ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL.

Facultad de Ingeniería en Electricidad y Computación.

"SISTEMA DE SEGURIDAD DE EQUIPOS DE LABORATORIO."

INFORME DE MATERIA DE GRADUACIÓN:

Previo a la obtención del Título de:

INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:
Juan Javier Domínguez Espinoza.
Miguel Ángel Iturralde Durán.

GUAYAQUIL – ECUADOR
2009

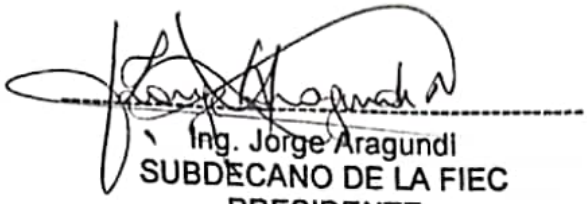
A G R A D E C I M I E N T O

Primeramente a Dios quien ha sido mi guía y supo darme fuerzas en los momentos más difíciles.
A mis padres, ya que gracias a sus sabios consejos pude encaminarme para convertirme en un hombre de bien y jamás dejaron de creer en mí, apoyándome siempre, tanto en los buenos tiempos como en los mas difíciles de mi vida.
A mis hermanos, quienes siempre me apoyaron y estuvieron a mi lado incondicionalmente.

D E D I C A T O R I A

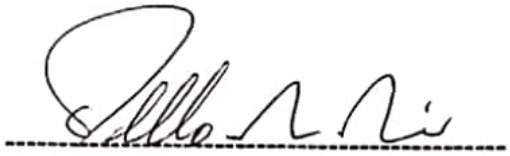
El trabajo, sacrificio y esfuerzo aplicado a este proyecto se los dedico a Dios y a mis padres. A Dios por ser guía de mi vida y a mis padres a quienes llevo dentro de mi corazón.

TRIBUNAL DE GRADUACIÓN



Ing. Jorge Aragundi
SUBDECANO DE LA FIEC
PRESIDENTE

Ing. Carlos Valdivieso A.
DIRECTOR DE TESIS



Ing. Hugo Villavicencio
VOCAL PRINCIPAL

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesis de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL)

Juan Domínguez

Miguel Iturralde

RESUMEN.

Este proyecto presenta como objetivo principal el desarrollo de un sistema de seguridad para los equipos de laboratorio. Esto consiste en el diseño de un sistema que permite un monitoreo constante de los equipos en cuestión, así como también, la implementación y puesta a prueba de un prototipo diseñado a menor escala que emule el comportamiento de dicho sistema.

En el primer capítulo de este texto se describen el alcance y las limitaciones que se puedan presentar durante la implementación de este sistema. Este capítulo nos proporciona una idea general del funcionamiento del sistema y de sus posibles fallas, así como también se mencionan soluciones similares utilizadas en implementaciones comerciales existentes en el mercado.

En el segundo capítulo se hace un análisis teórico de las tecnologías utilizadas en el diseño del sistema de seguridad. Estas tecnologías consisten en: un sensor de radiofrecuencia denominado RFID, que permite la identificación a distancia de cualquier objeto que posea una etiqueta transmisora adherida a el y un módulo de comunicación Ethernet. De igual manera, este sistema utiliza el software de instrumentación virtual LabVIEW que permite desarrollar una interfaz entre el usuario y el sistema. Las características y herramientas de dicho Software se presentarán en este capítulo.

En el tercer capítulo se presentarán los detalles del software y hardware utilizados durante el desarrollo del sistema. Para hacer una descripción detallada del

funcionamiento del sistema se recurrirá a un diagrama de bloques que explicará de manera sencilla la forma en que las distintas tecnologías electrónicas implementadas interactúan de tal manera que sea posible un desenvolvimiento armónico, eficiente e integral de cada uno de los elementos que conforman al sistema de seguridad implementado.

También se presentará el circuito impreso del prototipo implementado a menor escala, sus partes y los detalles de su funcionamiento, así como también el código estructurado en lenguaje G, que permitirá la comunicación del sistema con el usuario y con las distintas herramientas utilizadas que permiten el correcto funcionamiento del sistema.

Finalmente, el cuarto capítulo se refiere al funcionamiento del sistema, utilizando al prototipo desarrollado a menor escala como un módulo que emule el comportamiento del sistema diseñado, y de igual manera se presenta una guía del manejo y utilización del programa desarrollado en LabVIEW que permitirá la comunicación entre el sistema y el administrador del laboratorio.

ÍNDICE GENERAL

INTRODUCCIÓN.

Debido a que los costos de los equipos utilizados en los laboratorios son en su mayoría elevados, y a que en muchas ocasiones las dimensiones de dichos equipos podrían facilitar la sustracción de los mismos, la probabilidad de que estos equipos sean retirados de las inmediaciones del laboratorio, sin previo consentimiento del administrador del mismo, es muy elevada. Con el fin de minimizar las posibilidades de hurto de dichos equipos se ha desarrollado el presente proyecto, el cual consiste en el diseño de un sistema de seguridad para los equipos de laboratorio y la implementación de un prototipo desarrollado a menor escala de dicho sistema. La característica principal de este sistema es garantizar la estadía de los equipos en el área del laboratorio y permitir un monitoreo regular del sistema. Para lograrlo recurriremos a tecnologías de identificación por radiofrecuencia y a la implementación de chips microcontroladores, además se utilizará LabVIEW como software de instrumentación virtual que será la interfaz de comunicación entre el sistema y el administrador del laboratorio.

En el primer capítulo de este texto, presentado a continuación, se describirá en detalle el alcance y las limitaciones del presente proyecto

CAPÍTULO 1

DESCRIPCIÓN GENERAL DEL SISTEMA.

1.1 Alcance y limitaciones del proyecto.

El objetivo principal de este proyecto es el de desarrollar un sistema de seguridad para los equipos de laboratorio. Para lograrlo recurriremos a la implementación de tecnologías basadas en dispositivos microcontroladores, sistemas de comunicaciones inalámbricas utilizando la tecnología de identificación por radiofrecuencia RFID, dispositivos de comunicación serial basados en el protocolo de telecomunicaciones Ethernet, manejo de software de administración y gestión de bases de datos, y desarrollo de aplicaciones específicas utilizando LabVIEW como herramienta de instrumentación virtual.

1.1.1 Descripción del Proyecto.

Como se mencionó previamente, nuestro objetivo es desarrollar un Sistema de Seguridad para los equipos que se utilizan en los laboratorios. Este Sistema de seguridad consiste en garantizar la permanencia de los equipos en las inmediaciones del laboratorio, es decir que, el usuario tenga la certeza de que cada uno de los

equipos protegidos con este sistema de seguridad se encuentren dentro de un área específica definida por el usuario del sistema en cuestión.

Las principales características de este proyecto son:

- Garantiza la permanencia de los equipos del laboratorio en las inmediaciones del mismo.
- Proporciona una interfaz amigable con el usuario.
- El tiempo de instalación del sistema es reducido.

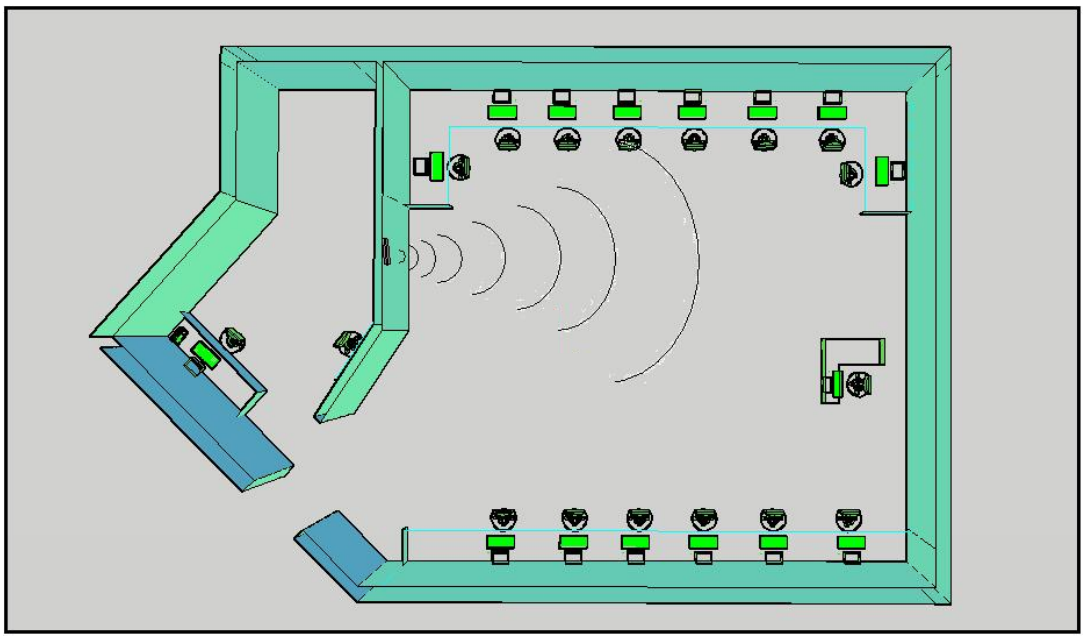


Figura 1.1 Laboratorio de Microcontroladores.

1.1.2 Estrategia Implementada.

El diseño consiste en la implementación de un sistema de comunicación inalámbrico que permitirá el monitoreo constante de los equipos que se encuentren bajo su administración. Este sistema está compuesto por un dispositivo transmisor y un dispositivo receptor de señales de radiofrecuencia. El dispositivo receptor deberá ubicarse en un punto estratégico del laboratorio de tal manera que se garantice una apropiada recepción de la señal de radiofrecuencia emitida por el transmisor, el cual está diseñado de tal manera que se pueda insertar en cada uno de los equipos cuya seguridad se desea garantizar. Este dispositivo transmisor de dimensiones reducidas y con la capacidad de insertarse en los equipos del laboratorio se denomina etiqueta o tag, la etiqueta emite una señal de identificación al receptor de manera periódica, de tal forma que el administrador del laboratorio puede monitorear de manera continua la permanencia de los equipos en las inmediaciones del laboratorio. Para lograrlo es necesario crear una interfaz que permita la comunicación entre el dispositivo receptor y el administrador del laboratorio.

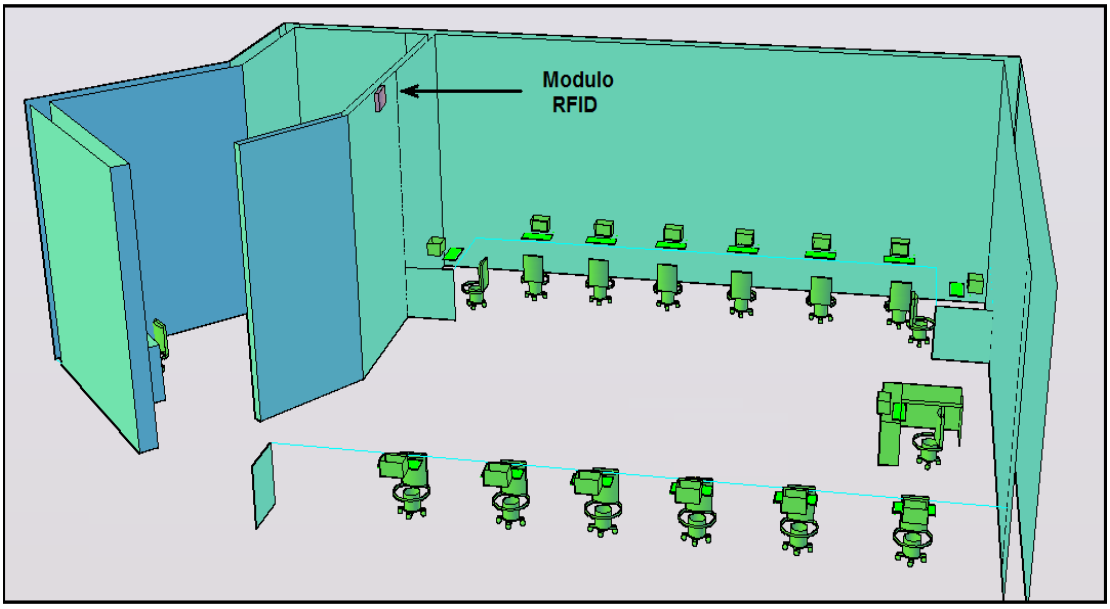


Figura 1.2 Ubicación del Módulo RFID.

Los detalles del diseño y funcionamiento de esta interfaz se describirán en capítulos posteriores.

El administrador del laboratorio podrá monitorear el sistema a través de un software de administración y gestión de bases de datos, en este caso particular utilizaremos la aplicación MySQL, que es un software que permite manejar bases de datos de manera gráfica y con una interfaz amigable para el usuario.

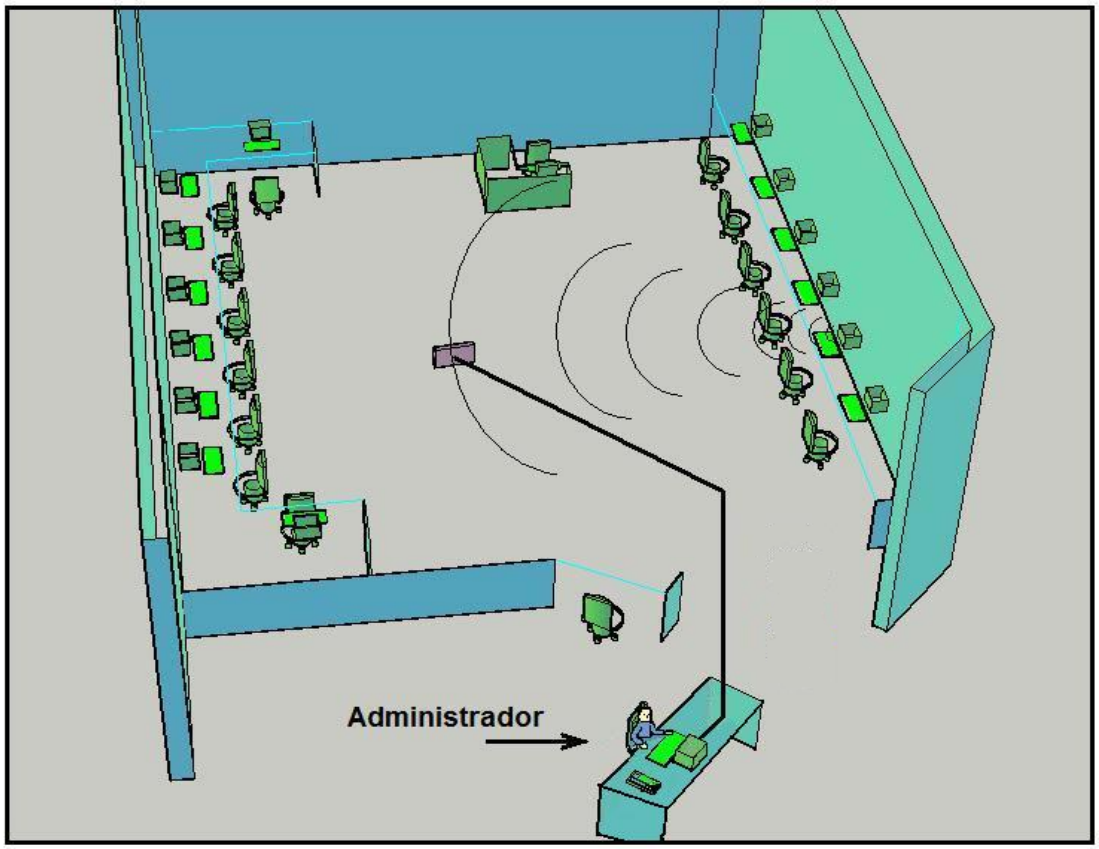


Figura 1.3 Funcionamiento del Sistema.

Si algún equipo que posea una etiqueta transmisora sale del área de cobertura del dispositivo receptor, el sistema emitirá una señal de alerta que le permitirá al administrador del laboratorio tomar las medidas pertinentes para evitar la salida del equipo en cuestión de las inmediaciones del laboratorio. Puesto que las etiquetas tienen un número de identificación único será posible determinar con exactitud cuál fue el equipo que salió del laboratorio en el preciso momento en que este abandonó las inmediaciones del mismo.

1.1.3 Limitaciones del Proyecto.

Este proyecto contiene algunas limitantes que es recomendable tener en consideración al momento de implementar el mismo, a continuación presentamos algunas de estas posibles limitantes y sus probables consecuencias:

- **Acceso a la Tecnología requerida.-** Tomando en consideración las limitaciones tecnológicas que padece nuestro país, es posible que haya inconvenientes en la adquisición de los módulos de identificación por radiofrecuencia, necesarios para la implementación del proyecto a gran escala, puesto que los distribuidores de equipos electrónicos no siempre tienen disponible en sus perchas los dispositivos necesarios para la implementación inmediata del proyecto, esto podría traer como consecuencia un retraso en la instalación del sistema en cuestión. Sin embargo es una limitante que, aunque podría causar inconvenientes, no se debe considerar como un factor restrictivo en la implementación del proyecto, pues en caso de no existir el artículo necesario en el mercado local, es posible adquirirlo en el extranjero pero a un mayor costo y con posibles demoras en la entrega del mismo.

- **Costo.-** Los costos en la implementación del proyecto variarán según el área de cobertura y la eficiencia operativa del sistema que el usuario desee

implementar. Esta podría ser una limitante restrictiva, pues en la actualidad existen diversas alternativas comerciales, que a pesar de ser menos efectivas, podrían proporcionar costos más accesibles al consumidor final.

- **Se requiere la presencia de un administrador del laboratorio.-** El sistema de seguridad desarrollado se limita a alertar al administrador del laboratorio sobre el posible robo de uno o varios equipos de laboratorio en el momento preciso de sucedido aquel acontecimiento, sin embargo cabe recalcar, que debe ser el administrador quien tome las medidas necesarias para evitar la salida de los equipos del área del laboratorio una vez que el sistema ha activado la señal de alerta.

- **Interferencia.-** Puesto que el sistema desarrollado utiliza módulos de identificación por radiofrecuencia, que en la mayoría de los casos operan en bandas de frecuencia no licenciadas, es posible que el desempeño del sistema se vea afectado por interferencias producidas por agentes externos. En caso de suceder esto, será necesario cambiar la frecuencia de operación a un rango que se encuentre libre de interferencias.

- **Cobertura.-** Es probable que el módulo de identificación por radiofrecuencia no tenga la cobertura requerida para la correcta implementación de un proyecto en particular. En cuyo caso será necesario adquirir un Lector RFID de mayor potencia, que opere en una banda de frecuencia que proporcione la cobertura necesaria.

Una vez analizadas las posibles limitaciones del proyecto, cabe recalcar que debido a restricciones presupuestarias y comerciales, nos resultará imposible implementar el hardware a escala real, pues para ello se requiere utilizar tecnología RFID de tipo activa, cuyos costos son sumamente elevados, además que esta tecnología actualmente no se ha posicionado de manera comercial en nuestro país, por lo que nos vemos limitados a desarrollar un prototipo a menor escala utilizando dispositivos RFID de tipo pasivos cuyos costos y comercialización son actualmente accesibles e ideales para nuestros propósitos.

Además, debido a que vamos a utilizar tecnología RFID de tipo pasiva se hará una pequeña variante en la implementación en hardware del prototipo desarrollado a menor escala, pues el sistema en lugar de dar la señal de alerta al detectar la ausencia de la señal proveniente de una de las etiquetas en su área de cobertura, como se explicó anteriormente, lo hará al ingresar la señal proveniente del tag al área de cobertura del lector RFID. Esta variante se hace debido a que el

lector RFID pasivo no tiene la cobertura requerida para poder implementar el sistema a escala real.

Figura 1.4 Módulo RFID Activo Kimaldi SYRD245-1N. [1]

Sin embargo, esta limitante no afecta en mayor manera la aplicabilidad del diseño propuesto en una escala real, pues si se desea implementar este diseño a mayor escala se puede hacer con absoluta normalidad, con la diferencia de que el lector RFID en lugar de detectar la presencia de una señal de radiofrecuencia, detectará la ausencia de la misma en sus inmediaciones, y además en lugar de utilizar un sistema RFID pasivo, se deberá adquirir un módulo RFID de tipo activo, preferiblemente un sistema desarrollado por la marca kimaldi.

Se recomienda adquirir el lector RFID activo SYRD245-1N desarrollado por kimaldi, ya que este equipo permite la lectura de tags activos a distancias comprendidas entre los 8 y 10m, opera en la banda no licenciada de 2.45 GHz y

---

Fuente: www.iberhardware.es[1]

tiene capacidad de lectura multitag, lo cual implica que podrá monitorear varias etiquetas de manera simultánea. El costo de este producto esta entre los €750 y €770.

1.2 Análisis de soluciones similares existentes en el mercado.

Actualmente en el mercado existe una gama de productos que ofrecen soluciones similares utilizando diferentes tecnologías, muchas de estas opciones ofrecen servicios más elaborados pero con costos restrictivos, mientras otros ofrecen precios accesibles pero menor eficiencia, todo depende de la capacidad adquisitiva del usuario y de sus necesidades. A continuación presentamos brevemente algunas soluciones comerciales similares al proyecto en cuestión:

1.2.1 Sistema de alarma contra intrusión.- Son sistemas de seguridad que incorporan tecnologías diseñadas para conectarse con un sistema central de alarmas que alerta a los usuarios de una posible incursión no autorizada. Estos sistemas son utilizados para evitar robos durante la ausencia del personal administrativo en las instalaciones protegidas, en este caso particular nos referimos al laboratorio. Este tipo de solución comercial ofrece además del sistema de alarma, un servicio de seguridad que proporciona respuesta inmediata ante el aviso de alerta. Se caracteriza por

ser muy costoso, pues el usuario tiene que cubrir el costo de la implementación de la tecnología y el servicio de seguridad inmediata en caso de activarse la alarma. Este tipo de sistemas utilizan detectores de intrusión o detectores periféricos conectados a una central de alarma.

1.2.2 Sistema de control de acceso.- Este tipo de sistemas de seguridad controlan el acceso de personas a un lugar determinado con el fin de precautelar la seguridad dentro de un área específica. En esta solución en particular se procura garantizar que el acceso al laboratorio se de únicamente por personas autorizadas (personal docente, estudiantes, personal administrativo, etc.). Esto se logra utilizando controles de acceso por tarjetas o códigos numéricos utilizando lectores de diferentes tecnologías, como bandas magnéticas detectores de proximidad, lectores enlazados a una unidad central, entre otros. De esta manera se garantiza que las personas que entren o salgan del laboratorio estén autorizadas para hacerlo y lo hagan previo consentimiento del administrador del laboratorio.

1.2.3 Incorporación de circuitos cerrados de TV (CCTV).- Los circuitos cerrados de TV permiten una vigilancia constante durante las horas laborales y el registro de eventos durante el cierre del laboratorio. Este sistema de seguridad ofrece varias ventajas como son la verificación al instante de la

causa del encendido de la alarma, identificación de un intruso, entre otras. Estos sistemas utilizan tecnologías captadoras de imágenes (cámaras), elementos reproductores de imágenes (monitores), videograbadoras, video sensores, entre otros. Todas estas características hacen de este sistema una opción eficiente pero al mismo tiempo sumamente costosa, lo que la convierte en una opción poco viable, si el presupuesto es limitado.

CAPÍTULO 2

ANÁLISIS TEÓRICO DEL DISEÑO PROPUESTO.

2.1 Revisión de la Base Teórica.

2.1.1 ¿Qué es la identificación por Radiofrecuencia?

El Sistema de Identificación por Radiofrecuencia o RFID (Radio Frequency Identification) es un sistema inalámbrico de almacenamiento y recuperación de datos que usa ondas de radio para determinar la identificación de pequeños dispositivos denominados etiquetas, transpondedores o tags RFID. La tecnología RFID requiere de dos dispositivos específicos para operar apropiadamente, un lector RFID y un tag o etiqueta RFID. El Tag RFID está compuesto por un microchip que almacena en su circuitería interna un número serial único denominado ID que lo identifica y puede adherirse o incorporarse a un producto o persona específico. Este microchip está acoplado a una antena que le permite recibir y responder a peticiones por radiofrecuencia desde un emisor-receptor RFID. El tag RFID debe trabajar en combinación con el lector RFID, el cual procesa los datos obtenidos y los envía al siguiente bloque de procesamiento de datos.

2.1.2 ¿Cómo funciona la identificación por Radiofrecuencia?

El modo de funcionamiento de los sistemas RFID es simple. La etiqueta RFID, que contiene los datos de identificación del objeto al que se encuentra adherido, genera una señal de radiofrecuencia con el ID de dicho objeto. Esta señal puede ser captada por un lector RFID, el cual se encarga de leer la información y pasarla en formato digital a la aplicación específica que utiliza RFID.

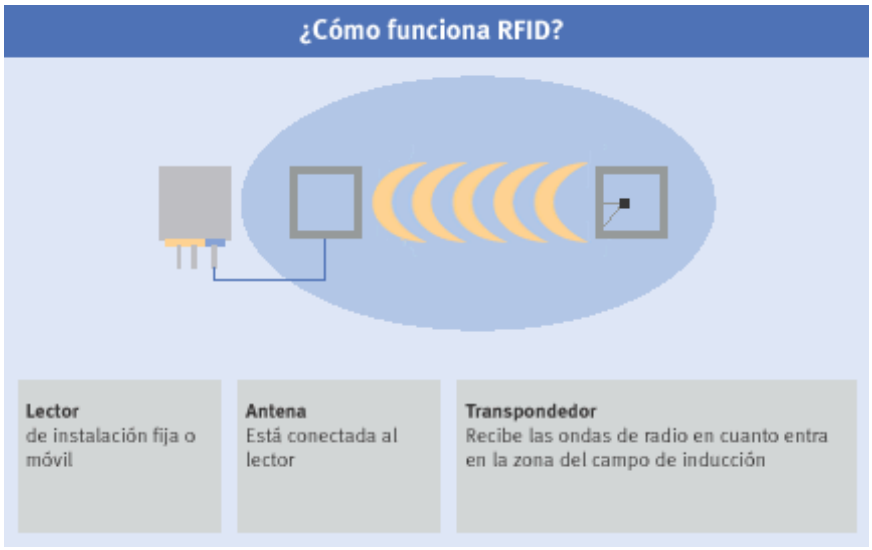


Figura 2.1 Funcionamiento de un sistema RFID.[2]

Un sistema RFID consta de los siguientes tres componentes:

- **Etiqueta RFID** o **transpondedor**: Como se mencionó anteriormente el tag

[2] Fuente: www.wikipedia.org

RFID está compuesto por una antena y un material encapsulado o chip. El propósito de la antena es permitirle al chip, el cual contiene almacenado el ID, transmitir la información de identificación de la etiqueta. El chip posee una memoria interna con una capacidad que depende del modelo. Existen dos tipos de memoria:

  - **Sólo lectura**: el código de identificación que contiene es único y es personalizado durante la fabricación de la etiqueta.
  - **De lectura y escritura**: la información de identificación puede ser modificada por el lector.

- **Lector de RFID** o **transceptor**: compuesto por una antena, un transceptor y un decodificador. El lector envía periódicamente señales para ver si hay alguna etiqueta en sus inmediaciones. Cuando capta una señal de una etiqueta, extrae la información y se la pasa al subsistema de procesamiento de datos.

- **Subsistema de procesamiento de datos o Middleware RFID**: proporciona los medios de proceso y almacenamiento de datos.

Ventajas de RFID

- Identificación sin contacto.
- Traspasa distintos materiales.
- Se pueden leer y registrar datos en la memoria según se necesite
- Identificación en menos de un segundo
- Captura simultánea de muchos transpondedores
- Resistente a condiciones adversas (temperaturas extremas, humedad, etc.)
- La forma y el tamaño del transpondedor se pueden adaptar según las necesidades
- Los transpondedores se pueden integrar completamente en el producto

Tipos de Tags RFID.

Los tags RFID pueden ser activos, semipasivos (también conocidos como semiactivos o asistidos por batería) o pasivos. Los tags pasivos no requieren ninguna fuente de alimentación interna y sólo se activan cuando un lector se encuentra cerca para suministrarles la energía necesaria. Los otros dos tipos necesitan alimentación, típicamente una pila pequeña. Los Tags semipasivos utilizan la batería únicamente para alimentar su circuitería interna, más no para generar la frecuencia de emisión como si ocurre con los tags de tipo activo.

La gran mayoría de las etiquetas RFID son pasivas, que son mucho más baratas de fabricar y no necesitan batería.

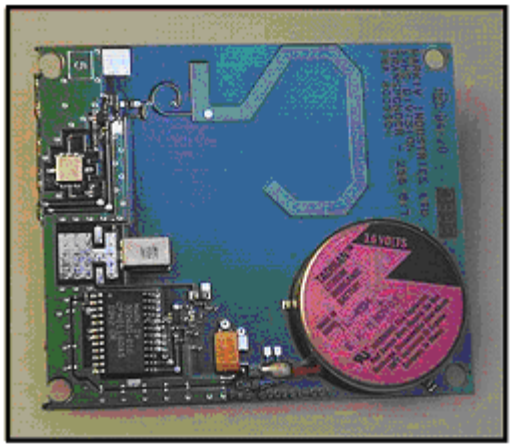


Figura 2.2 Tag Semipasivo[3].

A pesar de que las ventajas en cuanto al costo de las etiquetas RFID pasivas con respecto a las activas son significativas, otros factores; incluyendo exactitud, funcionamiento en ciertos ambientes como cerca del agua o metal, y confiabilidad; hacen que el uso de etiquetas activas sea muy común hoy en día.

Tags Pasivos.

Los tags pasivos no poseen alimentación eléctrica. La señal que les llega de los lectores induce una corriente eléctrica pequeña y suficiente para operar el circuito integrado del

[3] Fuente: www.wikipedia.org

tag, de forma que puede generar y transmitir una respuesta. Los tags pasivos suelen tener distancias de uso práctico comprendidas entre los 10 mm hasta cerca de 6 metros, dependiendo de la frecuencia de funcionamiento, el diseño y tamaño de la antena. Como no precisan de alimentación energética, el dispositivo puede resultar muy pequeño: pueden incluirse en una pegatina o insertarse bajo la piel (tags de baja frecuencia).

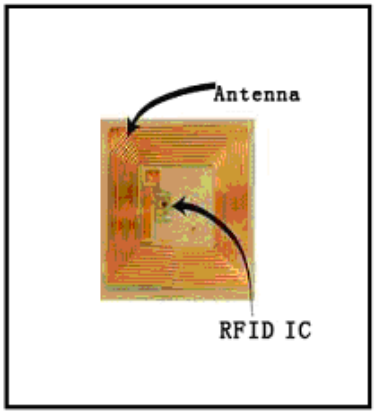


Figura 2.3 Tag Pasivo.[4]

Tags Activos.

A diferencia de los tags pasivos, los activos poseen su propia fuente autónoma de energía, que utilizan para dar corriente a sus circuitos integrados y propagar su señal al lector. Estos tags son mucho más fiables (tienen menos errores) que los pasivos. Gracias a su fuente de energía son capaces de transmitir señales más potentes que las de los tags pasivos, lo que les lleva a ser más eficientes en entornos dificultosos para la radiofrecuencia como el agua (incluyendo humanos y ganado, formados en su mayoría

[4] Fuente: www.wikipedia.org

por agua), metal (contenedores, vehículos). También son efectivos a distancias mayores pudiendo generar respuestas claras a partir de recepciones débiles. Pero a diferencia de los tags pasivos, suelen ser más caros, y su vida útil es en general mucho más corta.

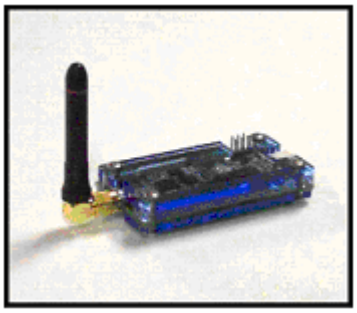


Figura 2.4 Tag Activo.[5]

Muchos tags activos tienen rangos efectivos de cientos de metros y una vida útil de sus baterías de hasta 10 años, además de mucho más rango (500 m), tienen capacidades de almacenamiento mayores y la habilidad de guardar información adicional enviada por el transceptor. Actualmente, las etiquetas activas más pequeñas tienen un tamaño aproximado de una moneda. Muchas etiquetas activas tienen rangos prácticos de diez metros, y una duración de batería de hasta varios años.

[5] Fuente: www.wikipedia.org

Clasificación:

Los sistemas RFID se clasifican dependiendo del rango de frecuencias que usan. Existen cuatro tipos de sistemas: De frecuencia baja o LF (entre 125 ó 134,2 KHz); de alta frecuencia o HF (13,56 MHz); UHF o de frecuencia ultra elevada (868 a 956 MHz); y de microondas (2,45 GHz).

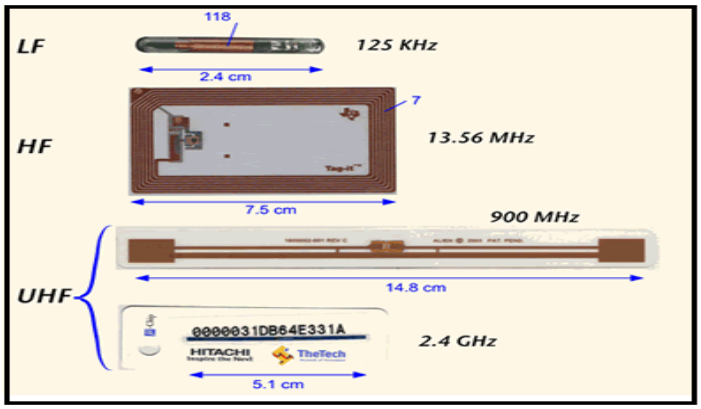


Figura 2.5 Clasificación de Sistemas RFID.[6]

2.1.3 Aplicaciones comerciales que utilizan RFID.

Dependiendo de las frecuencias utilizadas en los sistemas RFID, el costo, el alcance y las aplicaciones son diferentes. Los sistemas que emplean frecuencias bajas tienen igualmente costos bajos, pero también baja distancia de uso. Los que emplean frecuencias más altas proporcionan distancias mayores de lectura y velocidades de

[6] Fuente: www.wikipedia.org

lectura más rápidas. Así, las de baja frecuencia se utilizan comúnmente para la identificación de animales, seguimiento de mercadería, o como llave de automóviles con sistema antirrobo. En los Estados Unidos se utilizan dos frecuencias para RFID: 125 Khz. (el estándar original) y 134,5 Khz. (el estándar internacional). Las etiquetas RFID de alta frecuencia se utilizan en bibliotecas y seguimiento de libros, control de acceso en edificios, seguimiento de equipaje en aerolíneas, seguimiento de artículos de ropa y ahora último en pacientes de centros hospitalarios para hacer un seguimiento de su historia clínica.

Figura 2.6 Etiqueta RFID.[7]

Actualmente, la aplicación más importante de RFID es la logística. El uso de esta tecnología permitiría tener localizado cualquier producto dentro de la cadena de suministro. En lo relacionado a la trazabilidad, las etiquetas podrían tener gran aplicación ya que las mismas pueden grabarse, con lo que se podría conocer el tiempo que el

[7] Fuente: www.wikipedia.org

producto estuvo almacenado, en que sitios, etc. De esta manera se pueden lograr importantes optimizaciones en el manejo de los productos en las cadenas de abastecimiento teniendo como base el mismo producto, e independizándose prácticamente del sistema de información.

Implantes Humanos.

Los chips RFID implantables, diseñados originalmente para el etiquetado de animales se está utilizando y se está contemplando también para los seres humanos. Applied Digital Solutions propone su chip "*unique under-the-skin format*" (formato bajo-la-piel único) como solución a la usurpación de la identidad, al acceso seguro a un edificio, al acceso a un ordenador, al almacenamiento de expedientes médicos, a iniciativas de anti-secuestro y a una variedad de aplicaciones. Sin embargo, el implante de los chips supone un elevado riesgo para la salud, ya que resultan altamente cancerígenos.

Figura 2.7 Implante de un chip RFID[8]

[8] Fuente: www.wikipedia.org

Polémicas sobre su utilización.

El uso de la tecnología RFID ha causado una considerable polémica. Las cuatro razones principales por las que RFID resulta preocupante en lo que a privacidad se refiere son:

- El comprador de un artículo no tiene por qué saber de la presencia de la etiqueta o ser capaz de eliminarla.
- La etiqueta puede ser leída a cierta distancia sin conocimiento por parte del individuo.
- Si un artículo etiquetado es pagado mediante tarjeta de crédito, entonces sería posible enlazar la ID única de ese artículo con la identidad del comprador.

La mayoría de las preocupaciones giran alrededor del hecho de que las etiquetas RFID puestas en los productos siguen siendo funcionales incluso después de que se hayan comprado los productos y se hayan llevado a casa, y esto puede utilizarse para vigilancia y otros propósitos cuestionables sin relación alguna con sus funciones de inventario en la cadena de suministro. Aunque la intención es emplear etiquetas RFID de corta distancia, estas pueden ser interrogadas a mayores distancias por cualquier persona con una antena de alta ganancia, permitiendo de forma potencial que el contenido de una casa pueda ser explorado desde cierta distancia. Incluso un escaneado de rango corto es preocupante si todos los artículos detectados aparecen en una base de datos cada vez que una persona pasa un lector, o si se hace de forma malintencionada (por ejemplo, un robo empleando

un escáner de mano portátil para obtener una evaluación instantánea de la cantidad de víctimas potenciales). Con números de serie RFID permanentes, un artículo proporciona información inesperada sobre una persona incluso después de su eliminación; por ejemplo, los artículos que se revenden, o se regalan, pueden permitir trazar la red social de una persona.

Blindaje Faraday como una contramedida al RFID

Se puede utilizar una jaula de Faraday para evitar que las señales de radiofrecuencia se escapen o entren en una zona, actuando como un blindaje RF.

Si se rodeara un dispositivo RFID con un blindaje de Faraday tendría señales entrantes y salientes muy atenuadas, hasta el punto de que no podrían ser utilizables. Un blindaje de Faraday muy sencillo, válido para la mayoría de los propósitos, sería un envoltorio de papel de aluminio. Uno más efectivo sería un rectángulo de cobre alrededor del objeto. Un RFID implantado sería más difícil de neutralizar con dicho blindaje, pero incluso una cubierta simple de papel de aluminio atenuaría la componente de campo eléctrico de las señales. Neutralizar permanentemente el RFID podría necesitar una fuerte corriente eléctrica alterna adyacente al RFID, que sobrecargue la etiqueta y destruya su circuitería interna. En algunos casos, dependiendo de la composición del RFID, un imán fuerte puede servir para destruir mecánicamente la bobina o la conexión del chip por la fuerza mecánica ejercida en la bobina.

Las etiquetas de 125 kHz, 134 kHz (baja frecuencia), y en varios casos 13.56 MHz (alta frecuencia) están unidas por un campo magnético en lugar de un campo eléctrico, es lo que se denomina acoplamiento inductivo. Como la jaula de Faraday blinda solamente la componente eléctrica del campo electromagnético, el blindaje de papel de aluminio es ineficaz. Cualquier blindaje magnético, como por ejemplo una hoja fina de hierro o acero, encapsulando la bobina de la antena de la etiqueta, será eficaz.

2.2 Características y herramientas de LabVIEW.

2.2.1 Descripción de las funciones para el manejo de una base de datos.

Para poder implementar el proyecto será necesaria la utilización de librerías específicas de LabVIEW. Empezaremos describiendo las librerías Database que son las que permiten interactuar a LabVIEW con un programa de desarrollo de bases de datos específico, en este caso utilizaremos el software MySQL como la herramienta que permitirá el desarrollo de las bases de datos requeridas para la implementación del proyecto, como se menciona mas adelante.

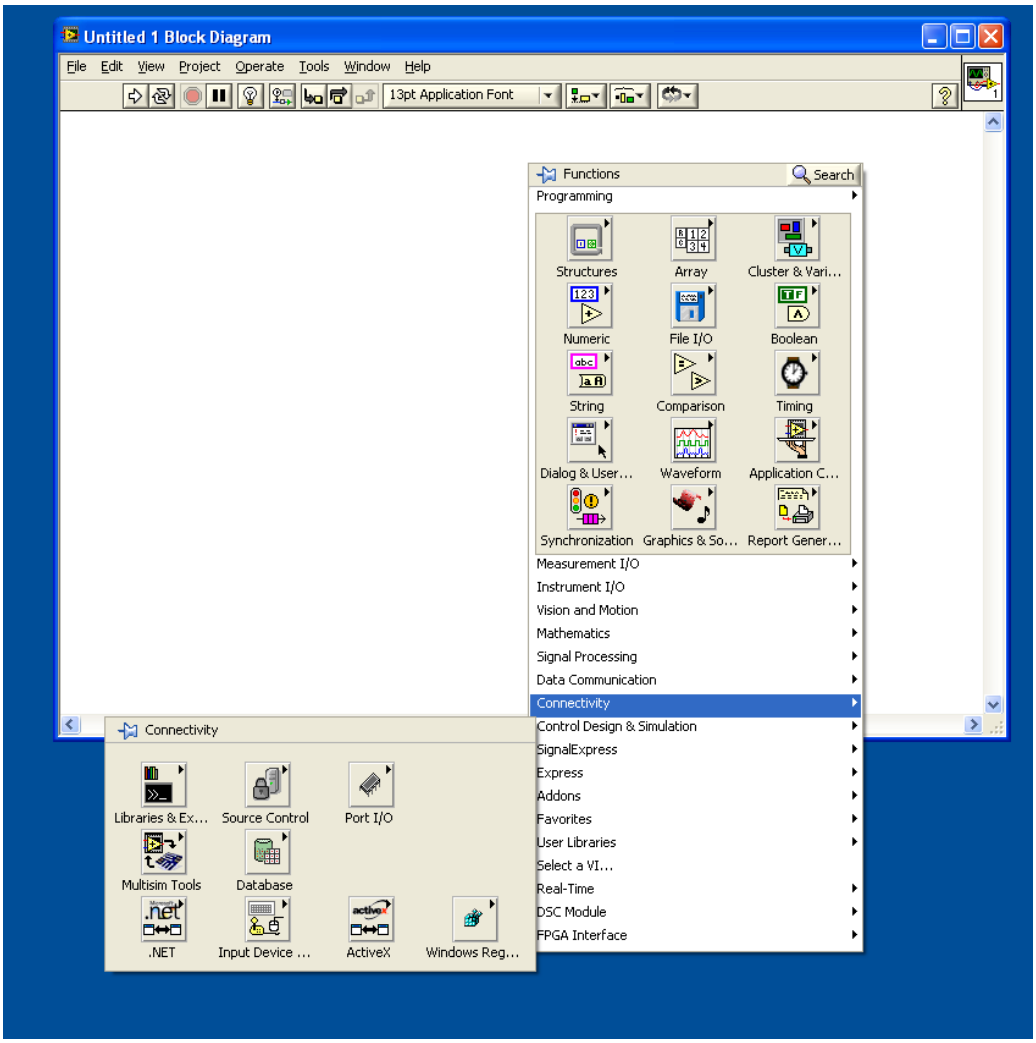


Figura 2.8 Despliegue del menú Connectivity.

Para poder acceder a este conjunto de funciones que permiten la interoperabilidad de LabVIEW con un software de desarrollo de bases de datos, será necesario activar el menú de funciones de LabVIEW, para lo cual se deberá hacer clic derecho en una región cualquiera de la ventana de diagrama de bloques (o Block Diagram) de LabVIEW, luego

se procede a escoger la opción Connectivity del menú de funciones previamente mencionado, al hacerlo se mostrará el menú de conectividad como lo muestra la figura.

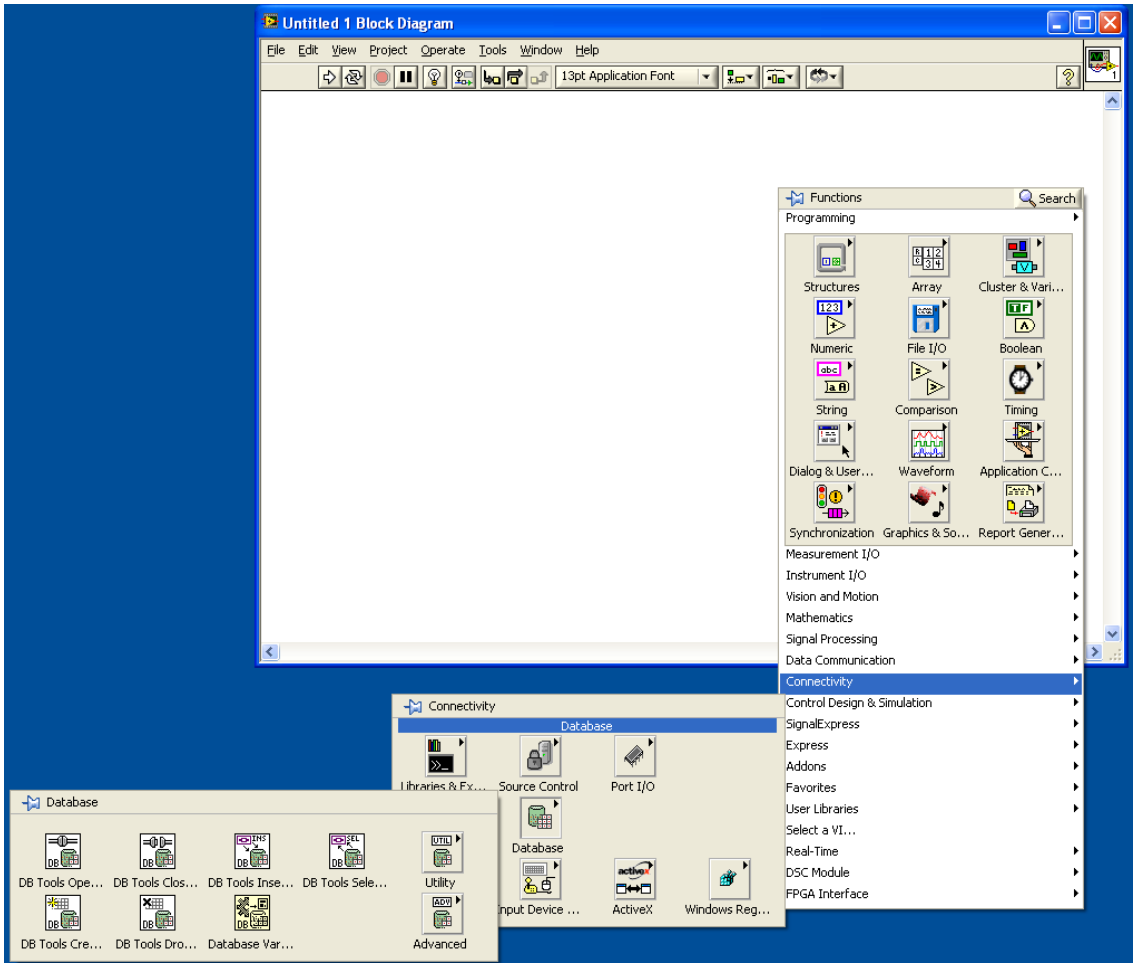


Figura 2.9 Librería Database.

Una vez activado el menú de conectividad (o connectivity), se deberá escoger la opción que nos permita interconectarnos con una aplicación de desarrollo de bases de datos, en este caso escogeremos la opción Database, luego de lo cual se despliega el menú

Database que nos muestra una gamma de funciones relacionadas con la manipulación y operación de bases de datos a través de LabVIEW. Como se muestra en las figuras.

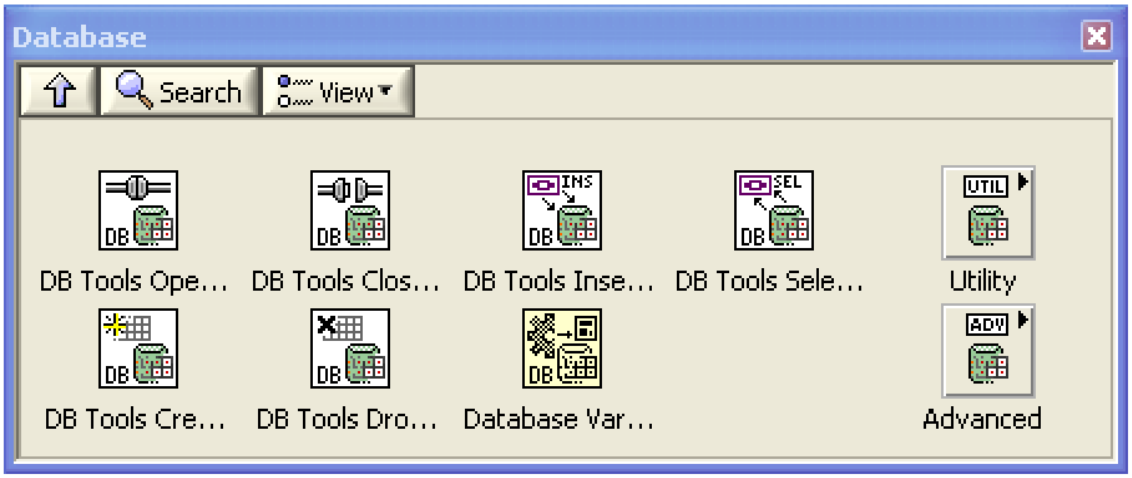


Figura 2.10 Detalle de la librería Database.

La Librería Database de LabVIEW es de suma importancia para una correcta y eficiente implementación del proyecto en cuestión. Debido a su importancia se presenta a continuación una descripción detallada de las funciones utilizadas en el desarrollo del proyecto.

Función DB Tools Open Connection.

La Función DB Tools Open Connection de la Librería Database permite abrir o iniciar una conexión de Base de datos utilizando la información de conexión proporcionada por

uno de sus parámetros, esta función devuelve o retorna un parámetro denominado connection reference. Como se puede apreciar en la figura esta función esta compuesta por siete parámetros, cinco parámetros de entrada y dos parámetros de salida. A continuación analizaremos la función de cada uno de sus parámetros.

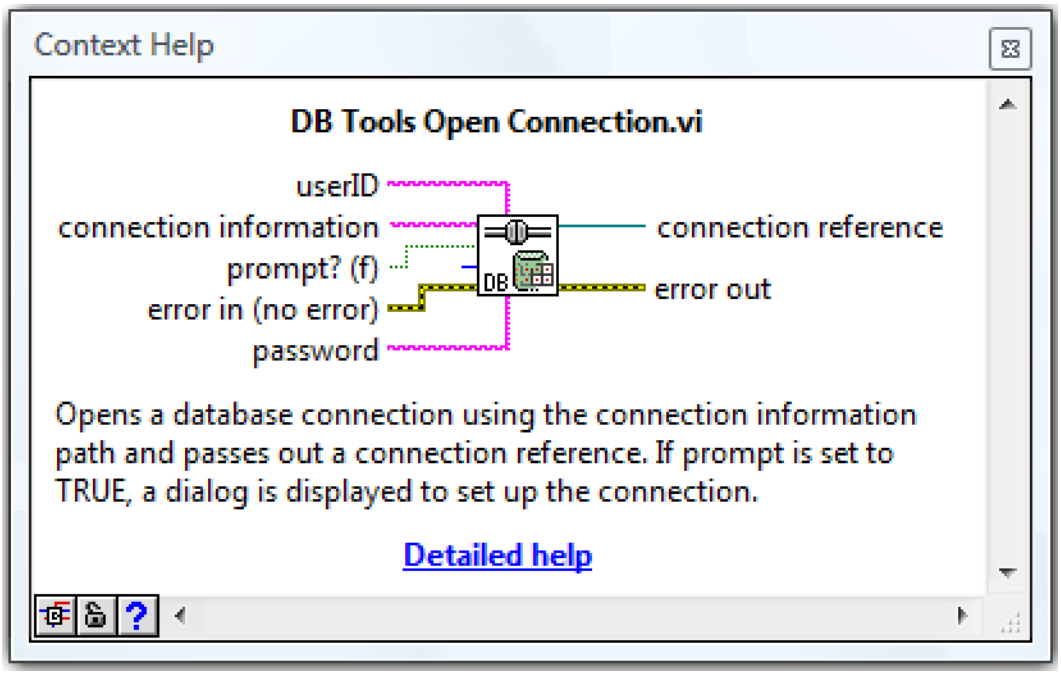


Figura 2.11 Función DB Tools Open Connection.

User ID.- Es la identificación de usuario que permite el ingreso a la base de datos, este parámetro debe ser de tipo string y no es necesario la introducción de este parámetro en la función para que esta funcione apropiadamente.

Connection Information.- Este parámetro acepta valores de tipo string y proporciona la información que se va a utilizar para realizar la conexión a la base de datos. Para utilizar un sistema ODBC, simplemente será necesario proporcionar el nombre del Data Source como esta configurado en el Windows ODBC Administrador.

Este parámetro también puede aceptar datos de tipo path que permiten determinar la ruta del archivo que almacena la información de la conexión. Un archivo de tipo Microsoft Data Link deberá tener una extensión .udl, mientras que un archivo DSN deberá tener una extensión .dsn.

Prompt?.- Este parámetro debe ser de tipo booleano. Si está establecida como TRUE, será posible determinar los parámetros de conexión a través de un prompt y la cadena de conexión será ignorada. El parámetro Prompt? es una alternativa a la utilización de una cadena de conexión, que se define en el parámetro connection information.

Connection timeout.- Este parámetro debe ser definido como tipo entero y determina el tiempo (en segundos) de espera para establecer una conexión con una base de datos especifica antes de interrumpirla y retornar un error. El valor establecido por defecto es de 15 segundos. Establecer este valor en 0 le indica a esta función que espere indefinidamente una conexión exitosa.

Error in.- Describe los errores ocasionados antes de que la función se haya activado. Este parámetro debe ser declarado como tipo cluster y el valor asignado por defecto es de “no error”. Si un error ocurre, este instrumento virtual retorna el valor de “error in” o

error presente en el parámetro error out. Esta función trabaja normalmente únicamente si no existen errores de llegada. En caso de que exista algún error, el instrumento virtual pasa el error al parámetro error out. El cluster denominado error in contiene los siguientes parámetros:

Status: El parámetro status es de tipo booleano, y es verdadero (TRUE) si un error ocurre. El valor establecido por defecto es FALSE.

Code: Es un parámetro de tipo integer y representa el error code del cluster. El valor designado por defecto es 0.

Source: Es un parámetro de tipo string y representa el nombre del instrumento virtual o función que ha producido el error. El valor por defecto es un empty string.

Password.- Es un parámetro de tipo string, y se utiliza para definir una contraseña que permita acceder a una base de datos específica. Este parámetro no es indispensable para el correcto funcionamiento del instrumento virtual en cuestión.

Connection Reference.- Es una referencia hacia un ADO connection object.

Error out.- Es un parámetro de tipo cluster que contiene información sobre los errores ocurridos. Si el parámetro “error in” indica la presencia de un error, el parámetro “error out” contendrá la misma información de error. De igual manera este parámetro describe

los errores producidos por este instrumento virtual. El cluster error out contiene los siguientes parámetros:

Status: Es de tipo booleano y adquiere el valor TRUE si un error ha ocurrido.

Code: Es de tipo integer y representa el error code de esta función.

Source: Es un parámetro de tipo string y representa el nombre del instrumento virtual o función que ha producido el error. El valor por defecto es un empty string.

Función DB Tools Close Connection.

La función DB Tools Close Connection es una función de la librería database que permite cerrar una conexión con una base de datos específica destruyendo la conexión de referencia asociada a este instrumento virtual. El DB Tools Close Connection maneja tres parámetros, dos parámetros de entrada y un parámetro de salida. A continuación presentamos la descripción detallada de cada uno de estos parámetros.

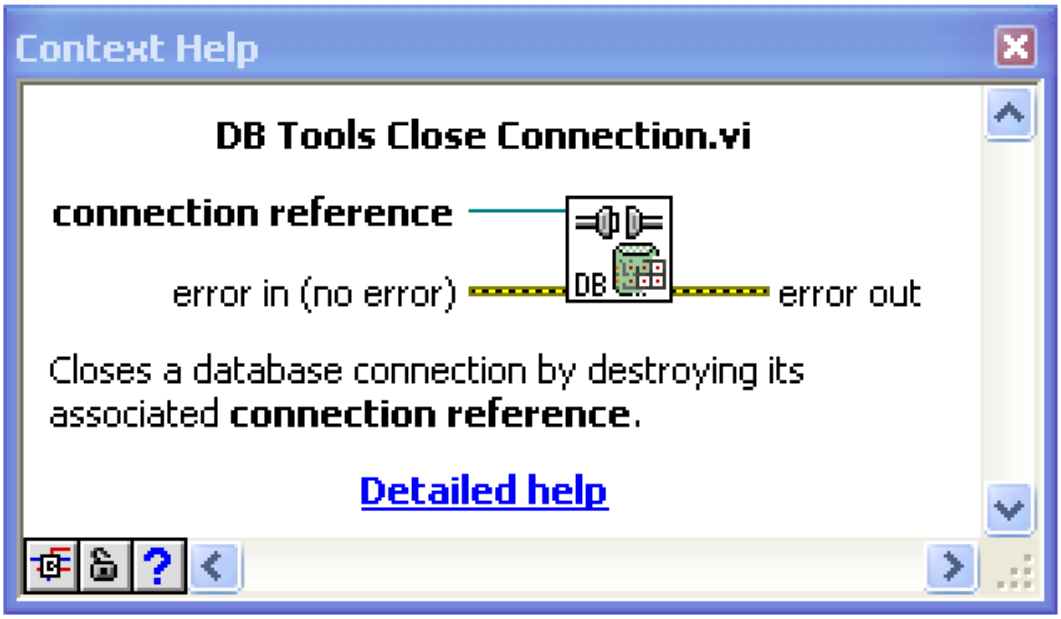


Figura 2.12 Función DB Tools Close Connection.

Connection Reference.- Es una referencia a un objeto de conexión tipo ADO.

Error in.- Describe los errores ocasionados antes de que la función se haya activado. Este parámetro debe ser declarado como tipo cluster y el valor asignado por defecto es de "no error". Si un error ocurre, este instrumento virtual retorna el valor de "error in" o error presente en el parámetro error out. Esta función trabaja normalmente únicamente si no existen errores de llegada. En caso de que exista algún error, el instrumento virtual pasa el error al parámetro error out. El cluster denominado error in contiene los siguientes parámetros:

Status: El parámetro status es de tipo booleano, y es verdadero (TRUE) si un error ocurre. El valor establecido por defecto es FALSE.

Code: Es un parámetro de tipo integer y representa el error code del cluster. El valor designado por defecto es 0.

Source: Es un parámetro de tipo string y representa el nombre del instrumento virtual o función que ha producido el error. El valor por defecto es un empty string.

Error out.- Es un parámetro de tipo cluster que contiene información sobre los errores ocurridos. Si el parámetro "error in" indica la presencia de un error, el parámetro "error out" contendrá la misma información de error. De igual manera este parámetro describe los errores producidos por este instrumento virtual. El cluster error out contiene los siguientes parámetros:

Status: Es de tipo booleano y adquiere el valor TRUE si un error ha ocurrido.

Code: Es de tipo integer y representa el error code de esta función.

Source: Es un parámetro de tipo string y representa el nombre del instrumento virtual o función que ha producido el error. El valor por defecto es un empty string.

Función DB Tools Insert Data.

Esta función permite la inserción de un nuevo renglón en la tabla de la base de datos previamente identificada por el parámetro de entrada "connection reference" de este instrumento virtual. Como se puede apreciar en la figura esta función esta compuesta por

nueve parámetros, siete parámetros de entrada y dos parámetros de salida. Estos parámetros se describen en detalle a continuación.

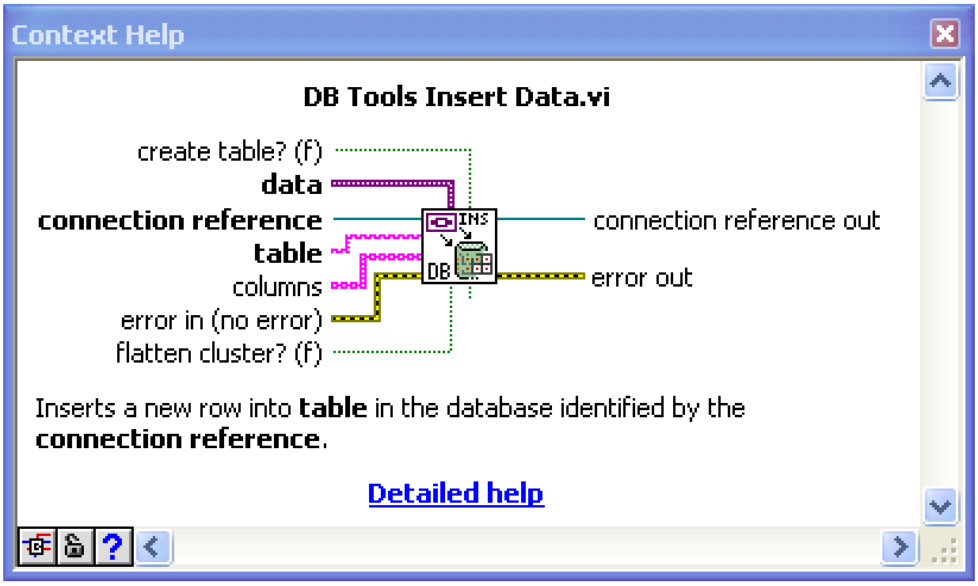


Figura 2.13 Función DB Tools Insert Data.

Create table?(f).- Este parámetro es de tipo booleano, y le indica al instrumento virtual que cree una tabla en caso de no existir una.

Data.- Le indica a la función los datos que se van a introducir en la base de datos. Si el dato es un cluster, entonces cada elemento del cluster será insertado en la tabla, de tal manera que corresponda con cada elemento del parámetro columns, de tipo array. Si el arreglo columns esta vacío entonces el primer elemento del cluster se insertara en la primera columna de la tabla.

Si el dato no es un cluster, entonces se insertara en la columna especificada por el parámetro columns.

Connection reference.- Es una referencia a un ADO connection object.

Table.- Es el nombre de la tabla en la base de datos en la cual se insertara el dato. Si el parámetro create table? tiene el valor de TRUE el instrumento virtual creara una tabla en caso de que esta no exista.

Columns.- Es un parámetro de tipo string y representa a la columna de la tabla en la cual se va a insertar el dato, si se introduce un arreglo vacío en este parámetro, este instrumento virtual asumirá que todas las columnas de la tabla están siendo utilizadas.

Error in.- Describe los errores ocasionados antes de que la función se haya activado. Este parámetro debe ser declarado como tipo cluster y el valor asignado por defecto es de "no error". Si un error ocurre, este instrumento virtual retorna el valor de "error in" o error presente en el parámetro error out. Esta función trabaja normalmente únicamente si no existen errores de llegada. En caso de que exista algún error, el instrumento virtual pasa el error al parámetro error out. El cluster denominado error in contiene los siguientes parámetros:

Status: El parámetro status es de tipo booleano, y es verdadero (TRUE) si un error ocurre. El valor establecido por defecto es FALSE.

Code: Es un parámetro de tipo integer y representa el error code del cluster. El valor designado por defecto es 0.

Source: Es un parámetro de tipo string y representa el nombre del instrumento virtual o función que ha producido el error. El valor por defecto es un empty string.

Flatten cluster? (f).- Este parámetro es de tipo boolean. Si adquiere el valor TRUE y además el parámetro data de la función es de tipo cluster, entonces el instrumento virtual insertara este cluster como un valor binario en lugar de tratar a cada uno de los elementos del cluster como columnas individuales, esto implica que la columna que este cluster esta insertando es una columna binaria.

Use file.- Es un parámetro de tipo booleano, y se utiliza cuando el parámetro create table es TRUE y le indica a la función que utilice un user-supplied file para determinar el tipo de base de dato.

Connection reference out.- Es una referencia a un ADO connection object.

2.2.2 Descripción de las funciones para una conexión con Ethernet.

Otra Librería de suma importancia para la implementación del proyecto en cuestión, es la librería relacionada con el manejo y conexión de ethernet a través de LabVIEW. Para acceder a esta librería será necesario ingresar al menú de funciones presionando el botón derecho del mouse sobre el área de trabajo de la ventana de Block Diagram de LabVIEW.

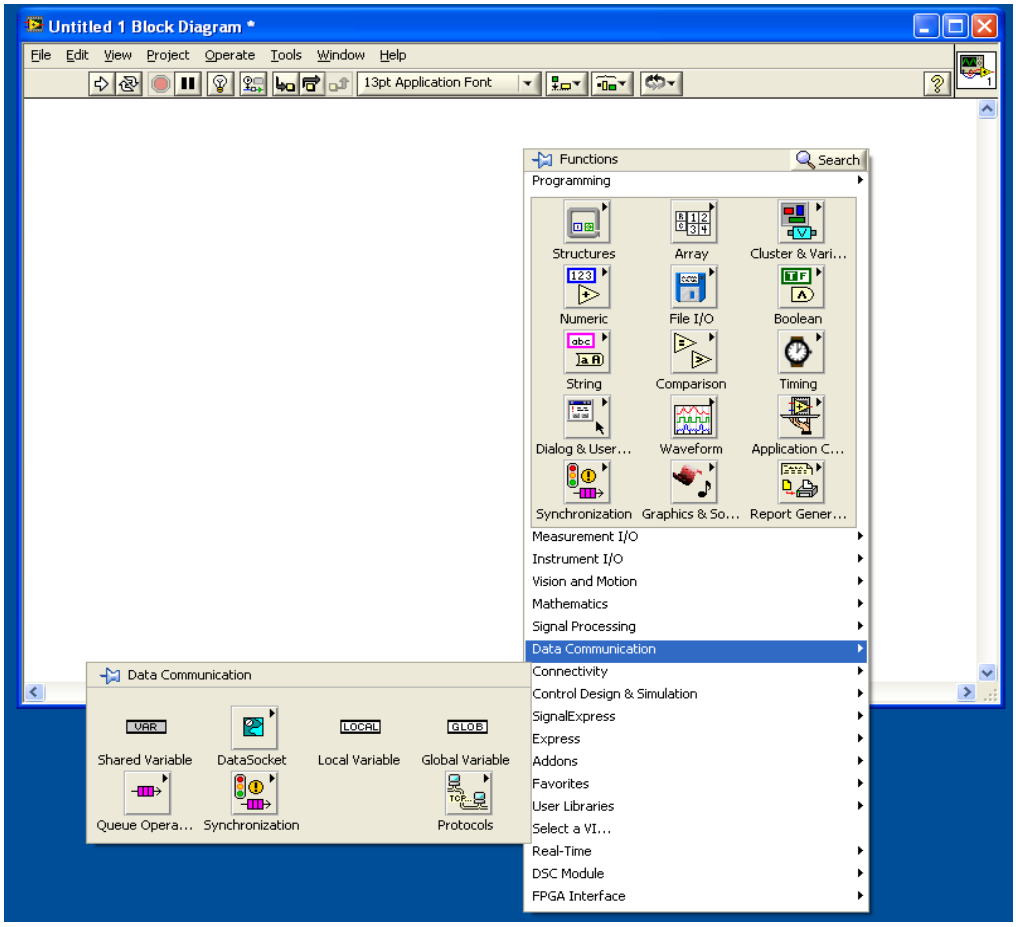


Figura 2.14 Despliegue del Menú Data Communication.

Una vez desplegado el menú de funciones escogemos la opción Data Communication, como se muestra en la figura 2.14.

Una vez que nos encontremos en el menú Data Communication seleccionamos la opción Protocols que nos permitirá acceder a las funciones de LabVIEW que permiten a este software utilizar distintos protocolos o estándares de comunicación. Como se muestra en la figura 2.15.

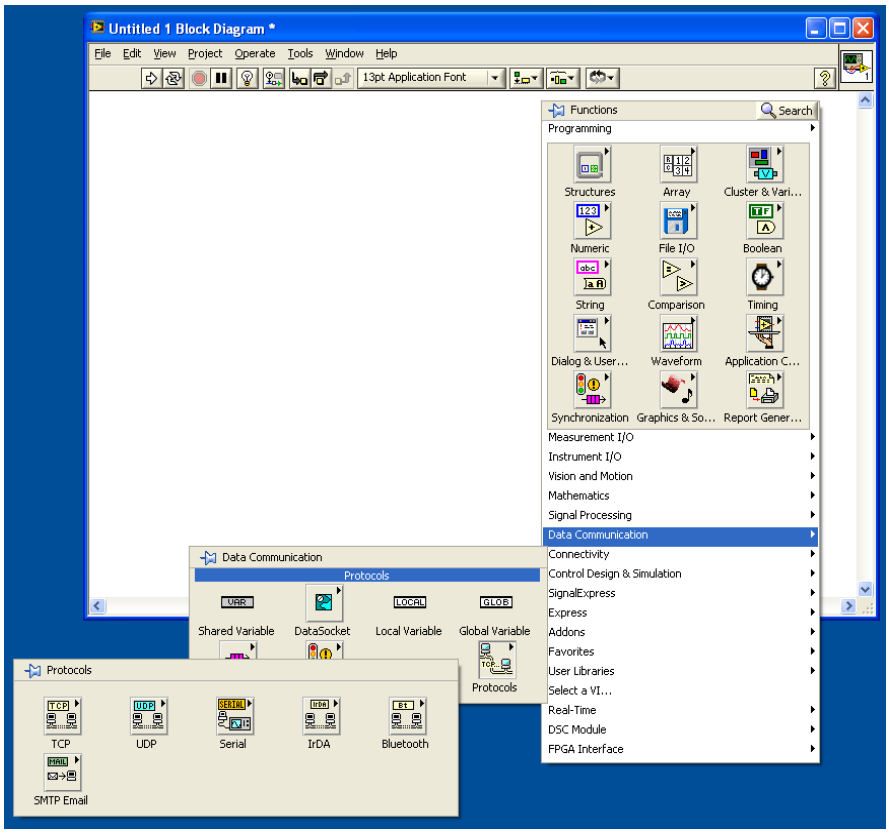


Figura 2.15 Menú Protocols.

Como podemos apreciar en la figura que presentamos en la parte inferior de esta página, el menú Protocols despliega seis posibles estándares de comunicación con los cuales LabVIEW puede trabajar, en este caso específico escogeremos la opción que despliega el menú del protocolo UDP, que es el protocolo que vamos a utilizar para poder realizar una comunicación ethernet, absolutamente necesaria para la implementación de nuestro proyecto, como se describirá en capítulos posteriores.

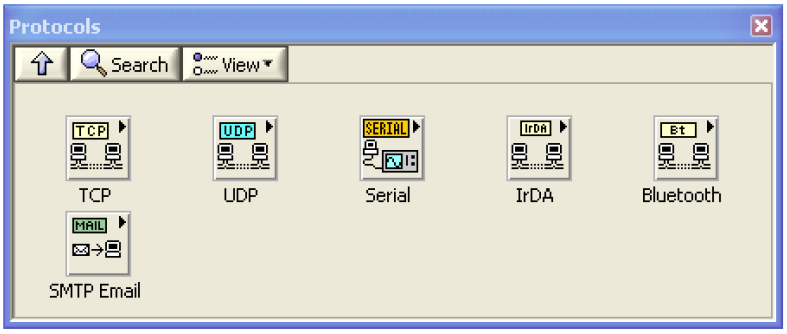


Figura 2.16 Detalle del Menú Protocols.

Al seleccionar esta opción se despliega la librería UDP de LabVIEW que muestra todos los instrumentos virtuales o funciones que permiten la interoperabilidad de LabVIEW con otros sistemas utilizando el Protocolo de Datagrama de Usuario (UDP).

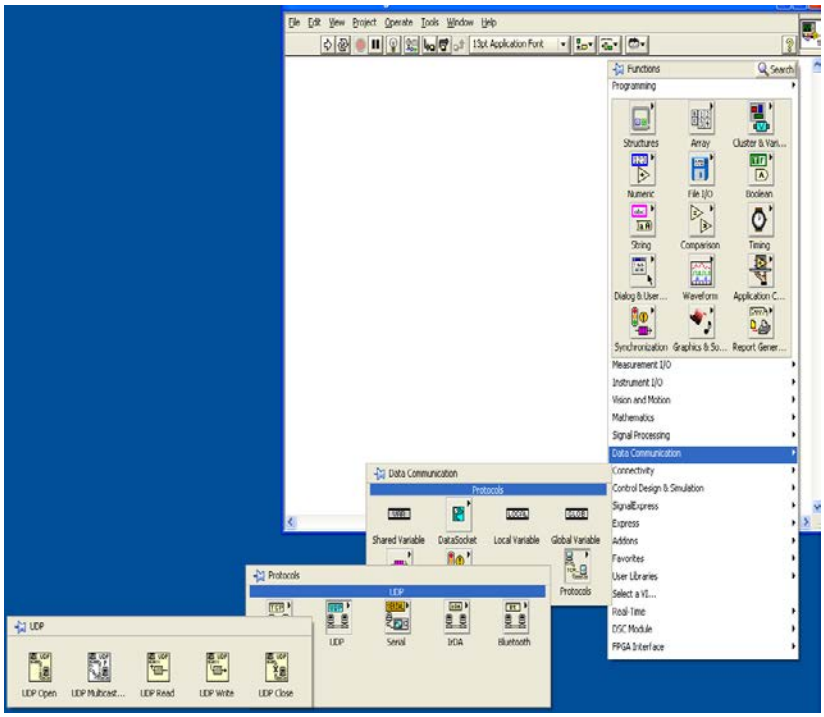


Figura 2.17 Despliegue de la librería UDP

En la figura presentada en la parte inferior de esta página podemos observar todas las funciones que pertenecen a la librería UDP. A continuación se describen en forma detallada las funciones que son utilizadas en la implementación del proyecto.

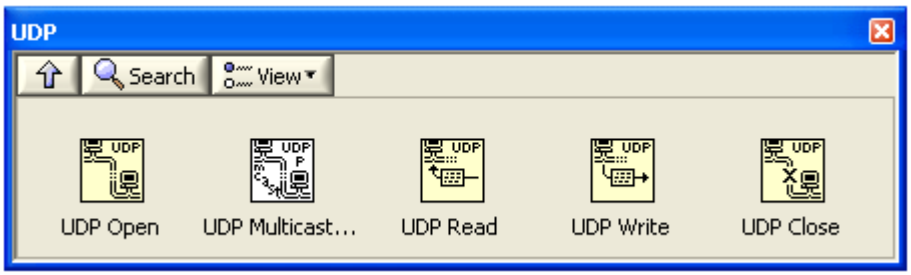


Figura 2.18 Detalle de la Librería UDP.

Función UDP Open.

Esta función permite iniciar una sesión UDP en LabVIEW en el puerto especificado en el parámetro port. Como se puede apreciar en la figura presentada en la parte inferior, este instrumento virtual trabaja con seis parámetros, de los cuales, tres son parámetros de entrada y tres son parámetros de salida. A continuación se presenta una descripción detallada de cada uno de los parámetros que conforman este instrumento virtual.

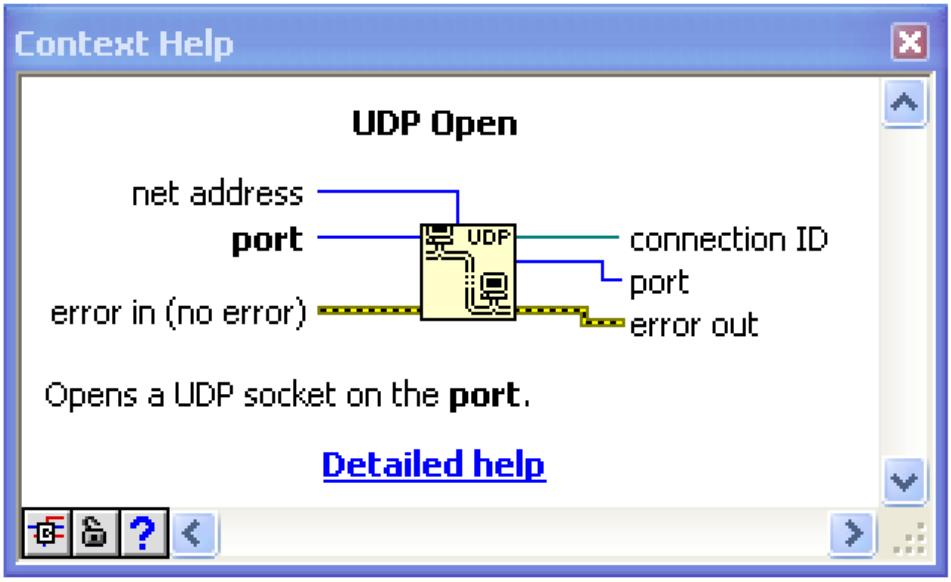


Figura 2.19 Función UDP Open.

Net address.- Es un parámetro de tipo Unsigned Long (32-bits). Este parámetro le indica a la función en cual dirección de red debe "escuchar". Especificar una dirección de red es útil si se esta trabajando con mas de una tarjeta de red y se desea "escuchar" únicamente en la tarjeta con la dirección de red especificada en este parámetro. Si no se especifica una dirección de red, LabVIEW escucha en todas las direcciones de red.

Port.- Es el puerto local con el cual se va a crear una sesión UDP. Este parámetro es de tipo Unsigned Word (16-bits).

Error in.- Describe los errores ocasionados antes de que la función se haya activado. Este parámetro debe ser declarado como tipo cluster y el valor asignado por defecto es de "no error". Si un error ocurre, este instrumento virtual retorna el valor de "error in" o

error presente en el parámetro error out. Esta función trabaja normalmente únicamente si no existen errores de llegada. En caso de que exista algún error, el instrumento virtual pasa el error al parámetro error out. El cluster denominado error in contiene los siguientes parámetros:

Status: El parámetro status es de tipo booleano, y es verdadero (TRUE) si un error ocurre. El valor establecido por defecto es FALSE.

Code: Es un parámetro de tipo integer y representa el error code del cluster. El valor designado por defecto es 0.

Source: Es un parámetro de tipo string y representa el nombre del instrumento virtual o función que ha producido el error. El valor por defecto es un empty string.

Connection ID.- Es una referencia a una conexión de red que únicamente identifica el UDP socket. Este valor se utiliza para hacer referencia a este socket en llamadas a funciones posteriores.

Port.- Es un parámetro de tipo Unsigned Word (16-bits) y retorna el número de puerto que la función utilizó. Si la entrada port de la función es diferente de cero, el parámetro port a la salida de la función tendrá el mismo valor que el de la entrada. Si se introduce en el parámetro port a la entrada el valor 0, entonces la función escogerá un puerto UDP asignado de forma dinámica y que el sistema operativo lo haya determinado como

disponible y utilizable. El IANA (Internet Assigned Numbers Authority), define los números de puertos válidos en el rango comprendido entre 49152 y 65535. No todos los sistemas operativos siguen el estándar IANA, por ejemplo los números de puerto asignados de forma dinámica por Windows están entre 1024 y 5000.

Error out.- Es un parámetro de tipo cluster que contiene información sobre los errores ocurridos. Si el parámetro "error in" indica la presencia de un error, el parámetro "error out" contendrá la misma información de error. De igual manera este parámetro describe los errores producidos por este instrumento virtual. El cluster error out contiene los siguientes parámetros:

Status: Es de tipo booleano y adquiere el valor TRUE si un error ha ocurrido.

Code: Es de tipo integer y representa el error code de esta función.

Source: Es un parámetro de tipo string y representa el nombre del instrumento virtual o función que ha producido el error. El valor por defecto es un empty string.

Función UDP Read .

Esta función lee un datagrama desde un socket UDP, y retorna los resultados en el parámetro data out. Este instrumento virtual retorna datos si recibe cualquier byte y permanece en modo de espera únicamente si no está recibiendo ningún byte. Esta función

está compuesta por nueve parámetros de los cuales seis son de entrada y tres de salida. Estos parámetros se describen en detalle a continuación.

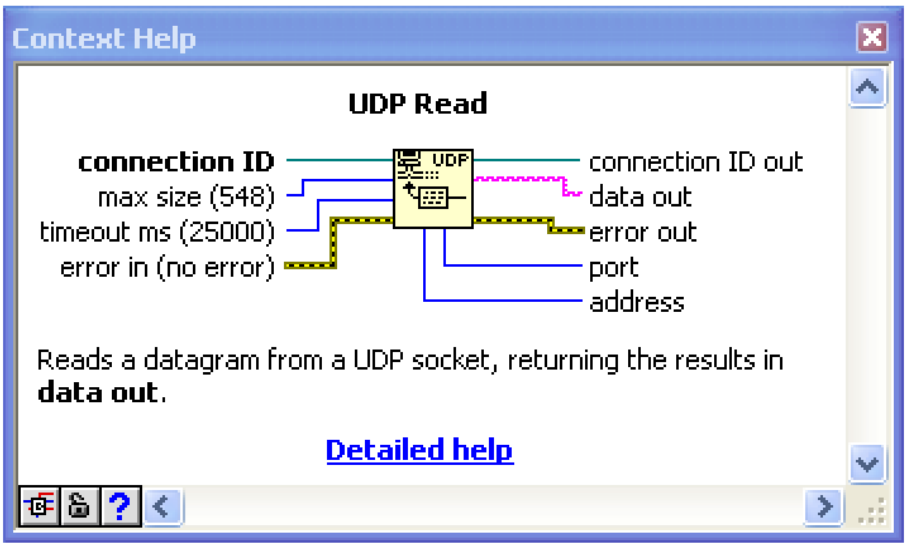


Figura 2.20 Función UDP Read.

Connection ID.- Es un número de referencia de la conexión a la red, que únicamente identifica al socket UDP.

Max Size.- Es un parámetro de tipo integer, y representa el número máximo de bytes que esta función puede leer. El máximo valor establecido por defecto es 548. Es posible que Windows retorne un error, si este valor se cambia puesto que la función no puede leer un número inferior de bytes que los que se encuentran en un paquete.

Timeout ms.- Es un parámetro de tipo integer. Si ningún byte ha sido recibido en el tiempo especificado por este parámetro, la función retorna un error. El valor establecido

por defecto de este parámetro es de 25000 ms. Si se introduce un valor de -1 en este parámetro, la función lo interpretara como un tiempo de espera indefinido.

Error in.- Describe los errores ocasionados antes de que la función se haya activado. Este parámetro debe ser declarado como tipo cluster y el valor asignado por defecto es de "no error". Si un error ocurre, este instrumento virtual retorna el valor de "error in" o error presente en el parámetro error out. Esta función trabaja normalmente únicamente si no existen errores de llegada. En caso de que exista algún error, el instrumento virtual pasa el error al parámetro error out. El cluster denominado error in contiene los siguientes parámetros:

Status: El parámetro status es de tipo booleano, y es verdadero (TRUE) si un error ocurre. El valor establecido por defecto es FALSE.

Code: Es un parámetro de tipo integer y representa el error code del cluster. El valor designado por defecto es 0.

Source: Es un parámetro de tipo string y representa el nombre del instrumento virtual o función que ha producido el error. El valor por defecto es un empty string.

Connection ID out.- Retorna el mismo valor recibido en el parámetro connection ID.

Data Read.- Es un parámetro de tipo string que contiene los datos leídos desde el datagrama UDP.

Error out.- Es un parámetro de tipo cluster que contiene información sobre los errores ocurridos. Si el parámetro “error in” indica la presencia de un error, el parámetro “error out” contendrá la misma información de error. De igual manera este parámetro describe los errores producidos por este instrumento virtual. El cluster error out contiene los siguientes parámetros:

Status: Es de tipo booleano y adquiere el valor TRUE si un error ha ocurrido.

Code: Es de tipo integer y representa el error code de esta función.

Source: Es un parámetro de tipo string y representa el nombre del instrumento virtual o función que ha producido el error. El valor por defecto es un empty string.

Port.- Es un parámetro de tipo Unsigned y representa al puerto del UDP socket que envía el datagrama.

Address.- Es la dirección de la computadora desde la cual se origina el datagrama. Este parámetro debe ser declarado como tipo integer.

Función UDP Close.

La función UDP Close permite cerrar una sesión UDP previamente inicializada. Esta función está compuesta por cuatro parámetros, dos de los cuales son parámetros de entrada y los dos restantes son los parámetros de salida del instrumento virtual.

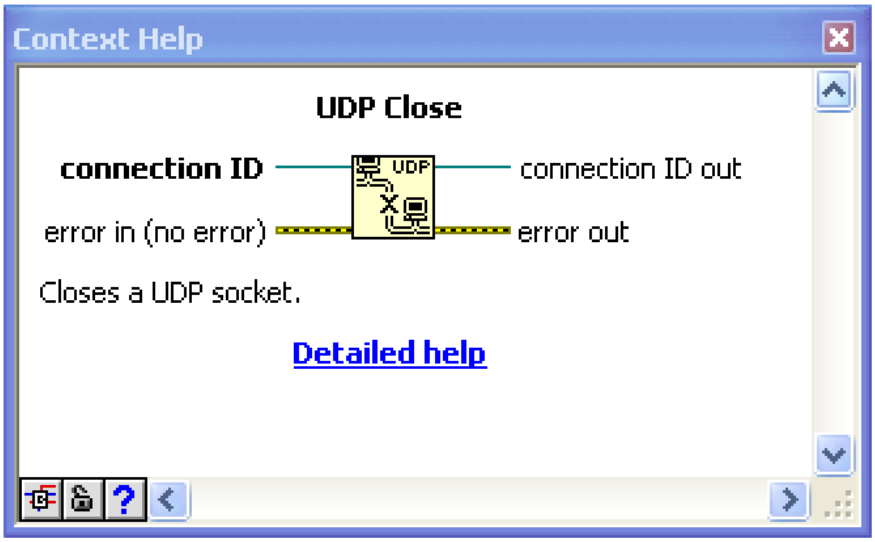


Figura 2.21 Función UDP Close.

Connection ID.- Es un número de referencia de la conexión a la red, que únicamente identifica a la sesión UDP que se desea cerrar.

Error in.- Describe los errores ocasionados antes de que la función se haya activado. Este parámetro debe ser declarado como tipo cluster y el valor asignado por defecto es de "no error". Si un error ocurre, este instrumento virtual retorna el valor de "error in" o error presente en el parámetro error out. Esta función trabaja normalmente únicamente si no existen errores de llegada. En caso de que exista algún error, el instrumento virtual pasa el error al parámetro error out. El cluster denominado error in contiene los siguientes parámetros:

Status: El parámetro status es de tipo booleano, y es verdadero (TRUE) si un error ocurre. El valor establecido por defecto es FALSE.

Code: Es un parámetro de tipo integer y representa el error code del cluster. El valor designado por defecto es 0.

Source: Es un parámetro de tipo string y representa el nombre del instrumento virtual o función que ha producido el error. El valor por defecto es un empty string.

Connection ID out.- Retorna el mismo valor recibido en el parámetro connection ID. Este parámetro no debe conectarse a otra función UDP.

Error out.- Es un parámetro de tipo cluster que contiene información sobre los errores ocurridos. Si el parámetro “error in” indica la presencia de un error, el parámetro “error out” contendrá la misma información de error.

De igual manera este parámetro describe los errores producidos por este instrumento virtual. El cluster error out contiene los siguientes parámetros:

Status: Es de tipo booleano y adquiere el valor TRUE si un error ha ocurrido.

Code: Es de tipo integer y representa el error code de esta función.

Source: Es un parámetro de tipo string y representa el nombre del instrumento virtual o función que ha producido el error. El valor por defecto es un empty string.

Port.- Es un parámetro de tipo Unsigned y representa al puerto del UDP socket que envía el datagrama.

Address.- Es la dirección de la computadora desde la cual se origina el datagrama. Este parámetro debe ser declarado como tipo integer.

2.3 Descripción de los módulos a implementar.

2.3.1 Módulo RFID Reader #28140.

El módulo RFID #28140 es un dispositivo de identificación por radiofrecuencia desarrollado por Parallax que permite identificar tags RFID de tipo pasivos. Este módulo trabaja exclusivamente con la familia EM4100 de tags pasivos de solo lectura desarrollados por EM Microelectronics-Marin S.A. Cada tag transpondedor contiene un número de identificación único que puede ser leído por el módulo RFID y transmitido al host a través de una interfaz serial simple. Este módulo puede integrarse a cualquier diseño utilizando sus cuatro conectores (VCC, /ENABLE, SOUT, GND). La tabla presentada a continuación indica la función de cada uno de los pines del módulo RFID 28140 reader.

Pin	Nombre del Pin	Tipo	Función
1	VCC	Alimentación	Alimentación del sistema. Es una entrada de +5V DC.
2	/ENABLE	Entrada	Es el habilitador del módulo. Se activa con una señal digital en bajo y al hacerlo se habilita el RFID reader y la antena se activa.
3	SOUT	Salida	Salida serial, 2400bps, 8 bits de datos, sin paridad, 1 stop bit.
4	GND	Tierra	Conexión a tierra.

Tabla I: Función de las terminales del Módulo RFID Reader #28140.[9]

[9] Fuente: Datasheet del Parallax #28140 RFID Reader.

Este módulo se controla por medio de un pulso aplicado en el pin /ENABLE. Si a esta entrada ingresa una señal cuyo nivel de voltaje es bajo, entonces el módulo se activará y la antena empezara a buscar la presencia de tags.

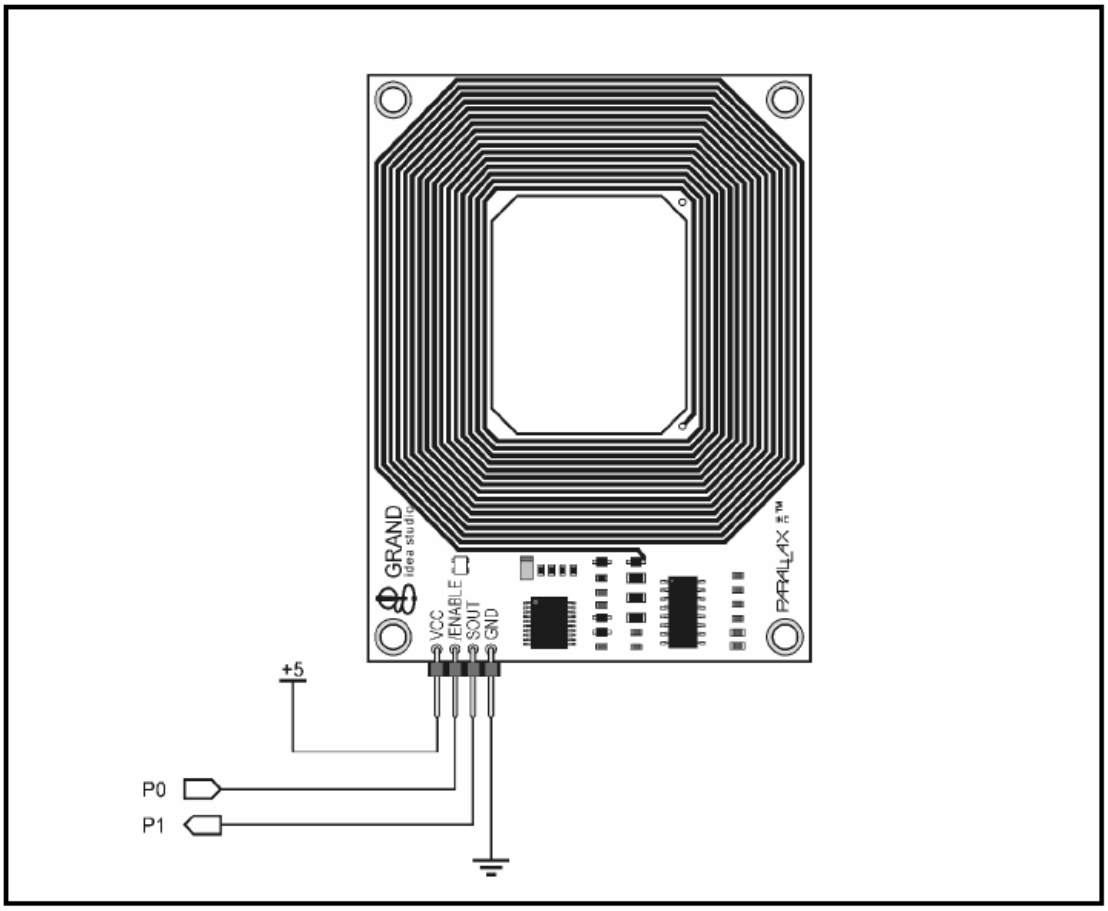


Figura 2.22 Lector RFID # 28140.[10]

[10] Fuente: Datasheet del Parallax #28140 RFID Reader.

Puesto que los tags utilizados son pasivos, la circuitería interna del tag se activa con la energía irradiada por la antena del reader. Una vez que el tag ha recibido la señal proveniente del reader, puede emitir una señal modulada que es recibida por el reader y convertida en una señal digital que luego es enviada a través de una interfaz serial por el pin SOUT. El módulo RFID reader está diseñado para operar específicamente a bajas frecuencias, en este caso a 125KHz. El tag RFID debe colocarse paralelo al módulo RFID reader. Si el tag se coloca perpendicular al módulo, es probable que no sea detectada la señal del tag o que la calidad de la señal recibida en el módulo sea de muy baja calidad. El módulo RFID reader únicamente puede detectar la presencia de un solo tag a la vez, colocar múltiples tags al mismo tiempo frente al módulo podría ocasionar una colisión de datos y el módulo no podrá distinguir el ID de ningún tag. Si se coloca una etiqueta RFID en el área de cobertura del módulo RFID reader, este tag empezara a transmitir su ID como una cadena de tipo ASCII de 12 bytes de longitud.

La trama de datos transmitida por el tag RFID tiene el siguiente formato:

Start Byte (0x0A)	ID único 1º Digito	ID único 2º Digito	ID único 3º Digito	ID único 4º Digito	ID único 5º Digito	ID único 6º Digito	ID único 7º Digito	ID único 8º Digito	ID único 9º Digito	ID único 10º Digito	Stop Byte (0x0D)

MSB **LSB**

El start byte y el stop byte se utilizan para identificar, de manera sencilla, si ha sido recibida exitosamente la trama en el RFID reader. Los 10 bytes del centro de la trama corresponden al número ID del tag.

Todas las comunicaciones deben usar los siguientes parámetros: 8 bits de datos, no parity, 1 stop bit. El baud rate está configurado para funcionar a 2400bps, y no puede ser cambiado. El módulo RFID reader incluye dos RFID tags de tipo pasivo, como se puede apreciar en la figura presentada a continuación.

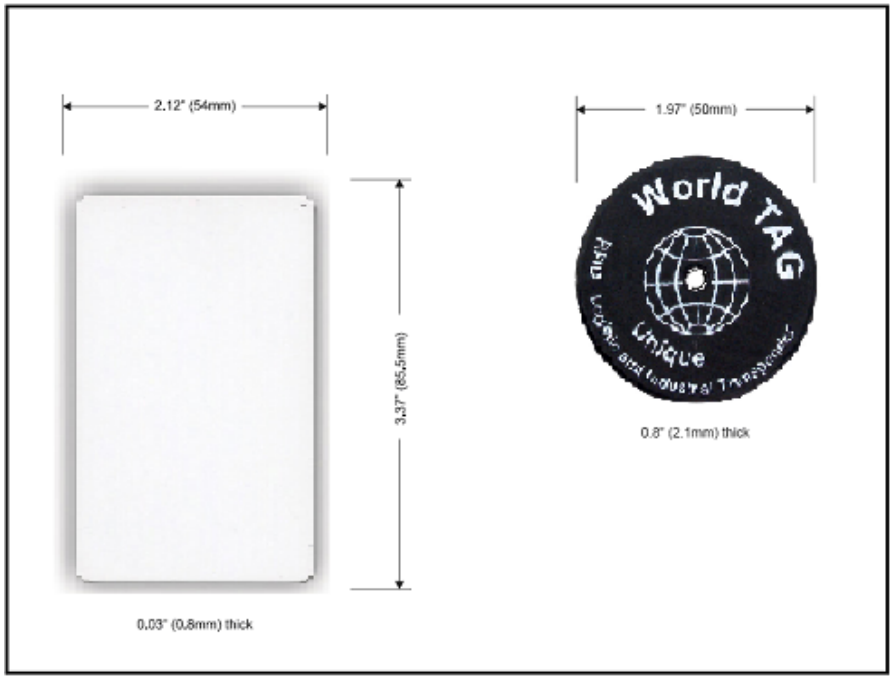


Figura 2.23 Etiquetas del Módulo RFID #28140.[11]

Las dimensiones de los tags se presentan en la figura, y tienen un área de cobertura aproximada de entre 1.75 y 3 pulgadas lo que equivale a un rango aproximado de entre 4.5 y 7.5 cm.

[11] Fuente: Datasheet del Parallax #28140 RFID Reader.

2.3.2 Módulo ET-MINI ENC28J60.

El módulo ET-MINI ENC28J60 está diseñado específicamente para permitir la comunicación entre un microcontrolador y una red Ethernet. Este módulo tiene la capacidad de trabajar con el protocolo TCP/IP debido a que en su circuitería interna tiene integrado el chip ENC28J60. Este circuito integrado es un controlador Ethernet que soporta el estándar de comunicación IEEE 802.3 y trabaja utilizando un bus SPI con una velocidad máxima de 10 Mbps.

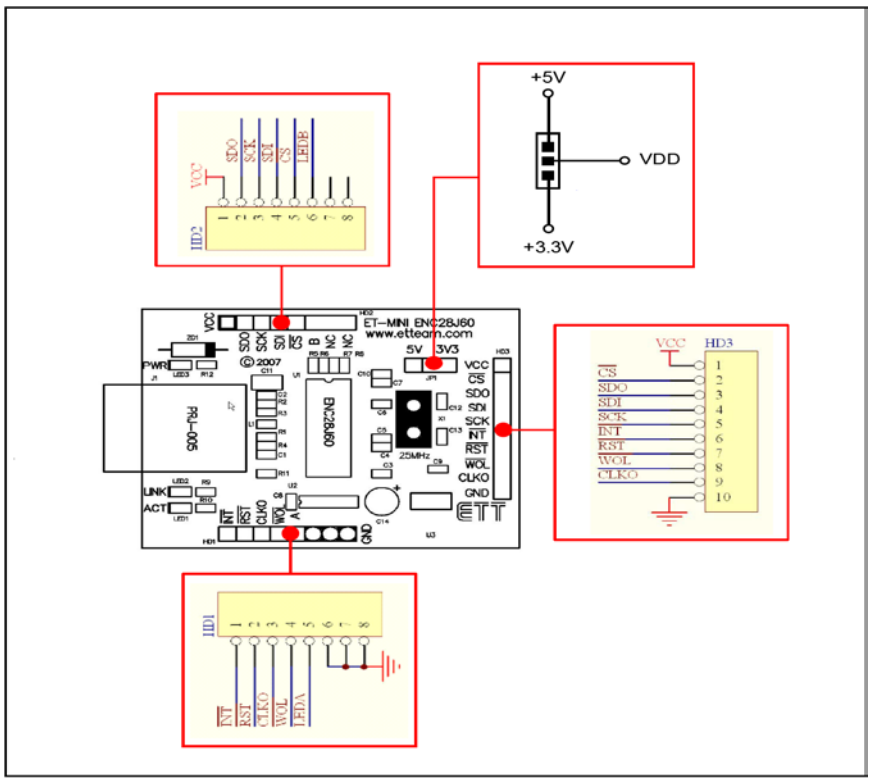


Figura 2.24 Módulo ET-MINI ENC28J60.[12]

[12] Fuente: Datasheet del Módulo ET-MINI ENC28J60.

Como se puede apreciar en la figura este módulo utiliza un conector de tipo PRJ-005, que permite la conexión a una red Ethernet utilizando un Jack RJ-45. El cerebro de este módulo es el chip ENC28J60, que es el que permite la comunicación Ethernet utilizando el estándar SPI como interfaz de comunicación serial entre el PIC 18F4520 y el chip ENC28J60. Además, se puede observar la presencia de tres conectores denominados HD1, HD2 y HD3. Los conectores HD1 y HD2 le permiten al módulo Ethernet comunicarse con el sistema, mientras que el conector HD3 le permite conectarse a otros microcontroladores. La función que cumplen cada uno de estos pines se detalla en la tabla presentada a continuación.

PIN	TIPO	FUNCION
CS	INPUT	Habilita o deshabilita el bus SPI del ENC28J60. CS= 0 habilita el bus SPI. CS= 1 deshabilita el bus SPI.
SDO	OUTPUT	Señal serial de salida de datos.
SDI	INPUT	Señal serial de entrada de datos.
SCK	INPUT	Señal de reloj.
INT	OUTPUT	Señal de interrupción.
RST	INPUT	Señal de reset.
WOL	OUTPUT	Señal Wake-up on LAN interrupt.
CLKO	OUTPUT	Señal de reloj programable.

Tabla II. Función de las terminales del Módulo ET-MINI ENC28J60.[13]

[13] Fuente: Datasheet del Módulo ET-MINI ENC28J60.

Otro elemento de suma importancia, cuyo funcionamiento se debe detallar, es el denominado puente o jumper (JP1). Este elemento funciona de la siguiente manera:

Si el módulo ET-MINI ENC28J60 se conecta a un microcontrolador que esta alimentado por una fuente de poder externa de +3.3V, el puente deberá colocarse entre los punto 2 y 3, es decir, entre VDD y +3.3V. Esto permitirá que el chip ENC28J60 se conecte directamente a la fuente de +3.3V, deshabilitando de esta manera el circuito regulador de voltaje (LM3940). Si el módulo de comunicación Ethernet se conecta a un microcontrolador alimentado por una fuente externa de +5V, entonces el jumper deberá colocarse entre los puntos 1 y 2, es decir, entre +5V y VDD. De esta manera se habilitará el circuito regulador de voltaje, proporcionándole al chip ENC28J60 una alimentación de +3.3V, que es el voltaje de operación de este integrado.

CAPÍTULO 3

DISEÑO DEL SOFTWARE Y HARDWARE DE MONITOREO Y CONTROL.

3.1 Diagrama de bloques del diseño propuesto.

En este capítulo describiremos en detalle el diseño del software y hardware utilizado para alcanzar el objetivo propuesto. Para empezar se describirá de manera general el diseño del hardware utilizado, para luego sistemáticamente abordar los detalles técnicos necesarios para comprender a cabalidad el comportamiento del sistema en cuestión.

El hardware del sistema esta conformado por tres bloques que garantizan un adecuado tratamiento de los datos, de tal manera que se puedan presentar de manera comprensible para el usuario, quien podrá acceder a estos datos a través de una computadora personal.

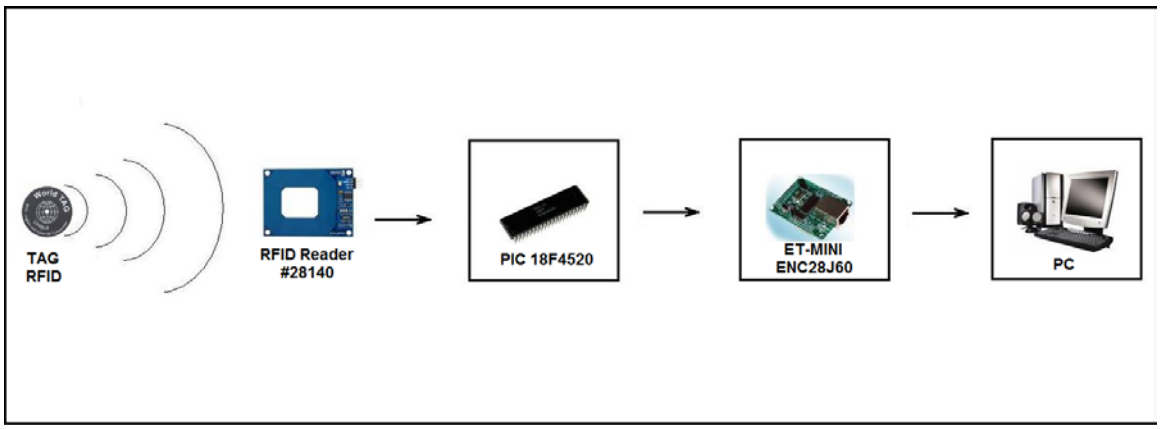


Figura 3.1 Diagrama del Diseño Propuesto.

El primer bloque, es el conformado por los sensores que permiten la adquisición y digitalización de los datos. El bloque de sensores esta conformado por el módulo RFID Reader #28140 y la etiqueta RFID que posee el número ID único necesario para garantizar la estadía de los equipos del laboratorio en las inmediaciones del mismo. El sensor RFID permite la conversión de las señales de radiofrecuencia obtenidas del tag RFID en señales digitales que pueden ser manipuladas por los bloques posteriores.

El siguiente bloque es el conformado por el PIC18F4520, este bloque microcontrolador es el encargado de proporcionar la interfaz necesaria entre el módulo RFID Reader y el módulo ET-MINI ENC28J60.

El PIC18F4520 adquiere los datos de tipo serial del RFID Reader y los procesa de tal manera que puedan ser enviados y manipulados, según la aplicación requerida, en el siguiente bloque del diseño presentado.

Una vez que los datos han sido procesados por el bloque microcontrolador, son enviados al bloque de comunicación ethernet, que está conformado por el módulo ET-MINI ENC28J60, este bloque permite la comunicación entre el sistema desarrollado y cualquier computador personal que utilice el estándar Ethernet. Esta comunicación es posible gracias a la presencia del chip ENC28J60 en su circuitería interna.

Finalmente, el bloque de comunicación ethernet envía los datos adquiridos por el sensor a través de la red hacia el host del administrador del laboratorio, quien podrá monitorear constantemente el sistema utilizando las herramientas de software pertinentes.

El análisis hecho previamente proporciona una idea general del sistema diseñado para proporcionar la seguridad del laboratorio. A continuación presentamos el diagrama esquemático del PCB implementado.

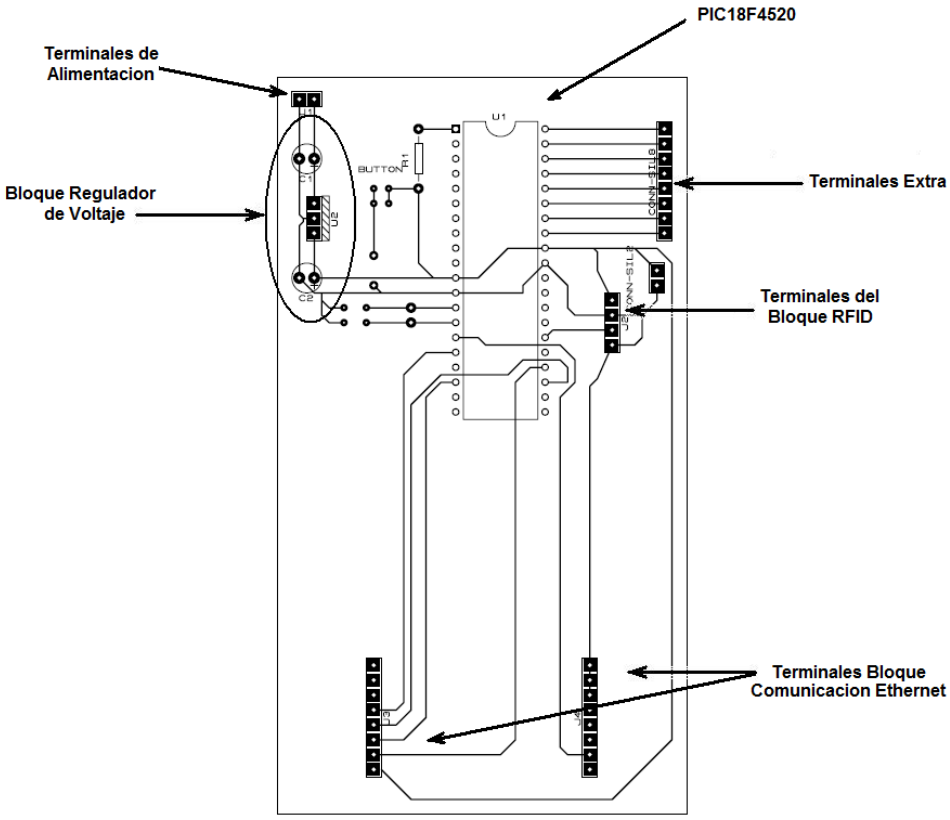


Figura 3.2 Detalle del diseño del Circuito Impreso.

Como podemos apreciar, el PCB esta conformado por tres bloques fundamentales que son los descritos previamente, estos bloques son: el bloque RFID, el bloque microcontrolador y el bloque de comunicación ethernet. Además fue necesaria la introducción de un bloque regulador de voltaje, que proporciona el potencial eléctrico necesario para que todo el sistema funcione adecuadamente. Este bloque esta conformado por dos capacitares electrolíticos y un chip regulador LM7805, que proporciona un voltaje de salida de 5 VDC, que es el requerido por el sistema.

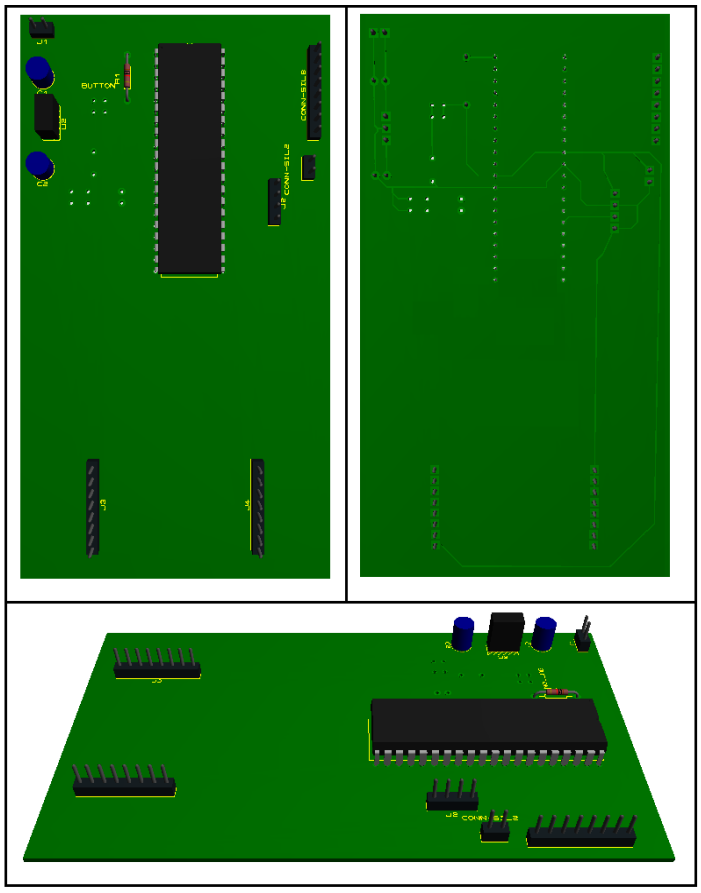


Figura 3.3 Simulación del Circuito Impreso.

3.2 Análisis del código en MikroBasic.

El código desarrollado en MikroBasic es de suma importancia en el desarrollo del proyecto, pues le proporciona al PIC18F4520 las instrucciones que requiere para poder operar apropiadamente, por lo tanto, debe ser desarrollado minuciosa y eficientemente para que el sistema pueda trabajar con absoluta confianza y utilizando la menor cantidad de memoria posible y de esta manera garantizar un tiempo de procesamiento relativamente bajo y un uso mas eficiente de los recursos proporcionados por el chip microcontrolador.

Este código y su respectivo análisis se presentan a continuación:

```
program Seguridad_equipos
include "eth_enc28j60"
include "eth_enc28j60_api"

  dim mymacaddr  as byte [6]
  dim myipaddr   as byte [4]
  dim getRequest as byte[20]
  dim dyna       as byte[30]
  dim txt        as string[100]
  dim i          as byte
  dim IpAddr     as byte[4]
  dim cnt        as byte[12]
  dim tag as string [12]
  dim cc as byte
  dim delim as char[1]
  sub function Spi_Ethernet_UserTCP(dim byref remoteHost as byte[4], dim
remotePort, localPort, reqLength as word) as word
    result=0
  end sub
  sub function Spi_Ethernet_UserUDP(dim byref remoteHost as byte[4], dim
remotePort, destPort, reqLength as word) as word
    result = 0
```

```
if destport = 200 then
  for i=0 to 3
    getRequest[i]=spi_ethernet_getbyte()
  next i
  if getrequest[0]="R" then
    if getRequest[1]="D" then
      if getRequest[2]="O" then
        select case getRequest[3]
          case "0"
            PORTD.0=1
            txt = "RD0 ENCENDIDO"
          case "1"
            PORTD.1=1
            txt = "RD1 ENCENDIDO"
          case "2"
            PORTD.2=1
            txt = "RD2 ENCENDIDO"
          case "3"
            PORTD.3=1
            txt = "RD3 ENCENDIDO"
          case "4"
            PORTD.4=1
            txt = "RD4 ENCENDIDO"
          case "5"
            PORTD.5=1
            txt = "RD5 ENCENDIDO"
          case "6"
            PORTD.6=1
            txt = "RD6 ENCENDIDO"
          case "7"
            PORTD.7=1
            txt = "RD7 ENCENDIDO"
        end select
      end if
    end if
  end if
  if getrequest[0]="R" then
    if getRequest[1]="D" then
      if getRequest[2]="F" then
        select case getRequest[3]
```

```
            case "0"
              PORTD.0=0
              txt = "RD0 APAGADO"

            case "1"
              PORTD.1=0
              txt = "RD1 APAGADO"
            case "2"
              PORTD.2=0
              txt = "RD2 APAGADO"
            case "3"
              PORTD.3=0
              txt = "RD3 APAGADO"
            case "4"
              PORTD.4=0
              txt = "RD4 APAGADO"
            case "5"
              PORTD.5=0
              txt = "RD5 APAGADO"
            case "6"
              PORTD.6=0
              txt = "RD6 APAGADO"
            case "7"
              PORTD.7=0
              txt = "RD7 APAGADO"
          end select
        end if
      end if
    end if
    result = 13 + reqLength
    spi_ethernet_putbytes(@txt,13)
    while(reqLength <> 0)
      spi_ethernet_putbyte(spi_ethernet_getByte())
      reqLength = reqLength - 1
    wend
  end if
end sub
```

```
main:
 adcon0=0
 adcon1=15
 trisa=0
 porta=0
 trisb=$80
 portb=0
 trisd=0
 portd=0

 mymacaddr[0]=0x00
 mymacaddr[1]=0x0F
 mymacaddr[2]=0x3D
 mymacaddr[3]=0xCB
 mymacaddr[4]=0x2F
 mymacaddr[5]=0x81

 myipaddr[0]=192
 myipaddr[1]=168
 myipaddr[2]=46
 myipaddr[3]=207


 spi_init()
 spi_ethernet_init(portc,0,portc,1,mymacaddr,myipaddr,1)

 IpAddr[0] = 192
 IpAddr[1] = 168
 IpAddr[2] = 46
 IpAddr[3] = 209

 Usart_Init(2400)
 cc = chr(13)
 bytetostr(cc,delim)
' delim =  "  "
  txt=""
  while true
  if Usart_Data_Ready = 1 then
     Usart_Read_text(tag, delim)
```

```
      delay_ms(1000)
      spi_ethernet_sendUDP(IpAddr, 10001, 4000, @tag, Strlen(tag))
      delay_ms(1000)
    end if
      spi_ethernet_dopacket()
 wend
end.
```

El análisis de este código se presenta a continuación:

El primer bloque por analizar será el correspondiente a la declaración de variables.

```
program Seguridad_equipos
include "eth_enc28j60"
include "eth_enc28j60_api"
 dim mymacaddr  as byte [6]
 dim myipaddr   as byte [4]
 dim getRequest as byte[20]
 dim dyna       as byte[30]
 dim txt        as string[100]
 dim i          as byte
 dim IpAddr     as byte[4]
 dim cnt        as byte[12]
 dim tag as string [12]
 dim cc as byte
 dim delim as char[1]
```

En este bloque se hace el llamado a las librerías Ethernet que permiten la comunicación utilizando este protocolo en conjunto con el chip ENC28J60. Luego se procede a la declaración de las variables, aquí especificamos al software cada una de las variables que vamos a utilizar durante el funcionamiento del PIC18F4520.

A continuación se procede a la programación de las subrutinas que permiten al microcontrolador manejar los datos utilizando el Protocolo de Datagrama de Usuario (UDP).

```
 sub function Spi_Ethernet_UserTCP(dim byref remoteHost as byte[4], dim
remotePort, localPort, reqLength as word) as word
  result=0
 end sub
 sub function Spi_Ethernet_UserUDP(dim byref remoteHost as byte[4], dim
remotePort, destPort, reqLength as word) as word
```

Estas funciones habilitan la comunicación entre el microcontrolador y el módulo ET-MINI ENC28J60. Todo esto se logra utilizando la interfaz SPI, que permite la transferencia de datos desde el PIC18F4520 hacia la tarjeta Ethernet, y el protocolo de comunicaciones informáticas UDP que permite la transferencia de información hacia el computador del administrador.

Una vez que se ha ejecutado la subrutina de comunicación Ethernet, el sistema ejecuta el proceso principal. En este proceso se le indica al bloque microcontrolador primeramente, cuales puertos deben habilitarse como entradas o salidas, y se especifican los puertos que han sido designados como analógicos o digitales. Esto se puede apreciar en el código presentado previamente.

Luego es necesario especificar las direcciones IP y MAC que van a ser utilizadas durante el proceso, para hacerlo, es necesario introducir dicha información en cada una de las variables designadas para cumplir este rol. Se sugiere tener mucho cuidado en la asignación de esta información al sistema, pues si se proporciona una dirección errada al bloque microcontrolador, el sistema no funcionará debidamente, pues no podrá proporcionar la información requerida al computador del administrador debido a la ausencia de conectividad en la red o a un mal enrutamiento de los datos adquiridos.

Finalmente se procede al enrutamiento de los datos hacia el host del administrador utilizando las funciones presentadas en las últimas líneas del código descrito.

3.3 Diagrama esquemático de la programación en lenguaje G.

Una vez que se ha implementado el hardware que permite el envío de los datos adquiridos desde la etiqueta, adherida a cada uno de los equipos del laboratorio, hacia la computadora personal del administrador, es necesario utilizar un software que permita al administrador visualizar y monitorear estos datos de tal manera que pueda tomar decisiones tomando como base la información proporcionada por este programa. Por lo tanto se requiere que el software utilizado sea confiable, y proporcione una interfaz amigable para el usuario.

Una de las herramientas utilizadas para este fin es LabVIEW, que es un software de instrumentación virtual que cumple con las características mencionadas previamente.

Este software utiliza un lenguaje de programación gráfico o Lenguaje G que permite desarrollar aplicaciones de manera más versátil y comprensible a diferencia de los lenguajes de programación tradicionales que en ocasiones suelen ser largos y confusos.

A continuación se presenta el programa desarrollado en lenguaje G que permite interactuar al usuario con el sistema implementado previamente.

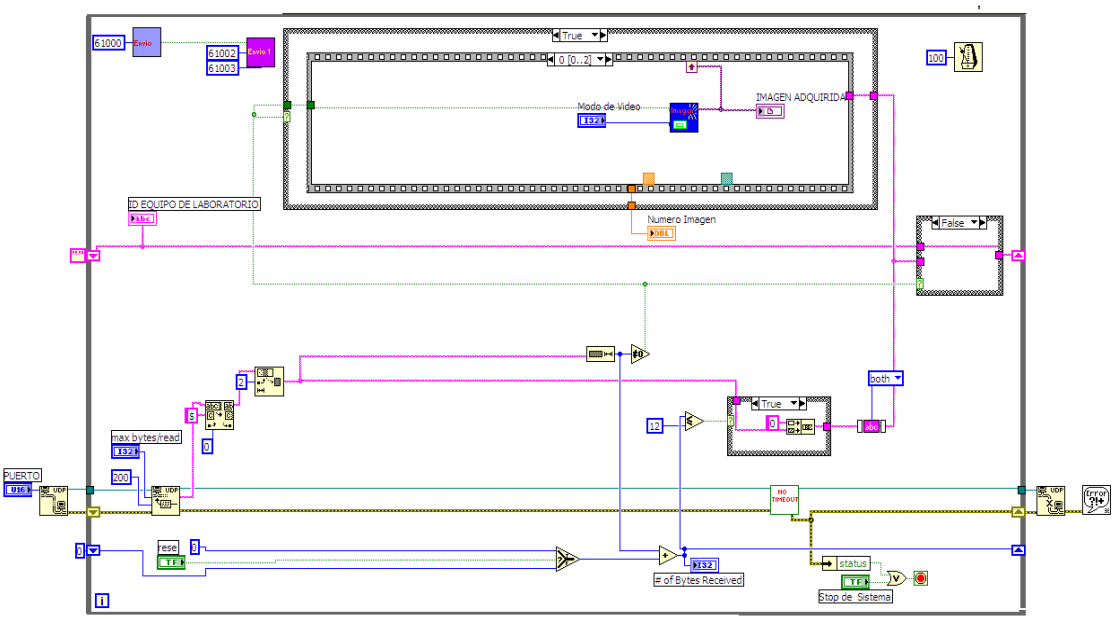


Figura 3.4 Diseño en LabVIEW del Sistema.

El diagrama de bloques de este programa es extenso y se subdivide en dos bloques.

El primer bloque permite la adquisición de imágenes y la comunicación con MySQL. Mientras que el segundo bloque permite la comunicación Ethernet y la adquisición del ID de la etiqueta RFID. A continuación se analizará en detalle el funcionamiento de cada uno de estos bloques.

Primeramente empezaremos con el bloque de adquisición de imágenes y comunicación DataBase, este bloque se presenta en la siguiente figura:

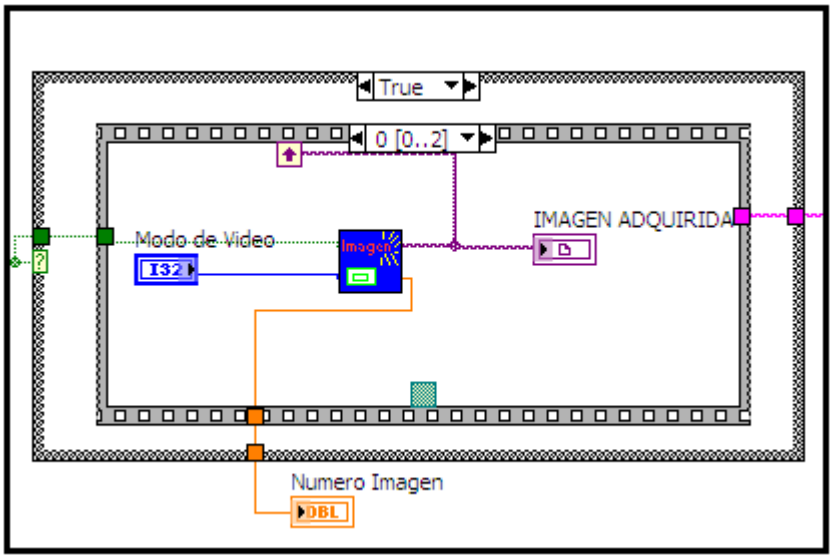


Figura 3.5 Secuencia de Manejo de Imágenes.

Este bloque tiene como función principal una estructura de tipo case, cuya condición es una variable de tipo booleano que indica si ha sido recibido o no el ID del tag RFID. En caso de que esta variable adquiera el valor TRUE, es decir que sea verdadera, se accederá a la estructura secuencial mostrada en la figura anterior. Esta estructura esta conformada por tres secuencias que se procederán a analizar a continuación.

En la primera secuencia se recurre al uso de un SubVI denominado imagen.

Esta función requiere como dato de entrada el parámetro Modo de Video, que es ingresado a través del panel frontal por el usuario del sistema, como se vera posteriormente y devuelve la imagen adquirida y el número de imagen.

El contenido de este SubVI se presenta a continuación:

La función imagen tiene como estructura principal un case, cuya condición es la variable Ingresa Dato, como se puede apreciar en la figura.

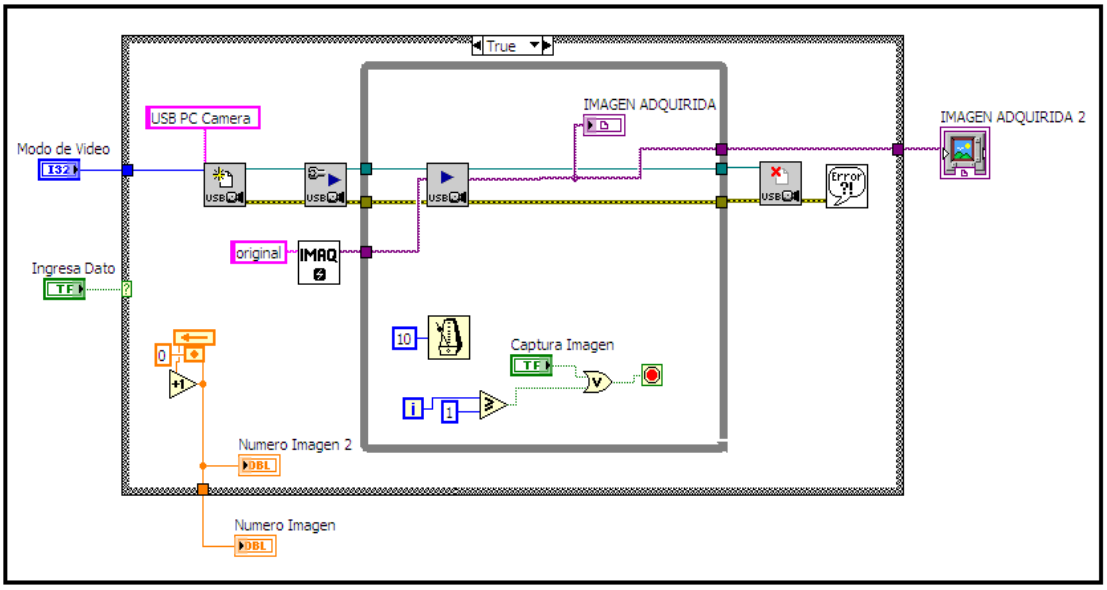


Figura 3.6 Detalle del SubVI Imagen.

Este bloque utiliza, en su mayor parte, funciones de la librería USB vision que permiten adquirir datos desde una cámara Web a través del puerto USB del PC. Como podemos apreciar en la figura mostrada en la parte superior, la primera función utilizada es la denominada IMAQ USB init, que requiere de dos parámetros para funcionar apropiadamente, el primero es el parámetro USB Camera Name que especifica el nombre de la cámara que va a ser utilizada, y el siguiente parámetro utilizado es el modo de video requerido.

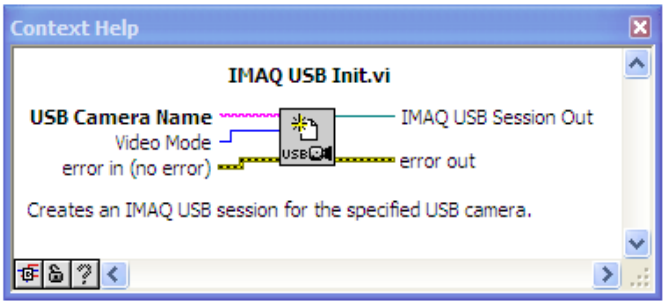


Figura 3.7 Función IMAQ USB Init.

Esta función permite iniciar una sesión IMAQ USB en la cámara especificada en el parámetro USB Camera Name.

Entonces este bloque empieza inicializando una sesión IMAQ USB, utilizando los controladores USB PC Camera y Modo de Video que son valores especificados por el usuario a través del Front Panel de LabVIEW.

Como se puede apreciar en esta imagen del diseño presentado en el panel frontal, los valores requeridos para inicializar el sistema son entre otros, el nombre de la cámara Web USB que se va a utilizar y el modo de video con el que desee trabajar el usuario.

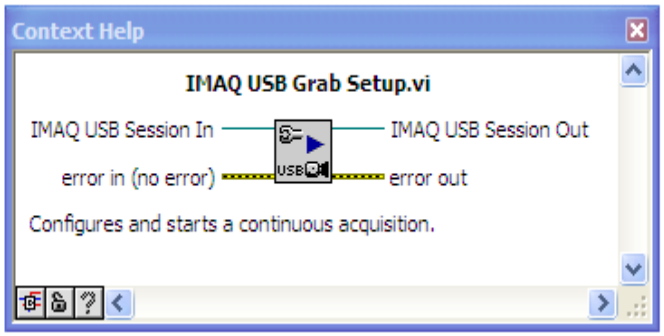


Figura 3.8 Función IMAQ USB Grab Setup.

Esta información es enviada a través de las variables de control, especificadas previamente, hacia el Block Diagram de LabVIEW para poder iniciar la sesión con la webcam. Una vez iniciada la sesión, el siguiente paso es activar la adquisición de datos de manera continua, esto se logra utilizando la función IMAQ USB Grab Setup. Este instrumento virtual presentado en la imagen inferior requiere básicamente de dos parámetros para operar. El primero es el denominado IMAQ USB Session In, que indica al instrumento virtual la sesión IMAQ USB desde la cual va a adquirir los datos de manera continua.

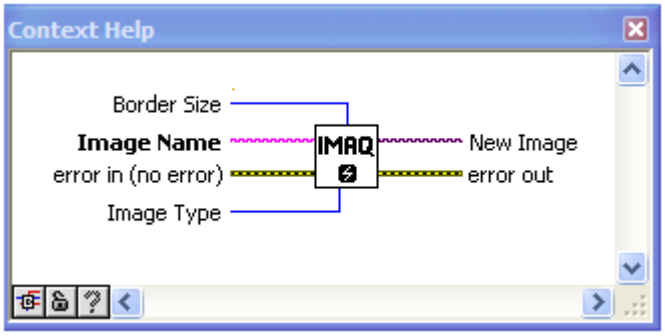


Figura 3.9 Función IMAQ.

En este caso específico ese parámetro se refiere a la sesión iniciada con la cámara Web especificada por el usuario en el panel frontal de LabVIEW.

El siguiente parámetro es el correspondiente al error de entrada que indica si ha existido algún error en el proceso previo a este instrumento virtual. Ambos parámetros son proporcionados por las salidas correspondientes de la función previamente implementada, como se puede apreciar en el gráfico correspondiente al diseño de este primer bloque de manejo de imágenes.

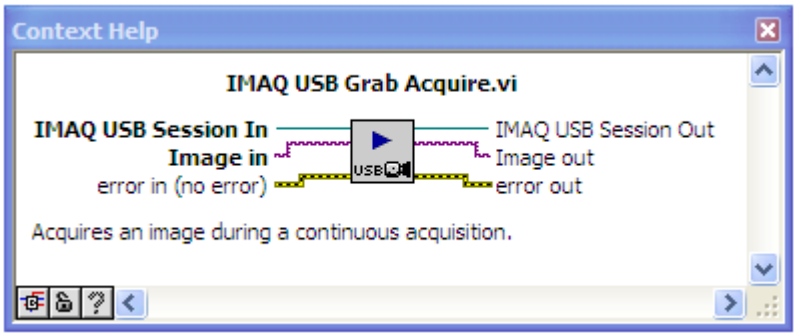


Figura 3.10 Función IMAQ USB Grab Acquire.

Paralelamente al instrumento virtual descrito y fuera del lazo while interno, está la función IMAQ Create. Este VI permite crear o reservar un espacio de memoria temporalmente para almacenar una imagen adquirida usando una cámara Web USB. Como se puede apreciar en la figura adjunta esta función puede manejar varios parámetros de entrada pero en la aplicación especifica que estamos analizando, se requiere únicamente el ingreso del parámetro Image Name, que indica el nombre que

se va a asignar a la imagen almacenada en el espacio de memoria reservado temporalmente.

La siguiente función implementada en el diseño de este primer bloque, es la denominada IMAQ USB Grab Acquire. Este instrumento virtual permite adquirir una imagen durante una adquisición continua, esta adquisición continua ha sido activada previamente por la función IMAQ USB Grab Setup mencionada anteriormente. Como se puede observar en la figura adjunta, este instrumento virtual opera utilizando tres parámetros de entrada, estos parámetros son proporcionados por las funciones previamente implementadas. A la salida de esta función tenemos el error out, el IMAQ USB Session Out y el Image Out, este último parámetro corresponde a la imagen adquirida por la cámara web en el momento preciso en que el tag RFID empezó a transmitir una señal de radiofrecuencia al lector RFID.

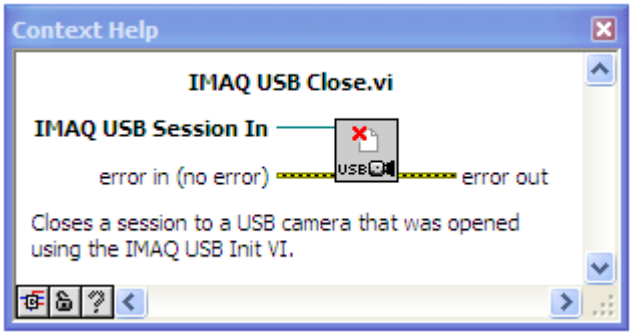


Figura 3.11 Función IMAQ USB Close.

Esta imagen es presentada a través de un indicador virtual en el panel frontal de LabVIEW.

Finalmente se procede a cerrar la sesión IMAQ USB utilizando la función IMAQ USB Close mostrada en la figura.

La siguiente secuencia de este bloque permite el almacenamiento de la imagen adquirida, para lograrlo se utiliza el ID de Relación el cual se concatena con la ruta de almacenamiento deseada.

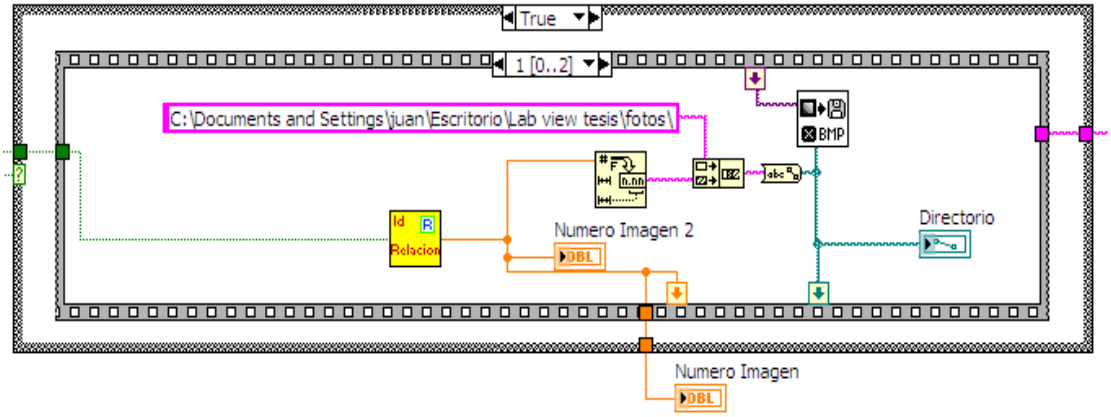


Figura 3.12 Secuencia de almacenamiento de imagen.

Como se puede apreciar en la figura de esta secuencia (figura 3.12), la ruta o path de la imagen adquirida se pasa a la siguiente secuencia.

Finalmente la última secuencia de este bloque se presenta a continuación.

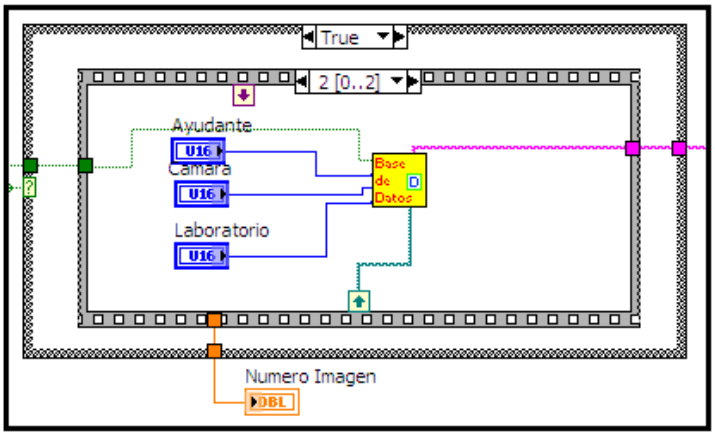


Figura 3.13 Secuencia de Manejo de Base de Datos.

Esta secuencia permitirá interactuar al usuario con un software de manejo y administración de bases de datos utilizando a LabVIEW como interfaz de comunicación. Esto se logra gracias al SubVI denominado Base de Datos, que permite iniciar una sesión con MySQL.

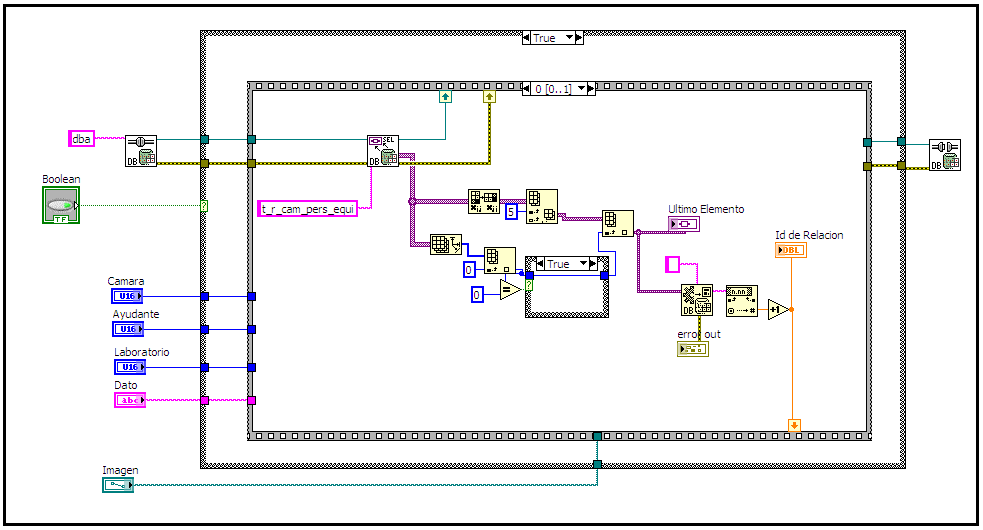


Figura 3.14 Detalle del SubVI Base de Datos.

El contenido de la función Base se Datos, que se mostró en la figura 3.14 utiliza como datos de entrada los parámetros cámara, ayudante, laboratorio y dato; que son ingresados por el usuario a través del panel frontal de LabVIEW, además utiliza el path de la imagen adquirida en secuencias previas. Este SubVI trabaja con funciones de la librería data base de LabVIEW, las cuales han sido presentadas en capítulos posteriores.

Como pudimos observar en la figura anterior, esta función empieza iniciando una conexión con una base de datos, en este caso específico nos referimos a una base de datos creada en MySQL, esto se logra gracias a la utilización de la función DB Tools Open Connection analizada anteriormente.

Este SubVI utiliza una estructura secuencial para el manejo de la información que se va a almacenar en la base de datos.

La primera secuencia inicia extrayendo los datos de la tabla t_r_cam_pers_equi, y luego manipula estos datos utilizando varias funciones de la librería array, de tal manera que finalmente se pueda obtener el ID de la relación, valor que se almacena en el indicador denominado ID de relación y se pasa a la siguiente secuencia para poder introducir este valor en uno de los campos de las tablas creadas por el sistema, en la base de datos de MySQL que permitirá monitorear de manera continua el estado de los equipos bajo la administración de este sistema de seguridad.

Ahora vamos a proceder a analizar la siguiente secuencia de este SubVI.

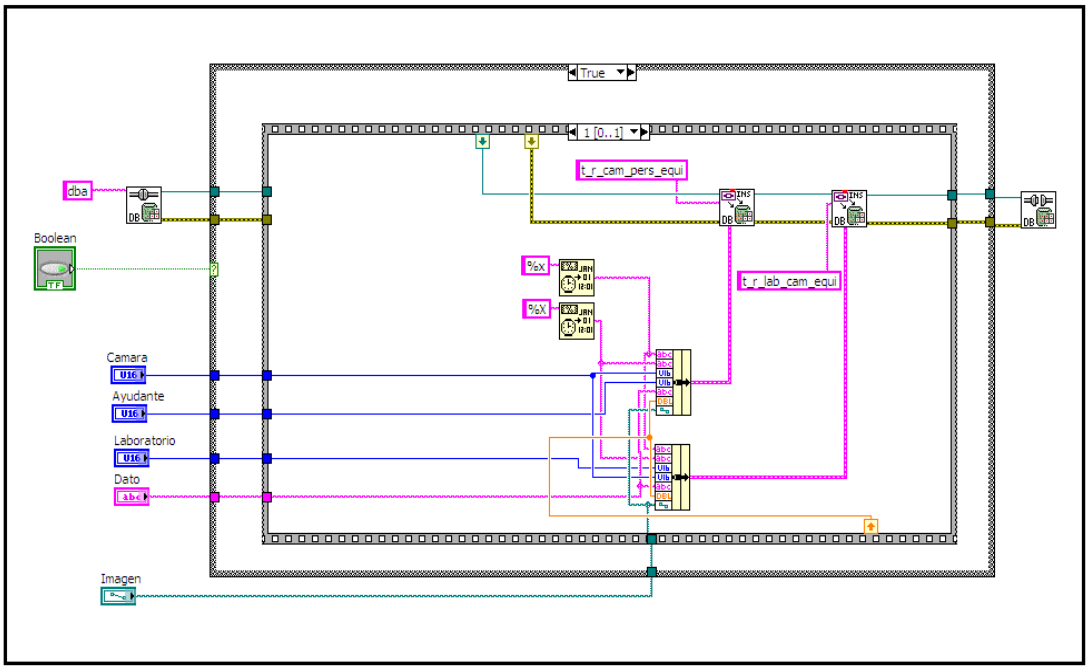


Figura 3.15 Segunda secuencia del SubVI Base de Datos.

En esta secuencia se crean dos clusters cuyos elementos son, en el caso del primer cluster la fecha en la que el sistema recibió datos del tag, la cámara utilizada en aquel momento, el ayudante que se encuentra presente en ese momento en el laboratorio, el ID del tag, la ruta de la imagen y el número de imagen. El segundo cluster esta conformado por la fecha, la cámara, el ID del tag, el laboratorio en el cual se recibió dicha información, la ruta de la imagen y el número de imagen.

Toda esta información se ingresa en el sistema de monitoreo manejado por MySQL a través de la función DB Tools Insert Data. Esta función le indica al sistema la información que debe almacenar y donde hacerlo. En este caso la información se

almacenará en dos tablas diferentes creadas en MySQL, la primera tabla denominada t_r_cam_pers_equi contendrá la información proporcionada por el primer cluster, esto es, fecha, cámara, ID y ayudante. La segunda tabla denominada t_r_lab_cam_equi contiene la información del segundo cluster, esto es, fecha, la cámara, ID y el laboratorio. Todo esto sucede asumiendo que la estructura case es verdadera (TRUE) en caso contrario, la conexión con MySQL se mantendrá activa aunque ningún dato se enviará a las tablas antes mencionadas.

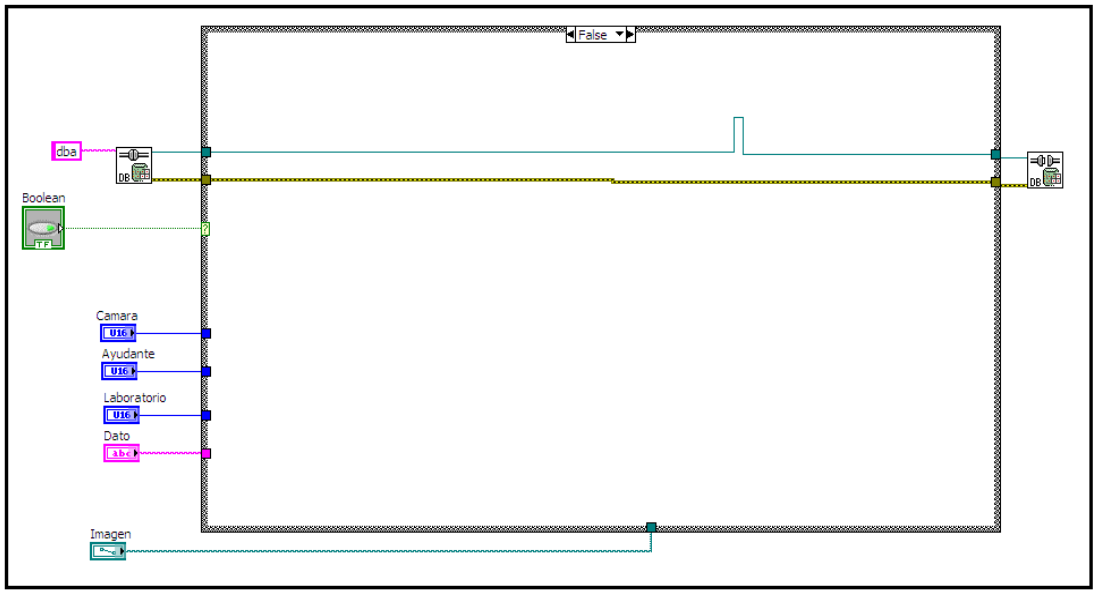


Figura 3.16 Tercera Secuencia del SubVI Base de Datos.

El segundo bloque es el que permite la comunicación ethernet utilizando el estándar UDP como protocolo de transporte para enviar los datos al computador personal. Este bloque utiliza funciones de la librería UDP analizadas previamente en un capítulo

posterior. Como se puede apreciar en la figura presentada a continuación la sesión UDP se inicializa tomando como parámetro de entrada la variable de control denominada Puerto, que es un valor ingresado por el usuario a través de la interfaz presentada en el panel frontal de LabVIEW.

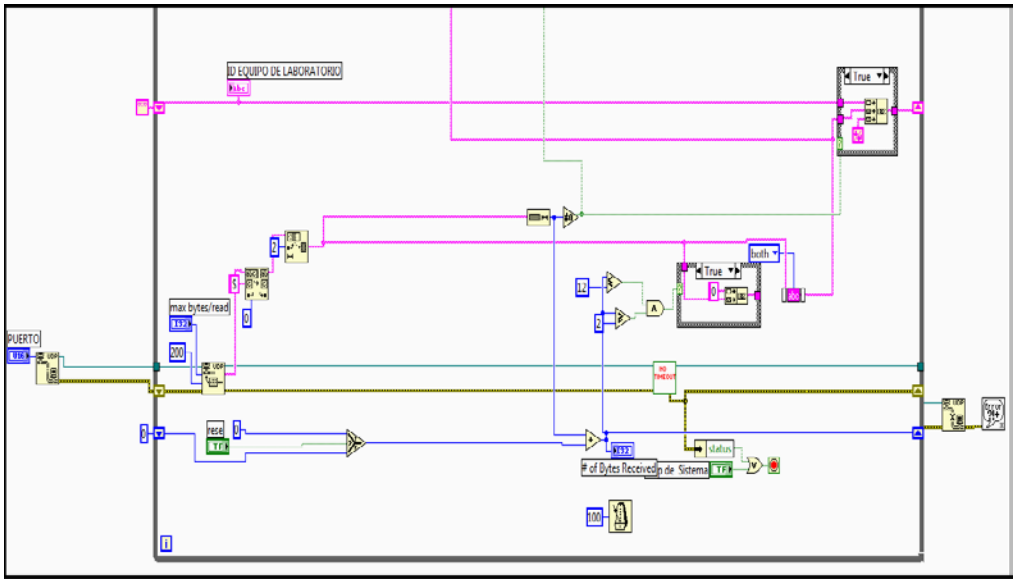


Figura 3.17 Bloque de comunicación UDP.

Una vez iniciada la sesión UDP se procede a la lectura de los datos utilizando la función UDP Read presentada anteriormente, a esta función se le ingresa el número máximo de bytes leídos y el retardo en milisegundos, que en este caso específico es de 200ms. Una vez hecho esto se eliminan los caracteres indeseados que han sido adquiridos por el sistema utilizando funciones de tipo string y estructuras de tipo case con el fin de obtener el ID de la etiqueta RFID. Una vez terminado el proceso de

eliminación de datos indeseados, se procede a cerrar la sesión UDP utilizando la función UDP Close.

Finalmente se analizará un bloque extra conformado por los SubVIs denominados Envío y Envío1. Estas funciones permiten visualizar la base de datos del sistema desde otra PC. A continuación se describe su funcionamiento.

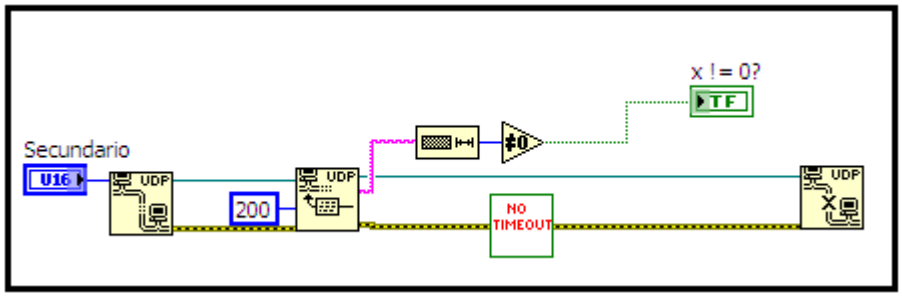


Figura 3.18 Función Envío.

Primeramente analizaremos la función Envío, que es la mostrada en la figura 3.18. Esta función devuelve un valor de tipo booleano que indica si la longitud del dato adquirido en la sesión UDP inicializada previamente es diferente de cero. Este dato se utiliza en el siguiente SubVI de este bloque como se verá a continuación. Luego de ello este SubVI cierra la sesión UDP utilizando la función UDP Close presentada en un capítulo posterior.

La siguiente función, denominada Envío1 tiene como parámetro de entrada una variable de tipo booleano, que en el bloque que estamos analizando corresponde al

parámetro de salida del SubVI Envío. El código de esta función se presenta en la figura 3.19.

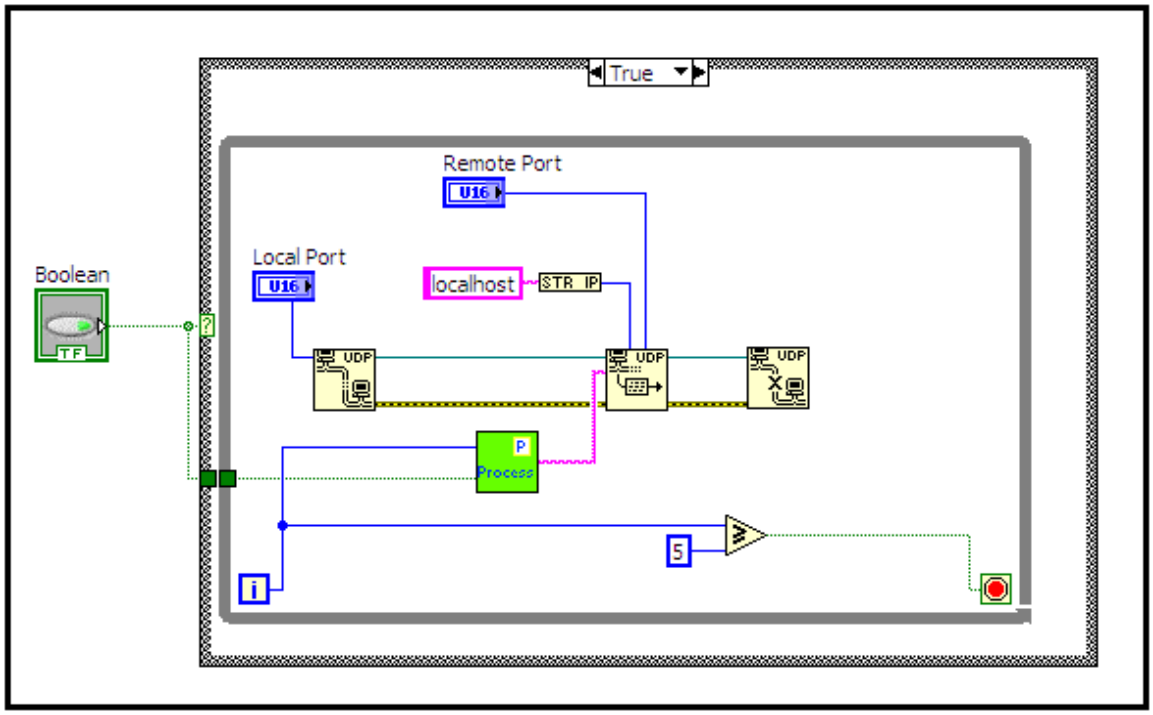
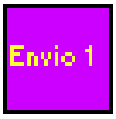


Figura 3.19 Función Envío1.



Esta función permite la inserción o escritura UDP de datos en la dirección especificada por la variable localhost. Esto se logra gracias a la utilización de la función UDP Write descrita con anterioridad. Como se puede observar en la figura 3.19, el SubVI denominado process proporciona los datos a UDP Write. A continuación presentamos el contenido de este SubVI.

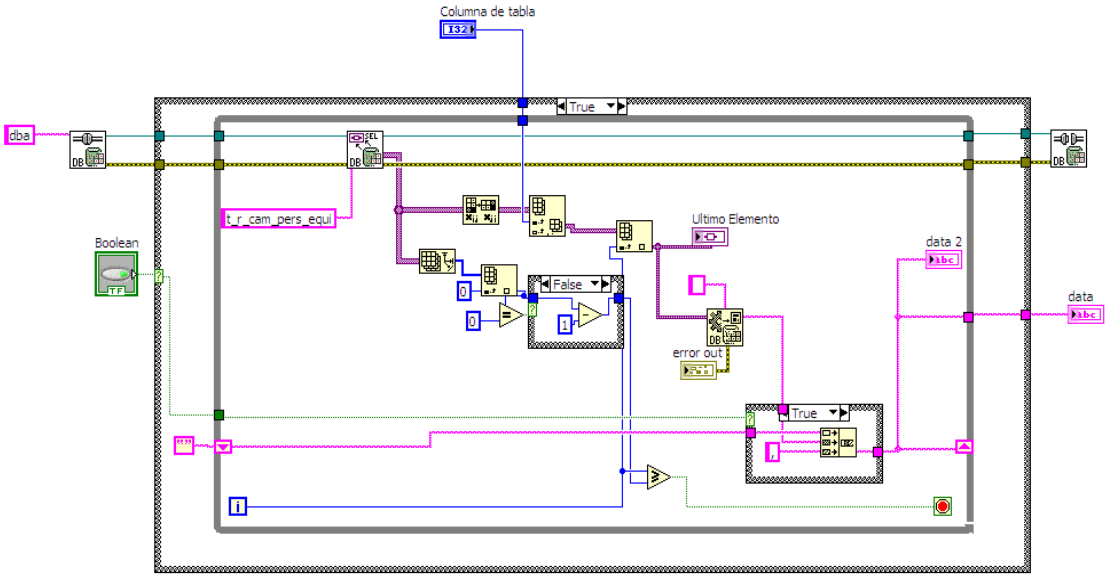


Figura 3.20 Función Process.

Esta función permite la adquisición de los datos provenientes de la tabla t_r_cam_pers_equi de MySQL. Esto se logra utilizando funciones de las librerías Database y Array analizadas con anterioridad. Una vez adquiridos estos datos los envía de vuelta a la función UDP Write del SubVI Envío1.

Finalmente luego de haber analizado el código en lenguaje G del sistema planteado, se procederá a presentar el Instrumento Virtual utilizado para la adquisición y manejo de los datos adquiridos desde la base de datos de MySQL en un computador remoto conectado a la red del sistema.

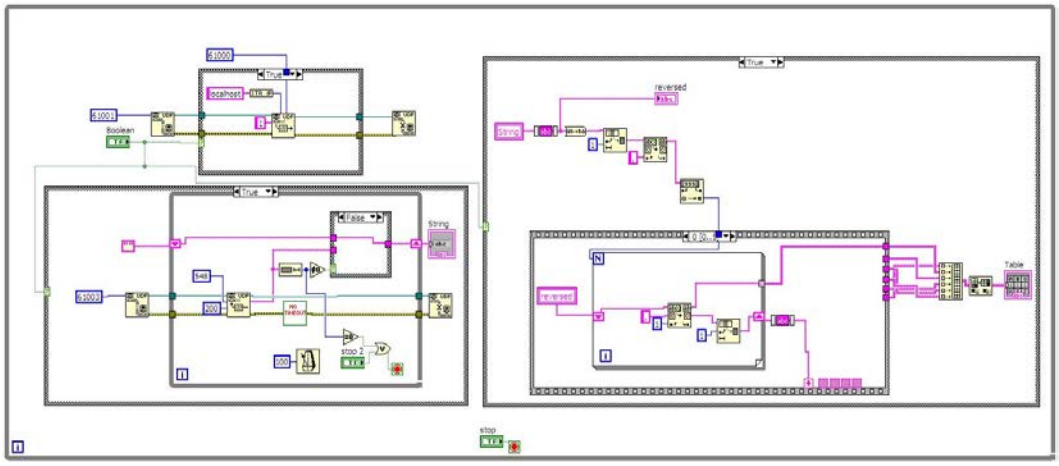


Figura 3.21 Instrumento Virtual de recepción de datos.

El Instrumento Virtual presentado en la figura 3.21 permite la recepción de los datos en un host remoto para poder monitorear y tener acceso a la base de datos del sistema desde una computadora que se encuentre conectada a la red. El panel frontal de este Instrumento Virtual se presenta en la figura 3.22.

El panel frontal ha sido diseñado de tal manera que sea fácil de manejar y al mismo tiempo, satisfaga las necesidades del usuario del host remoto. Como se puede apreciar, este panel puede proporcionar toda la información almacenada en las tablas de la base de datos del sistema pues posee una tabla con cada uno de los campos que se requiere para poder adquirir la información del computador de la administración del laboratorio, el cual funcionará como servidor del sistema.

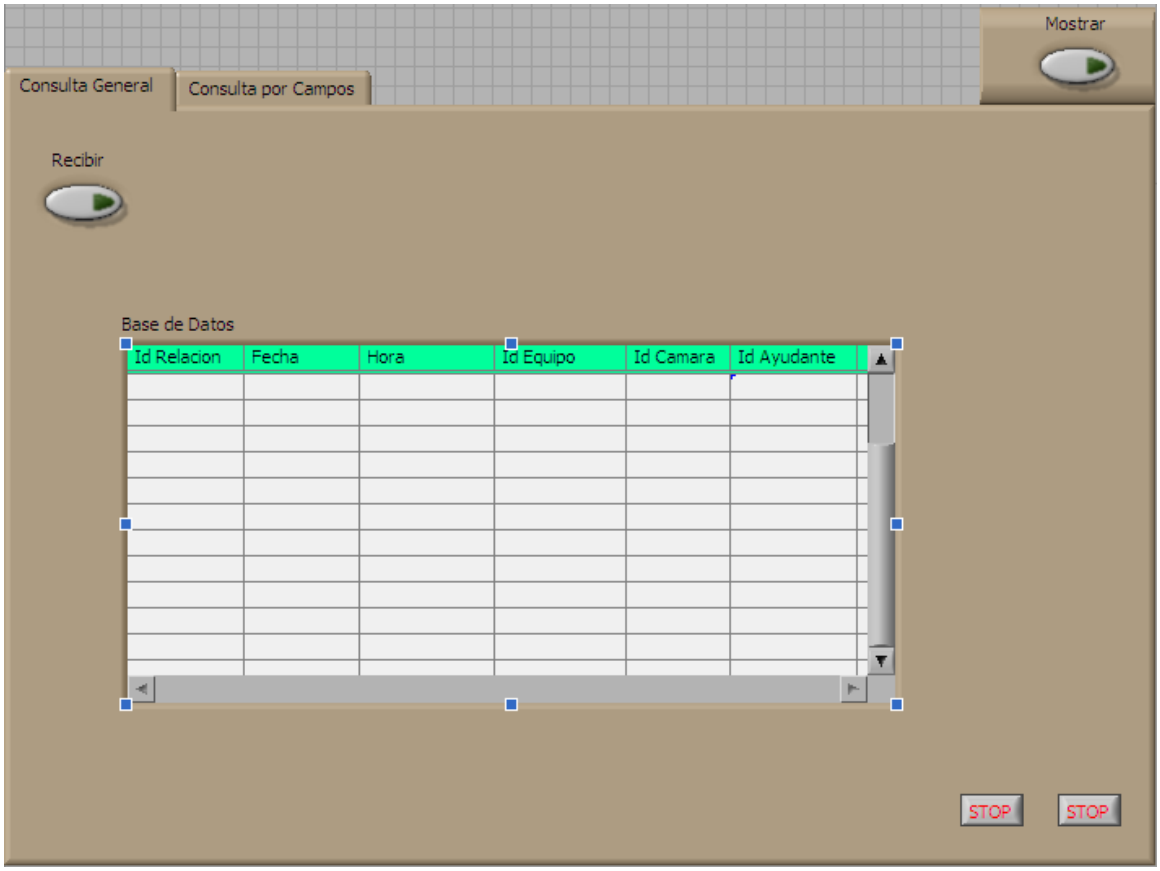


Figura 3.22 Panel Frontal de recepción de datos.

A continuación se presenta el panel frontal utilizado para hacer las veces de interfaz de comunicación con el usuario, en este caso el administrador del laboratorio, y LabVIEW. Se ha procurado utilizar una interfaz grafica sencilla y amigable con el usuario con la finalidad de facilitar la utilización del sistema. Esta interfaz esta conformada por varios campos que deben ser llenados por el administrador del laboratorio para que el sistema pueda funcionar apropiadamente.

Figura 3.23 Panel Frontal del Sistema.

Como se puede observar en la imagen, esta interfaz está conformada por cinco bloques. El primer bloque es el de datos adquiridos, el segundo es la imagen adquirida, el siguiente corresponde al sistema de control, luego viene el bloque de control de video y finalmente el bloque de parada de emergencia.

En el siguiente capítulo se describirá más ampliamente la forma de utilizar el sistema y la puesta prueba del mismo, con la finalidad de guiar al usuario en el manejo y comprensión de la interfaz desarrollada.

3.4 Descripción del sistema usado para la comunicación entre LabVIEW y MySQL.

Como se ha explicado en secciones anteriores, el proyecto se ha desarrollado de tal manera que el usuario pueda utilizar el panel frontal de LabVIEW como interfaz para la adquisición y exportación de datos hacia la herramienta de administración de base de datos utilizada por el usuario, para de esta manera, poder monitorear constantemente la información adquirida por el sistema. Además es posible administrar o monitorear la base de datos del sistema de manera remota gracias las herramientas proporcionadas por LabVIEW que hacen posible que la información de las tablas de la base de datos creadas en MySQL, se puedan enviar vía Ethernet hacia un host remoto que podrá acceder a esta información de manera periódica, haciendo posible de esta manera, realizar un monitoreo remoto cuando las circunstancias lo ameriten.

Debido a ello es imprescindible la comunicación entre LabVIEW y MySQL.

Como se mencionó en la sección anterior, el diseño en lenguaje G implementado hace posible esta comunicación, gracias a la utilización de las funciones de la librería database de LabVIEW. Para que el usuario pueda acceder a las tablas de MySQL creadas por LabVIEW será necesario iniciar una conexión MySQL.

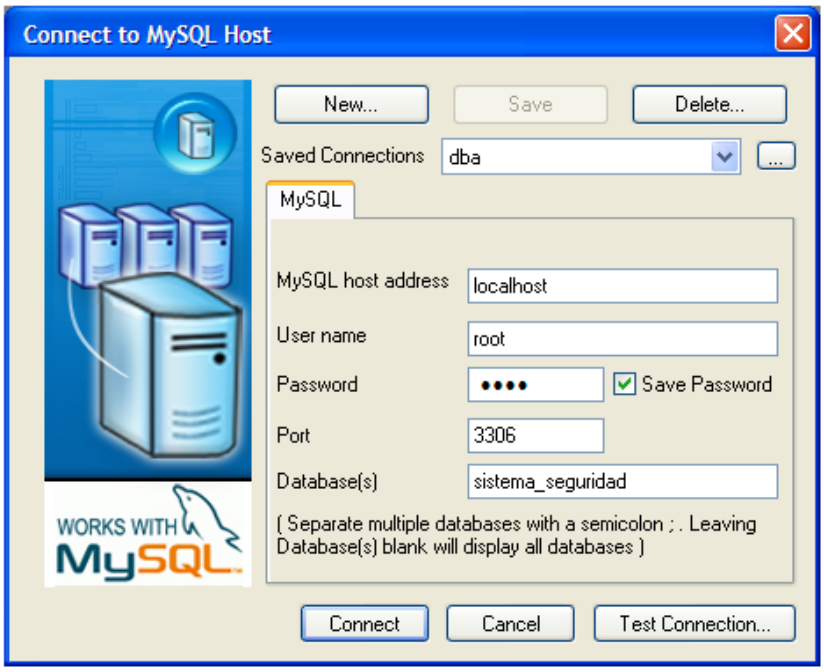


Figura 3.24 Conexión MySQL.

En este caso específico, iniciamos una conexión denominada dba, a través del puerto 3306 presentado por defecto y denominamos a la base de datos “sistema_seguridad”, tal y como se puede observar en la figura presentada en la pagina anterior.

El sistema enviará información a esta base de datos, pues en el código de LabVIEW se designó a la conexión dba, en el bloque de adquisición de imágenes y comunicación con MySQL visto en la sección anterior, como la conexión con la cual se comunicará LabVIEW. Entonces, cada vez que el sistema lo requiera, enviará información a las tablas correspondientes de esta base de datos.

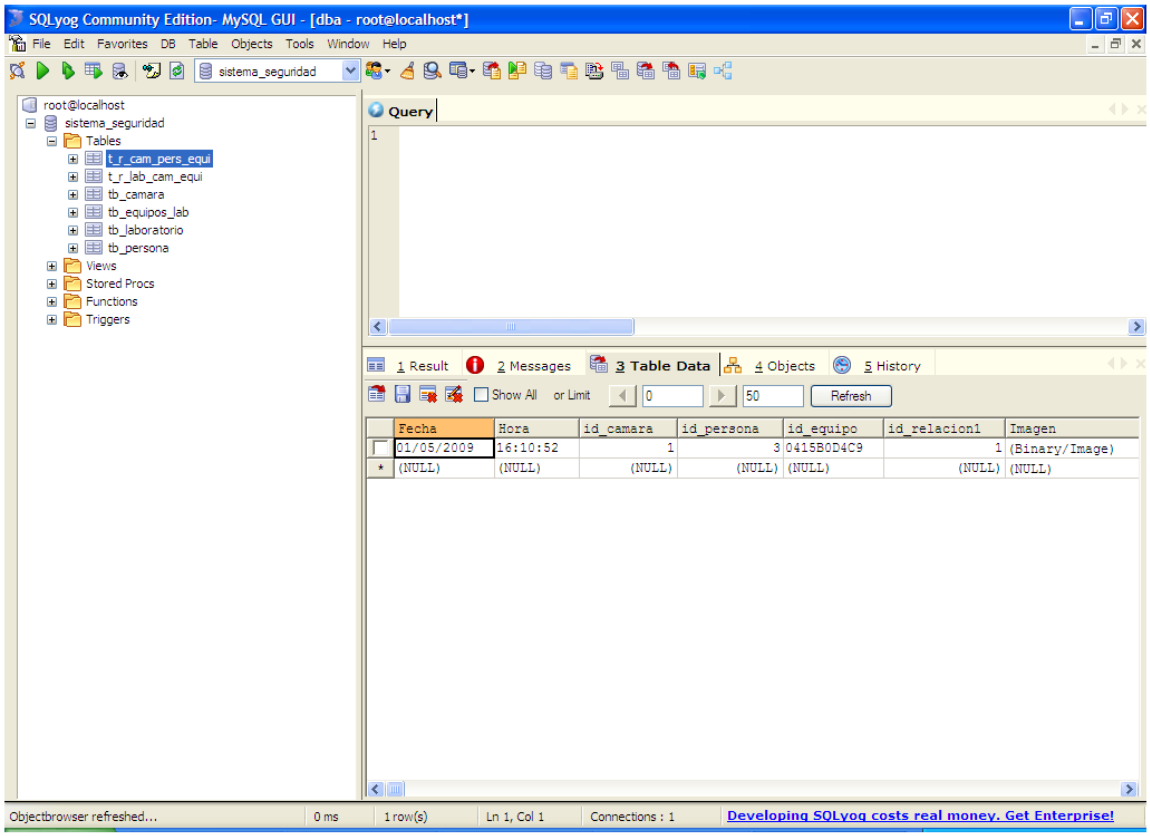


Figura 3.25 Base de Datos en MySQL.

Como se puede observar en la figura, esta base de datos tiene varias tablas que permiten monitorear, de manera periódica, el sistema.

A continuación presentamos las tablas más importantes de la base de datos creada, que se utilizan en la implementación del proyecto en cuestión.

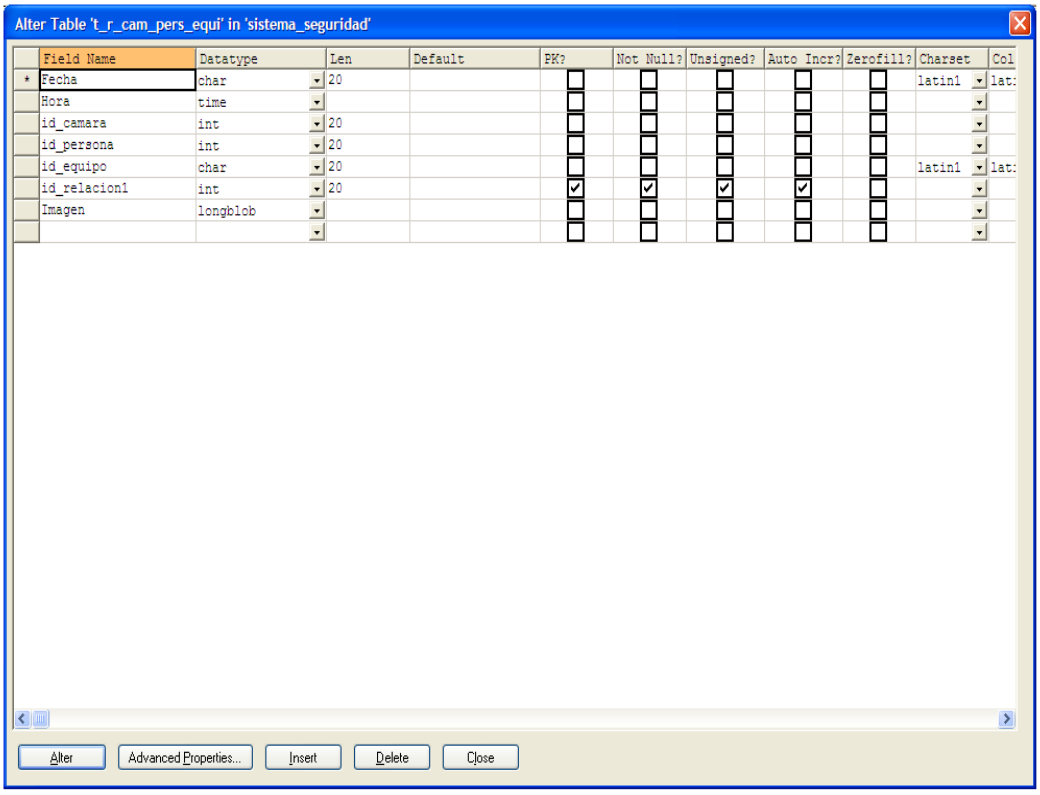


Figura 3.26 Campos de la Base de datos.

La primera tabla que vamos a mencionar es la denominada “t_r_cam_pers_equi”. Esta tabla de relación proporciona información sobre la fecha en la cual el sistema adquirió una imagen especifica, el ID de la cámara que tomo la imagen, el ID de la persona responsable del laboratorio en el instante en que se adquirió la imagen, el ID del equipo, el ID de relación y finalmente el path de la imagen.

La siguiente Tabla utilizada es la denominada “t_r_lab_cam_equi”. Esta tabla de relación tiene siete campos. La fecha, la hora, el ID del laboratorio, el ID de la cámara, el ID del equipo, el ID de la relación y finalmente el path de la imagen.

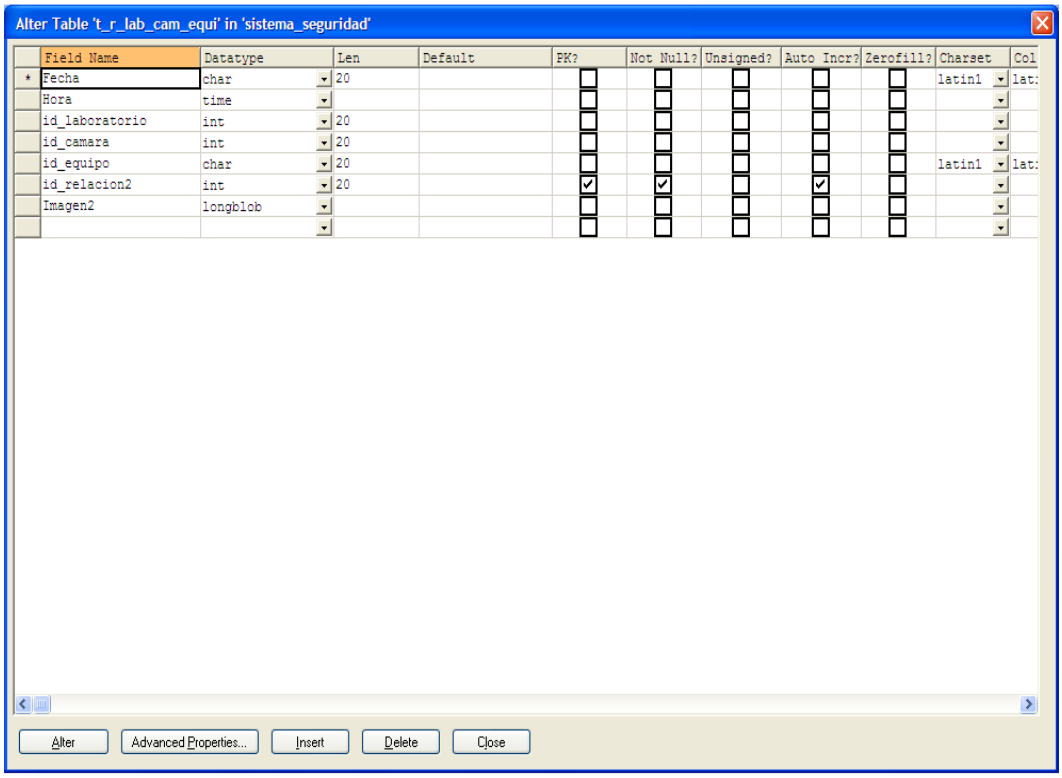


Figura 3.27 Campos en la Base de Datos.

Estas tablas se actualizan de manera periódica a medida que el sistema recibe información con la finalidad de permitirle al usuario monitorear constantemente el estado del sistema.

CAPÍTULO 4

FUNCIONAMIENTO DEL SISTEMA.

4.1 Implementación del Sistema.

Como se mencionó en el capítulo anterior el hardware del sistema esta conformado por tres bloques, que son el bloque del sensor RFID, el bloque microcontrolador y el bloque de comunicación ethernet. Estos tres bloques están integrados en el PCB mostrado anteriormente.

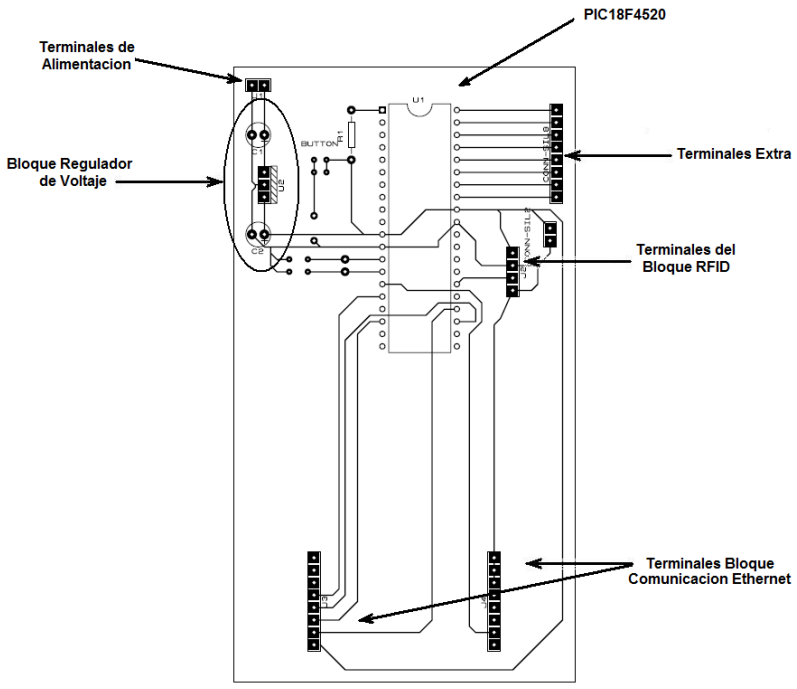


Figura 4.1 Circuito Impreso del Sistema.

La instalación del sistema es sumamente fácil, simplemente se requiere colocar el sensor RFID en el puerto indicado específicamente para este uso, es decir, en las terminales del bloque RFID mostrados en la figura anterior. De igual manera se debe colocar el módulo ET-MINI ENC28J60 en las terminales respectivas y alimentar el circuito utilizando las terminales de alimentación del PCB. Una vez montado el hardware como se ha indicado, se debe acercar la etiqueta RFID al área de cobertura del lector RFID, para que este pueda identificar al tag y el sistema pueda empezar la adquisición de los datos de manera inmediata.

Cabe recalcar que el procedimiento descrito se aplica para la implementación del prototipo desarrollado a menor escala. Si se desea implementar el hardware del sistema a escala real será necesario insertar un lector RFID de mayor potencia, preferiblemente de tipo activo, en las terminales del bloque RFID del PCB. De esta manera se proporcionara al sistema mayor cobertura y, según el lector RFID utilizado, capacidad de manejo multi-tag.

Una vez que se ha implementado el sistema como se ha acabado de describir, se debe colocar todo el circuito PCB en un lugar estratégico para garantizar la correcta y oportuna recepción de la señal proveniente de las etiquetas adheridas a los equipos del laboratorio.

4.2 Manejo y Prueba del Sistema.

Una vez garantizada la correcta instalación del dispositivo RFID, se requerirá el manejo de la interfaz desarrollada en LabVIEW para la adquisición y monitoreo de los datos adquiridos por el sistema. Como se mencionó en el capítulo anterior, la interfaz desarrollada en el panel frontal de LabVIEW esta conformada por cinco bloques que solicitan y presentan la información adquirida por el sistema de seguridad desarrollado. Pero antes de empezar la adquisición de datos desde LabVIEW es recomendable garantizar que exista comunicación ethernet entre el sistema de seguridad y el computador personal, para ello, se recomienda hacer una prueba de conectividad utilizando el comando PING del DOS. Para ello, tan solo es necesario abrir el command prompt de Windows y digitar el comando ping y luego la dirección IP asignada al dispositivo de seguridad.

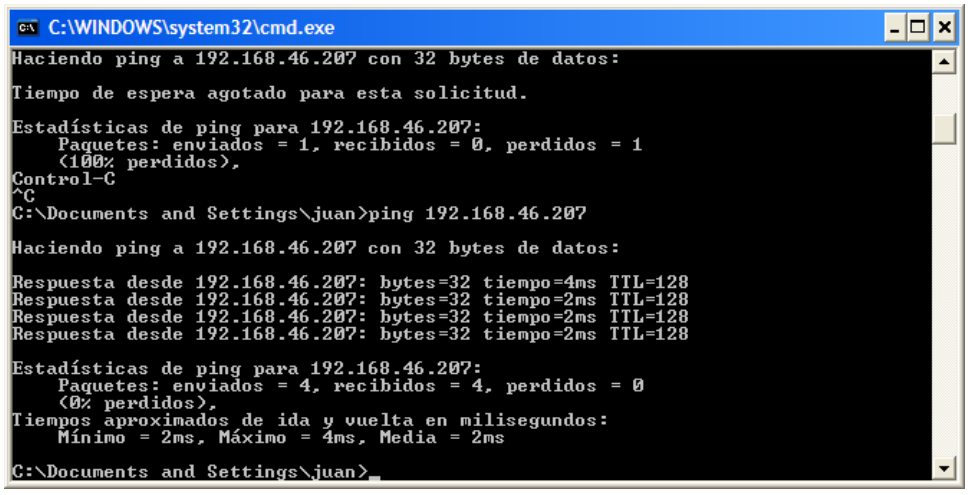


Figura 4.2 Prueba de Conectividad.

Si hay respuesta al PING por parte del sistema de seguridad, entonces tendremos la certeza de que el módulo ethernet esta funcionando apropiadamente y que por lo tanto hay conectividad entre el computador del administrador y el dispositivo de seguridad. Luego de haber culminado la etapa de prueba de conectividad, pasamos al manejo de la interfaz desarrollada en LabVIEW. Esta interfaz requiere el ingreso de distintos parámetros por parte del usuario. Entre ellos, los más indispensables son el puerto a través del cual se va a iniciar la sesión UDP por parte de LabVIEW.

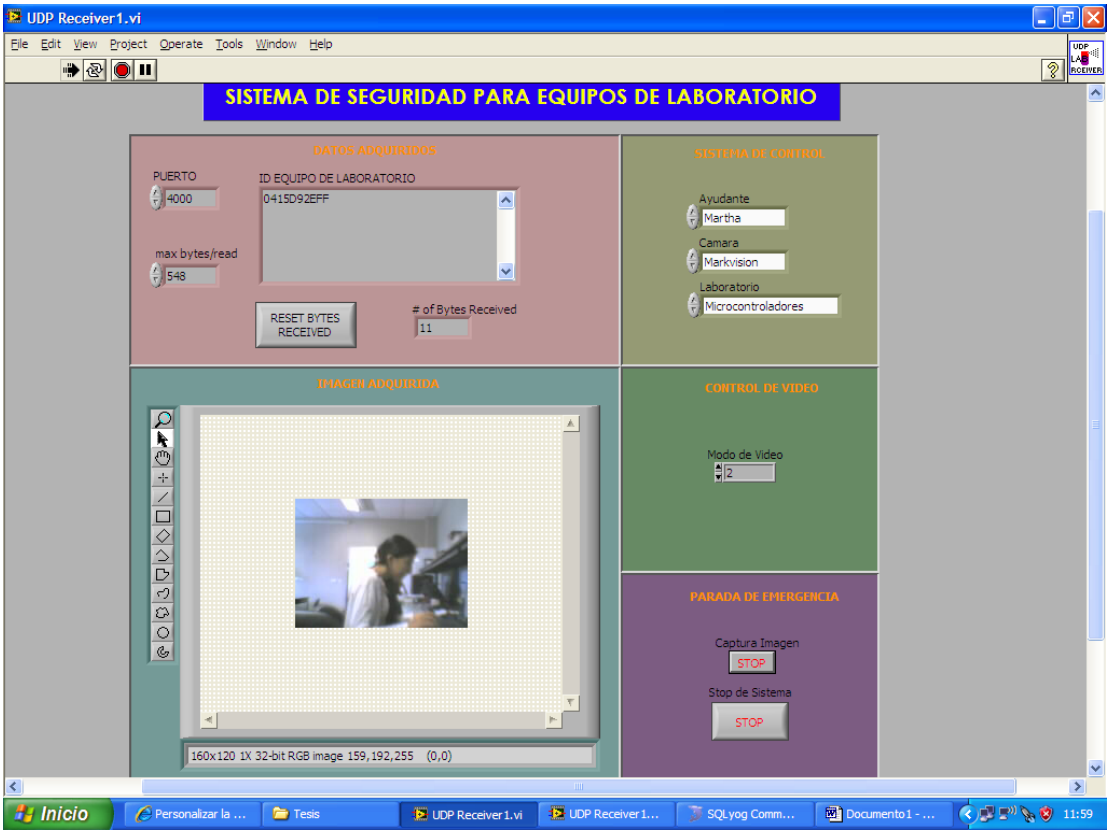


Figura 4.3 Puesta a prueba del Sistema.

También se requiere ingresar el nombre del ayudante que se encuentra administrando el laboratorio en aquel momento, el nombre de la cámara utilizada en ese momento específico, y el laboratorio que se esta resguardando. Además en el bloque de control de video se debe ingresar el modo de video que el usuario desea utilizar.

Una vez que se han ingresado estos parámetros, el sistema monitoreará de forma periódica la presencia de una señal proveniente de una etiqueta RFID.

Al ingresar el tag en el área de cobertura del dispositivo de seguridad, el ID de la etiqueta aparecerá de manera automática en la ventana del bloque denominado “datos adquiridos” del panel frontal, y la cámara tomará una imagen del suceso inmediatamente después de adquirir el ID de la etiqueta. Esta imagen se muestra en la pantalla del bloque denominado “imagen adquirida” del panel frontal. De esta manera el administrador del laboratorio podrá saber que equipo puede estar en riesgo, en que momento y gracias a la imagen adquirida por la webcam, el posible sospechoso de un potencial robo de los equipos del laboratorio.

Además de lo descrito anteriormente, el sistema activa una señal de alerta, de tal manera que el administrador pueda caer en cuenta de la posible sustracción de un equipo de las inmediaciones del laboratorio.

El sistema también permite llevar un registro de los eventos suscitados utilizando un software de administración de bases de datos.

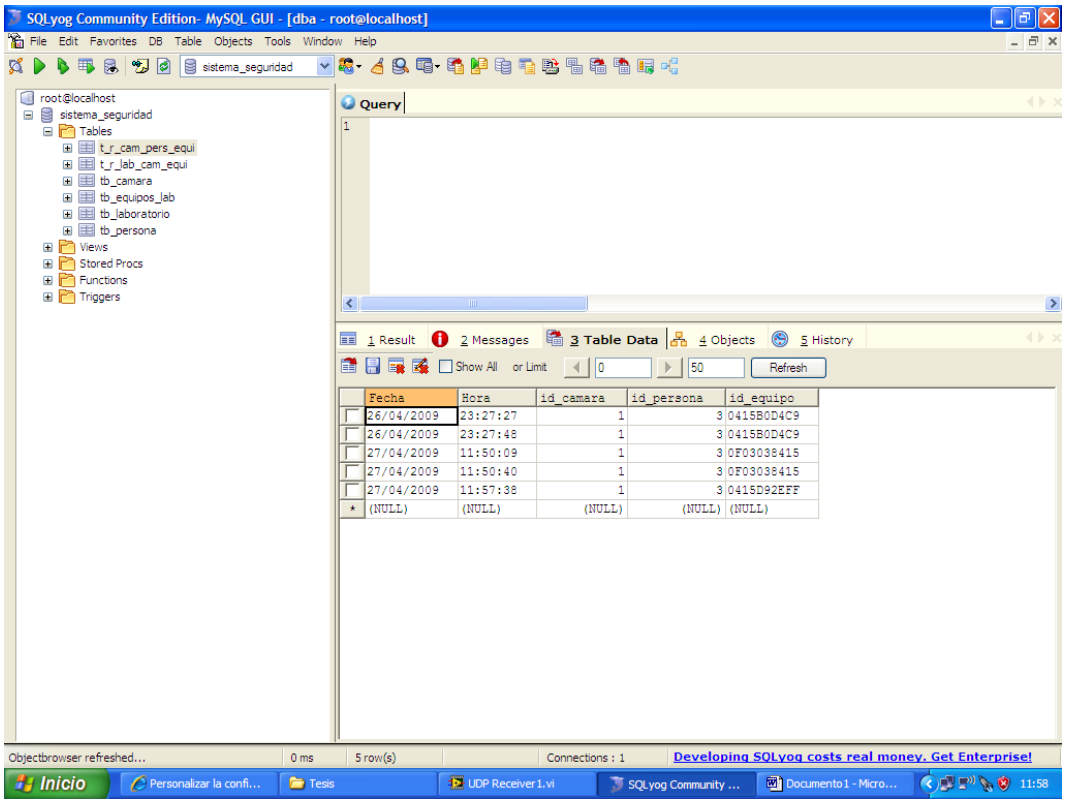


Figura 4.4 Base de datos del Sistema.

Esto es posible gracias a la utilización de las librerías database de LabVIEW que permiten iniciar una conexión con MySQL. Como se puede apreciar en la figura, el sistema registra cualquier robo potencial que se haya suscitado y lo almacena en una tabla de MySQL, de esta manera es posible llevar un control adecuado de los equipos del laboratorio y del personal que administra el mismo en cualquier instante.

CONCLUSIONES Y RECOMENDACIONES:

1. El Sistema proporciona una completa seguridad tanto proactiva como reactiva, para cada uno de los equipos del laboratorio, tanto de día como de noche. Esta es una de las principales diferencias respecto a otros sistemas, ya que el módulo RFDI puede trabajar al 100% de sus posibilidades en horario diurno, cuando los usuarios están utilizando sus equipos del laboratorio.

2. El sistema puede complementarse e integrarse con los sistemas de seguridad perimetrales actuales, como las cámaras de seguridad o Sistemas CCTV, sensores, Sistemas de control de acceso, entre otros; aportando una inmejorable prevención de los posibles riesgos que pueden padecer los equipos del laboratorio.

3. El hardware del sistema de seguridad para los equipos de Laboratorio, es pequeño y se lo puede colocar en un lugar estratégico, de tal manera, que ningún individuo que entre al laboratorio se percate de dicho dispositivo y así no pueda burlar la seguridad que se presta a los equipos.

4. Para el manejo del sistema de seguridad se ha utilizado el software Labview 8.5 que permite al usuario tener una interfase amistosa de tipo gráfica con el cual se puede tener un mayor control de los equipos que se encuentran dentro del laboratorio.

5. Es necesario que el administrador o computadora principal, la cual va a estar monitoreando la seguridad para los equipos dentro del laboratorio siempre tendrá que estar encendida, y con el panel frontal de Labview 8.5 abierta; siempre y cuando se quiera hacer una inspección de lo que sucede con los equipos del laboratorio, desde otro lugar que no sea este; es decir, se puede tener un control remoto desde otro computador utilizando la WEB.

6. Se recomienda conocer el código o tag que se va a emplear para la seguridad de cualquier equipo dentro del laboratorio debido a que dicho código tiene que estar registrado dentro de la base de datos, describiendo un equipo, al cual se quiere proteger; para que de esta forma no corran el riesgo de que hurten dicho equipo.

7. Es recomendable que las etiquetas electrónicas o tags, que se utilicen para la seguridad de los equipos del laboratorio, se encuentren en lugares no visibles para los usuarios, ya que podrían manipularlos y desprenderlos del equipo al cual prestan seguridad, y si esto ocurriera, el sistema no podrá garantizar la estadía del equipo en las inmediaciones del laboratorio.

8. Existe el riesgo de que se modifique fraudulentamente la información contenida en la etiqueta RFID mediante dispositivos portátiles, como PDAs o similares; esto podría suceder si se utilizan tarjetas electrónicas RFID activas, que tienen la capacidad de cambiar su identificación.

ÍNDICE DE FIGURAS:

CAPÍTULO 1.

CAPÍTULO 2.

CAPÍTULO 3.

CAPÍTULO 4.

ÍNDICE DE TABLAS:

CAPÍTULO 2.

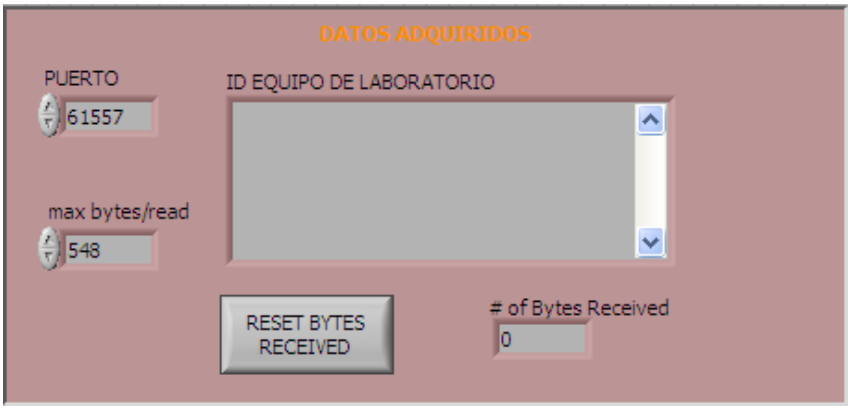
ANEXO A.

MANUAL DEL USUARIO.

El Sistema de seguridad de equipos de laboratorio propuesto en este proyecto requiere la utilización del software de instrumentación virtual LabVIEW y del software de administración de bases de datos MySQL. El usuario deberá interactuar constantemente con estos programas para poder monitorear el estado del sistema.

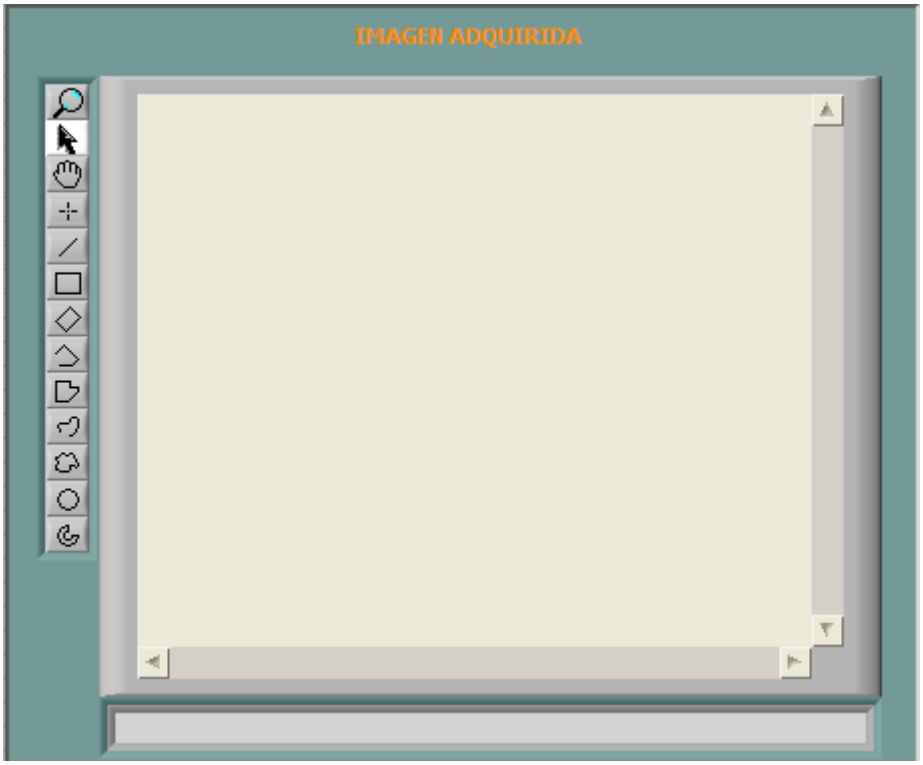
El Panel Frontal implementado en LabVIEW le permitirá al usuario indicarle al sistema los distintos parámetros necesarios para el correcto desempeño del mismo. Como se puede apreciar en la figura 3.23 presentada en el tercer capítulo de este texto, y reproducida nuevamente en la parte inferior de esta página, el panel frontal se subdivide en cinco secciones.

El primer panel, denominado “Datos Adquiridos”, proporciona el ID de la etiqueta cuya señal es recibida en el lector RFID. En este panel se debe ingresar el número de puerto y el número máximo de bytes y el botón “RESET BYTES RECEIVED” permite reiniciar los bytes recibidos.



El siguiente panel es el denominado “Sistema de control”. En este panel el usuario deberá ingresar el nombre del ayudante o persona responsable del laboratorio durante el funcionamiento del sistema, también la cámara utilizada para la adquisición de las imágenes y el laboratorio en el cual está operando el sistema.

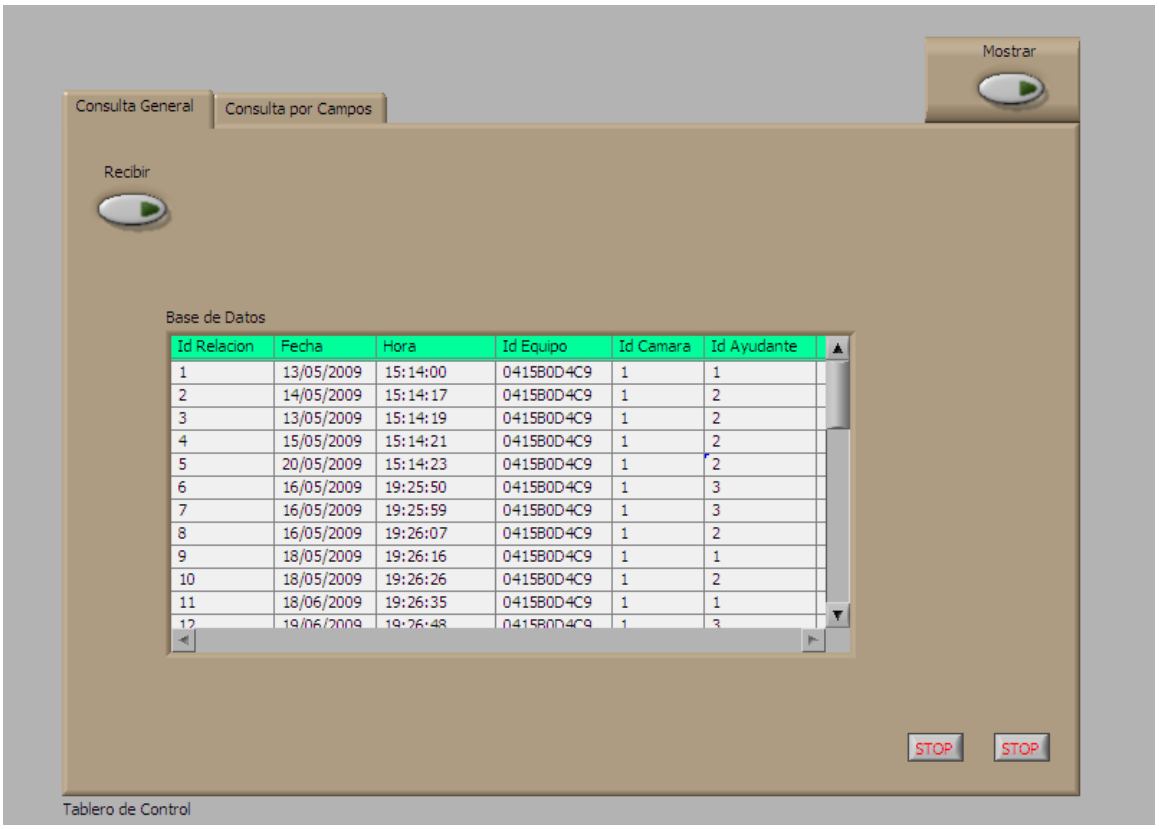
El siguiente es el panel “Imagen Adquirida” que presenta la imagen obtenida por la cámara especificada en el panel “sistema de control”. Este panel presenta un pequeño menú desplegado en la parte izquierda que le permite al usuario personalizar la imagen adquirida.



El panel “Control de video” posee una opción, en la cual el usuario podrá ingresar el modo de video deseado. Y finalmente el panel “Parada de emergencia” le permite al usuario detener el funcionamiento del sistema en caso de un mal funcionamiento del sistema o en caso de requerirlo el usuario.

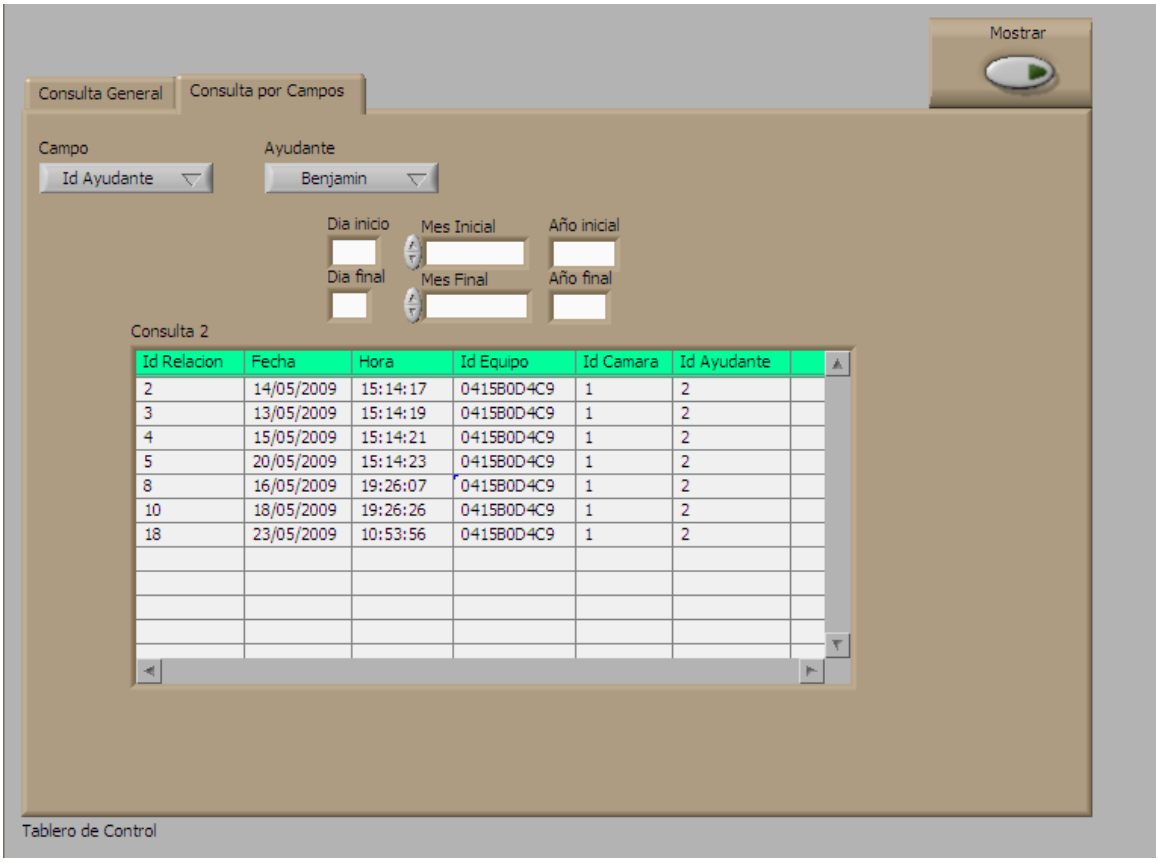
Finalmente, el sistema enviará la información adquirida de manera automática a una base de datos creada en MySQL que permitirá llevar un registro continuo de los datos recopilados por el sistema cuando se encuentre en operación. Esta información podrá ser consultada por el usuario a través de cualquier host de la red o del computador del administrador del laboratorio.

Además el sistema de seguridad para los equipos de laboratorio cuenta con un sistema de consultas, el cual es fácil de usar, ya que el ayudante o persona encargada que necesita un reporte de lo que sucede dentro del laboratorio lo único que tiene que hacer, es presionar el botón que dice: “Recibir”, el cual carga los datos que se encuentran dentro de la base de datos y luego “Mostrar” para que de esta manera se pueda observar los datos dentro de una tabla de consulta.



Id Relacion	Fecha	Hora	Id Equipo	Id Camara	Id Ayudante
1	13/05/2009	15:14:00	0415B0D4C9	1	1
2	14/05/2009	15:14:17	0415B0D4C9	1	2
3	13/05/2009	15:14:19	0415B0D4C9	1	2
4	15/05/2009	15:14:21	0415B0D4C9	1	2
5	20/05/2009	15:14:23	0415B0D4C9	1	2
6	16/05/2009	19:25:50	0415B0D4C9	1	3
7	16/05/2009	19:25:59	0415B0D4C9	1	3
8	16/05/2009	19:26:07	0415B0D4C9	1	2
9	18/05/2009	19:26:16	0415B0D4C9	1	1
10	18/05/2009	19:26:26	0415B0D4C9	1	2
11	18/06/2009	19:26:35	0415B0D4C9	1	1
12	19/06/2009	19:26:48	0415B0D4C9	1	3

Y a su vez si se quiere una consulta por campos, puede presionar la pestaña, “Consulta por Campos”, para que el usuario pueda revisar los datos por fecha, o por el ayudante que se encuentra encargado de la seguridad dentro del laboratorio



ANEXO B.

PIC 18F4520.

www.microchip.com

`

ANEXO C.

LM 7805.

www.fairchildsemi.com

BIBLIOGRAFÍA:

1. Parallax, Inc. [en línea] <www.parallax.com>. [Consulta: 17 de abril 2009].

2. Manual del Módulo Parallax, Inc. RFID Reader Module (#28140). Versión1.1, febrero 2006.

3. Kimaldi – Identificación y Biometría [en línea] <www.kimaldi.com> [Consulta: 18 de abril 2009].

4. Iberhardware, Soluciones Informáticas [en línea] <www.iberhardware.es>. [Consulta: 18 de abril 2009].

5. Fichet – Bauche [en linea] <www.fichet.es>. [Consulta: 19 de abril 2009].

6. Wikipedia, enciclopedia libre [en línea] <www.wikipedia.org> [Consulta: 19 de abril 2009].

7. MicroPic Servicios Profesionales [en línea], <www.micropic.es>. [Consulta: 21 de abril 2009].

8. Microchip Technology Inc. [en línea], www.microchip.com. [Consulta: 22 de abril 2009].

9. Etteam Co., LTD. [en línea]. < www.ett.co.th > [Consulta: 2 de mayo 2009].

10. Manual del Módulo ET-MINI ENC28J60 (Ethernet Controller). Versión 1, septiembre 2007.

11. Manual del PIC18F4520. Versión 1, 2004.

12. Fairchild Semiconductor Corporation. [en línea]. <www.fairchildsemi.com> [Consulta: 4 de mayo 2009].

13. Manual del LM7805. Versión diciembre 2005.

14. National Instruments [en línea] <www.ni.com>. [Consulta: 18 de abril 2009].