



UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

**“DIRECTRICES ORIENTADAS AL CONTROL DE INSTRUMENTOS
MEDIANTE SCPI, EJEMPLIFICADAS EN UNA HERRAMIENTA PARA EL
MANEJO DEL ANALIZADOR DE ESPECTROS ESA-L E4403B”**

TESIS

**PARA OBTENER EL TÍTULO DE
INGENIERO EN ELECTRÓNICA**

PRESENTA

FRANCISCO AGUILAR ACEVEDO

DIRECTOR DE TESIS

M.C. JOSÉ ANTONIO MORENO ESPINOSA

HUAJUAPAN DE LEÓN, OAX.; ABRIL DE 2006

**Tesis presentada el 5 de abril de 2006
ante los siguientes sinodales:**

**Ing. Hugo Suárez Onofre
M.C. Arturo Pablo Sandoval García
M.C. Jacob Javier Vásquez Sanjuan**

Director de Tesis:

M.C. José Antonio Moreno Espinosa

Dedicatoria

A la persona más maravillosa del mundo. Mi madre.

Paco

Agradecimientos

A mi abuela Consuelo por su gran cariño. A mis hermanas Yuridia y Gladys, por todo cuando hemos compartido. A toda mi familia; mi tía Beatriz, mis primos, mi cuñado y mis sobrinos.

Al Prof. José Antonio Moreno Espinosa por toda su confianza, que espero no defraudar.

A los sinodales Prof. Hugo Suárez Onofre, Prof. Arturo Pablo Sandoval García y Prof. Jacob Javier Vásquez Sanjuán, por el tiempo dedicado a la revisión del presente trabajo.

A todas y cada una de las personas que han contribuido en mi formación primero como persona y después como profesional.

Y sobre todo a Dios, por darme la oportunidad.

Paco

Índice

Índice	ix
Lista de tablas	xiii
Lista de figuras	xv
Resumen	xvii
Abstract	xix
1. Introducción	1
1.1. Planteamiento y objetivos de la tesis	2
1.2. Estructura de la tesis	2
2. Marco conceptual	5
2.1. Control de instrumentos	6
2.1.1. Tipos de instrumentos	6
2.1.1.1. Instrumentos de sobremesa	6
2.1.1.1.1. GPIB	7
2.1.1.1.2. Serial	8
2.1.1.1.3. Ethernet	8
2.1.1.1.4. USB	9
2.1.1.2. Instrumentos modulares	9
2.1.1.2.1. VXI	10
2.1.1.2.2. PCI/CompactPCI	10
2.1.1.2.3. PXI	11
2.1.2. Ambiente de desarrollo de aplicaciones	12
2.1.2.1. C/C++	13
2.1.2.2. Basic	14
2.1.2.3. LabVIEW	14
2.1.2.4. LabWindows/CVI	16
2.1.2.5. VEE	16
2.2. Analizador de espectros	18
2.2.1. Mediciones con analizadores de espectros	18
2.2.2. Tipos de analizadores de espectros	18
2.2.3. Fundamentos sobre analizadores de espectros superheterodinos	19
3. SCPI	21
3.1. Objetivo de SCPI	21
3.1.1. Intercambiabilidad de instrumentos	22

3.1.2. Ciclo de vida de SCPI.....	22
3.2. Construcción de ordenes.....	23
3.2.1. Cabeceras de programa.....	24
3.2.1.1. Cabeceras de ordenes comunes.....	24
3.2.1.2. Cabeceras de control de instrumento.....	24
3.2.1.2.1. Generación de mnemotécnicos.....	25
3.2.1.2.2. Generación de la estructura jerárquica.....	25
3.2.1.2.3. Preguntas.....	26
3.2.2. Parámetros.....	26
3.2.2.1. Carácter.....	26
3.2.2.2. Numérico.....	26
3.2.2.3. Boleano.....	26
3.2.2.4. Unidades de medida y sufijos.....	26
3.2.2.5. Expresiones.....	27
3.3. Instrumentos.....	27
3.3.1. Modelo de un instrumento.....	27
3.3.1.1. Encaminamiento de señal.....	29
3.3.1.2. Medida.....	29
3.3.1.3. Generación de señal.....	30
3.3.1.4. Memoria.....	30
3.3.1.5. Formato.....	31
3.3.1.6. Disparo.....	31
3.3.2. Clases de instrumentos.....	31
3.4. Agrupación de ordenes.....	33
3.4.1. Subsistemas.....	33
3.4.2. Instrucciones de medición.....	35
3.5. Registros de estado.....	36
3.5.1. Cola de errores y eventos.....	38
3.5.2. Byte de estado.....	38
3.5.3. Registro de eventos estándar.....	39
3.5.4. Registro de operación.....	40
3.5.5. Registro de cuestionamiento.....	40
4. Directrices para el control de instrumentos SCPI.....	43
4.1. Selección de ordenes.....	44
4.1.1. Administración de subsistemas.....	46
4.1.2. Administración de instrucciones de medición.....	46
4.2. Manejo de datos.....	48
4.2.1. Selección del formato de datos.....	49
4.2.2. Uso de bloques de datos de longitud definida.....	50
4.3. Acceso a memoria.....	51

4.3.1. Memoria interna	51
4.3.2. Memoria de almacenamiento masivo	52
4.4. Control del sistema de registros de estado.....	54
4.4.1. Método de sondeo.....	55
4.4.2. Método de solicitud de servicio.....	56
4.4.2.1. Generación de una solicitud de servicio	56
5. Herramienta de manejo del AS ESA-L 4403B.....	59
5.1. Especificación del software	60
5.1.1. Estudio de factibilidad.....	60
5.1.1.1. Descripción de alternativas.....	60
5.1.1.1.1. Agilent BenchLink para Analizadores de Espectros	60
5.1.1.1.2. Agilent Web Remote Control para AE de las series PSA, ESA-E/L y EMC.....	60
5.1.1.1.3. Agilent IntuiLink para AE de las series PSA, ESA-E/L y EMC.....	61
5.1.1.2. Factibilidad	61
5.1.1.2.1. Factibilidad operativa	61
5.1.1.2.2. Factibilidad técnica.....	62
5.1.2. Especificación de requerimientos	62
5.1.2.1. Requerimientos de usuario	63
5.1.2.2. Requerimientos del sistema.....	63
5.2. Diseño.....	64
5.2.1. Diseño de la arquitectura del software	64
5.2.2. Diseño de la interfaz	67
5.2.3. Diseño procedimental.....	69
5.3. Implementación	71
5.4. Validación.....	75
6. Conclusiones.....	81
Bibliografía.....	83
Acrónimos	87
Anexo A. Conectividad con instrumentos	A-1
A.1. Desarrollo de aplicaciones usando VISA	A-2

Lista de tablas

Tabla 2.1. Objetos para el control de instrumentos en LabVIEW.....	15
Tabla 2.2. Objetos para el control de instrumentos en VEE.	16
Tabla 2.3. Bloques de un analizador de espectro superheterodino.....	20
Tabla 3.1. Ordenes comunes IEEE 488.2.....	23
Tabla 3.2. Expresiones para ordenes SCPI.....	27
Tabla 3.3. Clases de instrumentos SCPI.....	31
Tabla 3.4. Subsistemas de ordenes SCPI.....	33
Tabla 3.5. Sintaxis de instrucciones de medición.....	36
Tabla 3.6. Bits del byte de estado.....	39
Tabla 3.7. Bits del registro de estado de eventos estándar.	39
Tabla 3.8. Bits del registro de operación.....	40
Tabla 3.9. Bits del registro de cuestionamiento datos/señal.....	40
Tabla 4.1. Directrices para el control de instrumentos.....	44
Tabla 4.2. Tareas comunes requeridas en aplicaciones de control de instrumentos.	45
Tabla 4.3. Empleo de instrucción de medición.	47
Tabla 4.4. Ordenes de formato SCPI.....	49
Tabla 5.1. Análisis de tareas para el diseño de la GUI para la HeMAE ESA-L E4403B.....	68
Tabla 5.2. Especificaciones del diseño grafico de la GUI para la HeMAE ESA-L E4403B.....	68
Tabla A.1. Consideraciones para el uso de software del I/O VISA.	A-3

Lista de figuras

Figura 1.1. Estructura del trabajo de tesis.	3
Figura 2.1. Componentes de un sistema de medición y automatización.....	5
Figura 2.2. Esquema básico para control de instrumentos de sobremesa.....	6
Figura 2.3. Esquema general para control de sistemas VXI, CompactPCI y PXI.	9
Figura 2.4. Diagrama a bloques de un analizador de espectros superheterodino clásico.....	19
Figura 3.1. Esquema para la construcción de cabeceras de ordenes comunes.	24
Figura 3.2. Esquema para la construcción de cabeceras de control de instrumento.....	24
Figura 3.3. Ejemplo de la estructura jerárquica SCPI.	25
Figura 3.4. Modelo básico de un instrumento programable.	28
Figura 3.5. Ejemplos de modelos SCPI simplificados.	28
Figura 3.6. Expansión del bloque de medida.....	29
Figura 3.7. Expansión del bloque de generación de señal.....	30
Figura 3.8. Grupo de instrucciones de medición.	35
Figura 3.9. Estructura mínima para el sistema de registros de estado en instrumentos SCPI.....	37
Figura 3.10. Jerarquía expandida del registro de estado de cuestionamiento.....	41
Figura 4.1. Uso de subsistemas e instrucciones de medición en una aplicación bajo Agilent VEE. .	48
Figura 4.2. Estructura para bloques de datos de longitud definida.....	50
Figura 4.3. Obtención de trazas de datos de un instrumento mediante una rutina en LabVIEW.....	52
Figura 4.4. Sistema de registros de estados del analizador de espectros ESA-L E4403B.	55
Figura 4.5. Uso de un sistema de registros de estado en una aplicación bajo Agilent VEE.	57
Figura 5.1. Estrategia de desarrollo incremental.	59
Figura 5.2. DFD de nivel contextual para la HeMAE ESA-L E4403B.....	65
Figura 5.3. DFD de nivel 1 para la HeMAE ESA-L E4403B.	65
Figura 5.4. DFD de nivel 2 que refina el proceso gestionar modificación de parámetros.	65
Figura 5.5. DFD de nivel 2 que refina el proceso gestionar obtención de datos.....	66
Figura 5.6. DFD de nivel 2 que refina el proceso monitorear condición del instrumento.	66
Figura 5.7. DFD de nivel 2 que refina el proceso almacenar datos en archivo(s).....	66
Figura 5.8. Primera iteración de la estructura del software HeMAE ESA-L E4403B.	67
Figura 5.9. Trazado para la GUI de la HeMAE ESA-L E4403B.	69
Figura 5.10. Prototipo para la GUI de la HeMAE ESA-L E4403B.	69
Figura 5.11. Diagrama de transición de estados para la HeMAE ESA-L E4403B.	70
Figura 5.12. Diagrama de flujo para el modelo operacional de la HeMAE ESA-L E4403B.....	70
Figura 5.13. Modulo de programa para la actualización de valores de parámetros del instrumento. .	71

Figura 5.14. Modulo de programa para la obtención de parámetros del instrumento.	73
Figura 5.15. Modulo de programa para la obtención de trazas datos y mediciones del instrumento.	73
Figura 5.16. Modulo de programa para la obtención de una imagen de la pantalla del instrumento.	74
Figura 5.17. Modulo de programa para el monitoreo del estado del instrumento.	75
Figura 5.18. Modulo de programa para la lectura de registros de estados.	75
Figura 5.19. Operación habitual de la HeMAE ESA-L E4403B.	76
Figura 5.20. Operación de la HeMAE ESA-L E4403B para la obtención de mediciones de potencia de canal.	77
Figura 5.21. Operación de la HeMAE ESA-L E4403B para la obtención de imágenes de la pantalla del instrumento.	78
Figura 5.22. Operación de la HeMAE ESA-L E4403B bajo un estado particular del instrumento. ...	78
Figura 5.23. Documento sobre una hoja de cálculo generado por la HeMAE ESA-L E4403B.	79
Figura A.1. Esquema general de componentes involucrados en la conectividad con instrumentos. A-2	

Resumen

El conjunto de ordenes normalizadas para instrumentos programables SCPI define uno de los modelos conceptuales más utilizados en el diseño de instrumentos programables. La meta principal de esta norma, es la reducción de tiempo en el desarrollo de aplicaciones para equipos de prueba automatizados, proporcionando un entorno consistente de programación para el control de instrumentos y manejo de sus datos.

Este trabajo de tesis presenta el desarrollo de un proceso de investigación aplicada motivado en la obtención de un conjunto de directrices encaminadas al control de instrumentos mediante SCPI que busquen facilitar el desarrollo de aplicaciones de esta índole, independientes del protocolo de comunicaciones y la plataforma de desarrollo empleados.

Con el fin de situar el ambiente en que la investigación se desarrolla, se presenta un marco conceptual referente el control de instrumentos en sistemas automatizados de medida. Asimismo y como resultado ilustrativo del trabajo de tesis, se exhibe la implementación de una herramienta software para el manejo del Analizador de Espectros ESA-L E4403B, que de manera esperada hace alusión al uso de las directrices planteadas.

Abstract

Standard Commands for Programmable Instruments (SCPI) define one of the conceptual models more utilized in the design of programming instrument. The goal of SCPI is to reduce Automatic Test Equipment (ATE) program development time, providing a consistent programming environment for instrument control and data usage.

This thesis presents the development of an applied investigation process motivated in the getting of a group of rules for instrument control by means of SCPI that provides the support from development of independent applications of the communication protocol and of the development environment used.

With the intention to rank the environment investigation, this work presents a conceptual framework about the instrument control in measurement and automation system. Likewise, and as part of the result of this thesis, it shows the implementation of a software tool for the management the Spectrum Analyzer ESA-L E4403B, that of manner expected to do usage of the rules establish.

1. Introducción

El uso de instrumentos de medida en la toma de decisiones sobre el control de procesos de fabricación, gestión, etc., fue el impulso principal para el desarrollo de los llamados equipos de prueba automatizados (ATEs, *Automated Test Equipments*). Concebidos inicialmente como una agrupación entre computadora, sistema de comunicación e instrumento, los sistemas ATE evolucionaron rápidamente para dar surgimiento a conceptos tan complejos como la instrumentación electrónica programable e instrumentación virtual.

El empleo de gran variedad de interfaces y protocolos de comunicación entre dispositivos no estandarizados en los años 60 y principios de los 70, tuvo su fin cuando en 1975, el IEEE (*Institute of Electrical and Electronic Engineers*) aprobó el estándar IEEE 488-1975 o bus GPIB (*General Purpose Bus Interface*). El IEEE 488 definió una interfaz estándar eléctrica y mecánica para conectores y cables, así como el control sobre la transferencia de datos, direccionamiento y protocolos generales para la transmisión individual de bytes de datos.

La limitación de esta norma al aspecto físico de la conexión del bus motivó que en 1987 esta se ampliara, mediante una segunda versión llamada IEEE 488.2. Esta extensión a la norma facilitaría la programación de los instrumentos mediante la normalización de protocolos de intercambio de mensajes, formatos de datos y sintaxis, informes de estado y ordenes generales comunes a distintos tipos de instrumentos, sin embargo, esta no consentía que dos instrumentos similares bajo este estándar, manejaran un mismo conjunto de ordenes, tarea que fue consumada hasta 1990, con el surgimiento de la norma SCPI (*Standard Commands for Programmable Instruments*) y la definición de un modelo conceptual único para el diseño de instrumentos programables [34].

Al mismo tiempo que se llevaron a cabo adelantos en el desarrollo de sistemas ATE compatibles, se realizaron esfuerzos para acrecentar la capacidad de procesamiento de los instrumentos GPIB. Así, en 1987 un consorcio de compañías fabricantes de instrumentos electrónicos presentó la arquitectura VXI (*VME eXtension for Instrumentation*) basada en el bus industrial VME (*Versa Module Europe*), con instrumentos modulares en tarjetas. La instrumentación programable basada en los buses GPIB y VXI permitió la creación de sistemas ATE sobre diversas plataformas para computadoras y múltiples sistemas operativos dando lugar al concepto de instrumentación virtual. El impulso al avance en las prestaciones del bus GPIB y nuevas arquitecturas de instrumentos programables, ha sido desde entonces una problemática de gran interés, un ejemplo claro son la presentación de la especificación HS488 en 1993, destinada a conseguir una velocidad máxima de 8MBps para GPIB, o bien la arquitectura PXI (*PCI eXtensions for Instrumentation*) en 1997, basada en bus PCI (*Peripheral Component Interconnection*), dirigida al manejo de instrumentos modulares en tarjetas. Mas aún, la popularidad de tecnologías de uso común han dado paso al desarrollo de nuevas arquitecturas, como LXI (*LAN eXtensions for Instrumentation*) en 2004, que buscan llevar a los sistemas ATE a nuevos horizontes.

Si bien, los avances tecnológicos, han acrecentado los componentes englobados alrededor de la construcción de una solución completa de medición y automatización, siguen siendo conceptos básicos como la adquisición de datos y el control de instrumentos, elementos imprescindibles en el desarrollo de los actuales sistemas ATE.

Día a día, la construcción de sistemas de medición y automatización eficientes, presenta desafíos para la industria productora de hardware y software destinados a esta tarea. Nuevos instrumentos, procesos cada vez más especializados, operaciones remotas, etc., hacen necesario el desarrollo de aplicaciones muy concretas bajo un entorno consistente de programación. Sin embargo el uso de hardware y software con mayores capacidades resulta en ocasiones no ser la solución más apropiada, siendo entonces alternativas como la elección adecuada de procedimientos, la respuesta más rápida y económica al desarrollo de sistemas calificados como eficientes. El manejo de protocolos de intercambio de mensajes entre dispositivos, así como el empleo de nuevas técnicas de programación, representan algunos de estos procedimientos. Dentro del control de instrumentos uno de estos mecanismos principales, es el empleo de argumentos SCPI, que si bien requieren de su comprensión, proporcionan también beneficios como el control total de dispositivos y la minimización de transacciones, esenciales en la construcción de dichos sistemas.

1.1. Planteamiento y objetivos de la tesis

El objetivo del presente trabajo es plantear una serie de directrices dirigidas al control de instrumentos mediante SCPI que permitan el desarrollo de aplicaciones independientes del protocolo de comunicaciones y/o la plataforma de desarrollo empleados. Dichas directrices serán ejemplificadas mediante la implementación de una herramienta software para el manejo del Analizador de Espectros ESA-L E4403B de la firma Agilent. El trabajo realizado en esta tesis forma parte de una línea de investigación, abierta por el cuerpo Académico de Redes de Instrumentación del Instituto de Electrónica y Computación de la Universidad Tecnológica de la Mixteca, consistente en el estudio de técnicas de descripción formal para el desarrollo de sistemas de instrumentación programable y protocolos de comunicaciones industriales y de instrumentación. Dichas técnicas permiten el diseño de aplicaciones que cumplan con los estándares para el desarrollo de software de calidad y aporten herramientas académicas para el conocimiento de los protocolos de comunicación en estudio.

Una vez planteado el objetivo principal y el ámbito donde se centra el trabajo de investigación se proponen los siguientes objetivos particulares:

- Realizar un estudio detallado de la norma de ordenes para instrumentos programables SCPI, haciendo énfasis en los criterios para la sintaxis y estilo de las ordenes, el formato de intercambio de datos y la clasificación de instrumentos.
- Aportar un conjunto de directrices que ayuden de manera significativa a reducir el tiempo en el desarrollo de aplicaciones sobre el control y automatización de instrumentos.
- Modelar y construir una herramienta software para el manejo del Analizador de Espectros ESA-L E4403B mediante GPIB y el uso de Agilent VEE, que muestre los procedimientos y beneficios aportados por las directrices planteadas. El modelado será realizado de acuerdo a las pautas que la Ingeniería de Software demanda para la implementación de un software de calidad.

La hipótesis origen de la realización de este trabajo, encuentra su base en el estudio de la norma SCPI, que permita, aportar un conjunto de directrices encaminadas al control de instrumentos SCPI; facilitando de esta forma el desarrollo de aplicaciones que busquen la utilización adecuada de los recursos que acompañan a un instrumento en particular y por tanto la optimización en el tiempo de gestión con su controlador.

1.2. Estructura de la tesis

La representación del trabajo realizado se estructura como muestra la Figura 1.1 y de la forma que se detalla a continuación.

El capítulo 2 presenta un marco conceptual alrededor del control de instrumentos, exponiendo los tipos de instrumentos programables existentes, sus características y plataformas de software comúnmente utilizadas en su control. De manera adjunta a esta sección, se exhibe al Analizador de Espectros como un ejemplo particular de instrumento a controlar.

El capítulo 3 describe la norma SCPI. Sus antecedentes, objetivos, sintaxis de ordenes y clases de instrumentos son los principales puntos abordados.

El capítulo 4 expone las directrices para el control de instrumentos SCPI generadas de la integración tanto del estudio de la norma como de las aportaciones originadas durante la construcción de la herramienta software propuesta. Los objetivos generales señalados por las directrices cubiertas en esta sección son: la selección de ordenes, el manejo de datos, el acceso a memoria de instrumentos y el control del sistema de registros de estado.

El capítulo 5 muestra el desarrollo de una herramienta software para el manejo del Analizador de Espectros ESA-L E4403B, como producto práctico de las directrices planteadas. El modelado y fases del software en cuestión, así como los resultados obtenidos que dan validez al proyecto de tesis establecido, son discutidos en este capítulo.

La exposición del trabajo de esta tesis finaliza con una serie de conclusiones y posibles líneas de investigación, las cuales se exponen en el capítulo 6.

Como punto final se presentan las referencias bibliográficas utilizadas en la realización de esta tesis, así como un anexo correspondiente a una guía práctica de ayuda para el alcance de la conectividad con instrumentos.

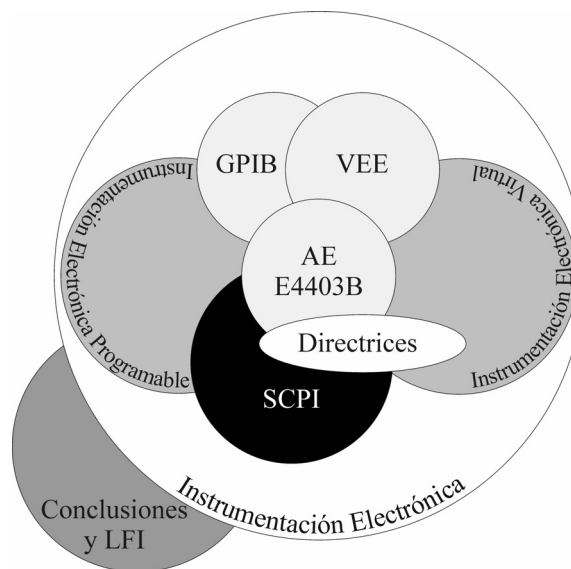


Figura 1.1. Estructura del trabajo de tesis.

2. Marco conceptual

Por muchos años, los sistemas de alto desempeño militares y de manufactura han requerido equipo de prueba automatizado de alto desempeño. A medida que la tecnología evoluciona, las tecnologías comerciales se desarrollan hasta igualar o exceder las tecnologías normalmente usadas en estos sistemas ATE. Igual que las computadoras que llenaban una habitación completa y ahora se ubican en un escritorio, los componentes centrales de los sistemas de medición se han reducido dramáticamente en tamaño mientras incrementan su diversidad y desempeño.

De la misma manera que el hardware empleado en el desarrollo de sistemas de prueba modernos han cambiado, los procedimientos y técnicas empleados en su construcción también han sufrido modificaciones, llevando a ramas como la instrumentación a impulsar nuevos conceptos, ahora ya consolidados como la instrumentación virtual, donde el software juega ahora, el rol mas importante.

Hoy en día, el software resulta ser un elemento esencial en todos los sistemas de pruebas, desde la verificación del diseño, hasta pruebas de manufacturación altamente automatizadas. La construcción de estos sistemas de medición y automatización requiere de un conjunto integrado de herramientas de software de desarrollo. Estas herramientas incluyen administradores de sistema, ambientes de desarrollo y servicios de control y medición. El software de administración de sistema hace referencia usualmente al software que un usuario desarrolla para una aplicación particular. El software para servicios de medición y control permite la manipulación de los medios físicos para la interacción entre el dispositivo a controlar y el que lleva el control. Ambos, tanto el software de administración como el de servicios, se integran en los llamados ambientes de desarrollo de aplicaciones (ADEs, *Application Development Environments*) con el propósito de lograr el máximo desempeño del sistema y productividad en el desarrollo [27].

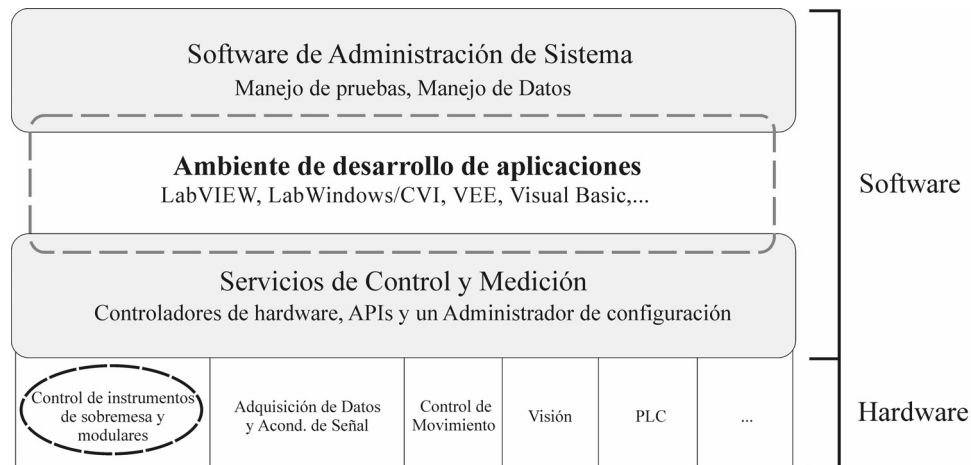


Figura 2.1. Componentes de un sistema de medición y automatización.

Las soluciones de medición y automatización hoy en día, integran a su construcción, equipos de adquisición de datos (DAQ, *Data Acquisition*), productos para el control de movimiento, soluciones de visión, instrumentos autónomos y modulares, entre otros elementos.

Aunque en la actualidad, el control de instrumentos es considerado solo una parte del entorno bajo el cual se desarrollan los sistemas de medición y automatización modernos, la inevitable presencia de instrumentos de medición como analizadores de señales, osciloscopios, etc. en estos sistemas, enfatizan su importancia y motivan su adaptación a las nuevas tecnologías, haciendo ineludible así, el estudio de las variables que acompañan a esta tarea, y que permiten mantenerla como punta de lanza en el desarrollo de los mencionados sistemas de medición.

2.1. Control de instrumentos

A lo largo del tiempo, tecnologías como GPIB, VXI, PXI, etc. han sido participes de la evolución del control de instrumentos, llevando al control clásico de un instrumentos mediante una PC, al empleo de arquitecturas modulares como VXI, PXI o mas recientemente LXI [URL5].

Dentro del control de instrumentos se pueden denotar dos roles, el asumido por el instrumento y el tomado por el elemento que realiza este control o bien llamado controlador. El tipo de instrumento utilizado determina los medios y procedimientos empleados para su control. El papel del controlador es encomendado normalmente a las computadoras personales (PCs, *Personal Computers*). Procesadores funcionando a velocidades de gigahertz, buses de alta velocidad, así como su amplia disponibilidad de software y bajo precio, hacen de la PC una plataforma ideal de prueba.

2.1.1. Tipos de instrumentos

Los diversos escenarios bajo las cuales son concebidos los instrumentos pueden ser agrupados en dos grandes conjuntos: instrumentos de sobremesa (*rack-and-stack*) y modulares (*cardcage*).

Los instrumentos de sobremesa, también llamados autónomos, son dispositivos usados de manera independiente (manualmente), sin embargo, estos no están ajenos al empleo un controlador externo, en base al empleo de una interfaz como GPIB, Serial, LAN, etc. A diferencia de los anteriores, los instrumentos modulares, son construidos dentro de tarjetas insertables, que requieren el uso de una infraestructura para su manipulación. Arquitecturas como VXI, CompactPCI y PXI posibilitan el uso de instrumentos modulares en tarjetas.

2.1.1.1. Instrumentos de sobremesa

Históricamente, la tarea de comunicación y control de instrumentos de sobremesa, ha sido desarrollado a través del uso de buses consolidados como GPIB y RS-232. Sin embargo, la popularidad de algunos buses comerciales tales como, Ethernet y USB ha llevado a estos, a incursionar de manera importante en el área de la instrumentación electrónica programable.

La Figura 2.2 muestra el esquema básico, empleado en cualquier aplicación de control de instrumentos de sobremesa. El uso de una computadora como medio de control es bastante común, sus múltiples prestaciones facilitan el manejo de diversas interfaces, transparentando al usuario la tarea del empleo de algún bus interno para la manipulación de los datos, consiguiendo así que sea solo el tipo de sistema de comunicación o interfaz empleada, la que sugerirá los medios y procedimientos usados en el control del instrumento. La utilización de sistemas de control particularmente diseñados para esta tarea, también es una solución, aunque poco usada.



Figura 2.2. Esquema básico para control de instrumentos de sobremesa.

2.1.1.1.1. GPIB

Por muchos años el bus de instrumentación de propósito general GPIB (*General Purpose Interface Bus*) ha sido uno de las interfaces más comunes en instrumentos de prueba y medida. Desarrollado en 1965 por la firma HP bajo el nombre HPIB (*Hewlett Packard Interface Bus*). Las particularidades de su diseño, encaminado a aplicaciones sobre el control de instrumentos, lo llevo a alcanzar su estandarización en 1975 bajo la denominación IEEE 488 o simplemente GPIB. Concebido en una primera versión (conocida ahora como 488.1), las limitaciones de la norma motivaron a que se diera lugar a una extensión o segunda versión (488.2) en 1987. La definición de la norma SCPI en 1990, permitió precisar las reglas para la abreviación de las ordenes del estándar IEEE 488.2.

La primera versión para el estándar IEEE 488 detalló las especificaciones mecánicas, eléctricas y funcionales para el bus en cuestión, entre las que destacan los siguientes [22]:

- *Tipos de mensajes*: dependientes del dispositivo y de interfaz con el controlador del bus.
- *Roles de los instrumentos*: como emisor (*talker*), con capacidad de enviar datos por el bus; como receptor (*listener*), con capacidad de acepta datos y ordenes del bus; y como controlador (*controller*), con capacidad de gestionar el bus, enviar ordenes, solicitar el estado de los dispositivos, así como controlar el flujo de datos.
- *Máximo de dispositivo conectados al bus*: establecido en 15, siendo necesariamente uno de ellos el controlador. En un sistema pueden existir varios controladores, aunque solo uno deberá encontrarse activo en un momento determinado. Más de la mitad de los instrumentos conectados deben de estar activos.
- *Configuraciones del bus*: en estrella o serie, con una longitud de cable máxima limitada a 20m, siendo la distancia máxima entre instrumentos de 2m.
- *Tipo de conector*: trapezoidal de 24 contactos, macho y hembra a la vez.
- *Transferencia de mensajes*: asíncrona, con una velocidad máxima de 1Mbps, para distancias largas esta puede disminuir hasta 200 Kbps.
- *Estructura del bus*: conformada de 16 líneas de señales y 8 de tierra. El bus utiliza lógica negativa con niveles TTL en sus líneas de señales, de las cuales 8 son líneas de datos; 3 son líneas de control del bus (*handshake*) para la transferencia de datos; y 5 son líneas de gestión de control y estado del bus.

La revisión de la norma IEEE 488.1 en 1987, introdujo una extensión o segunda versión de esta, la IEEE 488.2. Este hecho supuso un paso más hacia la compatibilidad de instrumentos, introduciendo soluciones como [24]:

- Especificar la estructura de los datos sobre el bus (en código ASCII, hexadecimal y octal).
- Precisar el protocolo para el envío de mensajes de dispositivo, así como la forma de enviar varios mensajes en una sola cadena de caracteres.
- Proporcionar un conjunto de ordenes comunes a todos los instrumentos: *CLS, *ESE, *ESE?, *ESR?, *IDN?, *OPC, *OPC?, *RST, *SRE, *SRE?, *STB?, *TST?, *WAI.
- Concretar el modelo estándar de bytes de información sobre el estado del dispositivo. De forma tal, que la norma obligue a los instrumentos al soporte de una estructura de bytes donde se guarde y actualice su estado. El byte de estado de la norma 488.1 es solo parte de esta nueva estructura.

Aportaciones como la hecha en 1993 por la empresa Alliance System *VXIplug&play*, con la introducción de la herramienta VISA (*Virtual Instrument Software Architecture*) la cual define interfaces de comunicaciones con los instrumentos VXI, GPIB y serial; y las impulsadas por la firma National Instruments con la presentación de un protocolo de comunicaciones GPIB de alta velocidad,

denominado HS 488 (hasta 8MBps), han fortalecido el uso de esta interfaz, no obstante su longevidad y limitaciones mostradas.

Los más grandes obstáculos atrás de GPIB los representan su tamaño y costo en conectores y cables, así como la necesidad de la instalación de un dispositivo de interfaz, dado que GPIB no es una interfaz estándar en PCs. Sin embargo su facilidad de conexión y manejo, su excelente fiabilidad eléctrica y mecánica, además de su diversidad de usos, han hecho de GPIB la opción más clara en aplicaciones como el control de instrumentos y automatización de procesos por más de dos décadas.

2.1.1.1.2. Serial

A pesar de su longevidad y particularidades, las interfaces seriales RS-232, RS-422 y RS-485, siguen hoy presentes en muchos dispositivos usados en aplicaciones industriales y de laboratorio [10, 27].

RS-232 es el más antiguo de los protocolos de comunicación serial usado en aplicaciones de prueba y medida. Introducido en 1962, esta sencilla interfaz es usada aun por algunos instrumentos analíticos y científicos, así como módems e impresoras. Si bien, los interfaces RS-422 y RS-485 exhiben mejoras significativas respecto a su antecesor RS-232, sus aportaciones se limitan a un incremento en el número de dispositivos manejados por la interfaz, y mejoras a su inmunidad al ruido.

Los principales beneficios de las interfaces seriales como medio de comunicación, los representan su bajo costo y ubicuidad entre PCs y viejos instrumentos. Sin embargo, su baja velocidad (alrededor de 20kbytes/s) y primitiva comunicación, han hecho de este un protocolo decadente. Eventualmente esta interfaz ha sido reemplazada, sin embargo su existencia en algunos dispositivos obliga aun a los usuarios, a luchar con sus peculiaridades y limitaciones.

2.1.1.1.3. Ethernet

Desarrollado por XEROX, DIGITAL e INTEL en la década de los 70's, y posteriormente estandarizado bajo el nombre IEEE 802.3 (CSMA/CD), su bajo costo y alta velocidad de transmisión ha convertido a Ethernet en el protocolo de comunicaciones más utilizado en los negocios y aplicaciones de tecnología de información. Las redes Ethernet más comunes, hoy en día permiten transferencias de datos a 10Mb/s, 100Mb/s, 1Gb/s y 10Gb/s. La utilización habitual de Ethernet en redes de área local (LAN, *Local Area Network*) ha conseguido adoptar en muchos casos a estos dos conceptos como sinónimos [9].

Su capacidad natural para el soporte de intercambio de información y distribución de recursos de hardware dentro de una red de equipos terminales, ha llevado a Ethernet a ser una de las mejores opciones para resolver un alto volumen de tráfico de I/O (*Input/Output*) en diversas aplicaciones.

La inclusión de Ethernet en aplicaciones para el control de instrumentos es una tecnología madura, y ampliamente usada en sistemas de medición. Ventajas debido al uso de Ethernet en el control de instrumentos incluyen control a distancia, facilidades de integración y publicación de datos.

Aunque la adición de conectores LAN en instrumentos recientes es cada vez más común, la aplicación de esta tecnología permite de igual forma la conexión de las actuales generaciones de PCs a sistemas y equipos con interfaces como GPIB, mediante dispositivos como pasarelas GPIB/LAN, entre otros.

Los beneficios observados por las conexiones LAN ha dado paso incluso al desarrollo de plataformas modulares basada en LAN como LXI (*LAN eXtensions for Instrumentation*) que representa un estándar de nueva generación para sistemas ATE.

2.1.1.1.4. USB

Ampliamente disponible en PCs, el bus serial universal (USB, *Universal Serial Bus*) es hoy una de las alternativas más usadas en el desarrollo rápido de sistemas de prueba [10].

Introducido al mercado en conjunto con el sistema Microsoft Windows 95, su demanda llevó al bus USB rápidamente a una primera revisión en 1996, numerada como 1.0. Pronto, el surgimiento de Microsoft Windows 98 permitió la incursión de una nueva versión, llamada USB 1.1 (12Mbits/s). Sin embargo, su aun limitado ancho de banda no permitía su uso común en áreas como la instrumentación electrónica programable, siendo finalmente la aparición de su versión 2.0 (480Mbits/s) en 2001, la que colocó al bus USB como una de las interfaces más usadas en el control de dispositivos.

Las utilidades ofrecidas por USB han hecho cada vez más común el desarrollo tanto de instrumentos con interfaz USB, como de dispositivos convertidores de esta a otras interfaces. El uso de USB es particularmente atractivo en entornos donde los instrumentos son frecuentemente movidos, compartidos y reconfigurados. Muchos laboratorios e instalaciones de producción son conformados de una mezcla de viejos y nuevos instrumentos. La inclusión de instrumentos GPIB o RS-232 en un sistema USB requiere solo del manejo de dispositivos de conversión entre interfaces. Algunos de los principales beneficios aportados por USB son su alta popularidad en PCs, bajo costo, conexión simple (*plug&play*), flexibilidad en los niveles de velocidad manejados acorde a la variedad de dispositivos existentes y conexión simultánea de más de 128 dispositivos.

2.1.1.2. Instrumentos modulares

Los instrumentos modulares combinan hardware de medición de alto desempeño y software de medición optimizado para suministrar la funcionalidad de los instrumentos electrónicos más populares. Los instrumentos modulares de hoy ofrecen: analizadores de señales de RF, digitalizadores y osciloscopios de alta velocidad, multímetros digitales, generadores de funciones, entre otros [27].

Una característica importante de los instrumentos modulares es la necesidad del uso de una infraestructura que permita alojar, administrar y controlar estos. Las arquitecturas más usadas para estas plataformas modulares son VXI, CompactPCI y PXI. Otras arquitecturas como PCI, integradas dentro de PCs, evitan el uso una estructura externa. Arquitecturas más recientes como LXI (2004) todavía se encuentran en un proceso de acreditación.

La Figura 2.3 muestra los medios, que de manera general son empleados en el control de arquitecturas VXI, CompactPCI y PXI. Existen básicamente dos procedimientos para el control de una arquitectura modular de este tipo. El primero, de mayor uso y eficiencia es el empleo de un controlador empotrado, que no es más que una PC en una tarjeta. Un segundo procedimiento, de menor empleo, es el manejo de una PC, o bien, otro sistema VXI, CompactPCI o PXI, como medio de control.

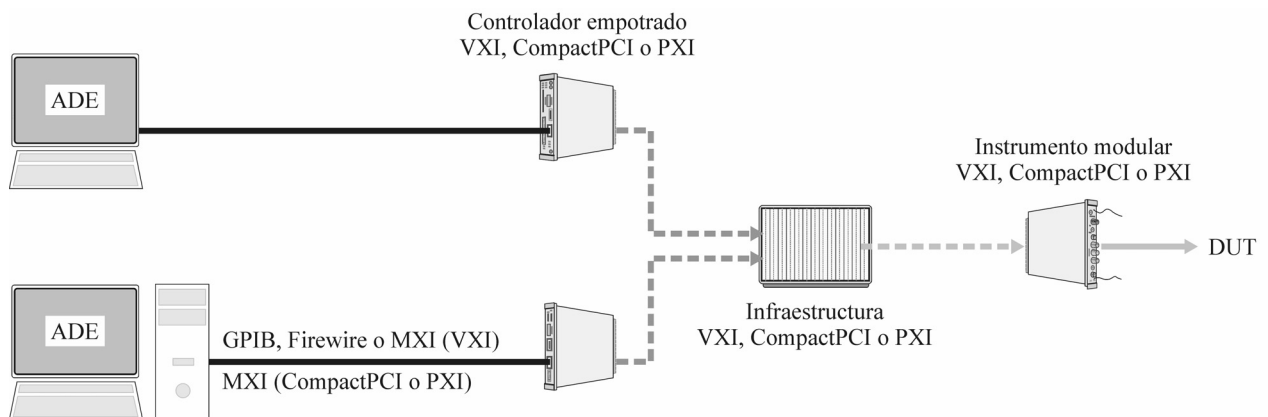


Figura 2.3. Esquema general para control de sistemas VXI, CompactPCI y PXI.

2.1.1.2.1. VXI

VXI es una arquitectura abierta para sistemas modulares que permite operar instrumentos modulares en tarjetas de diferentes fabricantes, sobre una misma infraestructura. El bus VXI (*VMEbus eXtension for Instrumentation*) basada en el bus VME (*Versa Module Europe*), es un estándar desarrollado por diferentes fabricantes para la construcción modular de instrumentos de medida [11, 33].

Históricamente, el bus VXI fue desarrollado en respuesta a la demanda de módulos de instrumentación VME. Durante 1987, cinco compañías líderes en el sector: Colorado Data Systems, Hewlett Packard, Racal Dana Instruments, Tektronix y Wavetek, anunciaron la intención de fabricar instrumentos basados en un estándar desarrollado conjuntamente, denominado VXI, que sería una mejora del bus VME, para aplicaciones de instrumentación. VXI ha llegado a ser un estándar aceptado por centenares de fabricantes dentro del ámbito de la instrumentación modular. Su estandarización se encuentra bajo la denominación IEEE-1155. El organismo regulador actual de este estándar es llamado consorcio VXI.

Alojadas dentro de una estructura o chasis, los módulos ó tarjetas VXI cumplen funciones tan diversas como las realizadas por dispositivos de adquisición de datos, instrumentos, interfaces, etc. Buses VME, sistemas de disparo y señales de reloj (10 y 100MHz) son algunos de los elementos incluidos en la placa interna del chasis (*backplane*), que permite el albergue de los módulos.

Los módulos son definidos en cuatro tamaños. Los dos mas pequeños, A y B, son los tamaños definidos para el bus VME, mientras que los más grandes, C y D, son adiciones pensadas para instrumentos más complejos. En la práctica, aproximadamente el 90% de los sistemas VXI del mercado utilizan el tamaño C. Si bien existe una relación entre los tamaños definidos para los módulos VXI y los manejados por el bus VME (3U, 6U y 9U), los términos asignados (A, B, C y D) corresponden solo con las especificaciones del bus VXI. De igual forma, VXI define tres tipos de conectores DIN de 96 pines denominados P1, P2 y P3, para la conexión de los módulos al chasis. Así, los módulos tipo A cuentan con un conector hembra (P1), los tipo B y C con dos (P1 y P2) y los tipo C con tres (P1, P2 y P3).

Como lo muestra la Figura 2.3, existen dos medios para controlar un sistema VXI: el uso de un bus para un control externo mediante otro dispositivo llámese PC, sistema VXI, etc., o bien el empleo de un controlador local empotrado. El uso de los buses GPIB, Firewire y MXI (*Multisystem eXtension Interface*) para la conexión se realiza mediante una interfaz GPIB/Firewire/MXI-VXI en un tarjeta, instalada sobre la ranura de control del sistema (ranura 0). La utilización de un controlador empotrado en una placa VXI, sobre la ranura correspondiente, da lugar a un mejor uso de la infraestructura VXI, obteniendo la máxima velocidad en la transmisión de ordenes al sistema y evitando el uso de otros protocolos de comunicación.

2.1.1.2.2. PCI/CompactPCI

Introducido por Intel en 1993 como un bus local, de bajo nivel, síncrono e independiente del procesador de 32 bits a 33 MHz, el bus PCI (*Peripheral Component Interconnection*) se ha convertido hoy en día en un estándar genérico para la industria de PC. Su alta tasa de transferencia de datos entre dispositivos y su expansión a través de puentes PCI-PCI son algunas de las razones que le han otorgado a este bus su gran popularidad. Hoy en día, es el PCISIG (*PCI Special Interest Group*) el organismo encargado de la evaluación y especificación del bus PCI [29, URL9].

Una arquitectura PCI implementada dentro de una computadora como plataforma de comunicación interna, normalmente permite la adición de cuatro o cinco ranuras de expansión del bus. Estas ranuras PCI son comúnmente usadas para adherir al control del bus y de manera inherente a la computadora, dispositivos de diferente índole, como controladores de interfaces, tarjetas de adquisición de datos e instrumentos modulares.

A pesar de la popularidad del bus PCI en aplicaciones de prueba y medida, la creciente necesidad de una arquitectura más robusta que permitiera en parte, un mayor número de ranuras de expansión y por tanto el control de una mayor cantidad de dispositivos, motivó en 1995, al desarrollo de una plataforma para computadoras industriales denominada CompactPCI, respaldada y normalizada por la organización PICMG (*PCI Industrial Computer Manufacturers Group*) [28, URL8].

La tecnología de bus CompactPCI combina las características eléctricas del bus PCI, con el estándar mecánico industrial Eurocard popularizado por el bus VME y la tecnología de conectores IEC (*Internacional Electrotechnical Commission*).

Un sistema CompactPCI es alojado usualmente dentro de una estructura rígida (chasis) que cobija, en su parte interior trasera (*backplane*), una placa con alrededor de 8 ranuras CompactPCI, para la inserción de módulos ó tarjetas. Una de estas ranuras, es destinada para el albergue de un módulo de sistema, mientras las restantes realizan el manejo de periféricos. La ranura destinada al sistema facilita la inicialización del sistema, el procesado de interrupciones y la distribución de la señal de reloj a todo el sistema; acciones que permiten el control y comunicación con los dispositivos conectados. La utilización de módulos puentes PCI permite de manera sencilla la extensión de un sistema CompactPCI mediante el agregado de otro.

Los módulos/tarjetas contemplados por CompactPCI son definidos en dos tamaños, de acuerdo al estándar Eurocard, los módulos/tarjetas 3U de 100x160 mm., y los 6U de 233.35x160 mm. Los conectores hembras usados en los módulos son del tipo industrial IEC. Tanto los módulos 3U y 6U, cuentan con dos conectores, uno tipo J1 (110 pines) dedicado al manejo de las señales de 32 bits PCI y otro tipo J2 (110 pines) consagrado a señales de 64 bits PCI, sin embargo los módulos 6U poseen hasta tres conectores adicionales J3 (95 pines), J4 (110 pines) y J5 (110 pines), destinados para aplicaciones particulares. Un tipo de conector opcional usado como puente a otros buses como VME o ISA en backplanes híbridos son los llamados conectores de 2mm.

Todo sistema CompactPCI está provisto de un controlador, que desempeña las funciones de administración del mismo. Casi siempre localizado en la primera ranura del chasis, un controlador puede ser de índole local o remoto. De manera local, el control del sistema se realiza mediante un módulo de sistema empotrado, constituido con componentes estándar de PCs. En contraste, un control remoto en base a una PC u otro dispositivo se lleva a cabo con la ayuda de un módulo de interfaz MXI, anclado dentro del chasis, que aprueba la manipulación de los módulos periféricos que se encuentren activos en el sistema.

2.1.1.2.3. PXI

PXI (*PCI eXtensions for Instrumentation*) es una arquitectura de instrumentación modular, que combina las características eléctricas del bus PCI, las aportaciones mecánicas Eurocard y IEC retomadas por CompactPCI con buses de disparo y sincronización, así como con nuevas características de software, para concebir una plataforma de alto rendimiento y bajo costo para el desarrollo de sistemas de medición y automatización [31, URL10].

Desarrollado en 1997 y divulgado en 1998, PXI fue introducido como un estándar industrial abierto, que diera solución a la creciente demanda de sistemas complejos de instrumentación. PXI es regulado por la PXISA (*PXI Systems Alliance*), un grupo de más de 60 compañías que promueven el estándar, aseguran su interoperabilidad y delimitan su especificación.

Un sistema PXI provee una estructura rígida y modular casualmente llamada chasis, para el alojamiento de hasta 18 elementos entre módulos y controlador. Características como fuentes de DC y acondicionamientos de algunas señales también son incluidas dentro del chasis. La placa interna de la estructura permite la conexión e interacción entre módulos y controlador mediante buses PCI y locales, señales de reloj del sistema (10 MHz) y buses de sincronización y disparo.

Un elemento indispensable dentro de una infraestructura PXI es el controlador, quien posibilita el control del sistema PXI. Este es situado usualmente en la primera ranura del chasis. Tal como lo muestra la Figura 2.3, un sistema PXI puede ser controlado desde un dispositivo externo mediante un modulo de interfaz, o bien de manera local, usando un controlador empotrado. El control remoto de una infraestructura PXI desde una PC u otro sistema PXI, es realizado mediante interfaces seriales de alta velocidad como lo es MXI. El uso de un controlador empotrado elimina la necesidad de una PC externa. Los controladores empotrados típicamente son construidos usando componentes estándares de PCs tales como un procesador, RAM, disco duro y periféricos estándares como USB, puertos paralelos y seriales. De igual forma, un controlador empotrado también permite la selección de un sistema operativo como Windows 2000/XP o LabVIEW Real-Time para la administración del sistema.

Los módulos o tarjetas periféricas insertadas dentro del chasis de un sistema PXI cubren una gran cantidad de funciones. La diversidad de módulos periféricos obedece al creciente uso de la arquitectura PXI en diversas áreas, llegando a ofrecerse en el mercado actualmente más de 1,000 productos. Estos productos incluyen: instrumentos, tarjetas de adquisición de datos, procesadores de señales e imagen, buses de interfaz, etc.

Al igual que la arquitectura CompactPCI, PXI define dos tamaños para los módulos/tarjetas usados, un tamaño pequeño llamado 3U con dos conectores (J1 y J2), y otro tamaño más grande denominado 6U con cuatro conectores (J1, J2, J3 y J4). Sin embargo las funciones asignadas para cada conjunto de líneas agrupadas en cada conector son distintas. Todas las características PXI son implementadas en el conector J2, mientras J3, J4 y J5 son reservados para particularidades PXI, o bien, empleados para propósitos generales. Las módulos de tipo 3U pueden ser utilizadas en infraestructuras PXI 6U mediante el uso de un adaptador.

Una característica importante ofrecida por la arquitectura PXI es el mantenimiento de la interoperabilidad con productos CompactPCI. Así, un modulo PXI puede ser usado dentro de un sistema CompactPCI y viceversa, con la reserva que, las características PXI de un modulo de este tipo no podrán ser desarrolladas dentro de una arquitectura CompactPCI.

2.1.2. Ambiente de desarrollo de aplicaciones

Un ambiente de desarrollo de aplicaciones, dentro del campo de la instrumentación electrónica programable, se entiende como un lenguaje o entorno mediante el cual es posible el desarrollo de programas para el control y coordinación de todos los instrumentos de un sistema [27].

La elección del entorno bajo el cual es desarrollada una aplicación, independientemente de su índole, se encuentra definida tanto por los objetivos trazados, como por los medios con los que se cuentan. Aunque no existen una serie de características específicas para el software usado en el control de instrumentos, algunos criterios útiles en la elección de estos, son [URL7]:

- *Conectividad con instrumentos*: los medios que un software presenta para el control de instrumentos y la forma en como estos son utilizados, determinara en gran medida el tiempo de desarrollo de la aplicación. Así, existirá una considerable diferencia entre el uso de una orden nativa (SCPI) donde un conocimiento del formato de intercambio de mensajes es necesario y la utilización de un controlador cuyo compilador asociado realiza la mayor parte de las operaciones, presentando al usuario objetos simples de manipulación.
- *Capacidades de análisis y presentación*: una vez que los datos han sido adquiridos desde un instrumento, se hace evidente el uso de algoritmos y funciones diseñadas especialmente para el análisis y procesamientos de señales. La inclusión de robustas capacidades de análisis y presentación facilitarían no solo generar, modificar, procesar y analizar la información ob-

tenida, sino también el desarrollo de una interfaz gráfica de usuario (GUI, *Graphical User Interface*) que cumpla con los propósitos de la aplicación.

Generalmente, el desarrollo de aplicaciones encaminadas al control de instrumentos, en base a algún protocolo de comunicaciones como GPIB, Serial o VXI, dentro de un entorno de instrumentación programable, se lleva a cabo mediante herramientas de software propietarias, como lo son VEE de la firma Agilent Technologies o LabVIEW y LabWindows/CVI de National Instruments. Sin embargo, esta no es una condicionante, que evite que lenguajes de propósito específico como C/C++ o BASIC gocen de una gran popularidad dentro del ámbito de la instrumentación programable.

2.1.2.1. C/C++

Una de las plataformas de desarrollo de software más populares hoy en día sigue siendo el lenguaje de programación C. Desarrollado entre 1969 y 1973 por Dennis Ritchie en los laboratorios Bell y diseñado inicialmente para ser usado en un entorno de programación bajo UNIX, su eficiencia y flexibilidad le permitieron rápidamente su expansión a otros sistemas operativos y diversos campos de aplicación, alcanzando su estandarización en 1989 bajo el nombre de ANSI C [URL3].

La aparición de la programación orientada a objetos (OOP, *Object Oriented Programming*) permitió la extensión del lenguaje C, a una primera versión llamada "C con clases", que más tarde en 1985 daría paso al lenguaje C++. Esta nueva transición de C concebida por Bjarne Stroustrup, adhirió al ya popular lenguaje conceptos como clases y herencia propios de la OOP, que potencializaron aun más su uso.

Al igual que otros campos de la ciencia, la instrumentación electrónica programable no ha sido ajena al manejo del divulgado uso de C/C++. Una gran variedad de aplicaciones para el control de instrumentos son escritas en estos lenguajes.

El desarrollo común, de una aplicación para control de instrumentos, bajo un lenguaje como C/C++ requiere la inclusión de rutinas o librerías, que permitan el manejo del hardware bajo control. Así librerías de I/O (*Input/Output*) VISA (*Virtual Instrument Software Architecture*) como NI-VISA de National Instrument ó Agilent VTL (*VISA Transition Library*) de la firma Agilent pueden ser empleadas. Otras librerías propias de algunos desarrolladores también pueden ser utilizadas, tal es el caso de SCLI (*Standard Instrument Control Library*) de Agilent. A continuación se muestra el código de un programa en C/C++, que permite obtener el identificador de un instrumento GPIB [4, 14].

```
#include <stdio.h>
#include <conio.h>
#include "visa.h"
void main(){
    ViSession defaultRM, vi;
    ViChar buf[256] = {0};
    /*Abrir la sesión para el dispositivo GPIB en la dirección 18*/
    ViOpenDefaultRM (&defaultRM); //Inicializar manejador de recursos
    viOpen (defaultRM, "GPIB0::18::INSTR", VI_NULL,VI_NULL, &vi);
    /*Enviar un orden de identificación *IDN? al dispositivo */
    viPrintf (vi, "*IDN?\n");
    /*Leer identificador */
    viScanf (vi, "%t", &buf);
    printf ("Cadena de identificación: %s\n",buf);
    /*Cerrar sesión*/
    viClose (vi);
    viClose (defaultRM);
    getch();
}
```

2.1.2.2. Basic

Diseñado en 1983 para fines educativos por John Kemeny y Thomas Kurtz BASIC (*standing for Beginner's All Purpose Symbolic Instruction Code*) se ha convertido en uno de los lenguajes de programación más comúnmente usados. De naturaleza secuencial, su simpleza fue la clave de su éxito y popularidad [URL6].

El tiempo trajo consigo el desarrollo de múltiples versiones de este lenguaje por parte de Microsoft, hasta que finalmente en 1991 un desarrollo llamada VB (*Visual Basic*) vino a consolidar este lenguaje. VB es hoy usado no solo para crear rápidas y simples interfaces con otros productos Microsoft tales como Excel y Access sin necesidad de gran cantidad de código, sino de igual forma es empleado en aplicaciones complejas de diversa índole.

La implementación de aplicaciones para el control de instrumentos mediante VB conlleva la adición de librerías que permitan esta tarea. Librerías como NI-VISA, Agilent VTL o SICL pueden ser incluidas como módulos al proyecto. A continuación se muestra el código de un programa en VB que permite obtener el identificador de un instrumento GPIB [4, 14].

```
Private Sub Main()
    Dim status As Long 'Código regresado por la función VISA
    Dim defrm As Long 'Manejador por defecto
    Dim vi As Long 'Sesión del instrumento
    'Abrir la sesión para el dispositivo GPIB en la dirección 18
    status=viOpenDefaultRM(defrm) 'Inicializar manejador de recursos
    status=viOpen(defrm, "GPIB0::18::INSTR", 0, 0, vi)
    If(status<0) Then GoTo VisaErrorHandler
    'Enviar un orden de identificación *IDN? al dispositivo
    status=viVPrintf(vi, "*IDN?" + Chr$(10), 0)
    'Leer identificador
    status=viVScanf(vi, "%t", strRes)
    If(status<0) Then GoTo VisaErrorHandler
    MsgBox (strRes)
    'Cerrar sesión
    Call viClose(vi)
    Call viClose(defrm)
VisaErrorHandler:
    MsgBox ("Error")
End Sub
```

2.1.2.3. LabVIEW

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) de la firma National Instruments es un programa de desarrollo basado en un ambiente gráfico destinado a realizar prototipos, diseños y sistemas de adquisición de datos, instrumentación y sistemas de control. Los programas en LabVIEW son llamados instrumentos virtuales (VIs, *Virtual Instruments*). Cada VI contiene tres partes principales: panel frontal, diagrama de bloques e icono/conector [24, URL7].

LabVIEW cuenta con bibliotecas de funciones dedicadas a la adquisición de datos, acondicionamiento de señal, comunicación serial, GPIB, VXI, PXI, control de instrumentos, adquisición de imágenes, control de motores, entre otros, lo que permite realizar programas distribuidos a través de una red bajo entornos Windows 9X/Me/2000/NT/XP, Mac, Linux y Solaris. LabVIEW se puede vincular con diferentes aplicaciones mediante herramientas .NET, ActiveX, WWW, XML, Datasocket, TCP/IP, etc.

LabVIEW contempla el control de instrumentos de manera directa (*direct I/O*) y mediante el uso de controladores universales (*plug&play*) e intercambiables (*IVT*).

La capacidad de control directo ofrecida por este software provee un manejo versátil del instrumento, particularmente en los casos en que los controladores para este no se encuentran disponi-

bles. Sin embargo su uso requiere de un mayor conocimiento sobre el instrumento y el formato de intercambio de mensajes.

Los controladores universales simplifican el control y comunicación con el instrumento a través de un modelo estándar de programación, estos controladores son producidos por *VXIplug&play* Systems Alliance o bien desarrollado dentro del mismo entorno. Los controladores intercambiables, elaborados por la fundación IVI (*Interchangeable Virtual Instrument*), ofrecen funcionalidad y flexibilidad en aplicaciones que requieren de intercambiabilidad y simulación de instrumentos.

La Tabla 2.1 resume las características de los objetos manejados por LabVIEW para el control de instrumentos.

Tabla 2.1. Objetos para el control de instrumentos en LabVIEW.

Objetos LabVIEW	Descripción
Control directo	Proporciona un manejo versátil del instrumento en circunstancias donde el control requerido es complejo o bien los controladores del instrumento no existen. El uso de este tipo de objetos requiere de un conocimiento previo tanto de las características del dispositivo bajo estudio, como de los protocolos de mensajes empleados (SCPI). El uso de este tipo de objetos solo requiere de la presencia de librerías para el control de operaciones (NI-VISA para National Instruments), que facilitan la interacción sobre interfaces GPIB, PXI, Serial, Ethernet, USB, entre otras.
Controlador	Universal
	Simplifica el control y comunicación con un instrumento a través de un simple modelo estándar de programación para todos los controladores. <i>VXIplug&play</i> Systems Alliance proporciona controladores universales que permiten la generación de código fuente nativo para LabVIEW y LabWindows/CVI. Así, un controlador universal en LabVIEW, sea <i>VXIplug&play</i> o diseñado por el usuario, no es más que un conjunto de VIs usados para el control y comunicación con un instrumento programable. Cada VI corresponde a una operación programática tal como una configuración, lectura de, escritura a, o manejo de un evento del instrumento. Estos VIs cumplen con los requerimientos y recomendaciones del estilo de programación citado por LabVIEW, como manejo de errores, paneles frontales, diagramas a bloques, iconos y ayuda en línea. Un controlador universal mantiene una arquitectura e interfaz común, que rápida y fácilmente permita conectar a, y comunicar con, algún instrumento sin la necesidad del desarrollo de una compleja aplicación, que implique el entendimiento de protocolos y paradigmas que envuelven esta tarea.
	Intercambiable
	Incrementa el rendimiento y flexibilidad de complejos sistemas de prueba que requieren intercambiabilidad, monitoreo o simulación de instrumentos. Basados en el estándar industrial desarrollado por la fundación IVI, este tipo de controladores ofrecen un alto desempeño en aplicaciones donde el control del instrumento demande el envío de ordenes que modifiquen su estado de operación. La fundación IVI define dos arquitecturas de controladores intercambiables, una basada en el lenguaje C y otra en un modelo de componentes (COM, <i>Component Object Model</i>) [13], ambas arquitecturas son diseñadas para coexistir. National Instrument hace uso de la arquitectura IVI-C, en base a los criterios de utilidad y longevidad de esta. Controladores IVI-C en LabVIEW incluyen VIs, así como códigos fuente (en C) de estos.

De igual forma, LabVIEW provee objetos para la gestión de eventos, como peticiones de servicio GPIB, activación de interrupciones VXI o incluso excepciones del procesador, empleados habitualmente en el control de instrumentos.

2.1.2.4. LabWindows/CVI

LabWindows/CVI (*Complete Virtual Instrumentation*) es un ambiente completo de desarrollo ANSI C para la creación de aplicaciones de instrumentación virtual. LabWindows/CVI cuenta con medios para la adquisición, análisis y visualización, una interfaz simplificada de edición para usuarios y herramientas automatizadas de generación de código. LabWindows/CVI posee módulos que facilitan el desarrollo de sistemas con necesidades específicas [27, URL7].

LabWindows/CVI es un ambiente completamente integrado, diseñado especialmente para la creación de sistemas de instrumentación basados en GPIB, PXI, VXI, tarjetas de adquisición de datos insertables, etc. El ambiente combina un entorno interactivo de desarrollo de fácil uso con el poder y flexibilidad de la programación de código compilado en ANSI C.

Si bien todos los medios empleados para el control de instrumentos dentro LabWindows/CVI implican la generación de rutinas de software en ANSI C, estas son muy similares a los utilizados en LabVIEW. Controladores universales e intercambiables, así como la opción de un control directo del instrumento, son contemplados de igual forma por LabWindows/CVI. Un aporte adicional de LabWindows/CVI al control de instrumentos es la inclusión de un asistente para el desarrollo de controladores de instrumentos, que permite generar código para el control del dispositivo.

2.1.2.5. VEE

VEE (*Visual Engineering Environment*) de la firma Agilent Technologies es un lenguaje de programación gráfica optimizado para el desarrollo de aplicaciones de prueba y automatización de medidas. Dentro de VEE existen tres medios para el control de instrumentos: objetos de control directo (*direct I/O*), controladores (*drivers*) e importación de bibliotecas (*PC Plug-in I/O*) [23, 17].

Aunque los objetos de control directo en VEE requieren el conocimiento de instrucciones estándares para instrumentos, estos proporcionan gran velocidad de comunicación sin la necesitan de un controlador, siendo por ello la forma de control más utilizada en aplicaciones avanzadas.

Los controladores pueden ser paneles de control (*panel driver*), componentes del controlador (*component driver*), universales (*VXIplug&play*) e instrumentos virtuales intercambiables (*IVI*). En aplicaciones orientadas a la enseñanza, la utilización de paneles de control es útil por su naturaleza educativa y facilidad de uso. Controladores universales e intercambiables proporcionan un control bajo estándares producidos por *VXIplug&play Systems Alliance* y *IVI Foundation* respectivamente. La importación de bibliotecas, sólo es utilizada cuando la interfaz es una tarjeta tipo Plug-in PC.

La Tabla 2.2 resume las características de los objetos manejados por VEE para el control de instrumentos.

Tabla 2.2. Objetos para el control de instrumentos en VEE.

Objeto VEE		Descripción
	Control directo	Permite un acceso total sobre las características programables de cualquier instrumento. Aunque ningún controlador es requerido, su utilización demanda el conocimiento de ordenes de programación del instrumento (SCPI). El uso de estos objetos en la comunicación con dispositivos GPIB, VXI o Serial es dispuesto por las librerías Agilent de I/O incluidas en el software. El soporte de la interfaz LAN por parte de este objeto no incluye instrumentos basados puramente sobre esta interfaz. VEE provee un objeto de control directo para múltiples dispositivos (<i>MultiInstrument Direct I/O</i>), que logra el control de varios instrumentos mediante la edición de un solo objeto, su aplicación es usual en interfaces VXI.
Controlador	Universal	Ofrece una comunicación estándar con instrumentos, citada por <i>VXIplug&play Systems Alliance</i> . <i>VXIplug&play</i> es una especificación de interfaz que permite a fabricantes suministrar hardware y software compatible. Un controlador de este tipo, es una biblioteca de funciones para el control de un instrumento específico. Un

Objeto VEE	Descripción
	<p>controlador universal <i>VXIplug&play</i> es escrito normalmente por el diseñador del instrumento. Este tipo de controladores no son exclusivos de instrumentos VXI, interfaces como GPIB y Serial también son soportadas. El uso de un controlador universal para el control de un instrumento, requiere de la instalación del archivo apropiado del controlador y de las librerías VISA I/O. El primer medio de interacción con un controlador universal en VEE es el objeto <i>To/From VXIplug&play</i>; Objetos Call también pueden ser utilizados para llamar funciones <i>VXIplug&play</i>. Los controladores universales son usados por múltiples plataformas de software (VEE/LabVIEW). Dentro de VEE, este tipo de controladores no pueden ser utilizados en conjunto con ningún otro objeto para el control de instrumentos.</p>
Panel	<p>Brinda un control fácil y educativo de gran variedad de instrumentos. Consienten la definición de un previo estado de medición, que permite especificar todas las funciones del instrumento, de esta manera, cuando un controlador de panel entra en operación, el instrumento es programado automáticamente para hacer coincidir sus parámetros con los definidos en el panel. Este controlador actúa como un panel de control del instrumento interactivo, esta forma es útil durante el desarrollo y depuración de programas. Si el instrumento no es conectado, la representación gráfica asociada, aportada por el controlador, puede ser usada para definir el estado de una medición, o bien, realizar operaciones de simulación. La utilización de este tipo de controlador requiere del archivo del controlador suministrado por VEE (exclusivo de la aplicación), y de las librerías de manejo de operaciones de I/O. Existen instrumentos que no cuentan con un controlador de panel. Este controlador es utilizado sobre interfaces GPIB y VXI. El desarrollo de un controlador de panel es posible siempre y cuando se conozca el conjunto de instrucciones manejadas por el compilador para controladores de instrumentos de VEE.</p>
Componente	<p>Proporciona una mejora de velocidad en las operaciones de I/O provista por un controlador de panel. Al igual un controlador de panel requiere de un archivo controlador de panel de instrumento suministrado por VEE, con la diferencia en su forma de uso. En un controlador de panel, cada función y valor de medición es llamado componente. Un componente es una variable dentro de un controlador que registra las características de una función o el valor de una medición. Así, un componente de controlador es un objeto que lee ó escribe únicamente el elemento que es especificado en sus terminales de entrada o salida. Este controlador es utilizado sobre interfaces GPIB y VXI.</p>
Intercambiable	<p>Facilita el uso de instrumentos provenientes de distintos fabricantes, sin la necesidad de un cambio en el software de control del dispositivo. Existen dos arquitecturas de IVI. Los controladores IVI-C, cuyo su nombre se atribuye al uso del lenguaje C para su desarrollo, permiten el intercambio de instrumentos sin la necesidad de reescribir el software de uso, sin embargo estos solo son soportados por particulares entornos de desarrollo. Los controladores IVI-COM no son limitados por alguna estructura y su uso es más general, VEE únicamente hace uso de este último tipo de controladores.</p>
Importación de bibliotecas	<p>Suministra medios para el manejo de tarjetas tipo Plug-in PC. VEE proporciona dos vías para el control de tarjetas PC plug-in: Interfaces de programación visual (VPI, <i>Visual Programming Interface</i>) y Librerías de enlace dinámico (DLL, <i>Dynamic Link Libraries</i>), ambas suministradas por los fabricantes de tarjetas.</p>

Adicionalmente, VEE provee soporte para todas las operaciones de sondeo serial, definidas por el estándar IEEE 488.1. Todos los instrumentos GPIB y VXI sobrellevan operaciones de sondeo serial. VEE no permite el soporte de operaciones de sondeo paralelo. Dos objetos son proporcionados por VEE para el manejo de operaciones de sondeo; objetos para el manejo de eventos en instrumentos (*Instrument Event*) y de control directo (*direct I/O*). De la misma forma VEE suministra un objeto para la administración de eventos de interfaces (*Interface Event*), como GPIB y Serial.

2.2. Analizador de espectros

El análisis espectral se define como el estudio de la distribución de la energía a través del espectro de frecuencia de una señal eléctrica dada. De manera sencilla un espectro es una colección de ondas sinusoidales que, combinadas apropiadamente producen la señal bajo estudio en el dominio de tiempo. De este estudio se obtiene información acerca del ancho de banda, efectos de diferentes tipos de modulación, generación de señales transitorias, lo cual es útil en el diseño y prueba de sistemas de diferente género.

El analizador de espectros originalmente diseñado para observar la energía de las señales de RF (*Radio Frecuencia*) en aplicaciones de radar, ha llegado a ser un instrumento sofisticado capaz de representar amplitudes en función de frecuencia en una porción del espectro de RF.

2.2.1. Mediciones con analizadores de espectros

Las mediciones más comunes con analizadores de espectros incluyen frecuencia, potencia, modulación, distorsión y ruido. Comprender el contenido espectral de una señal resulta importante, especialmente en sistemas con un ancho de banda limitado. La potencia de transmisión es otra medición clave. Una baja potencia puede expresar que una señal no es capaz de alcanzar su destino, así con un exceso puede advertir un desgaste rápido del suministro, generación de distorsiones o una alta temperatura de operación del sistema [5].

Medir la calidad de la modulación permite asegurar que un sistema trabaje apropiadamente y que la información sea transmitida correctamente por el mismo. Pruebas como el grado de modulación, la amplitud de bandas laterales, la calidad de la modulación y el ancho de banda ocupado son ejemplos comunes de mediciones de modulación analógica. Métricas de modulación digital incluyen magnitud del vector de error (EVM), error de fase contra tiempo y una variedad de otras mediciones.

En comunicaciones, la medida de la distorsión es crítica para ambos el receptor y transmisor. Medidas comunes de distorsión incluyen armónicos, intermodulación y emisiones esporádicas. Excesivos armónicos a la salida de un transmisor pueden interferir con otras bandas de comunicaciones. La etapa de preamplificación en un receptor deberá estar libre de intermodulación para prevenir diafonía en la señal. Un ejemplo es la intermodulación de portadores de TV por cable, que provoca distorsión entre canales sobre el mismo cable.

El ruido es a menudo una señal de interés de medida. Cualquier circuito activo o dispositivo generará una cantidad de ruido. Pruebas como la figura de ruido y la razón señal a ruido (SNR, *Signal to Noise Ratio*) permiten caracterizar el funcionamiento de un dispositivo y su contribución sobre un sistema.

2.2.2. Tipos de analizadores de espectros

Mientras los analizadores de espectros superheterodinos¹ representan la base de estudio del análisis espectral, existen otras arquitecturas de relevancia significativa. Un importante tipo lo son los analizadores de Fourier. Un analizador de este tipo, digitaliza la señal en el dominio del tiempo, para mediante el uso de técnicas de procesamiento digitales de señales (DSP, *Digital Signal Processing*) efectuar una transformada rápida de Fourier (FFT, *Fast Fourier Transform*) que permita desplegar la señal en el dominio de la frecuencia. El uso de una FFT permite caracterizar fenómenos simples de disparo, y la medición de la magnitud y fase de una señal. Sin embargo, los analizadores de Fourier presentan ciertas limitaciones en relación con los analizadores superheterodinos, particularmente en las áreas del rango de frecuencia, sensibilidad y rango dinámico. Los analizadores de Fourier son típicamente usados en aplicaciones de análisis de señales de banda base.

¹ Haciendo referencia al trato de frecuencias por arriba del rango audible.

Aunque bajo una definición diferente a la del análisis espectral, los analizadores vectoriales (VSAs, *Vector Signal Analyzers*) permiten al igual que los analizadores de Fourier, digitalizar señales en el dominio del tiempo, sin embargo estos extienden su capacidad de rango de frecuencias usando convertidores en lugar de digitalizadores. Estos analizadores ofrecen una alta resolución en la medición de espectros, demodulación y análisis avanzados en el dominio del tiempo. Son empleados usualmente para caracterizar señales transitorias, moduladas y en ráfagas usadas en comunicaciones, video y aplicaciones en imágenes de ultrasonido.

Mientras los conceptos de análisis espectral y análisis vectorial son manejados como distintos. El aumento en la complejidad de las señales motivo de análisis, han orillado a las nuevas generaciones de analizadores de espectros a incluir muchas capacidades de análisis vectorial previamente encontradas únicamente en analizadores de Fourier y vectoriales. Avances tecnológicos como pantallas de lento barrido, marcadores de pantalla, diversos tipos de promedios y la transferencia de datos a computadoras e impresoras han permitido esta fusión. Un ejemplo de este hecho es la inclusión de herramientas para el estudio de tecnologías para sistemas de comunicaciones móviles de 2G y 3G en muchos analizadores de espectros.

2.2.3. Fundamentos sobre analizadores de espectros superheterodinos

Mientras la tecnología actual hace todo lo posible por reemplazar muchos de los circuitos analógicos con modernas implementaciones digitales, es útil entender la arquitectura clásica de un analizador de espectros como un punto de partida para el estudio de estos.

La Figura 2.4 muestra un diagrama a bloques simplificado de un analizador de espectros superheterodino. Una señal de entrada pasa de un atenuador y un filtro pasa-bajas a un mezclador, donde ésta, es mezclada con una señal del oscilador local (LO, *Local Oscillator*). Debido a que el mezclador es un dispositivo no-lineal, la salida de este incluirá no solo las 2 señales originales sino también los armónicos de cada una de las señales, así como la suma y la diferencia de las señales junto con sus respectivos armónicos. Si algunas de las señales mezcladas cae dentro de la banda de paso del filtro pasabandas de frecuencia intermedia (IF, *Intermediate Frequency*), estas serán procesadas (amplificadas y quizás comprimidas en una escala logarítmica), esencialmente rectificadas mediante un detector de envolvente, digitalizadas y desplegadas. Un generador de rampa o barrido crea el movimiento horizontal a través de la pantalla de izquierda a derecha. La rampa también sintoniza el LO de modo que su frecuencia cambie en proporción a la rampa de voltaje.

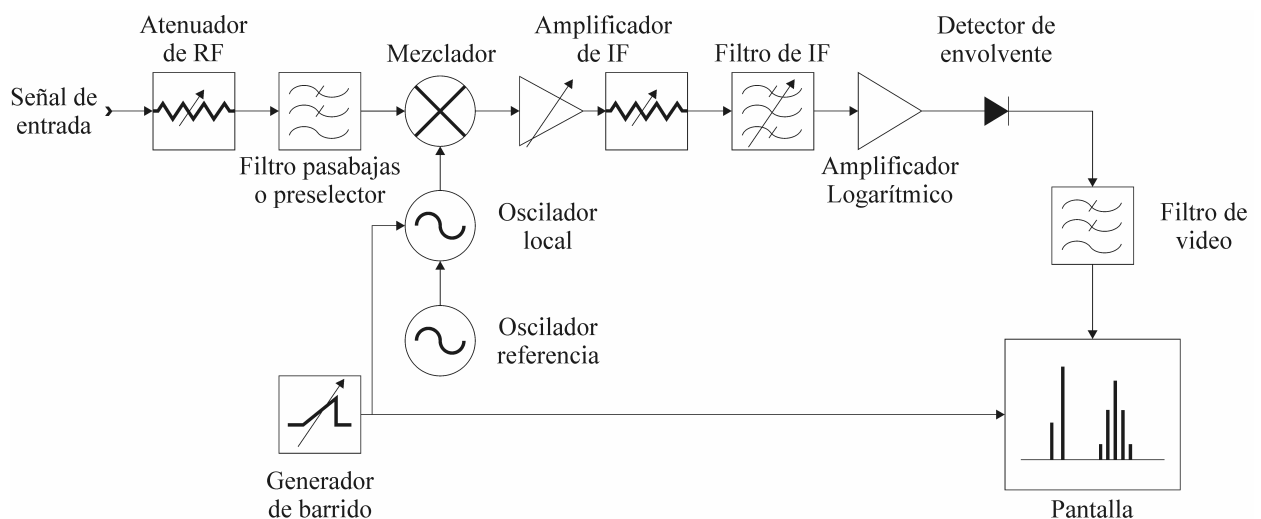


Figura 2.4. Diagrama a bloques de un analizador de espectros superheterodino clásico.

La Tabla 2.3 muestra a manera de resumen el propósito de cada uno de los bloques de un analizador de espectros superheterodino.

Tabla 2.3. Bloques de un analizador de espectro superheterodino.

Bloque	Descripción
Atenuador de RF	Asegura que la señal que ingresa al mezclador tenga un nivel óptimo, con el objeto de prevenir sobrecargas, compresión de ganancia y distorsión. Debido a que la atenuación es un sistema de protección para el instrumento, su valor es usualmente asignado de manera automática en base al nivel de referencia (valor máximo que es posible de visualizar en pantalla).
Filtro pasabajas o preselector	Rechaza las señales de alta frecuencia no usadas por el mezclador. Este impide la mezcla de señales fuera de ancho de banda establecido, que generen armónicos de IF. Analizadores de espectros para microondas rempazan este filtro por un preselector, el cual es un filtro sintonizable que rechaza todas las frecuencias excepto aquellas que se desean observar.
Mezclador	Genera la mezcla de la señal de entrada (f_{ent}) y la del oscilador local (f_{LO}), su salida incluye las dos señales originales, la suma y diferencia de las mismas, así como todos los armónicos generados (f_{ent} , f_{LO} , $f_{LO}+f_{ent}$, $f_{LO}-f_{ent}$, $f_{ent}-f_{LO}$, etc.).
Oscilador local	Sintoniza la señal que será ingresada al mezclado. Su valor es determinado por la señal de entrada y la frecuencia intermedia (f_{IF}) usada por el analizador, por lo que sus atributos variarían de acuerdo al elemento en frecuencia de la señal de entrada que se necesite procesar. Debido a que solo los productos de mezclado que caigan dentro de la IF son utilizados ($f_{LO}-f_{ent}$), el valor del LO estará fijado por la ecuación: $f_{LO} = f_{ent} + f_{IF}$. Cuando el valor del LO es igual al de la IF, la señal por si misma es procesada por el sistema y aparece en pantalla, como si esta fuera una señal de entrada de 0 Hz.
Oscilador referencia	Establece la referencia del oscilador local, su valor es establecido como fijo.
Generador de barrido	Controla la posición horizontal de la traza en pantalla y la frecuencia del LO.
Amplificador de IF	Ajusta la posición vertical de las señales en la pantalla sin afectar su nivel a la entrada del mezclador. Si la razón del amplificador de IF es modificada, el valor del nivel de referencia es rectificado con el fin de mantener el valor correcto de las señales en pantalla.
Filtro de IF	Selecciona productos mezclados, rechazando todas las otras señales no deseadas. La IF de este filtro pasabandas es un argumento fijo cuyo valor se establece por arriba del valor mas grande de frecuencia que el analizador puede sintonizar. La resolución del ancho de banda (RBW, <i>Resolution Bandwidth</i>) atribuida a este filtro, representa la capacidad del instrumento para separar dos entradas sinusoidales en dos distintas respuestas.
Amplificador logarítmico	Comprime la señal a una escala logarítmica, para fines esencialmente de visualización.
Detector de envolvente	Convierte la señal de IF a una señal de video ² . De manera simple un detector de envolvente consiste de un diodo, una resistencia de carga y un filtro pasabajas. La respuesta del detector sigue los cambios en la envolvente de la señal de IF, pero no el valor instantáneo de la onda seno de IF.
Filtro de video	Determina el ancho de banda de la señal de video (VBW, <i>Video Bandwidth</i>) que será a la poste digitalizada. Su naturaleza obedece a la de un filtro pasabajas. La frecuencia de corte del filtro de video puede ser reducida hasta un punto donde esta pueda ser más pequeña que el ancho de banda del filtro de IF. Cuando esto ocurre el sistema de video puede no ser capaz de seguir la variación de la envolvente de la señal, resultando la señal desplegada en pantalla un promedio de esta.
Pantalla	Muestra un espectro como producto del proceso. Cada punto en la traza representa una componente en frecuencia determinada por la resolución del generador de barrido, la RBW y VBW, por lo que cada punto es interpretado como un valor en un pequeño rango de frecuencias, derivado generalmente de múltiples mediciones.

² Señal cuyo rango de frecuencia se extiende de 0 (DC) a un valor superior de frecuencia. Su nombre tiene origen en su uso para el manejo de proyecciones en pantallas de CRT para algunos analizadores.

3. SCPI

Durante muchos años, los fabricantes de instrumentos trabajaron para estandarizar la interfaz mecánica y eléctrica entre instrumentos y computadoras. El surgimiento de la norma IEEE 488.1 en 1975, vino a dar en parte fin a esta tarea, e inició la búsqueda de la estandarización de los mensajes enviados sobre esta y otras interfaces, problema al cual, los fabricantes de instrumentos dieron como respuesta una gran variedad de ordenes para instrumentos, que forzaban a los usuarios a aprender un vocabulario para cada instrumento, así como a acrecentar el tiempo en el desarrollo, aprendizaje, mantenimiento y actualización de sistema ATE.

Aunque en 1987 el estándar IEEE 488.2 definió los roles de instrumentos y controladores en un sistema de medición, un esquema estructurado de comunicación y la manera de como enviar ordenes a un instrumento y como responder a ellas por parte del controlador. En general, no especificaba las ordenes o características que deberían ser implementadas en un instrumento, ocasionando que dos instrumentos similares, bajo este estándar manejaran conjuntos de ordenes diferentes.

Esfuerzos aislados fueron realizados con el objeto de alcanzar un modelo único para el diseño de instrumentos programables. En primera instancia por Hewlett Packard, ahora Agilent Technologies, quien desarrolló un conjunto de ordenes comunes basadas en los parámetros de las señales analizadas y un diagrama a bloques universal, para sus instrumentos. Este estándar interno fue el primero de su tipo en darse a conocer bajo el nombre de HP-SL y TMS posteriormente. Por otra parte, Tektronix examinó la posibilidad de almacenar y transportar mediciones de datos, así como compartir éstos entre diferentes instrumentos, concluyendo con la presentación de un formato de intercambio de datos (DIF, *Data Interchange Format*), combinando los datos generados de señal con información contextual acerca de cómo es hecha la medición [URL11].

Finalmente, fue la conferencia tecnológica para miembros del IEEE (AUTOTESTCON, *Automated Test Conference*) de 1989, la que concentro a este grupo de compañías interesadas en el desarrollo del anhelado estándar. Así fue como ocho meses después, y tras la colaboración de nueve de estas compañías, surgió la norma SCPI (*Standard Commands for Programmable Instruments*) y la consecuente creación de un consorcio bajo el mismo nombre, encargado de la regulación de este estándar.

Si bien el estudio de SCPI gira alrededor del conocimiento de los procedimientos empleados en la construcción de ordenes y las relaciones establecidas entre ellas, existen algunos otros argumentos que al margen de lo que se esperara representan la medula de la norma, tal como lo son el modelado y la clasificación de instrumentos. Aunado a esto, la norma SCPI destaca otros puntos de vital importancia en el desarrollo de cualquier aplicación relacionada con el control de instrumentos, tal es el caso del manejo de la estructura de registros de estado y el formato de intercambio de datos.

3.1. Objetivo de SCPI

La meta de SCPI es la reducción de tiempo en el desarrollo de aplicaciones ATE, proporcionando un entorno consistente de programación para el control de instrumentos y manejo de sus datos. Este entorno consistente de programación es alcanzado mediante el uso de mensajes, respuestas y formatos de datos definidos para todos los instrumentos bajo SCPI [34].

Un entorno de programación consistente usa las mismas ordenes y parámetros para controlar instrumentos que presentan una funcionalidad similar. Estas ordenes y parámetros son enviados desde un controlador al instrumento usando interfaces como IEEE 488.1, bus VXI, RS-232C, etc. Los instrumentos controlables mediante SCPI presentan gran flexibilidad en el uso de ordenes y parámetros. La respuesta del instrumento al controlador puede ser información de datos o estado. El formato de respuesta de los instrumentos a un cuestionamiento particular se encuentra bien definido y reduce el esfuerzo de programación para comprender dicha información.

La consistencia en la programación mediante SCPI es manejada de dos maneras: vertical y horizontal. La consistencia vertical define mensajes para una clase de instrumento, de esta forma se podrá usar una misma orden para leer voltaje de DC en diferentes multimetros. La consistencia horizontal hace referencia al uso de una misma orden para controlar funciones similares a través de diferentes clases de instrumentos, así una orden de disparo podrá ser usado en osciloscopios, generadores de funciones u otro instrumento que realice esta función.

Una clave para alcanzar esta consistencia es la reducción de caminos para el control de funciones similares de un instrumento. La filosofía de SCPI manifiesta el uso de una misma orden SCPI para todas las funciones afines en un instrumento. SCPI emplea nombres estándares en la industria y términos que apoyan fabricantes y usuarios.

SCPI proporciona diferentes niveles para el control de instrumentos. Ordenes simples de medida proveen un control fácil y rápido. Así, mientras más detalladas sean las ordenes usadas se obtendrá un control mas tradicional del instrumento.

SCPI esta diseñado para ser expandido con nuevas definiciones de ordenes en el futuro sin causar problemas de programación. Un objetivo implícito en todos los nuevos instrumentos controlables mediante SCPI es mantener la compatibilidad de programas con instrumentos bajo SCPI ya existentes, sin embargo este hecho puede resultar erróneo. Así, la compatibilidad de un programa de prueba es asegurada hacia versiones de instrumentos SCPI posteriores, pero no así para anteriores.

Con el objeto de promover su uso y aceptación en diferentes ámbitos, SCPI se encuentra disponible públicamente para su implementación por cualquiera, sea o no miembro del consorcio SCPI. El consorcio no da validez a documentos de investigación o documentación provisional, siendo únicamente estándares aprobados los puestos a disposición publica.

3.1.1. Intercambiabilidad de instrumentos

Uno de los principales fines perseguidos por SCPI es la capacidad de intercambio de instrumentos controlables bajo la norma, dentro de un sistema ATE. Sin embargo, SCPI no es un estándar que provee completamente intercambiabilidad entre instrumentos. SCPI ayuda a alcanzar esta intercambiabilidad mediante la definición de ordenes y respuestas de los instrumentos, pero no define funcionalidad, precisión, resolución y conexiones entre dispositivos, argumentos necesarios para alcanzar una verdadera intercambiabilidad de instrumentos sin afectar los sistemas ATE en hardware o software.

3.1.2. Ciclo de vida de SCPI

SCPI es un estándar “vivo”. El surgimiento de nuevas tecnologías e instrumentos hace necesaria la adición constante de nuevas pautas y ordenes al estándar. La inclusión de nuevas ordenes puede ser propuesta tanto por miembros del consorcio SCPI, como por otras partes interesadas. Las propuestas aceptadas por el consorcio son publicadas y distribuidas a las compañías miembros para su uso inmediato. Las propuestas aprobadas son revisadas anualmente. Después de cada revisión anual, se publica una nueva versión del estándar SCPI. Todos los instrumentos bajo la norma SCPI pueden

ser cuestionados mediante una orden específica, para determinar la versión (año) de la norma SCPI usada por estos.

En general, una orden propuesta como una adición a la norma SCPI deberá como primera condición ser construida de acuerdo a las reglas de estilo y sintaxis establecidas por la norma. Estas reglas establecerán el carácter de la orden, la categoría en la que será puesta, y si existe o no una orden que permita el control de la misma funcionalidad.

SCPI esta diseñado para crecer bajo un ambiente de compatibilidad. Para un programa de control de algún instrumento, esto significa que las adiciones hechas a SCPI no modificarán el significado de las ordenes existentes, por lo que este no necesitara ser rectificado, excepto por la inclusión de nuevas funcionalidades. Mientras que para un instrumento, esto significa que las extensiones hechas a la norma no harán obsoletos a los instrumentos ya existentes.

Es posible que una orden tenga que ser alterada o borrada para permitir que sean implementadas nuevas funcionalidades. Las propuestas de cambios que rompan el argumento de compatibilidad únicamente serán aceptadas si existe evidencia irrefutable de que los beneficios alcanzados por estas modificaciones son superiores.

Desde antes que un fabricante elija construir un instrumento controlable mediante SCPI con capacidades que no son cubiertas por la versión actual del estándar, este presumiblemente presentara al consorcio de manera anticipada, el conjunto de pautas y nuevas ordenes que pretenda, formen parte de la norma SCPI, en acuerdo con el proceso de acreditación correspondiente.

3.2. Construcción de ordenes

Todos los instrumentos definidos bajo la norma SCPI, se ajustan a las especificaciones para dispositivos citadas en el estándar IEEE 488.2, exceptuando la sección 4.1 de requerimientos, que específica que el dispositivo en cuestión deberá actuar sobre una interfaz 488.1.

En primera instancia todos los instrumentos subyugados a la norma SCPI deben implementar las ordenes comunes declaradas en el estándar IEEE 488.2. La Tabla 3.1 muestra las ordenes del estándar IEEE 488.2 de uso más general. Este conjunto de ordenes permite manipular y obtener información de algunos eventos y circunstancias que se presentan en el instrumento. Algunas de estas órdenes cuentan con una forma adicional de cuestionamiento, identificable mediante un signo de interrogación adherido.

Tabla 3.1. Ordenes comunes IEEE 488.2.

Orden	Nombre	Descripción
*CLS	Limpiar estado	Limpia las estructuras de registros de estado del dispositivo (inciso 3.4). Los elementos afectados por esta orden son: el byte de estado, el registro de eventos estándar, los registros de eventos en general, y la cola de errores y eventos. El registro de solicitud de servicio, los filtros de transición y registros de habilitación en general no son afectados.
*ESE, *ESE?	Habilitación de eventos estándar	Manipula la habilitación del registro de eventos estándar (inciso 3.5.3). Este registro monitorea errores y sincroniza condiciones del instrumento.
*ESR?	Eventos estándar	Cuestiona y limpia el registro de eventos estándar (lectura destructiva).
*IDN?	Identificador	Devuelve una cadena de información que identifica un instrumento. La cadena contiene el fabricante, modelo, número de serie y versión de firmware del instrumento, separados por una coma.
*OPC, *OPC?	Operación completada	Opera sobre el bit 0 de operación completada del registro de eventos estándar. Este es usado para sincronizar eventos de un instrumento con otros dispositivos en un bus externo.

Orden	Nombre	Descripción
*RST	Reinicio	Coloca al instrumento en un estado definido (de fabrica) donde este espera por una orden para iniciar una medición u otra acción. Las operaciones después de la ejecución de esta orden son optimizadas para la operación remota del dispositivo. Las estructuras de estado no son afectadas por esta orden.
*SER, *SRE?	Habilitación de solicitud de servicio	Manipula el registro de habilitación de solicitud de servicio (inciso 3.4).
*STB?	Byte de estado	Accede al octeto de estado sin modificar su contenido (inciso 3.5.2).
*TST?	Autopruueba	Usado en algunos instrumentos para realizar auto pruebas.
*WAI	Esperar para continuar	Ocasiona que el instrumento espere hasta que todas las ordenes pendientes sean completadas antes de la ejecución de ordenes adicionales.

Los elementos de impacto en la construcción y documentación de cualquier orden SCPI, son las cabeceras y los parámetros de la orden. Las cabeceras de orden o programa permiten definir el nombre de la orden, a partir de una o más palabras claves. Un parámetro se entiende como el argumento o valor manejado por una orden.

3.2.1. Cabeceras de programa

Las cabeceras de programa son mnemotécnicos ó palabras claves que identifican una orden. Las cabeceras de programa siguen la sintaxis descrita en la sección 7.6 del estándar IEEE 488.2. Los instrumentos bajo SCPI admiten caracteres en minúsculas o mayúsculas, sin distinción de los casos. Las cabeceras de programas son de dos tipos distintos: de orden común y de control de instrumento.

3.2.1.1. Cabeceras de ordenes comunes

Las cabeceras de programa para ordenes comunes son definidas en el estándar IEEE 488.2. La Figura 3.1 muestra la sintaxis planteada en dicho estándar, el termino mnemotécnico que se observa da correspondencia a la orden empleada, así por citar un ejemplo, una orden de operación completa será manejado como: *OPC para su ejecución y *OPC? para su cuestionamiento.

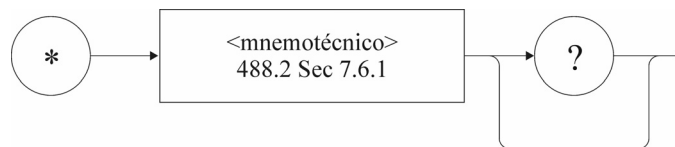


Figura 3.1. Esquema para la construcción de cabeceras de ordenes comunes.

3.2.1.2. Cabeceras de control de instrumento

Las cabeceras de control de instrumento son usadas por todas las otras ordenes existentes. La sintaxis para este tipo de cabecera se presenta en la Figura 3.2.

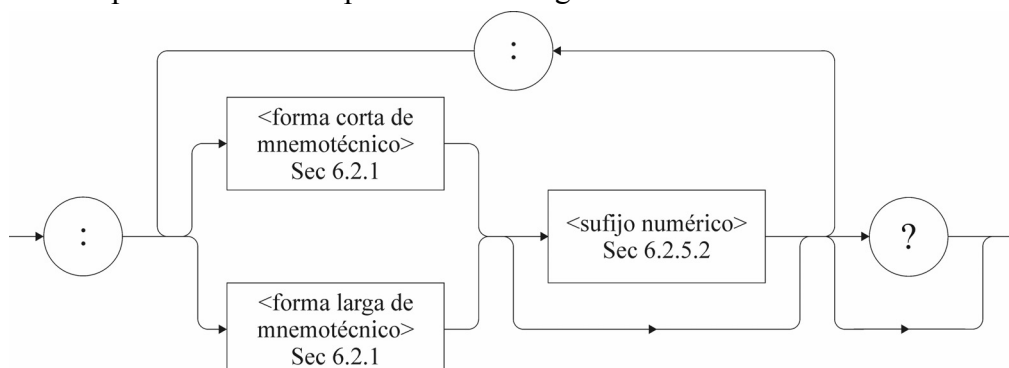


Figura 3.2. Esquema para la construcción de cabeceras de control de instrumento.

3.2.1.2.1. Generación de mnemotécnicos

Cada mnemotécnico usado en la construcción de cabeceras de control de instrumento cuenta con una representación larga y otra corta, por lo que una cabecera de este tipo puede presentarse de estas dos maneras. El envío de una cabecera a un instrumento que no es exactamente de ninguna de las dos formas será causa de un error. El estándar IEEE 488.2 limita la longitud de una cabecera a doce caracteres, incluyendo cualquier sufijo que pueda incluirse. La cabecera en su forma larga es una simple palabra o una abreviatura de una frase. Una representación corta es una abreviatura de la forma larga de la cabecera. Con el objetivo de mantener un alto grado consistencia, SCPI define reglas para la generación de mnemotécnicos que conforman las cabeceras de control de instrumento.

Durante el presente documento, se emplea una notación especial para diferenciar la forma larga de una palabra clave de su representación corta. Así, la forma larga de una palabra será presentada en base a una combinación de letras mayúsculas y minúsculas, en donde la porción en letras mayúsculas corresponderá a su forma corta (INPut, SENSE, etc.). Esta notación es muy particular de todos aquellos documentos referentes a la norma SCPI, incluyendo la norma misma.

3.2.1.2.2. Generación de la estructura jerárquica

La definición de ordenes SCPI se encuentra ligada directamente al manejo de una estructura jerárquica también conocida como árbol. Los elementos de primer nivel en la estructura jerarquía y raíces de esta, son producto del modelado mismo de los instrumentos (inciso 3.3.1), y representan la palabra clave de cada agrupación de ordenes llamada subsistema. De esta forma las ordenes asociadas con un subsistema en particular son agrupadas bajo un nodo común en la jerarquía, análogamente a ramas conectadas a otra común. Cada nuevo nodo que origine una rama será caracterizado con un mnemotécnico, este nodo dará origen a nuevas ramas y estas a otras sucesivamente, de acuerdo a la funcionalidad que se desea cubrir con la orden en desarrollo. Así, la lectura de una orden se hará siguiendo los nodos de cada rama generada a partir de la raíz del árbol o estructura jerárquica.

El procedimiento empleado permite a los mnemotécnicos que constituyen estas ordenes ser utilizados diversas veces para diferentes propósitos, sin que estos colisionen. De este modo, dos ordenes diferentes puedan hacer uso de un mismo mnemotécnico, poseyendo una independencia total una de otra.

La Figura 3.3 ejemplifica la construcción de ordenes SCPI bajo el nodo o subsistema SENSE en su nivel superior, nótese la existencia de dos nodos con mnemotécnicos iguales, que sin embargo no guardan relación alguna en la jerarquía. Así, la sintaxis de una orden para la modificación del ancho de banda de video en un instrumento, tendrá la forma siguiente:

```
:SENSE:BANDwidth:VIDeo <valor numérico>
```

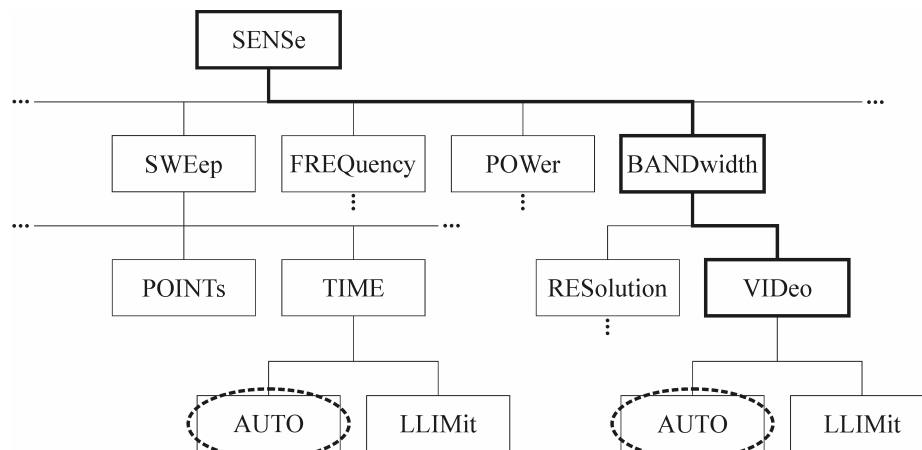


Figura 3.3. Ejemplo de la estructura jerárquica SCPI.

3.2.1.2.3. Preguntas

Todas las ordenes a menos que exista una contra indicación en su definición, cuentan con una forma adicional de cuestionamiento. Tal como lo define el estándar IEEE 488.2, una pregunta es una cabecera de orden con un signo de interrogación adherido. Cuando la forma de pregunta de una orden es recibida por un instrumento, la característica solicitada por la orden es puesta en el buffer de salida del instrumento. La respuesta a una pregunta no incluye una cabecera de programa. La ejecución de una orden en su forma de pregunta, no causa ningún cambio en los condiciones del instrumento, sin embargo una excepción a esta regla son el conjunto de ordenes agrupadas bajo el subsistema de medición MEASurement, quienes si modifican el estado del instrumento.

3.2.2. Parámetros

SCPI usa las formas de parámetros descritas en la sección 7.7 del estándar IEEE 488.2, con algunas restricciones. Un parámetro indica el número y disposición de los argumentos en una orden. Adicionalmente al uso de un tipo de parámetro particular, SCPI permite el uso de literales, o bien de una combinación parámetro-literal, como argumentos en una orden. Una literal es típicamente una palabra que enumera un argumento no descrito por algún parámetro definido por el estándar. Los tipos de parámetros manejados por SCPI son: caracteres, numéricos, booleanos, unidades y expresiones.

3.2.2.1. Carácter

Las reglas empleadas en la construcción de este tipo de parámetros son las mismas que las usadas en la especificación de cabeceras de programa, sin embargo en muchos casos los estándares industriales toman preferencia sobre estas pautas. Algunos parámetros de este género son usados para predefinir parámetros de tipo numérico.

3.2.2.2. Numérico

Los elementos numéricos son usados para representar cualidades numéricas. Estos no son empleados para funciones de selección (uno de n). Cualquier número que exceda el valor de $\pm 9.9 \text{ E } 37$ generara un error (-222, "Dato fuera de rango"). Los números usados por un instrumento son redondeados al más cercano valor que éste acepte, sin generar un error. Algunas expresiones de caracteres son definidas como formas especiales de números, estas son: DEFault, MINimum, MAXimum, UP, DOWN, NAN (*Not a Number*), INFinity, NINF (*Negative INFinity*).

Las expresiones numéricas y etiquetas son consideradas también dentro de este tipo de parámetros.

3.2.2.3. Boleano

Un valor de 0 o 1 sin ningún atributo de unidad es asignado a este parámetro. Dentro de la documentación SCPI este tipo de parámetro son referenciados mediante la expresión ON/OFF. Las ordenes de pregunta cuya respuesta sea un parámetro booleano devolverán como respuesta un 1 o 0, nunca una expresión ON/OFF.

3.2.2.4. Unidades de medida y sufijos

Las unidades y sufijos usados por la norma SCPI obedecen a las especificaciones establecidas en la sección correspondiente del estándar IEEE 488.2. La industria, quien juega un papel importante en la determinación de algunos conceptos de la norma SCPI, es la causante de la expansión de las unidades de potencia y amplitud definidas por el estándar IEEE 488.2 (Watts, Volts, dBm y dBV), mediante la adición de unidades como uV, dBuV, dBuW y dBmV.

3.2.2.5. Expresiones

Las expresiones manejadas por la norma SCPI se encuentran basadas en las definiciones establecidas por el estándar IEEE 488.2 para este tipo de parámetro. La Tabla 3.2 muestra los diferentes tipos de expresiones definidas por SCPI. El uso de algunas otras expresiones enmarcadas por el estándar IEEE 488.2, es permitido, siempre y cuando estas puedan ser documentadas adecuadamente.

El uso de expresiones es opcional en SCPI. Únicamente los tipos de expresiones que sean requeridas por una orden particular pueden ser reconocidas como parámetros de esta.

Tabla 3.2. Expresiones para ordenes SCPI.

Tipo de expresión	Descripción
Numéricas	Colección de términos usados en la evaluación de trazas, números, arreglos u otro tipo de elementos. Los términos usados incluyen símbolos como +, -, *, / y operadores lógicos como AND y OR. Los elementos considerados en una expresión de este tipo se encontraran de acorde al tipo de argumento evaluado. De esta forma una expresión relacionada con el manejo de trazas, será evaluada conforme a las reglas aritméticas establecidas para este argumento.
Listas de canales	Usadas para especificar puertos eléctricos de un instrumento, estas son típicamente empleadas en el enrutamiento de señales en instrumentos con múltiples canales o rutas de salida. Las listas de canales son comunes en ordenes de medición y configuración.
Listas numéricas	Formato de expresión utilizado para enunciar de manera compacta números y rangos mediante un simple parámetro.
Formato de intercambio de datos	Permite a paquetes de software e instrumentos compartir datos sobre formas de señal, entre otros. Su diseño flexible, les permite adecuarse a un ancho rango de estructuras de datos y formatos. La sintaxis del formato de intercambio de datos es compatible con la especificada en el estándar IEEE 488.2. Este formato es manipulado como un conjunto de bloques estructurados. Cada bloque contempla un aspecto diferente de la descripción del dato, mediante palabras claves y otros bloques subordinados.
Especificación de instrumento	Empleada en la descripción de funcionalidades presentada por un dispositivo, estas permiten representar el que hace y como lo hace de un instrumento. Una expresión de este tipo se compone de una serie palabras claves que detallan las funciones que el dispositivo presenta, enlazadas mediante operadores lógicos (AND u OR). Palabras claves adicionales son también usadas para la descripción de características de alguna funcionalidad en específico.

3.3. Instrumentos

Cualquier instrumento controlable mediante SCPI, dentro del ámbito normativo es citado como instrumento SCPI, de forma tal que dicha denominación sustente el hecho de pensar que el dispositivo en cuestión ha sido diseñado bajo los cánones que marca SCPI.

3.3.1. Modelo de un instrumento

El modelado de instrumentos en SCPI, es usado como un medio para conseguir la compatibilidad entre instrumentos. SCPI concierne por si mismo tres tipos de compatibilidad. La primera llamada compatibilidad vertical se da cuando dos instrumentos del mismo tipo tienen controles idénticos. Por ejemplo, dos osciloscopios de diferente fabricante, donde ambos tienen los mismos controles para sus tiempos base, disparos, etc. son compatibles verticalmente [35].

La segunda forma de compatibilidad denominada horizontal, hace referencia a dos instrumentos capaces de hacer una misma medición, independiente de la técnica usada. De esta manera ambos instrumentos pueden usar la misma orden para hacer esta medición. Por ejemplo, un osciloscopio y un contador pueden llevar a cabo una medición del tiempo de subida para un pulso particular, mediante dos técnicas diferentes, mostrando una compatibilidad horizontal.

El tercer tipo de compatibilidad evocada como funcional se establece cuando dos instrumentos realizan una similar función mediante una misma orden. Por ejemplo, un analizador de espectros y un generador de RF que efectúan con diferente propósito, operaciones de barridos en frecuencia mediante ordenes idénticas, presentaran una compatibilidad funcional en esta área.

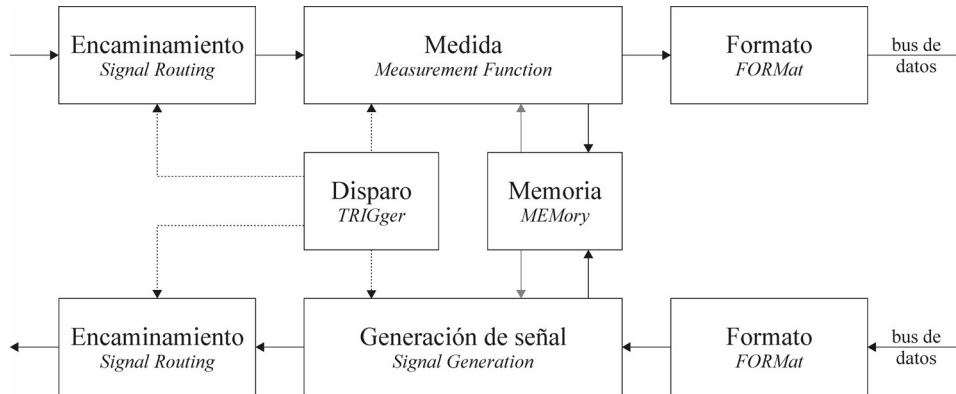


Figura 3.4. Modelo básico de un instrumento programable.

La Figura 3.4 representa la manera en como es visto funcionalmente un instrumento y categorizado por SCPI. El flujo de datos es representado por líneas sólidas, mientras que el flujo de control es descrito por líneas punteadas. El propósito de esta categorización es proporcionar organización y consistencia entre las diversas ordenes disponibles por SCPI. El modelo define como los elementos del lenguaje deben ser agrupados y asignados en la estructura jerarquía SCPI, para la construcción de las ordenes. Las áreas de mayor impacto en el aspecto funcional del instrumento son consolidadas como bloques; cada uno de estos bloques se encuentra en un primer nivel jerárquico de la estructura SCPI.

El modelo describe el flujo de la medición y aplicación de las señales de datos a través del instrumento así como la administración del flujo de datos asociados con las ordenes, cuestionamientos, calibraciones, accesos a memoria y otras funciones relacionadas que no son incluidas en este modelo. El modelo no define como un instrumento maneja o interpreta el formato de los datos. Los instrumentos únicamente implementaran los bloques que requieran. Por ejemplo, un voltímetro simple que no usa direccionamiento a memoria incluirá solo bloques de funciones de medición, disparo y formato, como se observa en la Figura 3.5(a). De manera similar, una fuente de alimentación simple incluirá solo un bloque generador de señal y uno de formato, como se muestra en la Figura 3.5(b).

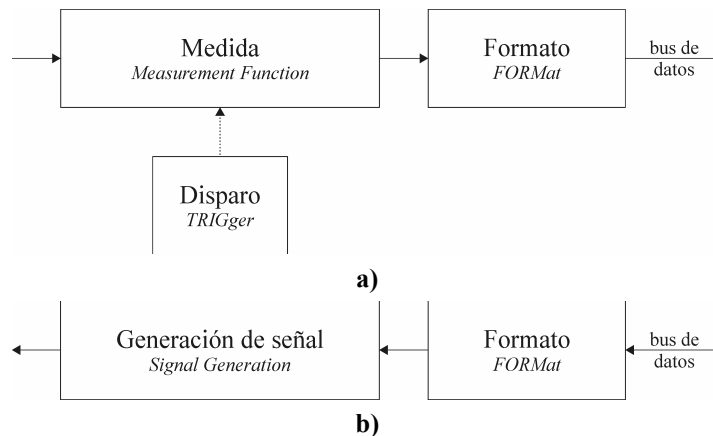


Figura 3.5. Ejemplos de modelos SCPI simplificados. (a) instrumento sensor. (b) instrumento fuente.

Un instrumento que cuenta con un bloque particular no siempre requiere implementar todas las ordenes asociadas con el bloque. Esto ocurre cuando, el bloque existe en un modo de configuración fija que no representa conflicto con alguna otra orden. Por ejemplo, un instrumento puede tener

un nivel externo de disparo TTL fijo, sin permitir un ajustable de este parámetro, en este caso una parte del conjunto de ordenes que forman el subsistema que hace referencia al bloque de disparo no tendrá la necesidad de ser implementada, sin embargo el control de disparo existe, y será implementado tal como lo define el subsistema en cuestión.

Los principales elementos o bloques denotados por el modelo básico de un instrumento SCPI citado en la Figura 3.4 son: encaminamiento de señal, medida, generación de señal, memoria, formato y disparo.

3.3.1.1. Encaminamiento de señal

El propósito del bloque de encaminamiento de señal es el control del recorrido de la señal (una señal física) desde el exterior al bloque de medida de señales o bien del interior del instrumento en el bloque de generación de señal al exterior. Las ordenes que controlar este bloque son descritas en la jerarquía SCPI bajo el subsistema ROUTE. La implementación de este subsistema es opcional para todos aquellos instrumentos que tienen conexiones fijas al bloque de medida o de generación de señal.

3.3.1.2. Medida

El bloque de medida convierte una señal física en un formato que admita ser procesado. Este bloque puede realizar tareas adicionales de acondicionamiento de señal y cálculos posteriores a la conversión.

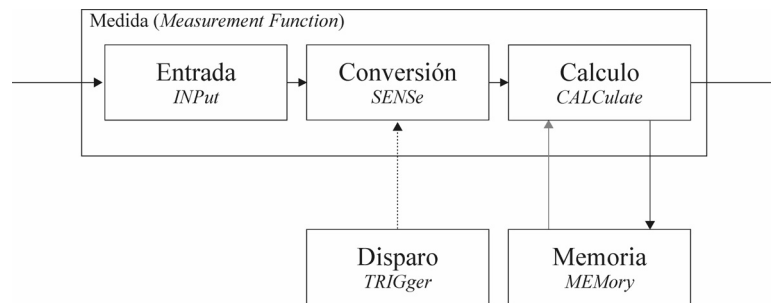


Figura 3.6. Expansión del bloque de medida.

Este bloque como se muestra en la Figura 3.6, se encuentra dividido en tres unidades:

- *Entrada*: acondiciona la señal de entrada antes de ser convertida a un formato digital, mediante funciones como: filtrado, escalado, conversión de frecuencia (mezcladores o funciones de preescala), y atenuación. El bloque de entrada aparece en la estructura jerárquica de ordenes SCPI bajo la denominación de subsistema INPut.
- *Conversión*: transforma la señal a un formato digital que se pueda manipular fácilmente. Las ordenes asociadas con este bloque controlan las diversas características del proceso de conversión. Ejemplo de ellos son el rango de medida, resolución, etc. Este bloque no incluye ninguna manipulación matemática de los datos después de haber sido convertidos. Las ordenes manejadas por el bloque de conversión aparecen en la estructura jerárquica SCPI bajo el nombre de subsistema SENSE. Dada su naturaleza, este bloque es requerido por todos los instrumentos sensores.
- *Cálculo*: convierte los datos adquiridos a un formato más útil para la aplicación que los demande. Cálculos típicos incluyen conversión de unidades, frecuencia, etc. Las ordenes referentes a este bloque son descritos en el subsistema CALCulate.

Las subdivisiones del bloque de medida no son detalladas en el modelo básico de un instrumento programable mostrado en la Figura 3.4 debido a que estas se encuentran en un nivel de división inferior, que puede dar lugar a considerar una incompatibilidad horizontal entre instrumentos.

De esta manera, la medición de la frecuencia de una señal podrá ser efectuada mediante el bloque de conversión en un contador, mientras es realizada en el bloque de cálculo en un analizador de espectros, mostrando siempre una compatibilidad horizontal entre dispositivos.

3.3.1.3. Generación de señal

El bloque de generación de señales es responsable de la conversión de datos internos en señales que se pueden utilizar en el mundo exterior. Este bloque puede realizar tareas adicionales de pre-conversión y acondicionamiento de señal.

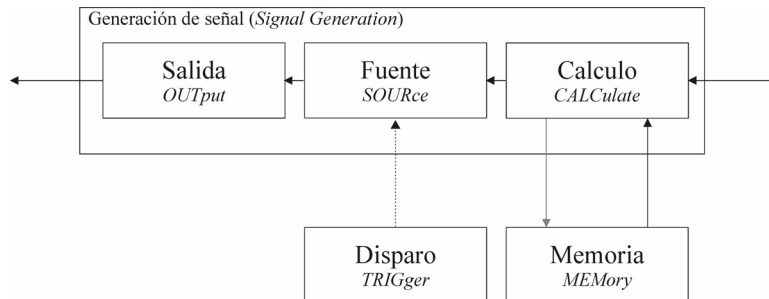


Figura 3.7. Expansión del bloque de generación de señal.

Este bloque como se muestra en la Figura 3.7, se encuentra dividido en tres unidades:

- *Salida*: acondiciona la señal que ha sido generada, antes de extraerla al exterior. El bloque incluye procedimientos de filtrado, escalado, conversiones de frecuencia y atenuación. El bloque de salida aparece en la jerarquía SCPI bajo el subsistema OUTPUT. Este bloque se encuentra presente en todos los instrumentos generadores, sin la necesidad de su uso total.
- *Fuente*: genera una señal de acuerdo a ciertas características, como lo pueden ser la modulación en amplitud, la corriente a proporcionar, la frecuencia, etc. Las ordenes asociadas a este bloque describen las particularidades de la señal generada. El bloque de fuente es requerido por todos los instrumentos que hacen uso del bloque de generación de señal. Las ordenes relacionadas con este bloque son descritos bajo el subsistema SOURCE.
- *Cálculo*: realiza operaciones de procesamiento posteriores a la adquisición de datos. Transforma los datos solicitados a manera de representar las anomalías en la generación de una señal que sean pertinentes, tales como correcciones para efectos externos, conversión de unidades y cambios de dominio. Las ordenes asociadas con este bloque son contemplados en el subsistema CALCulate.

Las subdivisiones de generación de señales no son puntualizados en el modelo básico de un instrumento programable de la Figura 3.4 debido a que pueden dan lugar a considerar una incompatibilidad horizontal entre instrumentos, por encontrarse en nivel de división inferior.

3.3.1.4. Memoria

El bloque de memoria retiene la información interna necesaria para el instrumento. La memoria puede ser implícita o inaccesible al usuario (datos internos de calibración), así como fija (medición de datos actual) o dinámica (manipulable por el usuario).

El argumento de memoria aparece en SCPI bajo los subsistemas MEMORY y MMEMORY. La manipulación de la memoria es realizada a través de ordenes de los subsistemas SENSE, SOURCE, FORMAT, DISPLAY entre otros. Aunque todo instrumento controlable mediante SCPI cuenta de manera implícita con un bloque de memoria, no todos proporcionan al usuario un control explícito de éste, en tales casos, la definición de este bloque no es necesaria.

3.3.1.5. Formato

El bloque de formato adapta la información que se genera en el dispositivo o es recopilada del exterior para que pueda ser transmitida por la interfaz externa o bus utilizado. Es en el subsistema FORMat dentro de la jerarquía SCPI donde las funciones de formato pertenecientes a este bloque son presentadas. Todos los instrumentos bajo SCPI hacen uso de los formatos descritos en el estándar IEEE 488.2.

3.3.1.6. Disparo

El bloque de disparo sincroniza los diversos bloques y acciones del instrumento con sucesos externos o con otros instrumentos. El bloque de disparo aparece en la jerarquía SCPI bajo los subsistemas TRIGger, ARM, INITiate y ABORT. Dos modelos existen para la concepción del bloque de disparo, uno complejo y otro simple, ambos pueden ser reducidos a un nivel que corresponda con las capacidades de disparo que el instrumento demande.

3.3.2. Clases de instrumentos

La clasificación de instrumentos planteada por SCPI esta basada tanto en la manipulación del modelo básico de instrumentos establecido por la norma, como en la funcionalidad presentada por el instrumento en cuestión. Los objetivos que esta tipificación persigue son [37]:

- Reducir el tiempo en el desarrollo de productos ATE basados en SCPI, guiando su diseño a través de un familiar punto de vista, como lo es la clase del instrumento.
- Alcanzar un alto grado de consistencia en implementaciones con la misma clase de instrumentos.

De manera sencilla la funcionalidad de un instrumento representa el que hace y como lo hace, así cada clase de instrumento expresara una funcionalidad en particular. A esta funcionalidad se le es asignado un mnemotécnico de acuerdo a las reglas estipuladas por SCPI. Estas palabras claves también definen funcionalidades adicionales.

Un instrumento es funcionalmente especificado usando uno o más de estos mnemotécnicos en una expresión del tipo <especificación de instrumento> (inciso 3.2.2.5). Algunos instrumentos implementan la orden de pregunta SYSTem:CAPability? cuya respuesta es una expresión de este genero, como medio de información sobre las capacidades del instrumento, de acuerdo a la clasificación hecha por SCPI. El fabricante del instrumento puede o no usar esta expresión para describir funcionalmente el dispositivo, tomando en cuenta las clases de instrumento descritas por la norma.

La Tabla 3.3 resume las clases de instrumentos manejadas por la norma SCPI versión 1999.0. Algunas de estos tipos de instrumentos solo hacen uso de determinados bloques del modelo SCPI, o bien mantienen una configuración fija para estos. Esta clasificación cubre la mayor parte de instrumentos contemplados por SCPI.

Tabla 3.3. Clases de instrumentos SCPI.

Clase de instrumento	Descripción	Bloques utilizados del modelo SCPI	Subsistemas utilizados
Dinamómetro-chasis	Instrumento generador básico. Provee una fuerza a un vehiculo con el fin de simular condiciones de uso. Una expansión de esta clase de instrumento permite la medición de parámetros de fuerza, velocidad y distancia. Su uso se delimita generalmente a la industria automovilística.	Encaminamiento de señal (fijo), Medida (presente), Generación de señal (presente), Memoria (presente), Formato (fijo), Disparo (fijo).	SOURce, SENSe, MEMory, CONTrol, CALibration, STATus.

Clase de instrumento	Descripción	Bloques utilizados del modelo SCPI	Subsistemas utilizados
Medidor digital	Instrumento de medición básico. La medición es típicamente un simple valor de un punto en el tiempo. Los instrumentos más comunes pertenecientes a esta clase son los voltímetros, amperímetros y ohmímetros. Un instrumento real suele caer una o más veces dentro de esta categoría.	Encaminamiento de señal (exterior-bloque de medida, fijo-opcional), Medida (presente), Generación de señal (no presente), Memoria (opcional), Formato (salida, fijo-opcional), Disparo (presente).	MEASurement FORMat, SENSe, TRIGger, STATus.
Digitalizador	Instrumento diseñado principalmente para mediciones de voltaje de formas de onda en el tiempo. Las entradas de un digitalizador, llamadas comúnmente canales son numeradas por números o letras, por lo que el uso de sufijos en las ordenes utilizadas por estos instrumentos es habitual.	Encaminamiento de señal (exterior-bloque de medida, fijo-opcional), Medida (presente), Generación de señal (no presente), Memoria (opcional), Formato (salida presente), Disparo (presente).	INPut, FORMat, SENSe(opc.), TRIGger, STATus.
Banco de emisiones	Dispositivo de medición complejo usado para determinar la concentración de una muestra de gas introducida en un banco. Los bancos de emisiones no solo se muestran como analizadores de gas, pues también incluyen otros instrumentos como divisores de gas y medidores de temperatura, presión y flujo. Su uso es habitual en la industria de procesos, como la petrolera.	Encaminamiento de señal (presente), Medida (presente), Generación de señal (presente), Memoria (presente), Formato (opcional), Disparo (presente).	SENSe, MEMory, TRIGger, CALibration, CONTrol, SYSTem, INSTRument, ROUTE, STATus.
Probador de celdas de emisión	Conglomerado de computadoras e instrumentos usados para desarrollar pruebas de emisión. Los instrumentos incluidos pueden ser bancos de emisiones, dinamómetros, entre otros. Su uso es común en la industria de las telecomunicaciones.	La naturaleza de esta clase de instrumentos impide definir un modelo a bloques específico. La existencia de diversos instrumentos de igual forma no hace posible concretar los subsistemas de ordenes utilizados.	SYSTem, TRIGger STATus.
Fuente de poder	Instrumento proveedor de un nivel de potencia significativa, que permita energizar un circuito eléctrico. Una fuente de poder puede operar en dos modos como fuente de voltaje o de corriente, modo del cual dependerá la utilización de un grupo de ordenes en particular.	Encaminamiento de señal (interior-bloque de generación de señal, fijo), Medida (no presente), Generación de señal (presente), Memoria (opcional), Formato (entrada, fijo-opcional), Disparo (opcional).	SENSe(opc.), OUTPut, SOURce, TRIGger, STATus.
Generador de RF y/o microondas	Instrumento típicamente generador de una señal sinusoidal a una frecuencia y nivel constante. La frecuencia de operación de estos dispositivos se halla usualmente entre 10 MHz y varios GHz. Su impedancia de salida es regularmente de 50 ohms. Funciones adicionales como modulación y barrido en frecuencia son contempladas también para este tipo de instrumentos.	Encaminamiento de señal (interior-bloque de generación de señal, fijo-opcional), Medida (no presente), Generación de señal (presente), Memoria (no presenten), Formato (entrada, fijo-opcional), Disparo (opcional).	SOURce, OUTPut, UNIT, STATus.
Conmutador de señal	Instrumento de enrutamiento de señal. En su forma más básica, este puede habilitar o deshabilitar conexiones de señales, usualmente referidas como canales. Una conexión se realiza cerrando un canal. El ejemplo más usual de este tipo de instrumentos son los multiplexor de n canales	Encaminamiento de señal (presente), Medida (no presente), Generación de señal (no presente), Memoria (no presente), Formato (no presente), Disparo (opcional).	ROUTE, TRIGger, STATus.

3.4. Agrupación de ordenes

Ajenos a los medios empleados en la construcción de cualquier orden, SCPI agrupa los subconjuntos de ordenes SCPI en dos grandes categorías, que a su vez permiten proporcionar dos diferentes niveles para el control de instrumentos: subsistemas e instrucciones de medición. Los subsistemas de ordenes, quienes proveen un control tradicional del instrumento, hacen uso de ordenes muy específicas, con el fin de obtener un control total sobre las características del dispositivo. Mientras que las instrucciones de medición proporcionan en base a un conjunto de instrucciones de alto nivel dirigidas a la adquisición de datos, un fácil y rápido control del instrumento [35].

3.4.1. Subsistemas

Dentro de SCPI un subsistema es una estructura jerárquica de ordenes independientes, que cumplen con un propósito particular de acuerdo al modelo general para instrumentos SCPI. Para cada subsistema es desarrollado un árbol o estructura jerárquica, que cubre las exigencias que se demandan de acuerdo a las reglas que la norma establece (inciso 3.2.1.2). La Tabla 3.4 describe de manera resumida, los subsistemas contemplados por la norma SCPI en su versión 1999.0.

Tabla 3.4. Subsistemas de ordenes SCPI.

Subsistema	Descripción
CALCulate	Establece operaciones para el procesamiento de datos, posteriores a la adquisición de ellos. Este subsistema opera con los datos adquiridos por las funciones SENSE.
CALibration	Provee el conjunto de funciones relativas al sistema de calibración del instrumento. Las ordenes de este subsistema tienden a ser muy específicas de acuerdo a la clase de instrumento del que se trate.
CONTRol	Aporta vías para control del encendido y apagado de elementos electromecánicos del instrumento implicados en algún procedimiento.
DIAGnostic	Proporciona una serie de rutinas de servicio y diagnostico usadas en el mantenimiento y reparación del instrumento. Cada fabricante es libre de adherir ordenes de servicio y diagnostico de acuerdo a sus necesidades, bajo este subsistema.
DISPlay	Controla la selección y presentación de texturas, gráficos e información de trazas. Este subsistema no modifica la forma en como los datos son enviados al controlador.
FORMat	Gestiona el manejo del formato de datos para la transferencia de información numérica y de arreglos. Este formato de datos es usado tanto para envío de ordenes de asignación como para la respuesta de datos.
HCOPY	Manipula parámetros de configuración para la impresión a un dispositivo externo.
INPut	Caracteriza los sensores de los puertos de entrada del instrumento.
INSTrument	Suministra mecanismos para identificar y seleccionar instrumentos lógicos mediante un nombre o número. Un instrumento puede soportar múltiples instrumentos lógicos. De esta manera una fuente de potencia dual es considerada como dos instrumentos lógicos, por lo que estos podrán ser manipulados de forma independiente mediante el uso de ordenes de este subsistema.
MEMory	Opera y administra la memoria del instrumento que no es usada para almacenamiento masivo. La memoria de un instrumento puede ser usada para guardar diversos tipos de datos como ASCII, binarios, estados de instrumento, trazas y macros. Los tipos de datos manejados dependen del dispositivo. La asignación de memoria para el manejo de datos en un instrumento puede ser dinámica o fija.
MMEMory	Aporta capacidades de almacenamiento masivo. Este almacenamiento puede ser interno ó externo al instrumento. Si el dispositivo de almacenamiento es externo entonces el instrumento deberá tener la capacidad de controlar el bus del dispositivo.
OUTPut	Caracteriza los puertos de salida del instrumento.

Subsistema	Descripción
PROGram	Administra características necesarias para generar y controlar uno o más tareas residentes en el instrumento, programadas por el usuario. Dos distintos métodos de acceso un programa son utilizados por este subsistema. El método de referencia explícita que requiere la especificación del nombre del programa, y el de selección que permite manipular un programa previamente seleccionado.
ROUTE	Aplica a instrumentos cuyo propósito es el encaminamiento de señales, como lo son los conmutadores de señales. Este subsistema puede ser opcional en algunos instrumentos. En casos en que el bloque de enrutamiento de señal no sea requerido, los puertos asociados con los bloques de entrada o salida se encontraran disponibles directamente para su uso.
SENSe	Ofrece controles que directamente afectan las especificaciones del dispositivo. Por su tamaño y complejidad este subsistema se encuentra dividido en varias secciones o subsistemas. En dispositivos que actúan básicamente como sensores este subsistema puede ser omitido. El nodo SENSe no puede ser opcional si los nodos SOURce o ROUTe son también de carácter opcional.
SOURce	Brinda cobertura sobre las particularidades de una señal generada. Este subsistema contiene un largo número divisiones que buscan cubrir las diferentes características de la señal. El nodo SOURce no puede ser opcional si los nodos SENSe o ROUTe lo son.
STATus	Controla el acceso a las estructuras de registros de estado definidas por la norma. SCPI define en adición a los establecidos por la norma IEEE 488.2, registros de cuestionamiento y operación, los cuales constan de registros de condición, evento, habilitación y filtros de transición negativos y positivos, conforme a las especificaciones del mismo estándar IEEE 488.2. Adicionales requerimientos son establecidos para instrumentos que soportan múltiples instrumentos lógicos o capacidades de disparo extendidas.
SYSTEM	Colecciona funciones que no son relacionadas con el rendimiento del instrumento. Incluye funciones relacionadas con parámetros y tareas globales tales como tiempo y seguridad.
TEST	Provee una sección de extensión para los procedimientos estándares de prueba desarrollados por los instrumentos basados en la orden *TST de la norma IEEE 488.2.
TRACe\DATA	Manipula el acceso a la memoria interna del instrumento de las entidades de trazas y/o datos. Concretamente no es un subsistema generado de un bloque del modelo para instrumentos SCPI. Particularmente, ya que solo forman parte del bloque de memoria, no obstante su extensión y utilidad permiten su consideración como un subsistema independiente. Las áreas de traza y/o datos pueden también ser usadas para almacenar otro tipo de datos, tales como arreglos constantes usados en correcciones de trazas, operaciones aritméticas realizadas entre trazas, o despliegues de formas de onda. Alternativamente estas áreas pueden ser usadas para el manejo de escalares (puntos simples).
TRIGger	Sincroniza las acciones del dispositivo con determinados eventos. El subsistema TRIGger hace uso de un modelo llamado ARM-TRIGger, para efectuar descripciones simples de sistemas de disparo de poca complejidad, de forma tal que este modelo represente una sección de las capacidades de disparo presentadas en el modelo para instrumentos SCPI, que describen sistemas de disparo muy complejos. El propósito de este subsistema es evaluar un evento antes dar paso a una secuencia de disparo, tal como habilitar un barrido, iniciar una medición o cambiar el estado del dispositivo.
UNIT	Determina mecanismos para la manipulación de las unidades, que por defecto son aplicadas en un dispositivo. Las unidades seleccionadas serán aplicadas a las ordenes, tanto en su forma de pregunta como de respuesta. Las ordenes de este subsistema pueden ser empleadas tanto a una escala global como a un nivel determinado. Cuando la orden es aplicado a un nivel o nodo en especial entonces todos los nodos que se encuentren en los niveles jerárquicos inferiores a este actuarán bajo la consigna de la unidad seleccionada. Por ejemplo, la orden [SOURce:]MODulation:UNIT:DBUv selecciona solo la unidades del modulador de una fuente y no para todo el instrumento. Las ordenes UNIT no pueden ser aplicadas a los nodos SENSe, SOURce ó ROUTe, dada que las unidades en estos subsistemas no pueden ser ambiguamente reconocidas por el instrumento.
VXI	Contiene ordenes de control, para las funciones de administración relacionadas con la operación de sistemas basados en la arquitectura VXI. Las ordenes y procedimientos empleados en este subsistema son producto de los estudios hecho por el consorcio VXI.

Todas las ordenes contempladas por SCPI, a excepción de un subconjunto obligatorio tomado de algunas secciones de los subsistemas de estado STATus y sistema SYSTem, son consideradas como opcionales. La implementación de ordenes particulares obedecerá a las capacidades presentadas por el instrumento. Cuando un dispositivo no es capaz de soportar la totalidad de valores manejados por algún parámetro de una orden, este podrá implementar solo un conjunto de estos valores.

3.4.2. Instrucciones de medición

El grupo de instrucciones de medición, derivado del subsistema de medida MEASurement, tiene como propósito la adquisición de datos mediante el uso de un conjunto de instrucciones de alto nivel. El referente de estas instrucciones son las características de la señal medida, lo que hace a estas, independientes del modelo para instrumentos planteado por SCPI.

El conjunto de instrucciones de medición se encuentran estructuradas de forma tal, que permitan al usuario un excelente control de los procesos de medición en compensación al sacrificio de la intercambiabilidad. El conjunto define acciones de configuración (CONFigure), recolección (FETCh), lectura (READ), y medida (MEASure).

La Figura 3.8 muestra las relaciones existentes entre las diferentes instrucciones de medición. La instrucción de medida MEASure aporta una secuencia de medición completa, configurando el instrumento, tomando la medición y devolviendo el resultado de una operación. Sin embargo, a menudo es requerido un control más preciso sobre la medición, por lo que es considerado el empleo de las instrucciones de configuración CONFigure y lectura READ. CONFigure realiza la parte de configuración de la medición, mientras que la adquisición de datos, pos procesamiento y obtención de la medición es realizada por READ. La instrucción READ contempla la ejecución de dos instrucciones adicionales, una de inicialización INITiate y otra de recolección FETCh. INITiate realiza la adquisición de datos, esta orden no es esencialmente una instrucción del grupo de medición ya que forma parte de las ordenes descritas por el subsistema de disparo TRIGger, las cuales obedecen al modelo para instrumentos SCPI. FETCh realiza funciones de pos procesamiento y obtención de la medición, que permiten al usuario obtener características o aplicar diferentes funciones sobre un simple conjunto de datos ya adquiridos. Del conjunto de instrucciones de medición, MEASure proporciona el mayor grado de compatibilidad entre instrumentos, debido a que no requiere de una base de conocimientos sobre el instrumento, para efectuar la operación. CONFigure y READ son menos compatibles si una reconfiguración del instrumento es requerida entre operaciones de configuración y lectura dado que la configuración es una especificación del instrumento. FETCh también presenta un bajo grado de compatibilidad, debido a que es precisa una comprensión adecuada de los procedimientos empleados por el instrumento para determinar si la información necesaria para una operación ha sido obtenida.

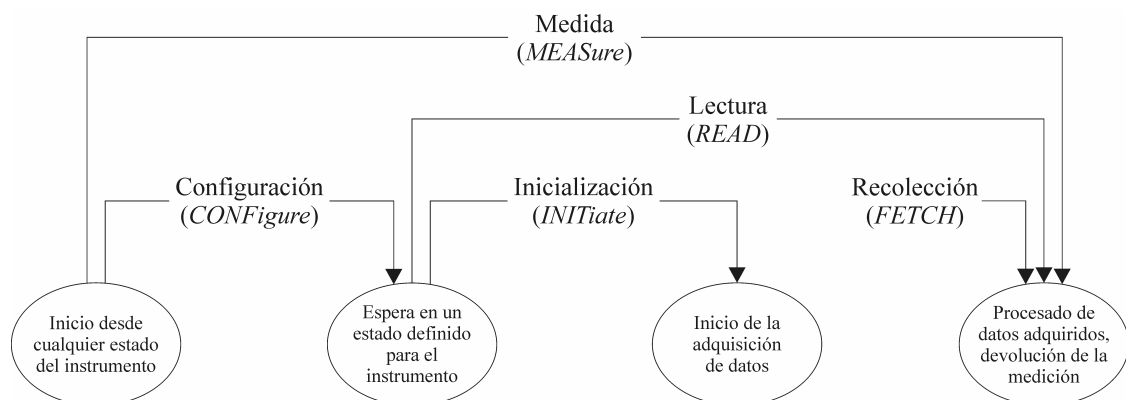


Figura 3.8. Grupo de instrucciones de medición.

La Tabla 3.5 describe la sintaxis de este tipo de instrucciones. Los elementos incluidos en la construcción de una instrucción de medición son el tipo función, los parámetros empleados y de manera opcional una lista de fuentes. El tipo de función define la operación de medición señalada por la instrucción. Si los parámetros son omitidos, la instrucción asume los valores atribuidos por defecto para la función particular. La lista de fuentes, sintacticamente igual al parámetro de lista de canal descrito por la norma (inciso 3.2.2.5), especifica el puerto físico que será fuente de la medición. De esta manera el formato general para una instrucción de medición es el siguiente:

```
CONFigure|FETCh|READ|MEASure: <función>? <parámetros> [,<lista de fuentes>]
```

Esta representación compagina tanto la forma de ejecución como de cuestionamiento de la instrucción, siendo esta la razón básica por la que se hace hincapié en llamarles instrucciones y no ordenes. La excepción a esta regla es la instrucción de configuración, quien presenta una forma para su ejecución y otra para su cuestionamiento.

Tabla 3.5. Sintaxis de instrucciones de medición.

Instrucción	Descripción de sintaxis
CONFigure	Presenta tanto una forma de orden como de pregunta. La diferencia en sintaxis entre una orden y una pregunta radica en el empleo o no del carácter de interrogación (?). La instrucción a manera de pregunta devuelve la última configuración realizada por una orden de configuración o medida. Su formato de respuesta es una listado con la función y sus parámetros. La ejecución de una orden de configuración puede afectar el valor de algún otro parámetro del instrumento. La orden de reinicio *RST reestablece la configuración del dispositivo.
FETCh, READ	Solo existen en forma de pregunta. Cuando el argumento de función es omitido, la última función recolectada, leída o medida es usada. Si un dato generado es inválido, la instrucción no será completada hasta que todos los datos sean validos. Si un parámetro usado resulta ser inválido será generado un error (-221). El formato de los datos devueltos por estas ordenes puede ser manipulado mediante el uso de las ordenes del subsistema de formato FORMat.
MEASure	Solo existe en forma de pregunta. Las funciones usadas en conjunción con esta y en general con cualquier instrucción de medición difieren a las descritas por el subsistema de especificación SENSE, debido a que están, orientadas a la señal y son potencialmente independientes del hardware. El formato de los datos devueltos se encuentra sujeto a lo señalado mediante el subsistema de formato FORMat.

3.5. Registros de estado

Fue el estándar ANSI/IEEE 488.2 de 1987 el primero en definir un modelo estándar de bytes de información sobre el estado de los dispositivos. Este hecho supuso un paso más hacia la compatibilidad de instrumentos.

SCPI establece en base a los mecanismos descritos en el capítulo 11 del estándar IEEE 488.2, una estructura estándar de registros de estado para instrumentos SCPI.

En adición a lo establecido por la norma IEEE 488.2 SCPI define registros de cuestionamiento y operación, así como ordenes de condición, evento y habilitación asociadas con estos. La Figura 3.9 muestra la estructura mínima requerida por el sistema de registros de estado en instrumentos SCPI. Requerimientos adicionales para dispositivos con múltiples instrumentos lógicos o extendidas capacidades de disparo también son estimados por esta norma.

En general, los registros de estado señalados por SCPI tienen la forma de un entero de 16 bits, con el más significativo bit puesto en cero, esto debido a que algunos controladores pueden tener dificultad en la lectura de un entero de 16 bits. Los registro establecidos por SCPI conforme a las especificaciones del estándar IEEE 488.2 constan de los siguientes elementos:

- *Registros de condición:* monitorea continuamente el estado del hardware y firmware del dispositivo. No existe una retención de datos por parte de este registro.

- **Filtros de transición:** define los bits del registro de condición que serán puestos de manera correspondiente en el registro de eventos. Existen dos tipos de filtros, positivos y negativos. Un filtro positivo permite a un evento ser reportado cuando una condición cambia de falso a verdadero, mientras un filtro negativo lo hará cuando esta cambie de verdadero a falso.
- **Registro de eventos:** retiene la transición de eventos del registro de condición tal como lo especifican los filtros de transición positivos y negativos. Los bits en el registro de eventos son retenidos, y una vez puestos, estos son mantenidos hasta ser limpiados por una orden que pregunte su contenido, o el envío de la orden *CLS.
- **Registro de habilitación de eventos:** especifica los bits en el registro de eventos que pueden generar un bit de resumen. Los bits de resumen son usados por el byte de estado o bien por un registro de jerarquía superior al que lo genera.

El sistema de registros de estado esta compuesto de múltiples registros en una disposición jerárquica. Los registros de estado de más baja prioridad propagan sus datos a los registro de más alta prioridad, mediante un bit que resume el contenido del registro.

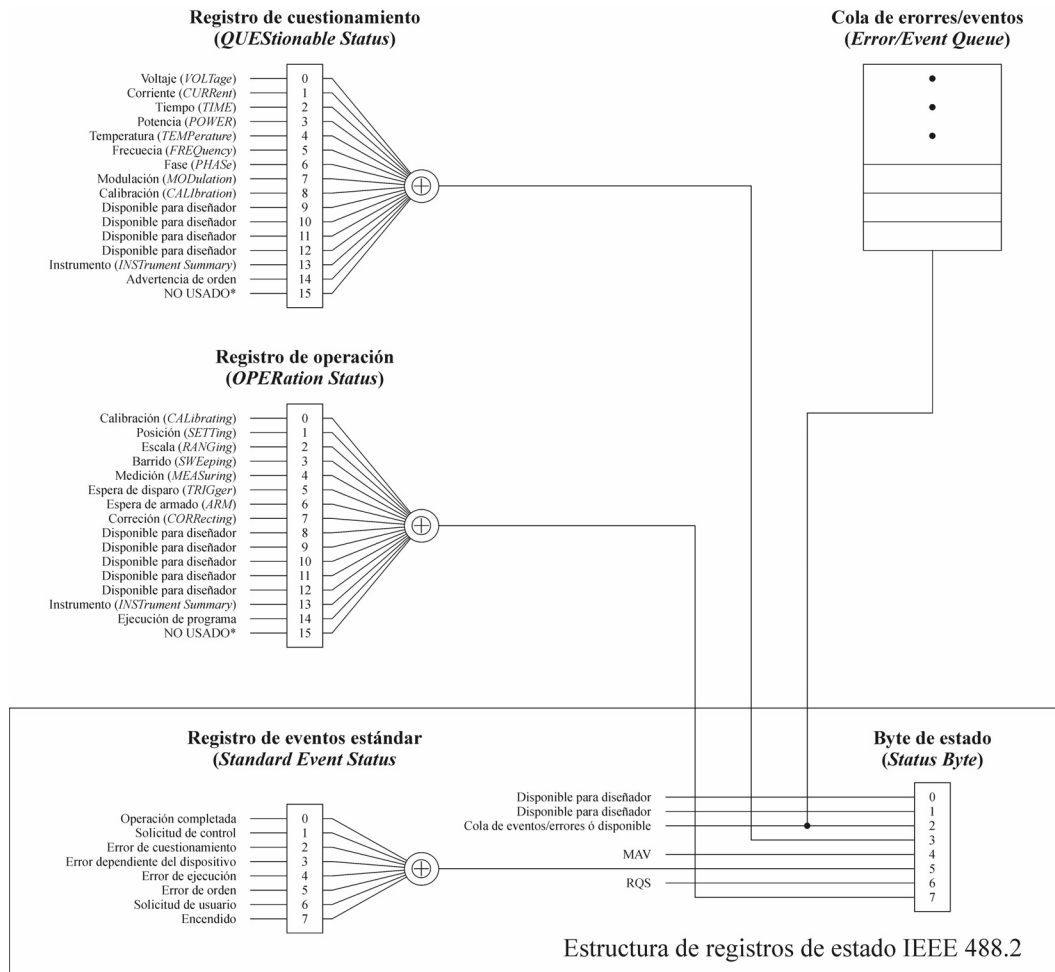


Figura 3.9. Estructura mínima para el sistema de registros de estado en instrumentos SCPI.

Los registros de estado (excepto por los registros de byte de estado y eventos estándar) están formados de elementos o registros, cuyo contenido puede ser usado para producir bits de resumen de estado. Estos bits de resumen son manipulados de la siguiente manera: El valor del registro de condición pasa a los filtros de transición negativos o positivos, para a continuación ser guardados en el registro de eventos. El contenido del registro de eventos son lógicamente multiplicados con el valor del registro de habilitación de eventos y el resultado es lógicamente sumado para producir un bit de

resumen de estado. Este bit de resumen es pasado al registro de byte de estado de manera directa o a través del registro de cuestionamiento. De esta forma será el byte de estado el que resumirá la condición del instrumento.

Dos métodos son usados para acceder de forma programática a la información contenida en los registros de estado. El método de sondeo y de solicitud de servicio. Mediante el método de sondeo, el dispositivo toma un rol pasivo, así el instrumento solo informara sobre su condición cuando el controlador lo demande. Este método resulta efectivo cuando no se requiere conocer los cambios en el estado del instrumento cuando estos ocurran. El método de solicitud de servicio, lleva al instrumento a un rol activo, de forma tal que este informe al controlador cuando una condición haya sido modificada. Este método requiere la manipulación de un registro de habilitación de solicitud de servicio, que no es contemplado por la estructura básica mostrada en la Figura 3.9. Bajo este método, si una condición cambia, el dispositivo habilita el bit 6 RQS del registro de byte de estado, así como la línea de solicitud de servicio (SRQ, *Service Request*) del bus usado, si este la contiene, tal es el caso de GPIB.

3.5.1. Cola de errores y eventos

La cola de errores y eventos es una estructura empleada para el almacenamiento de errores y eventos ocurridos en el instrumento, en una forma una numérica y textual. Las consultas a esta cola tendrán una respuesta con el formato siguiente:

```
<Número>,"<Descripción>;<Información dependiente del dispositivo>"
```

De esta manera, una respuesta particular se presentara como:

```
-131,"Invalid suffix;FREQuency:CENT 2.0E+5 dBuV"
```

El número de error o evento es un entero único en el rango de -32768 a 32767. Los números positivos son asignados al diseñador del instrumento y hacen referencia a errores o eventos dependientes del instrumento. Los números negativos son reservados y especificados por la norma SCPI. El valor de cero es reservado para indicar que ningún error o evento ha ocurrido. La longitud máxima de las cadenas de descripción y de información, es de 255 caracteres. La información textual dependiente del dispositivo puede ser opcional.

Si existe un sobre flujo en la cola, el último error o evento en esta, será remplazado con el error número -350 de sobre flujo, haciendo que cualquier nuevo error o evento sea puesto en la posición del antepenúltimo error o evento ingresado. La orden común IEEE 488.2 *CLS es empleada para limpiar dicha estructura. Las ordenes SCPI utilizadas para el acceso a la cola de errores y eventos se encuentran bajo el subsistema de sistema SYSTem en la subdivisión de error.

3.5.2. Byte de estado

El registro de byte de estado o normalmente llamado solo byte de estado (STB, *Status Byte*), surge como argumento en la norma 488.1, en él se marcan además de la solicitud de atención al controlador, sucesos como mensajes disponibles o el resumen de registros que le preceden. La Tabla 3.6 describe la acción que cada uno de los bits de este registro representa a su activación (valor de 1).

Las ordenes comunes IEEE 488.2 relacionadas con el byte de estado son: *STB, *SER y *SER?. La orden *STB permite obtener el contenido del byte de estado. Las ordenes *SER y *SER? son usadas sobre el registro de habilitación de solicitud de servicio (SRER, *Service Request Enable Register*), que en conjunto con el byte de estado facilitan la implementación del método para el acceso a la estructura de registros de estado llamado solicitud de servicio, característico de solo algunos buses de información. Aunque el SRER no es contemplado en el sistema básico de registros de estado, este se puede entenderse como un registro de habilitación del byte de estado, con la característica especial que el bit de resumen generado por este, es una retroalimentación al bit 6 del mismo byte de estado.

Tabla 3.6. Bits del byte de estado.

Bit	Nombre	Definición
0-1		Disponible para el diseñador.
2	Cola de errores/eventos	La cola de errores SCPI contiene al menos un mensaje de error.
3	Estado cuestionable	El registro de cuestionamiento ha sido modificado. Una cualidad de la señal ha cambiado.
4	Mensaje disponible (MAV, <i>Message Available</i>)	El instrumento dispone de un dato en la cola de salida.
5	Estado de eventos estándar	El registro de eventos estándar ha sido modificado. Una circunstancia como errores o acciones de control generadas por procesos internos o externos al dispositivo ha ocurrido.
6	Servicio a solicitud (RQS, <i>Request Service</i>)	El instrumento tiene al menos una razón para reportar un cambio en el estado del instrumento.
7	Estado de operación	El registro de operación ha sido modificado. Una condición que es parte de la operación norma del instrumento ha cambiado.

3.5.3. Registro de eventos estándar

El registro de eventos estándar (SESR, *Standard Event Status Register*), es usado para determinar circunstancias como errores o acciones generadas por procesos internos o externos al dispositivo. Este registro, ya considerado en el estándar IEEE 488.2, emplea solo un registro de habilitación de eventos. La Tabla 3.7 describe los eventos y errores que los bits de este registro representan a su activación (valor de 1). La modificación de este registro es reportada en el bit 5 del byte de estado. Las ordenes comunes IEEE 488.2 utilizadas sobre este registro son: *ESE, *ESE?, *ESR y *OPC.

Tabla 3.7. Bits del registro de estado de eventos estándar.

Bit	Nombre	Definición	Número de error/evento
0	Operación completada	El protocolo de sincronización del instrumento ha sido habilitado por la orden *OPC, completando todas las operaciones pendientes.	-899 a -800.
1	Solicitud de control	El instrumento requiere la atención del controlador. Evento propio del estándar IEEE 488.1.	-799 a -700
2	Error de cuestionamiento	El control de la estructura de salida del instrumento ha detectado un problema con el protocolo de intercambio de mensajes descrito en el estándar IEEE 488.2.	-499 a -400.
3	Error dependiente del dispositivo	Ciertas operaciones del dispositivo no han sido apropiadamente completadas, debido a una condición anormal del hardware o firmware.	-399 a -300 y de 1 a 32767.
4	Error de ejecución	Un error ha sido detectado en el bloque de control de ejecución del instrumento. Esto ocurre cuando el parámetro de datos que siguió a una cabecera fue evaluado por el dispositivo como inconsistente o fuera de rango. O bien cuando un mensaje valido no puede ser apropiadamente ejecutado debido a la condición actual del dispositivo. Los errores de ejecución son reportados por el dispositivo después que la evaluación de la expresión tenga lugar.	-299 a -200.
5	Error de orden	Un error de sintaxis ha sido detectado por el analizador gramático del instrumento, esto significa que un mensaje del controlador al dispositivo ha sido recibido con una violación al estándar IEE 488.2. Las posibles violaciones incluyen cabeceras de programa no reconocidas y formatos de datos no aceptados por el instrumento.	-199 a -100.

Bit	Nombre	Definición	Número de error/evento
6	Solicitud de usuario	El instrumento ha detectado la activación de una solicitud de control local del instrumento por parte del usuario.	-699 a -600
7	Encendido	Una transición de apagado a encendido de la fuente de alimentación del instrumento ha ocurrido.	-599 a -500

3.5.4. Registro de operación

El registro de estado de operación contiene condiciones que son parte de la operación normal del instrumento, cada bit del registro, describe una particular situación del dispositivo. La Tabla 3.8 resume la definición de cada uno de estos bits.

Tabla 3.8. Bits del registro de operación.

Bit	Nombre	Definición
0	Calibración	El instrumento se halla actualmente en un proceso de calibración.
1	Posición	El instrumento espera por una señal de control para iniciar la medición.
2	Escala	El instrumento se encuentra cambiando de escala.
3	Barrido	El instrumento esta efectuando un proceso de barrido.
4	Medición	El instrumento se encuentra en una actividad de medición.
5	Espera de disparo	El instrumento esta en un estado de espera de disparo “TRIG”.
6	Espera de armado	El instrumento esta en un estado de espera de disparo “ARM”.
7	Corrección	El instrumento esta ejecutando una corrección.
8-12		Disponibles para el diseñador.
13	Resumen de instrumento	Uno de múltiples instrumentos lógicos ha reportando un estado de operación.
14	Ejecución de programa	Un programa definido por el usuario se encuentra en un estado de ejecución.
15		No usado.

3.5.5. Registro de cuestionamiento

El registro de cuestionamiento marca ciertos sucesos propios del instrumento. Contiene bits que proporcionan un indicador de calidad de varios aspectos de la señal. Un bit puesto en este registro indica que los datos adquiridos o generados por el dispositivo son de cuestionable calidad, debido a alguna condición específica. La Tabla 3.9 expone la definición de cada uno de estos bits.

La Figura 3.10 muestra una expansión de este registro. El número de bits de resumen de este registro se implementen en un dispositivo, dependerá tanto del tipo de instrumento del que se trate, como de las características funcionales que éste desarrolle.

Tabla 3.9. Bits del registro de cuestionamiento datos/señal.

Bit	Nombre	Descripción
0-8	Bits de resumen (voltaje, corriente, tiempo, potencia, temperatura, frecuencia, fase, modulación, calibración)	Asociados con un parámetro singular, estos bits de resumen son generados por registros de jerarquía inferior que especifica claramente la condición particular del dispositivo, definiendo sucesos tales como un estado de desnivel, desbloqueo, invalidación, fallo del parámetro en cuestión, entre otros. La Figura 3.10 ilustra como estos bits de resumen son concretados.

Bit	Nombre	Descripción
9-12		Disponibles para el fabricante del instrumento. El diseñador del dispositivo debe ser consiente que la adición de registros al modelo de estado incrementa su complejidad, al mismo tiempo que este puede ser el único camino para notificar información sensitiva sobre estado del instrumento.
13	Resumen de instrumento	Asociado con el manejo de múltiples instrumentos lógicos.
14	Advertencia de orden	Indica una advertencia no fatal, relacionada con la interpretación de una orden, cuestionamiento o uno o más parámetros de una específica orden o pregunta. Este bit es una advertencia a la aplicación, indicando que el estado del instrumento o acción probablemente aceptada puede derivar en alguna situación particular.
15		No usado.

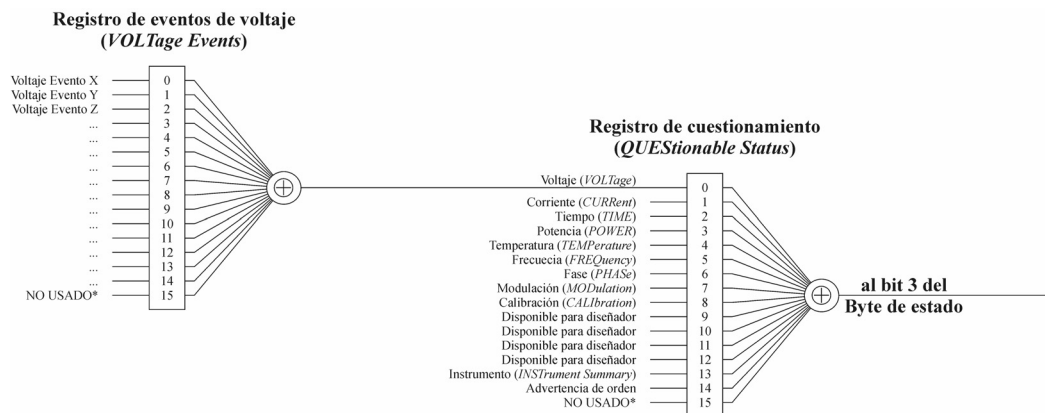


Figura 3.10. Jerarquía expandida del registro de estado de cuestionamiento.

4. Directrices para el control de instrumentos SCPI

La construcción de sistemas ATE presenta dos desafíos importantes. La selección de elementos y la utilización de procedimientos. El primero es representado por la naturaleza cambiante del hardware y software implícitos en la tarea. La diversidad de tecnologías de hardware, como GPIB, RS-232, VXI, PXI, etc. así como la cantidad de ambientes de desarrollo de aplicación existentes, como VEE, LabVIEW, C/C++ o BASIC, hacen difícil establecer un esquema general para el desarrollo de la tarea del control de instrumentos. Por otro lado, la utilización de procedimientos adecuados, representa la oportunidad de maximizar el rendimiento de los elementos de hardware y software involucrados, llevando a estos sistemas de eficaces a eficientes.

En la práctica, la mayor parte de los desarrolladores de sistema ATE enfocan su atención en los elementos de hardware y software a emplear, dejando de lado los mencionados procedimientos de control pese a la existencia de modelos conceptuales para el control de instrumentos, como SCPI, que conllevan de manera inherente la concepción de procedimientos particulares. La diversidad de elementos que son involucrados en la construcción de sistemas ATE, hace casi imposible determinar una secuencia específica de pasos para una tarea, sin embargo este hecho no aísla la posibilidad de contar con una serie de preceptos como producto del estudio de un modelo conceptual único como lo es la norma SCPI.

Bajo este escenario, el presente trabajo plantea una serie de directrices que ayuden de manera significativa al desarrollo de aplicaciones sobre el control de instrumentos mediante SCPI. Estas directrices que no son reglas sino consejos basados en el estudio y aplicación de la norma. Los juicios bajo los cuales han sido concebidas estas directrices son:

- La reducción del tiempo en el desarrollo de aplicaciones.
- El incremento en la eficiencia de los sistemas desarrollados.
- El empleo integral de los recursos provistos por los instrumentos.

Las directrices planteadas no pretenden ser un manual práctico, que instruya una afirmación como: «en caso de que ocurra X, se debe hacer Y». En cambio, estas directrices buscan ayudar a resolver problemas comunes ocurridos durante el desarrollo de aplicaciones referentes al control de instrumentos. Así, la decisión respecto al uso de estas directrices, se encuentra sujeta al conocimiento y sentido común del desarrollador.

Las directrices que en su mayoría procuran ser de carácter general, sugieren su uso de manera autónoma al bus empleado por el instrumento, y a la plataforma de software utilizada por el controlador.

Las mencionadas directrices se hallan enfocadas en cuatro aspectos básicos: la selección de órdenes, el manejo de datos, el acceso a memoria y el control del sistema de registros de estado. La Tabla 4.1 muestra las enunciaciones y objetivos atribuidos a cada una de las directrices planteadas.

Tabla 4.1. Directrices para el control de instrumentos.

Nombre	Enunciado	Objetivo
Selección de ordenes	<i>“Una política clara que defina objetivos, prioridades y servicios para una aplicación de control de instrumentos, permitirá seleccionar de manera adecuada el tipo y ordenes a emplear.”</i>	Proporcionar una serie de criterios que faciliten la elección y uso de ordenes para el control de instrumentos.
Manejo de datos	<i>“Las transacciones que impliquen la transferencia de información numérica y de arreglos, serán hechas obedeciendo un formato de datos determinado. Todas las operaciones de envío de grandes cantidades de datos, no accesibles de manera directa en un instrumento, requieren de la utilización de bloques de datos de longitud definida.”</i>	Denotar la importancia de la selección del formato de datos y empleo de bloques de datos en transacciones que implique un gran volumen de datos.
Acceso a memoria	<i>“La administración del recurso de memoria de un instrumento SCPI permite realizar tareas de almacenamiento y transferencia de datos en formas tan diversas, como trazas y archivos. El manejo de memoria en un instrumento SCPI es hecho a dos diferentes escalas de integración.”</i>	Proveer las herramientas necesarias para la distinción y uso de los diferentes tipos de memoria en instrumentos SCPI. Mostrar la utilidad del recurso de memoria del instrumento en tareas como la obtención de trazas de datos, archivos de imagen, etc.
Control del sistema de registros de estado	<i>“Toda aplicación que requiera la detección de cambios de estado o condiciones de error en un instrumento SCPI, hará uso de la estructura de registros de estado definida por la norma. Existen dos métodos empleados en el manejo de esta estructura, el método de sondeo y de solicitud de servicio.”</i>	Destacar la utilidad de la estructura de registros de estado SCPI, en aplicaciones donde sea necesario el conocimiento del estado actual de los instrumentos. Exponer los métodos usados en la manipulación de la estructura de registros de estado.

Las siguientes secciones, tratan los argumentos de cada una de las directrices desarrolladas. De igual manera, se exhibe el uso de estas directrices, en aplicaciones que permitan no solo exponer los procedimientos sugeridos, sino también proporcionen a las personas interesadas, una idea más clara de los mecanismos usados para el control de instrumentos bajo diferentes plataformas de desarrollo.

4.1. Selección de ordenes

Todas las ordenes definidas por SCPI se encuentran agrupadas en subconjuntos llamados subsistemas, estos a su vez, y con el fin de concebir dos diferentes niveles para el control de instrumentos, son categorizados en dos grandes bloques: subsistemas (en general) e instrucciones de medición (subsistema de medida). Los primeros proveen un control tradicional del instrumento, haciendo uso de ordenes muy específicas, mientras que los segundos suministran en base a un conjunto de instrucciones de alto nivel, dirigidas a la adquisición de datos, un fácil y rápido control del instrumento.

Aunque de algún modo, ambos tipos de ordenes SCPI permiten el manejo de las funciones provistas por un instrumento, en la mayoría de los casos estas son manejadas de manera independiente. Los subsistemas de ordenes son usados generalmente para el control de parámetros del instrumento como visualización, rangos de medición, resolución, almacenamiento, estado, etc. Las instrucciones de medición son designadas a la adquisición de datos generados por el instrumento, esta condición no aparta el hecho que los subsistemas de ordenes permitan la adquisición de algunos tipos de datos, mientras que las instrucciones de medición pueden también ser utilizadas en el control de ciertos parámetros del instrumento.

Una tarea primaria en el desarrollo de aplicaciones relacionadas al control de instrumentos, es la minimización de errores, productos del uso de las mismas ordenes. Los siguientes son algunos consejos utilices en el empleo de toda orden SCPI:

- Sobre la notación. La representación textual de una orden dentro de toda documentación SCPI, conlleva la utilización de algunos caracteres como corchetes, llaves y barras verticales. El uso de corchetes ([]) permite resaltar elementos opcionales. Las aplicación de llaves ({}), destacan uno o mas parámetros que pueden ser incluidos cero o más veces. La barra vertical (|) puede ser leída como “o” y es usada para separar parámetros alternativos.
- La omisión de argumentos opcionales ([]) permite una mayor versatilidad en el uso de ordenes, esto a razón que estos argumentos opcionales traducidos dentro de la estructura jerárquica SCPI como nodos, son adheridos con nuevas versiones de la norma con el fin de añadir funcionalidades a los instrumentos.
- El número de transacciones sobre el bus puede ser reducido mediante el envío de múltiples ordenes por transacción, separando estas con un carácter punto y coma (;). En el caso que las ordenes utilizadas formen parte de diferentes subsistemas, la orden subsiguiente deberá iniciar con el carácter dos puntos (:). Si las ordenes usadas pertenecen al mismo subsistema, el mnemotécnico inicial de la orden subsiguiente, no será incluido, pudiendo así también iniciar con el mismo mnemotécnico de la orden ejecutada anteriormente, si ambas se hallan en un mismo nivel de la estructura jerárquica de ordenes de un subsistema.
- El envío de múltiples ordenes por transacción puede ser causa de conflictos de ejecución. Esta situación se presenta, cuando existe una dependencia entre ordenes tal que, el envío de una orden seguida de otra, pueda provocar que las características especificadas por la primera orden sean modificadas a causa de la segunda o bien que la ejecución de la segunda no sea permitida a causa de la primera.
- Establecer y preguntar un valor para una característica del instrumento mediante una sola transacción, puede generar devoluciones de datos no esperados. Cuando esto sucede, el valor obtenido de la transacción corresponderá al valor de la característica anterior a la ejecución de la orden de asignación manejada en la transacción.
- El uso de instrucciones de alto nivel, llamadas también de medición, potencializa la compatibilidad entre instrumentos.

Establecer el conjunto de ordenes requeridas para el desarrollo de una aplicación en específico resulta una tarea realmente compleja, sin embargo el conocimiento general de la totalidad de subsistemas que la norma SCPI precisa, facilita la definición de ciertas tareas que de manera general son demandadas en la gran mayoría de aplicación relacionadas con el control de instrumentos. La Tabla 4.2 muestra las acciones implicadas en algunas de estas tareas críticas.

Tabla 4.2. Tareas comunes requeridas en aplicaciones de control de instrumentos.

Tarea	Acciones implícitas
Control de una medición específica	Configuración de parámetros referentes a la medición (subsistema SENSE). Selección de unidades empleadas (subsistema UNIT). Control de la medición (subsistemas INITiate y TRIGger). Selección de instrumentos de medición (subsistema INSTRUMENT). Medición (subsistema MEASUREMENT). Manejo de información del sistema. Reinicio, opciones, etc. (subsistema SYSTEM).
Transferencia de información numérica y de arreglos	Control de la medición (subsistema INITiate). Selección del formato de envío de datos (subsistema FORMat). Manipulación de trazas y datos. Memoria interna (subsistema TRACE DATA).

Tarea	Acciones implícitas
Manejo de fuentes y generadores	Configuración de referencia al puerto (subsistemas SENSE, ROUTE y TRIGGER) Control de las características del puerto. Encendido, etc. (subsistema OUTPUT). Control de las características de la señal. Amplitud, etc. (subsistema SOURCE).
Gestión de archivos de imagen, datos, etc.	Acceso a memoria de almacenamiento masivo (subsistema MEMORY).
Prestaciones de la interfaz con el usuario	Operaciones de procesamiento. Límites, marcadores, etc. (subsistema CALCULATE). Selección de representaciones textuales y gráficas (subsistema DISPLAY).
Monitoreo de acciones del instrumento	Control del sistema de registro de estados (subsistema STATUS). Operaciones sobre registros de estado (comandos comunes IEEE 488.2). Manejo de información del sistema. Errores, tiempo, etc. (subsistema SYSTEM).
Calibración del instrumento	Alineamiento del instrumento (subsistema CALIBRATION). Diagnósticos específicos sobre el estado del instrumento (subsistema DIAGNOSTIC). Ejecución de procedimientos de prueba (subsistema TEST).
Administración de dispositivos externos	Configuración de dispositivos de externos. Impresión, etc. (subsistema HCOPI). Manejo de información del sistema. Hardware, etc. (subsistema SYSTEM).
Control de arquitecturas VXI	Gestión de operaciones sobre sistemas VXI (subsistema VXI). Especificación del bus VXI (subsistema OUTPUT, SENSE y TRIGGER).

A pesar de las semejanzas en sintaxis, empleo y utilidad entre subsistemas de órdenes e instrucciones de medición existen argumentos que hacen a estas ser catalogadas en diferentes niveles de control.

4.1.1. Administración de subsistemas

Los subsistemas de órdenes SCPI son producto del modelo general para instrumentos SCPI (inciso 3.3.1). Las órdenes contenidas en estos subsistemas aunque son dedicadas casi en su totalidad al control de parámetros en los instrumentos, también permiten la obtención de trazas de datos, archivos, etc. La categoría de subsistemas deslinda a un conjunto particular de órdenes denominado de medida MEASUREMENT, que es manejado de manera distinta.

Existen subsistemas que por su tamaño y complejidad se encuentran divididos en subconjuntos denominados de igual forma subsistemas. Sin embargo estos subconjuntos se caracterizan por no estar ligados directamente al modelo general para instrumentos establecido por la norma.

Todas las órdenes contenidas en estos subsistemas a excepción de un conjunto obligatorio formado por algunas órdenes de los subsistemas de estado STATUS y sistema SYSTEM, son consideradas como opcionales.

4.1.2. Administración de instrucciones de medición

Dentro de SCPI se le conoce como instrucciones de medición al conjunto de órdenes destinadas a la adquisición de datos mediante el uso de un grupo de órdenes de alto nivel. Estas instrucciones son agrupadas bajo el subsistema de medida MEASUREMENT, sin embargo a diferencia del resto de los subsistemas contemplados por la norma, el subsistema de medida tiene como referente las características de la señal medida y no el modelo para instrumentos planteado por SCPI.

Aunque no se atiende como una regla, el empleo de instrucciones de medición es ineludible en algunos tipos de instrumentos SCPI, como en el caso de los medidores digitales. Así, instrumentos como voltímetros, amperímetros y ohmímetros siempre harán uso de este tipo de órdenes.

La presencia de instrucciones de medición es habitual en instrumentos que desarrollan algún tipo de medición particular. Mediciones como periodo, frecuencia, amplitud, fase, etc. en instrumen-

tos como osciloscopio, así como mediciones de distorsión armónica, potencia de canal, etc. definidas para algunos instrumentos como analizadores de espectros y vectoriales, son controlables mediante este conjunto de ordenes particular.

Este conjunto particular de instrucciones define acciones de configuración (CONFigure), recolección (FETCh), lectura (READ), y medida (MEASure), representadas a su vez en cuatro diferentes instrucciones. El formato general para cualquiera de estas instrucciones de medición es el siguiente:

```
<instrucción>: <función>? <parámetros> [,<lista de fuentes>]
```

El tipo de función define la operación de medición a realizar. Si los parámetros son omitidos, la instrucción asume los valores especificados por defecto para la función señalada. La lista de fuente especifica el puerto físico que será fuente de la medición. Esta representación compagina tanto la forma de asignación como de pregunta de la instrucción, siendo esta la razón por la que se hace hincapié en llamarles instrucciones y no ordenes. La excepción a esta forma es la instrucción de configuración, quien presenta una forma para el envío de una asignación y otra para el de una pregunta. Sí así es dispuesto por el instrumento, el formato de los datos devueltos por las instrucciones de medición es determinado por el subsistema de formato FORMat.

La Tabla 4.3 describe el uso y aplicación de las instrucciones de medición expuestas por SCPI. Adicionalmente se presentan algunas equivalencias funcionales entre instrucciones.

Tabla 4.3. Empleo de instrucción de medición.

Instrucción	Equivalencia	Descripción
Configuración (CONFigure)		Alista al instrumento para el desarrollo de una medición especificada por su argumento de función (necesario). Esta instrucción, suspende la actual medición (si existe alguna) y establece los parámetros para la medición en cuestión. Si la declaración de los parámetros es omitida o no permitida por el instrumento, los valores por defecto manejados por la medición serán usados. La instrucción de configuración no inicializa la adquisición de datos, solo establece parámetros y permite acceder al modo de medición. Algunos instrumentos realizan una adquisición de datos “falsa” como producto de la ejecución de esta instrucción.
Recolección (FETCh)		Selecciona los datos producidos por la más reciente medición y pone estos en el buffer de salida del dispositivo. Los datos usados son recuperados de la última orden de inicialización INITiate valida. El uso de la instrucción de recolección es recomendado en aplicaciones donde se requiere realizar operaciones sobre los datos de una medición ya realizada. Si un instrumento en un modo de medición particular, recibe una orden de recolección referente a un tipo de medición que no corresponde con el modo de medición actual, será generado un error, esto debido a que, la instrucción de recolección no selecciona un modo de medición.
Lectura (READ)	ABORt; INITiate; FETCh[]?<>[];	Provee un método de recolección de datos para una medición actual. Inicializa una medición y coloca los datos generados en el buffer de salida del instrumento, así una instrucción de lectura permite el inicio de un modo específico de medición, supliendo en parte el uso de instrucciones de configuración y haciendo de esta una instrucción recomendable sobre otras, en algunas tareas.
Medida (MEASure)	ABORt; CONFigure:<><>; READ:<?<>[];	Provee una secuencia de medición completa, incluyendo configuración y lectura de datos. La instrucción de medida es usada en situaciones donde sean requeridas modificaciones constantes sobre los parámetros de una medición en proceso. Mientras el proceso de medición es realizado el dispositivo suprime muchas veces algunas funciones de comunicación, este hecho reduce en parte la utilidad de esta instrucción.

Una orden de mención constante en el trato de instrucciones de medición es la orden de inicialización `INITiate` perteneciente al subsistema del mismo nombre. Esta orden inicializa un ciclo para una medición sin afectar sus parámetros, ni dispone datos al buffer de salida del instrumento. La utilidad de las ordenes de inicialización es destacada en el control de rutinas de disparo, iniciando, pausando y reiniciando mediciones.

La Figura 4.1 muestra un procedimiento desarrollado bajo Agilent VEE, que permite ilustrar de manera clara el uso indistinto de subsistemas e instrucciones de medición. La aplicación en cuestión realiza una medición de potencia de canales para 802.11, mediante el uso de un Analizador de Espectros que posibilita mediciones de potencia de canal (*Channel Power*). Como se observa la configuración de los parámetros para la medición son realizados a través de subsistemas de ordenes, en el caso particular del subsistema de especificación `SENSe`, mientras la adquisición de datos se lleva a cabo mediante la instrucción de medición `READ`.

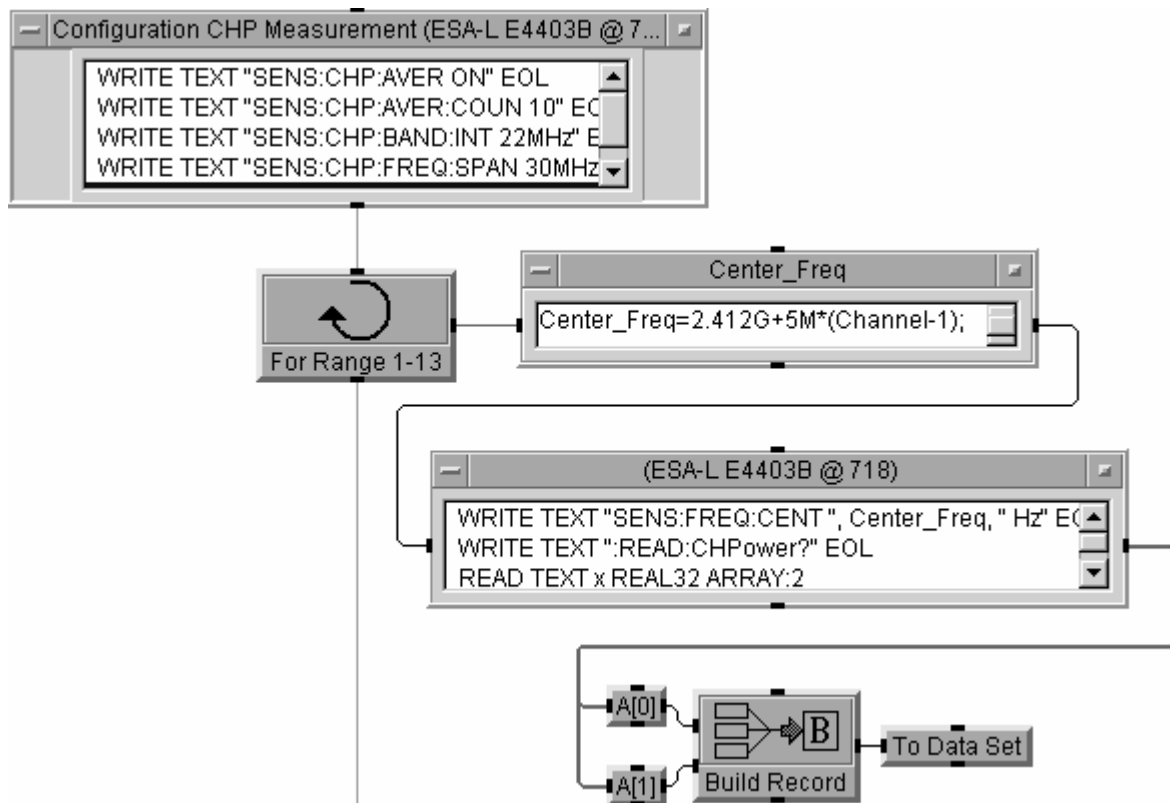


Figura 4.1. Uso de subsistemas e instrucciones de medición en una aplicación bajo Agilent VEE.

4.2. Manejo de datos

La definición de datos dentro de SCPI es un producto de la especificación de los parámetros permitidos en las ordenes (inciso 3.2.2). Un parámetro indica el valor y disposición de los argumentos en una orden. Los parámetros manejados por SCPI pueden ser de tipo carácter, numérico, booleano, unidad y expresión.

La forma en como son manejados los tipos de parámetros en un instrumento es propia del diseñador del mismo. Así, en un dispositivo particular se podrá hacer mención de infinidad de tipos de parámetros, sin embargo estos son solo categorías o construcciones de los parámetros considerados por la norma, tal es el caso del parámetro denominado tiempo `<time>`, tratado por algunas ordenes en ciertos instrumentos, que no es más que un conjunto de valores numérico.

Un elemento importante relacionado con el manejo de los datos es la selección de un formato, este argumento es común de los parámetros numéricos.

Las transacciones que involucran grandes volúmenes dentro de SCPI, son hechas a través de bloques de datos definidos de una manera particular. El uso de datos en arreglos es propio de las instrucciones de medición.

4.2.1. Selección del formato de datos

SCPI permite la elección del formato de datos usado en la transferencia de información numérica y de arreglos. Esta tarea es realizada mediante el conjunto de ordenes agrupadas bajo el subsistema de formato FORMat.

El formato de datos es usado sólo por ordenes y respuestas que son especialmente diseñadas para ser afectadas por este subsistema. Así, en un instrumento solo determinadas ordenes tomarán en cuenta la especificación de este formato de datos.

Para ciertas ordenes la especificación del formato de datos hecha mediante el subsistema de formato puede ser anulada durante la transferencia de datos, si el dato recibido indica por si mismo su tipo.

Las ordenes incluidas en el subsistema de formato permiten seleccionar el tipo de dato a manejar y el orden en como los octetos de información serán enviados. La Tabla 4.4 muestra las ordenes de formato manejadas por la norma SCPI versión 1999.0.

Tabla 4.4. Ordenes de formato SCPI.

Orden	Descripción
FORMat:BORDER NORMAL SWAPped	Selecciona el orden en que los octetos de datos binarios serán transferidos. En modo normal NORMAL, la secuencia de envío de octetos inician con el bit más significativo y finaliza con el menos significativo. El modo de cambio SWAPped permite realizar la transferencia de manera inversa.
FORMat[:DATA]<tipo>[,longitud]	<p>Cambia el formato de datos, en tipo y longitud. El tipo de dato a establecer pueden ser: ASCii, INTeger, UINteger, REAL, BINary, HEXadecimal, OCTal, y PACKed. El parámetro de longitud es opcional para todos los tipos de dato.</p> <p>Si el tipo de dato seleccionado es INTeger, UINteger, REAL o PACKed, los datos son transferidos en bloques de longitud definida. Bajo condiciones de reinicio todos los instrumento SCPI hacen uso del formato ASCii por defecto. Sin embargo la transferencia de datos es más rápida usando formatos binarios como enteros, enteros sin signo y reales. En una transferencia de datos en formato ASCII los números son separados por comas, tal como lo especifica la norma IEEE 488.2.</p> <p>Las longitudes validas para el tipo de dato entero INTeger son de 16 y 32, mientras que para el entero sin signo UINteger lo son 8, 16, 32 y 64. El tipo real REAL permite longitudes de 32 y 64 bits.</p> <p>Para los tipos de dato BINary, HEXadecimal y OCTal el parámetro de longitud representa el número de dígitos por cada número, en el formato correspondiente. Los datos son precedidos según sea el caso, de una cabecera (“#B”, “#H” o “#O”) tal como lo especifica la norma IEEE 488.2.</p> <p>El tipo de dato empaquetado PACKed transfiere un bloque de datos de longitud definida, de una manera especificada por el dispositivo. El parámetro de longitud puede ser ocupado para cubrir información adicional acerca del formato dependiente del dispositivo.</p>
FORMat:SREGister ASCii BINary HEXadecimal OCTal	Selecciona el tipo de dato de respuesta a una orden de pregunta sobre los registros de condición, evento ó habilitación, y filtros de transición manejados mediante el subsistema de estado STATus. Esta selección afecta de igual forma, al resto de los registros de estado contemplados por el estándar IEEE 488.2

4.2.2. Uso de bloques de datos de longitud definida

Un elemento particularmente útil en el envío de grandes cantidades de datos, tales como trazas de datos, archivos de imágenes entre otros datos, no accesibles en un dispositivo de manera directa, es la utilización de bloques de datos de longitud definida.

La estructura manejada en los bloques de datos de longitud usados en ordenes SCPI corresponde a la definición asentada en la sección 8.7.9.2 del estándar IEEE 488.2. Un bloque de datos de longitud definida conlleva una cabecera ASCII (iniciada con el carácter “#”) que señala la cantidad de datos adicionales en el bloque. La Figura 4.2 ilustra su estructura.

Cabecera		Octetos de datos	
#	d	n	m
#	4	1604	...1604 octetos

Donde: d = Numero de dígitos/octetos adicionales en la cabecera
 n = Cantidad de octetos de datos que siguen a la cabecera
 m = Octetos de datos

Figura 4.2. Estructura para bloques de datos de longitud definida.

En el ejemplo mostrado en la figura, “d” exhibe un valor de 4, esto significa que los próximos cuatro dígitos representan el número “n” (1604), cuyo valor revelara la cantidad de octetos que siguen a la cabecera. La cantidad de octetos requeridos para la representación de un dato estará en función del formato de datos empleado. De este modo, un entero de 32 bits requerirá de 4 octetos para la representación de un dato, un entero de 16 bits de 2 octetos, etc. En el ejemplo ilustrado, “n” ostenta un valor de 1604, así en el caso que el formato de datos seleccionado haya sido un entero de 32 bits (4 octetos), se concluye que el bloque contiene 401 (1604/4) datos del tipo correspondiente.

El código siguiente desarrollado sobre C++, permite ilustrar de manera más clara la obtención de bloques de datos, que para el caso particular representan la traza de datos de un instrumento. Nótese el empleo de ordenes para la selección del formato de datos, anticipadamente a la obtención de estos. La rutina mencionada elige el formato y orden para los datos, solicita el bloque de datos a través de una orden del subsistema de traza TRACe, elimina la cabecera del bloque obtenido y almacena los datos obtenidos de acuerdo al formato de datos manejado.

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#include "visa.h"
ViSession defaultRM, viESA;
void main(){
    ViStatus viStatus=0;
    ViChar _VI_FAR cResult[5000]={0};
    ViReal32 dTraceArray[401]={0};
    long lNumberBytes=0L;
    unsigned long lRetCount=0L;
    int iSize=0;
    //BytesPerPoint = 4 para Real32 o Int32, 8 para Real64, y 2 para Uint16
    //bloque de datos: cabecera + octetos de datos,
    //:= cabecera = #nmmmm, n - número de m dígitos en la cabecera
    //      mmmmm - número de octetos de datos que siguen a la cabecera
    //      octetos de datos (Real 32)->401 puntos*4 octetos/punto=1604 octetos
    int iBytesPerPnt=4, iSwpPnts=401;
    int iHeaderBytes=6;//= 2 ("#", "n") + 4 ("1604")
    lNumberBytes=(iSwpPnts*iBytesPerPnt);//1604
    FILE *fTraceFile;
    /*Abrir la sesión para el dispositivo GPIB en la dirección 18*/
    viStatus=viOpenDefaultRM(&defaultRM);
    viStatus=viOpen(defaultRM, "GPIB0::18", VI_NULL, VI_NULL, &viESA);
    if(viStatus){
        printf("Error al abrir sesión para el dispositivo en la dirección 18!\n");
```

```

    exit(0);}
/*Limpiar instrumento*/
viClear(viESA);
printf("\n\t\t Lectura de traza de datos (Real32,GPIB) \n\n" );
/*Selección del formato de datos (Real32)*/
viPrintf(viESA,"FORM:DATA REAL,32 \n");
/*Selección del orden de envío de datos*/
viPrintf(viESA, "FORM:BORD SWAP\n");
/*Calcular el tamaño del bloque de datos*/
/*= num. de dígitos en la cabecera + num. de octetos de datos + terminador*/
iSize = iHeaderBytes + lNumberBytes + 1;//6+1604+1=1611
/*Obtener traza de datos*/
viPrintf(viESA,"TRAC:DATA? TRACE1\n");
viReadAsync(viESA, (ViBuf)cResult,iSize,&lRetCount);//Leer bloque de datos
printf("\n%s\n",cResult);
/*Remover cabecera - Extraer traza de datos*/
memcpy(dTraceArray,cResult+iHeaderBytes,(size_t)lNumberBytes);
/*Salvar traza de datos en archivo ASCII*/
fTraceFile=fopen("C:\\\\READTRACE32GPIB.TXT","w");
for(long i=0;i<lNumberBytes/4;i++)
    fprintf(fTraceFile," Valor del punto [%d] =%.2lf dBm\n",i+1,dTraceArray[i]);
fclose(fTraceFile);
/*Cerrar sesión*/
viClose(viESA);
viClose(defaultRM);
getch();
}

```

4.3. Acceso a memoria

Un elemento indispensable en todo instrumento es la presencia de un elemento de memoria. La memoria dentro de un dispositivo facilita desde el almacenamiento transitorio de valores de medida hasta el de grandes conjuntos de información.

La presencia de un bloque de memoria en instrumentos SCPI no es una condición necesaria, sin embargo, su inevitable uso en todas las operaciones relacionadas con el control del dispositivo, sugiere siempre su existencia. La presencia de elementos de memoria en un instrumento no implica, que estos permitan ser accedados. Los conceptos de memoria en SCPI son apuntes del modelo para instrumentos definido por la norma.

La norma SCPI define dos tipos de memorias para instrumentos, una llamada interna y otra de almacenamiento masivo, para las cuales ofrece a su vez, subsistemas específicos de ordenes.

4.3.1. Memoria interna

La memoria interna de un instrumento es usada para guardar diversos tipos de datos, como lo son ASCII, binarios, estados de instrumento, trazas y macros. Los tipos de datos manejados dependen del dispositivo.

La diversidad de datos y procedimientos utilizados por la memoria interna de un instrumento, impide determinar un conjunto específico de ordenes para su manipulación, por lo que, las operaciones sobre este tipo de memoria son difundidas en diversos subsistemas como lo son, el de calculo CALCulate, trazas y datos TRACe|DATA, sistema SYSTem y programa PROGram.

No obstante, SCPI designa un subsistema de ordenes especial para el trato de la memoria interna de instrumento denominado MEMory. Las ordenes contenidas en este subsistema facilitan el acceso a un bloque de memoria interna del instrumento cuyos datos son agrupados esencialmente en tablas de tipos específicos de datos. Algunas operaciones realizadas mediante este subsistema son remover, copiar, cargar, borrar, intercambiar, liberar y especificar datos.

De forma general, un elemento a considerar en todas las transacciones de acceso a la memoria interna de un instrumento, es el tiempo requerido por las operaciones alrededor de la tarea encomen-

dada. Así, el envío de órdenes de memoria de una manera reiterativa tal, que impida que los datos sean procesados adecuadamente por el instrumento, ocasionara la obtención de información errónea que no representa el estado actual de los datos.

La Figura 4.3 muestra un procedimiento programático desarrollado en LabVIEW para el acceso a un segmento de memoria interna en un instrumento, dicha sección de memoria es utilizada para el almacenamiento de trazas de datos y controlada mediante el subsistema de órdenes de trazas y datos TRACe|DATA. Nótese el empleo de órdenes para la selección del formato de datos, de manera anticipada a la obtención de los mismos. La rutina mencionada solicita de manera repetitiva un bloque de datos a través de una orden del subsistema de trazas TRACe, elimina la cabecera del bloque y obtiene los datos de interés de acuerdo al formato empleado. El objeto de tiempo de espera (30) empleado en la rutina permite al instrumento la adquisición y procesamiento de datos, de forma tal que los datos tratados en cada ciclo sean producto de la medición actual, una alternativa al empleo de este objeto de tiempo es la utilización de la estructura de registros de estado como medio de monitoreo de las acciones del instrumento.

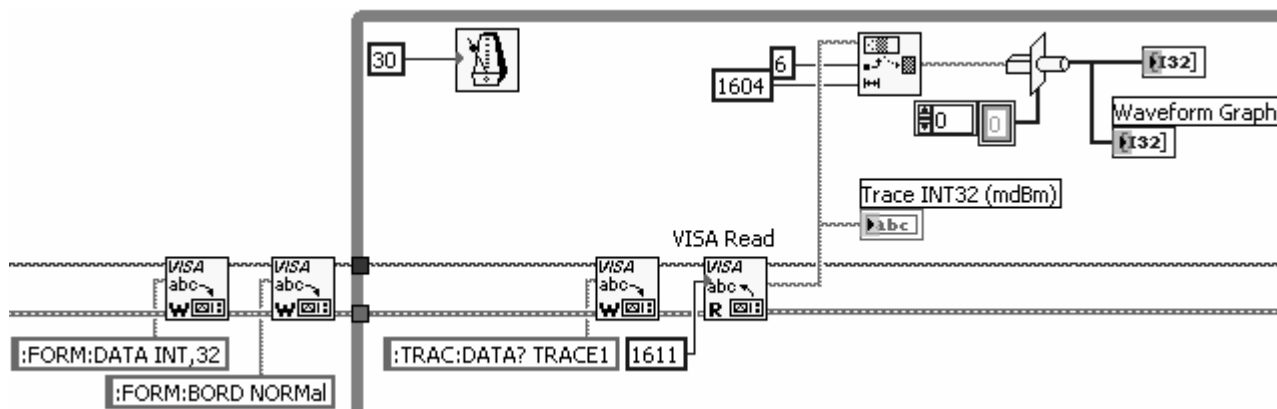


Figura 4.3. Obtención de trazas de datos de un instrumento mediante una rutina en LabVIEW.

4.3.2. Memoria de almacenamiento masivo

Físicamente la memoria de almacenamiento masivo dentro de un instrumento se encuentra conformado por los dispositivos internos y externos tales como discos flexibles y duros, que permiten salvaguardar información bajo un sistema de archivos. Cuando el dispositivo es externo el instrumento deberá tener la capacidad de controlar el bus al dispositivo. SCPI define un subsistema de órdenes de memoria llamado MMEMory (*Mass MEMory*), para el acceso a este tipo de memoria.

Los dispositivos de almacenamiento masivo dentro de un instrumento, son referidos mediante un identificador, cuya nomenclatura es propia de cada instrumento. Cadenas como “C:”, “:700,2”, etc. son usadas comúnmente, sin embargo algunos sistemas de archivos pueden hacer uso de otros identificadores particulares. Las órdenes del subsistema MMEMory hacen uso de un parámetro particular llamado nombre del archivo, el cual no es más que una cadena, que puede contener caracteres para la especificación de directorios (\ ó /) y separadores de extensiones (.).

Las operaciones tratadas por el subsistema MMEMory proveen medios que permiten informar, cerrar, copiar, transferir, borrar, inicializar, cargar, almacenar, seleccionar, mover, nombrar, abrir y recobrar datos y archivos. Todas estas operaciones, a excepción de las de carga y almacenamiento, son realizadas sobre los dispositivos de almacenamiento masivo. Las operaciones de carga LOAD y almacenamiento STORe posibilitan la transferencia de información entre un dispositivo de almacenamiento masivo y la memoria interna del instrumento. La operación STORe transfiere información de la memoria interna a un dispositivo de almacenamiento masivo, mientras que LOAD lo hace del dispositivo de almacenamiento masivo a la memoria interna. Las órdenes LOAD y STORe no cuentan con una forma de pregunta.

Las tareas realizadas normalmente a través del uso de esta memoria, incluyen en su mayoría procesos de obtención de archivo guardados en el instrumento. Los archivos tratados pueden incluir información tan diversa como estados del dispositivo, mediciones y trazas de datos, imágenes de pantallas, etc.

La ejecución de algunas ordenes referentes a la manipulación de memoria consumen un tiempo de ejecución largo, que en ocasiones impide al instrumento el desarrollo de otras actividades, tal es el caso de ordenes de carga LOAD y almacenamiento STORE. En tales casos es recomendable el uso de la orden común IEEE 488.2 de operación completada OPC o bien el control de la estructura de registros de estado asociada al dispositivo, con el fin de optimizar estos tiempos de ejecución.

El código siguiente desarrollado sobre C++, ilustra los medios usados para el acceso a un dispositivo de almacenamiento interno de un instrumento. La aplicación en cuestión guarda, obtiene y borra un archivo de la imagen actual de la pantalla del dispositivo. Cabe destacar el empleo de la orden común IEEE 488.2 de operación completada OPC, con el objeto de verificar que la tarea en el caso de almacenamiento de la imagen en pantalla en un archivo, haya sido completada. La obtención del archivo de imagen requiere del manejo de un bloque de datos de longitud definida.

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include "visa.h"
ViSession defaultRM, viESA;
void main(){
    ViStatus viStatus=0;
    ViChar_VI_FAR cResult[50000]={0}; //archivo de imagen <50K
    char dPictureArray[50000]; //archivo de imagen <50K
    long lNumberBytes=0L;
    unsigned long lRetCount=0L;
    int iSize=0;
    int iHeaderBytes=7; //1+1+5 ejemplo: #511365
    int opc;
    FILE *fPictureFile;
    lNumberBytes=50000; //archivo de imagen <50K ->tamaño de archivo desconocido
    iSize=lNumberBytes + iHeaderBytes+1; //+1 «CR»«LF»
    /*Abrir la sesión para el dispositivo GPIB en la dirección 18*/
    viStatus=viOpenDefaultRM(&defaultRM);
    viStatus=viOpen(defaultRM, "GPIB0::18", VI_NULL, VI_NULL, &viESA);
    if(viStatus){
        printf("Error al abrir sesión para el dispositivo en la dirección 18!\n");
        exit(0);
    }
    /*Limpiar instrumento*/
    viClear(viESA);
    printf("Obtener imagen \n\n");
    viPrintf(viESA, "MMEM:STOR:SCR 'C:PICTURE.WMF'\n"); //Guardar archivo de imagen
    viQueryf(viESA, "*OPC?\n", "%i", &opc); //OPC - Operación completada?
    do{viScanf(viESA, "%i", &opc);}while(opc!=1);
    viPrintf(viESA, "MMEM:DATA? 'C:PICTURE.WMF'\n"); //Obtener archivo de imagen
    viReadAsync(viESA, (ViBuf)cResult, iSize, &lRetCount); //Leer bloque de datos
    printf("%s\n", cResult);
    viPrintf(viESA, "MMEM:DEL 'C:PICTURE.WMF'\n"); //Borrar archivo de imagen
    /*Remover cabecera - Extraer datos de imagen*/
    memcpy(dPictureArray, cResult+iHeaderBytes, (size_t)lNumberBytes);
    /*Salvar en archivo*/
    fPictureFile=fopen("C:\\PICTURE.WMF", "wb");
    fwrite(dPictureArray, 1, lNumberBytes+1, fPictureFile);
    fclose(fPictureFile);
    /*Cerrar sesión*/
    viClose(viESA);
    viClose(defaultRM);
    getch();
}
```

4.4. Control del sistema de registros de estado

El sistema de registros de estado es un modelo estándar de bytes de información sobre el estado de los instrumentos. SCPI establece en base a los mecanismos puntualizados en el estándar IEEE 488.2, una estructura estándar de registros de estado para instrumentos SCPI. La Figura 4.4 muestra la estructura completa de registros de estado implementada en un instrumento particular.

La relación entre el controlador de un sistema de instrumentación y los instrumentos, en la cual el controlador envía una orden de medida y espera la respuesta, es muy usada, pero no siempre conveniente. Algunos instrumentos tienden a desarrollar tiempos de medida largos, situación que promueve la necesidad de predisponer de una secuencia de medidas temporizadas, de forma tal, que el controlador no deba preocuparse del instrumento hasta que este finalice la medición. En estos casos es útil disponer de mecanismos como el control del sistema de registros de estado, que permitan a los instrumentos notificar al controlador, que se ha llegado a un estado determinado.

Un elemento sin precedente en el control de instrumentos es el monitoreo de errores generados como producto de algún tipo de falla en el dispositivo o bien de la manipulación del mismo. Este escenario, destaca una vez más la importancia del uso de la estructura de registros de estado.

El uso adecuado de la estructura de registros de estado de un instrumento específico implica un conocimiento pleno del conjunto de registros, eventos y circunstancias que hacen que una condición sea detectable.

El monitoreo de las condiciones de un instrumento, es hecho al más alto nivel usando los comandos comunes IEEE:

- *CLS (*clear status*) limpia el byte de estado, así como la cola de eventos/errores y todos los registros de eventos.
- *ESE,*ESE? (*event status enable*) asigna y pregunta los bits en el registro de habilitación del registro de eventos estándar.
- *ESR? (*event status register*) pregunta y limpia el registro de eventos estándar.
- *OPC (*operation complete*) coloca el bit 0 en el registro de eventos estándar cuando todas las operaciones son completadas.
- *SER,*SER? (*service request enable*) coloca y pregunta el valor del registro de habilitación de solicitud de servicio.
- *STB? (*status byte*) pregunta el valor del registro de byte de estado son borrar su contenido

Existen dos métodos usados de manera programática en la detección y manejo de cambios de estado y condiciones de error en un instrumento. Ambos métodos permiten monitorear una o más condiciones. Estos métodos son el método de sondeo y de solicitud de servicio.

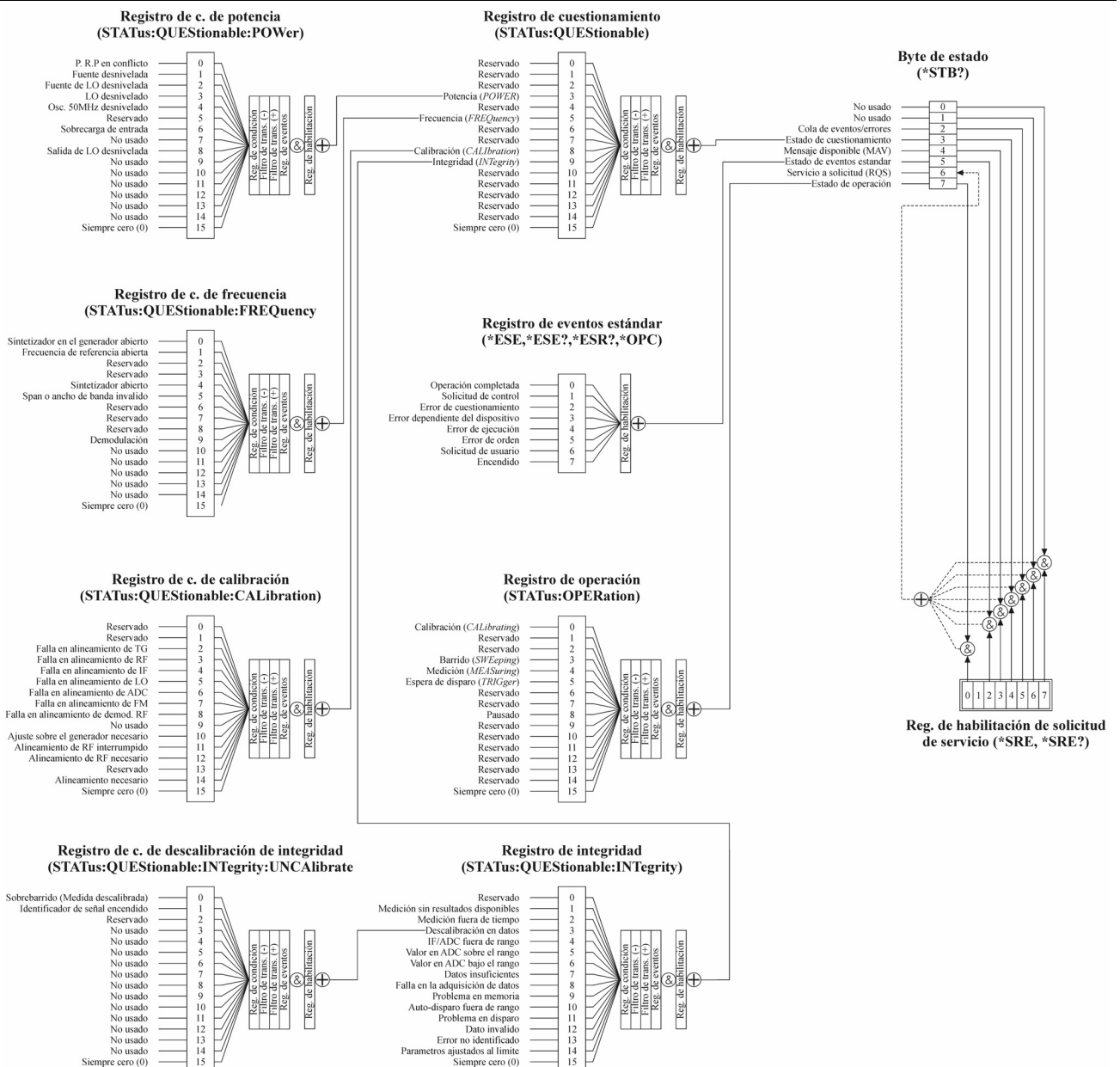


Figura 4.4. Sistema de registros de estados del analizador de espectros ESA-L E4403B.

4.4.1. Método de sondeo

Mediante este método, el instrumento asume un rol pasivo. Así, éste únicamente informara sobre los cambios sucedidos en sus condiciones cuando el controlador realice los cuestionamientos adecuados. Este método es adecuado en aplicaciones donde no es necesario conocer los cambios en el estado del instrumento en el momento mismo en estos ocurran. La utilidad de este método se incrementa cuando:

- El lenguaje de programación o entorno de desarrollo no soporta interrupciones SRQ.
- La aplicación desarrollada no requiere conocer el estado del instrumento de manera permanente.

Bajo este método se pueden denotar dos procedimientos para el sondeo de una condición. El primero de gran simplicidad sólo hace alusión a la lectura directa de los registros de condición, cuyo

valor representa el estado actual del registro implicado. Otro procedimiento, mucho más estructurado sobrelleva las actividades siguientes:

1. Establecer los bits de los registros del sistema de registros de estado del instrumento, que proporcionan el reporte de la condición que se desea conocer.
2. Habilitar los bits de los registros de habilitación de eventos que permitan reportar la condición deseada, mediante los bits de resumen correspondientes.
3. Opcionalmente, configurar los filtros de transición positivos o negativos, con el fin de controlar la condición de cambio en los registros.
4. Leer el byte de estado, con el fin de determinar los registros que han sido modificados. Realizar la lectura de los registros de eventos pertinentes.

4.4.2. Método de solicitud de servicio

El método de solicitud de servicio (SRQ, *Service Request*) permite la comunicación oportuna del estado del instrumento sin la intervención continua del controlador. Este método sitúa al dispositivo en un rol activo. El instrumento advertirá al controlador cuando una condición haya cambiado sin que este lo pregunte. El uso de este método es recomendable cuando:

- El tiempo para la notificación de cambios es crítico.
- Se requiere del monitoreo de uno más dispositivos con soporte de SRQ.
- La aplicación o controlador requiere efectuar alguna acción mientras el instrumento realiza una medición.

La primera consigna para la implementación de este método es, el soporte tanto entorno de desarrollo como del bus usado, de las interrupciones SRQ (por ejemplo C y C++ con GPIB). El monitoreo de una condición mediante el método SRQ, conlleva el siguiente procedimiento:

1. Determinar los bits que monitorean la condición.
2. Establecer como estos bits son reportados al bit 6 de servicio a solicitud (RQS, *Request Service*) del byte de estado. Mediante el estudio del sistema de registros de estado del instrumento, contemplado en un diagrama de jerarquía.
3. Habilitar los bits de los registros de habilitación de eventos.
4. Opcionalmente, configurar los filtros de transición positivos o negativos.
5. Habilitar los bits del registro de habilitación de solicitud de servicio adecuados.

Cuando una condición cambia, el dispositivo pone el bit RQS y la línea SRQ del bus. La controlador deberá determinar como responder a un SRQ, y realizar la lectura del byte de estado y los registros de eventos convenientes.

4.4.2.1. Generación de una solicitud de servicio

La Figura 4.4 muestra la estructura de un sistema de registros de estado de un instrumento que considera la implementación del método de SRQ. Como adición al sistema básico establecido por SCPI, un registro de habilitación de solicitud de servicio (SRER, *Service Request Enable Register*), es incluido en este. El SRER permite seleccionar que bits del byte de estado darán motivo de una solicitud de servicio. Este registro es concebido en el estándar 488.2. El manejo del SRER que faculta la aplicación de este método es realizada mediante las ordenes comunes IEEE 488.2 *SER y *SER?.

La forma en como se propagan los datos a través de esta jerarquía de registros es la misma que la considerada para una estructura SCPI mínima (inciso 3.5), sin embargo la adición del SRER implica algunas consideraciones adicionales. Una vez que el proceso usual de difusión de datos a través de los registros haya permitido colocar los bits de resumen del byte de estado, estos son lógica-

mente multiplicados por el contenido del SRER, este resultado a su vez, es lógicamente sumado para producir la habilitación del bit de servicio a solicitud (RQS, *Request Service*) en el registro de byte de estado.

Cuando un registro causa un bit de resumen en el byte de estado cambiando de un estado de falso (0) a verdadero (1), el dispositivo inicializa un proceso de solicitud de servicio. Sin embargo, este sólo dará comienzo si se cumplen las siguientes condiciones:

- El bit correspondiente del SRER se encuentra puesto (valor de 1).
- El instrumento no tiene una solicitud de servicio pendiente. Una solicitud es considerada pendiente entre el tiempo en que el dispositivo inicia un proceso SRQ, y el tiempo en que el controlador lee el estado del byte de estado.

El proceso SRQ habilita la línea SRQ del bus y el bit RQS del byte de estado. Ambas acciones son necesarias para informa al controlador que dispositivo requiere de servicio. La activación de la línea SQR sólo informa al controlador que algún dispositivo en el bus solicita servicio. Mientras que el bit RQS permite determinar que dispositivo demanda este servicio.

La Figura 4.5 ejemplifica el uso del sistema de registro de estado mostrada en la Figura 4.4. La aplicación realizada sobre Agilent VEE emula un método de solicitud de servicio para el monitoreo de las condiciones de la estructura de registros. Así como en otros entornos de desarrollo, el modo en como son tratados los objetos dentro de VEE no posibilita la inclusión de rutinas de interrupción por hardware, por lo que la implementación de un método de solicitud de servicio sólo es posible a un nivel de emulación. El procedimiento en cuestión, espera la generación de una solicitud de servicio en base a la configuración de registros, establecida inicialmente. Una vez que el evento ha ocurrido se hace la lectura del byte de estado para una posterior toma de decisiones.

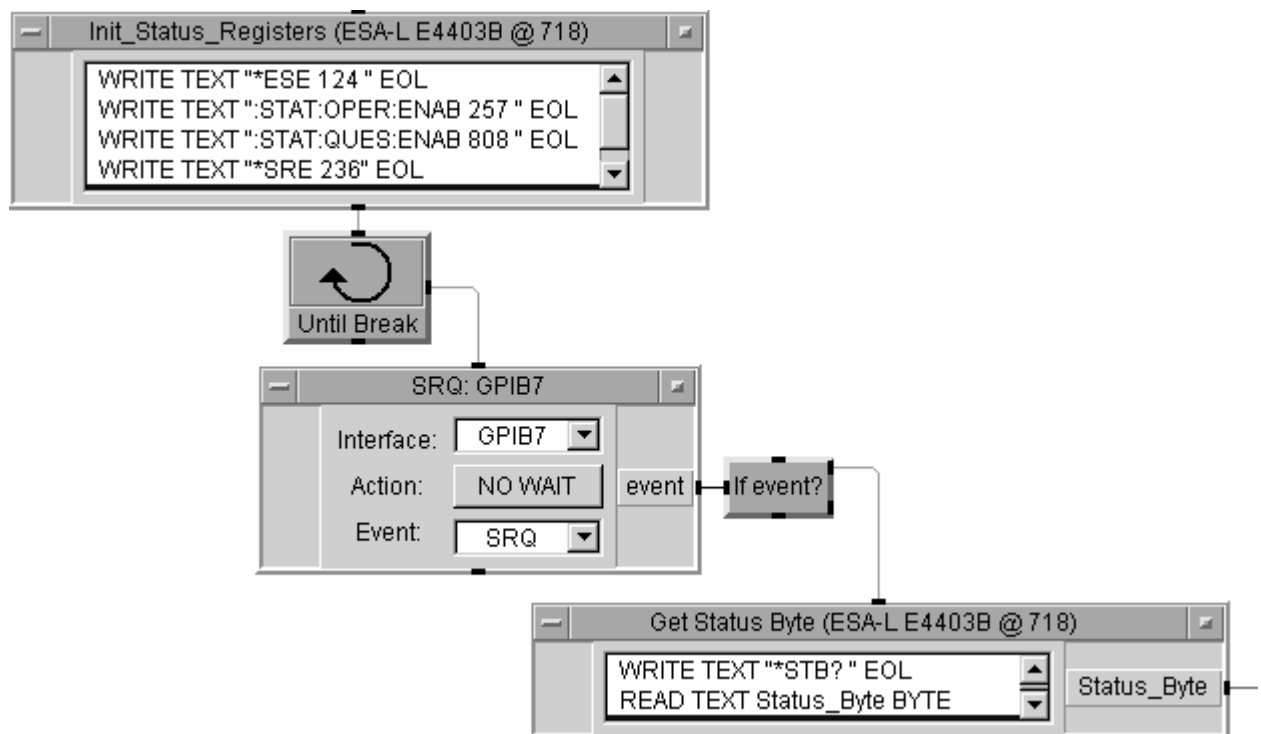


Figura 4.5. Uso de un sistema de registros de estado en una aplicación bajo Agilent VEE.

5. Herramienta de manejo del AS ESA-L 4403B

Los resultados esperados de cualquier proyecto de investigación aplicada involucran siempre el planteamiento de una tecnología aplicable para procesos o productos, sin embargo es común que estos resultados sean acompañados de una fase de aplicación demostrativa.

Si bien, el objetivo principal del presente trabajo de tesis es el planteamiento de una serie de directrices que faciliten el desarrollo de aplicaciones relacionada con el control de instrumentos mediante SCPI, su definición obliga a una demostración de su utilidad, es decir, como estas directrices son usadas en una problemática real. Es así, como atendiendo a esta demanda, se ha planteado el desarrollo de una herramienta software para el control del Analizador de Espectros ESA-L E4403B de la firma Agilent. La elección de este instrumento particular, permitirá mostrar todos los elementos contemplados por las directrices en cuestión. La aplicación mencionada será referida en la documentación siguiente por las siglas HeMAE (*Herramienta para el Manejo del Analizador de Espectros*) ESA-L E4403B.

La resolución de problemas reales de software, requiere de la incorporación de una estrategia de desarrollo que acompañe al proceso. Esta estrategia a menudo es llamada modelo ó paradigma de ingeniería del software. La selección de un modelo se lleva a cabo de acuerdo al tipo de proyecto y aplicación, los métodos y las herramientas a utilizarse, y los controles y entregas que se requieren [30]. La naturaleza demostrativa a la que se sujeta el software a desarrollar, sugiere la elección de una estrategia de desarrollo incremental.

Como lo muestra la Figura 5.1, un modelo incremental aplica secuencias lineales conforme progresa el tiempo. Cada secuencia lineal produce un incremento y por tanto una versión parcial pero utilizable del software. En el caso particular del software tratado, cada incremento permitirá exhibir el uso de una directriz particular.

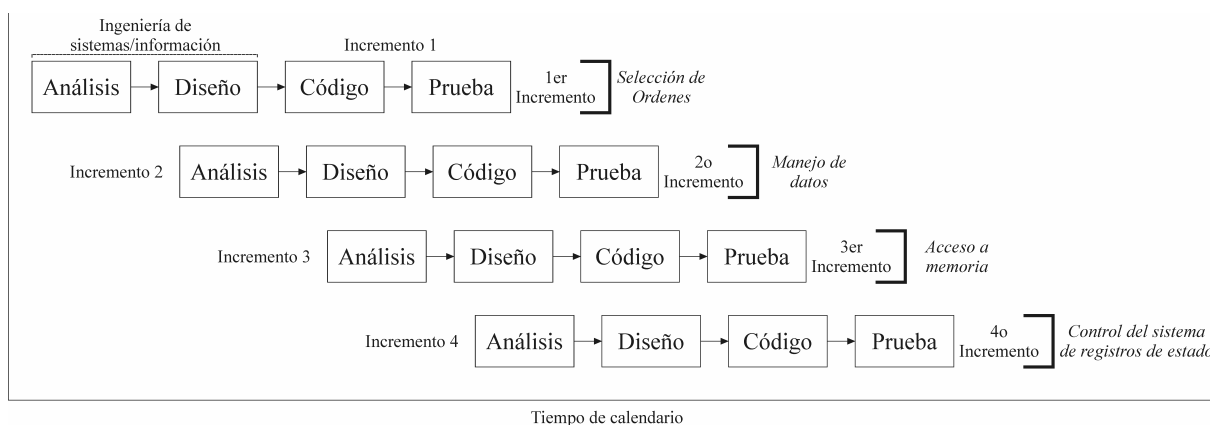


Figura 5.1. Estrategia de desarrollo incremental.

Independiente del modelo de software aplicado, fases como la especificación, el diseño, la implementación y la validación son habituales en todos los modelos. Las secciones siguientes describen las fases involucradas en el desarrollo de la herramienta software HeMAE ESA-L E4403B.

5.1. Especificación del software

La especificación del software permite establecer los servicios que se requieren del sistema y las restricciones de operación y desarrollo del mismo. Esta actividad es a menudo llamada ingeniería de requerimientos [39].

Al enfocarse en la definición de lo que se desea producir, la ingeniería de requerimientos cumple un papel primordial en el proceso de producción de software. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, y en forma consistente y compacta, el comportamiento del sistema; minimizando de esta manera los problemas relacionados al desarrollo de los mismos.

El proceso de ingeniería de requerimientos conduce a la producción de un documento de requerimientos, que representa la especificación del sistema. Lo requerimientos en este documento son presentados normalmente en dos niveles de detalle, el dirigido a los usuarios finales y el enfocado a los desarrolladores del sistema. Las fases más relevantes en el proceso de la ingeniería de requerimientos son el estudio de factibilidad del sistema, la obtención y análisis de requerimientos, y la especificación del software.

5.1.1. Estudio de factibilidad

La primera actividad que conlleva la creación de un software es un estudio de factibilidad del mismo. La factibilidad de un proyecto se refiere a la disponibilidad de los recursos necesarios para llevar a cabo los objetivos señalados. Dicho estudio permite estimar de alguna manera si las necesidades del usuario, pueden ser satisfechas con las tecnologías actuales de software y hardware.

5.1.1.1. Descripción de alternativas

El software de aplicación para el control de instrumentos existente en el mercado se caracteriza por mostrar una interfaz de usuario delimitada por un motivo de estudio. Este software es provisto habitualmente por el fabricante del instrumento, tal es el caso de Agilent BenchLink, Agilent Web Remote Control y Agilent IntuiLink, para analizadores de espectros de la firma Agilent.

5.1.1.1.1. Agilent BenchLink para Analizadores de Espectros

Proporciona un enlace de comunicaciones entre una PC y algunos analizadores de espectros. Algunas de sus principales utilidades que este software presenta son [1]:

- Soporte para instrumentos: serie 8560, 8590, ESA-E/L, E7400, CaLan 2010/3010, serie EM 8590 y ESA-L1500A. Soporte de interfaz GPIB y RS-232.
- Transferencia de imágenes de pantalla y datos de trazas del analizador a la PC.
- Captura y restauración de algunas condiciones de medida del instrumento.
- Uso de marcadores X/Y y ampliaciones por zonas para el análisis de trazas.

5.1.1.1.2. Agilent Web Remote Control para AE de las series PSA, ESA-E/L y EMC

Permite el control de manera local y remota de analizadores de espectros de las series PSA, ESA-E/L, y analizadores EMC de la serie E7400A. El instrumento es conectado vía GPIB o LAN a una PC que actúa como servidor local, y la cual se encontrara ligada a Internet. Los usuarios clientes tendrán acceso al analizador mediante una computadora conectada a Internet usando un navegador Web estándar. Dentro de las principales características funcionales que este software presenta se encuentran [19]:

- Control directo sobre funciones del analizador de espectros tales como: frecuencia central, de inicio, de paro y rango (*span*), nivel de referencia, atenuación, resolución y ancho de banda video, promedios, tiempo y número de puntos de barrido, reinicio del instrumento, marcadores, detectores y auto-acoplamiento.
- Monitoreo de la actividad del instrumento mediante pantallas estándares, acumulativas, espectrogramas, etc.
- Captura de imágenes de pantalla y datos de trazas del analizador.
- Múltiples usuarios permitidos, en razón al servidor Apache asociado durante su instalación.
- Marcadores, líneas de referencia y ampliaciones por zona para la grafica en pantalla.
- Envío de comandos estándares SCPI basados en ASCII desde el cliente remoto al analizador de espectros o desde otros instrumentos conectados al servidor.

5.1.1.1.3. Agilent IntuiLink para AE de las series PSA, ESA-E/L y EMC

Brinda un conjunto de herramientas de conectividad que permiten fácil y rápidamente transferir datos desde un analizador de espectros de la serie PSA, ESA-E/L y EMC a una PC. El software esta diseñado para proveer un alto control de instrumentos usando aplicaciones comunes como Microsoft Excel y Word. Las utilidades ofrecidas son dirigidas tanta a usuarios programadores, como para aquellos que no requieran compromisos. Dos singularidades destacan en este software [2]:

- Una barra de herramientas de fácil uso que hace posible la conexión con el analizador de espectros desde una aplicación como Microsoft Excel[®] o Word[®]. Una vez realizada una conexión se podrá obtener y salvar datos e imágenes de la pantalla del analizador, capturar y restaurar condiciones de medida, así como realizar mediciones repetitivas.
- Un servidor de automatización *ActiveX* que facilita la tarea de programación bajo el uso de un entorno como Visual Basic, VEE, LabVIEW, etc. El servidor provee funciones para el control del instrumento y descarga de datos e imágenes de pantalla.

5.1.1.2. *Factibilidad*

Los elementos tratados en un estudio de factibilidad se encuentran ligados directamente con la definición del sistema en cuestión, sin embargo aspectos como el técnico, operativo y económico son comunes en cualquier estudio de factibilidad.

La factibilidad técnica obedece a un análisis tecnológico de todos aquellos factores que justificaran la mejor combinación de elementos que determinar la viabilidad del proyecto. La factibilidad operativa permite la evaluación del impacto del proyecto sobre la organización. El estudio de la factibilidad económica del sistema permite conocer la totalidad de gastos en los se incurrirán, este ultimo estudio es omitido en proyectos de investigación donde el impacto monetario no es un factor.

5.1.1.2.1. Factibilidad operativa

Las compañías dedicadas a la construcción de hardware y/o software de cualquier tipo buscan proveer un entorno confortable y eficaz que les permita una productividad más alta del uso y por tanto un mayor tiempo de vida del producto [32]. La capacidad de aprendizaje, eficiencia y satisfacción son algunas de las directrices de evaluación que los usuarios siguen cuando se enfrentan a cualquier tipo de interfaz sea de software o hardware [30, 38].

Los desafíos que presentan los sistemas automatizados de medida a la industria del software de este tipo, hacen necesario el desarrollo de aplicaciones específicas, que ofrezcan un alto grado de flexibilidad que lleve al usuario a determinar por si mismo las tareas a realizar y los resultados que desea visualizar ó almacenar [25].

De acuerdo a lo anterior, y en base al estudio de las alternativas de software existentes se establece un factor de validez para la solución planteada. La ventaja comparativa que el software propuesto presenta en el área de estudio, se base en el análisis de las virtudes y carencias de las alternativas existentes. El software Agilent BenchLink Web Remote Control representa una buena solución a la problemática en cuestión, sin embargo su rigidez y limitaciones en cuanto al control de funciones del instrumento minimizan sus atributos.

El enfoque ilustrativo bajo el cual es concebida la aplicación, da el atributo de utilidad, necesario para su implementación. La libertad que proporcione el software para la elaboración de tareas, no dará lugar a una infactibilidad operativa por resistencia del usuario final.

5.1.1.2.2. Factibilidad técnica

La tecnología para la solución propuesta se encuentra actualmente disponible. Será el lenguaje de programación grafica VEE 6.01 para Windows, que se muestra dentro del software para el desarrollo de este tipo de aplicación como una de las mejores opciones, el ambiente de desarrollo elegido para la implementación de la herramienta software propuesta.

El equipo mínimo a emplear para el desarrollo y prueba del sistema, será el siguiente:

- PC con procesador Pentium II a 500MHz o superior, 64 Mb de memoria RAM, disco duro de 1Gb, sistema operativo Windows 9x/Me/NT/2000/XP e interfaz GPIB.
- Analizador de espectros ESA-L E4403B de la firma Agilent Technologies, interfaz GPIB (opción 4H) presente.

Siendo el control del instrumento el factor más incidente en una aplicación de instrumentación programable, las pruebas serán realizadas a diferentes escalas de integración, de manera tal que se pueda garantizar en un porcentaje significativo, el buen funcionamiento de la herramienta software desarrollada.

La disposición del ambiente de desarrollo seleccionado, ha sido probada en aplicaciones de este género. La amplia difusión sobre su uso, no permite estimar dificultades de este tipo. En lo que respecta al conocimiento técnico necesario por parte del desarrollador, las variables que inciden en el desarrollo se encuentran dentro del mismo dominio de la instrumentación electrónica, por lo que se asume su comprensión.

La incorporación de nueva tecnología de software al usuario final se encuentra ligada al uso de la herramienta. La totalidad de usuarios finales potenciales solo requerirán del soporte de la aplicación VEE o VEE Run Time y del equipo mencionado, para el uso de la mencionada herramienta software.

5.1.2. Especificación de requerimientos

El documento de requerimientos del software o simplemente especificación de requerimientos es una declaración oficial, de qué es lo que requieren los desarrolladores del sistema. Incluye tanto los requerimientos del usuario para el sistema como una especificación detallada de los requerimientos del sistema. Una gran variedad de organizaciones han definido estándares para los documentos de requerimientos, los cuales en su mayoría tratan de manera progresiva el desarrollo de dicho documento, pasando de la obtención de los requerimientos del usuario a un bosquejo preliminar de la aplicación y una definición de requerimientos ahora del sistema. Algunas veces, un apartado de análisis de requerimientos es incluido como conclusión a este documento y preámbulo a una fase de diseño. Sin embargo, la información que se incluye en un documento de requerimientos debe depender del tipo de software a desarrollar y del enfoque de desarrollo que se utilice.

5.1.2.1. *Requerimientos de usuario*

1. *Facilidad de manejo*: la aplicación deberá proporcionar un entorno familiar que permita al usuario un fácil aprendizaje. Los resultados producidos por el software deberán ser predecibles al usuario.
2. *Manipulación de funciones primarias y mediciones particulares del instrumento*: los mecanismos de interfaz a suministrar permitirán al usuario operar sobre las funciones que el dispositivo presenta. El usuario seleccionará de forma indistinta los parámetros del instrumento sobre los que desea hacer modificaciones.
3. *Presentación de trazas en pantalla*: las trazas de datos presentes en el instrumento deberán ser exhibidas por la aplicación, en diferentes formatos, adaptados de acuerdo al tipo de medición realizada y de las necesidades del usuario. Utilidades sobre el manejo en pantalla también deberán ser incluidas.
4. *Obtención de datos e imágenes del instrumento*: los datos, parámetros e imágenes ofrecidas por el instrumento deberán encontrarse disponibles al usuario para su manejo directo en esta u otra aplicación. Se deberán proveer utilidades que permitan mostrar la información generada, en aplicaciones como Microsoft Excel[®] y Word[®], en un formato de reporte conveniente.
5. *Monitoreo de condiciones de error y eventos en el instrumento*: el estado en que el dispositivo se encuentre deberá ser verificado de manera que permita al usuario conocer la situación en la cual se encuentre el instrumento. Una unidad de interfaz informativa deberá ser implementada para cumplir con este propósito.

5.1.2.2. *Requerimientos del sistema*

1. *Usabilidad*: los mecanismos usados por la aplicación buscarán potencializar el uso de la herramienta. Los criterios de desarrollo empleados buscarán la facilidad de aprendizaje, la eficiencia en el uso, la facilidad para recordar y la libertad de errores del software.
2. *Interfaz de usuario*: se emplearán una serie de metáforas visuales que proporcionen al usuario un alto grado de consistencia aplicación-instrumento, sin buscar crear discrepancias sobre el modelo mental que el usuario tenga del instrumento.
3. *Funciones del instrumento*: los elementos de interacción con el usuario que faciliten el control de las funciones provistas por el instrumento serán agrupados siguiendo una estrategia de descubrimiento progresivo que permita definir una jerarquía entre argumentos. Los mecanismos de interfaz incluirán paneles de control, gráficos, ventanas de información, barras de herramientas y botones de control general.
4. *Manejo de datos*: el sistema incluirán el manejo de trazas de datos, archivos de imágenes, mediciones y argumentos para las funciones del instrumento. Los métodos usados para la obtención y presentación de los datos obedecerá al tipo del dato en cuestión.
5. *Control sobre estado del instrumento*: el monitoreo de las condiciones de errores y eventos del instrumento requerirá de la manipulación del sistema de registros de estado del dispositivo. Se implementarán una serie de rutinas que permitan visualizar el estado actual del instrumento. Como un medio de prevención de errores, se llevará a cabo una validación anticipada de los datos enviados al dispositivo.
6. *Conectividad*: se proveerán una serie de medios que faciliten la conectividad de la herramienta con otras aplicaciones tales como Microsoft Excel[®] y Word[®]. La generación de reportes será el principal objetivo de estos medios.

5.2. Diseño

Un diseño de software es una descripción de la estructura del software que se va a implementar, los datos que son parte del sistema, las interfaces entre los componentes del sistema y algunas veces, los algoritmos utilizados. El diseño es la única manera de materializar con precisión los requerimientos del cliente [30].

Una vez que se analizan y especifican los requerimientos del software, el diseño de software es la primera actividad técnica que se requiere para la construcción y verificación del software. Los requisitos del software, manifestados por los modelos de datos, funcionales y de comportamiento producidos durante su análisis, alimentan la tarea del diseño, proporcionando la información necesaria para crear los modelos de diseño que se requieren para una especificación completa del diseño. El diseño de datos, arquitectónico, a nivel de componentes y de interfaz. Es una práctica común, englobar el diseño de datos y arquitectónico, en una sola escala de integración llamada arquitectura del software.

El diseño de la herramienta software para el control de instrumentos en cuestión, es concedido bajo un panorama de análisis estructurado, consigna bajo la cual son descritos a continuación los diferentes niveles de la pirámide de diseño para la herramienta software tratada. Todos los esquemas producto del diseño de la arquitectura de software y de componentes han sido realizados mediante el uso del software para modelado de procesos BPwin versión 4.1.4.

5.2.1. Diseño de la arquitectura del software

La arquitectura del software es construida en base a dos niveles de la pirámide de diseño, el diseño de datos y arquitectónico. El diseño de datos facilita la representación de los componentes de datos de la arquitectura. El diseño arquitectónico se centra en la representación de la estructura de los componentes del software, sus propiedades e interacciones. Así, la obtención de una arquitectura de software se resume a la conversión de requisitos, modelos de datos y funcionales creados durante el análisis de requisitos, en un esquema que proporcione una visión global del sistema a construir.

La primera encomienda en la concepción de una arquitectura de software es la existencia de un modelo funcional del sistema, que provea un indicador de cómo se transforman los datos a medida que estos fluyen por el sistema. Dentro del análisis estructurado, una de las técnicas preferentemente usadas en la representación de modelos funcionales, son los diagramas de flujos de datos (DFD). A continuación se presentan el conjunto de DFDs, derivados del modelo funcional obtenido durante el análisis de los requerimientos de la herramienta software HeMAE ESA-L E4403B.

La Figura 5.2 muestra el DFD de nivel 0 o contextual para la HeMAE ESA-L E4403B. Las entidades externas principales que producen información para el sistema, son denotadas por un panel de control como medio de interacción con el usuario, así como por el instrumento mismo como una fuente de información. Las entidades de consumo de información son del tipo visual, de almacenamiento y control.

La Figura 5.3 muestra el DFD de nivel 1 para la HeMAE ESA-L E4403B, producto de la expansión del DFD de nivel contextual. Los seis procesos derivados permiten contemplar las principales tareas desarrolladas por la herramienta software, así como las dependencias existentes entre datos que acompañan a dichas tareas.

La Figura 5.4 muestra el DFD de nivel 2 para el proceso que gestiona la modificación de parámetros del instrumento, de acuerdo a las exigencias del usuario. El proceso es numerado como 2 en su DFD precedente.

La Figura 5.5 muestra el DFD de nivel 2 para el proceso que tramita la obtención de trazas de datos y mediciones. Los procedimientos de interacción con el instrumento no son destacados a este nivel de abstracción. El proceso es numerado como 3 en su DFD precedente.

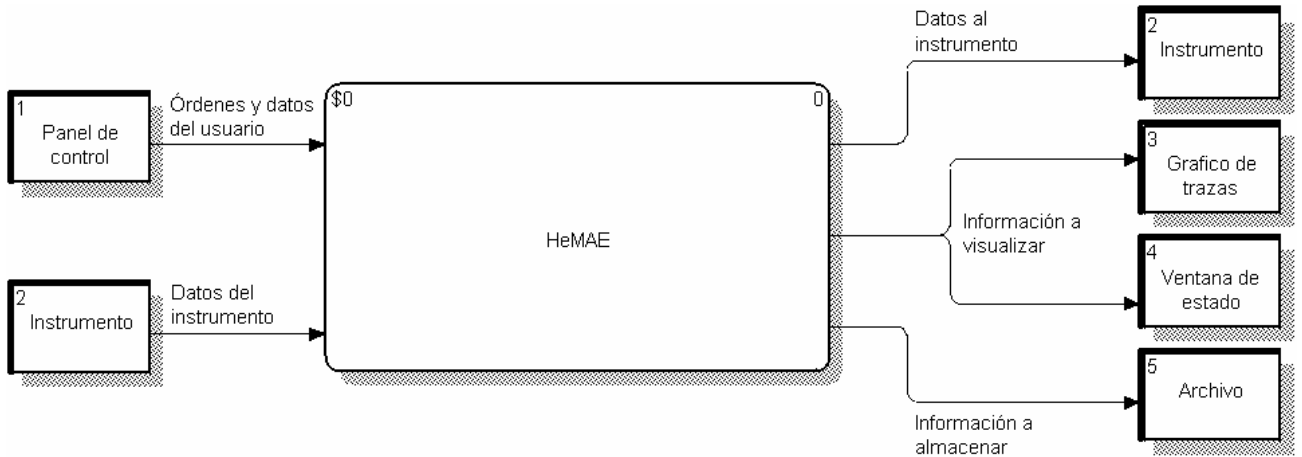


Figura 5.2. DFD de nivel contextual para la HeMAE ESA-L E4403B.

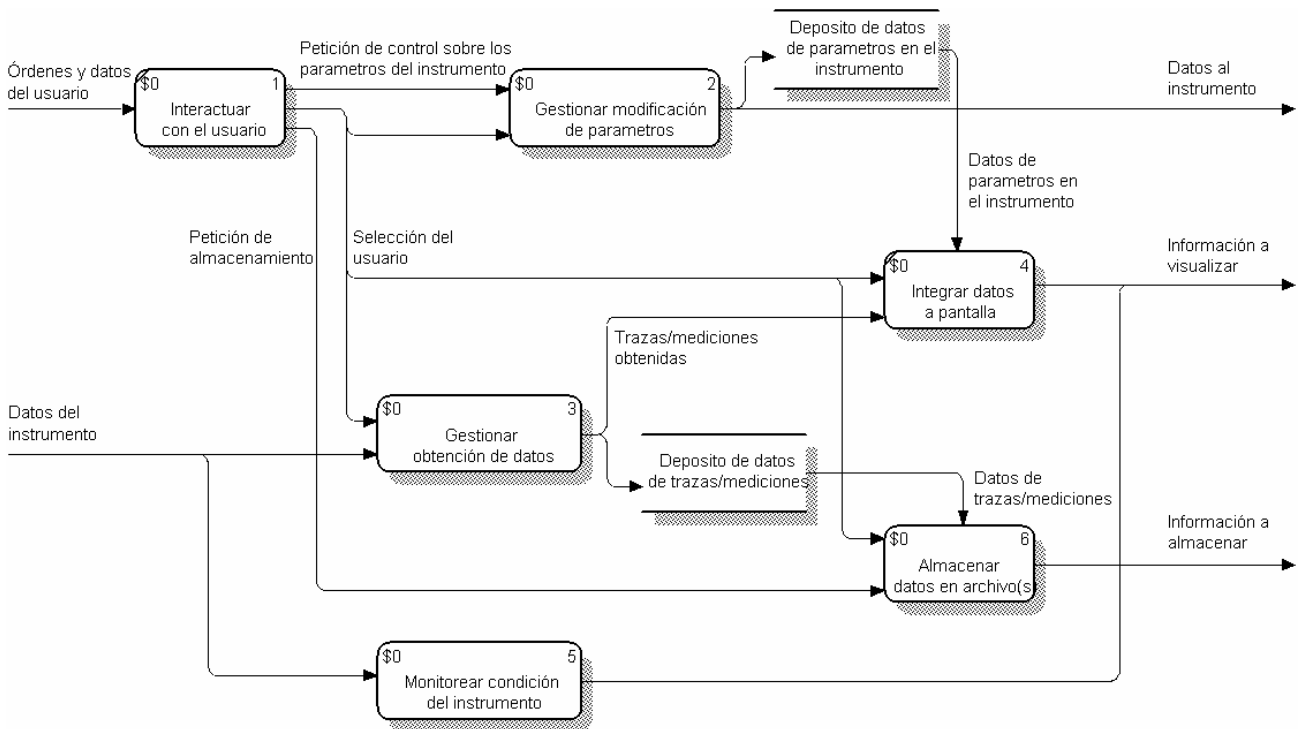


Figura 5.3. DFD de nivel 1 para la HeMAE ESA-L E4403B.

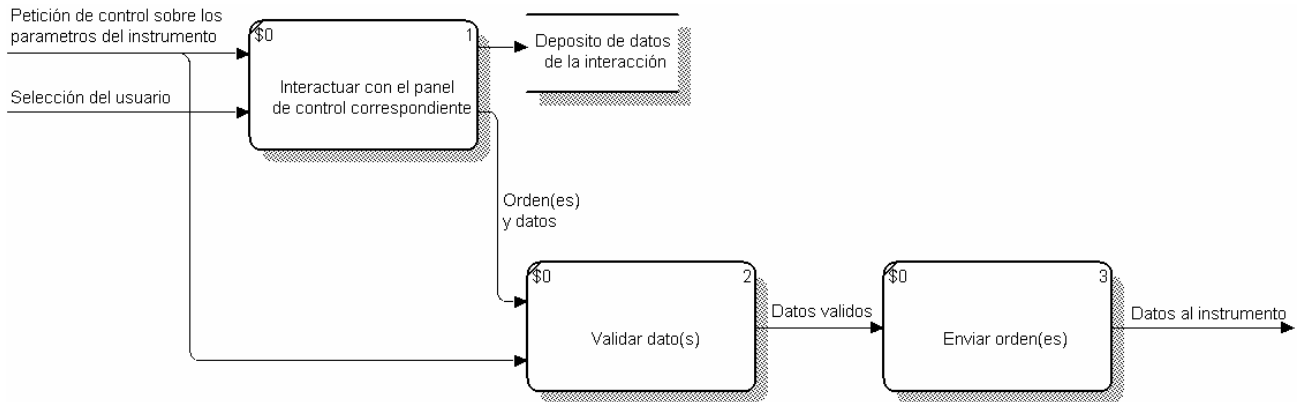


Figura 5.4. DFD de nivel 2 que refina el proceso *gestionar modificación de parámetros*.

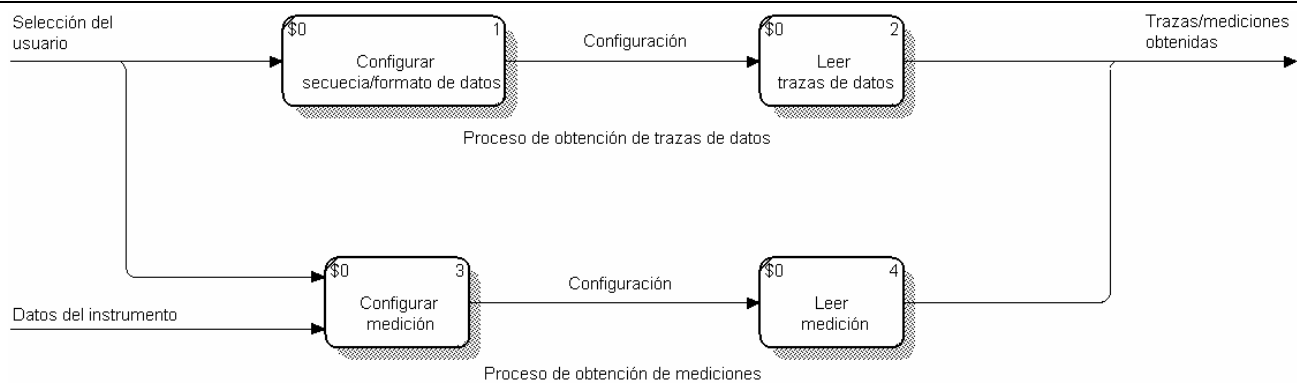


Figura 5.5. DFD de nivel 2 que refina el proceso *gestionar obtención de datos*.

La Figura 5.6 muestra el DFD de nivel 2 para el proceso que obra sobre la inspección del estado del instrumento. El subproceso de configuración del instrumento es tratado como una tarea inicial, independiente al proceso normal de monitoreo. El proceso es numerado como 5 en su DFD precedente.

La Figura 5.7 muestra el DFD de nivel 2 para el proceso que facilita el almacenamiento de datos en archivo. El subproceso de construcción de archivos de hojas de cálculo y de texto es englobado dentro de un proceso general de generación de reportes, que implica otras fuentes de datos. Los procedimientos para la obtención de archivos de imagen son destacados a un alto nivel de abstracción. El proceso es numerado como 6 en su DFD precedente.

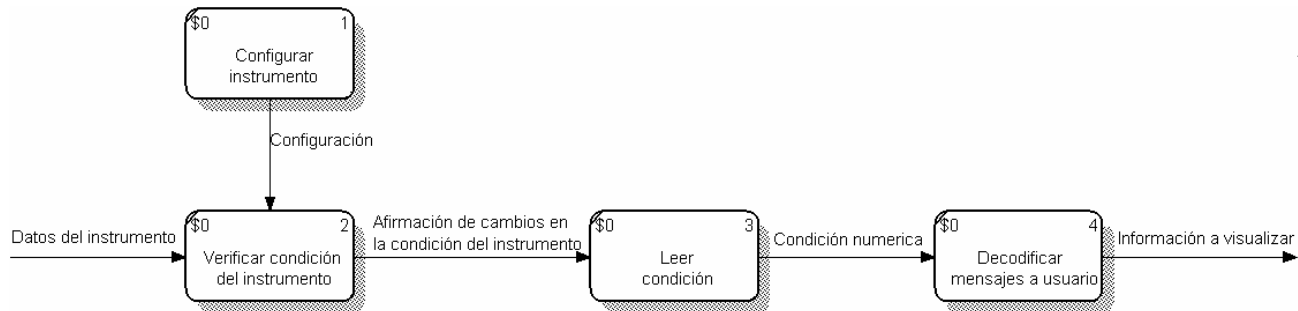


Figura 5.6. DFD de nivel 2 que refina el proceso *monitorear condición del instrumento*.

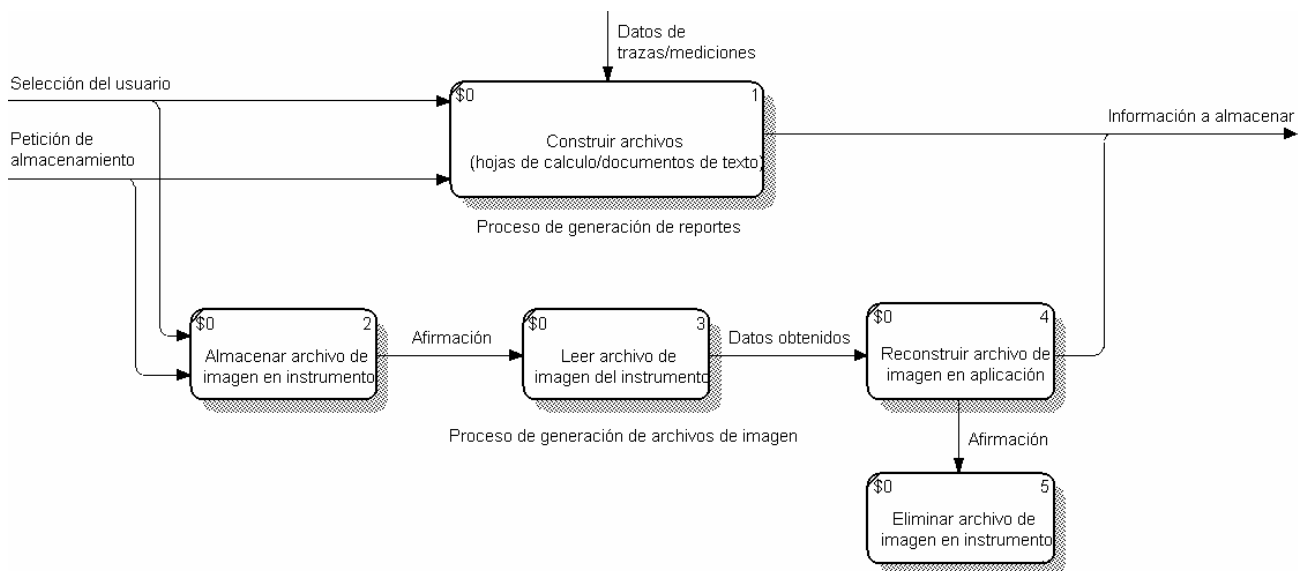


Figura 5.7. DFD de nivel 2 que refina el proceso *almacenar datos en archivo(s)*.

Una vez que se cuenta con un modelo funcional para el sistema, es posible en base a un estilo arquitectónico y método de diseño particular, el planteamiento de una arquitectura de software.

El diseño estructurado, caracterizado como un método de diseño orientado al flujo de datos permite la transición del flujo de información representado en DFDs, en una estructura de programa o arquitectura de software. El tipo de flujo de información determina los procedimientos usados por el método de conversión empleado.

La herramienta de software HeMAE ESA-L E4403B ha sido ideada siguiendo un estilo arquitectónico de llamada y retorno. La Figura 5.8 muestra la estructura del software generada, en base a un análisis de transformaciones, para la conversión de los DFDs mostrados.

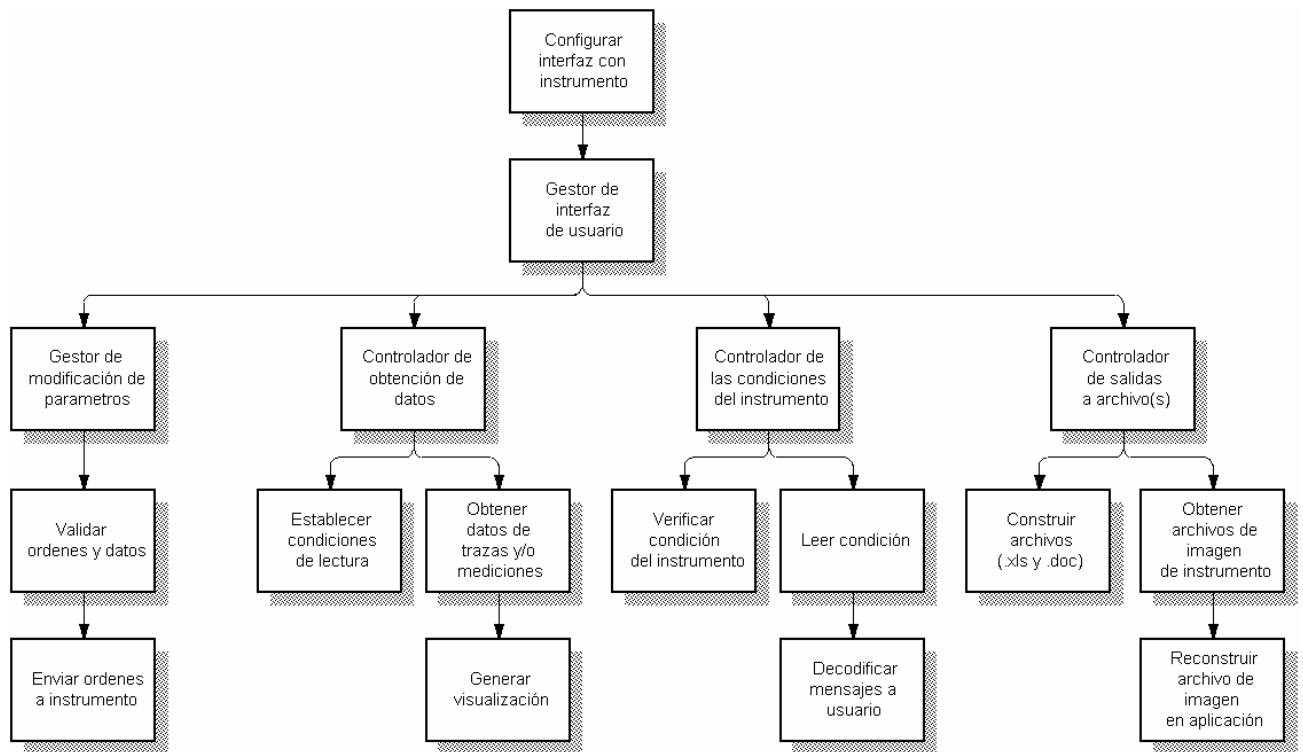


Figura 5.8. Primera iteración de la estructura del software HeMAE ESA-L E4403B.

5.2.2. Diseño de la interfaz

Un diseño integral de la interfaz describe la comunicación del software consigo mismo, con los sistemas que operan junto con el, y con los usuarios que lo emplean, sin embargo algunos de estos elementos son normalmente absorbidos por áreas como el diseño arquitectónico y de datos, dejando al diseño de la interfaz ser relacionado sólo con el trato de la interfaz grafica de usuario (GUI, *Graphical User Interfaz*).

El objetivo del diseño de la interfaz grafica de usuario es definir un conjunto de objetos y acciones de interfaz (y sus representaciones en la pantalla) que posibiliten al usuario llevar a cabo todas las tareas definidas, de forma que cumplan con los objetivos de usabilidad definidos por el sistema. La noble meta de un diseñador de interfaces es proveer una confortable y eficiente entorno virtual en que los usuarios puedan laborar.

Dentro del proceso de diseño de GUIs se destacan dos pasos primordiales, el análisis de tareas y el diseño grafico de la interfaz. El análisis de tareas es usado como un mecanismo para refinar las labores necesarias para que el software lleve acabo la función deseada. El diseño grafico refleja el modelo de interacción utilizado por la GUI, con el fin de alcanzar un alto grado de eficiencia [32].

La Tabla 5.1 resume los elementos obtenidos del análisis de tareas mediante casos de uso, para la herramienta software HeMAE. La ya reiterada naturaleza demostrativa que acompaña a este software da motivo a la selección de las tareas y medios expuestos por este análisis.

Tabla 5.1. Análisis de tareas para el diseño de la GUI para la HeMAE ESA-L E4403B.

Tarea	Objetos/acciones derivados	Descripción
Control de parámetros del instrumento	Selector y paneles de control. Elemento de reinicio de parámetros del instrumento.	La modificación y visualización de parámetros en el instrumento se llevara a cabo mediante el uso de paneles de control interactivos. El elemento de selección contemplado, permitirá acceder a los distintos paneles de control, cada panel cubrirá una sección de parámetros del instrumento de acuerdo a su función. Se implementaran dos tipos de paneles, los que cubrirán los parámetros globales del instrumento, y los que permitan el acceso a la configuración de mediciones particulares.
Obtención de trazas de datos y mediciones del instrumento	Pantalla de trazas y mediciones.	La obtención y exposición de trazas de datos y mediciones del instrumento, requerirá del empleo de un grafico de datos, el cual será acompañado por diversos indicadores que harán referencia tanto a parámetros en el instrumento, como a valores de las mediciones realizadas. La presentación de la pantalla variará de acuerdo con el tipo de medición solicitada.
Almacenamiento de información	Objetos (barra de herramientas).	Se proveerá de una serie de herramientas, que permitan, el almacenamiento de la información referente a parámetros, trazas y mediciones del instrumento, en diferentes formatos de archivo, hojas de cálculo, documentos de texto e imágenes.
Monitoreo del estado del instrumento	Ventana de estado.	El estado en que se encuentre el dispositivo será expuesto mediante un objeto de inspección. Se mostraran al usuario de manera clara las razones de la condición generada, transparentando los mecanismos de operación usados.

El diseño grafico de la interfaz conlleva el desarrollo de dos argumentos, el trazado efectivo y las consideraciones de color/fuente en el diseño. La Tabla 5.2 resume las particularidades planteadas para el diseño grafico de la interfaz. La Figura 5.9 muestra el trazado de la interfaz, destacando la implementación de los objetos generados por el análisis de tareas previo. La Figura 5.10 presenta la aplicación de las consideraciones de trazado y color en un prototipo para la interfaz a desarrollar.

Tabla 5.2. Especificaciones del diseño grafico de la GUI para la HeMAE ESA-L E4403B.

Diseño	Especificación	Concepto	Descripción
Trazado	Jerarquía visual	Número de tareas cubiertas	4
		Guía de estilo utilizado	Microsoft Windows
		Número máximo de descubrimientos progresivos	3
	Balance	Parámetro de búsqueda visual	Izquierda a derecha
		Arquetipo	Asimétrico
		Densidad de espacios aproximado	45%
	Proporción de elementos visuales	1:1.414 a 1:1.732	

Diseño	Especificación	Concepto	Descripción
Color/Fuente	Consistencia visual	Metáfora visual	AE ESA-L 4403B
		Cuadrícula de trazado entre paneles	2x8
	Consideraciones de color	Ajuste a resolución mínima	800x600
		Modelo de colores primarios	RGB
Fuente	Fuente	Cantidad de colores implícitos	5
		Informativa	Arial Narrow, 11
		Descriptiva	Arial, 10-12

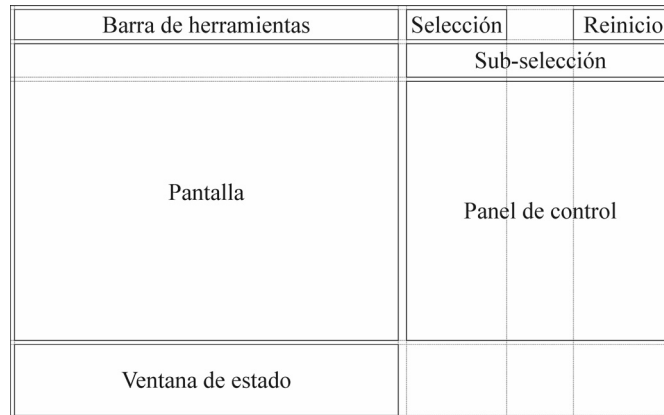


Figura 5.9. Trazado para la GUI de la HeMAE ESA-L E4403B.

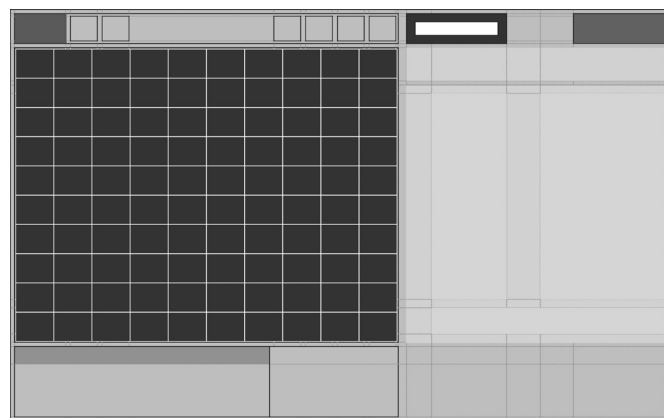


Figura 5.10. Prototipo para la GUI de la HeMAE ESA-L E4403B.

5.2.3. Diseño procedimental.

El diseño a nivel de componentes, llamado también diseño procedimental, tiene lugar posteriormente al establecimiento del diseño de datos, arquitectónico e interfaz. Su objetivo es convertir un modelo de comportamiento del sistema en un esquema de software operacional, que sea posible implementar. Una de las técnicas de diseño más usadas a nivel de componentes es la construcción de estructuras o programación estructurada. Los diagramas de flujo son una de los medios gráficos utilizados por excelencia en la representación de datos procedimentales.

La Figura 5.11 muestra el diagrama de transición de estados (DTE) del modelo de comportamiento desarrollado en la fase de análisis de requerimientos. La creación de este modelo permite representar el comportamiento del sistema, y dar pauta al diseño de componentes del software. La

Figura 5.12 esquematiza en base a un diagrama de flujo, el modelo operacional trazado para la herramienta software.

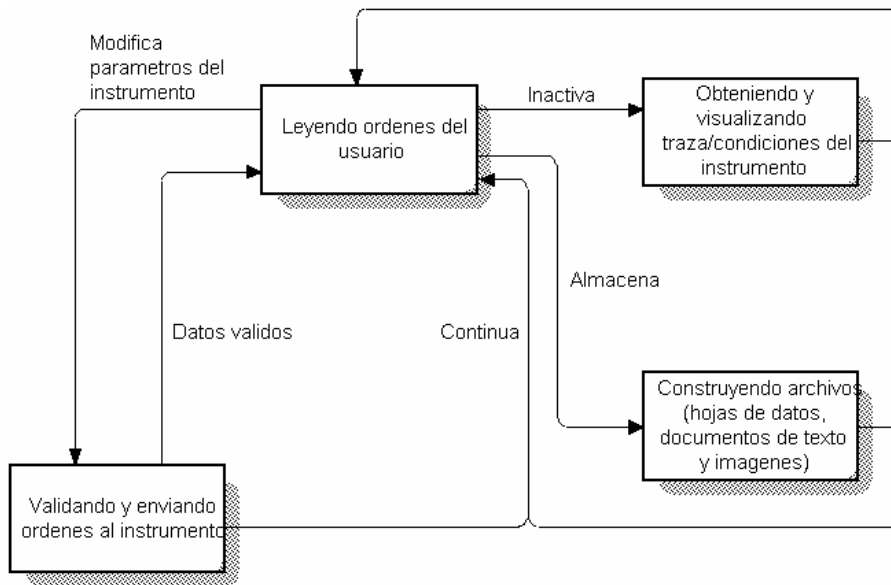


Figura 5.11. Diagrama de transición de estados para la HeMAE ESA-L E4403B.

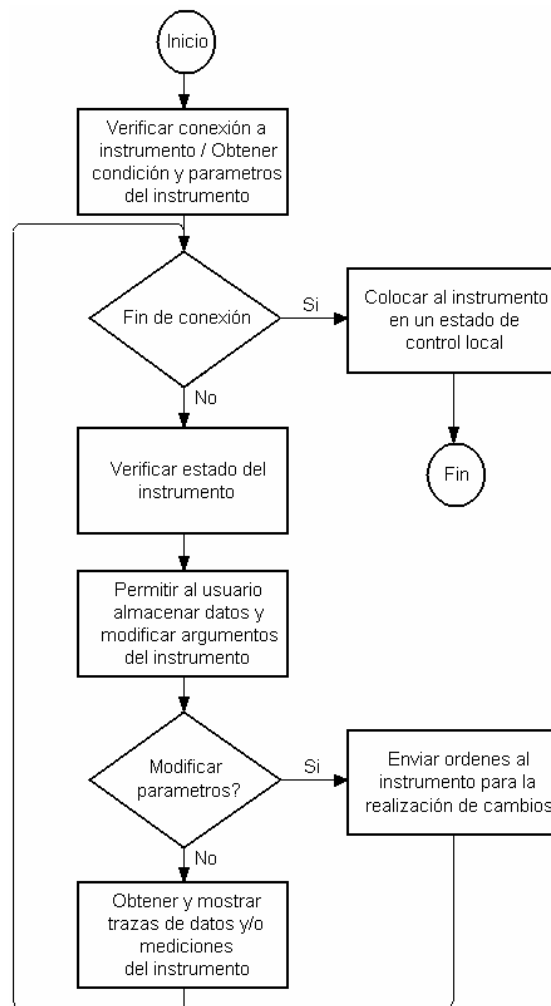


Figura 5.12. Diagrama de flujo para el modelo operacional de la HeMAE ESA-L E4403B.

5.3. Implementación

La etapa de implementación del software es el proceso de convertir una especificación en un sistema ejecutable. El desarrollo de un programa que permita implementar un sistema se sigue de forma natural al proceso de diseño. Es común que las primeras etapas de diseño y desarrollo de programas se encuentren entrelazadas.

Tal como se planteo de manera inicial, el objetivo del desarrollo de la herramienta software HeMAE ESA-L E4403B, es ilustrar el empleo de las directrices para el control de instrumentos trazadas durante el presente trabajo de tesis. Siendo entonces así, la etapa de implementación del software el escenario idóneo para la justificación práctica del uso de estas directrices. De manera previa, las fases de ingeniería de requerimiento y diseño han sido forjadas alrededor del aspecto demostrativo del software, denotando los argumentos contemplados por las directrices en cuestión, y asiendo a la etapa de implementación el punto de culminación de esta tarea demostrativa. La existencia de versiones progresivas del software se hace evidente en razón a la estrategia de software seleccionada.

El estudio de factibilidad realizado en los albores de este trabajo permitió establecer la factibilidad técnica de la herramienta software. Sin embargo, existen algunos otros argumentos como el conocimiento y conectividad del instrumento (anexo A) que requieren ser contemplados como parte de la implementación del sistema. A continuación, se exponen algunas secciones de código, que permiten ilustrar el proceso de implementación de la herramienta software, en base al uso de las directrices trazadas para el control de instrumentos.

La Figura 5.13 muestra una rutina de programa empleada en la actualización de parámetros del instrumento. Como se observa, dentro del ambiente de desarrollo seleccionado, se hizo uso de objetos de control directo para el envío de ordenes SCPI al instrumento. La elección de este tipo de objetos reside en las facilidades presentadas para el manejo de protocolos de mensajes como SCPI, mediante operaciones de lectura, escritura, ejecución y espera, sobre interfaces GPIB, VXI y Serial. La posibilidad de un control total del instrumento debido al uso de ordenes SCPI, y la rapidez en la ejecución de ordenes al no requerir elementos secundarios de software (controladores), hacen de los objetos de control directo la mejor elección de control de instrumentos dentro de VEE.

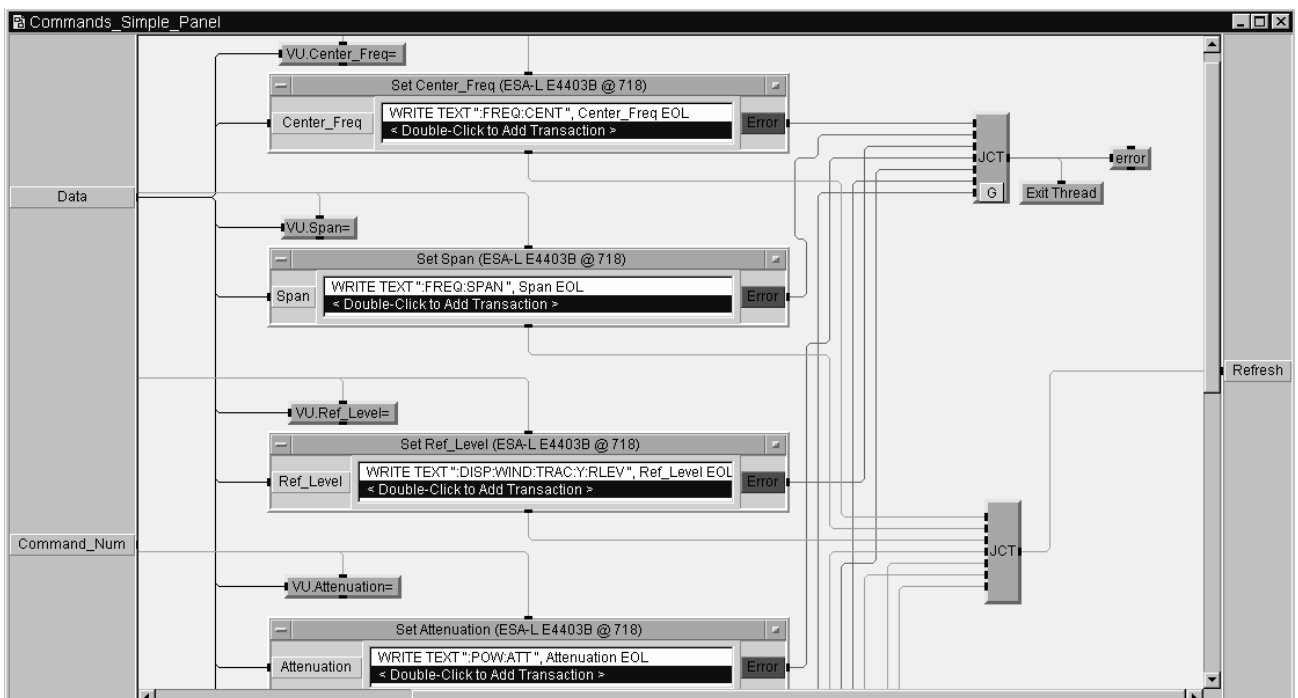


Figura 5.13. Modulo de programa para la actualización de valores de parámetros del instrumento.

La Figura 5.14 muestra el uso de ordenes SCPI para la obtención de valores de los parámetros del instrumento. Las ordenes SCPI son expresadas usando una notación corta (inciso 3.2.1.2.1), que si bien podría incrementar la probabilidad de error en el envío de ordenes, también resulta obvio pensar que el uso de una menor cantidad de caracteres minimizara el tiempo requerido por el analizador gramático del instrumento para la interpretación de ordenes, aunque de cierta manera esta disminución de tiempo no es perceptible. Otro aspecto a considerar, es la estructura mostrada por cada una de las ordenes SCPI empleadas, no obstante que la norma SCPI hace opcional el uso de algunos mnemotécnicos en diversas ordenes, esta consideración se encuentra sujeta a la decisión del diseñador del instrumento. Como se observa, algunas ordenes no demanda la presencia del mnemotécnico inicial que hace referencia al conjunto o subsistema de ordenes al que pertenecen, tal es el caso de las ordenes pertenecientes al subsistema de conversión SENSE, que permiten modificar parámetros de frecuencia, ancho de banda o potencia. Sin embargo, algunas otras ordenes como las referentes al subsistema pantalla DISPLAY si requieren el uso de este primer mnemotécnico. Así, la forma en como son conjuntadas las ordenes SCPI y el conocimiento de las características del instrumento bajo control, hacen casi imposible la selección errónea de ordenes para el manejo de un parámetro particular del instrumento.

La Figura 5.15 muestra de manera programática la obtención de trazas de datos y mediciones específicas del instrumento. La existencia y uso subsistemas e instrucciones de medición, como dos conjuntos de ordenes de naturaleza distinta, es apreciable de manera clara en instrumentos que facilitan tanto la adquisición simple de datos, como la realización de mediciones que requieren de un procesamiento explícito.

La adquisición de trazas de datos resulta ser una tarea a priori en todo instrumento, que facilite la visualización de señales. La realización de esta labor en instrumentos SCPI, demanda la comprensión de los mecanismos implícitos en la solicitud y transferencia de este tipo de datos. El envío de grandes cantidades de información dentro de SCPI, es manejada a través de bloques de datos de longitud definida, la cantidad de octetos contenidos en el bloque dependerá del formato de datos seleccionado y de la cantidad de elementos de información implicados (número de puntos en la traza, tamaño del archivo, etc.). El objeto que se muestra en la Figura 5.15, usado para la obtención de trazas de datos, permite la lectura de un bloque de datos de una longitud hasta cierto punto desconocida (señalada por un terminador), bajo un formato determinado. En algunos ambientes de desarrollo es necesaria una serie de operaciones adicionales a fin de establecer las características de un bloque de datos singular (inciso 4.2.2).

La toma de mediciones específicas es un compromiso casi inevitable en todo instrumento de medida. La norma SCPI enmarca el empleo de cada una de las instrucciones de medición que ésta define. El analizador de espectros ESA-L E4403B es por defecto puesto en un modo de operación de análisis espectral, dicho modo permite realizar una serie de mediciones, tales como potencia de canal, distorsión armónica, entre otras. La elección de otros modos posibles de operación del analizador de espectros requiere de elementos adicionales. Con el fin de demostrar el uso de instrucciones de medida, la herramienta software facilita la operación de la medición de potencia de canal (CHP, *Channel Power*), con la que cuenta el instrumento. Desde el punto de vista programático, la herramienta software implementada, tal como se observa en la Figura 5.15, hace uso de la instrucción de recolección FETCh para la obtención de mediciones (inciso 4.1.2). Dado que la configuración del modo de medición es realizada de manera preliminar y que proceso de adquisición de datos es realizado de modo continuo, el uso de instrucciones de medida MEASure y lectura READ para la obtención de mediciones resultan ser poco eficientes, llevando así a la instrucción FETCh, la cual no encierra un proceso de configuración y adquisición de datos, sino solo la lectura de la ultima medición valida realizada, a ser considerada la mejor elección, logrando así un mayor desempeño del control sobre la medición.

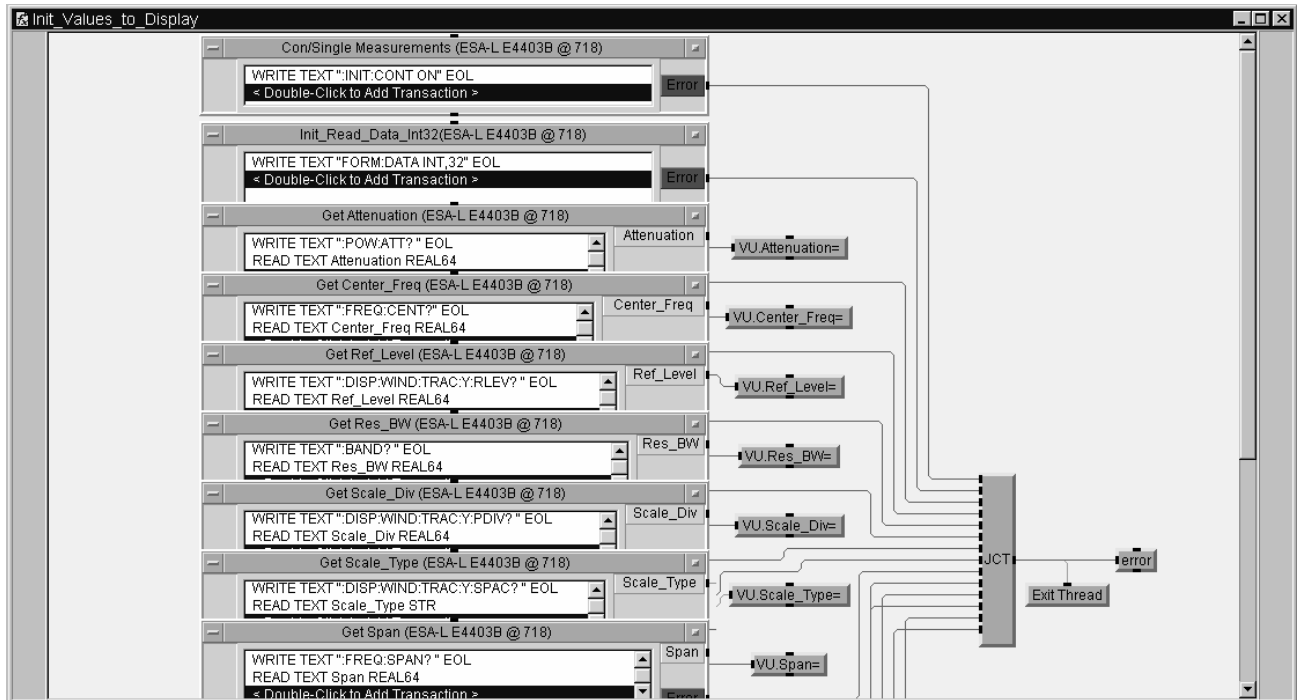


Figura 5.14. Módulo de programa para la obtención de parámetros del instrumento.

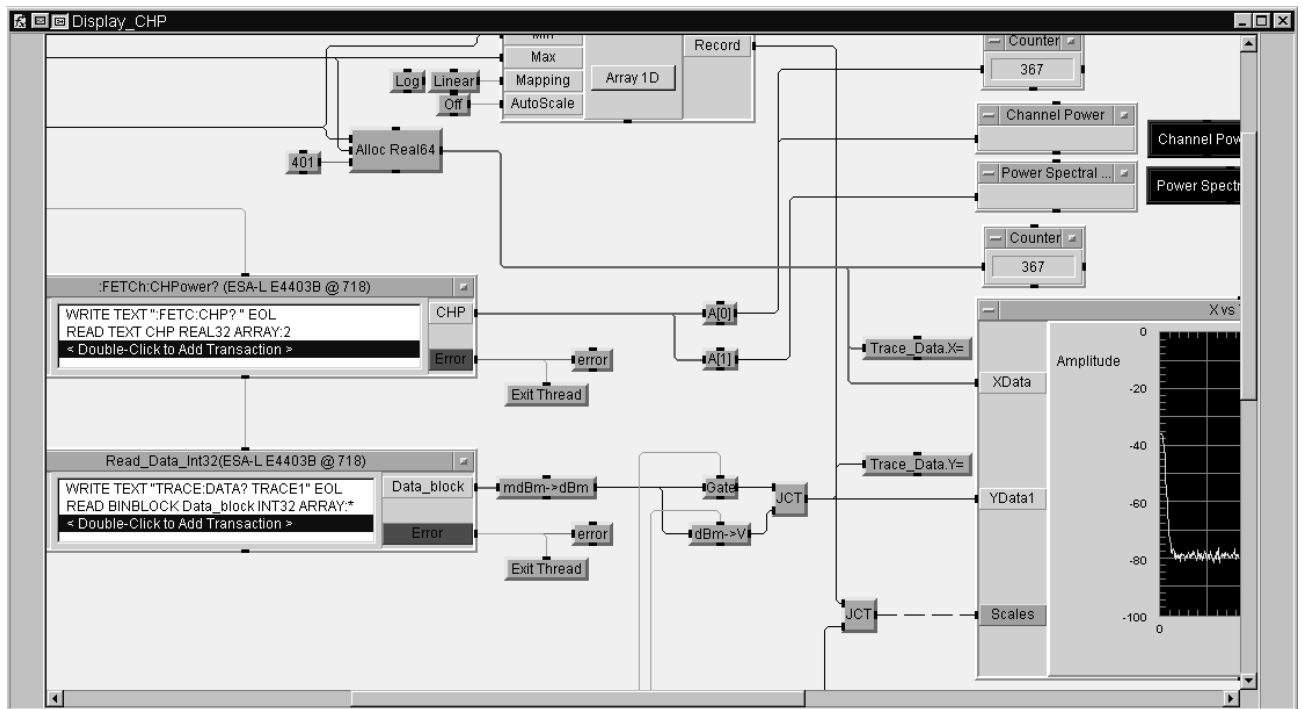


Figura 5.15. Módulo de programa para la obtención de trazas datos y mediciones del instrumento.

La Figura 5.16 muestra un procedimiento para la obtención de una imagen de la pantalla actual del instrumento. Este tipo de procesos implican el uso de la memoria de almacenamiento masivo del instrumento. El analizador de espectros ESA-L E4403B cuenta con dos elementos de almacenamiento masivo, una unidad flexible de 3 ½ y otra de disco duro referenciada como “C”. Dichas unidades permiten el almacenamiento de imágenes, datos de trazas, mediciones, estados, límites y tablas de correcciones. De manera general, el método utilizado en la adquisición de estas imágenes involucra tres operaciones básicas; el almacenamiento de la imagen en pantalla en un archivo, la transferencia de datos del archivo y la eliminación del archivo generado. La figura mencionada ejemplifi-

ca el uso de ordenes SCPI para la realización de estas tareas. Este tipo de ordenes se encuentran agrupadas dentro de la norma SCPI bajo el subsistema MMEMory. Los datos del archivo de imagen en el instrumento son manejados en bloques de datos. Los objetos del ambiente de desarrollo VEE usados, permiten la lectura y escritura de bloques de datos de longitud desconocida, lo cual facilita en gran medida esta tarea. En el caso del empleo de ambientes de desarrollo de propósito general como lo C++ y BASIC, donde no se cuentan con objetos de este tipo, se sugiere la consulta del tamaño del archivo a leer, mediante la consulta del catalogo de archivos (:MMEMory:CATalog?), con el fin de determinar la cantidad de datos que se obtendrán.

La Figura 5.17 muestra un proceso desarrollado para la inspección del estado del instrumento. Una actividad inicial recomendada en toda aplicación de control de instrumentos es la confirmación de la conectividad con el instrumento. Si bien basta con uso de alguna orden común IEEE 488.2 para la realización de esta tarea, existe un conjunto de ordenes agrupadas bajo el sistema SYSTEM que proveen de algunos procedimientos complementarios igualmente útiles, como lo pueden ser la obtención de características del hardware, configuraciones, información descriptiva de errores ocurridos, entre otros. De manera directa el ambiente de desarrollo VEE no cuenta con mecanismos para la implementación de un método de solicitud de servicio (inciso 4.4.2), para la detección y manejo del cambio de estado y condiciones de error en un instrumento, lo que llevo a realizar una emulación del mismo. El objeto empleado para esta labor, etiquetado en la figura mencionada con el nombre SRQ:GPIB7 permite verificar de manera cíclica pero no permanente, la línea SRQ del bus GPIB usado. Una vez que esta línea haya sido habilitada se procederá a la lectura en primer plano del registro de byte de estado, en el cual son reflejadas las solicitudes de servicio generadas por los registros de nivel inferior que han sufrido alguna modificación, y de los cuales se hará lectura con el fin de determinar la condición actual del instrumento, tal como lo muestra la Figura 5.18. De manera preliminar a las actividades mencionadas se requerirá de la configuración de los registros de habilitación de eventos, filtros de transición y el registro de habilitación de solicitud de servicio, que permitan la generación de las solicitudes de servicio deseadas.

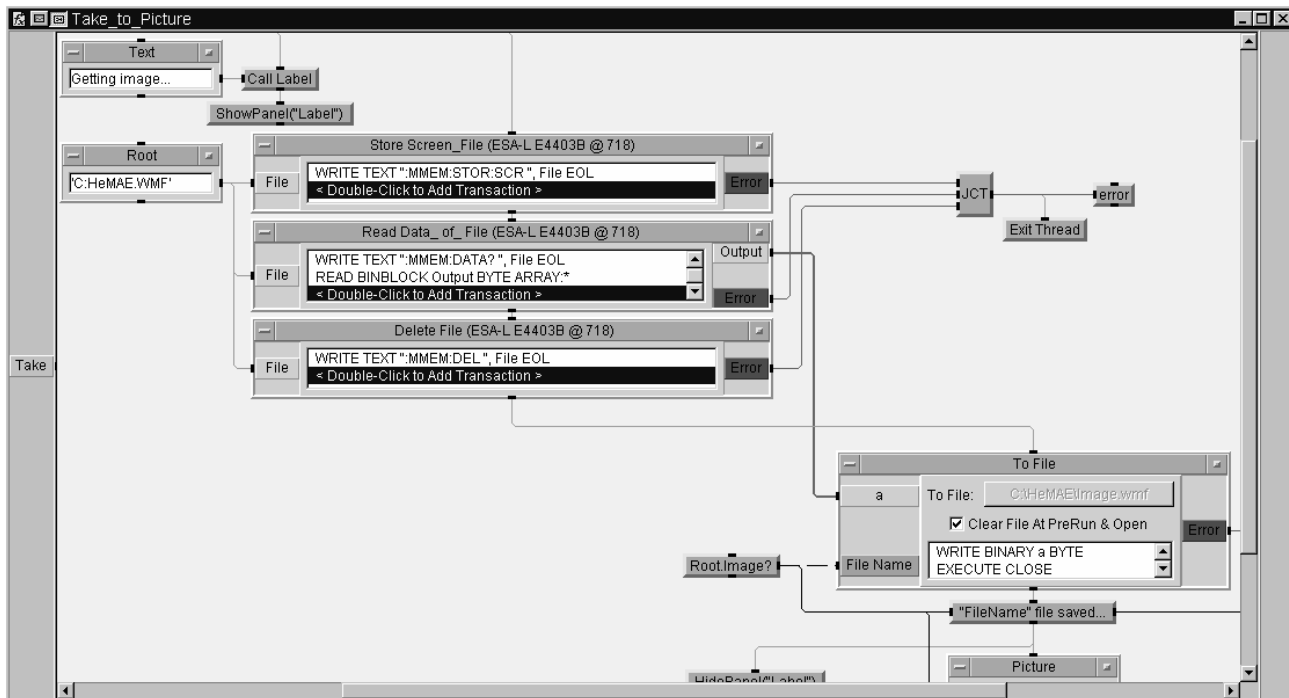


Figura 5.16. Modulo de programa para la obtención de una imagen de la pantalla del instrumento.

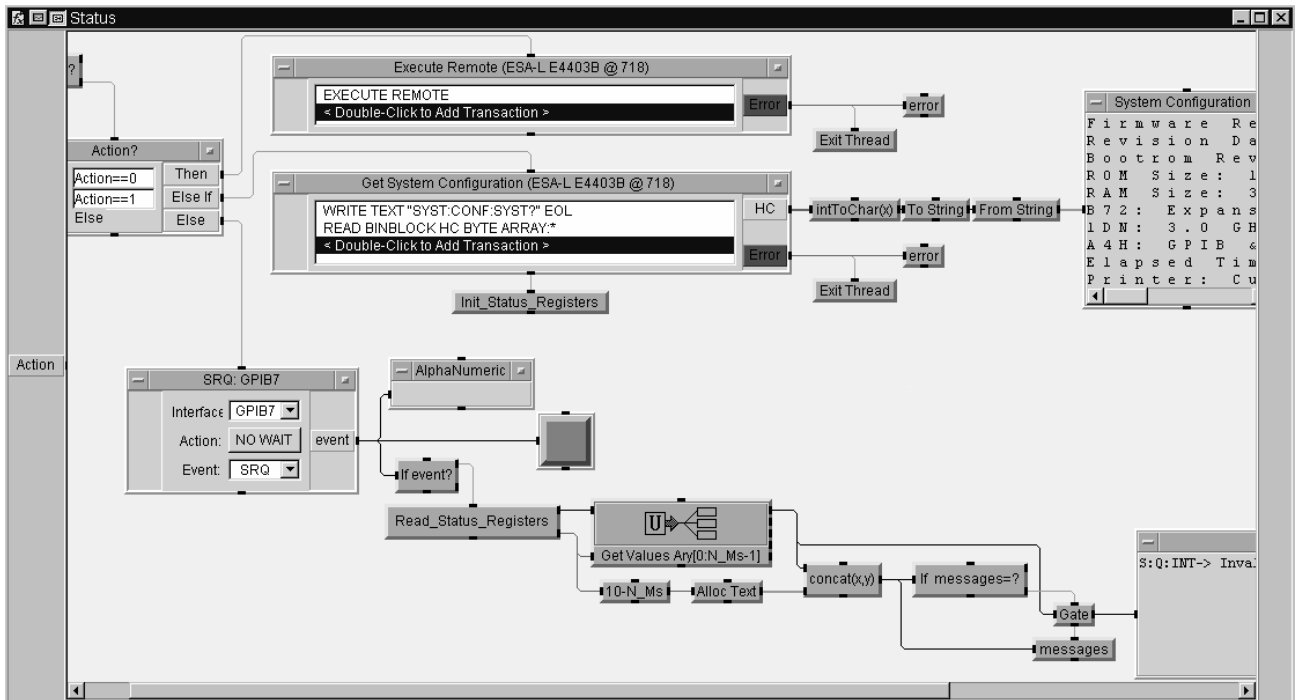


Figura 5.17. Módulo de programa para el monitoreo del estado del instrumento.

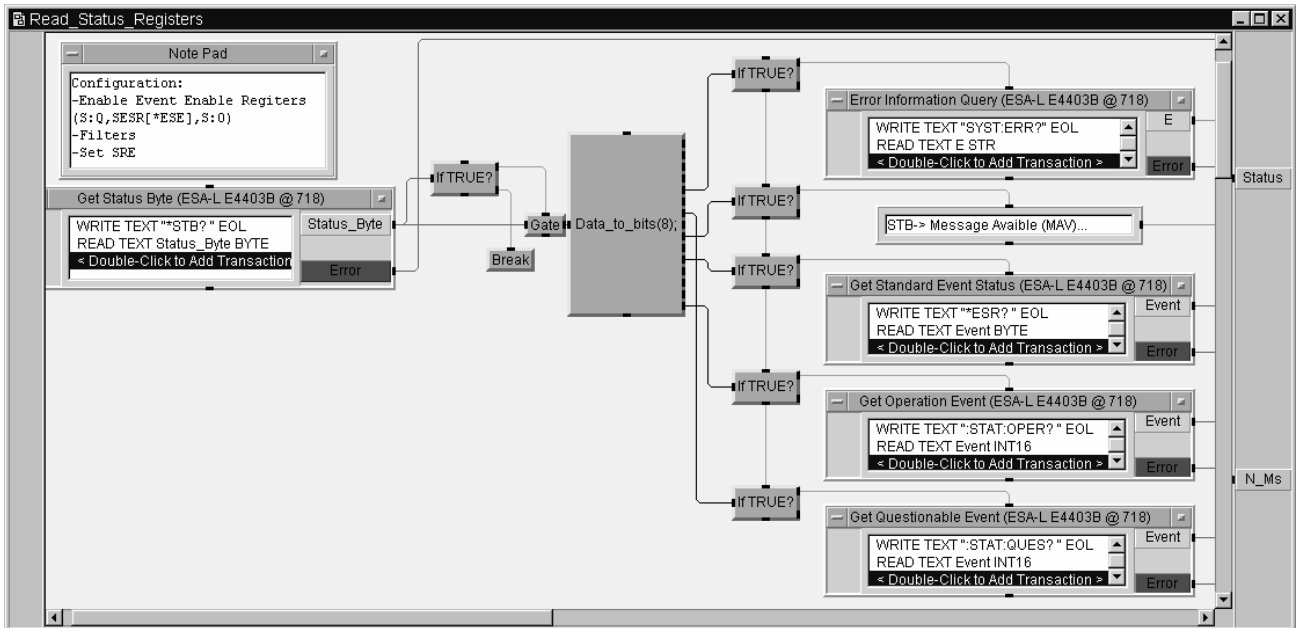


Figura 5.18. Módulo de programa para la lectura de registros de estados.

5.4. Validación

Las pruebas del software son un elemento crítico para la garantía de calidad del software, y representa una revisión final de las especificaciones, del diseño y de la codificación [30]. Cualquier producto de ingeniería puede probarse usando dos perfiles de prueba, llamados de caja blanca y de caja negra. De manera simple, el proceso de prueba del software encierra tres etapas, las cuales buscan probar los componentes del software, su integración y la validación del software ya consumado.

La validación de un software permite mostrar que un sistema cumple con las expectativas del usuario de acorde a su especificación. La validación es una tarea que incluye comprobaciones, inspecciones y revisiones, en cada etapa del proceso software, desde la definición de requerimientos

hasta la implementación, sin embargo su impacto es perceptible de manera más clara cuando el sistema final es puesto a prueba [39]. La validación del software se consigue mediante una serie de pruebas de caja negra que demuestran la conformidad con los requisitos.

A continuación se describen algunas pruebas de validación efectuadas sobre la herramienta software HeMAE ESA-L E4403B, el plan de prueba usado busca comprobar el cumplimiento de los requisitos establecidos para el software, en base al índole ilustrativo bajo el cual ha sido concebido.

La Figura 5.19 muestra una operación habitual de la herramienta software. Todas y cada una de las operaciones que impliquen el control del instrumento son realizadas mediante ordenes SCPI. La inspección constante del estado y trazas de datos del instrumento inducen un flujo de información constante entre instrumento y controlador, el cual es incrementado con la modificación de parámetros del instrumento. Dicha situación impide estimar una razón del flujo de información real para una operación habitual de la herramienta. El indicador numérico situado en la esquina superior derecha de la pantalla de trazas de la GUI (inciso 5.2.2), permitirá estimar de manera aproximada la cantidad de transacciones ocurridas durante la ejecución de la aplicación.

Tal como se plantea y muestra en la Figura 5.20 la herramienta software desarrollada facilita también el control de mediciones particulares del instrumento. La medición seleccionada para ejemplificar esta tarea ha sido la medición de potencia de canal, la cual permite calcular la potencia y densidad espectral contenida en un ancho de banda específico. La herramienta software permite el control de los argumentos de esta medición. Los parámetros de frecuencia central y ancho de banda de integración de la medición permiten situar el espectro de frecuencias del cual se requiere una medición de potencia. En la figura cuestionada se muestra un objeto de trazas que expresa la pantalla de la medición de potencia de canal en el instrumento. El cálculo de la potencia de canal es un mecanismo útil en sistemas de monitoreo de canales específicos de frecuencia, como lo son los servicios de análisis de enlaces de radio para redes 802.11b.

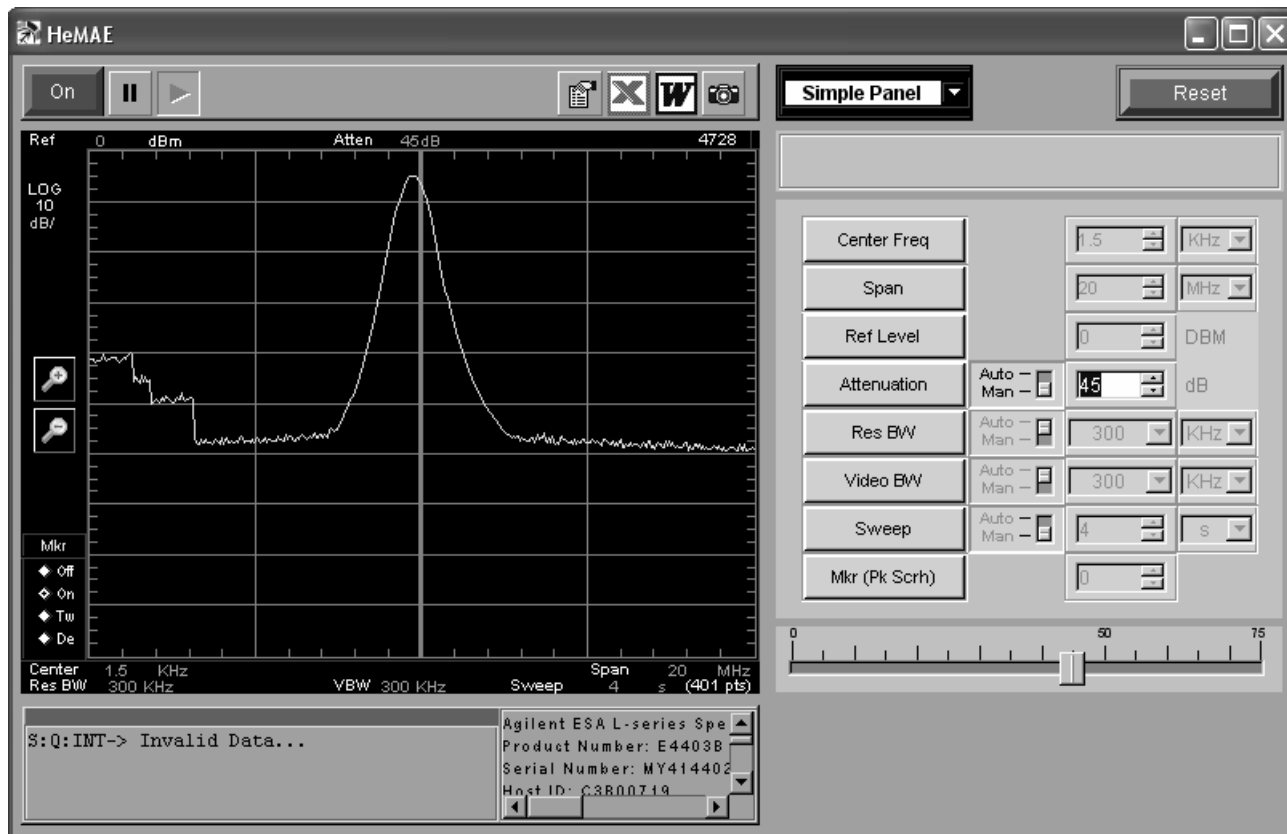


Figura 5.19. Operación habitual de la HeMAE ESA-L E4403B.

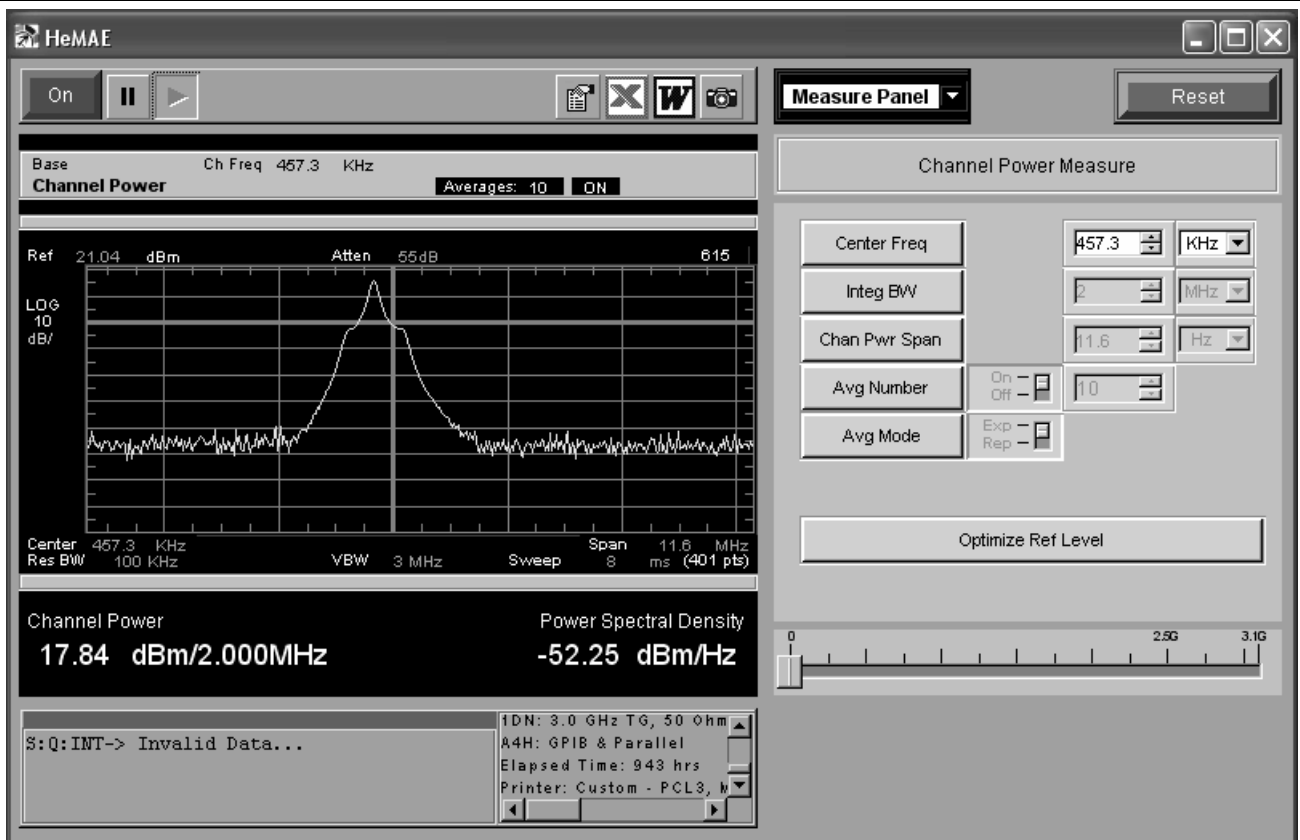


Figura 5.20. Operación de la HeMAE ESA-L E4403B para la obtención de mediciones de potencia de canal.

La Figura 5.21 ilustra la obtención de una imagen de la pantalla actual del instrumento. El medio de acceso a esta utilidad es presentado en la barra de herramientas de la aplicación. El tiempo de ejecución requerido por este proceso, es una variable dependiente del instrumento. Las operaciones, de almacenamiento, transferencia y eliminación del archivo de imagen en el instrumento, realizadas durante esta tarea, inducen un incremento considerable en el tiempo de ejecución. Con la finalidad de proveer mayor versatilidad a la tarea de obtención de imágenes, la herramienta software facilita la elección de una ruta de almacenamiento para el archivo de imagen obtenido, a través de un objeto de preferencias situado también en la barra de herramientas de la aplicación. El empleo de una ruta inadecuada para el almacenamiento del archivo de imagen en el controlador, generara un ejercicio inconcluso de la tarea.

En la Figura 5.22 se observa la operación de la herramienta software en presencia de un cambio en el estado del instrumento. En la ventana de estado, observada en la figura, se muestra una serie de mensajes que exhiben la circunstancia actual del instrumento. La inspección de un cambio en el estado del instrumento tiene como consecuencia un incremento en el número de transacciones entre instrumento y controlador. La validación por parte de la aplicación, de los valores asignados a los parámetros del instrumento, minimiza en la generación de errores de cuestionamiento, ejecución y orden, que son independientes del funcionamiento del instrumento, reduciendo indirectamente así, la cantidad de transacciones entre instrumento y controlador. Una tarea inherente al monitoreo del estado del instrumento, es la verificación de la conectividad con el dispositivo. El primer esfuerzo programático requerido en toda aplicación de instrumentación electrónica programable es la afirmación del posible control del instrumento. La herramienta software cuenta con una serie de mecanismos que le permiten el reconocimiento de dispositivos y la revisión de errores de conectividad ocurridos durante la ejecución de la aplicación.

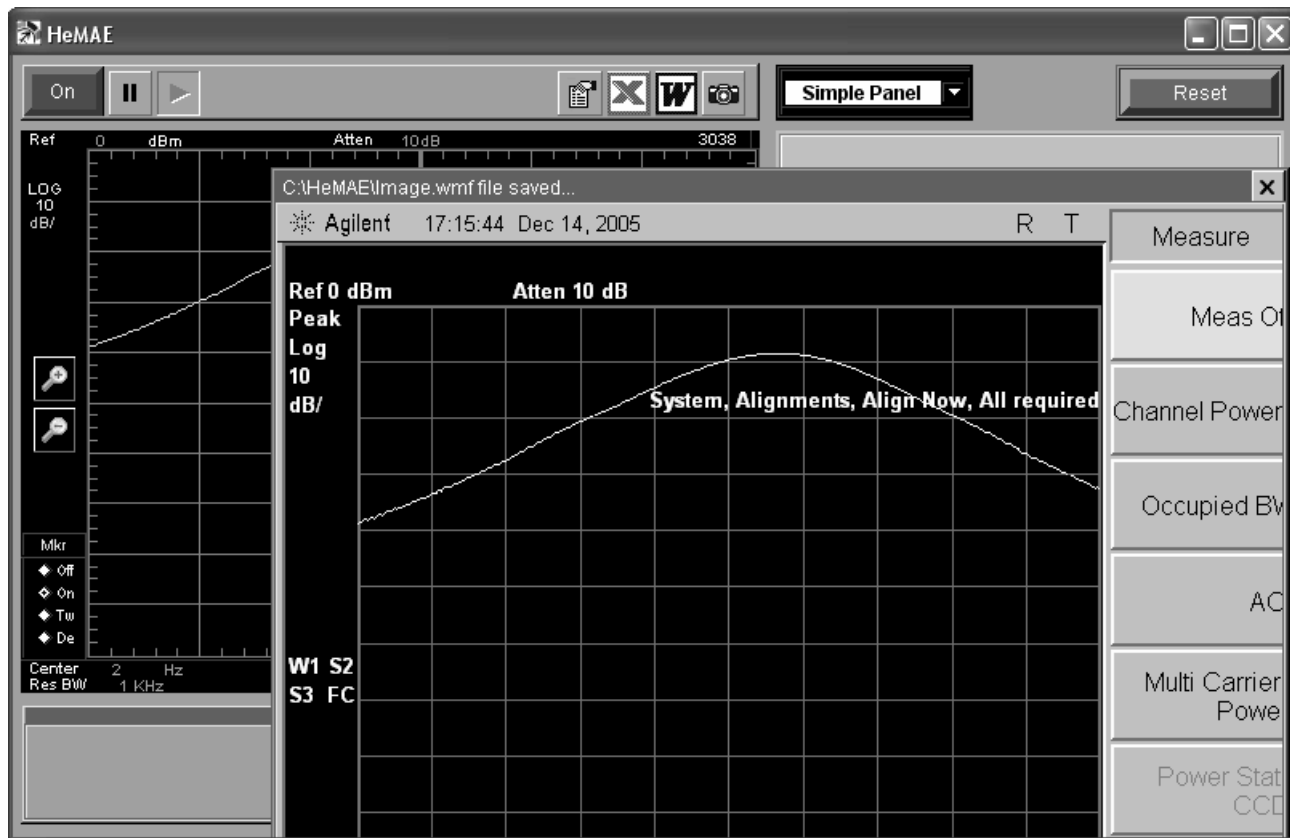


Figura 5.21. Operación de la HeMAE ESA-L E4403B para la obtención de imágenes de la pantalla del instrumento.

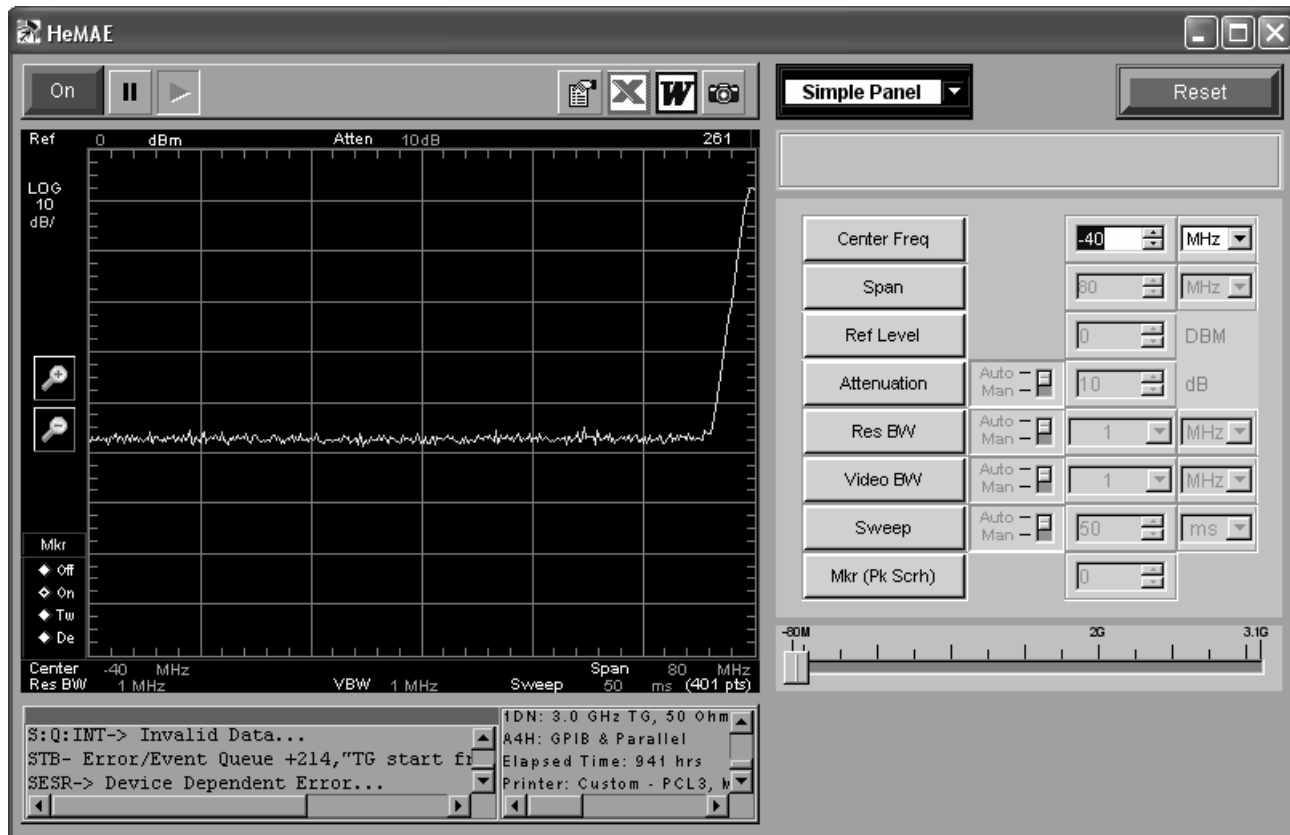


Figura 5.22. Operación de la HeMAE ESA-L E4403B bajo un estado particular del instrumento.

En razón a su encomienda de usabilidad, la herramienta software incorpora una serie de medios, que facilitan la generación de esquemas de reporte en aplicaciones como Microsoft Excel® y Word®. La Figura 5.23 muestra un documento en una hoja de cálculo, construido a partir de un conjunto de datos exhibidos por la herramienta. El tiempo requerido para la consumación de un documento obedece a la cantidad de operaciones requeridas para su construcción. La generación de este tipo de reportes demanda de manera evidente la existencia de la aplicación que permita su existencia, en el caso particular de esta herramienta software, su desarrollo y prueba involucro el uso de la versión 2003 para Microsoft Excel®, motivo por el cual su uso reclama una condición similar. La modificación de las rutas de almacenamiento de hojas de cálculo y documentos de texto generados por la herramienta software es llevada a cabo mediante un objeto de preferencias, situado de igual forma, en la barra de herramientas de la aplicación.

La existencia de algunos otros mecanismos tales como la pausa de la aplicación y por tanto del control sobre el instrumento, el reestablecimiento de las condiciones por defecto del instrumento, y la posibilidad de realizar ampliaciones y reducciones visuales sobre la pantalla de trazas, brindan a la herramienta software desarrollada un grado de usabilidad aun mayor.

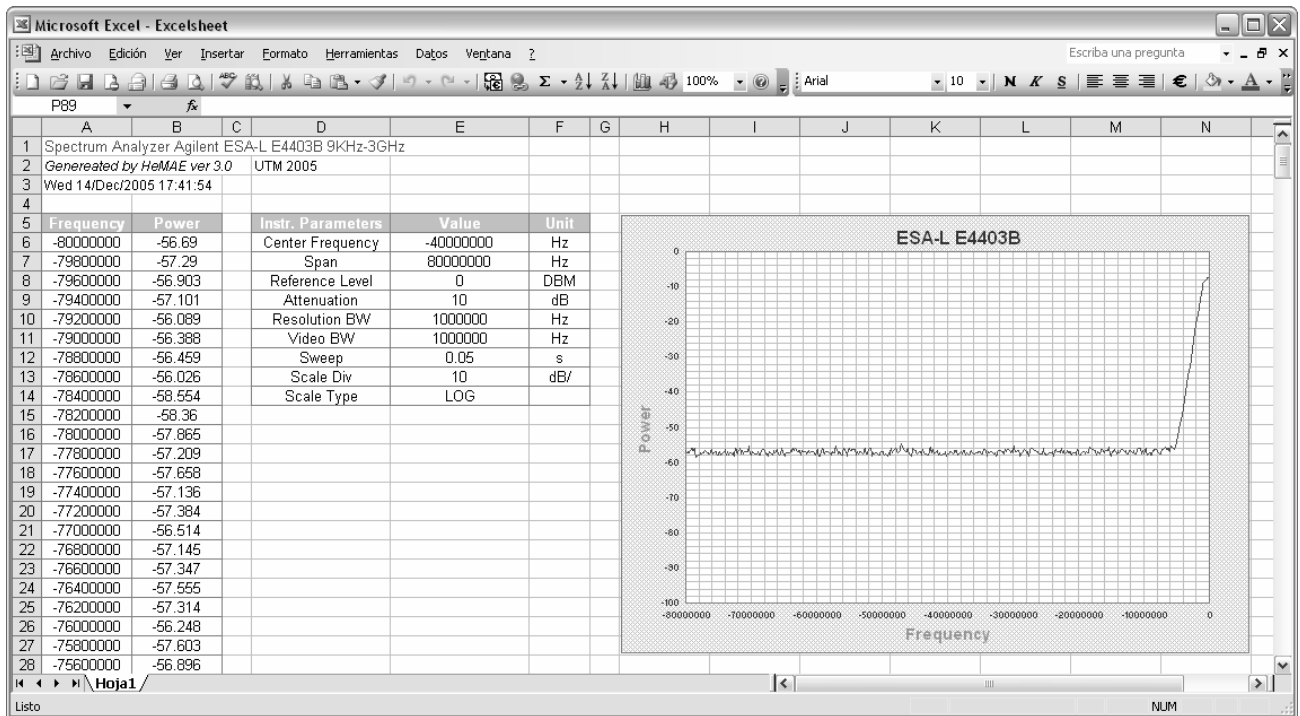


Figura 5.23. Documento sobre una hoja de cálculo generado por la HeMAE ESA-L E4403B.

6. Conclusiones

El proyecto de investigación aplicada en el campo de la instrumentación electrónica programable, y de manera particular referida al control de instrumentos mediante SCPI, que el presente trabajo de tesis involucró, permitió el planteamiento de un conjunto de preceptos teóricos, que fundamentaron el desarrollo de aplicaciones de este género.

Se mostró mediante el desarrollo de una herramienta software para el control del analizador de espectros ESA-L E4403B, el empleo de las directrices trazadas, permitiendo así validar, el trabajo de investigación realizado.

De las tareas complementarias demandadas durante el desarrollo del presente trabajo de tesis, se pueden mencionar las siguientes aportaciones:

- Con el fin de comprender los argumentos que sobrelleva la utilización de la norma SCPI, como un modelo conceptual para el control de instrumentos programables, se efectuó un estudio profundo sobre los procedimientos que la norma establece para la construcción y envío de ordenes, manejo de datos, modelado y clasificación de instrumentos.
- En base a la investigación de campo realizada, el estudio de la norma SCPI y el desarrollo de aplicaciones ejemplo sobre el control de instrumentos mediante SCPI, se delimitó un conjunto de pautas o directrices enfocadas al desarrollo de aplicaciones sobre el control de instrumentos mediante SCPI. Cubriendo aspectos como el uso de ordenes, la manipulación de datos, el acceso a memoria y el control del sistema de registro de estado, indispensable en el control de instrumentos SCPI. Buscando como meta la reducción del tiempo en el desarrollo de aplicaciones de este género.
- Se diseñó y desarrolló en base a un modelado de ingeniería de software, una herramienta software para el control del analizador de espectros ESA-L E4403B, que permitió mostrar de manera práctica la utilidad de las directrices planteadas, logrando un producto de software comparable en rendimiento con otros existentes en el mercado.

La valoración positiva del trabajo realizado, está cimentada en la documentación del proyecto de investigación aplicada desarrollado en esta tesis. Investigación que pretende sentar un importante referente sobre el control de instrumentos mediante SCPI, que lleva a personas de conocimiento menudo en el campo a tener una perspectiva más amplia sobre el tema, y a su vez le capacite en el desarrollo de sistemas de instrumentación electrónica programable.

Los resultados obtenidos han permitido sugerir la realización de algunos trabajos sobre la misma línea de investigación, que reafirmen la utilidad de la labor efectuada:

- Automatización de los servicios de análisis de enlace de radio en formato 802.11b manuales, ofrecidos en el mercado, mediante el uso de una antena de prueba omnidireccional, un analizador de espectros y un elemento de control (PC). Este tipo de análisis llamado también prueba de cobertura consiste en la exploración de las diferentes bandas de frecuencias adoptadas por la tecnología 802.11b con el fin determinar la banda o canal mas adecuado para el establecimiento de un enlace de radio particular.

- Integración de un sistema que permita determinar la longitud, impedancia característica y localización de fallos a lo largo de cables coaxial y par trenzado, mediante la técnica de reflectometría en el dominio del tiempo (TDR, *Time Domain Reflectometry*); cuya operación básica consista en la inyección en un extremo del conductor, de un pulso de energía que viaje hasta encontrar un cambio en la impedancia del cable que genere, que parte de la señal o toda ella, sea reflejada hacia la fuente. El tiempo, tamaño y polaridad de las señales reflejadas indicarán la ubicación y naturaleza de las faltas de continuidad en el cable. El sistema empleará un generador de pulsos, un analizador de espectros ó osciloscopio (medidor de la reflexión), elementos de conexión y una PC para el procesamiento de los datos.
- Construcción de software didáctico para la medición de parámetros en líneas de transmisión, que en base a desarrollos prácticos, permitan a los usuarios interesados, una comprensión más efectiva sobre el tema, transparentando los procesos que conlleva el desarrollo de esta tarea. Dicho software incluirá el manejo de instrumentos como analizadores de espectros, osciloscopios, fuentes de alimentación, etc. como parte esencial de los sistemas pedagógicos desarrollados.
- Desarrollo de controladores o herramientas software para el control de instrumentos del LabCD como osciloscopios modelo 54621A y fuentes de alimentación modelo E3646A, que no cuentan con interfaces que sean fáciles de operar para el alumnado ajeno al manejo de estos instrumentos. Para el caso particular de los osciloscopios 54621A, cuyo modo de programación no incluye la posibilidad de un control mediante ordenes SCPI, se sugiere la utilización de la investigación efectuada, como un medio para la comprensión de las ordenes basadas en el estándar 488.2 de las cuales hace uso.

Bibliografía

- [1] Agilent Technologies: Agilent E4444A BenchLink Spectrum Analyzer Software, Product Overview. Agilent Technologies Inc., January, 2002.
- [2] Agilent Technologies: Agilent IntuiLink for PSA and ESA E/L/EMC-series Spectrum Analyzers Software User's Guide. Agilent Technologies Inc., 2000.
- [3] Agilent Technologies: Agilent SICL, User's Guide for Windows. Agilent Technologies Inc., February, 2002.
- [4] Agilent Technologies: Agilent VISA, User's Guide. Agilent Technologies Inc., February, 2002.
- [5] Agilent Technologies: Application Note 150, Agilent Spectrum Analysis Basics. Agilent Technologies Inc., April, 2004.
- [6] Agilent Technologies: Application Note 1286-1, 8 Hints for making Better Spectrum Analyzer Measurements. Agilent Technologies Inc., 2000.
- [7] Agilent Technologies: Application Note 1318, Optimizing Spectrum Analyzer Measurement Speed. Agilent Technologies Inc., 2000.
- [8] Agilent Technologies: Application Note 1465-3, Test-System Development Guide, Understanding Drivers and Direct I/O. Agilent Technologies Inc., November, 2002.
- [9] Agilent Technologies: Application Note 1465-9, System Developer Guide, Using LAN in Test Systems: The Basics. Agilent Technologies Inc., July, 2004.
- [10] Agilent Technologies: Application Note 1465-12, System Developer Guide, Using USB in the Test and Measurement Environment. Agilent Technologies Inc., October, 2004.
- [11] Agilent Technologies: Feeling Comfortable with VXIbus. Agilent Technologies Inc., April, 2001.
- [12] Agilent Technologies: Getting Started Guide, ESA Series Spectrum Analyzers. Agilent Technologies Inc., April, 2004.
- [13] Agilent Technologies: IVI Standards Tutorial. Agilent Technologies Inc., November, 2002.
- [14] Agilent Technologies: Measurement Guide and Programming Examples, PSA and ESA Series Spectrum Analyzers. Agilent Technologies Inc., April, 2004.
- [15] Agilent Technologies: User's and Programmer's Reference, Volume 1, Core Spectrum Analyzer Functions, ESA Series Spectrum Analyzers. Agilent Technologies Inc., April, 2004.
- [16] Agilent Technologies: User's and Programmer's Reference, Volume 2, One-Button Power Measurements, PSA and ESA Series Spectrum Analyzers. Agilent Technologies Inc., April, 2004.
- [17] Agilent Technologies: VEE Pro Advanced Techniques. Agilent Technologies Inc., 2000.
- [18] Agilent Technologies: VEE Pro User's Guide. Agilent Technologies Inc., 2000.
- [19] Agilent Technologies: Web Remote Control SA Software, Installation and Setup. Agilent Technologies Inc., 2001.

- [20] García, Josué N.: Administración de los recursos de un laboratorio de comunicaciones digitales mediante su instalación, configuración y puesta en funcionamiento. Universidad Tecnológica de la Mixteca, 2004, México.
- [21] Gast, Matthew S.: 802.11 Wireless Networks, The definitive guide. O'Reilly & Associates Inc., 2002.
- [22] González, A.: Estudio del protocolo IEEE 488 mediante el desarrollo de una herramienta de simulación. Universidad Tecnológica de la Mixteca, 2003, México.
- [23] Helsel, R.: Visual Programming with HP VEE. Tercera Edición, Prentice Hall, 3th edition, 1998.
- [24] Lázaro, Antonio M.: LabView 6i, Programación Gráfica para el Control de Instrumentación, Paraninfo/Thomson Learning, España, 2001.
- [25] Mariño, P., Nogueira, J. y Hernández, H.: "Laboratorio de instrumentación programable para prueba de circuitos electrónicos y análisis de señales de Bus GPIB". Revista TEMAS, Universidad Tecnológica de la Mixteca, Vol. 4, pp. 29-36, Mayo-Agosto 2000, México.
- [26] National Instruments: PXI Product Guide, CompactPCI Optimized for Test, Control, and Design. National Instruments Co., 2004.
- [27] National Instruments: The Measurement and Automation Catalog. National Instruments Co., 2005.
- [28] PCI Industrial Computers Manufacturers: CompactPCI Short Form Specification. PCI Industrial Computers Manufacturers, Revision 2.1, September, 1997.
- [29] PCI Special Interest Group: PCI Local Bus Specification. PCI Special Interest Group, Revision 2.2, December, 1998.
- [30] Pressman, Roger S.: Ingeniería de Software, Un enfoque práctico. McGraw-Hill/ Interamericana de España, 1998.
- [31] PXI Systems Alliance: PXI Hardware Specification. PXI Systems Alliance, Revision 2.2, September, 2004.
- [32] Ritter, David J.: LabVIEW GUI, Essential Techniques. McGraw-Hill, 2002.
- [33] Ron Wolfe: Application Note 030, Short Tutorial on VXI/MXI. National Instruments Inc., April, 1998.
- [34] SCPI Consortium: Standard Commands for Programmable Instruments, Volume 1, Syntax and Style. SCPI Consortium. May, 1999.
- [35] SCPI Consortium: Standard Commands for Programmable Instruments, Volume 2, Command Reference. SCPI Consortium. May, 1999.
- [36] SCPI Consortium: Standard Commands for Programmable Instruments, Volume 3, Data Interchange Format. SCPI Consortium. May, 1999.
- [37] SCPI Consortium: Standard Commands for Programmable Instruments, Volume 4, Instrument Classes. SCPI Consortium. May, 1999.
- [38] Shneiderman, Ben: Designing the User Interface, Strategies for effective human-computer interaction. Addison-Wesley Longman, 1998.
- [39] Sommerville, Ian: Ingeniería de Software. Pearson Educación, 2002.

Sitios de Internet

- [URL1] <http://adn.tm.agilent.com>, "Pagina de desarrollos Agilent", Agilent Technologies 2004. Accesada el 22 de febrero de 2005.

- [URL2] <http://www.agilent.com>, “Pagina de la firma Agilent Technologies”, Agilent Technologies 2000-2002. Accesada el 31 de enero de 2005.
- [URL3] <http://docs.hp.com>, “Pagina de soporte HP”, Hewlett Packard, 2004. Accesada el 4 de febrero de 2005.
- [URL4] <http://www.ivifoundation.org>, “Pagina de la fundación IVI”, IVI Foundation, 2004. Accesada el 18 de noviembre de 2004.
- [URL5] <http://www.lxistandard.org/>, “Pagina del consorcio LXI”, LXI Consortium, 2004-2005. Accesada el 29 de octubre de 2004.
- [URL6] <http://msdn.microsoft.com>, “Pagina de desarrollos Microsoft”, Microsoft Corporation, 2005. Accesada el 22 de junio de 2005.
- [URL7] <http://www.ni.com>, “Pagina de la firma National Instruments”, National Instrument Corporation, 2003. Accesada el 15 de noviembre de 2004.
- [URL8] <http://www.picmg.org>, “Pagina del grupo de fabricantes de computadoras industriales basadas en PCI”, PCI Industrial Computers Manufacturers, 2001-2005. Accesada el 26 de octubre de 2004.
- [URL9] <http://www.pcisig.org>, “Pagina del grupo de interés especial en el bus PCI”, PCI Special Interest Group, 2004. Accesada el 20 de octubre de 2004.
- [URL10] <http://www.pxisa.org>, “Pagina de la alianza de sistemas basados en PXI”, PXI Systems Alliance, 2004. Accesada el 28 de octubre de 2004.
- [URL11] <http://www.scpiconsortium.org>, “Pagina del consorcio SCPI”, SCPI Consortium, 2003. Accesada el 2 de diciembre de 2004.
- [URL12] <http://www.vxibus.org>, “Pagina del consorcio para el desarrollo de sistemas basados en VXI”, VXI Consortium, 2004. Accesada el 22 de octubre de 2004.

Acrónimos

ADE (*Application Development Environment*, Ambiente de Desarrollo de Aplicaciones)

API (*Application Programming Interface*, Interfaz de Programación de Aplicaciones)

ATE (*Automated Test Equipment*, Equipo, Equipo de Prueba Automatizado)

AUTOTESTCON (*Automated Test Conference*, Conferencia sobre Automatización de Pruebas)

BASIC (*standing for Beginner's All Purpose Symbolic Instruction Code*)

DAQ (*Data Acquisition*, Adquisición de Datos)

DIF (*Data Interchange Format*, Formato de Intercambio de Datos)

DLL (*Dynamic Link Library*, Biblioteca de Enlace Dinámico)

DSP (*Digital Signal Processing*, Procesamiento Digital de Señales)

FFT (*Fast Fourier Transform*, Transformada Rápida de Fourier)

GPIB (*General Purpose Interface Bus*, Bus de Interfaz de Propósito General)

GUI (*Graphical User Interfaz*, Interfaz Gráfica de Usuario)

HPIB (*Hewlett Packard Internal Bus*, Bus Interno de Hewlett Packard)

IEC (*Internacional Electrotechnical Comisión*, Comisión Internacional de Electrotécnica)

IEEE (*Institute of Electrical and Electronic Engineers*, Instituto de Ingenieros Eléctricos y Electrónicos)

IF (*Intermediate Frequency*, Frecuencia Intermedia)

IVI (*Interchange Virtual Instrument*, Instrumento Virtual Intercambiable)

IVI-COM (*IVI Component Object Model*)

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*)

LabWindows/CVI (*Complete Virtual Instrumentation*)

LAN (*Local Area Network*, Red de Área Local)

LO (*Local Oscillator*, Oscilador Local)

LXI (*LAN eXtensions for Instrumentation*)

MATE (*Modular Automated Test Equipment*, Equipo de Prueba Automatizado Modular)

MXI (*Multisystem eXtension Interface*)

OOP (*Object Oriented Programming*, Programación Orientada a Objetos)

PC (*Personal Computer*, Computadora Personal)

PCI (*Peripheral Component Interconnection*)

PCISIG (*PCI Special Interest Group*, Grupo de Interés Especial en el bus PCI)

PICMG (*PCI Industrial Computer Manufacturers Group*, Grupo de Fabricantes de Computadoras Industriales basadas en PCI)

PXI (*PCI eXtensions for Instrumentation*)

PXISA (*PXI Systems Alliance*, Alianza de Sistema basados en PXI)
RBW (*Resolution Bandwidth*, Resolución del Ancho de Banda)
RF (*Radio Frecuencia*, Radio Frecuencia)
RQS (*Request Service*, Servicio a Solicitud)
SA (*Spectrum Analyzers*, Analizador de Espectros)
SCPI (*Standard Commands for Programmable Instruments*, Estándar de Ordenes para Instrumentos Programables)
SESR (*Standard Event Status Register*, Registro de Eventos Estándar)
SICL (*Standard Instrument Control Library*, Biblioteca de Control para Instrumentos Estándares)
SRER (*Service Request Enable Register*, Registro de Habitación de Solicitud de Servicio)
SRQ (*Service Request*, Solicitud de Servicio)
STB (*Status Byte*, Byte de Estado)
TDR (*Time Domain Reflectometry*, Reflectometría en el Dominio del Tiempo)
USB (*Universal Serial Bus*, Bus Serial Universal)
VBW (*Video Bandwidth*, Ancho de Banda de Video)
VEE (*Visual Engineering Environment*, Entorno Visual de Desarrollo para Ingeniería)
VI (*Virtual Instrument*, Instrumento Virtual)
VISA (*Virtual Instrumentation Software Application*, Software de Aplicación para Instrumentación Virtual)
VME (*Versa Module Europe*)
VPI (*Visual Programming Interface*, Interfaz de Programación Visual)
VSA (*Vector Signal Analyzer*, Analizador de Señales Vectorial)
VXI (*VME eXtension for Instrumentation*)

Anexo A. Conectividad con instrumentos

La primera actividad que conlleva el desarrollo de cualquier aplicación relacionada con el control de instrumento es la aseveración de la conectividad con el dispositivo. Si bien esta tarea es una suma de esfuerzos, que implican consideraciones tanto de hardware como software, resulta ser la interacción entre estos elementos el punto crucial de esta labor. Los siguientes son algunos consejos útiles en la realización de esta tarea:

- Sobre el instrumento. El papel más importante en el alcance de la conectividad con los instrumentos, es el asumido por el mismo dispositivo. Este define no sólo la interfaz física que es requerida, sino también el protocolo de mensajes empleado, así un instrumento podrá utilizar interfaces GPIB, Serial, USB, LAN, etc., y ser controlable mediante ordenes SCPI, ordenes derivadas del estándar 488.2, etc., definiendo así las particulares de su control.
- Sobre el hardware del controlador. Si bien los mecanismos usados por el controlador, se encuentran sujetos a las condiciones establecidas por el dispositivo a controlar, la posibilidad de la adición de dispositivos de conversión entre estas interfaces GPIB, Serial, USB, LAN, etc., hacen de esta condición algo menos intrascendente.
- Sobre el software del controlador. Los mecanismos que llevan a alcanzar la conectividad con los instrumentos, están determinados en gran parte por los componentes versátiles del sistema, tal es el caso del software usado en el control del dispositivo. De manera general, son dos los elementos de software que destacan en un sistema de control de instrumentos, el software de aplicación y el de entrada y salida (I/O, *Input/Output*). Tal como lo muestra la Figura A.1, estos son relacionados particularmente con el controlador. El software de aplicación es o bien provisto por el fabricante del instrumento tal es el caso de Agilent BenchLink, Agilent Web Remote Control, Agilent IntuiLink, etc. o desarrollado mediante lenguajes de programación y utilidades para aplicaciones de instrumentación programable, como lo son VEE, LabVIEW, Visual Basic, C, C++, etc. Aunque el software de aplicación sobrelleva en mayor parte las tareas asumidas por el controlador, es el software de I/O, el encargado de la interacción entre el hardware del controlador y la aplicación. Si bien la mayoría del software de I/O es referido a través de lenguajes de propósito general tal como C y C++, su distinción es hecha principalmente a razón del uso herramientas de software propietarias como VEE, LabVIEW y LabWindows/CVI en donde el software de I/O es visto como una unidad independiente encargada de manejo de la interfaz física con el instrumento. La elección del software de I/O se encuentra sujeta a disposiciones como el empleo de ordenes SCPI, el uso de controladores específicos, etc. que el software de aplicaciones establezca para el control de instrumentos. El manejo del software I/O representa tal vez el mayor desafío dentro de la labor de conectividad. Firmas como Agilent Technologies y National Instruments han buscado facilitar esta tarea con contribuciones particulares de software de I/O, sin embargo el uso de este software se encuentra regularmente sujeto al hardware empleado, dificultando nuevamente la labor y rompiendo con la cualidad de estándar que se desea establecer. Una de las arquitecturas estándares de software de I/O de mayor demanda, es la concebida por la empresa Alliance System VXIplug&play, denominada VISA (*Virtual*

Instrument Software Architecture). Otras arquitecturas como SICL (*Standard Instrument Control Library*) de la firma Agilent Technologies también son empleadas regularmente.

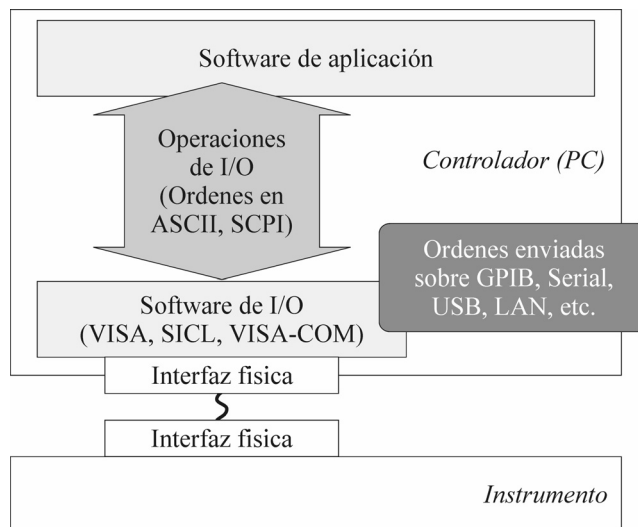


Figura A.1. Esquema general de componentes involucrados en la conectividad con instrumentos.

A.1. Desarrollo de aplicaciones usando VISA

Una de las alternativas de software de I/O para el control de instrumentos de uso más popular en entornos de instrumentación programable como VEE y LabVIEW, así como en lenguajes basados en C/C++ o BASIC es la herramienta VISA (*Virtual Instrument Software Architecture*). Esta define interfaces de comunicaciones con los instrumentos VXI, GPIB y serial. Desarrollada por la empresa Alliance System *VXIplug&play*, su condición de estándar ha hecho de este software de I/O, un elemento de uso confiable, en el desarrollo de aplicaciones de control de instrumentos mediante protocolos de intercambio de mensajes como SCPI. Firmas inmersas profundamente en el ámbito de la instrumentación electrónica programable, como Agilent Technologies y National Instruments proporcionan soluciones de software de I/O basadas en la arquitectura VISA.

Agilent Technologies agrupa la arquitectura VISA (Agilent VTL), junto con un particular desarrollo llamado SICL en un software propietario de I/O (*Agilent IO Libraries Control*). Dicho software facilita la configuración de interfaces de I/O, tales como GPIB, VXI, LAN, etc., al igual que provee utilidades (librerías) que pueden ser usadas por diversos ambientes de desarrollo, para el control de instrumentos. Este software tiene la facultad de poder coexistir con otra herramienta VISA de distinto proveedor en un mismo controlador (PC), sin embargo esta característica puede traer consigo inconvenientes para algunos ambientes de desarrollo.

El software de I/O de National Instruments llamado NI-VISA, provee de igual forma, no solo herramientas para la verificación de la conectividad con el instrumento mediante VISA, sino también suministra código utilitario para desarrollos sobre diferentes ambientes.

Una nota relevante en el uso de software de I/O propietario, en la existencia de una dependencia con el hardware bajo operación, así el manejo de productos de interfaz de una firma particular requerirá el uso exclusivo de software de I/O de la misma firma. De esta manera una tarjeta de interfaz PCI/GPIB de Agilent Technologies requerirá para su manejo exclusivamente el uso de software de I/O de Agilent. Afortunadamente esta dependencia se encuentra establecida en este nivel de software, dejando al software de aplicación libre de esta atenuante.

El uso de utilidades VISA demanda, la atención de ciertas situaciones. La Tabla A.1 resume algunas de estas consideraciones para diversas plataformas de software.

Tabla A.1. Consideraciones para el uso de software del I/O VISA.

Ambiente de desarrollo	Observaciones
Lenguajes basados en C/C++ (Visual C++, Borland C++)	La inclusión de utilidades VISA es hecha a nivel programático. El desarrollo de una aplicación para el control de instrumentos sobre C++, requiere de la consumación de dos tareas, la inclusión de archivos (include file) y la de librerías de referencia (library files). La inclusión de archivos de cabecera varía de acuerdo a la aplicación, sin embargo esta operación en general solo requiere de la señalización de la ruta en donde estos archivos se encuentran (Ej. C:\...\VISA\WINNT\INCLUDE). De igual forma, la adición de librerías conlleva la especificación del archivo o la ruta en que este se encuentre, dependiendo de tipo de aplicación (16 o 32 bits) y del ambiente de desarrollo empleado (Ej. C:\...\VISA\WINNT\LIB\MSC\VISA32.LIB para compiladores Microsoft, o bien C:\...\VISA\WINNT\LIB\BC\VISA32.LIB para compiladores Borland).
Lenguajes basados en Basic (Visual Basic)	El uso de funciones particulares provistas por algún tipo de software de I/O basada en un arquitectura VISA, dentro de un lenguaje como Visual Basic, se encuentra sujeto, únicamente a la adición del modulo de programa Visual Basic (visa32.bas), que contenga dicha especificación. La ruta del archivo esta sujeta a la elección hecha durante la instalación de la referencia VISA (Ej C:\...\VISA\WINNT\INCLUDE).
LabVIEW	El manejo de hardware de interfaz dentro de VEE reclama la existencia de un software de I/O tal como NI-VISA. La herramienta NI-VISA provee medios para la configuración de gran variedad de hardware, sin embargo su utilidad se reduce generalmente a la administración de equipo propietario de la firma. De este modo, dentro de LabVIEW el manejo de dispositivos de otro fabricante exige regularmente la presencia de software de I/O del fabricante.
VEE	La existencia de software de I/O adecuado a VEE, como lo son las librerías de control de I/O de Agilent, hace pensar en una dependencia absoluta de este ambiente a este particular software de I/O, sin embargo este es un argumento solo valido para algunos objetos en VEE, tales como los objetos de control directo, y controladores exclusivos del lenguaje (paneles de control), quienes hacen uso de arquitectura alternativa a VISA, denominada SICL, exclusiva del software de I/O cuestionado.

