

# DIRECCIÓN DE SERVICIOS DE CERTIFICACIÓN LABORATORIO DE ENSAYOS DE COMPATIBILIDAD ELECTROMAGNÉTICA RADIADA

## ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE CENDITLAB

FORMATO:		N DOC:	
Originado por:	Elaborado por:	Revisado por:	Aprobado por:
Br. Arias B., Jose A.	Br. Arias B., Jose A.	-	-
Fecha: 16/06/2017	Fecha: 16/06/2017		

Versión 0.1		1 / 15
-------------	---	--------

# Índice

<b>1. Introducción</b>	<b>3</b>
1.1. Propósito . . . . .	3
1.2. Alcance . . . . .	3
1.3. Definiciones, acronimos y abreviaturas . . . . .	3
1.4. Referencias . . . . .	3
1.5. Visión general . . . . .	4
<b>2. Descripción general</b>	<b>4</b>
2.1. Perspectiva de producto . . . . .	4
2.2. Funciones de producto . . . . .	5
2.3. Características de los usuarios . . . . .	6
2.4. Restricciones . . . . .	6
2.5. Suposiciones y dependencias . . . . .	6
<b>3. Requerimientos específicos</b>	<b>6</b>
3.1. Requerimientos de interfaces externas . . . . .	6
3.1.1. Interfaces de usuario . . . . .	6
3.1.2. Interfaces de hardware . . . . .	8
3.1.3. Interfaces de software . . . . .	9
3.1.4. Interfaces de comunicaciones . . . . .	10
3.2. Requerimientos funcionales . . . . .	11
3.2.1. Requerimiento funcional 1 . . . . .	11
3.2.2. Requerimiento funcional 2 . . . . .	11
3.2.3. Requerimiento funcional 3 . . . . .	11
3.2.4. Requerimiento funcional 4 . . . . .	11
3.2.5. Requerimiento funcional 5 . . . . .	12
3.2.6. Requerimiento funcional 6 . . . . .	12
3.2.7. Requerimiento funcional 7 . . . . .	12
3.2.8. Requerimiento funcional 8 . . . . .	12
3.2.9. Requerimiento funcional 9 . . . . .	13
3.2.10. Requerimiento funcional 10 . . . . .	13
3.2.11. Requerimiento funcional 11 . . . . .	13
3.2.12. Requerimiento funcional 12 . . . . .	13
3.2.13. Requerimiento funcional 13 . . . . .	14
3.3. Requerimientos de desempeño . . . . .	14
3.3.1. Requerimiento de desempeño 1 . . . . .	14
3.3.2. Requerimiento de desempeño 2 . . . . .	14
3.4. Restricciones de diseño . . . . .	14
3.5. Atributos del sistema de software . . . . .	14
<b>4. Asignación de prioridades y plan de entregas</b>	<b>14</b>
4.1. Elección del método para asignar prioridades . . . . .	14
4.2. Plan de entregas . . . . .	15

## 1. Introducción

Esta sección brinda un panorama general del contenido del presente documento de especificaciones de requisitos de software. Se da al final una lista de abreviaturas.

### 1.1. Propósito

El propósito de este documento es dar una descripción detallada de los requerimientos para el software CENDITLAB. Ilustrará el propósito y sentará de manera formal y completa la base inicial para el proceso de desarrollo. Explicará la restricciones del sistema, interfaces e interacciones con aplicaciones externas.

Este documento se dirige principalmente a los futuros usuarios de la software CENDITLAB para fines de aprobación, y para desarrolladores, que se involucren en un futuro en el soporte y mejoras de la base de código.

### 1.2. Alcance

El software CENDITLAB es una aplicación de escritorio que brinda a los usuarios del *sistema de medición de ruido* un entorno de trabajo o laboratorio virtual

El software CENDITLAB es una aplicación para PC de escritorio que sirve como interfaz de software para el *sistema de medición de ruido* presente en el Cendit. Permite realizar muchas de las tareas propias de los equipos de un sistema de medición de ruido dentro de un entorno de trabajo en una aplicación centralizada, o laboratorio virtual.

CENDITLAB permite a los usuarios realizar mediciones de manera automatizada y remota, configuración los instrumentos del sistema de medición de ruido, capturar y visualizar de datos además de generar reportes en formatos digitales, como pdf o html.

CENDITLAB utiliza la capacidad de transferencia de datos a través de buses de comunicaciones que disponen los instrumentos del sistema de medición de ruido, como GPIB, USB o LAN. Para ello el PC donde se ejecute la aplicación debe disponer de los aditamentos de hardware apropiado que permitan acceder a estos buses, así como también el respectivo soporte de software en forma de librerías o controladores de dispositivo, que permitan a un PC el acceso a estos buses.

### 1.3. Definiciones, acronimos y abreviaturas

Término	Definición
Stakeholder	Participante
VM	Virtual Machine, maquina virtual que se encarga de ejecutar una aplicación
JVM	Java Virtual Machine
UML	Unified Modeling Language, lenguaje unificado de modelado, permite modelar gráficamente la estructura y comportamiento de unidades de software
SO	Sistema Operativo

### 1.4. Referencias

- [1] ISO/IEC/IEEE 29148. *Systems and Software Engineering - Lyfe Cycle Processes - Requirements Enginering*. ISO/IEC/IEEE, 2011.

- [2] ISO/IEC/IEEE 42010. *Systems and Software Engineering - Architecture Description*. ISO/IEC/IEEE, 2011.
- [3] Rajesh Narang. *Software Engineering Principles and Practices*. Mc Graw Hill, New Delhi, India, 2015.
- [4] Ian Sommerville. *Software Engineering*. Addison-Wesley, Boston, Massachusetts, 2011.

## 1.5. Visión general

El resto de este documento incluye tres capítulos y un apéndice. El segundo provee un panorama de la funcionalidad del sistema y su interacción con el sistema de medición de ruido. Este capítulo además introduce los diferentes tipos de *participantes* y su interacción con el sistema. Más adelante, el capítulo menciona las restricciones del sistema y las suposiciones de la aplicación.

El tercer capítulo describe la especificación de requerimientos al detalle y describe las diferentes interfaces del sistema. Se emplean diferentes técnicas de especificación con el objeto de mostrar los requerimientos de manera precisa para distintas audiencias.

El cuarto capítulo trata con el establecimiento de la prioridad de los requerimientos. Este incluye una motivación. Expone los motivos para los métodos de prioridad escogidos y el por qué de las alternativas rechazadas.

El apéndice al final de este documento incluye los resultados de la priorización de requerimientos y el plan de entrega basado en las prioridades.

## 2. Descripción general

Esta sección brindará una perspectiva general de todo el sistema de software, CENDITLAB. El sistema será explicado dentro de su contexto para mostrar cómo éste interactúa con el sistema de medición de ruido por medio de su funcionalidad básica. Describirá además qué tipo de participantes –*stakeholders*– usarán el sistema y qué funcionalidad estará disponible para cada tipo de ellos. Al final, se presentan las restricciones y suposiciones para el sistema.

### 2.1. Perspectiva de producto

CENDITLAB será una aplicación de escritorio que servirá como interfaz de usuario de software para el sistema de medición de ruido. El software del sistema consiste de tres partes: el soporte para interfaz gráfica, la capa de abstracción de comunicaciones y la automatización de mediciones.

CENDITLAB será una aplicación dirigida inicialmente a PC que ejecute un sistema operativo propietario (Windows) o alguna distribución en software libre (Linux). Esto garantiza que CENDITLAB será una aplicación portable, podrá ser ejecutada por cualquier equipo o dispositivo que posea una *máquina virtual* apropiada, como JVM en Java o el entorno .NET de Microsoft, tecnologías que se encuentran disponibles bajo licencias de software libre.

La aplicación resultará en una interfaz de software para los tres equipos principales que componen el sistema de medición de ruido. En sí resultará en un *instrumento virtual* que concentra toda la funcionalidad del SMR en un solo sitio: el dispositivo o PC que ejecute la aplicación.

Conceptualmente se ha dividido la aplicación en cuatro grandes paquetes, como ilustra el diagrama UML de la figura 1

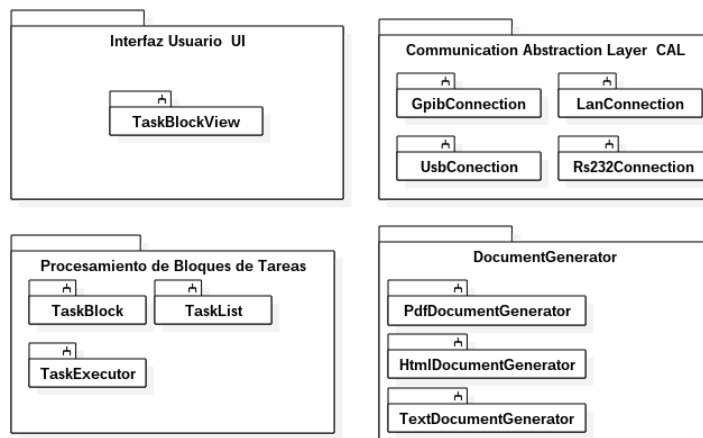


Figura 1: Diagrama de paquetes UML que describe la organización del software

La aplicación necesita establecer comunicación de datos con los instrumentos, por medio de los buses GPIB o USB, red LAN o un puerto de comunicaciones RS232. La capa en la figura ?? conocida como *Capa de Abstracción de Comunicaciones* es la encargada de gestionar todo lo relativo a la comunicación de datos a través de múltiples interfaces o buses disponibles, mientras brinda una interfaz uniforme a las capas de software superiores.

La funcionalidad de los instrumentos que componen el SMR será representada en software por unidades funcionales que se llamarán *bloques de tareas*. El usuario, por medio de la interfaz gráfica, podrá configurar la instrumentación agregando, conectando y programando bloques de tareas. Es responsabilidad del paquete *Procesamiento de Bloques de Tareas* la compilación y ejecución de la programación hecha por el usuario.

Los instrumentos presentes en el SMR serán representados en la interfaz gráfica de la aplicación mediante diagrama de bloques. Es responsabilidad del paquete *Interfaz de Usuario* la gestión de la presentación de diagramas de bloques.

La aplicación CENDITLAB presentará los resultados de una tarea de medición en pantalla o en documentos digitales en formato de documento portable (pdf), en lenguaje de marcado de hipertexto (html) o en formato de texto plano, como valores separados por comas (csv). El paquete *DocumentGenerator* se encargará de cubrir esta funcionalidad.

## 2.2. Funciones de producto

Las tareas de medición que realizaría un usuario por medio de las interfaces físicas empleando de los instrumentos que conforman SMR podrán realizarse a través CENDITLAB, la cual brindará una única interfaz de software al SMR.

A través de CENDITLAB no se necesita presencia física del usuario en el SMR, la aplicación brindará acceso remoto a través de buses de comunicación de datos o por red LAN.

La aplicación se encargará de buscar y presentar al usuario los instrumentos que forman SMR que se encuentren en línea, es decir, conectados a un bus de datos o a una red LAN, a la cual la PC donde se ejecute la aplicación tenga la facilidad de acceso.

La funcionalidad principal de CENDITLAB será la de automatizar el proceso de medición con el SMR.

La interfaz gráfica de la aplicación permitirá al usuario seleccionar los instrumentos en línea y así configurar la instrumentación adecuada para la tarea de medición, por medio de la representación gráfica de los instrumentos en forma de diagrama de bloques. Estos bloques admiten programación en forma de comandos SCPI, estándar para instrumentos programables.

El usuario podrá ejecutar la secuencia de comandos y observar los resultados en tiempo real en pantalla. La aplicación podrá además guardar los resultados de la mediciones en formatos de documento digital como pdf, html o texto plano.

## 2.3. Características de los usuarios

La aplicación CENDITLAB está dirigida a usuarios con formación técnica en el área de radio frecuencia, comunicaciones, antenas, electrónica y afines. En estos campos se consideran ingenieros y técnicos universitarios como *usuarios técnicos* en este documento.

## 2.4. Restricciones

La aplicación CENDITLAB requiere que en el PC o dispositivo donde se ejecute se encuentre instalada la *maquina virtual* adecuada y con la versión correcta (Java Virtual Machine, JVM para Java o el entorno .NET ).

CENDITLAB hace uso intensivo de las comunicaciones a través de los buses GPIB, USB o través de una red LAN. Por ello en el dispositivo o PC donde se ejecuta la aplicación se requiere de los periféricos de hardware apropiados, como adaptadores USB a GPIB o puente LAN a GPIB. En software también se necesita el soporte apropiado para las comunicaciones de datos, requiere de las librerías VISA de National Instruments, Linux-GPIB, VXI-11 y libUSB.

## 2.5. Suposiciones y dependencias

Se parte del supuesto de que el dispositivo donde se ejecute la aplicación posee la cantidad de memoria adecuada y la capacidad de procesamiento necesario para que la maquina virtual ejecute la aplicación, sin una degradación del desempeño apreciable.

# 3. Requerimientos específicos

Esta sección presenta todos los requerimientos funcionales y de calidad para la aplicación CENDITLAB. Muestra una descripción detallada del sistema y de todas sus características.

## 3.1. Requerimientos de interfaces externas

### 3.1.1. Interfaces de usuario

Cuando el usuario inicia la aplicación, se mostrará la pantalla inicial de la figura ???. Esta ventana principal de la aplicación, representa el centro de control de la aplicación ya que concentra múltiples funciones en forma de controles de ventana, comúnmente llamados *widgets*, que el usuario debe seleccionar para ejecutar la acción asociada a este control.

En la pantalla un control en la ventana principal de la aplicación, será llevado a otra ventana con la funcionalidad asociada al control. Por ejemplo, al seleccionar el control *Buscar Instrumentos* se le presentará la ventana de *Explorador de Instrumentos*.

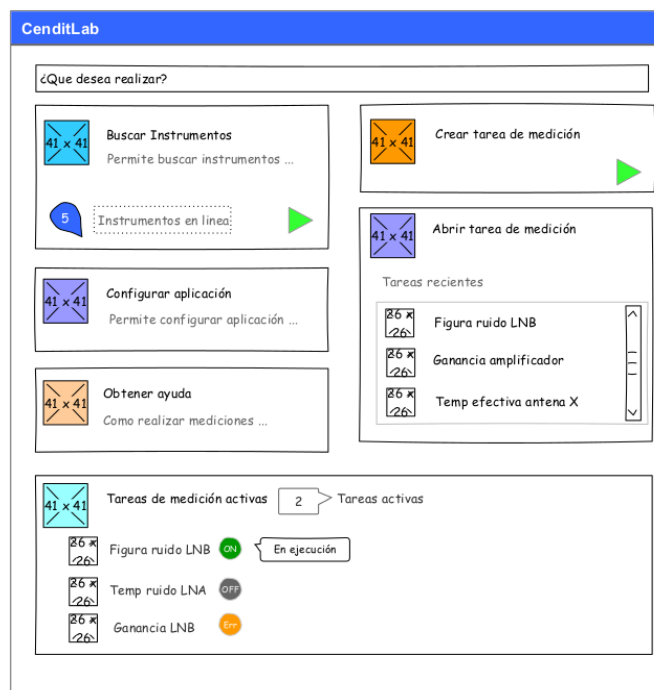


Figura 2: Sketch para la ventana principal de la aplicación

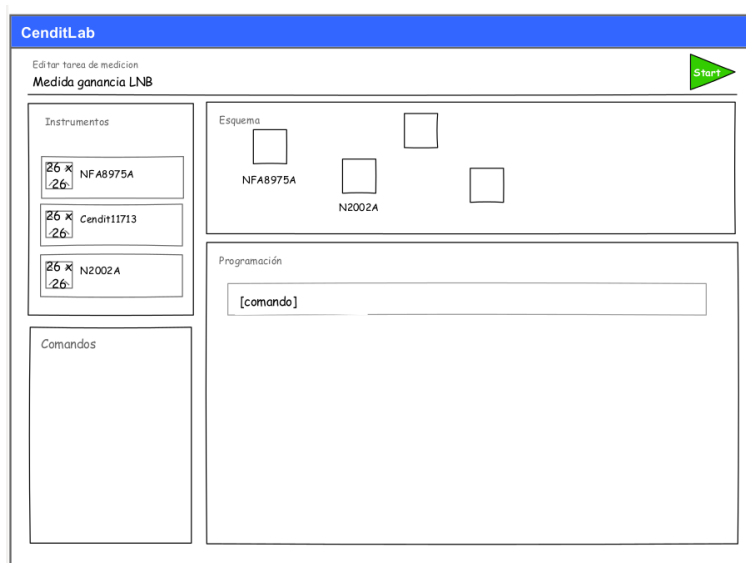


Figura 3: Sketch para la ventana de programación de tareas

Al seleccionar *Crear tarea* el usuario sera llevado a la ventana de programación de tareas de la figura 3. Por medio del control *Abrir tarea* el usuario puede cargar una tarea de medición, previamente almacenada, a la ventana de programación. Presionando el control *Configurar aplicación* se presentará

la ventana en donde el usuario puede ajustar la configuración global de la aplicación. Presionando el control *Obtener ayuda* se presentará una ventana en donde el usuario podrá indagar por ayuda.

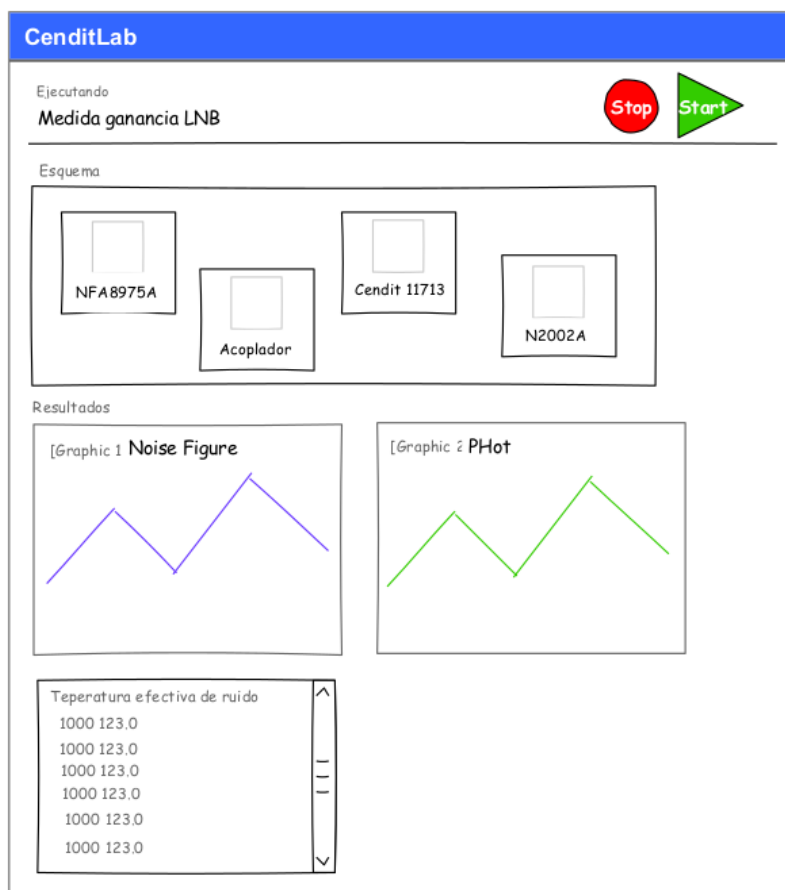


Figura 4: Sketch para la ventana de ejecución de tareas

Dentro de la ventana de programación de tareas de la figura 3 el usuario puede seleccionar los bloques que representan a los instrumentos y arrastrar cada bloque al panel de esquema (parte superior derecha), en donde puede configurar un diagrama de bloques con la instrumentación a utilizar para el proceso de medición. El panel de programación, ubicado en la parte inferior derecha, permite realizar la programación de los instrumentos. El usuario selecciona bloques de instrucciones del panel inferior izquierdo a panel de programación. El usuario edita los comandos IEEE488 o SCPI dentro de los bloques de instrucciones.

Al presionar el botón *start* en la ventana de la figura 3 se procederá a ejecutar el programa y se mostrará la ventana de la figura 4. En esta ventana se muestran los resultados de las mediciones en tiempo real.

### 3.1.2. Interfaces de hardware

La aplicación CENDITLAB requiere un intercambio de datos continuo y confiables con el sistema de medición de ruido. En la figura 5 se muestra un diagrama compuesto UML que indica muestra los *paquetes* de software (CENDITLAB) y de hardware (sistema de medición de ruido) y el *canal de datos*,



puente de comunicación entre estos dos paquetes.

Se tiene una interfaz de hardware, el bus de datos, representada en la figura 5 por los círculos y semicírculos con la etiqueta **Bus Datos**. Esta puede tomar la forma de buses Usb, Gpib, Rs232 o de una red Lan. Las interfaces de hardware la constituyen todos los dispositivos físico que deben estar presentes en el canal de datos, entre el computador que ejecuta la aplicación y los instrumentos que integran el SMR.

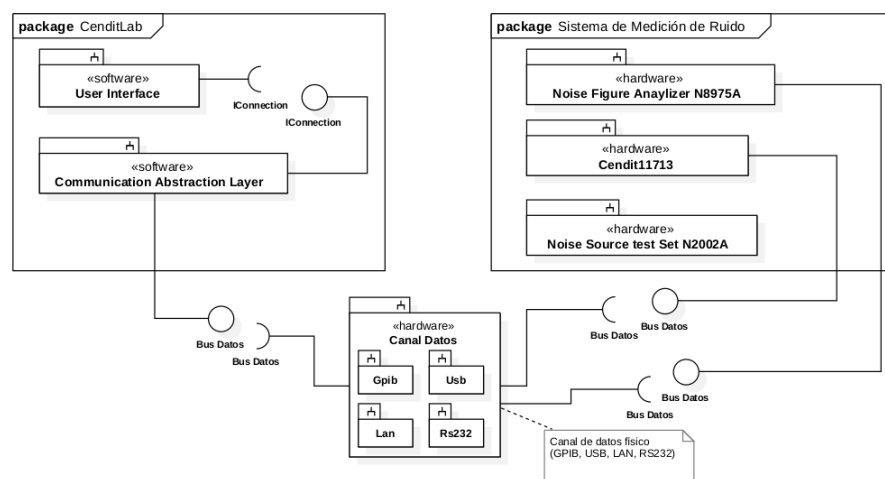


Figura 5: Vista general de interfaces de software y hardware

### 3.1.3. Interfaces de software

Dentro del *marco de paquetes* etiquetado como CENDITLAB de la figura 5 se encuentra la principal interfaz de software, llamada **IConnection**. Esta interfaz representa una abstracción de alto nivel del canal de comunicaciones físicos, bien sea un bus Gpib, Usb, Rs232 o una conexión a red Lan. La interfaz **IConnection** es provista por una capa de software que se ha llamado como **Communication Abstraction Layer** y se indica en la figura 5, interfaz que será utilizada o consumida por niveles superiores dentro del software de CENDITLAB.

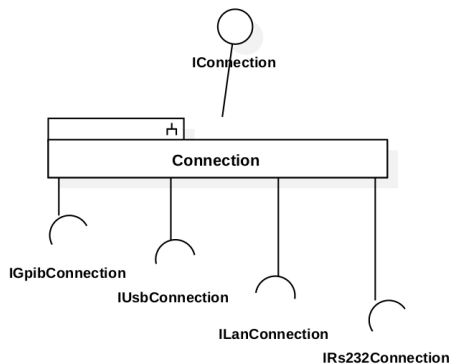


Figura 6: Principales interfaces de software

El paquete de software **Communication Abstraction Layer**, como se muestra en la figura 6, necesita consumir las interfaces **IGpibConnection**, **IUsbConnection**, **ILanConnection**, **IRs232Connection** que presentan funcionalidad de software para acceso a los respectivos buses de datos.

Cada una de las interfaces mencionadas requieren el uso de librerías nativas, instaladas en el sistema operativo donde se ejecute la aplicación. Estas representan interfaces de software que permiten el intercambio de datos a bajo nivel. La dependencia entre las librerías de alto nivel con sus respectivas librerías nativas se indica en las figuras 7 y 8.

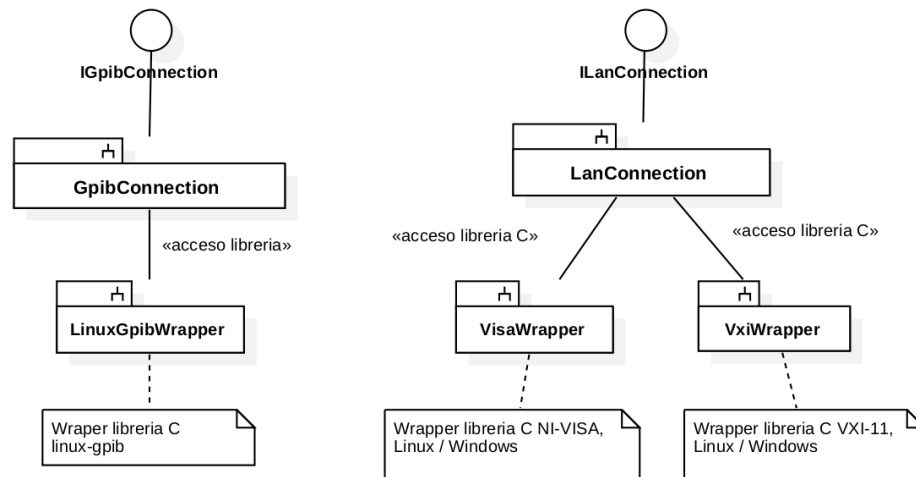


Figura 7: Interfaces de software y sus dependencias, para conexión con buses Gpib y redes Lan

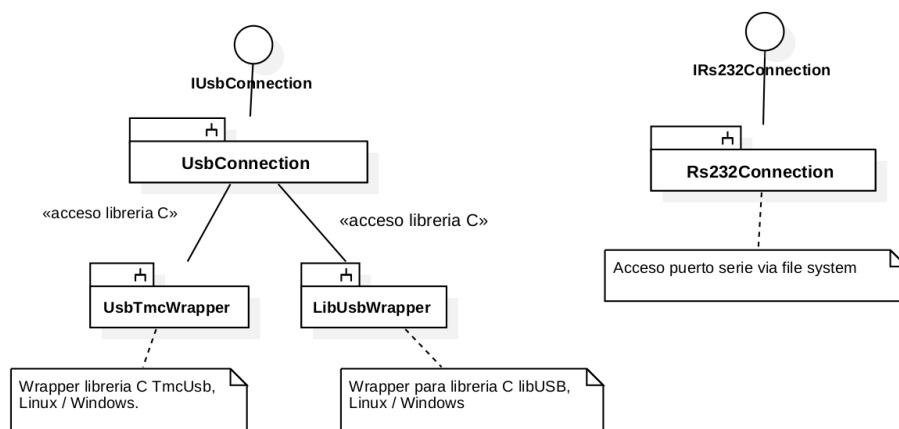


Figura 8: Interfaces de software y sus dependencias, para conexión con buses Usb o Rs232

### 3.1.4. Interfaces de comunicaciones

Las interfaces de comunicaciones para la aplicación CENDITLAB están representadas por el software de interfaz para los buses Gpib, Usb, Rs232 y el software de interfaz para acceso a redes Lan.

## 3.2. Requerimientos funcionales

A continuación los requerimientos funcionales

### 3.2.1. Requerimiento funcional 1

**ID:** FR1

**Titulo:** Automatizar tareas de medición.

**Descripción:** Los usuarios pueden programar los instrumentos y generar tareas de medición para sus posterior ejecución, de forma automatizada y remota.

**Justificación:** Programar y automatizar las mediciones con elSMR.

**Dependencias:** Ninguna.

**Usuario:** Técnico.

### 3.2.2. Requerimiento funcional 2

**ID:** FR2

**Titulo:** Presentar una interfaz gráfica para el SMR.

**Descripción:** La aplicación servirá como una representación integral en software de los tres instrumentos principales que componen el SMR, será un instrumento virtual.

**Justificación:** Simplificar las tares de medición con el SMR.

**Dependencias:** FR1.

**Usuario:** Técnico.

### 3.2.3. Requerimiento funcional 3

**ID:** FR3

**Titulo:** Gestión de las comunicaciones con los instrumentos del SMR de manera simple y uniforme desde la perspectiva de usuario.

**Descripción:** La aplicación facilitará el intercambio de datos con los instrumentos del SMR.

**Justificación:** Simplificar la adquisición de datos del SMR.

**Dependencias:** FR1, FR2.

**Usuario:** Técnico.

### 3.2.4. Requerimiento funcional 4

**ID:** FR4

**Titulo:** Explorador de instrumentos conectados a los buses.

**Descripción:** El usuario podrá visualizar los instrumentos conectados a los buses, al cual el PC tiene acceso.

**Justificación:** Con el fin de mostrar un listado de instrumentos en línea.

**Dependencias:** Ninguna.

**Usuario:** Técnico.

### 3.2.5. Requerimiento funcional 5

**ID:** FR5  
**Título:** Configurar los instrumentos de medición.  
**Descripción:** Al usuario se le presentara una interfaz gráfica en donde podrá establecer la configuración de cada instrumento en línea.  
**Justificación:** Los instrumentos requieren una configuración previa antes de iniciar la medición.  
**Dependencias:** FR2.  
**Usuario:** Técnico.

### 3.2.6. Requerimiento funcional 6

**ID:** FR6  
**Título:** Mostrar una representación gráfica del montaje a utilizar para una tarea de medición, orientada a diagrama de bloques.  
**Descripción:** La aplicación permitirá al usuario describir de forma gráfica la interconexión de instrumentos para una tarea de medición.  
**Justificación:** Servir de ayuda gráfica al usuario para una tarea de medición.  
**Dependencias:** FR2.  
**Usuario:** Técnico.

### 3.2.7. Requerimiento funcional 7

**ID:** FR7  
**Título:** Interfaz gráfica orientada a diagrama de bloques.  
**Descripción:** El usuario puede programar las tareas de medición por medio de la creación de diagramas de bloques, que representan funcionalidades diversas.  
**Justificación:** Con el objeto de automatizar el proceso de medición.  
**Dependencias:** Ninguna.  
**Usuario:** Técnico.

### 3.2.8. Requerimiento funcional 8

**ID:** FR8  
**Título:** Capacidad de programar el proceso medición.  
**Descripción:** El usuario puede elegir un conjunto de instrumentos y establecer una secuencia de comandos para estos, que podrá almacenar en disco y ejecutar en el futuro.  
**Justificación:** Con el objeto de automatizar los procesos de medición.  
**Dependencias:** FR2, FR3, FR4 FR5.  
**Usuario:** Técnico.

### 3.2.9. Requerimiento funcional 9

**ID:** FR9  
**Título:** Capacidad para simular el proceso medición.  
**Descripción:** El usuario puede simular una tarea de instrumentación dada, sin necesidad de conexión con el SMR.  
**Justificación:** Con el objeto de automatizar los procesos de medición.  
**Dependencias:** FR2, FR3, FR4 FR5.  
**Usuario:** Técnico.

### 3.2.10. Requerimiento funcional 10

**ID:** FR11  
**Título:** Capacidad para el montaje y configuración de tareas de medición.  
**Descripción:** El usuario puede salvar la configuración y programación que haya hecho sobre una tarea de medición.  
**Justificación:** Con el objeto de automatizar los procesos de medición.  
**Dependencias:** FR3.  
**Usuario:** Técnico.

### 3.2.11. Requerimiento funcional 11

**ID:** FR11  
**Título:** Capacidad para almacenar y cargar datos de calibración de instrumentos.  
**Descripción:** El usuario puede guardar almacenar los datos de calibración y cargarlos nuevamente antes de cada proceso de medición.  
**Justificación:** Con el objeto de automatizar los procesos de medición.  
**Dependencias:** FR3.  
**Usuario:** Técnico.

### 3.2.12. Requerimiento funcional 12

**ID:** FR12  
**Título:** Capacidad de programar la secuencia de pasos en el proceso de medición.  
**Descripción:** El usuario podrá programar la secuencia de pasos a realizar en un proceso de medición dado.  
**Justificación:** Con el objeto de automatizar el proceso de medición.  
**Dependencias:** FR2, FR3.  
**Usuario:** Técnico.

### 3.2.13. Requerimiento funcional 13

**ID:** FR13

**Título:** Capacidad programar la presentación gráfica reporte con resultados de medición.

**Descripción:** El usuario podrá escoger los datos resultado de la medición y configurar su presentación y disposición en el documento de salida.

**Justificación:** Con el objeto de configurar el documento que presenta los resultados de medición.

**Dependencias:** FR3,FR4.

**Usuario:** Técnico.

## 3.3. Requerimientos de desempeño

### 3.3.1. Requerimiento de desempeño 1

**ID:** QR1

**Título:** Diseño de interfaz gráfica limpio, des congestionado y ordenado.

**Descripción:** La interfaz gráfica deberá presentar un diseño limpio y estructurado, con funcionalidad común agrupada.

**Justificación:** Con el objeto de facilitar la navegación por las pantallas de la aplicación.

**Dependencias:** Ninguna.

### 3.3.2. Requerimiento de desempeño 2

**ID:** QR2

**Título:** La aplicación para PC deberá ser portable entre los SO Windows y Linux.

**Descripción:** La aplicación debe ejecutarse de manera uniforme en los sistemas operativos Windows y Linux.

**Justificación:** Facilitar al usuario el uso de la aplicación en ambos sistemas operativos..

**Dependencias:** Ninguna.

## 3.4. Restricciones de diseño

[Por completar]

## 3.5. Atributos del sistema de software

[Por completar]

## 4. Asignación de prioridades y plan de entregas

[Por completar]

### 4.1. Elección del método para asignar prioridades

[Por completar]

## 4.2. Plan de entregas

[Por completar]