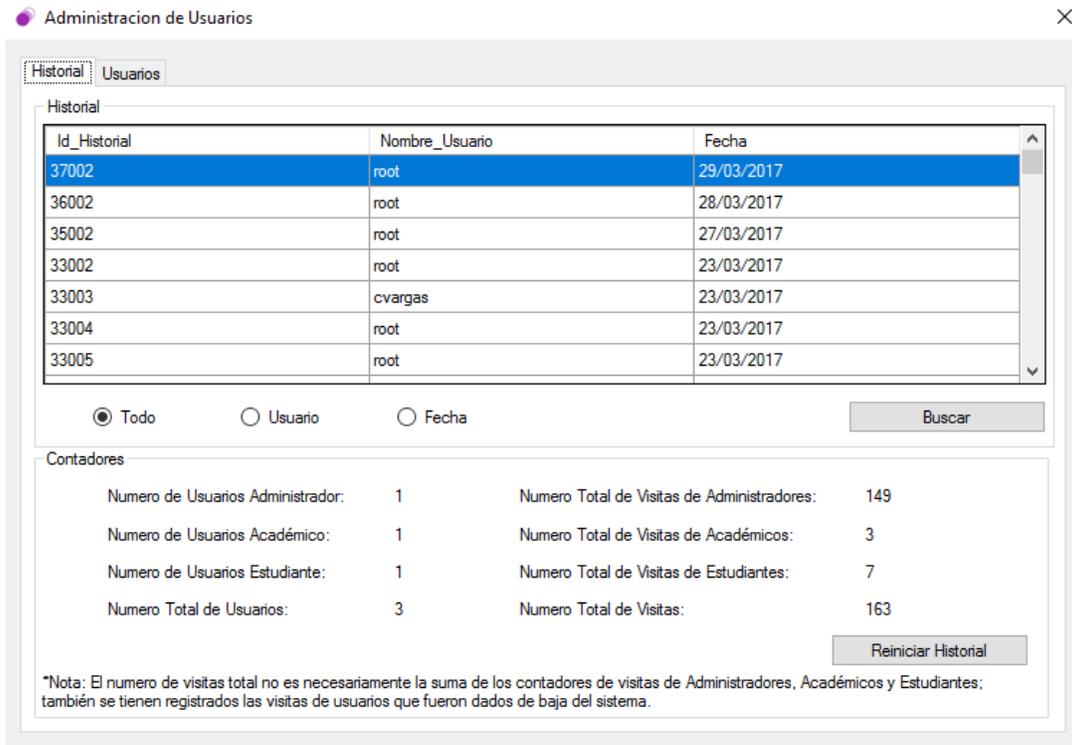


Figura 3.4 Pestaña Historial del form Administración de Usuario

Cuando se inicializa el *form* Administración de Usuarios, por defecto, en la tabla del historial se muestra el historial de las visitas totales a la plataforma en la parte superior del *form*. Sin embargo, también se puede realizar una búsqueda de las visitas totales de un usuario en específico como también de una fecha.

La obtención del historial se realiza mediante consultas SQL a la base de datos (Tabla 3.1) que son luego mostradas en la tabla del historial (*DataGridView*).

	Consulta SQL
Historial Completo	<code>Select * from Historial ORDER BY Fecha DESC;</code>
Historial por Usuario	<code>Select * from Historial WHERE Nombre_Usuario = @nombre_usuario ORDER BY Fecha DESC;</code>
Historial por Fecha	<code>Select * from Historial WHERE Fecha = @fecha ORDER BY Fecha DESC;</code>

Tabla 3.1 Consultas SQL Historial

Por otro lado, en la parte inferior del panel de la pestaña historial se muestra una serie de contadores que se encuentran divididos en dos partes: los contadores de usuario

registrados (columna izquierda) y los contadores de visitas (columna derecha). Ambos contadores muestran la información por tipo de usuario.

Sin embargo, los contadores de visitas pueden ser reiniciados, a diferencia de los contadores de usuarios registrados, ya que estos se encuentran directamente ligados con el historial. En el panel de la pestaña Historial existe un botón llamado Reiniciar Historial, que como su nombre lo indica reinicia los registros de visitas hechas a la plataforma. Cuando es presionado el botón Reiniciar Historial antes de ejecutar la acción se le pide confirmar la operación al usuario Administrador. El reinicio del historial corresponde como tal a la ejecución de la siguiente una instrucción SQL:

TRUNCATE TABLE Historial;

3.6.3.2 Agregar Usuarios

La segunda pestaña del *form* Administrador de Usuarios, la pestaña Usuarios (Figura 3.5) corresponde como tal a la administración de usuarios. Desde el panel de esta pestaña es posible eliminar usuarios como también agregarlos a la base de datos.

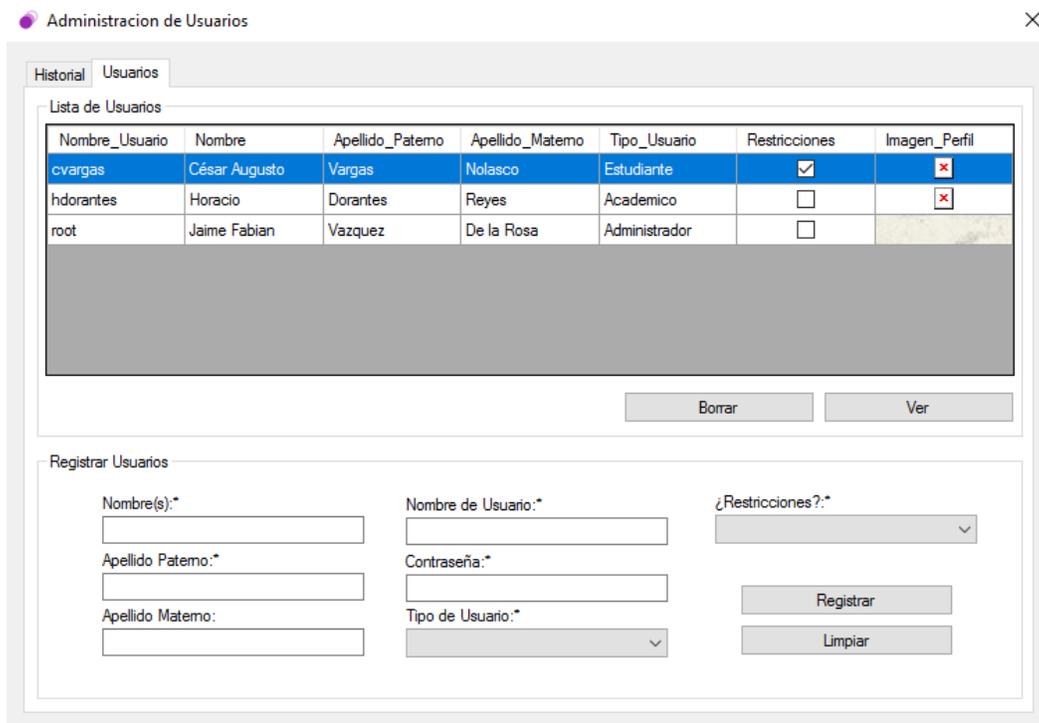


Figura 3.5 Pestaña Usuarios del *form* Administración de Usuario

En la parte superior del panel de la pestaña Usuarios nos encontramos con la lista de todos los usuarios registrados en la plataforma. En esta lista al seleccionar algún usuario es posible realizar dos acciones: ver con a detalle su información o eliminarlo.

Para mostrar su información solo se necesita presionar el botón Ver. Este desplegará un nuevo *form* con la información del usuario y la fotografía del usuario (Figura 3.5).

Figura 3.5 Form Vista de Usuario

Para realizar la eliminación de un usuario solo es necesario seleccionarlo en la lista y presionar el botón Borrar. Antes de completar la operación se le pide al usuario confirmar la operación. Al igual que las operaciones realizadas en historial la eliminación es una actualización sobre la base de datos mediante una instrucción en SQL. La siguiente instrucción realiza la eliminación del registro de un usuario en la tabla Usuarios de la base de datos.

```
DELETE FROM Usuario
WHERE Nombre_Usuario = @nombre_usuario
```

La eliminación se hace a partir del nombre de usuario ya que este parámetro en la definición de la tabla Usuario de la base de datos es la llave primaria, por lo que lo identifica de forma única a cada usuario.

En la segunda parte, la parte inferior del panel de la pestaña Usuario, se tiene un formulario para el registro de un nuevo usuario en la aplicación. Para el registro es necesario ingresar su nombre, su apellido paterno, un nombre de usuario, una contraseña, el tipo de usuario y las restricciones. Aunque también es requerido el apellido materno no es necesario para llevar a cabo el registro de un usuario. La imagen de perfil es agregada por el nuevo usuario posterior a su registro.

Dependiendo del tipo de Usuario que se elija -solo es posible elegir entre Académico o Estudiante- las restricciones pueden o no aplicar. En el momento del registro si se elige que el tipo de usuario es un Estudiante, por defecto el usuario tendrá restricciones de uso con la aplicación. Sin embargo, con el tipo de usuario Académico se tiene la posibilidad de elegir aplicar restricciones o no.

Las restricciones consisten como tal en poder hacer uso de la sección de Lista de dispositivos como de poder ejecutar el experimento de ionización de gases por RF.

El registro de un usuario es como tal una inserción en la tabla Usuario de la base de datos. La operación mediante la cual se realiza el registro de una la información de un usuario es la siguiente:

```
INSERT INTO Usuario(Nombre_Usuario,
Nombre,
Apellido_Paterno,
```

```
Apellido_Materno,  
Contrasena,  
Tipo_Usuario,  
Restricciones)  
VALUES (@nombre_usuario,  
@nombre,  
@apellido_paterno,  
@apellido_materno,  
@contrasena,  
@tipo_usuario,  
@restricciones)
```

3.6.4 Form Lista de Dispositivos

La aplicación RFION no solo ofrece la posibilidad manipular en conjunto los instrumentos involucrados en el Experimento de Ionización de Gases por RF. También es posible la ejecución de algunas operaciones con los instrumentos de forma independiente: el Generador de Señales, el Amplificador de Potencia y el Multi-Device Controller. Este último no fue integrado a la rutina del Experimento. Sin embargo, se integró debido al contexto tan específico en el que se inscribe la aplicación: manejar de forma remota la instrumentación de la Cámara Semianecoica de la Facultad de Ciencias de la UNAM. El Multi-Device Controller juega un papel nodal dentro de la Cámara ya que es el instrumento encargado de controlar el sistema de posicionamiento. Específicamente la Torre con Antena y la Mesa Rotatoria. Como se comentó con anterioridad: cualquier aplicación que intente integrarse al entorno de un lugar como la Cámara Semianecoica, debe controlar el sistema de posicionamiento necesariamente.

El *form* Lista de Dispositivos es un menú que muestra los dispositivos implementados en la aplicación para realizar ajustes (Figura 3.6). A través de este se puede acceder a las configuraciones de cada uno. Sin embargo, no es posible para todos los usuarios acceder a esta sección de la aplicación; es solo para aquellos que no tienen restricciones. Para acceder a este *form* es necesario seguir la ruta *Menú -> Configuración -> Lista de Dispositivos*.

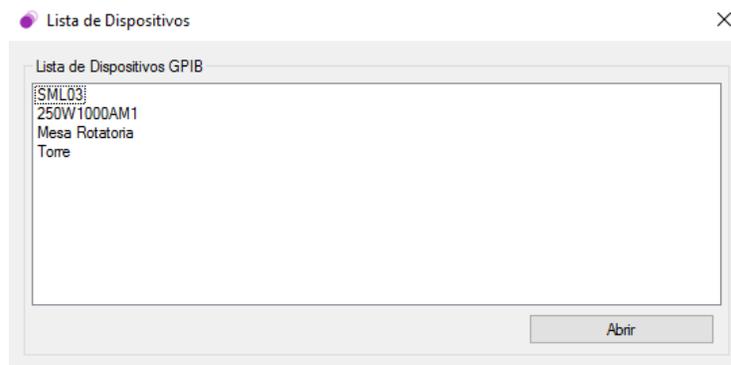


Figura 3.6 Form Lista de Dispositivos

3.6.5 Form Ajustes

A través de lista de dispositivos es posible acceder a las configuraciones de los instrumentos de la aplicación. Cada dispositivo cuenta con un *form* Ajustes que está compuesto, en el caso del Generador de Señales y el Amplificador de Potencia de 2

pestañas, y en el de la Torre y la Mesa Rotatoria de 3. Estas describen las configuraciones de los instrumentos como también, solo en el caso de la Torre y la Mesa, controles para su manejo remoto de forma independiente.

3.6.5.1 General

Todos los instrumentos en su *form* de Ajustes correspondiente cuentan con la pestaña General (Figura 3.7). Esta contiene la información básica del instrumento: el estado de conexión en el que se encuentra, su disponibilidad para operar; el nombre del instrumento; y la dirección GPIB del instrumento mediante el cual se identifica dentro de la red de comunicación. La información contenida en esta pestaña no es posible editarse.

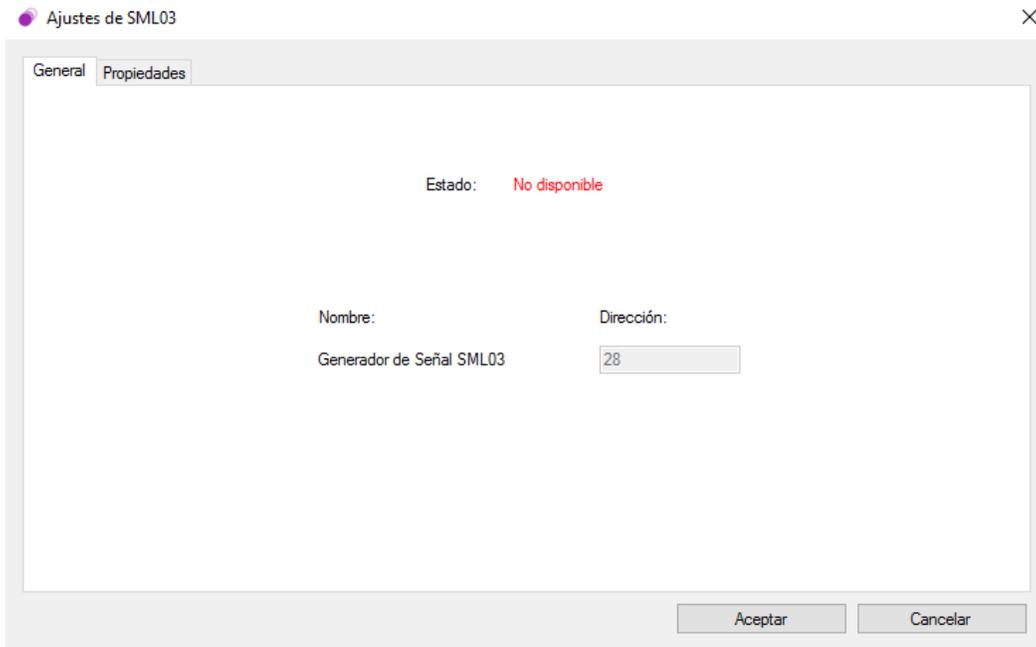


Figura 3.7 Pestaña General del *form* Ajustes de SML03

3.6.5.2 Propiedades

La pestaña Propiedades (Figura 3.8), también disponible para todos los instrumentos, muestra las configuraciones mediante cuales operan los instrumentos: los límites de operación de la frecuencias y amplitud del generador de señales, el límite de la ganancia del amplificador de potencia, los límites de altura y velocidad de la Torre, y los limites en grados y velocidad en que gira la mesa rotatoria.

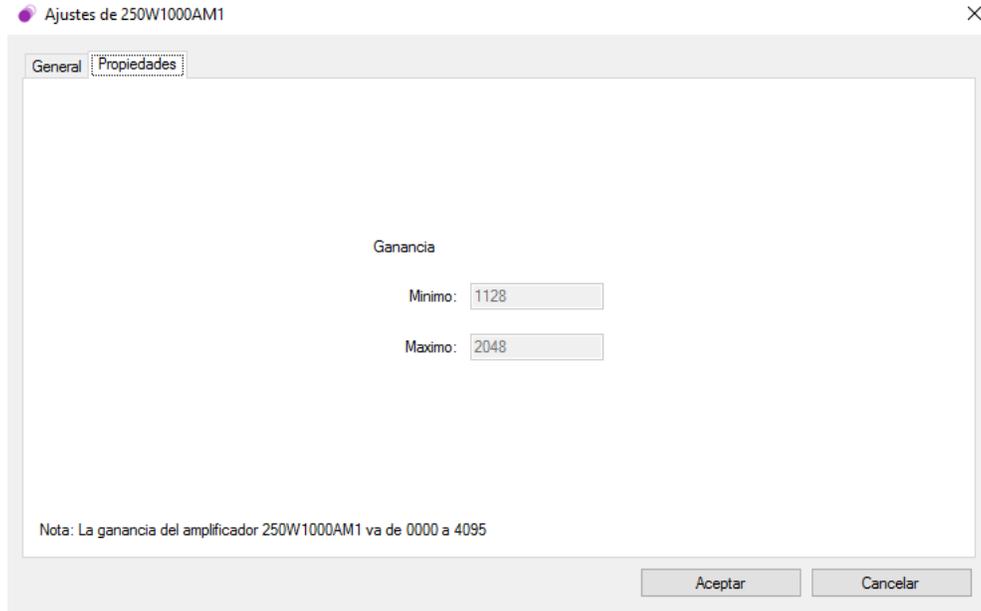


Figura 3.8 Pestaña Propiedades del *form* Ajustes de 250W1000AM1

3.6.5.2 Test

Solo para un par de instrumentos se tiene disponible la pestaña Test: La Torre y la Mesa Rotatoria. Esto se debe a que a diferencia del generador de señales y el amplificador de potencia el funcionamiento de la Torre y la Mesa Rotatoria puede darse de forma totalmente independiente. En el caso de los otros dos instrumentos, si alguno no se encuentra operando, por la configuración de conexión en la que se encuentran, no es posible que radien señal alguna; no se puede emitir como tal una salida.

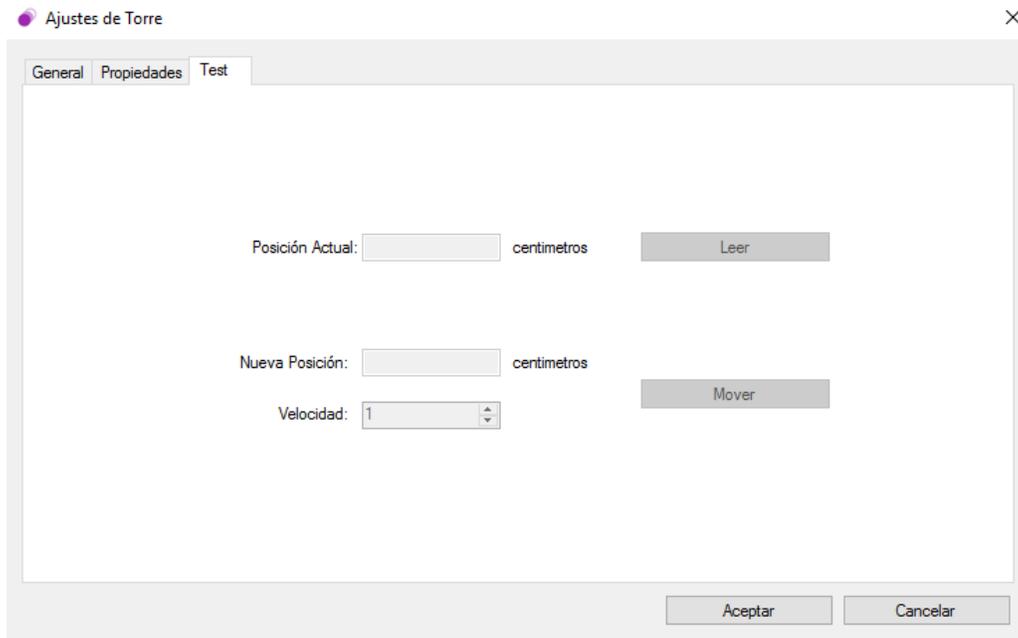


Figura 3.8 Pestaña Test del *form* Ajustes de Torre

La pestaña Test para el caso de la Torre (Figura 3.8) permite cambiar la altura de la antena como la velocidad en que lo hace. Se ingresa la altura a la que se quiere llevar la torre, en centímetros, se define la velocidad en que se moverá (existen 8 velocidades diferente), y finalmente se presiona el botón Mover. El botón Mover manda llamar 2 funciones de forma consecutivas: la función *cambiarVelocidad()* que recibe como parámetro la velocidad en la que se moverá la Torre; y la función *cambiarPosición()* que recibe la nueva posición de la antena.

La función *cambiarVelocidad()*, como su nombre lo indica, modifica la velocidad de la Torre, entre las 8 velocidades posibles. Como en cada instrucción que se envía a un instrumento GPIB posteriormente se verifica el estado de este para saber si el envío de la instrucción no genero un error.

```
public void cambiarVelocidad(int velocidad)
{
    string instruccion = "S" + velocidad;

    try
    {
        li.ibwrt(Id, instruccion, instruccion.Length);
        if ((li.ibsta & c.ERR) != 0)
            throw new Exception("No se pudo enviar la instruccion: " + instruccion);
    }
    catch (Exception ex)
    {
        Console.WriteLine(Environment.NewLine +
            "Error: " + ex.Message + Environment.NewLine + //Nombre del error
            "ibsta = " + li.ibsta + Environment.NewLine + //ibsta
            "iberr = " + li.iberr + Environment.NewLine + //iberr
            NombresError[li.iberr] + Environment.NewLine); //Nombre del error
    }
}

//cambiarVelocidad
```

La función *cambiarPosición()* mueve la antena de la Torre a la nueva posición establecida. Una vez que se da la instrucción al instrumento se inicia una iteración en la cual constantemente se está preguntando al mismo cuál es su posición actual como si este sigue moviéndose. Hasta que el instrumento responda que se ha detenido se sale de este ciclo y concluye de ejecutarse la función.

```
public void cambiarPosicion(double nueva_posicion)
{
    try
    {
        string instruccion = "SK " + nueva_posicion.ToString();

        li.ibwrt(Id, instruccion, instruccion.Length);
        if ((li.ibsta & c.ERR) != 0)
            throw new Exception("No se enviar la instruccion: " + instruccion);

        DateTime localDate;
        bool TorreOpTerminada = false;

        do
        {
            li.ibwrt(Id, "CP?", 3);
            if ((li.ibsta & c.ERR) != 0)
                throw new Exception("No se enviar la instruccion: CP?");

            li.ibrd(Id, out ValStr, TAMARRAY);
        }
    }
}
```

```

        if ((li.ibsta & c.ERR) != 0)
            throw new Exception("No se puedo leer la respuesta de la mesa
retoratoria");

        Console.WriteLine("La posicion actual de la Torre es: " + ValStr);

        localDate = DateTime.Now;
        int i = localDate.Millisecond + (localDate.Second * 1000);

100)        while ((DateTime.Now.Millisecond + (DateTime.Now.Second * 1000)) < i +

        {
            Application.DoEvents();
        }

        if (!TorreOpTerminada)
        {
            li.ibwrt(Id, "*OPC?", 5);
            if ((li.ibsta & c.ERR) != 0)
                throw new Exception("No se enviar la instruccion: *OPC?");

            li.ibrd(Id, out ValStr, TAMARRAY);
            if ((li.ibsta & c.ERR) != 0)
                throw new Exception("No se puedo leer la respuesta de la mesa
retoratoria");

            // Comprobamos si se ha detenido la mesa
            if ((String.CompareOrdinal(ValStr, "0")) == 0)
                TorreOpTerminada = false;
            else
                TorreOpTerminada = true;

        }

        Application.DoEvents();
    } while (!TorreOpTerminada);
}
catch (Exception ex)
{
    Console.WriteLine(Environment.NewLine +
        "Error: " + ex.Message + Environment.NewLine + //Nombre del error
        "ibsta = " + li.ibsta + Environment.NewLine + //ibsta
        "iberr = " + li.iberr + Environment.NewLine + //iberr
        NombresError[li.iberr] + Environment.NewLine); //Nombre del error
}

} //cambiarPosicion

```

Otra de las funciones que contiene la pestaña Test es poder saber la posición actual del instrumento. Solo es necesario presionar el botón Leer para ejecutar la función *posicionActual()* que se encarga de preguntar al instrumento su posición actual. La siguiente función lo describe:

```

public double posicionActual()
{
    try
    {
        //dar posicion actual
        li.ibwrt(Id, "CP?", 3);
        if ((li.ibsta & c.ERR) != 0)
            throw new Exception("No se pudo enviar la instruccion: CP?");

        li.ibrd(Id, out ValStr, TAMARRAY);
        if ((li.ibsta & c.ERR) != 0)
            throw new Exception("No se pudo leer la respuesta de la torre");

        //Devolvemos la posicion actual
    }
}

```

```

        return Convert.ToDouble(ValStr);
    }
    catch (Exception ex)
    {
        Console.WriteLine(Environment.NewLine +
            "Error: " + ex.Message + Environment.NewLine + //Nombre del error
            "ibsta = " + li.ibsta + Environment.NewLine + //ibsta
            "iberr = " + li.iberr + Environment.NewLine + //iberr
            NombresError[li.iberr] + Environment.NewLine); //Nombre del error

        return -1; //error
    }
}

```

La configuración de la pestaña Test es la misma para el caso de la Mesa Rotatoria con la diferencia que el cambio de posición de esta se da en grados y no en centímetros. Las funciones *cambiarVelocidad()*, *cambiarPosicion()* y *posicionActual()* también existen en la programación de la Mesa Rotatoria con diferencias mínimas.

3.7 Carga y Guardado de Configuraciones

Las configuraciones anteriormente descritas pueden ser almacenadas presionando el botón Aceptar del *form*. Sin embargo, es necesario que los instrumentos se encuentren activos, en el caso contrario la aplicación no permitirá realizarlos.

Cuando se inicializa la aplicación estas configuraciones son cargadas. Estas son almacenadas en archivos de texto plano. Uno por cada instrumento integrado a la aplicación. Los archivos de texto contienen el nombre del instrumento, su dirección GPIB y la definición de los parámetros entre los cuales pueden operar. La aplicación toma esta información y configura los instrumentos como también la interfaz del Experimento de Ionización de Gases por RF acotando el rango de los controles de configuración de los instrumentos.

La decisión de que la información fuera almacenada en archivos de texto se debe a que también pueden ser editados directamente desde el archivo. Una incorrecta configuración de estos, como establecer parámetros de forma incorrecta daría como resultado la imposibilidad de operar los instrumentos desde la aplicación. También, en el caso de que los archivos fueran eliminados estos podrían ser fácilmente sustituidos ya que la legibilidad de los archivos es fácil para el usuario.

3.8 Referencias

- [1] *Microsoft SQL Server 2012*, Microsoft Corporation, USA, 2013
- [2] *La biblia de C#*, Jeff Ferguson, Brian Patterson, Jason Beres, Pierre Boutquin, Meeta Gupta, España 2003
- [3] *NI-488.2 Function Reference Manual for Windows*, National Instruments, USA 1999
- [4] *Manual Multi-Device Controller Model 2090*, EMC Test Systems, 2001
- [5] *Operating and Service Manual Model 250W1000AM1*, Amplifier Research, USA 2002
- [6] *Operating Manual Signal Generator*, Rohde&Schwarz, Munich 2001
- [7] *Keysight 82357B USB/GPIB interface Quick Start Guide*, Keysight Technologies, Malasia 2014

Capítulo 4: Observaciones y Trabajo Futuro

4.1 Observaciones del Amplificador Modelo 250W1000AM1

Durante la implementación de las instrucciones para el manejo del Amplificador Modelo 250W1000AM1 se presentaron problemas al establecer comunicación. Para poder programar las rutinas de cada uno de los instrumentos es necesario acudir a revisar los manuales. Sin embargo, en el caso del Amplificador, el manual contenía errores, como también, se puede decir que faltaba contenido.

4.1.1 Cadena de terminación <LF>

Como ya fue descrito en el capítulo anterior, cuando se envía una instrucción al amplificador, el manual indica que a toda instrucción, se le debe agregar al final la cadena <LF> para indicar que finaliza la instrucción. Y aunque esto no es necesario cuando se ha habilitado el modo EOS, resulta preferible implementar ambos métodos en dado caso que la inicialización del instrumento no se contemple la habilitación el modo EOS.

El problema surgió al implementar las instrucciones del amplificador Modelo 250W1000AM1 ya que sólo se ejecutaron algunas de las instrucciones. Las instrucciones fueron escritas tal y como el manual lo indica.

Dentro de la Cámaras Semianecoica existe otro amplificador, el Modelo 500A100AM1, del mismo fabricante (Rohde & Schwarz), cuyas características son similares, así como también las instrucciones de manejo remoto descritas en el manual respectivo. El script, con las mismas instrucciones que se utilizaron para manipular el Modelo 250W1000AM1, fue ejecutado para el Modelo 500A100AM1. La ejecución de todas las instrucciones fue satisfactoria.

Posteriormente, para poder lograr la comunicación completa con el amplificador se tuvo que realizar una revisión de los archivos de ejecución del programa EMC32. Dependiendo del amplificador conectado el programa EMC32 elige las instrucciones correspondientes de un archivo de texto plano, dependiendo del fabricante del amplificador: existen varios archivos de texto con las mismas instrucciones, pero diferente sintaxis, para cada amplificador compatible con el software. Ya que el archivo con las instrucciones del amplificador Modelo 250W1000AM1 no funcionaron en su totalidad se probaron alternativamente con las instrucciones de otros modelos. Por este medio se logró concluir que el error radicaba en la cadena <LF> que se agregaba al final, y que el manual explícitamente marcaba. Esta no debía ser agregada en solo algunas instrucciones. Esto podría suponer que el firmware del amplificador tenga errores.

4.1.2 Estado actual

Por otro lado, el amplificador no cuenta con una instrucción para saber en qué modalidad se encuentra (Operate, StandBy), o si está encendido o apagado. Las únicas instrucciones de petición corresponden al estado de las variables de su funcionamiento. Tampoco es posible conocer algunas de estas características desde los comandos comunes ya que está inscrito al estándar IEEE 488.1.

Sin embargo, el manejo actual de la red GPIB de la Cámara Semianecoica se realiza mediante el software de propietario EMC32 de Rohde & Schwarz, con el cual sí es posible conocer el estado actual del amplificador. Se puede suponer, por lo tanto, que el manual del amplificador 250W1000A10 no cuenta con todas las especificaciones respecto al manejo vía remota del dispositivo, aunque se ha intentado con las posibles instrucciones que podría compartir con otros amplificadores bajo el estándar IEEE 488.1 no ha sido posible conocer su estado.

4.2 Trabajo Futuro

La creación de RFION no solo ha satisfecho los requerimientos del proyecto al que estaba inscrito. Sino que en la base del mismo se obedece a otros proyectos que se han venido contemplando para la extensión del uso de la Cámara Semianecoica por estudiantes y académicos.

4.2.1 Ampliación de RFION

Primeramente, los planes de extensión que se ha contemplado para RFION buscarían la implementación de todas las funcionalidades de los instrumentos ya agregados especialmente las del Generador de Señales SML03. Esto permitiría poder ampliar las posibilidades para otro tipo de experimentos que académicos y alumnos que estudian los fenómenos electromagnéticos podrían utilizar, como el estudio de transmisión de información inalámbrica, el estudio de blindajes electromagnéticos y la deposición de energía de RF y microondas sobre la materia.

También se tiene contemplado la implementación de un nuevo instrumento a la plataforma: el EMI Test Receiver ESIB26 que es un dispositivo de medición, es decir, un dispositivo receptor y no emisor como el generador de señales. Su implementación permitiría extender la funcionalidad de RFION para poder agregar una rutina para la caracterización de antenas y pruebas completas de Compatibilidad Electromagnética.

4.2.1.1 Reportes en línea

Como se había discutido anteriormente, es importante llevar un registro del uso que se le da a la Cámara Semianecoica, por protocolo. Actualmente el almacenamiento de los reportes se han venido llevando a cabo dentro de la computadora que administra la red GPIB y tiene instalado el programa EMC32.

La creación de una base de datos para el almacenamiento de reportes y consulta en línea es un proyecto que ya se tenía contemplado. En conjunto con esto, una página web dedicada a la Cámara Semianecoica de la Facultad de Ciencias se encargaría de proveer información sobre su uso comercial, académico y de investigación, como también de la consulta de los reportes.

A la extensión de RFION podría agregársele perfectamente este proyecto, permitiendo desde la misma la actualización de la base de datos con los nuevos reportes que se generen.

4.2.2 Software de testeo propio

Con la implementación de RFION, específicamente con el estudio del protocolo GPIB y la lectura de los manuales del Generador de Señales SML03, el Amplificador 250W1000AM1 y el Multi-Device Controller Modelo 2090, se han dado las bases para el desarrollo de una plataforma de código abierto para investigación y testeo. La aplicación EMC32 pertenece a Rohde & Schwarz, por lo que es imposible manipular los instrumentos con total libertad.

El desarrollo de una aplicación propia permitiría principalmente lo siguiente:

- Se podría manipular cualquier instrumento conectado a la red GPIB, siempre y cuando se programen las rutinas e instrucciones correspondientes y sean respetadas las restricciones de conexión. Las actuales aplicaciones EMC32 y ESK1 (también fabricado por Rohde & Schwarz, que actualmente ya no recibe soporte por parte de la compañía) solo permiten realizar ensayos con los instrumentos que traen precargados, sin la posibilidad poder agregar o reconfigurar alguno.
- Se podrían realizar ensayos fuera de los parámetros establecidos por los estándares de testeo. Las actuales aplicaciones, no permiten realizar ensayos fuera de dichos parámetros (establecidos por FCC, IEC y la IEEE), por lo que limita la funcionalidad de los instrumentos en su modo remoto y no sirve con fines de investigación.
- Se podrían configurar diferentes pruebas y ensayos con configuraciones específicas de instrumentos y parámetros.

4.3 Referencias

[1] *Operating and Service Manual Model 250W1*

000AM1, Amplifier Research, USA 2002

[2] *Operating and Service Manual Model 500A100AM1*, Amplifier Research, USA 2002

[3] *Microsoft SQL Server 2012*, Microsoft Corporation, USA, 2013

[4] *Operating Manual EMI TEST RECEIVER*, Rohde&Schwarz Munich 2001

Capítulo 5: Conclusiones

A lo largo de los capítulos anteriores se presentó una investigación del bus estándar GPIB, la descripción del Generador de Señales SML03, el Amplificador de potencia 250W1000AM1, y el Multi-Device Controller Modelo 2090, para desarrollar una interfaz capaz de realizar Experimentos de Ionización de Gases por RF, de forma remota, dentro de la Cámara Semianecoica de la Facultad de Ciencias de la UNAM.

El manejo remoto de los dispositivos de la Cámara Semianecoica al inicio del desarrollo de este trabajo se realizaba a través del bus GPIB. Esa fue la razón por la que se decidió elegir este bus y su protocolo de transmisión como medio de comunicación para los instrumentos, frente al bus RS-232, al que también responden los instrumentos.

Debido al enfoque específico del Experimento de Ionización de Gases por RF solo fue necesario implementar algunas de las funciones de los instrumentos. Sin embargo, el estudio de los instrumentos a partir de los manuales, mostraron que para la realización de la totalidad de la implementación de las funcionalidades de cada uno de los instrumentos es necesario un amplio conocimiento del protocolo de transmisión GPIB, específicamente de los estándares IEEE 488.1 y IEEE 488.2. Aunque, como ya se ha discutido anteriormente, el estándar IEEE 488.2 es solo una extensión de las funcionalidades del IEEE 488.1, es necesario contemplar ambas ya que para el estándar IEEE 488.1 depende totalmente de las instrucciones que son señaladas en los manuales.

Cada instrumento cuenta con su correspondiente manual de uso. Sin embargo, no en todos se encontró la información necesaria para su correcta manipulación ya que suponen conocimientos previos sobre el protocolo GPIB. En el caso de Generador de Señales SML03 que implementa los SCPI, se tuvo que acudir a otras fuentes de información ajenas a su manual, para el manejo de los Comandos Comunes.

El desarrollo de una plataforma propia y de código abierto, como RFION, permite no solo implementar los instrumentos inscritos a la Cámara Semianecoica. A partir de la misma librería con la que se ha logrado la comunicación con los instrumentos involucrados en el Experimento de Ionización de Gases por RF, se ha podido establecer comunicación, con scripts independientes, con instrumentos como los osciloscopios Tektronik. Así, es posible una variedad de opciones para la implementación de ensayos o rutinas que involucren una diversidad de instrumentos que no necesariamente pertenezcan al mismo fabricante.

Una extensión del programa, o tomada como base para la creación de una plataforma que se adecúe a todas las necesidades requeridas dentro de las actividades de la Cámara Semianecoica, para pruebas e investigación, se presenta viable y necesaria después de este trabajo.

Apéndices

Apéndice A: Comandos del Multi-Device Controller

Se describen a continuación una serie de comandos para el manejo del Multi-Device Controller. Solo se incluyeron los más relevantes para este trabajo. Si se desea consultar la totalidad de estos debe remitirse al manual.

Comando	Descripción
CC	Ordena a la Mesa Rotatoria moverse en dirección contra el sentido del reloj. EL movimiento está limitado por el límite contra sentido del reloj. Sintaxis: CC Ejemplo: CC
CL	Cambia el límite de la Mesa Rotatoria en contra del sentido del reloj. El valor tiene que ser menor al límite en sentido del reloj y entre -999.9 y 999.9. Sintaxis: CL [+/-]XXX[.X] El signo + es opcional. El signo – es necesario para valores negativos. Los valores decimales deber ir en el siguiente formato XXX[.X] Ejemplo: CL -100.2
CL?	Consulta el límite en contra del sentido del reloj de la Mesa Rotatoria. Debe leerse después la respuesta. Sintaxis: CL? Ejemplo: CL?
CP?	Consulta la posición actual de la Mesa Rotatoria o la Torre. Debe leerse después la respuesta. Sintaxis: CP? Ejemplo: CP?
CW	Ordena a la Mesa Rotatoria moverse en dirección del sentido del reloj. El movimiento es limitado por el límite en sentido del reloj. Sintaxis: CW Ejemplo: CW
CY	Cambia el contador de ciclos de la Torre o Mesa. El valor debe estar entre 0.0 y 999.5. El valor 0 representa un contador infinito de escaneo. Sintaxis CY XXX.X Los valores decimales deber ir en el siguiente formato XXX[.X] Ejemplo: CY 2.5
DN	Ordena a la torre moverse hacia abajo. El movimiento es limitado por el limite interior. Sintaxis: DN Ejemplo: DN

LL	<p>Cambia el límite inferior de la Torre. El valor tiene que ser menor al límite superior y entre -999.9 y 999.9.</p> <p>Sintaxis: LL [+/-]XXX[.X] El signo + es opcional. El signo – es necesario para valores negativos. Los valores decimales deber ir en el siguiente formato XXX[.X] Ejemplo: LL 200.5</p>
LL?	<p>Consulta el limite inferior de la Torre. Debe leerse después la respuesta.</p> <p>Sintaxis: LL? Ejemplo: LL?</p>
PH	<p>Ordena a la Torre cambiar su polarización a horizontal.</p> <p>Sintaxis: PH Ejemplo: PH</p>
PV	<p>Ordena a la Torre cambiar su polarización a vertical</p> <p>Sintaxis: PV Ejemplo: PV</p>
P?	<p>Consulta a la torre como esta polarizado. Debe leerse después la respuesta. Una vez leída la respuesta, si esta es “1” la torre esta polarizada horizontalmente, si es “0” esta polarizada verticalmente.</p> <p>Sintaxis: P? Ejemplo: P?</p>
S#	<p>Cambia la velocidad de la Mesa o Torre.</p> <p>Sintaxis S# El símbolo # representa el numero de velocidad seleccionada. Ejemplo: S1</p>
S?	<p>Consulta la velocidad de la Mesa o Torre.</p> <p>Sintaxis: S? Ejemplo: S?</p>
SC	<p>Ordena al dispositivo iniciar un escaneo entre los límites predefinidos. El número de Escaneos es determinado por el valor de los ciclos (CY).</p> <p>Sintaxis: SC Ejemplo: SC</p>
SC?	<p>Pregunta al 2090 si el modo escaneo esta activo. Debe leerse después la respuesta.</p> <p>Sintaxis: SC? Ejemplo: SC?</p>
SK	<p>Ordena al dispositivo iniciar la búsqueda de del valor especificado (Torre o Mesa Rotatoria). El valor debe encontrarse entre los límite inferior/en contra del sentido del reloj y superior/en el sentido del reloj actual definido.</p> <p>Sintaxis: SK [+/-]XXX[.X]</p>

	<p>El signo + es opcional. El signo – es necesario para valores negativos. Los valores decimales deber ir en el siguiente formato XXX[.X] Ejemplo: SK 100</p>
ST	<p>Detiene el movimiento de la Torre o Mesa.</p> <p>Sintaxis: ST Ejemplo: ST</p>
UL	<p>Cambia el límite superior de la Torre. El valor tiene que ser mayor al límite inferior y entre -999.9 y 999.9.</p> <p>Sintaxis: UL [+/-]XXX[.X] El signo + es opcional. El signo – es necesario para valores negativos. Los valores decimales deber ir en el siguiente formato XXX[.X] Ejemplo: UL 400</p>
UL?	<p>Consulta el límite superior de la Torre. Debe leerse después la respuesta.</p> <p>Sintaxis: UL? Ejemplo UL?</p>
UP	<p>Ordena a la Torre moverse en hacia arriba. El movimiento está limitado por el límite superior.</p> <p>Sintaxis: UP Ejemplo: UP</p>
WL	<p>Cambia el límite en sentido del reloj de la Mesa. El valor definido debe ser mayor que el límite en contra del sentido del reloj y entre -999.9 y 999.9.</p> <p>Sintaxis: WL [+/-]XXX[.X] El signo + es opcional. El signo – es necesario para valores negativos. Los valores decimales deber ir en el siguiente formato XXX[.X] Ejemplo: WL 359.9</p>

Apéndice B: Comandos del Amplificador

Se describen a continuación una serie de comandos para el manejo del Amplificador 250W1000AM1. Solo se incluyeron los más relevantes para este trabajo. Si se desea consultar la totalidad de estos debe remitirse al manual.

Comando	Descripción
P	Enciendo o apaga el Modelo 250W1000AM1. Sintaxis: Px Cuando el valor de x es 0, el amplificador se apagar; cuando el valor de x es 1 se enciende. Ejemplo: P1<LF>
S	Pone el Modelo 250W1000AM1 en modo Standby. Sintaxis: S Ejemplo: S<LF>
O	Pone el Modelo 250W1000AM1 en modo Operate. Sintaxis: O Ejemplo: O<LF>
R	Limpia los errores y reinicia el Modelo 250W1000AM1. Sintaxis: R Ejemplo: R<LF>
G	Establece el nivel de ganancia remota del Modelo 250W1000AM1 con 4095 pasos de resolución. Sintaxis: Gxxxx Los valores de xxxx van de 0000 (mínimo de ganancia) a 4095(máximo de ganancia) que son los pasos de resolución de la ganancia. Ejemplo: G0000<LF>
G?	Consulta el ajuste de la ganancia RF actual. Sintaxis: G? Ejemplo: G? <LF>
I?	Consulta la identidad del Model 250W1000AM1 Sintaxis: I? Ejemplo: I?<LF>

Apéndice C: Comandos del Generador de Señales

Se describen a continuación una serie de comandos para el manejo del Generador de Señales SML03. Estos corresponden a los SCPI correspondiente del tipo de instrumento .Solo se incluyeron los más relevantes para este trabajo. Si se desea consultar la totalidad o a detalle de los comandos posibles debe remitirse al manual.

Comando	Descripción
FREQuency	<p>Cambia la frecuencia del SML03.</p> <p>Sintaxis: FREQuency val ó FREQ val val es el valor de la frecuencias. El valor numérico debe ser seguido por un espacio en blanco y a continuación la unidad de medida MHz ó GHz. Ejemplo: FREQ: 300 MHz</p>
OUTPut:STATus	<p>Enciende o apaga la emisión de radio frecuencia.</p> <p>Sintaxis: OUTPut:STATus ON/OFF ó OUTP:STAT ON/OFF ON en para iniciar la emisión de RF; OFF para detenerla. Ejemplo: OUTP:STAT ON</p>
POWer	<p>Cambia el nivel del SML03.</p> <p>Sintaxis: POWer val ó POW val val es el valor del nivel . El valor numérico debe ser seguido por un espacio en blanco y a continuación la unidad de medida dBm. Ejemplo: POW 0 dBm</p>
SYSTEM:PRESET	<p>Establece en instrumento en un estado predefinido.</p> <p>Sintaxis: SYSTEM:PRESET Ejemplo: SYSTEM:PRESET</p>