UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 5

ENTORNOS VIRTUALES

Análisis y diseño de un sistema de instrumentación virtual para simular un osciloscopio.

Trabajo para optar por el Título de Ingeniero en Ciencias Informáticas.

Autores: Ramiro Ramírez García.
Reynaldo Gari Danger.

Tutores: Ing. Ivanier Martínez Benítez.
Ing. Tte.Leonel Salazar Videaux.

Consultante: Tte. Cor. Gerardo García Pierrat.

Ciudad de la Habana

Octubre, 2007

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo, y autorizamos al Proyecto Modernización de la Técnica y el Armamento (MTA) del Centro de Compatibilización e Integración y Desarrollo de Productos Informáticos para la Defensa de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año ___2008____.

Autores:

Reynaldo Gari Danger

_____

Ramiro Ramírez García

_____

Tutores:

Ing. Ivanier Martínez Benítez.

_____

Ing. Tte. Leonel Salazar Videaux.

_____

Datos de contacto

Nombre y Apellidos: Leonel Salazar Videaux
Edad: 24 años.

Ciudadanía: cubano.

Institución: Centro de Compatibilización e Integración y Desarrollo de Productos Informáticos para la Defensa.

Título: Ingeniero en Ciencias Informáticas.

E-mail: leonelsv@uci.cu

Nombre y Apellidos: Ivanier Martínez Benítez.

Edad: 24 años.

Ciudadanía: cubano.

Institución: Centro de Compatibilización e Integración y Desarrollo de Productos Informáticos para la Defensa.

Título: Ingeniero en Ciencias Informáticas.

E-mail: ivaniek@uci.cu

Agradecimientos

Agradezco a todas las personas que me alentaron cuando lo necesité, en especial a mi madre, mi hermanita quien ha sido un ejemplo a seguir para mí, a mi padre, quien siempre tiene un consejo que darme, a mi Abuelita que no está entre nosotros pero sé que está tan feliz como yo, a mis tías, Miriam, Patricia y Margarita quienes han sido mis madres también, a mis primas todas, que forman parte de este triunfo.

A mis amigos, que hubiera sido sin ellos, cuyos nombres no los escribí aquí pero los recordaré siempre.

A mis tutores cuya labor fue decisiva en esta emprendedora tarea

¡¡ A todos gracias!!

Reynaldo.

A mi madre, por confiar en mí y apoyarme en cada momento.

A mi hermana, ella es mi otra madre.

A mis tíos y mi padre que siempre han creído en mí y sé que esperan ansiosos este momento.

A mis amigos, que venimos juntos desde hace tiempo. La de ustedes está cerca.

A mi grupo por aguantarme todos estos años, ustedes son únicos.

A los tutores, si no es por ellos…no nos apuramos con el trabajo.

A Lázaro, cuida a mi hermana.

Ramiro.

Dedicatoria

A mi mamá, mi hermana, mi padre, mi familia y a todos mis amigos.

Reynaldo.

A mi mamá, mi hermana, mi papá y a todos mis tíos.

A mis amigos.

Ramiro.

Resumen

En la actualidad las FAR no cuentan con un osciloscopio virtual para medir las señales emitidas por algunos transmisores que poseen determinados equipos. Este trabajo tiene como objetivo elaborar el diseño de software para una aplicación de instrumentación virtual que simule un osciloscopio. Para ello se hace un estudio de las principales tecnologías que se emplean en la construcción de estos tipos de software; quedando definida la metodología, el lenguaje de modelado y las herramientas de desarrollo a utilizar.

Debido a la complejidad de este instrumento virtual se realizó el modelo de dominio correspondiente, se dividieron en módulos las clases del análisis, además de agrupar las entidades del diseño en tres subsistemas: adquisición, procesamiento y presentación de datos respectivamente; los cuales realizan el intercambio de datos a través de interfaces de comunicación.

Así, el proyecto pretende servir de soporte para la implementación del instrumento virtual. Finalmente se concluye la investigación dando respuesta al objetivo principal y recomendando una serie de aspectos a tener en cuenta para su futuro desarrollo.

PALABRAS CLAVE

Adquisición de datos, simulación, instrumento virtual, instrumentación virtual, señales, osciloscopio virtual.

Contenido

Introducción

La evolución de la investigación en el diseño, desarrollo y aplicación de los sistemas de medición evidencian una marcada tendencia hacia la automatización. En la actualidad, las computadoras personales (PCs) están cambiando la forma en que los instrumentos son construidos y utilizados. La combinación de los métodos tradicionales con las nuevas técnicas digitales, que pueden ser aplicadas a partir de la incorporación de la PC al campo de la instrumentación, permite el desarrollo de sistemas flexibles y fáciles de operar.

El constante desarrollo de las Tecnologías de la Información y las Comunicaciones (TICs) ha permitido que desde la década de los ochenta se hayan empezado a crear programas de simulación por computadora, lo cual se ha convertido en una parte útil del modelado de muchos sistemas naturales en física, química, biología y sistemas humanos como la economía y las ciencias sociales. La realización de simulaciones en física, muchas veces lleva consigo el uso de instrumentos tales como generadores de señal, osciloscopios, analizadores lógicos, analizadores de espectro, por citar algunos ejemplos. En los entornos desarrollados últimamente se incorporan módulos que simulan perfectamente estos y otros instrumentos, de tal manera que se puede realizar la medición de señales o su análisis utilizando la PC como una herramienta.

La creciente capacidad y empleo generalizado de las PC otorgan nuevas herramientas para apoyar los distintos problemas en que se ven inmersas las diversas disciplinas y áreas del mundo moderno; por tanto, se enmarca la presente investigación en la tecnología de la instrumentación virtual.

Los inicios de la instrumentación controlable desde el ordenador, y de hecho de los sistemas de instrumentación, se sitúan a finales de los años 60 cuando Hewlett Packard, desarrolló su bus para la instrumentación. Desde aquellos días hasta ahora fue sufriendo varias modificaciones y hoy se le conoce habitualmente como General Purpose Interface Bus (GPIB), nombre por el que se ha convertido en uno de los más populares en el campo de la instrumentación programable.

La industria de la instrumentación está sufriendo importantes cambios como resultado de la revolución de las PCs. Estos cambios están ocurriendo tanto en el componente hardware como en el software. Un elevado número de científicos e ingenieros en todo el mundo usan PC para automatizar sus tareas de investigación, diseño y fabricación. Este desarrollo de hardware y software ha dado lugar a que aparezca un nuevo tipo de instrumentación, que es la denominada instrumentación virtual (I.V de ahora

en lo adelante), como se mencionaba antes. El término virtual, empleado en la industria de la instrumentación, hace referencia a la combinación de instrumentos programables con las PCs. [1]

En Cuba existe una experiencia previa en el desarrollo de la I.V, tal es el caso del Sistema de Instrumentación Virtual para Prácticas de Laboratorio de Caracterización de Pequeñas Partículas en Suspensión, desarrollado por el Instituto Superior de Tecnologías y Ciencias Aplicadas (InSTEC) de La Habana. Otro ejemplo de instrumentación virtual lo constituyen los osciloscopios como el ADQO-1, desarrollado por el Instituto Superior de Ciencias y Tecnología Nucleares (ISCTN) de La Habana, cuyo funcionamiento se basa solamente en la tarjeta de adquisición de datos. Este programa fue creado para el entorno MS-DOS, debido a las prestaciones de las computadoras personales que se disponía en dicho centro docente. También hace uso de un instrumento virtual, el Centro de Estudios de Innovación y Mantenimiento del Instituto Superior Politécnico José Antonio Echeverría de La Habana, para el registro y análisis de vibraciones aplicándolo al diagnóstico mecánico.

Nuestro país no está exento del desarrollo científico técnico existente y como parte de su inmersión ha desatado un proceso de informatización que se lleva a cabo en todos sectores de la sociedad. Un ejemplo de lo antes mencionado lo constituye las Fuerzas Armadas Revolucionarias (FAR), que empleando la tecnología de software libre moderniza la técnica y el armamento para elevar la disposición combativa de estos.

Actualmente las FAR se valen de un osciloscopio para medir las señales emitidas por algunos transmisores que poseen determinados equipos, el cual, mediante marcas auxiliares en su pantalla permite conocer los principales parámetros para realizar un dictamen sobre el estado y funcionamiento de los transmisores. Estos osciloscopios empleados se encuentran obsoletos tecnológicamente, fallan con mucha frecuencia y la adquisición de nuevas tecnologías resultaría muy costosa para esta institución y el país, debido a los elevados precios en el mercado. Por estas razones es necesaria una aplicación de instrumentación virtual que represente al equipo y sus funcionalidades.

Es por la necesidad de automatizar esta técnica y la problemática presente, que se plantea como base de estudio el siguiente **problema de investigación**: ¿Cómo elaborar el análisis y diseño para desarrollar una aplicación de instrumentación virtual que simule un osciloscopio, utilizando tecnologías libres?

El Centro de Investigación y Desarrollo Número 3 (MECATRONICS) en conjunto con la Facultad 5 de la Universidad de las Ciencias Informáticas (UCI) y el Centro de Compatibilización e Integración y

Desarrollo de Productos Informáticos para la Defensa desarrollarán la presente investigación que posee como objeto de estudio el desarrollo de instrumentos virtuales de medición, de ello se deriva que el campo de acción de este trabajo lo constituyen los osciloscopios virtuales.

Para guiar la investigación y tomando en cuenta lo analizado hasta el momento se decide trabajar sobre la base de la siguiente **idea a defender**: el desarrollo de un diseño de software para simular un osciloscopio guiará hacia una correcta implementación del instrumento virtual.

Por tanto el **objetivo general** de esta investigación es elaborar el diseño de software para una aplicación de instrumentación virtual que simule un osciloscopio.

Con vista a dar respuesta a este objetivo se plantean las siguientes **tareas de investigación**:

- Consultar la bibliografía relacionada con el desarrollo de instrumentos virtuales de medición.
- Elaborar la fundamentación teórica sobre la instrumentación virtual y las principales herramientas de desarrollo.
- Analizar métodos y técnicas para la representación gráfica de señales eléctricas.
- Investigar sobre algoritmos y tecnologías utilizadas en el desarrollo de osciloscopios virtuales.
- Diseñar una solución para cumplir con el objetivo propuesto.

Capítulo 1 Fundamentación teórica

Este capítulo resume los conceptos fundamentales de la teoría de los osciloscopios, desde las generalidades hasta las especificaciones. Además comprende la descripción de las principales tecnologías que se emplean en la construcción de estos tipos de software incluyendo la metodología, el lenguaje de modelado y las herramientas de desarrollo propuestas.

1.1. Estado actual

En la actualidad existen varias empresas que se dedican al desarrollo de instrumentos virtuales de medición tal es el caso de la empresa National Instruments, una de las más destacadas a nivel mundial en la instrumentación virtual, la cual se ha usado para traer la potencia de software flexible y la tecnología de las PC para probar, controlar y diseñar aplicaciones haciendo mediciones análogas y digitales exactas de corriente directa.

Cada instrumento virtual consiste de dos partes: software y hardware. Un instrumento virtual similarmente tiene un precio accesible y muchas veces mucho menor que los instrumentos tradicionales de medición similares para una tarea actual. Sin embargo, los ahorros son compuestos a través del tiempo, debido a que los instrumentos virtuales son mucho más flexibles al cambiar las tareas de medición. A continuación se listan varias empresas desarrolladoras de instrumentos basados en PC.

Pico Technology

Pico Technology es una empresa que se fundó en 1991 e inmediatamente obtuvo una gran popularidad por el desarrollo alcanzado en el campo de los instrumentos basados en PC y adquisición de datos. Tiene la de línea productos PicoScope series 2000,3000 y 5000, donde el modelo más económico mide señales de hasta 50 Mhz (2 canales) y cuesta USD $783. Tiene un software propietario para Windows y drivers para C, Pascal, Delphi y Visual Basic. También tiene un modelo que mide hasta 200 Mhz cuyo costo de es U$S 1530. [15]

TiePie Engineering

TiePie Engineering es una empresa holandesa que desarrolla y vende instrumentos de medida controlados por computadora. Siempre ha trabajado en el mismo rubro desde su fundación, en el año 1987. Algunos de los modelos que ofrecen son la serie Handyscope4 (HS4) de 50 Mhz, 12 bit y 4 canales con comunicación USB 2.0 con precios desde USD $648 hasta USD $1380; la serie Handyscope3 (HS3) de 100 Mhz y 2 canales con precios desde USD $640 hasta USD $1108 y la serie Handyscope4 (HS4-DIF) de 50 Mhz, 12 bit y 4 canales con comunicación USB 2.0 con precios desde USD $828 hasta USD $1540. Tanto el software como el driver vienen exclusivamente para Windows. [16]

ETC

ETC es una empresa fundada en Enero de 1996 y dedicada al diseño y producción de dispositivos de medida para PC, utilizando FPGAs y CPLDs. Uno de sus productos es el M521 de 60 Mhz y 2 canales por EUR $427, mientras que el M524 de 120 Mhz cuesta EUR $607. El software es para Windows y tiene disponibles drivers para Delphi y Visual Basic con un costo adicional. [17]

Bitscope

Bitscope es una empresa australiana con firmes creencias en el software y hardware libre, y su osciloscopio es quizás el más peculiar de todos puesto que es el único que se caracteriza por ser de hardware abierto, lo cual significa que en su página se encuentran publicados todos los esquemáticos del mismo, junto con abundante documentación sobre su construcción y decisiones de diseño, como así también los protocolos usados. Todo esto permite que un usuario, a partir de la información de la página, pueda construirse el osciloscopio que ellos venden, siempre y cuando tenga los medios y conocimientos necesarios. Uno de sus osciloscopios es el BS310 de 100 Mhz y 2 canales por USD $575. El software, por supuesto, corre tanto en Windows como en Linux. [18]

Como se ha podido apreciar existen muchas empresas productoras de instrumentos virtuales, pero la mayoría son de software propietario y las aplicaciones están hechas para Windows solamente. Lo que se quieres es una aplicación propia y basada en los términos del software libre, además presentan un alto precio en el mercado, de ahí que ninguno de estos instrumentos resulta conveniente para esta institución. Es por eso que surge esta investigación, para la realización de un software con tecnologías libres.

Se ha estado hablando de instrumentos virtuales y de instrumentación virtual, pero no se ha dicho en que consiste cada uno. En el epígrafe siguiente se abordaran estos temas para lograr un mayor entendimiento de lo que se está tratando.

1.1. Instrumento virtual. Características

El concepto de instrumentación virtual nace a partir del uso de la PC como forma de reemplazar equipos físicos por software, permite a los usuarios interactuar con la computadora como si estuviesen utilizando un instrumento real.

Además de utilizar la PC como instrumento de medición de señales de corriente, implica adquisición de señales, funciones de procesamiento y análisis, almacenamiento, distribución y despliegue de los datos e información relacionados con la medición de una o varias señales, involucra la interfaz hombre-máquina, visualización, la comunicación con otros equipos, monitoreo y supervisión remota del proceso. Entonces se puede plantear que la instrumentación virtual es un sistema de medición, análisis y control de señales físicas con una computadora por medio de instrumentos virtuales.

De aquí surge el término de instrumento virtual de cual existen varias definiciones pero las que más se ajustan con lo que se esta realizando son las siguientes:

- Una capa de software y hardware que se le agrega a una PC en tal forma que permite a los usuarios interactuar con la computadora como si estuviesen utilizando su propio instrumento electrónico hecho a la medida.[3]

- Una computadora del tipo industrial, o una estación de trabajo, equipada con poderosos programas (software), hardware económico, tales como placas para insertar, y manejadores (drivers) que cumplen, en conjunto, las funciones de instrumentos tradicionales. [4]

- “Un instrumento que no es real, se ejecuta en una computadora y tienes sus funciones definidas por software.” (National Instruments, 2001).

Los instrumentos virtuales representan un apartamiento fundamental de los sistemas de instrumentación basados en el hardware a sistemas centrados en el software que aprovechan la potencia de cálculo, productividad y capacidad de conexión de las computadoras de escritorio y estaciones de trabajo. Aunque la PC y la tecnología de circuitos integrados han experimentado avances significativos en las últimas dos décadas, es el software el que realmente provee la ventaja para construir sobre esta potente base de hardware para crear los instrumentos virtuales, proveyendo mejores maneras de innovar y de reducir los costos significativamente. [3]

Con los instrumentos virtuales, se pretende construir sistemas de medición y automatización que se ajustan exactamente a las necesidades definidas por el usuario, en lugar de estar limitados por los instrumentos tradicionales de funciones fijas, definidas por el fabricante. La tendencia actual es permitir que el usuario disponga de un sistema configurable y adaptable a sus necesidades.

1.1.1. Flexibilidad

A excepción de los componentes especializados y los circuitos hallados en los instrumentos tradicionales, la arquitectura general de los instrumentos autónomos es muy similar a la hallada en un instrumento virtual basado en computadora. Ambos requieren uno o más microprocesadores, puertos de comunicación, por ejemplo: serie y GPIB, y capacidad de mostrar resultados así como también módulos de adquisición de datos. Lo que diferencia uno del otro es su flexibilidad y el hecho que se puede modificar y adaptar el instrumento a necesidades particulares. Un instrumento tradicional podría contener un circuito integrado para llevar a cabo un conjunto particular de instrucciones de procesamiento de datos; en un instrumento virtual estas funciones podrían llevarse a cabo por el programa que corre en el procesador de la computadora. Se puede fácilmente extender ese conjunto de funciones y estar sólo limitado por la potencia del software que utilice. [3]

Entonces se deduce que utilizando soluciones basadas en la instrumentación virtual, se puede reducir los costos de inversión, desarrollo de sistemas y mantenimiento, al mismo tiempo que mejora el tiempo de fabricación y la calidad de los productos.

1.1.2. Arquitectura abierta.

Se denomina así a la disponibilidad que tiene una PC de que se le conecten nuevos periféricos a través de sus zócalos de expansión. Esto le ha brindando una gran versatilidad para la configuración del sistema requerido. En este tipo de aplicación de instrumentación virtual, el periférico utilizado consiste en plaquetas de adquisición, control y procesamiento digital con el software correspondiente. El desarrollo de esta tecnología involucra el conocimiento de cuatro aspectos fundamentales: adquisición de datos, procesamiento digital, almacenamiento y presentación. Contar con este tipo de arquitectura permite aprovechar también otras ventajas, como las relacionadas con la distribución y transferencia de los datos e información provenientes de una medición entre diferentes equipos a través de redes, lo que sumado a sus características de bajo costo, alto poder de procesamiento, capacidad de almacenamiento y calidad de graficación de la PC, ha provocado su incorporación en forma definitiva como unidad central alrededor de la cual se desarrollan las distintas aplicaciones que forman el sistema completo de medición. [5]

1.2. Osciloscopios

Hay muchos artefactos de medidas capaces de cuantificar diferentes magnitudes. Por ejemplo, el voltímetro mide tensiones, el amperímetro intensidades y el vatímetro potencia. Pero, sin duda alguna, el más importante que se conoce es el osciloscopio. Con él, no sólo podemos averiguar el valor de una magnitud, sino que, entre otras cosas, se puede saber la forma que tiene dicha magnitud, es decir, podemos obtener la gráfica que la representa.

1.2.1. ¿Qué es un osciloscopio?

Un osciloscopio es un instrumento de medición electrónico para la representación gráfica de señales eléctricas que pueden variar en el tiempo. [2] La forma de trabajo de un osciloscopio consiste en dibujar una gráfica, que no es más que una curva que tiene dos ejes de referencia, el denominado de abscisas u horizontal y el eje de ordenadas o vertical. Para representar cada punto de la gráfica tememos que dar dos coordenadas, una va a corresponder a su posición respecto al eje horizontal y la otra va a ser su posición respecto al en el vertical [6]

Los osciloscopios, clasificados según su funcionamiento interno, pueden ser tanto analógicos como digitales, siendo el resultado mostrado idéntico en cualquiera de los dos casos, teóricamente. Un osciloscopio tradicional, como el que se muestra en la Figura 1.1, tiene una funcionalidad ya predefinida en su diseño, producción y ensamblado, no se pueden modificar los controles de operación y presentación una vez salidos de fábrica.

Figura 1.1 Ejemplo de osciloscopio tradicional

Con el osciloscopio se puede:

- Determinar directamente el período y el voltaje de una señal.
- Determinar indirectamente la frecuencia de una señal.
- Determinar que parte de la señal es DC y cual AC.
- Localizar averías en un circuito.
- Medir la fase entre dos señales.
- Determinar que parte de la señal es ruido y como varia este en el tiempo.

1.2.2. Osciloscopio virtual

Un osciloscopio virtual queda definido en función del software y hardware que se le agrega a una PC, es el que permite al usuario configurar por medio de la computadora su propio instrumento electrónico, a través de interruptores virtuales que se encuentran en la pantalla y emulan los controles reales. El hardware consiste en una placa de adquisición de datos con su software apropiado. En la Figura 1.2 se indica el diagrama en bloques de este instrumento virtual. Se observa que el resultado final de este proceso es la presentación en pantalla de las magnitudes medidas por el sistema.

En este instrumento, el software es la clave del sistema, a diferencia del instrumento tradicional, donde la clave es el hardware. Se puede construir un osciloscopio personalizado, con la interfaz gráfica que se desee, lo que le agrega inclusive más funcionalidad.

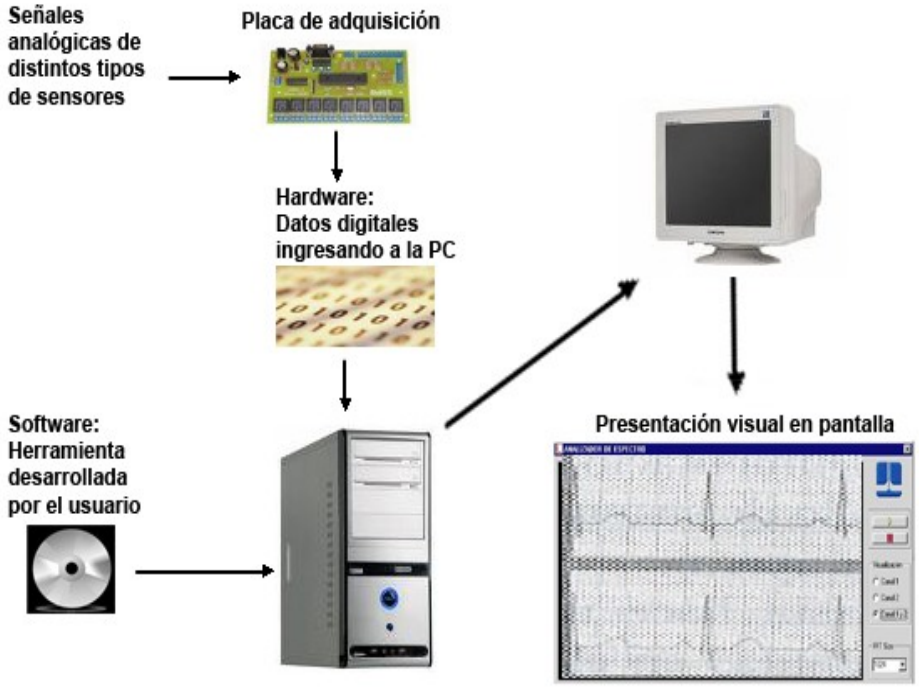

Figura 1.2 Esquema de un instrumento virtual basado en PCs.

Los principales beneficios del instrumento virtual, son su gran flexibilidad y adaptabilidad a las condiciones requeridas por las exigencias del sistema de medición y monitoreo necesarios.

1.3. Señales eléctricas en los osciloscopios

La representación de la señal nos proporciona una valiosa información. En cualquier momento se puede visualizar la altura que alcanza y, por lo tanto, saber si el voltaje ha cambiado en el tiempo, si se observa, por ejemplo, una línea horizontal se puede concluir que en ese intervalo de tiempo la señal es constante. Con la pendiente de las líneas diagonales se puede conocer la velocidad en el paso de un nivel a otro, se pueden observar también cambios repentinos de la señal: ángulos muy agudos, generalmente debidos a procesos transitorios. [9]

1.3.1. Tipos de señales

Se pueden clasificar las señales en los tres tipos siguientes:

- Señales senoidales.
- Señales cuadradas.
- Señales triangulares y en diente de sierra.

Señales senoidales

Son las señales fundamentales, pues poseen unas propiedades matemáticas muy interesantes, por ejemplo, con combinaciones de señales senoidales de diferente amplitud y frecuencia se puede reconstruir cualquier señal, la señal que se obtiene de las tomas de corriente de cualquier casa tienen esta forma, las señales de prueba producidas por los circuitos osciladores de un generador de señal son también senoidales, la mayoría de las fuentes de potencia en corriente alterna (AC) producen señales senoidales. [9]

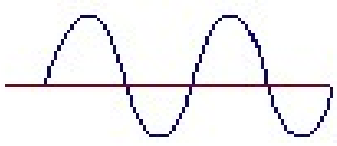

Figura 1.3 Ejemplo de una señal senoidal.

Señales cuadradas

Las señales cuadradas son básicamente señales que pasan de un estado a otro de tensión, a intervalos regulares, en un tiempo muy reducido. Son utilizadas usualmente para probar amplificadores. Esto es debido a que este tipo de señales contienen en sí mismas todas las frecuencias. La televisión, la radio y los ordenadores utilizan mucho este tipo de señales, fundamentalmente como relojes y temporizadores.[9]

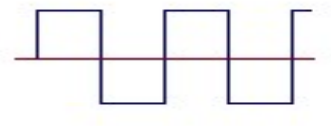

Figura 1.4 Ejemplo de una señal cuadrada.

Señales triangulares y en diente de sierra

Se producen en circuitos diseñados para controlar voltajes linealmente, como pueden ser, por ejemplo, el barrido horizontal de un osciloscopio analógico o el barrido tanto horizontal como vertical de una televisión. Las transiciones entre el nivel mínimo y máximo de la señal cambian a un ritmo constante. Estas transiciones se denominan rampas.

La señal en diente de sierra es un caso especial de señal triangular con una rampa descendente de mucha más pendiente que la rampa ascendente. [9]

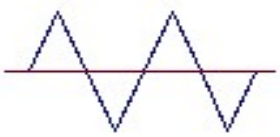

Figura 1.5 Ejemplo de una señal triangular

Figura 1.6 Ejemplo de una señal en diente de sierra

1.3.2. Caracterización de las señales E/S.

Los datos que resultan del muestreo son los que deben ser representados por la aplicación de software, estos llegan a la PC por el puerto USB y el puerto Serie (utilizando el protocolo RS-232).

Las señales a recibir tienen las siguientes características:

Tabla 1.1 Características de las señales

Puerto	Denominación	Tipo de señal	Rango	U.M.	F.Adquisición
USB	Señales rápidas	Analógica antes de muestrearla	Se muestrea una señal con escala de 0 a +20v de amplitud y de 0 a 1000μks de tiempo	[amplitud]= volts [tiempo]= μks	Se recibe cada 500μks
USB	Señales lentas	Analógica antes de muestrearla	Se muestrea una señal con escala de 0 a +20v de amplitud y de 0 a 0.5ms duración.	[amplitud]= volts [tiempo]= ms	Se recibe cada 50ms
Serie	Señal de retardo	Digital	Se muestra con caracteres alfanuméricos	volt	Se recibe cada 500μks

U.M: unidad de medida.

F. Adquisición: frecuencia de adquisición.

¿Qué parámetros se le miden a estas señales?

Los parámetros que se le miden a estas señales son los siguientes:

Amplitud de la señal: Es la magnitud, por el eje vertical, que se le mide desde la parte inferior de la señal hasta el punto donde su valor es máximo.

Duración de la señal: Es la magnitud, por el eje horizontal, que se mide desde el punto donde se intercepta la línea que representa el nivel 0,5 de la amplitud de la señal con la línea de subida de la señal, hasta la intersección del mismo nivel 0,5 con la línea de caída de la señal.

Duración del frente delantero de la señal: Es la magnitud, por el eje horizontal, que se mide desde el punto donde se intercepta la línea que representa el nivel 0,1 de la amplitud de la señal con la línea de subida de la señal, hasta la intersección de la misma línea de subida de la señal con el nivel 0,9 de la amplitud de la señal.

Duración del frente trasero de la señal: Es la magnitud, por el eje horizontal, que se mide desde el punto donde se intercepta la línea que representa el nivel 0,9 de la amplitud de la señal con la línea de caída de la señal, hasta la intersección de la misma línea de caída de la señal con el nivel 0,1 de la amplitud de la señal.

Deformación de la señal δ_u): Es la medida, expresada en por ciento, que se emplea para evaluar el deterioro del aspecto de la señal. Esta deformación se calcula mediante la expresión:

$$\delta_u = \frac{U_2 - U_1}{U_2 + U_1} * 100$$

Donde U1 y U2 son los valores que alcanza una misma señal en los extremos mínimo y máximo de su amplitud respectivamente.

1.3.3. Métodos de muestreo de señales

Se trata de explicar cómo se las arreglan los osciloscopios digitales para reunir los puntos de muestreo. Para señales de lenta variación, los osciloscopios digitales pueden perfectamente reunir más puntos de los necesarios para reconstruir posteriormente la señal en la pantalla. No obstante, para señales rápidas, como de rápidas dependerá de la máxima velocidad de muestreo del equipo se posea, el osciloscopio no puede recoger muestras suficientes y debe recurrir a una de estas dos técnicas:

Interpolación: estimar un punto intermedio de la señal basándose en el punto anterior y posterior.

Muestreo en tiempo equivalente: Si la señal es repetitiva es posible muestrear durante unos cuantos ciclos en diferentes partes de la señal para después reconstruir la señal completa.

Muestreo en tiempo real con interpolación

El método estándar de muestreo en los osciloscopios digitales es el muestreo en tiempo real en el cual el osciloscopio reúne los suficientes puntos como para reconstruir la señal. Para señales no repetitivas o la parte transitoria de una señal es el único método válido de muestreo. Este método de muestreo es el más simple de todos y es el único que permite digitalizar completamente señales no periódicas y transitorias. Cada muestra y el tiempo en que fue tomada tienen una correspondencia directa con su equivalente en tiempo real. A mayor tasa de muestreo en comparación al ancho de banda de la señal, se obtiene una mayor definición en el resultado.

Los osciloscopios utilizan la interpolación para poder visualizar señales que son más rápidas que su velocidad de muestreo. Existen básicamente dos tipos de interpolación:

Lineal: Simplemente conecta los puntos muestreados con líneas.

Senoidal: Conecta los puntos muestreados con curvas según un proceso matemático, de esta forma los puntos intermedios se calculan para rellenar los espacios entre puntos reales de muestreo. Usando este proceso es posible visualizar señales con gran precisión disponiendo de relativamente pocos puntos de muestreo. [10]

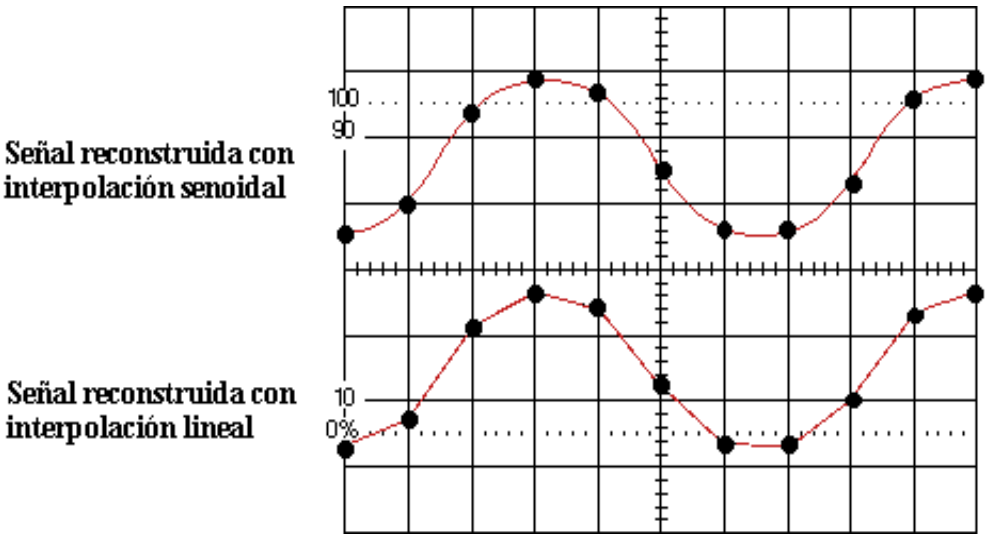

Figura 1.7 Aplicación de la interpolación en la reconstrucción de señales

Tiempo equivalente

El muestreo en tiempo equivalente cuenta con las siguientes características:

- solo para señales periódicas
- la señal se va construyendo en barridos sucesivos
- 3 tipos de barridos:
    - aleatorio repetitivo
    - secuencial
    - submuestreo

Desventajas:

- No es apto para mediciones de señales no periódicas y de alta velocidad.
- No puede medir transitorios.

Por lo tanto no es aconsejable utilizar este método de muestreo y se eligió el método de muestreo en tiempo real con interpolación senoidal.

1.4. Instrumentos Virtuales vs. Instrumentos Tradicionales

Los instrumentos tradicionales, tales como osciloscopios y generadores de ondas, son muy poderosos, caros y diseñados para llevar a cabo una o más tareas específicas definidos por el fabricante. Sin embargo, el usuario por lo general no puede extender o personalizar esas tareas. Los botones del instrumento, sus circuitos electrónicos y las funciones disponibles para el usuario son todas específicas a la naturaleza del instrumento. Además, deben desarrollarse una tecnología especial y costosos componentes para construirlos, lo cual los hace muy caros y lentos para adaptarlos.

Debido a que están basados en la PC, los instrumentos virtuales aprovechan inherentemente los beneficios de la última tecnología de las computadoras personales corrientes. Estos avances en tecnología y rendimiento, que están cerrando rápidamente la brecha entre los instrumentos autónomos y las PCs, incluyen poderosos procesadores, tales como el Pentium 4 y sistemas operativos tales como el Microsoft Windows XP, Linux y otros. Los instrumentos tradicionales también adolecen frecuentemente de falta de portabilidad, en tanto que los instrumentos virtuales que corren en las computadoras portátiles automáticamente incorporan esta naturaleza portátil. [3]

Poseer un osciloscopio virtual trae grandes ventajas, es por ello que las FAR quiere reemplazar el osciloscopio existente (tradicional) por un sistema automatizado adaptado a una PC, ya que se puede adaptar un instrumento virtual a necesidades particulares de un cliente sin necesidad de reemplazar todo el instrumento, dado que posee el software de aplicación instalado en la computadora y al amplio rango disponible de hardware para instalar en ella. La Tabla 1 resume las principales diferencias entre el instrumento convencional o tradicional, y el instrumento virtual.

Tabla 1.2 Comparación entre instrumentos tradicionales y virtuales

Instrumento Tradicional	**Instrumento Virtual**
Definido por el fabricante.	Definido por el usuario.
Funcionalidad específica, con conectividad limitada.	Funcionalidad ilimitada, orientado a aplicaciones, conectividad amplia.
Hardware es la clave.	Software es la clave.
Alto costo/función.	Bajo costo/función, variedad de funciones, reusable.
Arquitectura "cerrada”.	Arquitectura "abierta".

Lenta incorporación de nuevas tecnologías.	Rápida incorporación de nuevas tecnologías, gracias a la plataforma PC.
Bajas economías de escala, alto costo de mantenimiento.	Altas economías de escala, bajos costos de mantenimiento.

Reemplazar los instrumentos tradicionales por instrumentos virtuales, permite que las funciones de los mismos vayan a la par del desarrollo de las nuevas tecnologías.

1.5. Metodologías de desarrollo de software

Esta sección describe los conceptos fundamentales en el desarrollo de software, incluyendo la metodología Rational Unified Process (RUP) y el Lenguaje Unificado de Modelado (UML). El Proceso Unificado es un marco de trabajo genérico que puede especializarse para gran variedad de sistemas. [7]

Las principales características de RUP son:

- centrado en la arquitectura
- dirigido por casos de uso
- iterativo e incremental

RUP define cuatro fases para el desarrollo del software: inicio, elaboración, construcción y transición. Además define un grupo de flujos de trabajo básicos: Modelamiento del Negocio, Requerimientos, Análisis y Diseño, Implementación, Prueba y Despliegue; incluye flujos de trabajo de apoyo como Gestión del Cambio y la Configuración, Gestión del Proyecto y Ambiente.

UML [8] (Unified Modeling Language) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Actualmente UML ofrece un estándar para describir un "plano" del sistema incluyendo aspectos conceptuales como procesos del negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software utilizados.

Es importante destacar que UML es un lenguaje de modelado, no un método o un proceso, se emplea para definir, detallar y documentar los artefactos de un sistema de software. En la última versión se adicionaron diversas novedades que resuelven carencias desde el punto de vista práctico fundamentalmente. Entre los diagramas que propone UML para modelar un sistema encontramos:

- diagramas de estructura(clases, componentes, objetos, despliegue. paquetes)
- diagramas de comportamiento(actividades, casos de uso, estado)
- diagramas de interacción(secuencia, comunicación)

1.6. Herramientas de desarrollo propuestas

1.6.1. Toolkit gráfico

GTK+

GTK+ [11] son las siglas de GIMP Toolkit, es una biblioteca que permite crear interfaces gráficas de usuario, se distribuye bajo la licencia pública general (GPL), lo que hace de GTK+ un producto completamente libre.

GTK+ es multiplataforma, se ha extendido hasta Microsoft Windows y muchos derivados de Unix, como Linux y Mac OS. Está escrita en C y tiene extensiones en varios lenguajes como C++, Python, Perl y muchos más.

Se empleó inicialmente en el proyecto GNU Image Manipulation Program Tool Kit, por eso su nombre: GTK+, se ha extendido rápidamente por su estabilidad y una de las implementaciones que más se ha usado es la de C++, llamada Gtkmm [12].

Qt

Se propone utilizar el toolkit gráfico Qt, pues es un framework escrito en C++ que permite crear aplicaciones con interfaces gráficas. Qt es totalmente orientado a objetos, fácil de usar, extensible y multi-plataforma (soportada en Windows, Unix y derivados). Además se cuenta con una buena documentación para la preparación de los programadores. [13]

Qt es un producto creado por la compañía Trolltech, esta comercializa una versión de Qt no libre para los sistemas Windows, además distribuye una versión completamente libre y gratuita para los sistemas Unix, Linux y derivados, que se distribuye bajo licencia GPL y QPL[14] , que está aprobada por la Fundación de Software Libre (FSF).

Qt provee mecanismos para la visualización de imágenes vectoriales y raster, utilizando widgets para hacer el render y el procesamiento de las operaciones principales: zoom, escala, rotación y traslación. Provee también mecanismos para la edición de las imágenes y los objetos gráficos en general, entre ellos cambio de brocha, pincel, estilo de los textos, terminaciones de brocha, estilos de línea entre otros.

Entorno de desarrollo integrado (IDE)

Un entorno de desarrollo integrado (IDE) muy utilizado es Eclipse, un proyecto de software libre que ha ido creciendo gracias a la fuerte comunidad que lo desarrolla. Eclipse es un IDE multiplataforma desarrollado por IBM, en la actualidad lo mantiene la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos, capacidades y servicios complementarios.

Inicialmente Eclipse se desarrolló para los programadores que usaban el lenguaje Java, presentaba varios componentes como: la Plataforma Principal (inicio y ejecución de plug-ins), es Standar Widget Toolkit (portables widgets) y el Workbench (vistas, editores, perspectivas y asistentes). Actualmente el IDE emplea módulos (plug-ins) para adicionar funcionalidades según las necesite el desarrollador, a diferencia de otros entornos monolíticos que las tienen todas incluidas, sean necesarias o no. Gracias a estos plug-ins se ha extendido el soporte de Eclipse hasta lenguajes como C/C++, Phyton y PHP entro otros, además permite utilizar lenguajes de procesado de texto como LaTeX, aplicaciones de red como Telnet, Sistemas de Gestión de Bases de Datos (DBMS). Para la gestión de la configuración y el control de versiones tiene msoporte para CVS y Subversion, incluye plug-ins para realizar pruebas de unidad (Junit, CxxTest).

Actualmente el Proyecto Eclipse exhibe su versión 3.2, y entre las características fundamentales que presenta encontramos:

- editor de textos
- resaltado de sintaxis para varios lenguajes de programación
- compilación en tiempo real
- soporte para pruebas unitarias
- integración con sistemas de control de versiones

Por último el Proyecto Eclipse esta compuesto por varios sub-proyectos muy bien definidos y aceptados por su comunidad de usuarios, por solo citar algunos encontramos:

- Plataforma de Herramientas para Pruebas y Desempeño
- Plataforma de Herramientas Web

- Edición Visual
- UML2
- Plataforma de Herramientas de Datos
- Plataforma de Desarrollo de Software para Dispositivos

Como herramienta CASE se propone utilizar el Visual Paradim, pues cumple con las características de la metodología de desarrollo de software RUP, mencionadas anteriormente.

1.7. Otra herramienta

Existen otras herramientas para el desarrollo de este tipo de software, que indudablemente disminuiría el tiempo de elaboración del producto y la calidad seria considerablemente buena, pero por sus características privativas y sus altos costos, resulta de difícil adquisición. Tal es el caso de LabView (Laboratory Virtual Engineering Workbench), que es un revolucionario entorno de desarrollo gráfico con funciones integradas para realizar adquisición de datos, control de instrumentos, análisis de medida y presentaciones de datos. Está basado en un nuevo sistema de programación gráfica, llamado lenguaje G. Es un programa enfocado hacia la instrumentación virtual, por lo que cuenta con numerosas herramientas de presentación, en gráficas, botones, indicadores y controles, los cuales son muy esquemáticos y de gran elegancia. Estos serían complicados de realizar en bases como C++ donde el tiempo para lograr el mismo efecto sería muchas veces mayor. Cuenta con librerías especializadas para manejos de Data Acquisition System (DAQ), redes, comunicaciones, análisis estadístico, comunicación con bases de datos, etc.

A pesar de toda esta gama de funcionalidades y facilidades no se puede elegir LabView como herramienta para desarrollar la aplicación, pues se esta tratando de que toda herramienta desarrollada en nuestro país cumpla con las requerimientos de software libre.

También se pudo haber utilizado Open Modélica, herramienta que es una implementación libre del lenguaje Modélica, que incluye un Plugin del entorno de desarrollo Eclipse donde la compilación y simulación están basadas en un ambiente de software libre, además de que distribuye el código binario fuente. No se utilizo porque no es aún una implementación completa.

Capítulo 2 Descripción de la solución propuesta

Para llevar a cabo el desarrollo de un instrumento virtual de medición fue necesario dividir este proceso en varios módulos o subprocesos. En este capítulo se refleja la solución que se propone para resolver el problema científico identificado. Se hace una descripción de los módulos que se utilizan en la construcción del software. Se incluyen aspectos del análisis que sirven de punto de partida para la elaboración del diseño del sistema de instrumentación virtual.

2.2. Modelo del dominio

Este modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema:

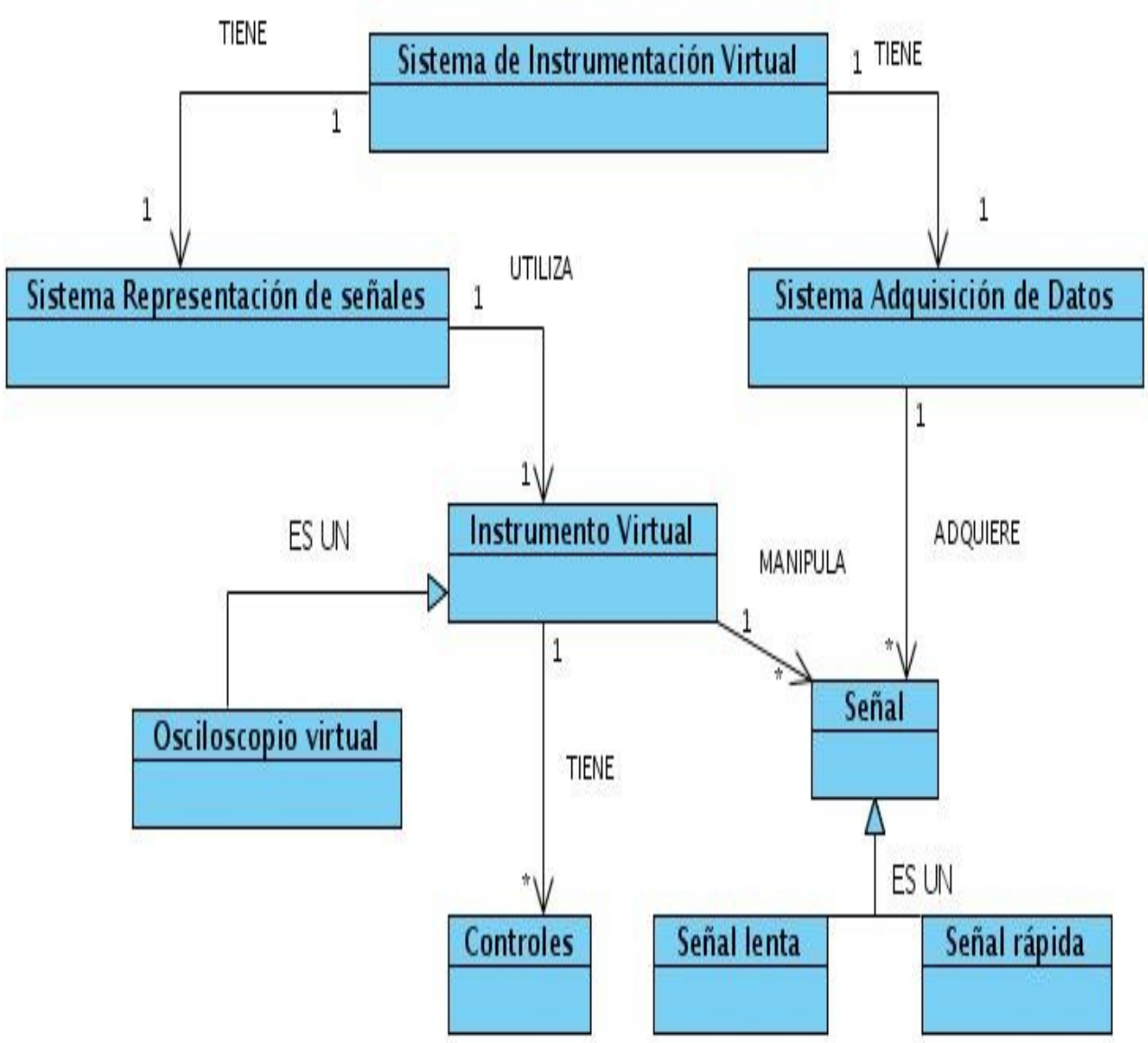

Figura 2.1 Modelo del dominio

2.1.1. Glosario de términos del dominio

Sistema de instrumentación virtual: sistema de medición, análisis y control de señales físicas con una computadora por medio de instrumentos virtuales.

Sistema de adquisición de datos: comprende todas aquellas soluciones de aplicación para referirse a la captura de información.

Instrumento virtual: un instrumento que no es real, se ejecuta en una computadora y tienes sus funciones definidas por software. (National Instruments, 2001).

Osciloscopio virtual: es un instrumento virtual que permite al usuario configurar por medio de la computadora su propio instrumento electrónico, a través de interruptores virtuales que se encuentran en la pantalla y emulan los controles reales.

Controles: botones de control del instrumento virtual.

Señal: Conjunto de ondas propagadas a lo largo de un canal de transmisión y que sirven para actuar sobre un dispositivo receptor.

Señal rápida: señal cuya presencia dura un tiempo de pocos microsegundos.

Señal lenta: señal cuya presencia dura un tiempo de varios milisegundos.

Duración del frente delantero de la señal: Es la magnitud, por el eje horizontal, que se mide desde el punto donde se intercepta la línea que representa el nivel 0,1 de la amplitud de la señal con la línea de subida de la señal, hasta la intersección de la misma línea de subida de la señal con el nivel 0,9 de la amplitud de la señal.

Duración del frente trasero de la señal: Es la magnitud, por el eje horizontal, que se mide desde el punto donde se intercepta la línea que representa el nivel 0,9 de la amplitud de la señal con la línea de caída de la señal, hasta la intersección de la misma línea de caída de la señal con el nivel 0,1 de la amplitud de la señal.

Deformación de la señal (δu): Es la medida, expresada en por ciento, que se emplea para evaluar el deterioro de l aspecto de la señal.

Duración de la señal: Es la magnitud, por el eje horizontal, que se mide desde el punto donde se intercepta la línea que representa el nivel 0,5 de la amplitud de la señal con la línea de subida de la señal, hasta la intersección del mismo nivel 0,5 con la línea de caída de la señal.

2.1.2. Definición de los actores del negocio

Tabla 2.1 Actores del negocio

Actores del negocio	Justificación
Operador	Es quien inicia los procesos de: • Amplificar señal de entrada. • Medir retardos de tiempo • Establecer marcas • Seleccionar nivel medio

2.1.3. Diagrama de casos de uso del negocio

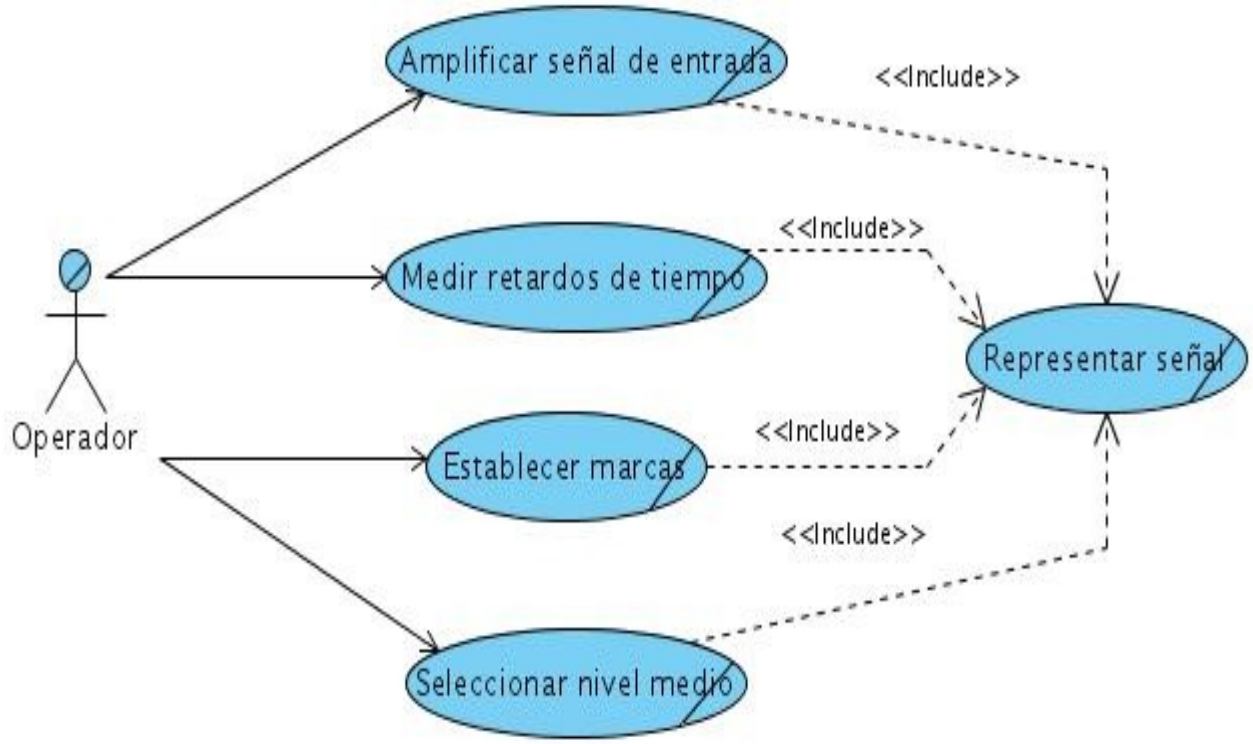

Figura 2.2 Diagrama de casos de uso del negocio.

2.1.4. Descripción textual de los casos de uso del negocio.

Tabla 2.2 Descripción textual CU-Amplificar señal de entrada

Caso de Uso:	Amplificar señal de entrada	
Actor:	Operador(inicia)	
Resumen:	El procedimiento de amplificar una señal comienza cuando se selecciona la posición de Amplificación, que permite el paso de la señal por un amplificador para que se vea más grande en pantalla.	
Flujo Normal de Eventos		
Acción del Actor		**Respuesta del Negocio**
1- El caso de uso inicia cuando el especialista selecciona la opción Amplificación en el botón de amplificar señal.		2- Pasa la señal por un amplificador para que se vea más grande.
		3- Visualiza en pantalla la señal con mayor tamaño.
4- Observa la señal amplificada.		
Prioridad	Crítico	

Diagrama de Actividades <Amplificar señal>

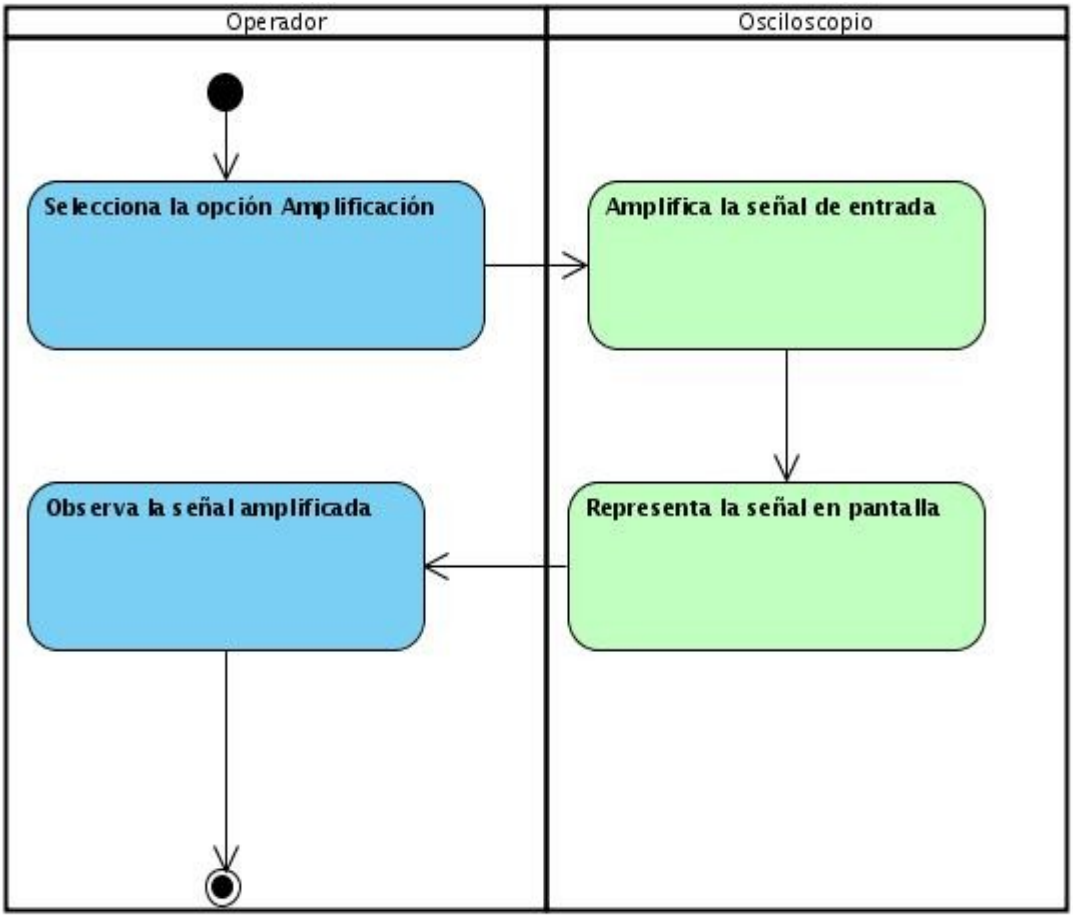

Figura 2.3 Diagrama de actividades del CU-Amplificar señal

Tabla 2.3 Descripción textual CU-Medir tiempo

Caso de Uso:	Medir retardos de tiempo	
Actor:	Operador(inicia)	
Resumen:	El procedimiento de medir retardos de tiempo comienza cuando se mueve una marca en la pantalla del osciloscopio con la cual es posible medir retardos de tiempo entre las señales que se están observando con precisión de 0.01µks.	
Flujo Normal de Eventos		
Acción del Actor		**Respuesta del Negocio**
1- El caso de uso inicia cuando se selecciona la cantidad de tiempo a medir.		2- Mueve una marca en la pantalla del osciloscopio.
		3- Mide los retardos de tiempo entre las señales, según las marcas establecidas.
4- Observa los resultados y evalúa.		
5- Emite un dictamen.		
Prioridad	Critico	

Diagrama de Actividades < Medir tiempo>

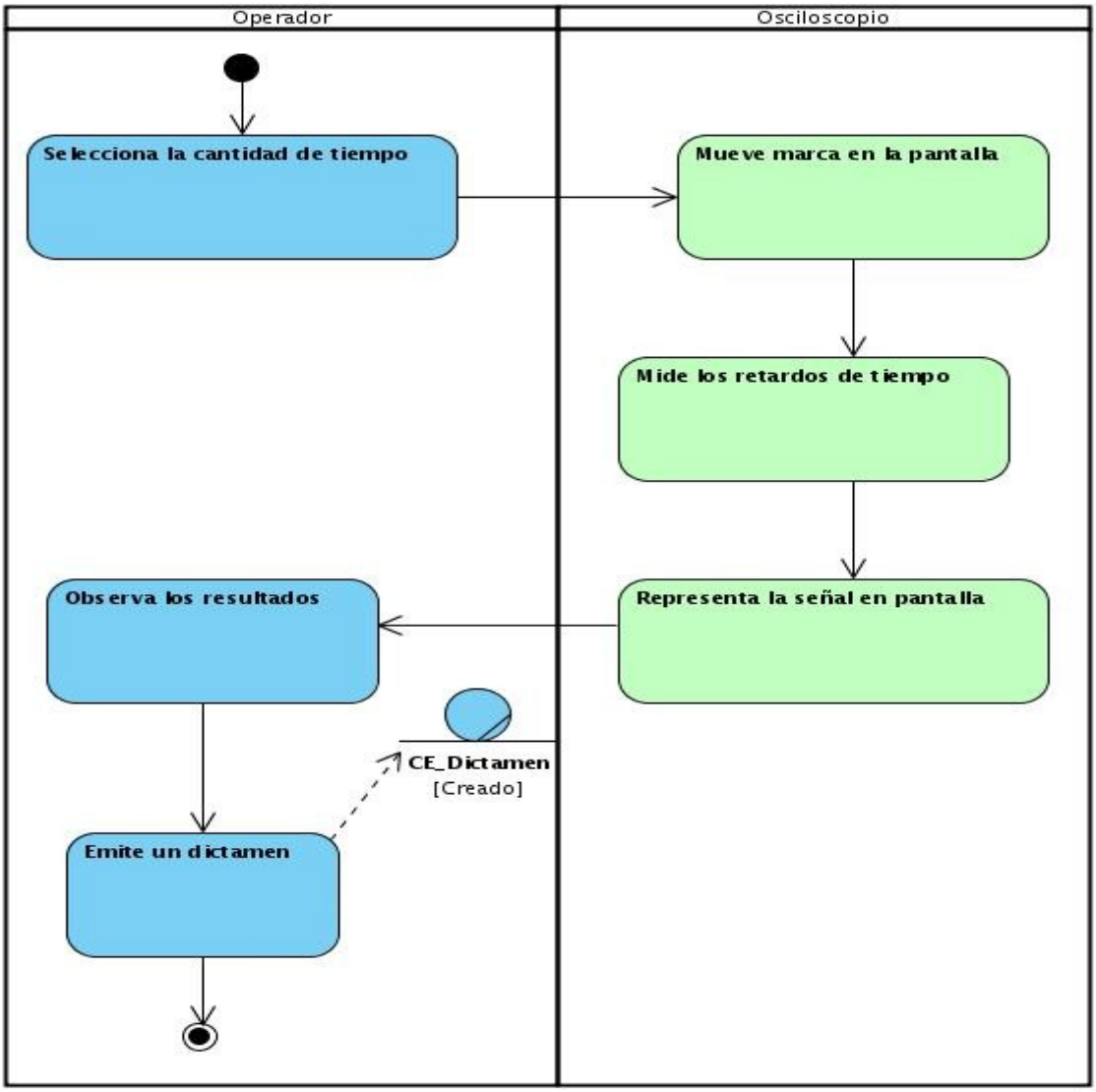

Figura 2.4 Diagrama de actividades del CU-Medir tiempo

Tabla 2.4 Descripción textual CU-Establecer marcas.

Caso de Uso:	Establecer marcas	
Actor:	Operador(inicia)	
Resumen:	El procedimiento de establecer marcas comienza cuando se acciona el interruptor de "conexión de marcas de 1μks", con este se establecen unas marcas en la pantalla del osciloscopio, que están espaciadas cada 1μks. Esto permite realizar mediciones de retardo entre señales.	
Flujo Normal de Eventos		
Acción del Actor		**Respuesta del Negocio**
1- El caso de uso inicia cuando se acciona el interruptor de conexión de las marcas de 1μks.		2- El osciloscopio posiciona las marcas según lo elegido por el especialista.
3- Observa las marcas.		
4-Realiza el dictamen y concluye el caso de uso.		
Prioridad	Crítico.	

Diagrama de Actividades < Establecer marcas

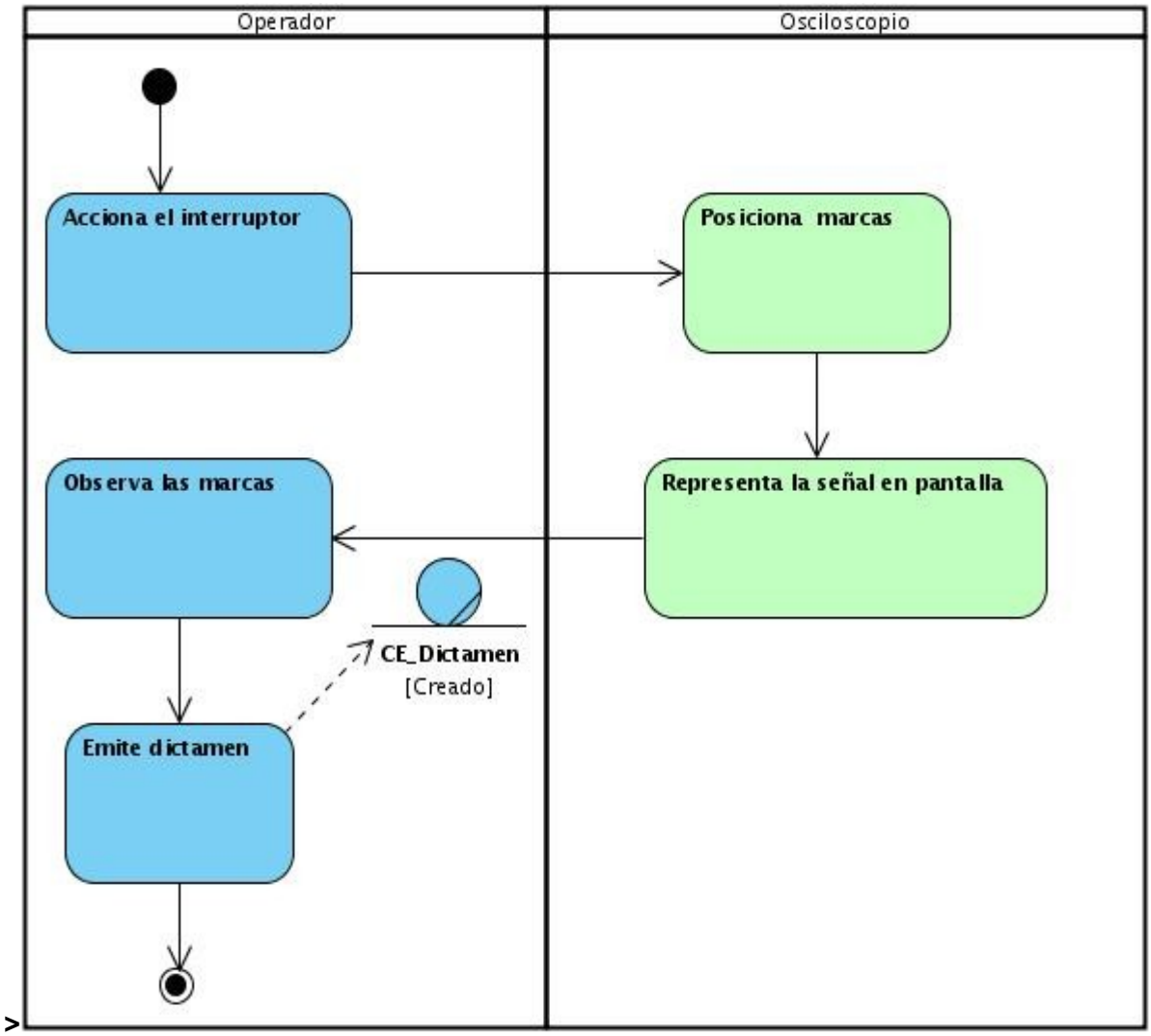

Figura 2.5 Diagrama de actividades del CU-Establecer marcas

Tabla 2.5 Descripción textual CU-Seleccionar nivel medio

Caso de Uso:	Seleccionar nivel medio.	
Actor:	Operador(inicia).	
Resumen:	El procedimiento de seleccionar nivel medio comienza cuando se acciona el interruptor selección del nivel medio, con el cual se desplaza una señal por la vertical, quedando marcada por el nivel medio de su amplitud.	
Flujo Normal de Eventos		
Acción del Actor		**Respuesta del Negocio**
1- El caso de uso inicia cuando el especialista selecciona el interruptor “Selección del nivel medio”.		2- Se desplaza la señal por la vertical, quedando marcada por la amplitud.
3- Observa la señal modificada.		
4-Realiza el dictamen y concluye el caso de uso.		
Prioridad	Crítico	

Diagrama de Actividades < Seleccionar nivel medio>

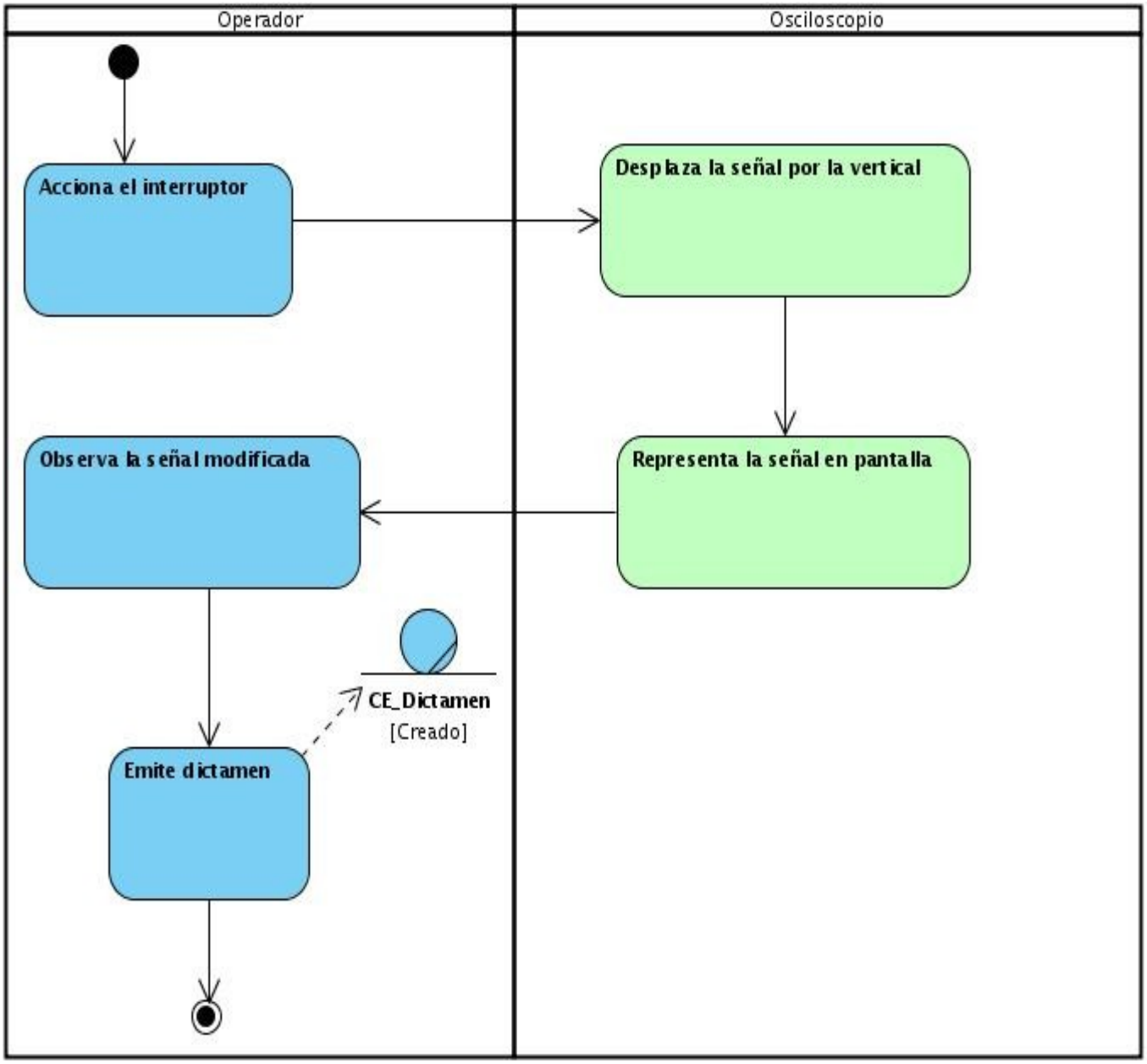

Figura 2.6 Diagrama de actividades del CU-Seleccionar nivel medio

Tabla 2.6 Descripción textual CU-Representar señal

Caso de Uso:	Representar señal	
Actor:	Operador(inicia)	
Resumen:	El procedimiento de representar una señal comienza cuando se tienen las muestras de señal y se decide realizar una representación de las mismas.	
Flujo Normal de Eventos		
Acción del Actor		**Respuesta del Negocio**
1- El caso de uso inicia cuando el especialista conecta la sonda del osciloscopio a un circuito.		2- La señal atraviesa la sonda y se dirige a la sección vertical.
		3- La sección vertical ajusta la amplitud de la señal e inicia el barrido vertical.
		4- La sección horizontal determina la toma de una señal.
		5- La sección de disparo inicia el barrido horizontal.
		6- Representa la señal en pantalla.
7- Observa la gráfica de la señal y concluye el caso de uso.		
Cursos Alternos		
Línea **2**: Si no hay señal externa, la señal de barrido obligará al punto a trazar una línea horizontal en la pantalla del osciloscopio.		
Prioridad		Critico

Diagrama de Actividades < Representar señal>

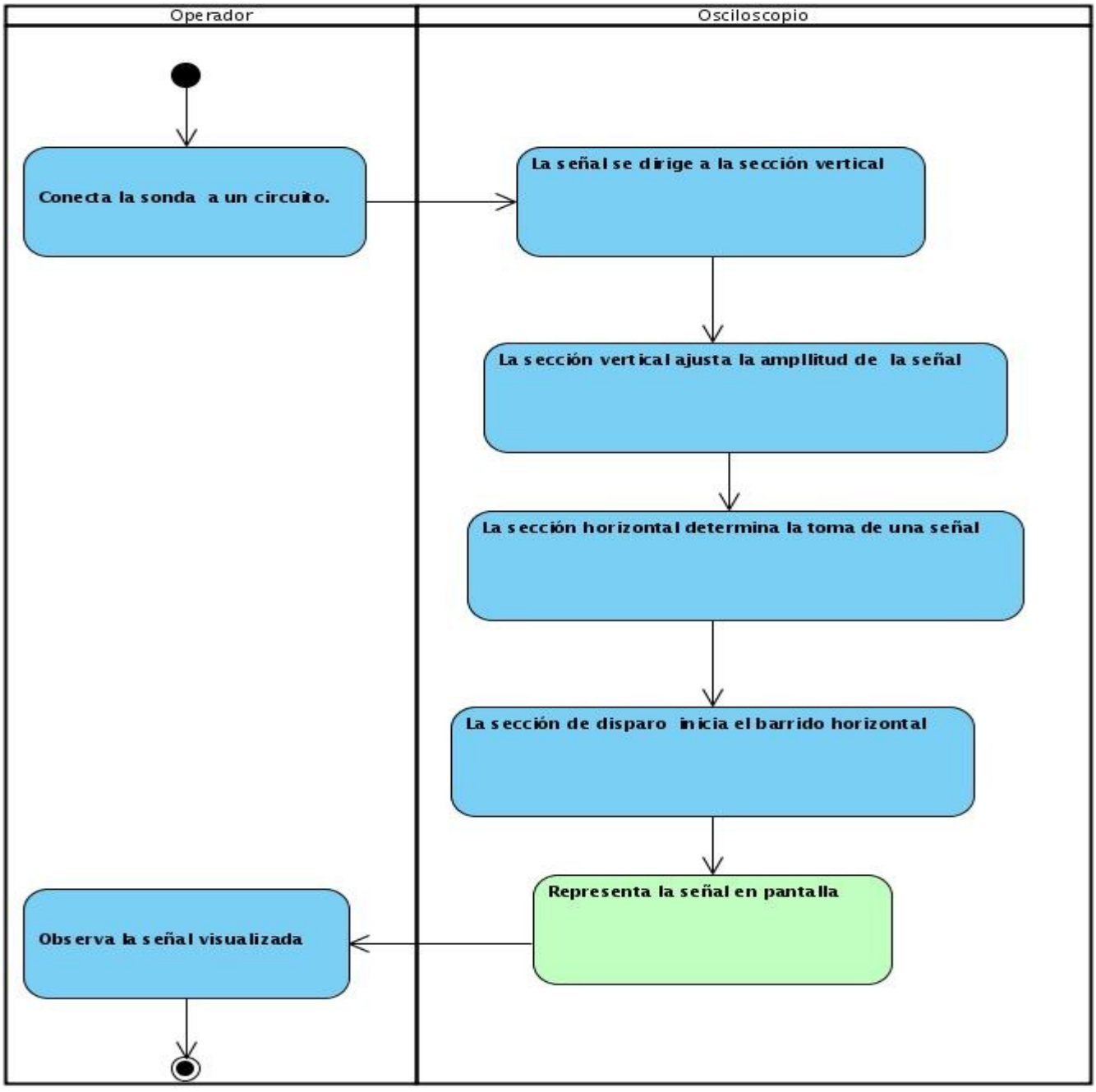

Figura 2.7 Diagrama de actividades del CU-Representar señal

2.2. Reglas del negocio

- La velocidad de transmisión de los datos debe ser en tiempo real.
- Los datos que resultan del muestreo serán los representados por la aplicación.
- Los datos llegaran a la PC por los puertos USB y serie (RS232).
- El ancho de banda debe ser como mínimo 4Mhz.
- La frecuencia de muestreo del hardware debe ser 100MS/s.

2.3. Especificación de los requisitos de software.

2.3.1. Requisitos funcionales

1. **R1** <Representar señal>
   1.1. Adquirir señal
   1.2. Procesar señal.
2. **R2** <Amplificar señal de entrada>
   2.1. Seleccionar cantidad de amplificación.
   2.2. Cambiar escala de los ejes.
   2.3. Procesar la señal de entrada
   2.4. Representar la señal.
3. **R3** <Medir tiempo >
   3.1. Seleccionar cantidad de tiempo.
   3.2. Establecer tiempo
   3.3. Procesar la señal de entrada
   3.4. Representar la señal.
4. **R4** <Establecer marcas>
   4.1. Establecer marcas en la pantalla.
   4.2. Calcular los parámetros de la señal
   4.3. Mostrar los valores de los parámetros.
5. **R5** <Seleccionar nivel medio>
   5.1. Calcular la amplitud de la señal.
   5.2. Establecer nivel medio de la amplitud.
6. **R6** <Imprimir>
7. **R7** <Guardar>
8. **R8** <Tabular>

2.3.2. Requisitos no funcionales

Rendimiento.

- Haga uso óptimo de la memoria.

Portabilidad.

- Multiplataforma.

Legales

- El sistema se basa en el manual de normas y principios establecidos por el MINFAR.

Usabilidad.

- Va a contar con Ayuda y especificaciones de uso.
- El sistema debe ser de fácil manejo para los usuarios que tengan niveles básicos sobre la computación o hayan realizado algún trabajo previo con aplicaciones desktop.

Interfaz y Apariencia

- La interfaz a diseñar debe ser sencilla, de fácil uso y con rápida respuesta del sistema.

2.4. Definición de los casos de uso del sistema

2.4.1. Definición de los actores

Tabla 2.7 Definición de actores del sistema

Actores	Justificación
Especialista	Es quien se beneficia con las funcionalidades del sistema y quien interactúa de forma directa con él.

2.4.2. Listado de casos de uso

Tabla 2.8 CU-1 Configurar dispositivo

CU-1	Configurar dispositivo
Actor	Especialista
Descripción	Permite al especialista establecer los datos de la configuración del dispositivo para la lectura de datos.

Tabla 2.9 CU-2 Adquirir señal

CU-2	Adquirir señal
Actor	Especialista
Descripción	Adquiere señales emitidas de los transmisores.

Tabla 2.10 CU-3 Procesar datos

CU-3	Procesar datos
Actor	Especialista
Descripción	Procesa los valores de las muestras adquiridas.

Tabla 2.11 CU-4 Representar señal

CU-4	Representar señal
Actor	Especialista
Descripción	Representa las señales en la pantalla del osciloscopio.

Tabla 2.12 CU-5 Amplificar señal de entrada

CU-5	Amplificar señal de entrada
Actor	Especialista
Descripción	Amplifica la señal de entrada.

Tabla 2.13 CU-6 Medir tiempo

CU-6	Medir tiempo
Actor	Especialista
Descripción	Mide retardo de tiempo entre señales.

Tabla 2.14 CU-7 Establecer marcas

CU-7	Establecer marcas
Actor	Especialista
Descripción	Establece unas marcas en la pantalla.

Tabla 2.15 CU-8 Seleccionar nivel medio.

CU-8	Seleccionar nivel medio.
Actor	Especialista
Descripción	Desplaza una señal por la vertical quedando marcada por el nivel medio de su amplitud.

Tabla 2.16 CU-9 Imprimir datos

CU-9	Imprimir datos
Actor	Especialista
Descripción	Imprime datos seleccionados.

Tabla 2.17 CU-10Guardar datos

CU-10	Guardar datos
Actor	Especialista
Descripción	Guarda los datos seleccionados.

Tabla 2.18 CU-11 Tabular datos

CU-11	Tabular datos
Actor	Especialista
Descripción	Crea una tabla con los datos seleccionados.

2.4.3. Diagrama de casos de uso del sistema

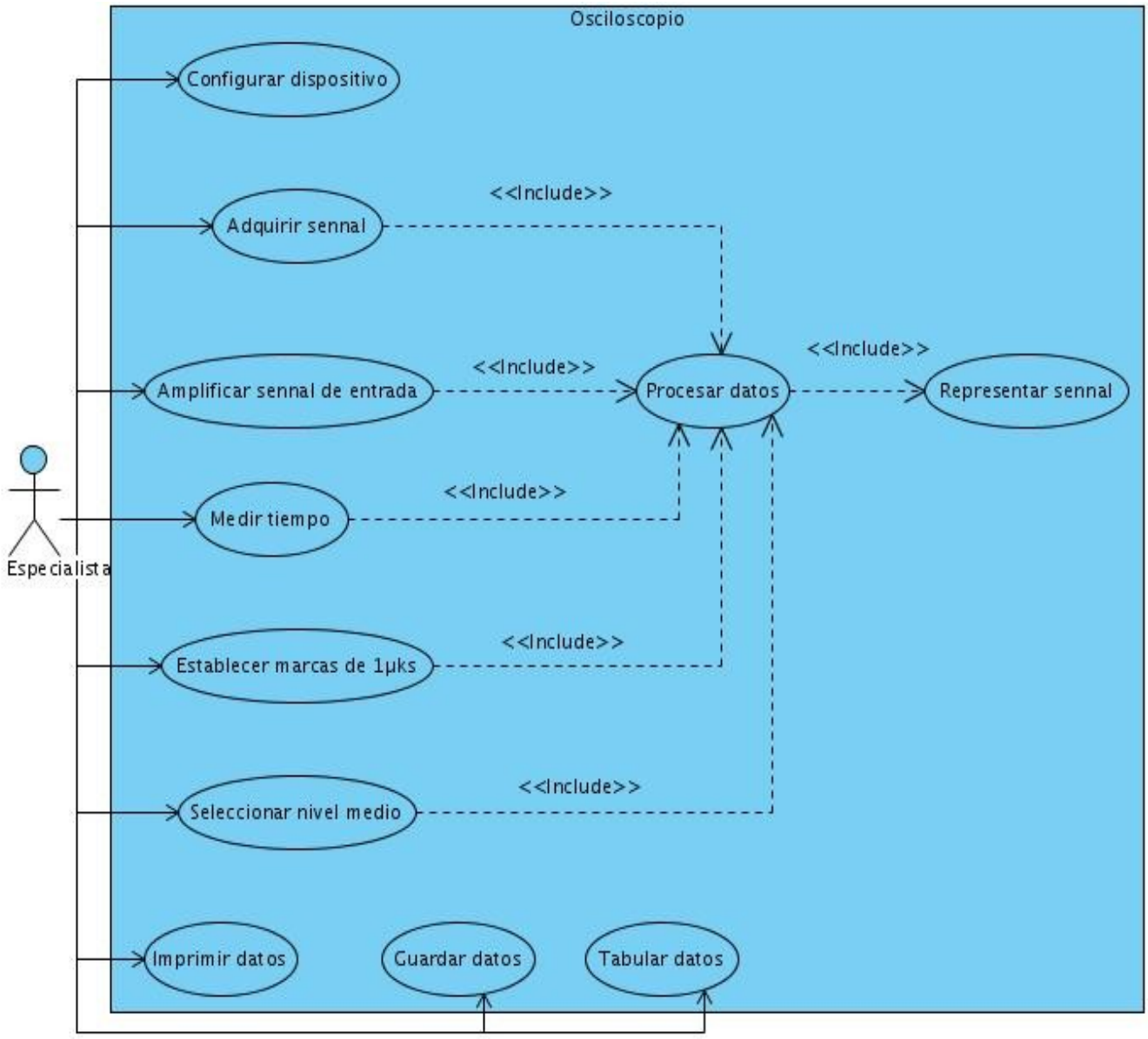

Figura 2.8 Diagrama de casos de uso del sistema

2.4.4. Casos de uso por módulos.

Tabla 2.19 Casos de uso por módulos

Cód.	Nombre del caso de uso	Módulo	Justificación
CU1	Configurar dispositivo	Módulo de visualización	
CU2	Adquirir señal	Módulo de adquisición de datos	
CU3	Procesar datos	Módulo de procesamiento	
CU4	Representar señal	Módulo de procesamiento	
CU5	Amplificar señal	Módulo de procesamiento	
CU6	Medir retardo entre señales	Módulo de procesamiento	
CU7	Establecer marcas	Módulo de procesamiento	
CU8	Seleccionar nivel medio	Módulo de procesamiento	
CU9	Imprimir	Módulo de visualización	
CU10	Tabular	Módulo de visualización	
CU11	Guardar	Módulo de visualización	

2.4.5. Casos de uso expandidos

Tabla 2.20 Expansión CU-1

Caso de Uso	Configurar dispositivo
Actores	Especialista(inicia)
Propósito	El caso de uso permite configurar un dispositivo
Resumen	El caso de uso inicia cuando el especialista selecciona la opción Configurar Dispositivo donde selecciona el puerto por el que se va a realizar la lectura de datos.
Precondiciones	Debe haber un dispositivo
Poscondiciones	Queda configurado el dispositivo
Prioridad	Crítico.
Interfaz	

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1- El caso de uso inicia cuando el Especialista selecciona la opción “Configurar dispositivo”	**2-** Muestra la interfaz de usuario para la configuración del dispositivo.
	3- Muestra los puertos disponibles para la adquisición de datos.
4- Selecciona el puerto de lectura y acepta.	**5-** Crea un archivo con la configuración establecida y concluye el caso de uso.

Tabla 2.21 Expansión CU-2

Caso de Uso	Adquirir señal	
Actores	Especialista(inicia)	
Propósito	El CU permite adquirir señales emitidas por determinados equipos.	
Resumen	El CU comienza cuando el Especialista oprime el botón Iniciar, dando comienzo a la adquisición de un bloque de datos.	
Precondiciones	Deben estar conectados los dispositivos para la adquisición de datos y el equipo a evaluar.	
Poscondiciones	Señal adquirida	
Prioridad	Crítico.	
Responsabilidades	1.1	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1- El caso de uso inicia cuando el Especialista oprime el botón “Iniciar“, en el osciloscopio.		**2-** Obtiene los datos guardados en el archivo de configuración.
		3- Estos datos son enviados al subsistema de adquisición para iniciar la captura.
		4- Inicializa el dispositivo con los datos de la configuración.
		5- Comienza el proceso de adquisición de un bloque de datos.
		6- Una vez adquirido un bloque de datos, detiene la captura.
		7- Cierra el puerto de lectura del dispositivo y concluye el caso de uso, dando inicio al caso de uso Procesar datos.(Ver caso de uso Procesar datos)

Tabla 2.22 Expansión CU-3

Caso de Uso	Procesar datos
Actores	Especialista(inicia)
Propósito	El CU permite procesar señales emitidas por determinados equipos.
Resumen	El CU inicia cuando se han adquirido los datos y están listos para procesarlos
Precondiciones	Deben haberse adquirido un bloque de muestras.
Poscondiciones	Señal procesada
Prioridad	Crítico.
Responsabilidades	1.2
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1- El caso de uso inicia cuando el subsistema de adquisición de datos le envía los valores muestreados para iniciar el procesamiento.
	2- Se realizan los cálculos necesarios para obtener la secuencia de puntos a representar.
	3- Esta secuencia es enviada al subsistema de interfaz de usuario para la representación de la señal y concluye el caso de uso, dando inicio al caso de uso Representar señal(ver caso de uso Representar señal)

Tabla 2.23 Expansión CU-4

Caso de Uso	Representar señal
Actores	Especialista(inicia)
Propósito	El CU permite representar señales emitidas por determinados equipos.
Resumen	El CU inicia cuando el Especialista oprime el botón “Iniciar”, el sistema toma las muestras, las procesa y comienza a representar la señal.
Precondiciones	Debe haberse adquirido y procesado los valores muestreados.
Poscondiciones	Queda representada la señal adquirida.
Prioridad	Crítico.
Responsabilidades	R1, 1.1 1.2
Interfaz	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	1- El caso de uso inicia cuando el subsistema de procesamiento le envía la secuencia de puntos a representar.
	2- Representa en una grafica la señal muestreada.
Cursos Alternos	
Línea **2**: Si no recibe la secuencia de puntos a representar, se trazará una línea horizontal en la pantalla del osciloscopio.	

Tabla 2.24 Expansión CU-5

Caso de Uso	Amplificar señal de entrada	
Actores	Especialista(inicia)	
Propósito	El CU permite amplificar y representar una señal emitida por un equipo determinado	
Resumen	El CU inicia cuando el Especialista escoge la opción “Amplificar”, para que se vea la señal más grande.	
Precondiciones	Debe estar adquiriéndose una señal.	
Poscondiciones	Queda la señal amplificada.	
Prioridad	Critico	
Responsabilidades	R2,2.1,2.2,2.3	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1- El caso de uso inicia cuando el Especialista selecciona la cantidad de amplificación.		**2-** Redimensiona los valores de los ejes de representación para obtener una vista mas ampliada de la señal.
		3- Representa la nueva señal con el valor de amplificación.

Tabla 2.25 Expansión CU-6

Caso de Uso	Medir tiempo	
Actores	Especialista(inicia)	
Propósito	El CU permite variar el tiempo de medición entre las señales que se están mostrando en la gráfica.	
Resumen	El caso de uso inicia cuando el Especialista acciona el botón de “Medición de tiempo”, con el cual es posible medir retardos de tiempo entre las señales que se están observando.	
Precondiciones	Debe estar adquiriéndose una señal.	
Poscondiciones	Se muestran en una gráfica las señales adquiridas	
Prioridad	Critico.	
Responsabilidades	R3,3.1,3.2,3.3,3.4,	
Interfaz		
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1- El caso de uso inicia cuando el Especialista selecciona la cantidad de tiempo entre una señal y otra, por medio del control de sensibilidad del osciloscopio.		**2-** Establece el tiempo de retardo.
		3- Representa la señal con el valor de retardo establecido y concluye el caso de uso.

Tabla 2.26 Expansión CU-7

Caso de Uso	Establecer marcas	
Actores	Especialista(inicia)	
Propósito	El CU permite el establecimiento de marcas en la pantalla de osciloscopio para realizar mediciones a la señal.	
Resumen	El CU inicia cuando el Especialista establece marcas en la pantalla del osciloscopio y permitiendo hacer cálculos de los parámetros de la señal.	
Precondiciones	Debe estar una señal en la pantalla del osciloscopio.	
Poscondiciones	Se muestran los parámetros calculados y el nivel medio de la señal.	
Prioridad	Critico.	
Responsabilidades	R4,4.1,4.2,4.3	
Interfaz		
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1- El caso de uso inicia cuando el Especialista acciona sobre la pantalla del osciloscopio para indicar la posición donde desea las marcas.		**2-** Establece las marcas en las posiciones que el Especialista indicó.
		3- Calcula los valores de los parámetros de la señal en cuestión.
		4- Muestra los valores de los parámetros y concluye el caso de uso.

Tabla 2.27 Expansión CU-8

Caso de Uso	Seleccionar nivel medio	
Actores	Especialista(inicia)	
Propósito	El CU permite calcular y representar el nivel medio de la señal en cuestión	
Resumen	El CU inicia cuando el Especialista establece las marcas en la pantalla del osciloscopio, permitiendo calcular el nivel medio entre dichas marcas.	
Precondiciones	Debe haber una señal en la pantalla del osciloscopio.	
Poscondiciones	Se representa gráficamente el nivel medio.	
Prioridad	Critico.	
Responsabilidades	R5,5.1,5.2	
Interfaz		
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
		1- El caso de uso inicia cuando el Especialista ha posicionado las marcas en la pantalla del osciloscopio.
		2- Calcula la amplitud de la señal.
		3- Establece el nivel medio con la amplitud de la señal y concluye el caso de uso.

Tabla 2.28 Expansión CU-9

Caso de Uso	Imprimir datos	
Actores	Especialista(inicia)	
Propósito	El CU permite imprimir los datos relacionados con una señal.	
Resumen	El CU inicia cuando el Especialista selecciona la opción “Imprimir”.	
Precondiciones	Debe estar adquiriéndose una señal.	
Poscondiciones	Quedan impresas las gráficas.	
Prioridad	Opcional.	
Responsabilidades	R6	
Interfaz		
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1- El caso de uso inicia cuando el Especialista decide imprimir la gráfica de la señal en pantalla y selecciona la opción de Imprimir.		**2-** Muestra la interfaz de usuario para configurar las opciones de impresión.
3- Configura las propiedades de impresión y acepta.		**4-** Imprime la gráfica y concluye el caso de uso.

Tabla 2.29 Expansión CU-10

Caso de Uso	Guardar gráfica	
Actores	Especialista(inicia)	
Propósito	El CU permite guardar la gráfica como una imagen	
Resumen	El CU inicia cuando el Especialista selecciona la opción Guardar abriéndose un cuadro de diálogo para guardar la gráfica como una imagen.	
Precondiciones	Debe haber una imagen en la pantalla del osciloscopio.	
Poscondiciones	Quedan guardada la señal como imagen.	
Prioridad	Opcional.	
Responsabilidades	R7	
Interfaz		
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1- El caso de uso inicia cuando el Especialista decide guardar la gráfica como una imagen y selecciona la opción Guardar.		**2-** Muestra la ventana para guardar la gráfica como imagen.
3- Establece la dirección y nombre donde se guardará la imagen.		**4-** Guarda la gráfica como una imagen y concluye el caso de uso

Tabla 2.30 Expansión CU-11

Caso de Uso	Tabular datos.	
Actores	Especialista(inicia)	
Propósito	El CU permite realizar una tabla con los parámetros a medir de la señal.	
Resumen	El CU inicia cuando el Especialista selecciona la opción de “Tabular datos” y el sistema le brinda una tabla con los datos adquiridos.	
Precondiciones	Debe estar adquiriéndose una señal.	
Poscondiciones	Quedan en una tabla los datos de medición.	
Prioridad	Opcional.	
Responsabilidades	R8	
Interfaz		
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Sistema
1- El caso de uso inicia cuando el Especialista decide crear una tabla con los parámetros de la señal y selecciona la opción “Tabular datos”.		**2-** Obtiene valores de los parámetros.
		3- Crea una tabla con los datos obtenidos.

Capítulo 3 Análisis y diseño del sistema.

En el presente capítulo se presenta ya el diseño del sistema propuesto, los patrones que se utilizaron en el diseño de la aplicación, una propuesta de arquitectura, un diagrama de clases, donde se definen las responsabilidades de estas y sus relaciones. Además se presentan otros artefactos involucrados en el diseño como los diagramas de secuencia por casos de uso, facilitando con esto el entendimiento de cada subsistema, y los patrones que se utilizaron en el diseño de la aplicación.

3.1. Patrones de diseño.

Constantemente los diseñadores se enfrentan a difíciles situaciones a la hora de diseñar las clases que conformarán la solución de su proyecto, en las que las siguientes preguntas son comunes: ¿quién es el responsable de...?; ¿quién debería ser responsable a la hora de crear una nueva instancia de alguna clase?; ¿cómo dar soporte a una dependencia escasa y a un aumento de la reutilización?; ¿cómo mantener la complejidad dentro de límites manejables?; ¿quién debería encargarse de atender un nuevo evento del sistema?. Para esas interrogantes hay una solución: patrones GRASP. Los patrones de diseño GRASP son patrones de asignación de responsabilidades a objetos que responden a una pareja de problema/solución y que son aplicables a otros contextos.

En la realización del diseño se utilizaron los patrones GRASP:

Experto: que propone como solución asignar la responsabilidad a la clase que cuenta con la información necesaria para cumplir la responsabilidad. Permitiendo que se conserve el encapsulamiento, soportando un bajo acoplamiento y una alta cohesión.

Creador: con la solución de asignarle a una clase la responsabilidad de crear los objetos de la otra en los casos de: contener, agregar, registrar o utilizar. Brindando soporte de bajo acoplamiento, lo cual supone menos dependencias entre clases y posibilidades.

Bajo Acoplamiento que brinda como solución asignar responsabilidades de manera que las clases no dependan fuertemente de otras. Ofreciendo como beneficio que son fáciles de entender por separadas, fáciles de reutilizar y no se afectan por cambios de otros componentes.

Alta Cohesión: Este patrón propone asignar la responsabilidad de manera que la complejidad se mantenga dentro de límites manejables asumiendo solamente las responsabilidades que deben manejar, evadiendo un trabajo excesivo. Su utilización: mejora la claridad y facilidad con que se entiende el diseño, simplifica el mantenimiento y las mejoras de funcionalidad, generan un bajo acoplamiento, soporta mayor capacidad de reutilización.

3.2. Propuesta de arquitectura: Arquitecturas en Capas

Esta premisa supone la base de la separación en capas del sistema. Distintas responsabilidades no deben ser delegadas en la misma clase, y llevado esto algo más allá, en el mismo conjunto de clases. En la actualidad, la tendencia más aceptada es la aplicación de patrones de diseño de arquitectura que dividen la responsabilidad en distintas capas que interaccionan unas con otras a través de sus interfaces. Se trata de los sistemas denominados multicapa o n-capas. Aplicados al software a desarrollar, se propone la utilización del modelo de aplicaciones mencionado. Entre los motivos por los cuales se recurre a la arquitectura multicapas se encuentran los siguientes:

- Aislamiento de la lógica de la aplicación en componentes independientes susceptibles de reutilizarse después en otros sistemas.
- Asignación de los diseñadores para que construyan determinadas capas; por ejemplo un equipo que trabaje exclusivamente en la capa de presentación (HMI). Así se brinda soporte a los conocimientos especializados en las habilidades de desarrollo y también a la capacidad de realizar actividades simultaneas en equipo.

Atendiendo a los criterios para aplicar este tipo de arquitectura y teniendo en cuenta que el sistema esta dividido en tres subsistemas, se puede afirmar que este estilo arquitectónico es aplicable al diseño en cuestión, lo que da lugar a que las capas queden definidas de la siguiente manera:

- Capa de representación de la señal (Subsistema HMI)
- Capa para el procesamiento de los datos adquiridos (Subsistema de procesamiento)
- Capa para la adquisición de datos (Subsistema de adquisición)

3.3. Diagrama de clases del diseño.

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación.

La figura 3.1 muestra las clases agrupadas en tres subsistemas. Estos subsistemas se eligieron de forma que todas las clases relacionadas con la interfaz-hombre-máquina (HMI) se ubican en un subsistema, todas las que tienen que ver con el procesamiento de datos en otro y las clases involucradas en la adquisición de datos en un tercero. La ventaja de colocar todas las clases separadas por subsistemas es que se puede reemplazar un subsistema por otro que ofrezca la misma funcionalidad.

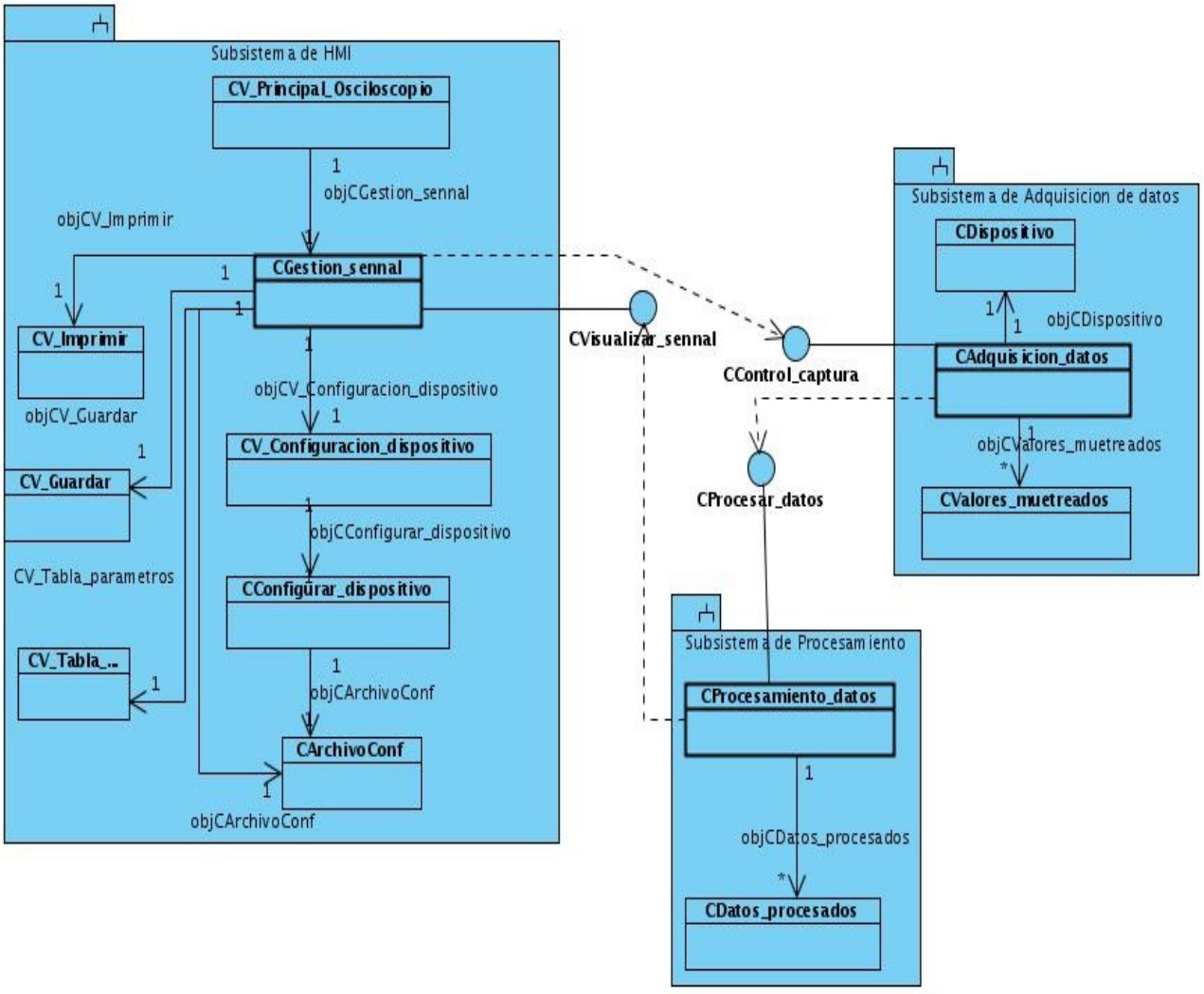

Figura 3.1 Diagrama de clases del diseño

En la figura anterior también se muestra las interfaces entre los subsistemas. Un círculo representa una interfaz. La línea continua de una clase a una interfaz significa que la clase proporciona la interfaz, este tipo de clase se denomina activa, que no es más que una colección de objetos activos que poseen un proceso o hilo y puede iniciar actividad de control; y por tanto se dibuja con un borde grueso. Una línea de trazo discontinuo de una clase a una interfaz significa que la clase usa la interfaz.

Las interfaces entre subsistemas se consideran relevantes para la arquitectura debido a las interacciones que definen operaciones que son accesibles desde fuera del subsistema.

3.3.1. Subsistema de adquisición de datos

En la figura 3.2 se puede observar las relaciones de asociación entre las clases dentro de los subsistemas, las cuales se representan con una flecha unidireccional desde la clase fuente hasta la clase destino indicando la visibilidad de los atributos. Esto se traduce como si la clase fuente tuviera un atributo que se refiere a una instancia de la clase destino.

En este subsistema es donde se adquieren las muestras de señales. Un mensaje es enviado por un objeto de la clase CGestion desde el subsistema HMI para iniciar la captura. Este llega a la interfaz CControl_captura, que es la utilizada en la comunicación entre estos dos subsistemas, quien está relacionada con la clase CAdqusicion_datos, esta última se encarga de realizar un conjunto de operaciones para comenzar la lectura de datos, como por ejemplo, seleccionar el puerto de lectura, abrir el dispositivo, establecer el canal por el cual se va a leer y otras más, necesarias para llevar a cabo este proceso. Una vez establecidos estos parámetros, se adquieren las muestras almacenadas en el buffer del dispositivo y se almacenan en un la entidad, CValores_muestreados, es decir se pasan los datos del dispositivo a la PC.

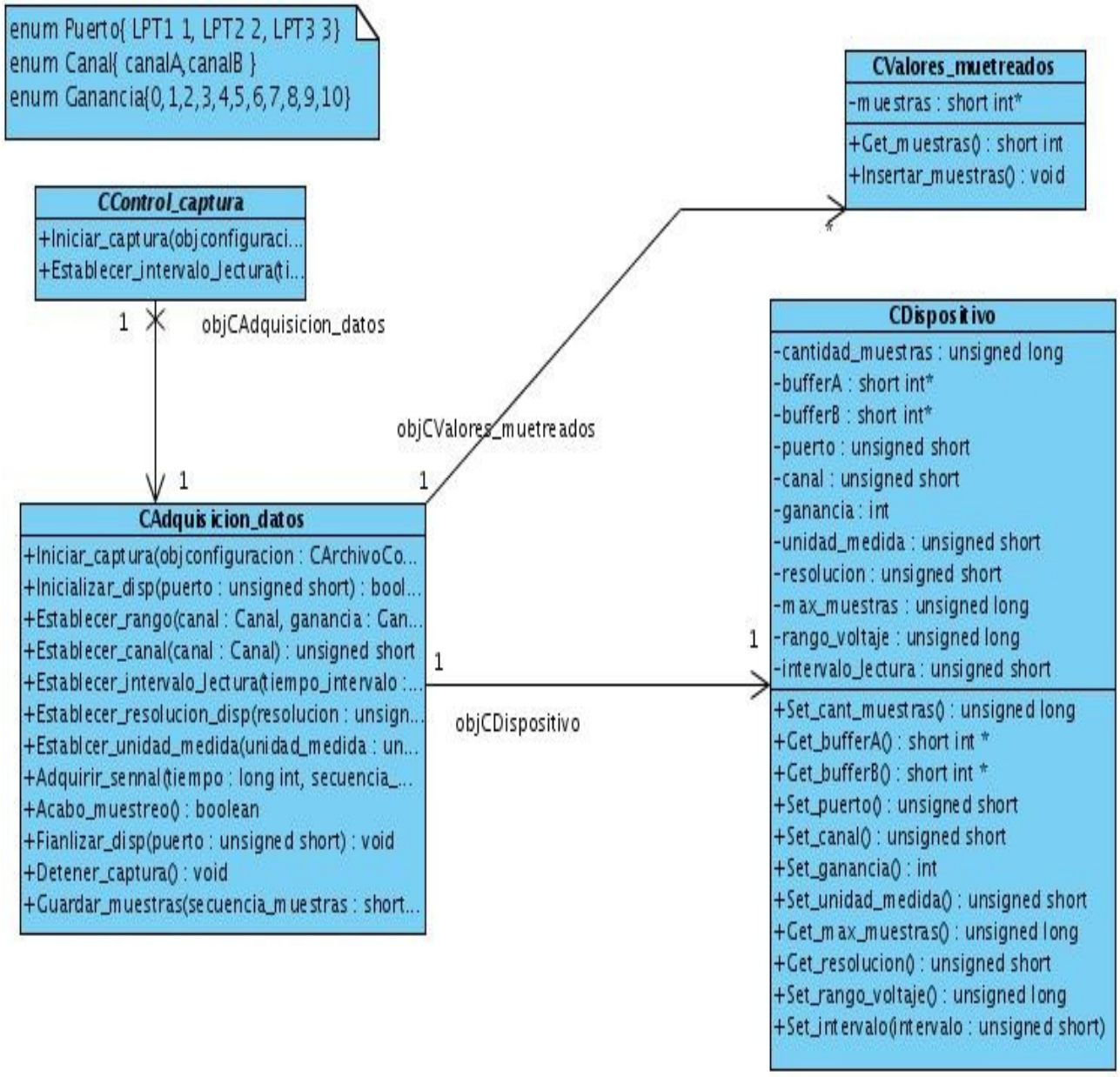

Figura 3.2 Diagrama de clases del diseño (Módulo de adquisición de datos).

3.3.2. Subsistema de procesamiento de datos

En este subsistema, figura 3.3, es donde se procesan las muestras adquiridas para realizar la posterior representación de la señal, es donde se llevan a cabo los procesos del dominio. Aquí llegan las muestras a la interfaz CProcesar_datos, que es la utilizada para la comunicación entre estos dos subsistemas pues contiene las operaciones que se pueden invocar desde afuera. Se inicia el procesamiento de los valores muestreados en la clase CProcesamiento_datos donde se generan los valores sinc, que son los utilizados para generar la secuencia de puntos nuevos en la interpolación senoidal, dando como resultado la secuencia de puntos a representar los cuales son enviados al subsistema HMI.

La entidad CDatos_procesados es la responsable de almacenar los valores de los parámetros que se le miden a la señal.

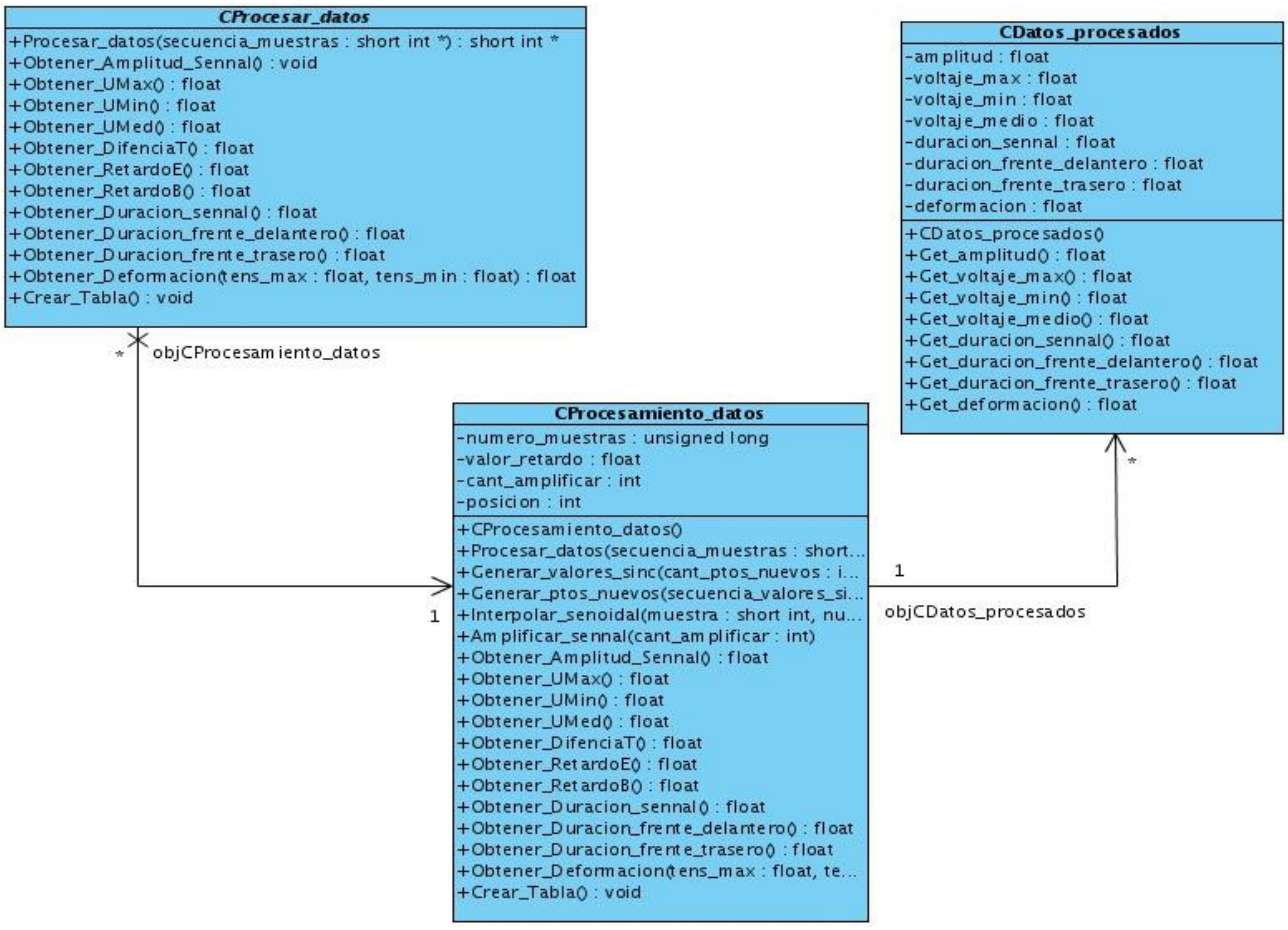

Figura 3.3 Diagrama de clases del diseño (Módulo de procesamiento).

3.3.3. Subsistema de HMI

En la figura 3.4 se muestra el subsistema HMI, contenido por las clases relacionadas con la configuración del dispositivo y la representación de la señal. Este muestra la interfaz de usuario (CV_Principal_Osciloscopio) con todas las funcionalidades del sistema, quien se relaciona con la clase controladora CGestion_sennal para llevar a cabo parte de las operaciones pues la otra parte esta relacionada con la configuración del dispositivo y para eso existe una interfaz (CV_Configuracion_dispositivo) que se relaciona con la controladora CConfigurar_dispositivo encargada de crear el archivo de configuración.

Por otra parte, la clase interfaz CVisualizar_sennal es la que define la comunicación entre los subsistemas, establece las operaciones que se pueden acceder desde afuera.

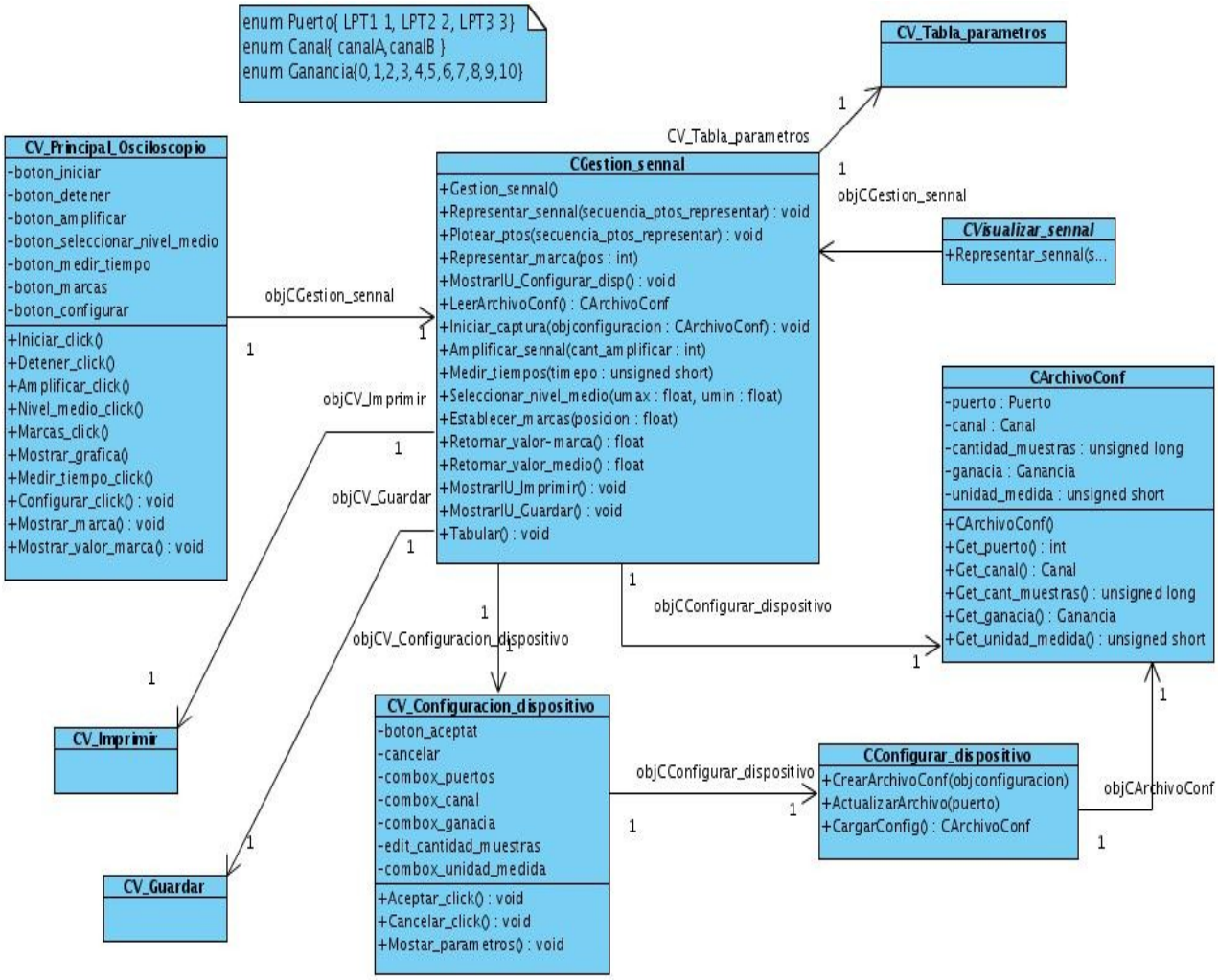

Figura 3.4 Diagrama de clases del diseño (Módulo de visualización).

3.4 Descripción de las clases

Tabla 3.1 Descripción de la clase CV_Principal_Osciloscopio

Nombre: CV_Principal_Osciloscopio	
Interfaz gráfica de usuario	
Atributo	**Tipo**
isp._iniciar	Tbutton
isp._configurar	Tbutton
isp._medir_tiempo	Tbutton
isp._amplificar	Tbutton
isp._salvar_imagen	Tbutton
isp._imprimir	Tbutton
performanceGraph	TperformanceGraph
Para cada responsabilidad:	
Nombre:	CV_Principal_Osciloscopio()
Descripción:	Constructor de la clase.
Nombre:	Iniciar():void
Descripción:	Envía el mensaje LeerArchivoConf() a la clase Cgestion_sennal
Nombre:	Configurar():void
Descripción:	Envía el mensaje MostarIU_Configuracion_disp() a la clase Cgestion_sennal
Nombre:	Medir_tiempo():void
Descripción:	Envía el mensaje Medir_tiempo(tiempo: foat) a la clase Cgestion_sennal
Nombre:	Amplificar():void
Descripción:	Envía el mensaje Amplificar(cant_amplificar: int) a la clase Cgestion_sennal
Nombre:	Salvar_como_imagen():void
Descripción:	Envía el mensaje Salvar_como_imagen()
Nombre:	Imprimir():void
Descripción:	Envía el mensaje Imprimir()
Nombre:	Establecer_marcas(pos :int):void
Descripción:	Establece marcas en la pantalla.

Tabla 3.2 Descripción de la clase Cgestion_sennal

Nombre: CGestion_sennal	
Controladora	
Para cada responsabilidad:	
Nombre:	Cgestion_sennal()
Descripción:	Constructor de la clase.
Nombre:	MostarIU_Configuracion_disp():void
Descripción:	Muestra la interfaz relacionada con la configuración del dispositivo.
Nombre:	Representar_sennal(secuencia_ptos_representar: short int *):void
Descripción:	Pinta la gráfica en la pantalla.
Nombre:	Plotear_ptos(secuencia_ptos: short int *):void
Descripción:	Representa los puntos en la pantalla del osciloscopio.
Nombre:	Representar_marcas(posición: int):void
Descripción:	Representa las marcas en la posición seleccionada.
Nombre:	LeerArchivoConf():CarchivoConf
Descripción:	Retorna un objeto de tipo CarchivoConf.
Nombre:	Amplificar_sennal(cant_amplificar:int)
Descripción:	Redimensionar los ejes de representación.
Nombre:	Medir_tiempo(tiempo: float)
Descripción:	Envia mensaje a la clase CcontrolCaptura.
Nombre:	Seleccionar_nivel_medio(umax:float, umin: float)
Descripción:	Establece el nivel medio entre las marcas seleccionadas
Nombre:	Establecer_marcas(posición:int)
Descripción:	Establece marcas en la pantalla.

Tabla 3.3 Descripción de la clase CV_Configuracion_dispositivo

Nombre: CV_Configuracion_dispositivo	
Interfaz gráfica de usuario	
Atributo	**Tipo**
isp._aceptar	Tbutton
isp._cancelar	Tbutton
comboBox_puerto	TcomboBox
comboBox_canal	TcomboBox
cant_muestras	Tedit
ganancia	Tedit
unidad_medida	Tedit
Para cada responsabilidad:	
Nombre:	CV_Configuracion_dispositivo()
Descripción:	Constructor de la clase.
Nombre:	Aceptar(): void
Descripción:	Manda el mensaje CrearArchivoConf(puerto, canal, cant_muestras, ganancia, unidad_medida) a la clase Cconfigurar_dispositivo.
Nombre:	Cancelar():void
Descripción:	Cierra la ventana

Tabla 3.4 Descripción de la clase Cconfigurar_dispositivo

Nombre: Cconfigurar_dispositivo	
Controladora	
Atributo	**Tipo**
path	**string**
Para cada responsabilidad:	
Nombre:	Cconfigurar_dispositivo()
Descripción:	Constructor de la clase.
Nombre:	Actualizar_Dispositivo(puerto, canal, cant_muestras, ganancia, unidad_medida):void
Descripción:	Actualiza los datos de configuración.
Nombre:	CrearArchivoConf(puerto, canal, cant_muestras, ganancia, unidad_medida):void
Descripción:	Crea un archivo con los datos de configuración.

Tabla 3.5 Descripción de la clase CarchivoConf

Nombre: CarchivoConf	
Entidad	
Atributo	**Tipo**
puerto	Puerto
canal	Canal
ganancia	Ganancia
cant_muestras	int
unidad_medida	int
Para cada responsabilidad:	
Nombre:	CarchivoConf (puerto, canal, cant_muestras, ganancia, unidad_medida)
Descripción:	Constructor de la clase
Nombre:	Set_ Puerto(isp. :Puerto)
Descripción:	Establecer el valor del puerto de lectura
Nombre:	Set_ canal(canal:Canal)
Descripción:	Establecer el valor del canal de lectura
Nombre:	Set_ ganancia(ganancia:Ganancia)
Descripción:	Establecer el valor del ganancia.
Nombre:	Set_ cantidad_muestras(cant_muestras:int)
Descripción:	Establecer el valor de la cantidad de muestras
Nombre:	Set_ unidad_medida(unidad_medida:int)
Descripción:	Establecer el valor de la unidad de medida
Nombre:	Get_ Puerto()
Descripción:	Retorna el puerto de lectura
Nombre:	Get_ canal()
Descripción:	Retorna el canal
Nombre:	Get_ ganancia()
Descripción:	Retorna la ganancia
Nombre:	Get_ cantidad_muestras()
Descripción:	Retorna la cantidad de muestras a adquirir(en este caso serán 50)
Nombre:	Get_ unidad_medida()
Descripción:	Retorna la unidad de medida.

Tabla 3.6 Descripción de la clase Cvisulazar_sennal

Nombre: Cvisulazar_sennal	
Interfaz	
Para cada responsabilidad:	
Nombre:	Cvisulazar_sennal()
Descripción:	Constructor de la clase.
Nombre:	Representar_sennal(secuencia_ptos_representar: short int *)
Descripción:	Envía el mensaje a la clase CgestionSennal.

Tabla 3.7 Descripción de la clase Cprocesamiento_datos

Nombre: Cprocesamiento_datos	
Controladora	
Atributo	**Tipo**
numero_muestras	unsigned long
valor_retardo	float
cant_amplificar	int
posición	int
Para cada responsabilidad:	
Nombre:	Cprocesamiento_datos()
Descripción:	Constructor de la clase
Nombre:	Procesar_sennal(valores_muestras:short int)
Descripción:	Devuelve los valores a representar.
Nombre:	Generar_valores_sinc(cant_ptos_nuevos: int, secuencia_valores_sinc:short int*): short int*
Descripción:	Retorna la secuencia de valores sinc
Nombre:	Generar_ptos_nuevos(secuencia_valores_sinc:short int*, valores_muestras:short int*):short int*
Descripción:	Retorna secuencia de puntos nuevos
Nombre:	Interpolar_senoidal(muestra:short int, num_muestra:int, num_ptos_nuevos:int, secuancia_ptos_nuevos:short int,secuencia_ptos_representar:short int):short int*
Descripción:	Retorna la secuencia de puntos a representar
Nombre:	Obtener_amplitud_señal():float
Descripción:	Retorna la amplitud
Nombre:	Obtener_Umax():float
Descripción:	Retorna la tensión máxima

Nombre:	Obtener_Umin():float
Descripción:	Retorna la tensión mínima
Nombre:	Obtener_Umed():float
Descripción:	Retorna la tensión media
Nombre:	Obtener_DiferenciaT():float
Descripción:	Retorna la diferencia de periodos
Nombre:	Obtener_RetardoE():float
Descripción:	Retorna el retardo E
Nombre:	Obtener_RetardoB():float
Descripción:	Retorna el retardo B
Nombre:	Obtener_Duración_señal():float
Descripción:	Retorna la duración de la señal
Nombre:	Obtener_Duración_frente_trasero():float
Descripción:	Retorna la duración de la parte trasera de la señal
Nombre:	Obtener_Duración_frente_delantero():float
Descripción:	Retorna la duración de la parte delantera de la señal
Nombre:	Obtener_Deformación(ten_max: float, ten_min: float):float
Descripción:	Retorna la deformación de la señal

Tabla 3.8 Descripción de la clase Cdatos_procesados

Nombre: Cdatos_procesados	
Entidad	
Atributo	**Tipo**
amplitud	float
voltaje_max	float
voltaje_min	float
voltaje_medio	float
duración _señal	float
duración_frente-delantero	float
duración_frente-trasero	float
deformación	float
Para cada responsabilidad:	
Nombre:	Get_amplitud():float
Descripción:	Retorna la amplitud
Nombre:	Get_voltaje_max():float
Descripción:	Retorna el voltaje máximo
Nombre:	Get_voltaje_min():float
Descripción:	Retorna el voltaje mínimo
Nombre:	Get_voltaje_med():float
Descripción:	Retorna el voltaje medio
Nombre:	Get_duración _señal():float
Descripción:	Retorna la duración de la señal
Nombre:	Get_duración _frente_delantero():float
Descripción:	Retorna la duración del frente delantero de la señal
Nombre:	Get_duración _frente_trasero():float
Descripción:	Retorna la duración del frente trasero de la señal
Nombre:	Get_deformación():float
Descripción:	Retorna la deformación de la señal

Tabla 3.9 Descripción de la clase CControl_captura

Nombre: CControl_captura	
Interfaz	
Para cada responsabilidad:	
Nombre:	Adquirir_señal():void
Descripción:	Envía mensaje a la clase Cadquisicion_datos

Tabla 3.10 Descripción de la clase Cdispositivo

Nombre: Cdispositivo	
Entidad	
Atributo	**Tipo**
cant_muestras	unsigned long
bufferA	short int
bufferB	short int
puerto	unsigned short
canal	unsigned short
ganancia	int
unidad_medida	unsigned short
max_muestras	unsigned long
rango_voltage	unsigned long
Para cada responsabilidad:	
Nombre:	Get_cant_muestras():unsigned long
Descripción:	Retorna la cantidad de muestras
Nombre:	Get_bufferA():short int
Descripción:	Retorna el buffer si están almacenadas las muestras en este buffer
Nombre:	Get_bufferB():short int
Descripción:	Retorna el buffer si están almacenadas las muestras en este buffer
Nombre:	Set_puerto():unsigned short
Descripción:	Establece el puerto correspondiente
Nombre:	Set_canal():unsigned short
Descripción:	Establece el canal correspondiente
Nombre:	Set_ganancia():int
Descripción:	Establece la ganancia correspondiente
Nombre:	Set_unidad _medida():unsigned short

Descripción:	Establece la unidad de medida correspondiente
Nombre:	Get_max_muestras():unsigned long
Descripción:	Retorna el máximo de muestras obtenidas
Nombre:	Get_resolución():unsigned short
Descripción:	Retorna la resolución obtenida
Nombre:	Get_rango_voltaje():unsigned long
Descripción:	Retorna el rango de voltaje que se va a utilizar

Tabla 3.11 Descripción de la clase Cadquisicion_datos

Nombre: Cadquisicion_datos	
Controladora	
Para cada responsabilidad:	
Nombre:	Iniciar_captura(obj_CarchivoConf : CarchivoConf*) : void
Descripción:	Inicia la captura de los datos del dispositivo.
Nombre:	Seleccionar_puerto(puerto unsigned short) : boolean
Descripción:	Se selecciona el puerto correspondiente
Nombre:	Inicializar_dispositivo(puerto : int) : boolean
Descripción:	Establecer la configuración del dispositivo para la lectura de los datos.
Nombre:	Establecer_rango(canal:Canal, ganancia:Ganancia):unsigned short
Descripción:	Se establece el rango de lectura de datos
Nombre:	Establecer_canal(canal:Canal):unsigned short
Descripción:	Se establece el canal por el cual se deben adquirir los datos
Nombre:	Establecer_intervalo_lectura(tiempo_intervalo:unsigned long , es_lento:unsigned char*,tiempo_lectura:Tiempo):unsigned short
Descripción:	Se establece el intervalo de lectura de los datos
Nombre:	Establecer_ resolución _disp(resolución:unsigned short):void
Descripción:	Se establece la resolución para realizar la adquisición de datos
Nombre:	Comenzar_captura(muestras:unsigned long):unsigned short
Descripción:	Comienza la captura de los datos emitidos por el l dispositivo
Nombre:	Adquirir _señal(tiempo:long int,valores_muestrasA:short int*, valores_muestrasB:short int*,cant_muetras:unsigned long):void
Descripción:	Se adquieren los datos de la señal del dispositivo
Nombre:	Finalizar_disp(puertos:unsigned short):void
Descripción:	Se cierra el puerto por el que se adquirieron los datos
Nombre:	Detener_captura():void
Descripción:	Se detiene el proceso de captura

Tabla 3.12 Descripción de la clase Cvalores_muetreados

Nombre: Cvalores_muetreados	
Entidad	
Atributo	**Tipo**
muestras	short int*
Para cada responsabilidad:	
Nombre:	Get_muetras():short int
Descripción:	Retorna la cantidad de muestras
Nombre:	Guardar_muestras():void
Descripción:	Guarda las muestras adquiridas

3.5. Diagramas de secuencia

El diagrama de la figura 3.5 muestra la interacción entre los objetos para llevar a cabo la operación de configurar el dispositivo.

Se puede apreciar en la imagen que el Especialista es el actor que inicia la secuencia al interactuar con la interfaz CV_Principal_Osciloscopio, especificando que desea configurar el dispositivo. Luego se trasmite el mensaje hasta llegar a la clase responsable de esta operación, CConfigurar_dispositivo, quien se encargara de generar un archivo con la configuración establecida.

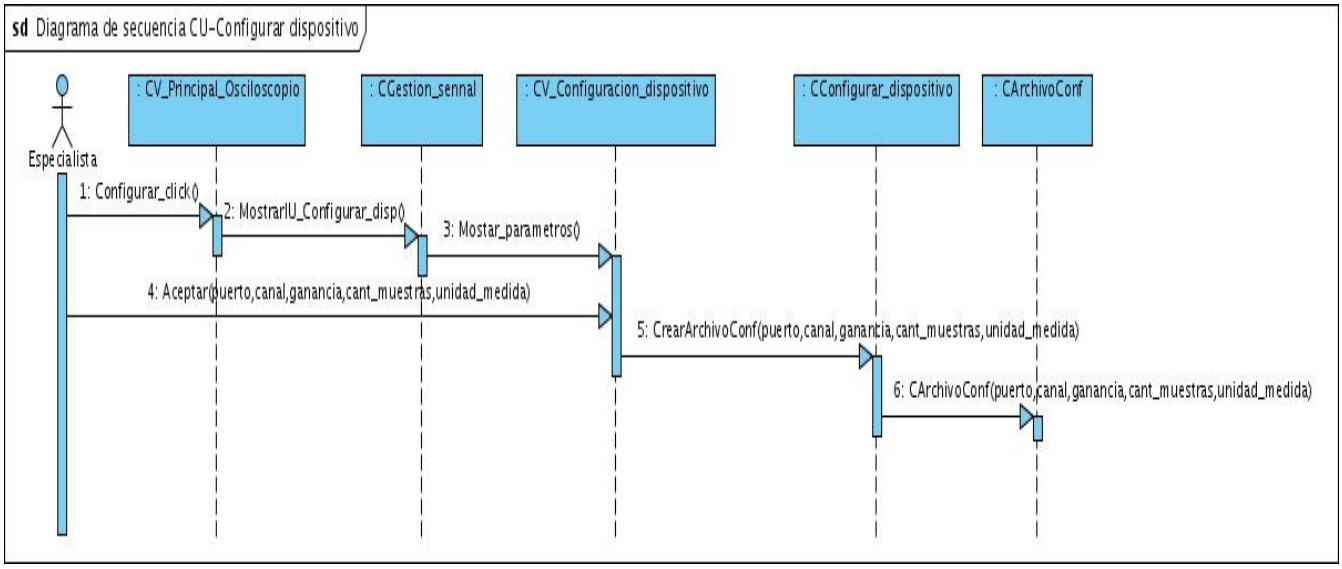

Figura 3.5 Diagrama de secuencia CU-Configurar dispositivo

El diagrama de la figura 3.6 muestra la interacción entre los objetos para llevar a cabo la operación de adquirir las muestras.

Es apreciable en la imagen que el Especialista es el actor que inicia la secuencia al interactuar con la interfaz CV_Principal_Osciloscopio, especificando que desea iniciar la captura de las muestras. Luego se trasmite el mensaje hasta llegar a la clase responsable de esta operación, CAdquisicion_datos,

quien se encargara de tomar las muestras almacenadas en el buffer del dispositivo y guardarlas en la memoria de la PC.

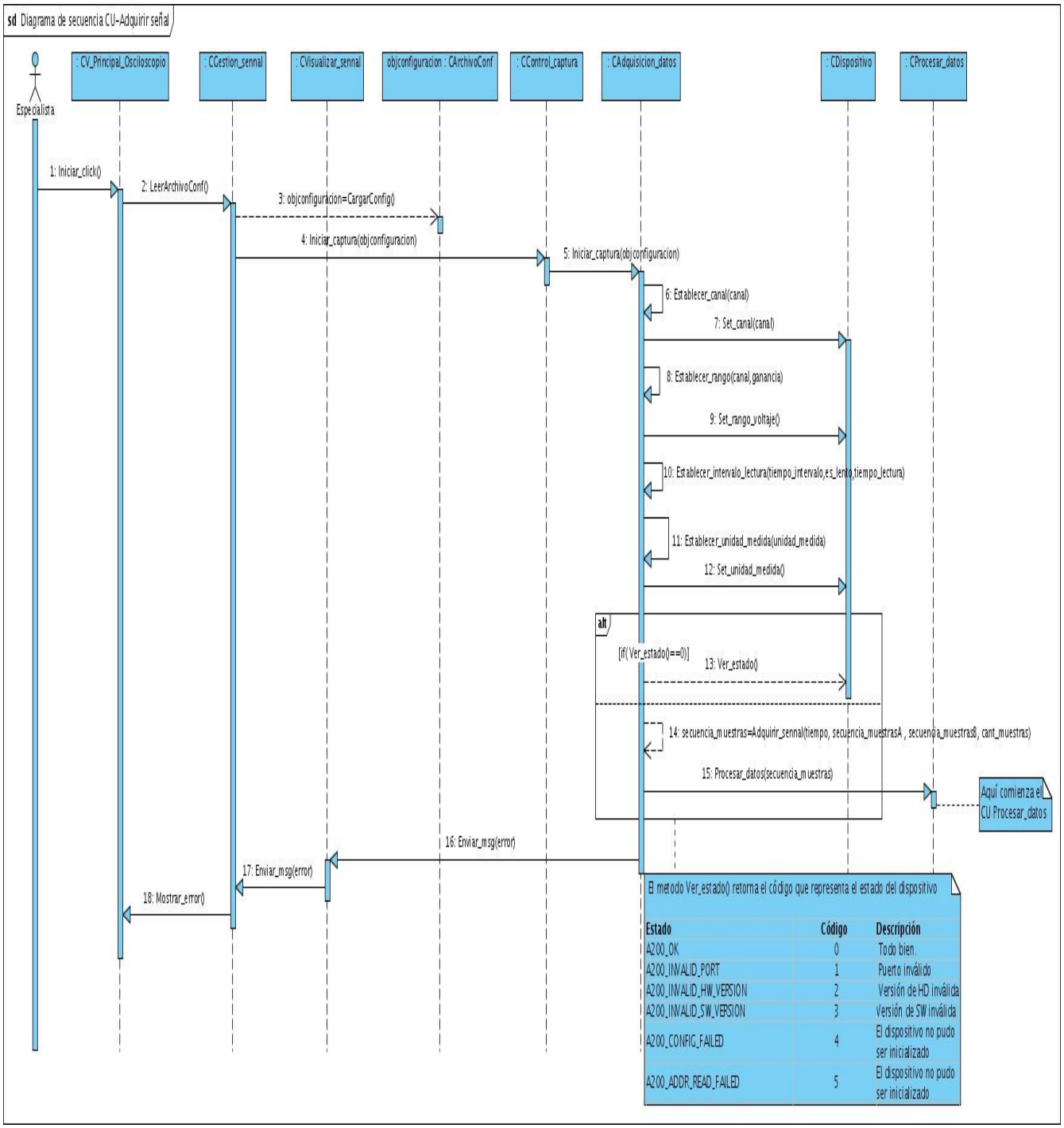

Figura 3.6 Diagrama de secuencia CU-Adquirir señal

El diagrama de la figura 3.7 muestra la interacción entre los objetos para llevar a cabo la operación de procesar las muestras adquiridas.

Se puede apreciar en la imagen que la secuencia inicia cuando se reciben las muestras del subsistema de adquisición de datos en la interfaz del subsistema de procesamiento invocando al método Procesar_datos(). Luego se trasmite el mensaje hasta llegar a la clase responsable de esta operación, CProcesamiento_datos, quien se encargara de realizar los métodos necesarios para obtener los puntos que van a representar la señal adquirida en una gráfica.

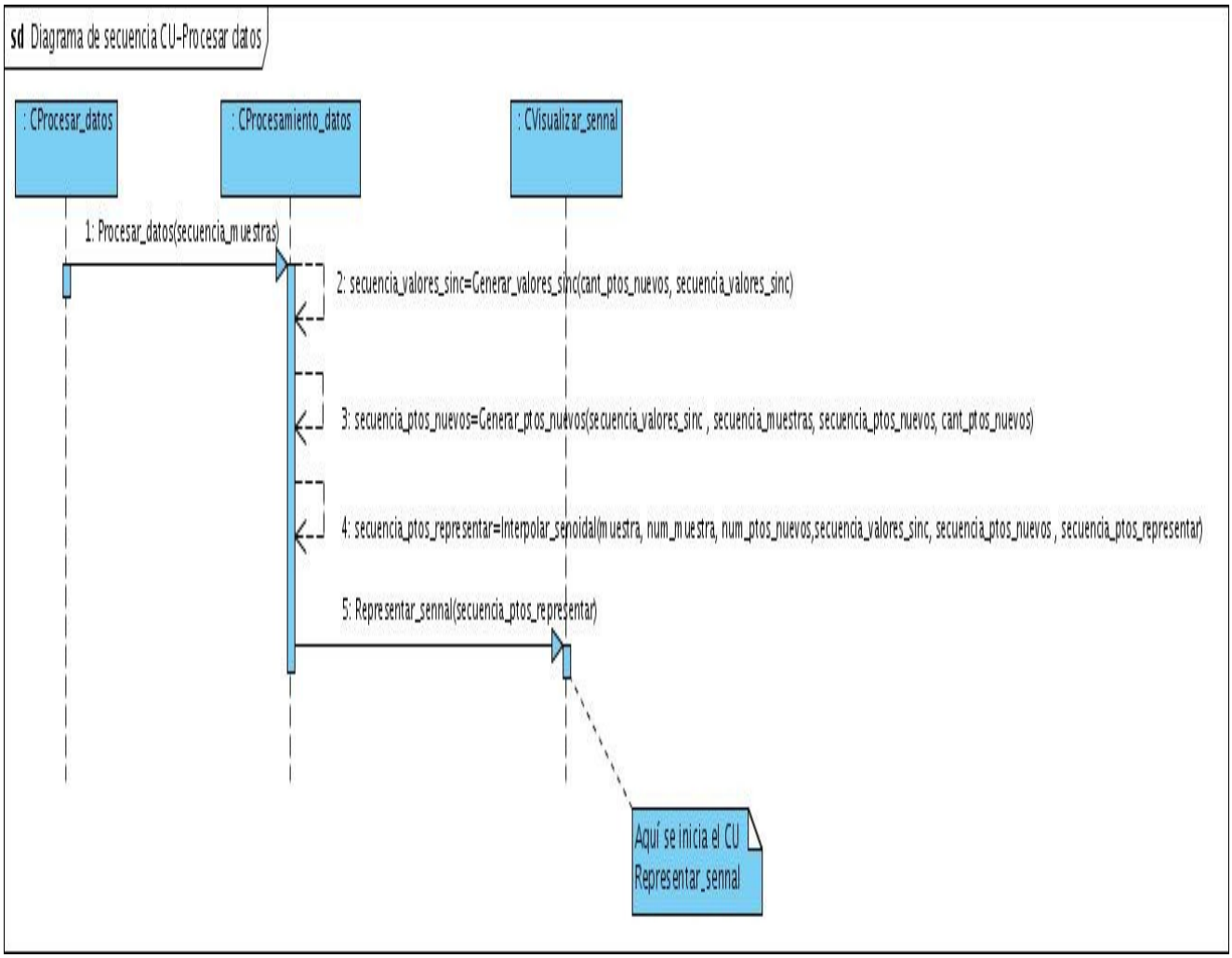

Figura 3.7 Diagrama de secuencia CU-Procesar datos

El diagrama de la figura 3.8 muestra la interacción entre los objetos para llevar a cabo la operación de reconstrucción de la señal adquirida.

Como se observa en la imagen, la secuencia inicia cuando se recibe la secuencia de puntos que conformaran la grafica de la señal, desde el subsistema de procesamiento de datos en la interfaz del subsistema HMI invocando al método Representar_sennal)(). Luego se trasmite el mensaje hasta llegar a la clase responsable de esta operación, CGestion_sennal, quien se encargara de realizar los métodos necesarios para obtener la gráfica de la señal muestreada.

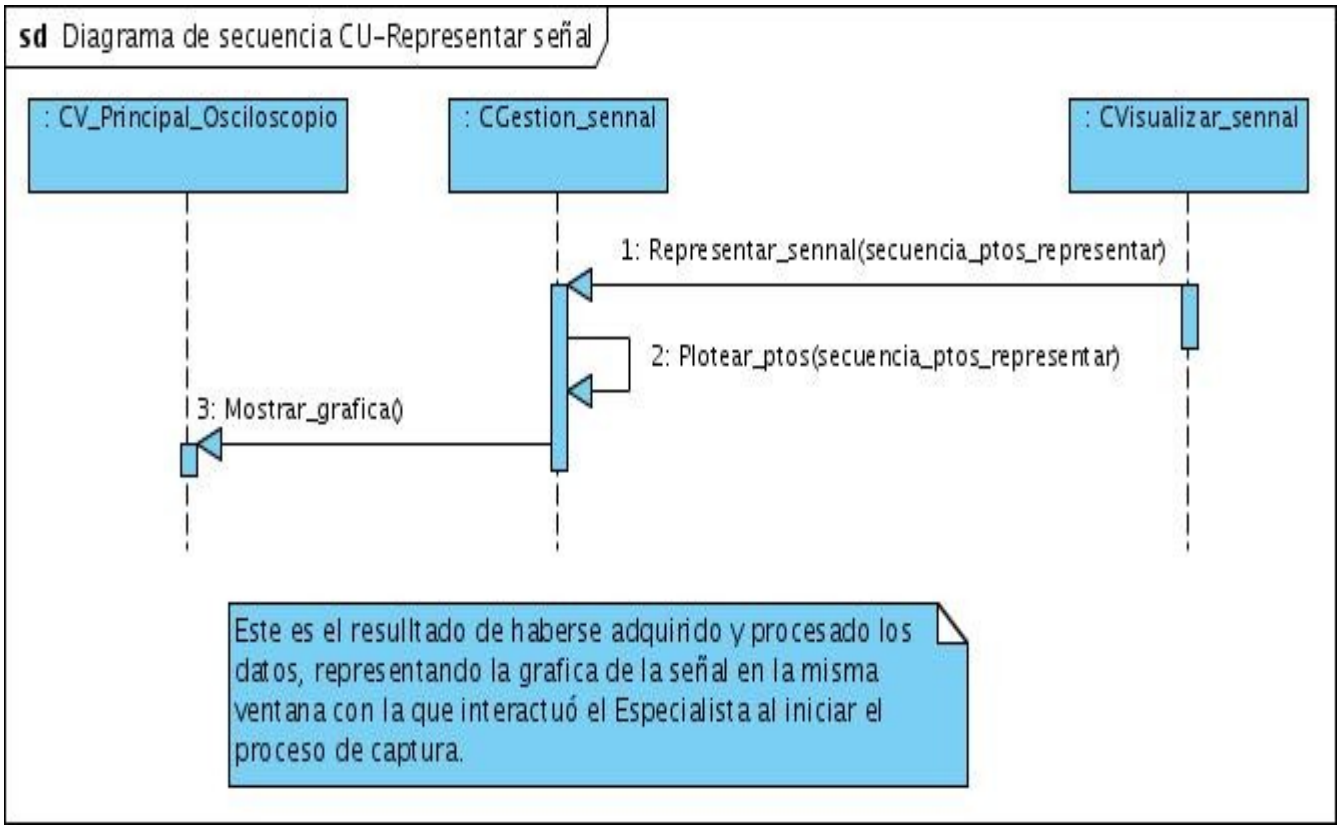

Figura 3.8 Diagrama de secuencia CU-Representar señal

El diagrama de la figura 3.9 muestra la interacción entre los objetos para llevar a cabo la operación de amplificar la señal.

Se puede apreciar en la imagen que el Especialista es el actor que inicia la secuencia al interactuar con la interfaz CV_Principal_Osciloscopio, especificando que desea amplificar la gráfica. Luego se trasmite el mensaje hasta llegar a la clase responsable de esta operación, CProcesamiento_datos, quien se encargara de procesar las nuevas muestras y amplificarlas. El diagrama concluye con la visualización de la señal amplificada.

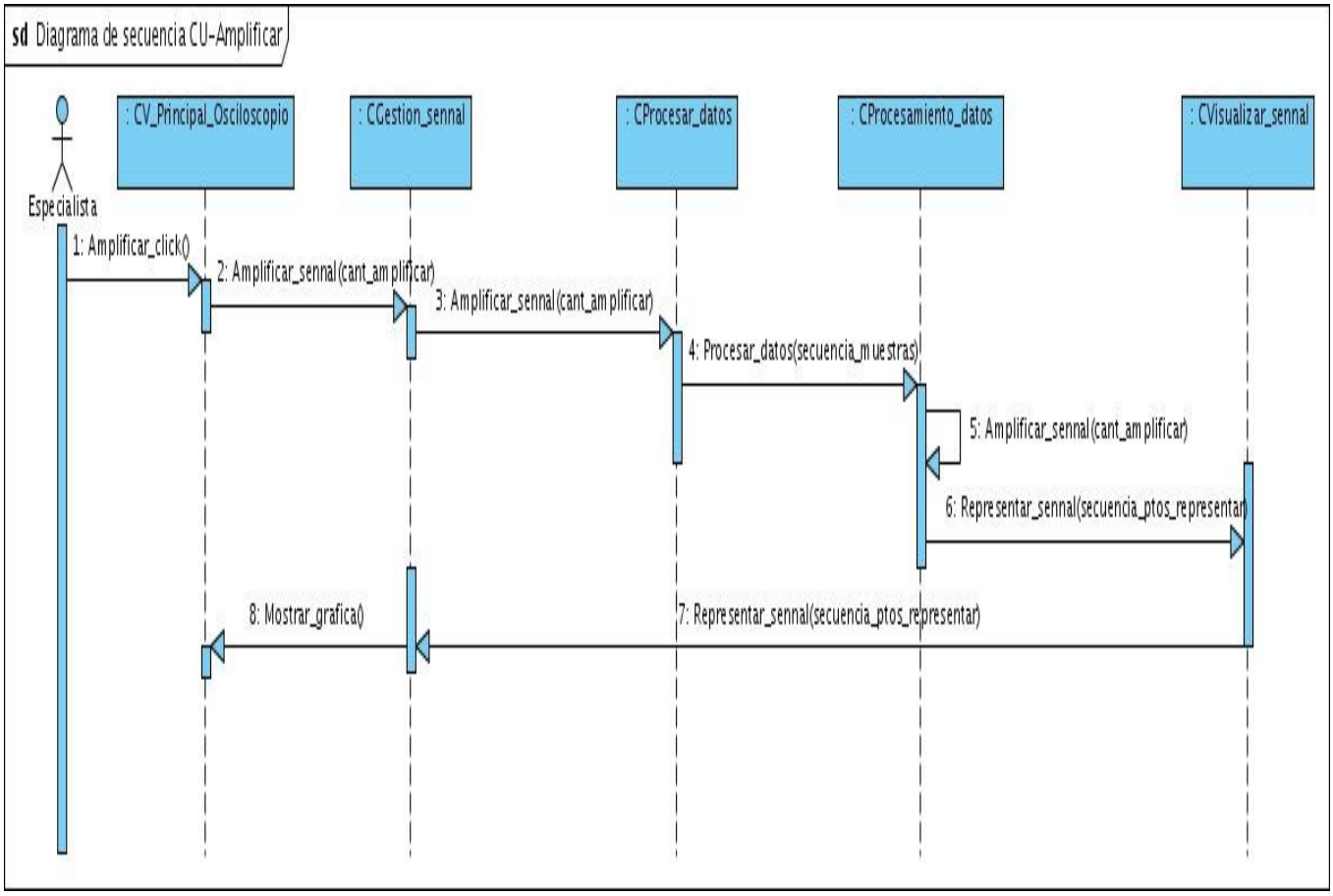

Figura 3.9 Diagrama de secuencia CU-Amplificar señal

En la figura 3.10 se puede apreciar cómo el actor inicia la secuencia al interactuar con la interfaz CV_Principal_Osciloscopio, especificando que desea cambiar el intervalo de lectura. Luego se trasmite el mensaje hasta llegar a la clase responsable de esta operación, CAdquisicion_datos, quien se encargara de establecer el intervalo de lectura.

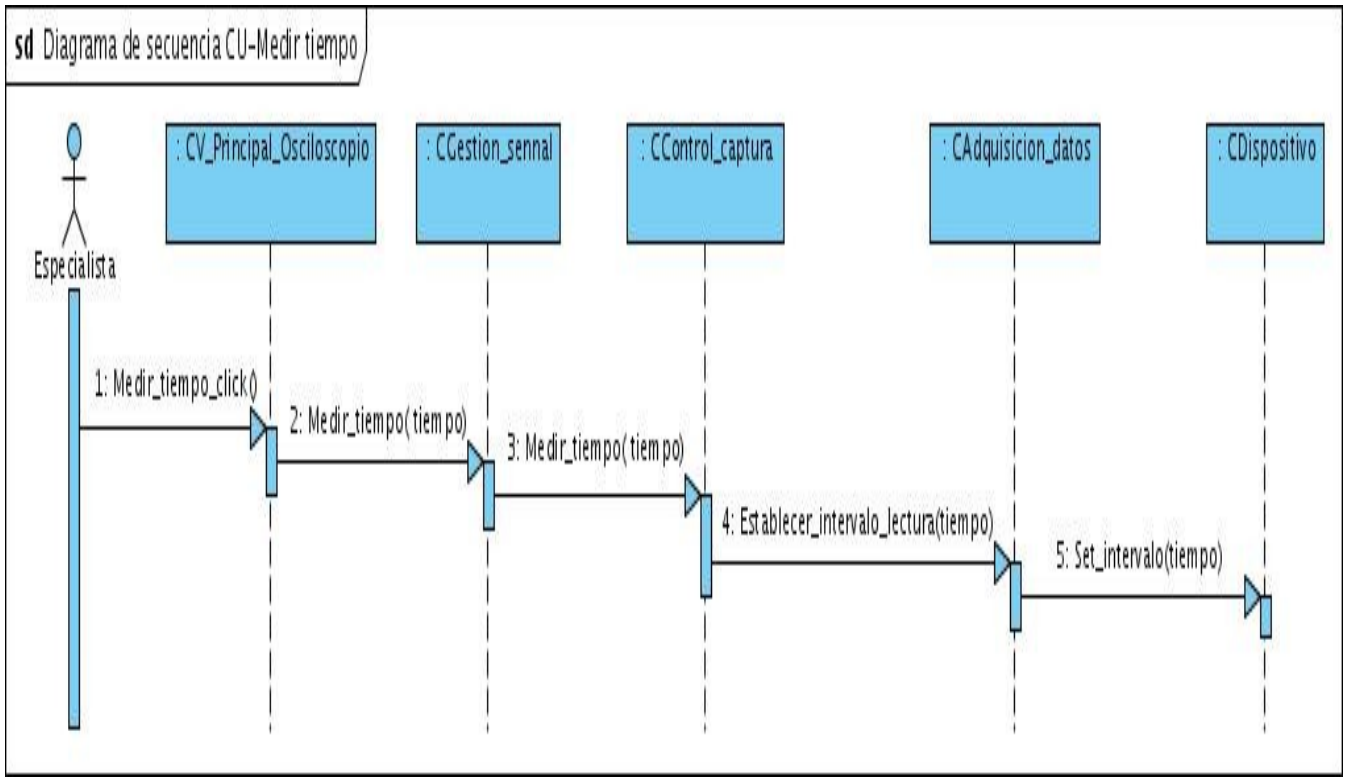

Figura 3.10 Diagrama de secuencia CU-Medir tiempo

En la figura 3.11 se puede apreciar cómo el actor inicia la secuencia al interactuar con la interfaz CV_Principal_Osciloscopio, especificando que desea establecer marcas en la pantalla. Luego se trasmite el mensaje hasta llegar a la clase responsable de esta operación, CGestion_sennal, quien se encargara de establecer las posiciones de las marcas y devolver el valor donde están situadas las mismas.

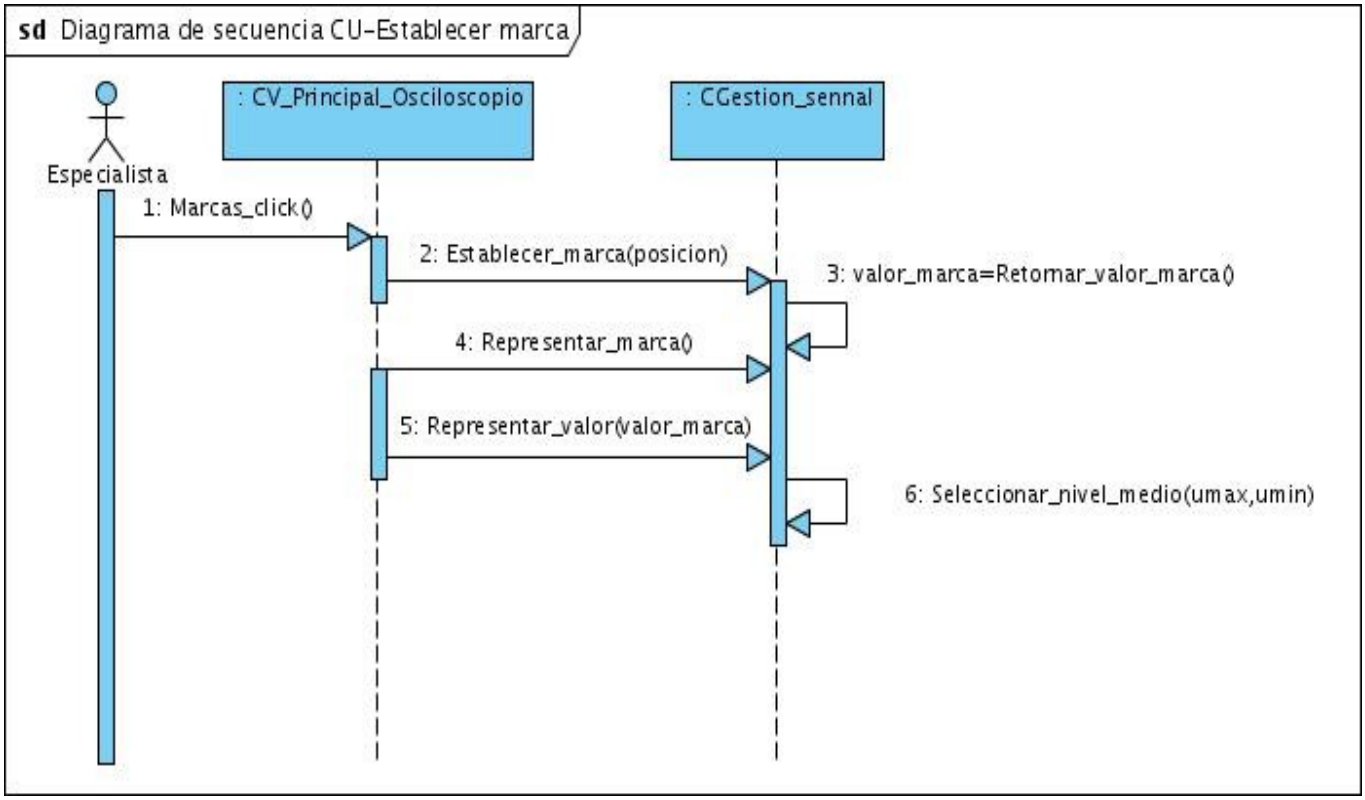

Figura 3.11 Diagrama de secuencia CU-Establecer marcas

En la figura 3.11 se puede apreciar cómo una vez establecidas las marcas se estable el nivel medio de las mismas.

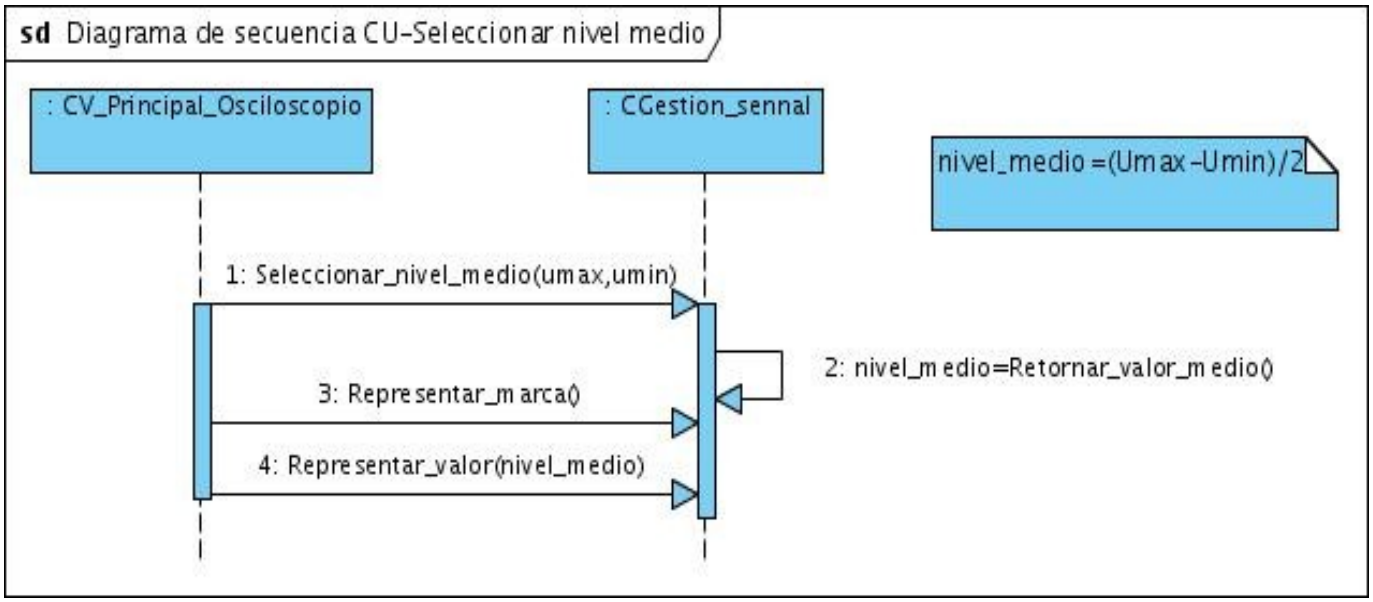

Figura 3.12 Diagrama de secuencia CU-Seleccionar nivel medio

El diagrama de la figura 3.13 muestra la interacción entre los objetos para llevar a cabo la operación de imprimir la gráfica. Aquí se muestran la secuencia de pasos lógicos para realizar dicha operación .

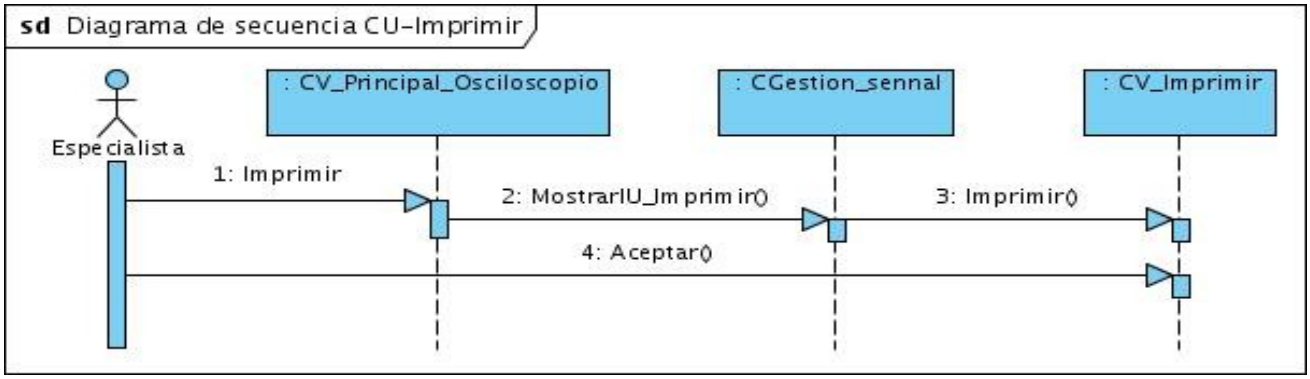

Figura 3.13 Diagrama de secuencia CU-Imprimir

Para guardar la grafica el actor indica en la interfaz principal esta operación, cuya secuencia se muestra a continuación en la figura 3.14Ñ

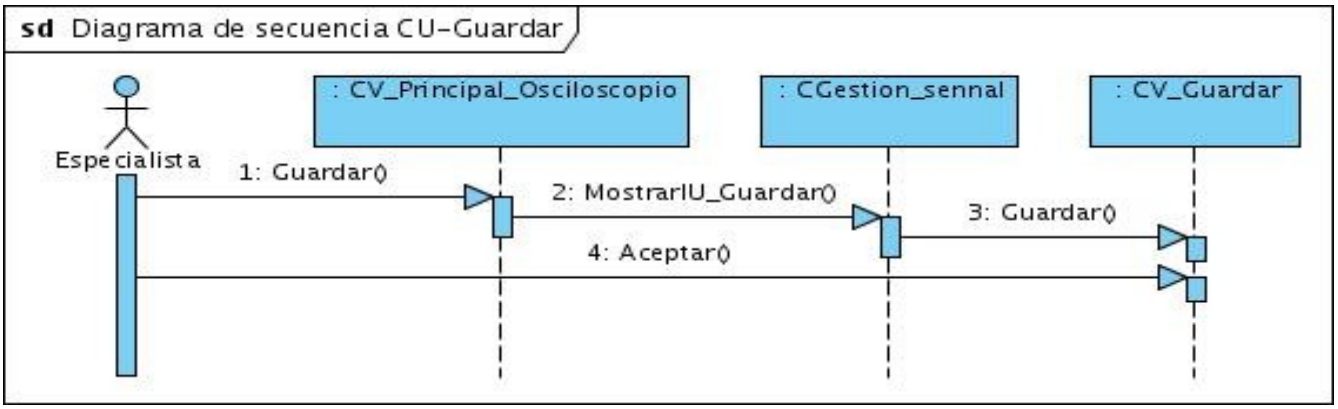

Figura 3.14 Diagrama de secuencia CU-Guardar

El diagrama de la figura 3.15 muestra la interacción entre los objetos para llevar a cabo la operación de crear una tabla con los parámetros que se le miden a la señal.

Se puede apreciar en la imagen que el Especialista es el actor que inicia la secuencia al interactuar con la interfaz CV_Principal_Osciloscopio, especificando que desea tabular los parámetros. Luego se trasmite el mensaje hasta llegar a la clase responsable de esta operación, CProcesamiento_datos, quien se encargara de generar la tabla requerida. El diagrama concluye con la visualización de la tabla.

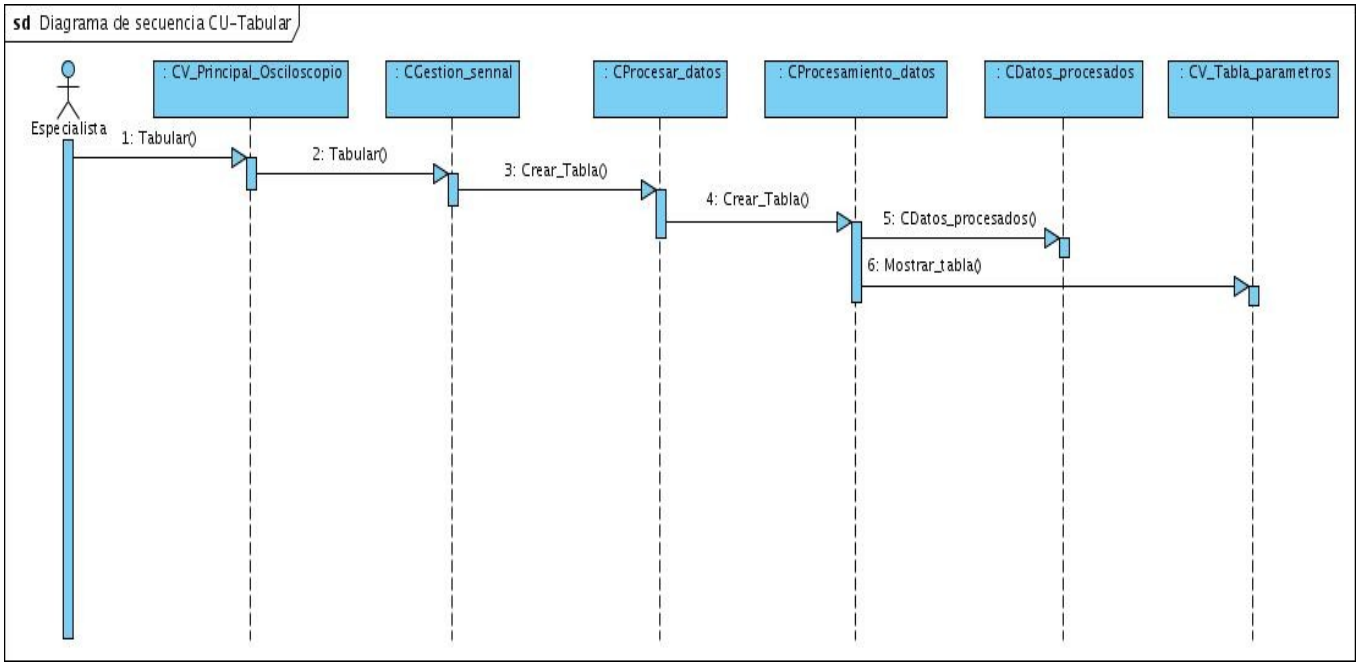

Figura 3.15 Diagrama de secuencia CU-Tabular

Conclusiones

- Concluida la investigación del proceso a automatizar, se logro elaborar un diseño de software para implementar el osciloscopio virtual donde se aplican los resultados de la investigación realizada dando cumplimiento al objetivo de este trabajo.

- El estudio de las herramientas y tecnologías actuales, sus tendencias en el campo de la instrumentación virtual, sobre todo las relacionadas con software libre; son precisos para el desarrollo de un diseño efectivo.

- Se citaron dos métodos de muestreo de señales, uno muestreo en tiempo real con interpolación y el otro muestreo en tiempo equivalente, de los cuales se eligió el método de muestreo en tiempo real con interpolación senoidal por sus características

- Para lograr capturar los tipos más importantes de objetos que existen o los eventos que suceden en el entorno del sistema, es necesario desarrollar un modelo de dominio.

- El agrupar todas las clases por subsistemas, facilita el desarrollo de cada modulo así como su posterior mantenimiento.

- En cada etapa del proceso de desarrollo del software se obtuvieron los artefactos necesarios para guiar la investigación hacia el diseño del sistema correcto.

Concluido este trabajo de diploma y una vez desarrollado los temas que en él se exponen, se dejó listo para comenzar la próxima etapa de desarrollo y lograr la implementación de la solución propuesta.

Recomendaciones

- Implementar completamente la solución propuesta.
- Realizar un estudio exhaustivo de Open Modélica para una posible utilización.
- Estudiar y analizar otros tipos de arquitectura utilizadas para el desarrollo de instrumentos virtuales.
- Continuar investigando las necesidades de los clientes, para incorporar nuevas funcionalidades a la aplicación.

Referencias bibliográficas

[1] Universidad, de Córdova. Grupo de Arquitecturas Avanzadas de Computadores. [En línea] 17 de Mayo de 2008. [Citado el: 25 de Mayo de 2008.] Dsiponible en: http://www.uco.es/grupos/gaac/?L%EDneas_de_Trabajo:Instrumentaci%F3n_Virtual.

[2] Wikipedia. Osciloscopio. [En línea] 10 de Junio de 2007. [Citado el: 5 de Enero de 2008.] Disponible en: http://es.wikipedia.org/wiki/Osciloscopio.

[3] National Instruments. ¿Qué es la instrumentación virtual? [En línea] 2008. [Citado el: 7 de Enero de 2008.] Disponible en: http://digital.ni.com/worldwide/latam.nsf/web/all/01E4BFF8EC93532086256B6000669953.

[4] Asociación Argentina de Control Automático. La Instrumentación Virtual. [En línea] 2008. [Citado el: 10 de Enero de 2008.] Disponible en: http://www.aadeca.org/socios/tracnova/La_Instrumentacion_Virtual.pdf.

[5] Universidad Nacional de Mar del Plata. Secretaria de Ciencia y Técnica. [En línea] Junio de 2001. [Citado el: 15 de Enero de 2008.] Disponible en: http://www.mdp.edu.ar/rectorado/secretarias/investigacion/nexos/13/osciloscopio.htm.

[6] F.J.M. Electrónica Fácil. Características del osciloscopio y del generador de funciones. [En línea] 2004. [Citado el: 20 de Enero de 2008.] Disponible en: http://www.electronicafacil.net/tutoriales/CARACTERISTICAS-OSCILOSCOPIO-GENERADOR-FUNCIONES.php.

[7] Ivar Jacobson, Grady Booch, James Rumbaugh. 2004. *El Proceso Unificado de Desarrollo de Software.* La Habana: Félix Varela, 2004. II.

[8] JACOBSON, I.; BOOCH, G.; Rumbaugh, J. El Lenguaje Unificado de Modelado. Manual de Referencia. Addison Wesley, 2000. 3 p.v

[9] Estable, Cristóbal Cuesta. Laboratorios Virtuales en Ciencia y Tecnología. [En línea] 2004 [Citado el: 20 de Enero de 2008.] Disponible en: http://rabfis15.uco.es/lvct/tutorial.

[10] Hameg Instruments. Oscilloscopes. [En línea] 2008. [Citado el: 5 de Febrero de 2008.] Disponible en: http://www.hameg.com/15.0.html.

[11] GIMP ToolKit, [En línea]. Fundación GTK, 2004. [2007]. Disponible en: http://www.gtk.org/

[12] Documentación Oficial de Gtkmm, [En línea]. Fundación GTK, 2005. [2007]. Disponible en: http://www.gtkmm.org/gtkmm2/docs/

[13] Elementos técnicos del producto Qt, [En línea]. Trolltech, 2006. [2007]. Disponible en: http://www.trolltech.com/products/qt/features/index

[14] Licencia QPL, [En línea]. Trolltech, 1999. [2007]. Disponible en: http://www.trolltech.com/products/qt/licenses/licensing/qpl

[15] . PC Oscilloscope and Data Acquisition Products. [En línea] [Citado: Febrero 2, 2008.] Disponible: http://www.picotech.com/.

[16] PC based Oscilloscope and other measuring instruments. [En línea] 2005[Citado el: 10 de Febrero de 2008.] Disponible en: http://www.tiepie.com/.

[17] PC measuring device. [En línea] 1997-2007. [Citado: Febrero 15, 2008.] Disponible en: http://www.etcsk.com/.

[18] PC OSCILLOSCOPES AND ANALYZERS. [En línea] [Citado Febrero 20, 2008.] Disponible en: http://www.bitscope.com/

Glosario de términos

Bus: canal por el que circula información electrónica en forma de bits. El ancho de bus es el número de bits transmitidos simultáneamente por el bus.

GPIB: es un estándar bus de datos digital de corto rango para conectar dispositivos de test y medida, por ejemplo multímetros, osciloscopios y otros, con dispositivos que los controlen como un ordenador.

Simulaciones: Es un intento de modelar situaciones de la vida real por medio de un programa de computadora, lo que requiere ser estudiado para ver cómo es que trabaja el sistema.

Instrumentación virtual: es un sistema de medición, análisis y control de señales físicas con una computadora por medio de instrumentos virtuales.

Instrumento programable: es un sistema electrónico de medida o generación de señales basado en un procesador digital, en cuya memoria se sitúa un programa que automatiza todas las medidas.

Osciloscopio: es un instrumento de medición electrónico para la representación gráfica de señales eléctricas que pueden variar en el tiempo

Tecnologías libres: La tecnología libre incluye todos aquellos conocimientos tecnológicos que respetan las libertades del conocimiento libre. Incluyen entre ellos el software libre, el open source o código abierto, el hardware libre o el hardware abierto, y los estándares abiertos. Son tecnologías que permiten su libre reutilización. Los productos/servicios generados con ellas no tienen necesariamente por que ser gratuitos.

Instrumento virtual: Un instrumento que no es real, se ejecuta en una computadora y tienes sus funciones definidas por software.

FPGA (del inglés Field Programmable Gate Array): es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad se puede programar. La lógica programable puede reproducir desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinacional hasta complejos sistemas en un chip.

CPLD (del acrónimo inglés Complex Programmable Logic Device): Extienden el concepto de un PLD (del acrónimo inglés Programmable Logic Device) a un mayor nivel de integración ya que permite

implementar sistemas más eficaces, ya que utilizan menor espacio, mejoran la fiabilidad del diseño, y reducen costos.

Interfaz hombre-máquina: Es el medio de comunicación entre el hombre y la computadora.

Índice de figuras y tablas