

**UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA MECÁNICA**



**DESARROLLO, IMPLEMENTACIÓN Y PUESTA EN
OPERACIÓN DE UN SISTEMA DE ADQUISICIÓN,
REGISTRO Y MONITOREO DE DATOS DE TERMINALES
DE PESAJE METTLER – TOLEDO MODELO ID3S**

INFORME DE SUFICIENCIA

**PARA OPTAR EL TÍTULO PROFESIONAL DE:
INGENIERO MECATRÓNICO**

**ELABORADO POR:
PAUL RAFAEL ARRIZ TISOC**

PROMOCIÓN 2007 - II

LIMA - PERÚ

2014

DEDICATORIA

A mis padres, por su constante e incondicional apoyo.

AGRADECIMIENTOS

A mis profesores de la Universidad Nacional de Ingeniería por todas sus enseñanzas y conocimientos impartidos.

ÍNDICE

PRÓLOGO	1
CAPÍTULO I	
INTRODUCCIÓN	3
1.1. Antecedentes.....	3
1.2. Definición del problema	4
1.3. Objetivos.....	5
1.3.1. Objetivo general.....	5
1.3.2. Objetivos específicos	5
1.4. Justificación	6
1.5. Alcances	6
1.6. Limitaciones.....	7
1.7. Viabilidad.....	7
1.7.1. Viabilidad técnica	7
1.7.2. Viabilidad económica	8
1.7.3. Viabilidad operativa.....	9
1.8. Alternativas.....	10

CAPÍTULO II

FUNDAMENTO TEÓRICO Y FORMULACIÓN DEL PRODUCTO.....	11
2.1. Fundamento teórico.....	11
2.1.1. Modelo TCP/IP.....	11
2.1.2. RS-232.....	30
2.1.3. Bases de datos	32
2.2. Formulación del producto	35

CAPÍTULO III

DESARROLLO DEL SISTEMA	37
3.1. Comunicación con los terminales de pesaje	37
3.1.1. Convertidor Ethernet-serial	38
3.1.2. Formato de tramas de mando y respuesta.....	39
3.1.3. Conexión de datos hacia los terminales de pesaje.....	40
3.1.4. Lectura de pesos de los terminales de pesaje.....	41
3.1.5. Identificación de los pesajes correctos e incorrectos.....	42
3.2. Registro en base de datos	47
3.2.1. Campos de la tabla en la base de datos	47
3.2.2. Escritura de pesos en la base de datos.....	49
3.2.3. Lectura de pesos desde la base de datos	51
3.2.4. Agregar observaciones a los pesos	51
3.2.5. Eliminación de pesos en la base de datos	53
3.3. Diseño de la interfaz gráfica.....	55
3.3.1 Panel principal de la aplicación	55
3.3.2 Pantalla de configuración.....	57
3.3.3 Pantalla de generación de reportes.....	60

3.3.4	Manipulación de datos en las tablas	60
-------	---	----

CAPÍTULO IV

IMPLEMENTACIÓN DEL SISTEMA.....	63
4.1. Implementación de los algoritmos.....	63
4.1.1. Proyecto en Labview.....	65
4.1.2. Inicialización de parámetros.....	65
4.1.3. Conexión de datos hacia los terminales de pesaje.....	65
4.1.4. Lectura de pesos desde los terminales de pesaje.....	70
4.1.5. Generación de reportes	71
4.1.6. Generación de mensajes y sonidos de alerta.....	71
4.1.7. Configuración del sistema.....	73
4.1.8. Grabar datos en base de datos	78
4.1.9. Actualización de datos en las tablas	80
4.1.10. Lectura de pesos desde la base de datos.....	83
4.1.11. Reacción a eventos de teclado en las tablas.....	84
4.1.12. Clasificación de pesajes.....	86
4.2. Validación de algoritmos por simulación	91

CAPÍTULO V

PUESTA EN OPERACIÓN DEL SISTEMA EN UNA PLANTA DE ENVASADO	92
5.1. Descripción de la planta de envasado	92
5.2. Instalación del sistema.....	93
5.2.1. Instalación de hardware	93
5.2.2. Instalación de software	96

5.3. Puesta en operación	97
5.3.1. Versión preliminar	97
5.3.2. Levantamiento de observaciones del cliente	97
5.3.3. Lanzamiento final	98
CONCLUSIONES	99
RECOMENDACIONES	100
BIBLIOGRAFÍA	101
ANEXOS	102

PRÓLOGO

El presente Informe detalla el desarrollo de un sistema de adquisición, registro y monitoreo de datos para terminales de pesaje, el cual permite con un bajo costo aumentar la eficiencia del área de envase de la planta objetivo, reduciendo las pérdidas en tiempo y en producto causadas por los errores en el pesaje por parte de los operarios del área de envase, lo cual no sólo tiene un impacto en la eficiencia de esta área, sino que además puede dañar la confianza del cliente que recibe menos producto del que se indica en el envase. Este sistema permite además registrar toda la información en la base de datos de la empresa y generar reportes en base a estos registros en formatos amigables para el usuario.

El informe consta de cinco capítulos tratando los siguientes temas:

El primer capítulo corresponde a la introducción del informe, la cual está compuesta por los antecedentes, la definición del problema, los objetivos, la justificación, las limitaciones, la viabilidad de solución y las alternativas.

El segundo capítulo comprende una descripción del producto formulado para la solución del problema propuesto, así como una presentación concisa del fundamento teórico utilizado.

El tercer capítulo corresponda al desarrollo del sistema, en el cual se desarrollan los principales algoritmos para la comunicación con los terminales de pesaje, el registro en la base de datos, así como el diseño de la interfaz gráfica.

El cuarto capítulo contiene la implementación de los algoritmos de software en la plataforma de desarrollo seleccionada.

El quinto capítulo refiere a la puesta en operación de este producto en una planta de envase, aquí se describe la planta de envase, la instalación del sistema, y la propia puesta en operación.

Finalmente, se presentan las conclusiones y recomendaciones de este informe, así como la bibliografía y anexos.

En conclusión, el presente informe muestra como el sistema propuesto provee una solución al problema descrito.

CAPÍTULO I

INTRODUCCIÓN

1.1 ANTECEDENTES

El producto del presente trabajo es un sistema de adquisición, registro y monitoreo de datos de terminales de pesaje de la marca Mettler-Toledo, modelo ID3S.

El cliente para el cual se desarrolló este producto es una empresa de productos alimenticios. Más específicamente, este sistema se instaló en el área de envase de la planta industrial de esta empresa. Esta área de envase cuenta con un bajo nivel de automatismo; el personal recoge manualmente los empaques del producto desde una faja transportadora y los introduce en sacos hasta llegar al peso adecuado en cada saco, momento en el que estos son sellados. Los sacos son finalmente colocados manualmente en un sistema que los envía a otra área a fin de ser almacenados para posteriormente ser despachados al cliente.

El personal de la planta de envase identificó distintas oportunidades de mejora en este proceso. Algunas de estas oportunidades tenían como causa raíz los errores humanos durante el proceso de pesaje de los sacos del producto. El producto del presente informe fue desarrollado en respuesta a esta problemática.

1.2 DEFINICIÓN DEL PROBLEMA

La problemática que motivó la implementación del proyecto tratado en el presente informe viene definido por los errores en el peso final de los sacos debido a una mayor o menor cantidad de producto en estos. Este problema se vuelve relevante debido a que estos errores causan reprocesamiento ya que estos sacos deben regresar nuevamente al área de envase luego de no pasar el control de calidad, o aun peor, llegan a manos del cliente el cual puede presentar un reclamo por el producto no conforme con las especificaciones.

La empresa cliente identificó la necesidad de implementar un sistema que permita mitigar el impacto o probabilidad del problema previamente definido; lamentablemente, desde un inicio dejó claro que solo contaba con un presupuesto muy limitado asignado para la implementación de esta mejora, el cual hacía inviable la automatización del proceso de envase. Esta limitación económica redujo drásticamente el espectro de posibles soluciones al problema y motivó finalmente que la empresa cliente considere como solución el diseño de un sistema que permita detectar los sacos que no estén dentro de especificaciones, en base a los pesos obtenidos en los terminales de pesaje al momento del envase; así como el registro de esta información en la base de datos de la empresa, para tener una mejor registro de estos productos no conformes, teniendo en cuenta que no todos los sacos son pesados nuevamente durante el control de calidad. Esto permitía aprovechar la capacidad de comunicación serial RS-232 que tienen estos terminales de pesaje y la plataforma de base de datos ya existente en la planta, reduciendo así los costos.

1.3 OBJETIVOS

1.3.1. Objetivo General

Desarrollar e implementar un sistema de adquisición, registro y monitoreo de datos de terminales de pesaje de la marca Mettler-Toledo, modelo ID3S.

1.3.2. Objetivos específicos

Los objetivos específicos a tratar en el presente informe desarrollado son los siguientes:

- Desarrollar el algoritmo de comunicación entre la base de datos corporativa y la aplicación cliente.
- Desarrollar el algoritmo de comunicación entre la aplicación cliente y los terminales de pesaje.
- Desarrollar una interfaz gráfica conforme a las especificaciones del cliente.
- Implementar los algoritmos diseñados.
- Estimar el costo necesario para el desarrollo del sistema.
- Describir la puesta en operación realizada para este sistema en una planta de envase.
- Un objetivo interno, por parte de la empresa que contrató a quien escribe este informe, fue el exceder las expectativas de la empresa cliente debido a que este sistema fue el primer producto que se le proveía, y se habían ya identificado varias soluciones potenciales para otros problemas en esta planta las cuales representaban potenciales proyectos. Un resultado sobresaliente podría motivar la aprobación de otros proyectos adicionales.

- RIP (Routing Information Protocol): Usa una métrica basada en saltos (rutas corta, 16 saltos). Utiliza datagramas UDP para pasar la información entre routers
- HELLO: usa una métrica basada en retardos (rutas largas)
- BGP(Border Gateway Protocol)
- EGP(Exterior Border Protocol)

2.1.1.7 Protocolo ICMP

Tiene como misión comunicar los mensajes de error, solicitudes y respuestas entre nodos de la red. Este protocolo se define en el RFC 792. Los mensajes son enviados dentro de un datagrama IP.

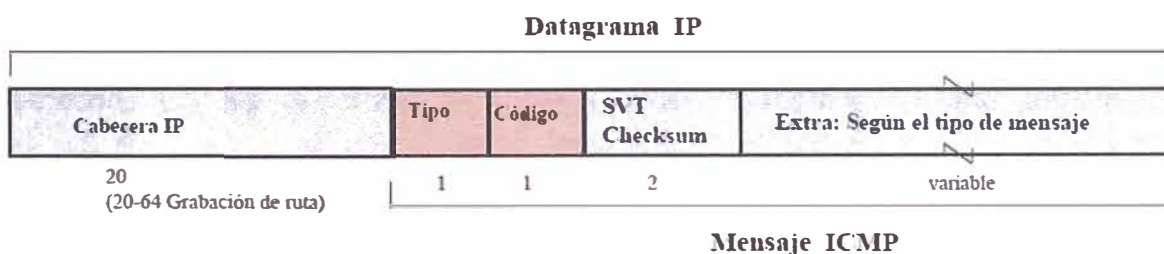


Figura 2.8 – Mensaje ICMP en un datagrama IP

Existen dos tipos de mensajes:

- Solicitud ICMP

Se solicita o se informa de un acontecimiento concreto que no es causa de ningún error. Ejemplo: el envío de un eco, o la presencia de un router.

- Notificación de error ICMP

1.4 JUSTIFICACIÓN

La justificación económica del desarrollo de este sistema está dado por los costos del producto no conforme proveniente de los errores de pesaje en los sacos del producto.

Adicionalmente, la justificación académica se encuentra en la aplicación de herramientas informáticas que forman parte del contenido curricular de pregrado de la especialidad de ingeniería mecatrónica, así como el aprendizaje de otras herramientas relacionadas a la especialidad.

Por otro lado, el aporte tecnológico viene dado por el desarrollo de una solución personalizada de este tipo localmente, proveyendo un pequeño pero sincero aporte al desarrollo local de tecnología, fomentando el desarrollo de productos tecnológicos en nuestro país.

Así mismo, el aporte productivo está constituido por la disminución del tiempo de reprocesamiento en el área de envase causado por los errores de pesaje cometidos por los operarios. Esta disminución de reprocesamiento permite un aumento de la productividad del área de envase y por ende potencialmente permite también una mejora en la eficiencia del proceso productivo.

Los aportes mencionados previamente justifican el valor del presente informe así como del producto aquí desarrollado.

1.5 ALCANCES

Los alcances del proyecto son los siguientes:

- Se formulará el producto requerido para la solución del problema
- Se desarrollará un diagrama de flujo del algoritmo de comunicación entre la aplicación cliente y la base de datos corporativa.

- Se desarrollará un diagrama de flujo del algoritmo de comunicación entre la aplicación cliente y las balanzas.
Se desarrollará una interfaz gráfica conforme a las especificaciones del cliente.
- Se implementarán los algoritmos diseñados.
- Se estimarán los recursos necesarios para el desarrollo del sistema.
- Se realiza la instalación del sistema así como la puesta en marcha en la planta de envase del cliente.

1.6 LIMITACIONES

Se deberán tener en cuenta las siguientes limitaciones:

- El sistema fue diseñado para comunicar con un máximo de tres terminales de pesaje del modelo ya previamente indicado.
- El sistema fue diseñado para tener una sola estación de operación con la aplicación de software cliente. No es un sistema multiusuario.
- El sistema depende de la red Ethernet de la planta para el registro de datos en la base de datos. No se realizó una conexión dedicada entre el computador donde se instaló la aplicación cliente y el servidor de base de datos de la empresa cliente.

1.7 VIABILIDAD

1.7.1. Viabilidad técnica

Los siguientes recursos son necesarios para la implementación del producto:

- Software NI Labview 8.0 Professional de la empresa National Instruments
- Kit de herramientas de software para conexión a base de datos de NI Labview 8.0
- Un Computador proveído por la empresa cliente, conforme a las especificaciones indicadas en el presente informe.
- Licencia de Microsoft Excel suministrador por la empresa cliente.
- Conectividad de red a la base de datos corporativa de la empresa cliente.
- Un Conversor Ethernet-Serial de 4 puertos.
- 3 balanzas modelo ID3S de la empresa Mettler-Toledo.

Ya que se cuenta con acceso a todos estos recursos se considera el producto técnicamente viable.

El principal riesgo técnico del sistema provenía de las posibles complicaciones que pudieran aparecer al desarrollar la comunicación entre la aplicación cliente y los terminales de pesaje. Se redujo la probabilidad e impacto de este riesgo realizando pruebas en una etapa temprana del proyecto.

1.7.2. Viabilidad económica

Se evaluaron los costos del sistema. Sin embargo, la información cuantitativa de los beneficios obtenidos al evitar los errores en el peso de los sacos no fue compartida por el cliente. La falta de esta información impide calcular los beneficios económicos generados por el producto de este informe. Por este motivo, se considera el proyecto económicamente viable al haber sido aprobado por el cliente mediante las órdenes de compra respectivas.

Los costos del sistema se encuentran detallados en la siguiente tabla:

Tabla 1.1 – Costos de los componentes del sistema

Item	Recurso	cantidad	Costo unitario	Costo
1	Software Labview 8.0 Professional*	1	\$ -	\$ -
2	Kit de herramientas de software para conexión a base de datos de NI Labview*	1	\$ -	\$ -
3	Computador (existente)	1	\$ 1,250.00	\$ 1,250.00
4	Licencia de Microsoft Excel (existente)	1	\$ 387.00	\$ 387.00
5	Balanza Mettler toledo ID35 (existentes)	3	\$ 425.00	\$ 1,275.00
6	MOXA Nport 5400 - Conversor Ethernet-Serial	1	\$ 519.00	\$ 519.00
7	Switch Ethernet 8 puertos	1	\$ 117.50	\$ 117.50
8	Tablero para conversor Ethernet-Serial y switch de red (tercerizado)	1	\$ 480.00	\$ 480.00
9	Instalación de tablero, conduits y cableado (tercerizados)	1	\$ 1,850.00	\$ 1,850.00
10	Horas de ingeniería para desarrollo	160	\$ 9.00	\$ 1,440.00
			Total	\$ 7,318.50

*Los costos de los items 1 y 2 no son considerados ya que la empresa que desarrollo la solución cuenta con licencias gratuitas para estos dos items, esto debido a que es representante en el Perú de National Instruments, quien desarrolla estos dos softwares.

1.7.3. Viabilidad operativa

El sistema desarrollado puede entrar en operación sin afectar el proceso normal de pesaje de los sacos. Tampoco es requerido reiniciar o alterar de ninguna manera los terminales de pesaje al conectar el puerto serial de los terminales de pesaje al sistema.

Si bien el sistema fue desarrollado para tres balanzas, este puede ser fácilmente escalado a un mayor número de balanzas, para esto solo es necesario agregar otros conversores Ethernet-serial o modificar el actual, de 4 puertos, por uno de más puertos; así como también realizar las modificaciones respectivas en el software. Esto fue también conversado desde un inicio con el cliente ya que potencialmente se podría extender el sistema para todos los terminales de pesaje del área de envase del cliente.

1.8 Alternativas

La primera evidente solución al problema descrito en este informe es la automatización del proceso completo de envase; sin embargo, el cliente dejó claro desde un inicio que esta solución ya había sido evaluada y resultó no ser viable en el corto plazo.

La empresa cliente evaluó internamente distintas alternativas de solución al problema planteado y concluyó que un producto conforme al desarrollado en el presente informe era la mejor opción. Posteriormente comparó propuestas de distintas empresas para el desarrollo de esta solución y finalmente colocó la orden de compra que generó el desarrollo, implementación y puesta en marcha de este sistema.

CAPÍTULO II

FUNDAMENTO TEÓRICO Y FORMULACIÓN DEL PRODUCTO

2.1 FUNDAMENTO TEÓRICO

El fundamento teórico se describen las bases de los protocolos TCP/IP, del standard RS-232 y de bases de datos.

2.1.1 MODELO TCP/IP

El modelo TCP/IP es un modelo de descripción de protocolos de red desarrollado por Vinton Cerf y Robert E. Kahn en los años 70. El modelo consta de cuatro capas:

- Capa de aplicación
- Capa de transporte
- Capa de internet o capa de red
- Capa de acceso a la red o capa de enlace.

El origen de esta familia de protocolos fue la red ARPANET (en ella se desarrollaron los conceptos fundamentales de diseño y gestión de redes). Los niveles más bajos (enlace y físico) no están implementados ya que se diseñó para no depender de una red física concreta.

Las semejanzas entre los modelos OSI y TCP/IP se representan en la siguiente gráfica, el cual muestra la correlación que existe entre las capas de ambos modelos.

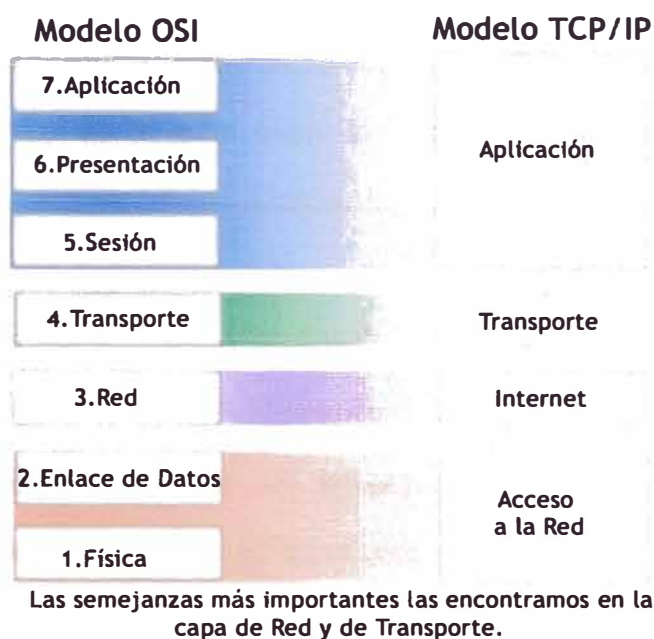


Figura 2.1 – Semejanzas entre modelo OSI y Modelo TCP/IP.

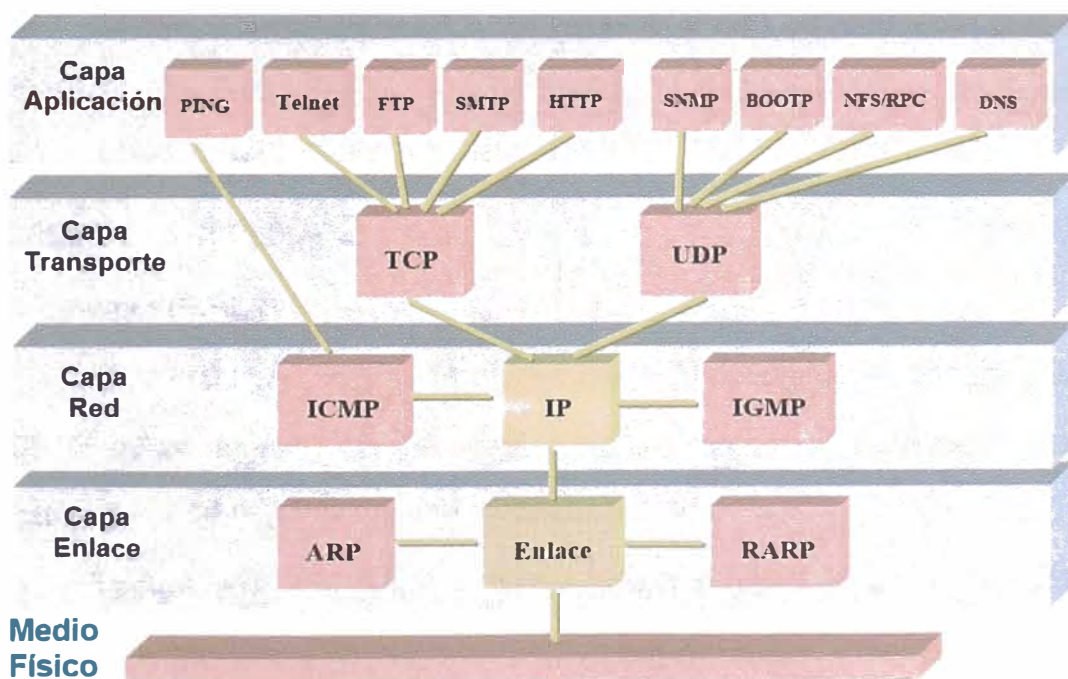


Figura 2.2 – Familia de protocolos TCP/IP.

2.1.1.1 Capa de acceso a la red o capa de enlace

Esta es la capa que maneja todos los aspectos que un paquete IP requiere para efectuar un enlace físico real con los medios de la red. Esta capa incluye los detalles de la tecnología LAN y WAN y todos los detalles de las capas físicas y de enlace de datos del modelo OSI.

Son funciones de esta capa: la asignación de direcciones IP a las direcciones físicas, el encapsulamiento de los paquetes IP en tramas. Basándose en el tipo de hardware y la interfaz de la red, la capa de acceso de red definirá la conexión con los medios físicos de la misma.

Los protocolos ARP (Address Resolution Protocol) y RARP (Reverse Address Resolution Protocol) se encargan de enlazar los sistemas de direccionamiento IP y el de la red física utilizada.

2.1.1.2 Nivel de enlace de datos LLC/MAC

TCP/IP no define el nivel de enlace sin embargo un caso muy usual es usual usar Ethernet. Existen básicamente los dos siguiente tipos de tramas dentro de Ethernet:

- IEEE 802.2/802.3(RFC 1042)
- Ethernet (RFC 894)

La utilización de una u otra trama depende del protocolo utilizado. Así, TCP/IP utiliza la encapsulación denominada Ethernet II, mientras que el protocolo IPX/SPX, utiliza la encapsulación 802.3

Tanto IEEE 802.2/802.3 como Ethernet utilizan 48 bits para la dirección de receptor y 48 bits para la dirección de emisor. Ethernet identifica el tipo de datos que prosiguen.

IEEE 802 subdivide la capa de enlace de datos en las subcapas LLC y MAC. LLC provee mecanismos de multiplexado que permiten a distintos protocolos de red (por ejemplo IP, IPX, Decnet) coexistir en una misma red multipunto y ser transportadas sobre el mismo medio. Por otro lado, MAC provee mecanismos de direccionamiento y control de acceso que permiten a distintos nodos de red comunicarse sobre una misma red.

El protocolo IEEE 802.2/802.3 identifica además la cantidad de bytes que siguen sin considerar el CRC, mientras que el tipo es identificado dentro del campo SNAP. El campo LLC está compuesto del punto de acceso de servicio destino (DSAP), punto de acceso de servicio fuente (SSAP) y del campo de control.

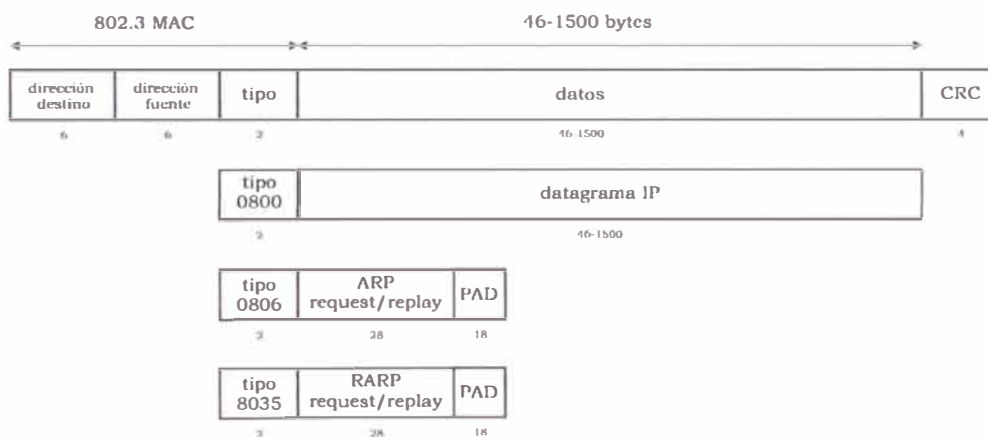


Figura 2.3 – Encapsulación Ethernet

2.1.1.1 MTU (Maximum Transmission Unit).

La unidad máxima de transmisión (MTU) depende del tipo de adaptador de red. Si un datagrama IP es mayor que el MTU, entonces el datagrama se fragmentará a nivel IP.

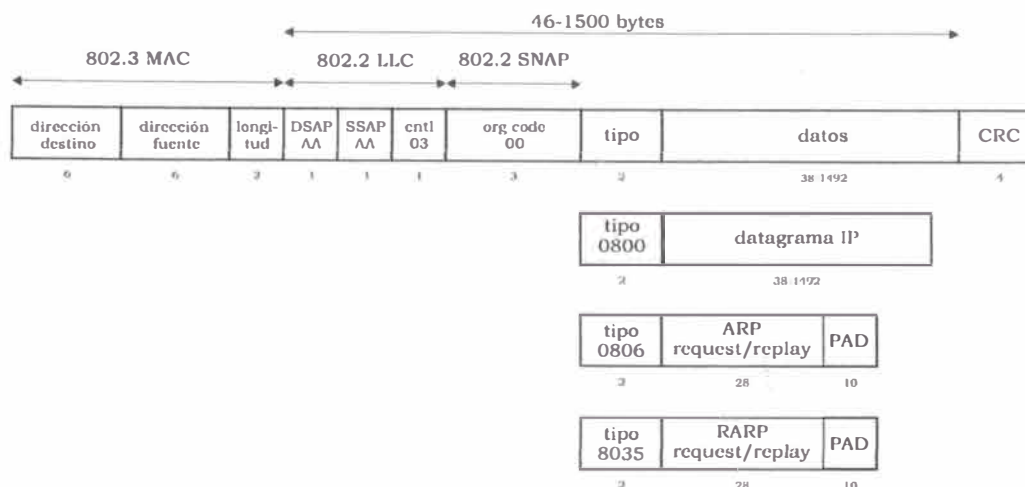


Figura 2.4 – Encapsulación IEEE 802.2/802.3 (RFC1042)

2.1.1.2 Direccionamiento MAC

Para que un datagrama IP alcance su destino es preciso conocer su dirección física. Esta dirección física es una combinación de 48 bits agrupados en 6 bytes. Los tres primeros bytes identifican al fabricante y los tres últimos bytes identifican la tarjeta. La dirección MAC es única e irrepetible y existe una dirección reservada que tiene todos los bits a 1 (FF:FF:FF:FF:FF:FF) la cual se utiliza para enviar una trama a todas las máquinas de la red (broadcast). Estas direcciones son únicas porque son escritas directamente en el hardware al momento de su fabricación.

2.1.1.3 Protocolo ARP

Para enviar un datagrama necesitamos también la dirección MAC de destino. El protocolo ARP (Address Resolution Protocol) obtiene la dirección MAC correspondiente a una dirección IP. Para esto envía una trama de 'Petición ARP' a todas las máquinas de la red (broadcast). El protocolo ARP

de la máquina destino verifica si el solicitante pide su dirección IP y emite una trama de 'Respuesta ARP' incorporando su dirección MAC e IP. Al recibir la respuesta se guarda el resultado en una tabla, pudiendo emitir a partir de entonces datagramas IP. ARP está documentado en el RFC826.

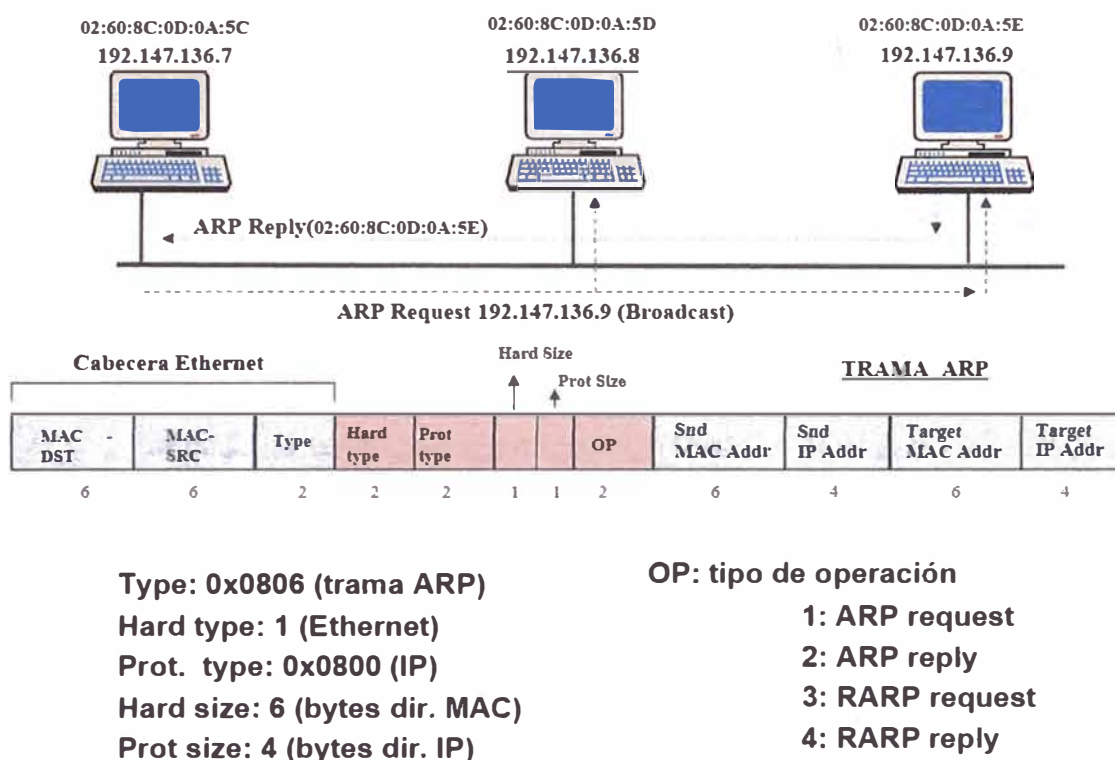


Figura 2.5 – Petición y respuesta ARP

Los datos se almacenan en una tabla dinámica. El plazo normal de expiración de una entrada es de 20 minutos después de su creación (depende del sistema operativo).

El Protocolo RARP (Reverse Address Resolution Protocol) realiza la operación inversa a ARP y se encuentra descrito en el RFC903. RARP ya no es usado, fue reemplazado por BOOTP (protocolo de red que es usado para

obtener una dirección IP de un servidor), el cual fue tiempo más tarde sustituido por el Protocolo de Configuración Dinámica de host (DHCP).

```

$ arp -a
Net to Media Table
Device      IP Address          Mask                Flags              Phys Addr
-----
hme0       193.147.136.216    255.255.255.255    00:60:08:64:39:07
hme0       193.147.136.1      255.255.255.255    00:e0:1e:3d:fe:83
hme0       192.168.2.20       255.255.255.255    00:a0:c9:78:75:bb
hme0       lorca               255.255.255.255    SP                08:00:20:a2:55:fe
hme0       BASE-ADDRESS.MCAST.NET 240.0.0.0          SM                01:00:5e:00:00:00

```

Figura 2.6 – Ejemplo de una tabla de dirección MAC e IP

2.1.1.4 Capa de internet o capa de red

La base de la familia de protocolos TCPIP es el nivel de Red IP. Es un protocolo de conmutación de paquetes muy sencillo, de tipo datagrama, de forma que se pueda implementar en cualquier tipo de máquina. A diferencia de OSI, no se distingue claramente entre servicio, interfaz y protocolo. Existen actualmente dos versiones: IPv4 e IPv6. Existen además los siguientes protocolos de apoyo:

- ICMP (Internet Control Message Protocol): encargado de la comunicación de mensajes entre nodos de la red
- IGMP (Internet Group Mangement Protocol): encargado del envío de mensajes a grupos de usuarios.

2.1.1.5 Protocolo IPv4

En el datagrama IP (RFC 791) el tamaño normal de la cabecera es de 20 bytes, a no ser que presente opciones. El tamaño máximo del datagrama se ajusta para que quepa en el MTU del nivel de enlace.

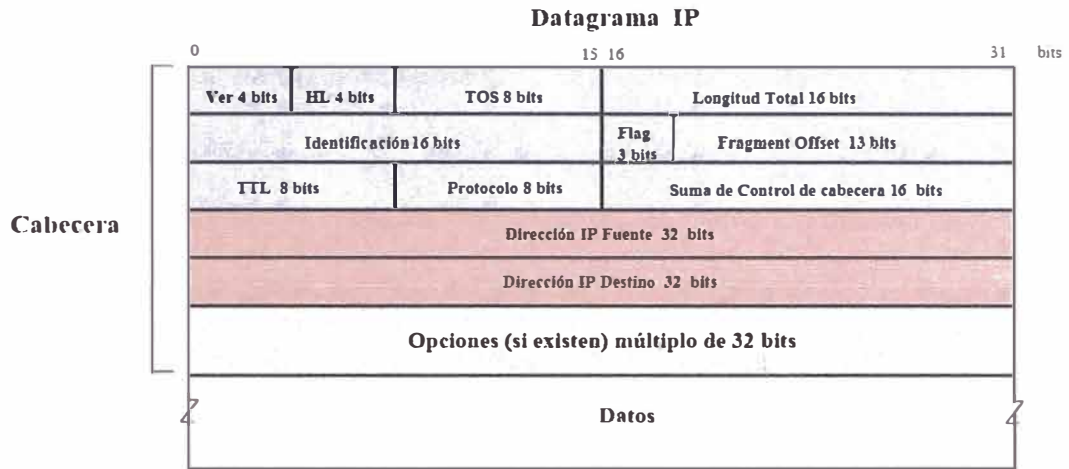


Figura 2.7 – Cabecera del datagrama IP

- 'Ver': Estos cuatro bits indican la versión
- HL: longitud de la cabecera en palabras (4 bytes), normalmente vale 5 (20 bytes) pero puede alcanzar las 15 palabras (60 bytes) cuando se utilizan las opciones de marcado de rutas.
- TOS: Tipo de servicio. Los tres primeros bits son ignorados, los 4 bits siguientes representan el TOS y el último vale siempre 0. Cada bit selecciona un tipo de comportamiento (solo uno puede estar activado). Si todos los bits están en cero significa un servicio normal:
 - Bit 1: Minimiza el Retardo
 - Bit 2: Minimiza el Caudal
 - Bit 3: Maximiza la fiabilidad
 - Bit 4: Minimiza Coste
- Longitud Total: tamaño en bytes del datagrama IP incluyendo la cabecera (máximo valor 65535).
- Identificación: permite marcar de forma única cada datagrama enviado por una máquina (16 bits). Asegura la secuenciación. Asociado al mecanismo

de fragmentación, su valor se copia en todos los trozos de datagrama que componen el datagrama original)

- **Flag:** El tercer bit en uno indica que hay fragmentación 'more fragments' (se fragmenta si se ha superado el MTU). El último fragmento coloca el bit a cero. El segundo bit del campo Flag se denomina 'don't fragment' . Si está activado, fuerza un mensaje de error si el datagrama debe ser fragmentado y lo elimina.
- **Fragment Offset:** Índice del fragmento a partir del datagrama original (en bloques de 8 bytes) El campo longitud total es reajustado al tamaño de cada fragmento. Cada fragmento lleva su propia cabecera IP pero con el mismo identificador de datagrama.
- **TTL: (Time To Live)** tiempo de expiración del datagrama. Cada nodo por el que pasa el datagrama decrementa este campo. Si alcanza el valor cero el datagrama es destruido y se genera un mensaje ICMP. Permite evitar que un datagrama se quede atrapado en un bucle.
- **Protocolo:** identifica el protocolo que utiliza el datagrama IP. Por ejemplo TCP, UDP, ICMP, IGMP, etc.
- **Suma de Control:** (suma complemento a 1 de palabras de 16 bits). Permite comprobar errores en la cabecera (no tiene en cuenta los datos).
- **Direcciones IP fuente y destino:** contienen estas direcciones.
- **Opciones:** Permiten almacenar un registro de rutas, grabación de tiempos de paso por los routers, restricciones de seguridad, especificar la ruta. Está limitado a 40 bytes.

2.1.1.6 Enrutamiento IP

Se cuenta con una tabla de enrutamiento, la cual es un documento electrónico que almacena las rutas a los diferentes nodos en una red informática. Los nodos pueden ser cualquier tipo de dispositivo electrónico conectado a la red. La Tabla de encaminamiento generalmente se almacena en un router o en una red en forma de una base de datos o archivo. Cuando los datos deben ser enviados desde un nodo a otro de la red, se hace referencia a la tabla de encaminamiento con el fin de encontrar la mejor ruta para la transferencia de datos. La tabla enrutamiento contiene los siguientes campos:

- Dirección IP destino: puede ser una máquina o una red
- Dirección IP del siguiente router al que se debe enviar el datagrama
- Flags:

G: la dirección de IP de destino es una red (ruta indirecta)

H: la dirección de IP de destino es una máquina (ruta directa)

U: la ruta está en servicio

D: la ruta ha sido creada por redirección

M: ruta modificada por redirección

Debe existir una ruta por defecto ('default gateway')

Se pueden crear rutas manualmente (estáticamente) y existen además diferentes algoritmos de enrutamiento dinámico según el tipo de ruta (se basan todos ellos en el intercambio de las tablas entre routers):

Por ejemplo: una máquina inaccesible, una fragmentación requerida, etc...Un mensaje de error no puede ser producido por otro mensaje de error. Un mensaje de error contiene siempre la cabecera IP (20 bytes) y los 8 primeros bytes del datagrama que lo provocó (permite asociar el mensaje a un protocolo TCP/UDP y un puerto (usuario))

2.1.1.8 Capa de transporte

La capa de transporte implementa protocolos de extremo a extremo (entre el nodo origen y el destino de la información). Se definen dos protocolos: UDP y TCP.

2.1.1.9 UDP (User Datagram Protocol)

Proporciona un nivel de transporte no fiable de datagramas, ya que apenas añade la información necesaria para la comunicación extremo a extremo al paquete que envía al nivel inferior. Lo utilizan aplicaciones como NFS (Network File System) y RCP (comando para copiar ficheros entre ordenadores remotos), pero sobre todo se emplea en tareas de control y en la transmisión de audio y vídeo a través de una red. No introduce retardos para establecer una conexión, no mantiene estado de conexión alguno y no realiza seguimiento de estos parámetros. Así, un servidor dedicado a una aplicación particular puede soportar más clientes activos cuando la aplicación corre sobre UDP en lugar de sobre TCP.

UDP es un protocolo no orientado a la conexión. Cada operación de salida efectuada por un proceso genera un único datagrama UDP. Cada datagrama es autocontenido, no se mezclan. Cada datagrama UDP genera un datagrama IP aunque este puede ser fragmentado dependiendo del MTU.

UDP no ofrece garantía de secuencia (los datagramas pueden llegar desordenados). UDP no ofrece garantía de fiabilidad (los datagramas pueden ser incorrectos o no llegar). UDP no ofrece control de flujo. Es misión del protocolo de nivel aplicación el control de flujo y errores.

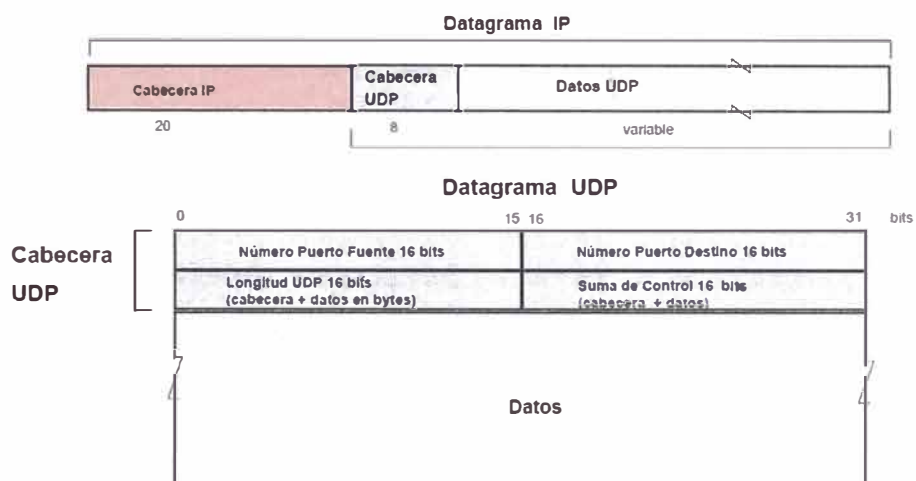


Figura 2.9 – Cabecera del datagrama UDP

2.1.1.10 TCP (Transmission Control Protocol)

TCP proporciona un transporte fiable de flujo de bits entre aplicaciones. Está pensado para poder enviar grandes cantidades de información de forma fiable, liberando al programador de la dificultad de gestionar la fiabilidad de la conexión (retransmisiones, pérdida de paquetes, orden en el que llegan los paquetes, duplicados de paquetes...) que gestiona el propio protocolo. Pero la complejidad de la gestión de la fiabilidad tiene un coste en eficiencia, ya que para llevar a cabo las gestiones anteriores se tiene que añadir bastante información a los paquetes que enviar. Debido a que los paquetes para enviar tienen un tamaño máximo, cuanto más información añade el protocolo para su gestión, menos información que proviene de la

aplicación podrá contener ese paquete (el segmento TCP tiene una sobrecarga de 20 bytes en cada segmento, mientras que UDP solo añade 8 bytes). Por eso, cuando es más importante la velocidad que la fiabilidad, se utiliza UDP. En cambio, TCP asegura la recepción en destino de la información para transmitir.

Es un protocolo de ventana deslizante no selectivo. Cuando TCP emite un segmento, mantiene un temporizador esperando su asentimiento por el otro extremo. Si expira el temporizador se reenvía el segmento. Si TCP recibe datos del otro extremo de la conexión, emite un asentimiento. TCP mantiene una suma de control en su cabecera y datos. Si aparece un segmento inválido, se rechaza y no se emite asentimiento. Los segmentos TCP son enviados como datagramas IP, por tanto pueden llegar desordenados. Los segmentos TCP llevan un número de secuencia por lo que TCP los reordena si es necesario.

Los segmentos TCP pueden llegar duplicados. TCP los elimina. TCP proporciona un control de flujo (tamaño de la ventana). Adecua un emisor rápido con un receptor lento. Una conexión TCP actúa como un flujo de bytes, pero TCP no interpreta su contenido.

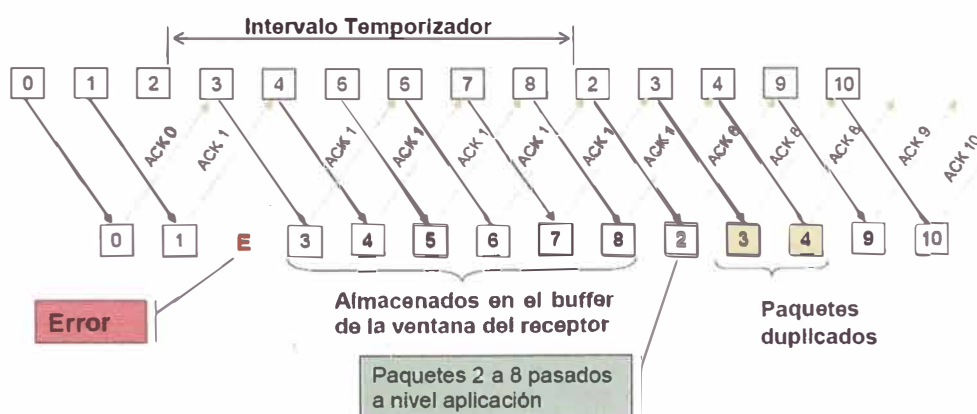


Figura 2.10 – Reenvío de un segmento cuando expira el temporizador.

Este número de secuencia es de 32 bits sin signo que retorna a 0 al llegar a $2^{32} - 1$. Cuando una conexión está siendo establecida el flag SYN se activa y el campo del número de secuencia contiene el ISN (initial sequence number) elegido por el host para esa conexión. El número de secuencia del primer byte de datos será el ISN+1 ya que el flag SYN consume un número de secuencia.

- Número de acuse de recibo (32 bits): Contiene el valor del siguiente número de secuencia que el emisor del segmento espera recibir. Una vez que la conexión ha sido establecida, este número se envía siempre y se valida con el flag ACK activado. Enviar ACKs no cuesta nada ya que el campo de acuse de recibo siempre forma parte de la cabecera, al igual que el flag ACK. TCP se puede describir como un protocolo sin asentimientos selectivos o negativos ya que el número de asentimiento en la cabecera TCP significa que se han recibido correctamente los bytes anteriores pero no se incluye ese byte.
- Longitud de cabecera (4 bits): especifica el tamaño de la cabecera en palabras de 32 bits. Es requerido porque la longitud del campo "opciones" es variable. Por lo tanto el tamaño máximo de la cabecera está limitado a 60 bytes, mientras que sin "opciones" el tamaño normal será de 20 bytes. A este campo también se le suele llamar "data offset" por el hecho de que es la diferencia en bytes desde el principio del segmento hasta el comienzo de los datos.
- Reservado (3 bits): para uso futuro. Debe estar a 0.
- Flags (9 bits):

NS (1 bit): Para proteger frente a paquetes accidentales o maliciosos que se aprovechan del control de congestión para ganar ancho de banda de la red.

CWR (1bit): Congestion Window Reduced. El flag se activa por el host emisor para indicar que ha recibido un segmento TCP con el flag ECE activado y ha respondido con el mecanismo de control de congestión.

ECE (1 bit): Para dar indicaciones sobre congestión.

URG (1 bit): Indica que el campo del puntero urgente es válido.

ACK (1 bit): Indica que el campo de asentimiento es válido. Todos los paquetes enviados después del paquete SYN inicial deben tener activo este flag.

PSH (1 bit): Push. El receptor debe pasar los datos a la aplicación tan pronto como sea posible.

RST (1 bit): Reset. Reinicia la conexión.

SYN (1 bit): Synchronice. Sincroniza los números de secuencia para iniciar la conexión.

FIN (1 bit): El emisor finaliza el envío de datos.

- Tamaño de ventana (16 bits): Tamaño de la ventana de recepción que especifica el número máximo de bytes que pueden ser metidos en el buffer de recepción o dicho de otro modo, el número máximo de bytes pendientes de asentimiento.
- Suma de verificación (16 bits): Checksum utilizado para la comprobación de errores tanto en la cabecera como en los datos.
- Puntero urgente (16 bits): Cantidad de bytes desde el número de secuencia que indica el lugar donde acaban los datos urgentes.

- Opciones: Para poder añadir características no cubiertas por la cabecera fija.
- Relleno: Se utiliza para asegurarse que la cabecera acaba con un tamaño múltiplo de 32 bits.

2.1.1.11 Conexión y desconexión TCP

Las conexiones TCP se componen de tres etapas: establecimiento de conexión, transferencia de datos y fin de la conexión. Para establecer la conexión se usa el procedimiento llamado negociación en tres pasos (3-way handshake). Para la desconexión se usa una negociación en cuatro pasos (4-way handshake). Durante el establecimiento de la conexión, se configuran algunos parámetros tales como el número de secuencia con el fin de asegurar la entrega ordenada de los datos y la robustez de la comunicación.

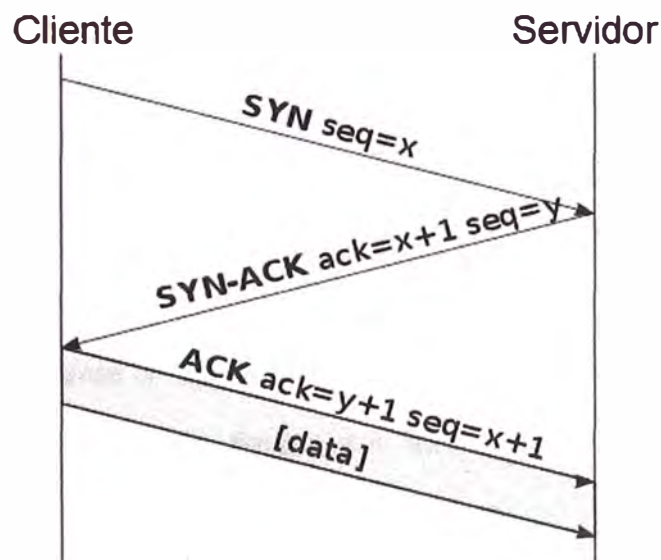


Figura 2.12 – Reenvío de un segmento cuando expira el temporizador.

Aunque es posible que un par de entidades finales comiencen una conexión entre ellas simultáneamente, normalmente una de ellas abre un

socket en un determinado puerto TCP y se queda a la escucha de nuevas conexiones. Es común referirse a esto como apertura pasiva, y determina el lado servidor de una conexión. El lado cliente de una conexión realiza una apertura activa de un puerto enviando un paquete SYN inicial al servidor como parte de la negociación en tres pasos. En el lado del servidor (este receptor también puede ser una PC o alguna estación terminal) se comprueba si el puerto está abierto, es decir, si existe algún proceso escuchando en ese puerto, pues se debe verificar que el dispositivo de destino tenga este servicio activo y esté aceptando peticiones en el número de puerto que el cliente intenta usar para la sesión. En caso de no estarlo, se envía al cliente un paquete de respuesta con el bit RST activado, lo que significa el rechazo del intento de conexión. En caso de que sí se encuentre abierto el puerto, el lado servidor respondería a la petición SYN válida con un paquete SYN/ACK. Finalmente, el cliente debería responderle al servidor con un ACK, completando así la negociación en tres pasos (SYN, SYN/ACK y ACK) y la fase de establecimiento de conexión. Es interesante notar que existe un número de secuencia generado por cada lado, ayudando de este modo a que no se puedan establecer conexiones falseadas (spoofing).

Durante la etapa de transferencia de datos, una serie de mecanismos claves determinan la fiabilidad y robustez del protocolo. Entre ellos están incluidos el uso del número de secuencia para ordenar los segmentos TCP recibidos y detectar paquetes duplicados, checksums para detectar errores, y asentimientos y temporizadores para detectar pérdidas y retrasos.

La fase de finalización de la conexión usa una negociación en cuatro pasos (four-way handshake), terminando la conexión desde cada lado independientemente. Cuando uno de los dos extremos de la conexión desea

Señal		DB-25	DE-9 (DB-9, TIA-574)	EIA/TIA 561	Host	RJ-50	MMJ
Common Ground	G	7	5	4	4,5	6	3,4
Transmitted Data	TD	2	3	6	3	8	2
Received Data	RD	3	2	5	6	9	5
Data Terminal Ready	DTR	20	4	3	2	7	1
Data Set Ready	DSR	6	6	1	7	5	6
Request To Send	RTS	4	7	8	1	4	-
Clear To Send	CTS	5	8	7	8	3	-
Carrier Detect	DCD	8	1	2	7	10	-
Ring Indicator	RI	22	9	1	-	2	-

Figura 2.14 – Reenvío de un segmento cuando expira el temporizador.

La interfaz RS-232 puede operar en distancias de hasta 15 metros con velocidades de transmisión no superiores a 20 kbps. La velocidad máxima absoluta puede variar en función de las condiciones de la línea y la longitud del cable. La interfaz RS-232 opera frecuentemente a 38,4 Kbps sobre distancias muy cortas.

Los niveles de voltaje definidos por el estándar RS-232 van desde -12 a +12 voltios. La interfaz RS-232 es una interfaz no balanceada o de una sola terminación, lo que significa que una señal eléctrica única es comparada con una señal común (tierra) para determinar los estados lógicos binarios. Un voltaje de +12 voltios (normalmente de +3 a +15 voltios) representa un 0 binario (espacio), y -12 voltios (de -3 a -15 voltios) representa un 1 binario (marca).

2.1.2.1 Control de flujo en RS-232

Existen dos tipos de control de flujo, por hardware (CTS/RTS) y por software (XON/XOFF).

En el caso del control de flujo con las señales CTS/RTS, el lado transmisor revisa el estado de su CTS antes de transmitir, y si el CTS está activo enviará datos, caso contrario no transmitirá datos. Por otro lado, el receptor usa el RTS para indicar al transmisor si debe transmitir más datos o pausar la transmisión.

Si bien ambos tipos de control de flujo utilizan software, se llama control de flujo por software al que usa los caracteres ASCII XON y XOFF debido a que en este caso no es necesario usar líneas adicionales de cableado para las señales de control de flujo, si no que estos dos caracteres ASCII son usados directamente en la línea de transmisión de datos para realizar el control de flujo. Enviar un carácter XOFF causará que el transmisor deje de enviar datos, análogamente enviando el carácter XON se le indicará al transmisor que puede continuar con la transmisión de datos.

2.1.3 Bases de datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

2.1.3.1 Sistemas de gestión de bases de datos

Un sistema de gestión de bases de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto Los SGBD

también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permite presentar la información de la base de datos en variados formatos. La mayoría de los SGBD incluyen un generador de informes. También puede incluir un módulo gráfico que permita presentar la información con gráficos y tablas.

Generalmente se accede a los datos mediante lenguajes de interrogación, lenguajes de alto nivel que simplifican la tarea de construir las aplicaciones. También simplifican la interrogación y la presentación de la información. Un SGBD permite controlar el acceso a los datos, asegurar su integridad, gestionar el acceso concurrente a ellos, recuperar los datos tras un fallo del sistema y hacer copias de seguridad.

2.1.3.2 Lenguajes de interrogación a bases de datos

Los lenguajes de consulta de bases de datos y de generación de informes permiten interrogar a la base de datos, analizar los datos y actualizarlos según los privilegios de cada usuario. También controla la seguridad de la base de datos para prevenir accesos no autorizados que vean, borren o cambien los datos. Mediante el uso de claves se permite el acceso a toda la base de datos o a parte de ella.

2.1.3.3 Lenguaje de consulta estructurado (SQL)

El lenguaje de consulta estructurado o SQL (por sus siglas en inglés Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo

relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información de interés de bases de datos, así como hacer cambios en ella.

El SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales y permite así gran variedad de operaciones.

Es un lenguaje declarativo de "alto nivel" o "de no procedimiento" que, gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros permite una alta productividad en codificación y la orientación a objetos. De esta forma, una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros. SQL también tiene las siguientes características:

- Lenguaje de definición de datos: El LDD de SQL proporciona comandos para la definición de esquemas de relación, borrado de relaciones y modificaciones de los esquemas de relación.
- Lenguaje interactivo de manipulación de datos: El LMD de SQL incluye lenguajes de consultas basado tanto en álgebra relacional como en cálculo relacional de tuplas.
- Integridad: El LDD de SQL incluye comandos para especificar las restricciones de integridad que deben cumplir los datos almacenados en la base de datos.
- Definición de vistas: El LDD incluye comandos para definir las vistas.
- Control de transacciones: SQL tiene comandos para especificar el comienzo y el final de una transacción.

- SQL incorporado y dinámico: Esto quiere decir que se pueden incorporar instrucciones de SQL en lenguajes de programación como: C++, C, Java, PHP, Cobol, Pascal y Fortran.
- Autorización: El LDD incluye comandos para especificar los derechos de acceso a las relaciones y a las vistas.

Algunos de los tipos de datos básicos de SQL son:

- Date: una fecha de calendario que contiene el año (de cuatro cifras), el mes y el día.
- Time: La hora del día en horas minutos segundos (el valor predeterminado es 0).
- Timestamp: la combinación de Date y Time.

2.2 Formulación del producto

El producto final propuesto como solución a la problemática definida previamente es un sistema que permita tomar ventaja de las capacidades de comunicación serial RS-232 ya existente en los terminales de pesaje Mettler-Toledo ID3S para leer los pesajes de los sacos de producto. Dado que la solución se desarrollará para tres terminales de pesaje con comunicación RS-232 punto a punto y que es posible esto se extienda en un futuro a más terminales de pesaje, se plantea el uso de un conversor Serial-Ethernet con múltiples puertos, de tal manera que el sistema propuesto se comunique por TCP/IP a un solo equipo de hardware (dicho conversor), pero manteniendo comunicaciones independientes a nivel TCP/IP a cada puertos serial RS-232 de este conversor, los cuales estarán conectados a cada terminal de pesaje.

Se plantea además el diseño de un software que realice la identificación y clasificación de los pesajes dentro de rango, fuera de rango, y malos pesajes, y que esta data sea grabada en la base de datos de la empresa cliente. Debido a las limitaciones de costos establecidas por el cliente así como al hecho de que la operación de pesaje es lenta (en el orden de uno a dos minutos para completar un saco de producto), se determina instalar este software en un computador personal(hardware no-industrial), con la posibilidad de ser exportado posteriormente a otra plataforma de hardware. Se acuerda también que toda la información de pesos sea guardada en la base de datos SQL ya existente en la planta.

CAPÍTULO III

DESARROLLO DEL SISTEMA

3.1. Comunicación con los terminales de pesaje

La comunicación entre el software desarrollado y los terminales de pesaje consta de dos etapas; una etapa serial RS-232, entre cada uno de los tres terminales de pesaje y un conversor Ethernet-Serial; y una etapa Ethernet, desde este conversor hasta el computador en el cual se encuentra instalada la aplicación cliente.

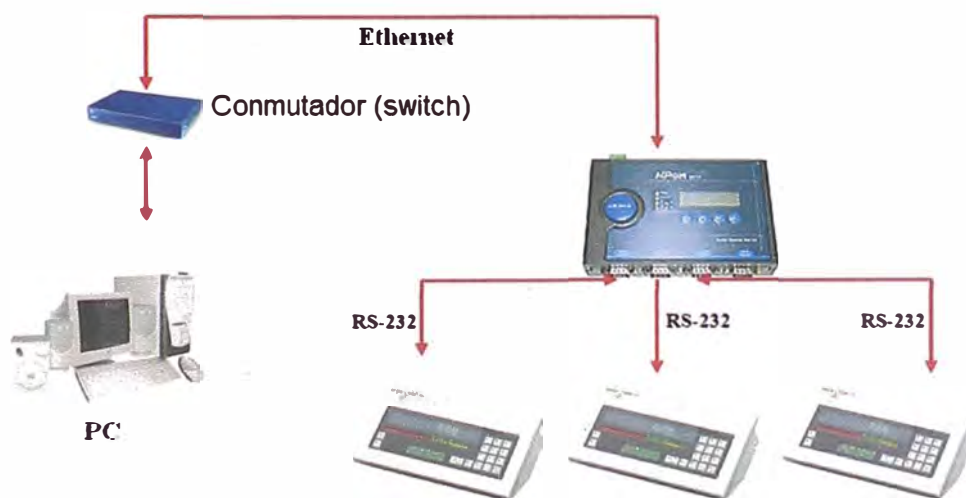


Figura 3.1 – Esquema de equipos utilizados en las comunicaciones del sistema

3.1.1 Conversor Ethernet-serial

El conversor serial Ethernet se encarga de proveer conectividad a cada una de sus cuatro interfaz seriales mediante una sola interfaz Ethernet. A nivel TCP, cada interfaz serial está asociada a un puerto TCP distinto.

El conversor fue configurado para operar como un servidor TCP, de tal manera que espera pasivamente la conexión por parte de la aplicación cliente instalada en el computador. Cada uno de los tres primeros puertos seriales va conectado a uno de los tres terminales de pesaje. La configuración del IP y puertos TCP usados debe coincidir con los configurados en la aplicación cliente.

Una vez establecida la conexión el conversor funciona de manera transparente, permitiendo el flujo bidireccional de datos.

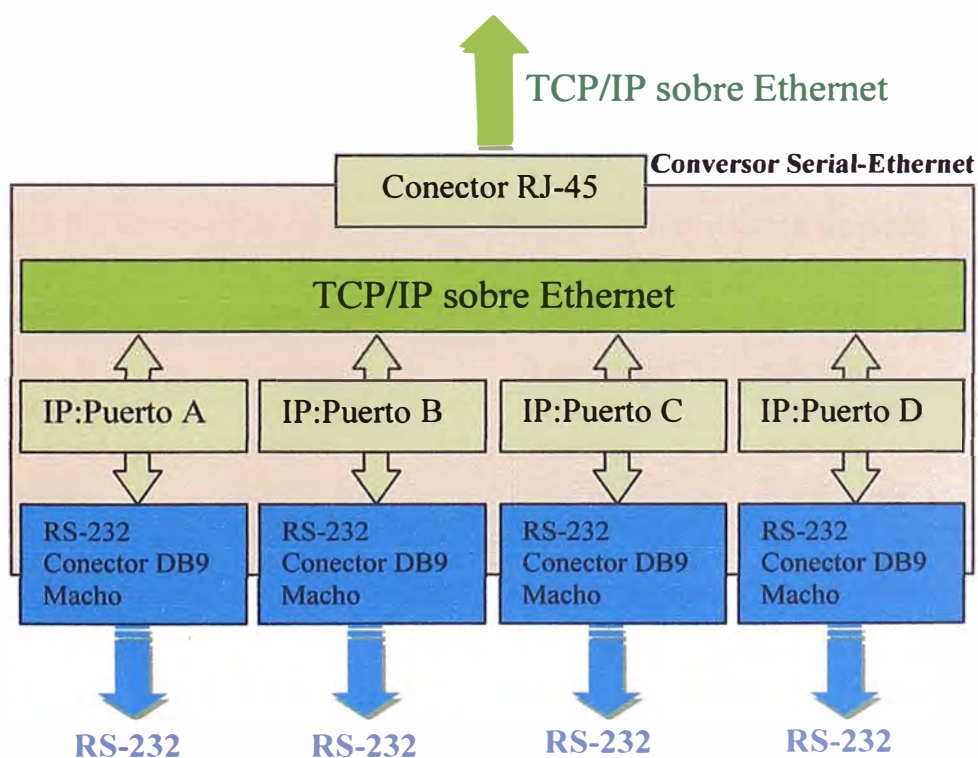


Figura 3.2 – Esquema del conversor Ethernet-serial.

3.1.2 Formato de tramas de mando y respuesta

La comunicación entre el conversor Ethernet-serial y los terminales de pesaje es serial RS-232. La comunicación consiste en la transmisión de tramas hacia el terminal de pesaje, las cuales incluyen un identificador de mando, un valor de peso y unidades, y un delimitador que indica el fin de la trama. Los tres campos iniciales están separados por un carácter de guion inferior. La sintaxis de esta trama de mando es descrita en la Tabla 3.1.

Tabla 3.1 Formato de mando en la transmisión de valores de peso

Identificación	_	Valor de peso	_	Unidad	Limitación
Secuencia de caracteres para la especificación del mando (1 ... 4 caracteres)		1 ... 8 cifras, número de cifras variable		1 ... 3 caracteres, número de caracteres variable	definible en Master Mode, ajuste de fábrica: C _R L _F

Tabla 3.2 Formato de respuesta en la transmisión de valores de peso

Identificación	_	Valor de peso	_	Unidad	Limitación
Secuencia de caracteres para la especificación de la respuesta (2 ... 3 caracteres)		10 cifras, alineadas a derecha, llenado con espacios		3 caracteres, alineados a izquierda, llenado con espacios	definible en Master Mode, ajuste de fábrica: C _R L _F

Si la trama de mando enviada al terminal de pesaje cumple con la trama y contenido correcto, el terminal de pesaje responderá al mando correspondiente con otra trama de datos comenzando por un identificador de la respuesta, el valor del peso, la unidad utilizada, y un delimitador que indica

el final de la trama. Los tres campos iniciales están separados por un carácter de guion inferior. La sintaxis de esta trama de respuesta se describe en la Tabla 3.2.

Para la lectura de pesos de la balanza se utiliza el mando SI el cual permite transmitir los pesos independientemente de la estabilización de la plataforma de pesaje. Luego, un carácter en la respuesta nos permitirá identificar si el peso corresponde a un valor estabilizado o no, es decir, si el peso del saco sobre la balanza llegó a un valor estable, o si está cambiando dinámicamente (durante el llenado de los sacos por ejemplo, o por alguna perturbación).

Tabla 3.3 Mandos para solicitar el peso leído por el terminal de pesaje

Mando	S	Transmitir con estabilización de la plataforma de balanza un valor de peso estable.
	S, I	Independiente de la estabilización de la plataforma de pesaje, transmitir un valor de peso estable o dinámico.
Respuesta	S, _ _	<u>Valor de peso</u> _ <u>Unidad</u> Transmitido el valor de peso estable
	S, D _	<u>Valor de peso</u> _ <u>Unidad</u> Transmitido el valor de peso dinámico
	S, I	Valor no válido
	S, I, -	Plataforma de pesaje en régimen de carga de menos
	S, I, +	Plataforma de pesaje en régimen de carga de más

3.1.3 Conexión de datos hacia los terminales de pesaje

La primera etapa para la comunicación con los terminales de pesaje a través de la aplicación cliente es abrir las conexiones TCP/IP hacia el conversor Ethernet-Serial, el cual está configurado como servidor TCP. Cada puerto serial del conversor cuenta con una conexión separada en un puerto TCP distinto. La figura 3.3 muestra el diagrama de flujo para abrir la conexión TCP/IP.

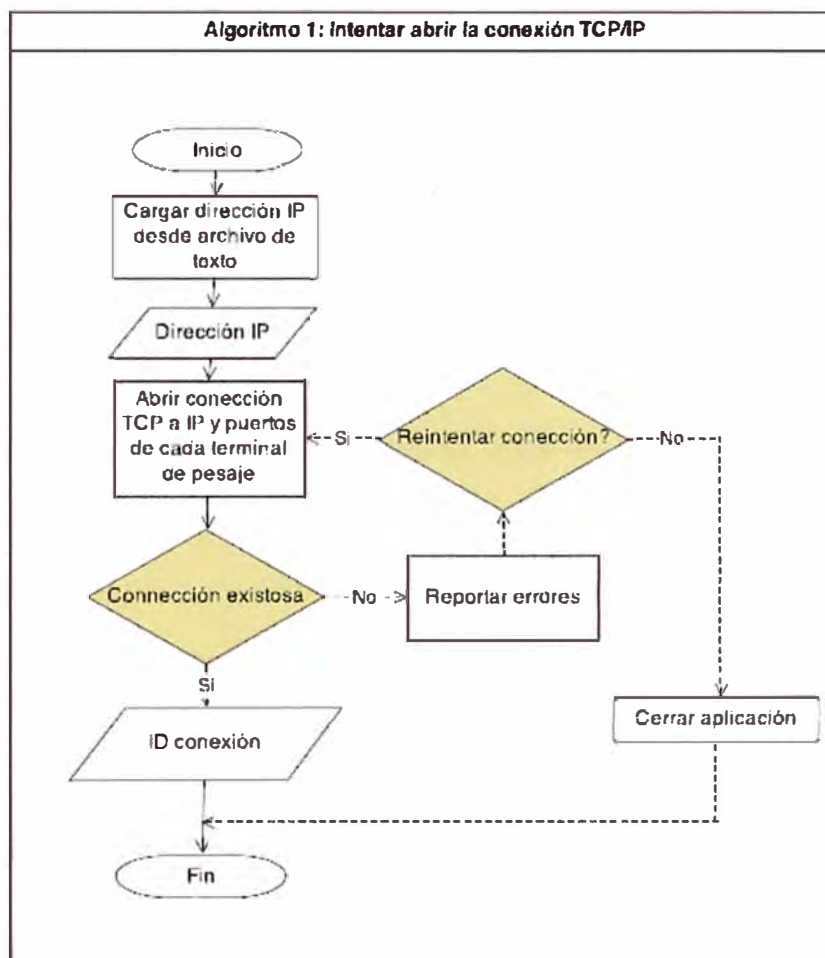


Figura 3.3 – Diagrama de flujo para el intento de apertura de la conexión
TCP/IP

Una vez la conexión está abierta se obtiene un identificador de conexión el cual será posteriormente usado para interactuar con esta conexión TCP/IP.

3.1.4 Lectura de pesos de los terminales de pesaje

Luego de establecer una conexión exitosa, se ejecutan los lazos de lectura de datos de pesajes. Para esto envía el comando "SI" seguido por los caracteres ASCII CR y LF a cada balanza.

Luego del envío del comando de transmisión (SI) el lazo correspondiente queda a la espera de recibir la respuesta de la balanza en el mismo puerto TCP. Dado que el delimitador de la respuesta son también los caracteres ASCII CR y LF, el comando de lectura TCP finaliza si recibe esta terminación, si alcanza el valor máximo de 25 bytes, o si alcanza el tiempo límite de espera de 2 segundos.

Si la trama es recibida exitosamente, esta es luego fragmentada en sus distintas componentes conforme a la estructura descrita en la tabla 3.3, así como el valor de peso convertido a formato de doble precisión.

El identificador booleano de nombre Estático es usado para identificar si el peso actual corresponde a una lectura de un valor estable en el terminal de pesaje. Por otro lado, el identificador booleano EstaticoAnt indica el estado estático de la lectura anterior. El identificador booleano CambioAEstatico indica que el peso anterior fue dinámico y el actual es estático.

3.1.5 Identificación de los pesajes correctos e incorrectos

El fabricante utiliza los términos “peso dinámico” y “peso estático” para diferenciar la etapa de respuesta transitoria del equipo de pesaje de la condición de estado estable en el cual el valor medido por el terminal de pesaje puede considerarse constante para fines prácticos.

El fabricante no indica cual es el umbral de error que utiliza para diferenciar estas dos condiciones, sin embargo según las pruebas en operación normal esta condición se alcanza en pocos segundos.

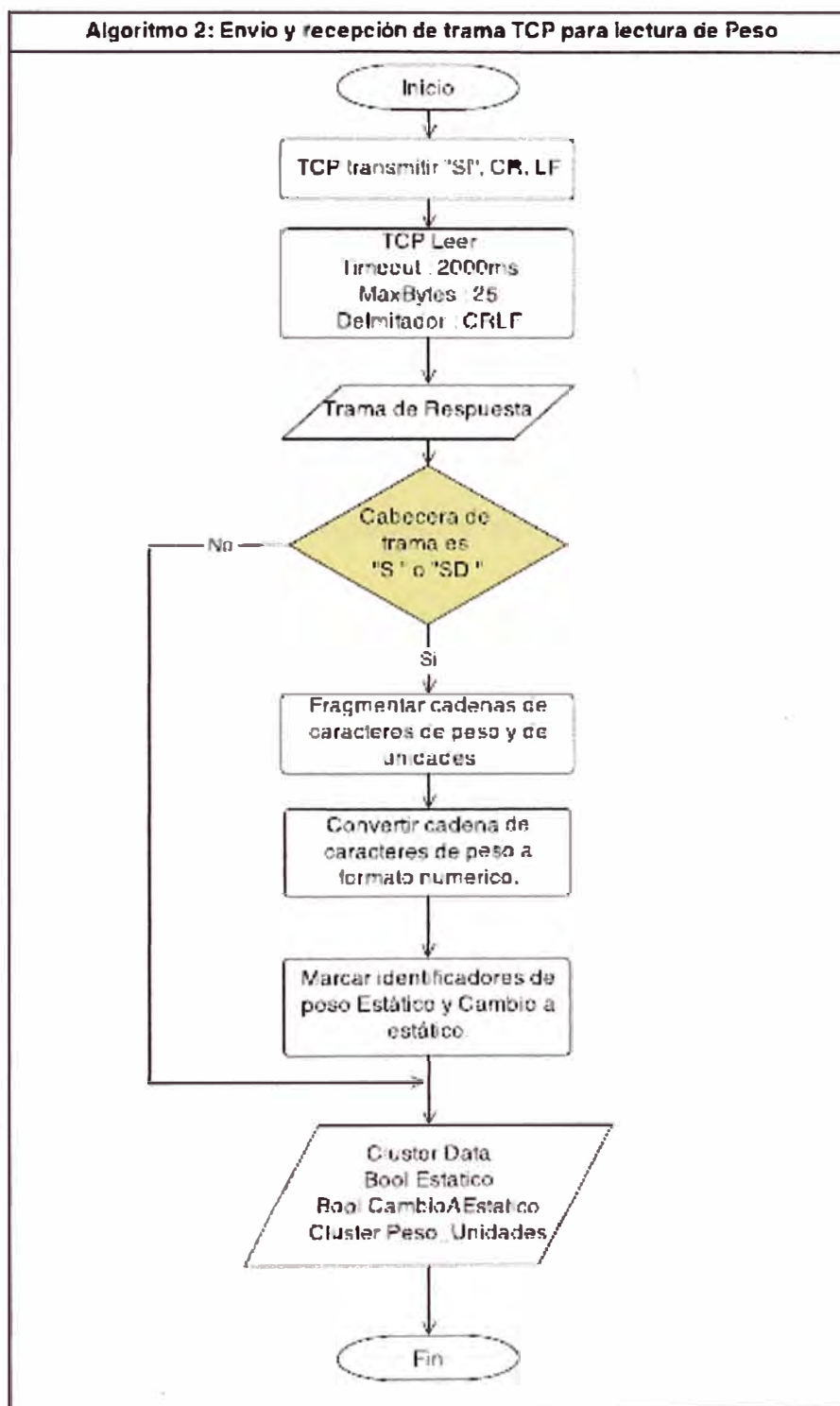


Figura 3.4 – Diagrama de flujo para el envío y recepción de la trama TCP para la lectura de peso.

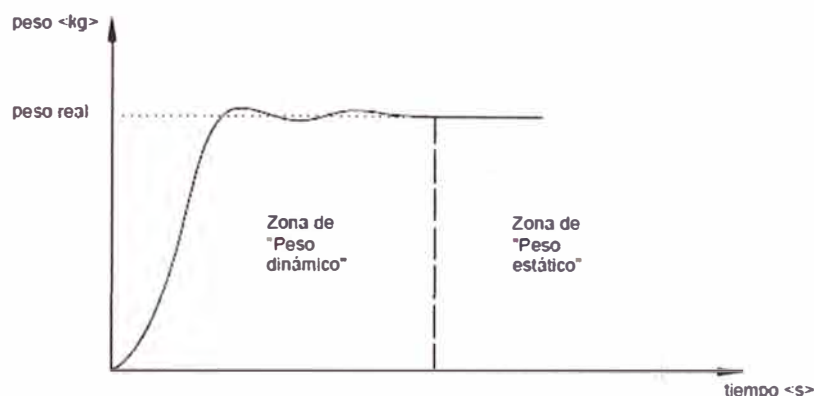


Figura 3.5 – Diferencia entre los términos “peso dinámico” y “peso estático” usados por el fabricante del terminal de pesaje

El proceso de pesaje estándar es esquematizado en la figura 3.6. El operador coloca un saco vacío en el terminal de pesaje y a continuación introduce las bolsas del producto en este. Esta operación la repite hasta que el peso del saco esté dentro de las especificaciones definidas por un límite mínimo y límite máximo preestablecido. Durante este proceso el terminal de pesaje registrará varios pesos dinámicos y posiblemente también pesos estáticos si es que el operador pausó la operación de llenado el tiempo suficiente. Es indispensable sin embargo que al llegar al peso final del saco el operador deje este en reposo hasta que el terminal de pesaje indique que ha alcanzado la estabilidad. La balanza utiliza el símbolo ~ para indicar un peso dinámico, y la ausencia de este símbolo indica entonces un pesaje estático. Una vez sucede esto el operador debe retirar el saco de la balanza y permitir que el peso se estabilice nuevamente sin carga (idealmente en cero).

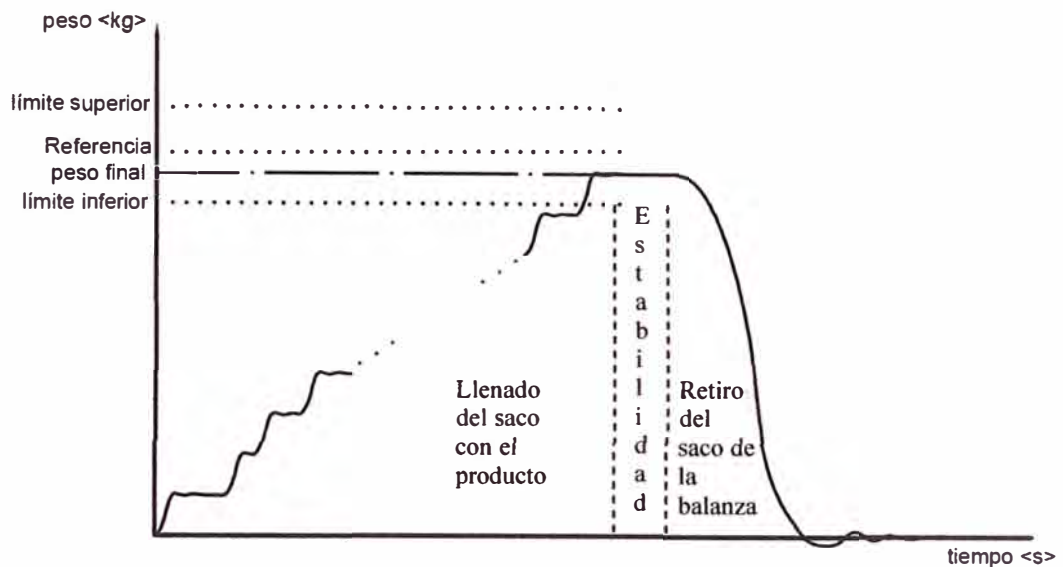


Figura 3.6 – Proceso estándar de pesaje de sacos de producto

El operador debe permitir al terminal de pesaje llegar a esta estabilidad antes de retirar el saco de producto de la balanza, caso contrario el sistema considerará el último peso estático que haya obtenido desde que se colocó el saco sobre el terminal de pesaje, y de no haber habido ninguno considerará entonces la última operación realizada como un mal pesaje.

A medida que el saco va siendo llenado la aplicación cliente guarda constantemente el último valor de peso estático recibido del terminal de pesaje correspondiente. Una vez el saco es retirado del terminal de pesaje la aplicación cliente considera este último valor de pesaje estático como el valor final de peso del saco. La identificación del retiro del saco se hace utilizando un umbral igual a la quinta parte de la referencia, cuando la aplicación cliente recibe un peso estático por debajo de este umbral habiendo previamente tenido un peso por encima, considera entonces que el saco fue retirado del terminal de pesaje. Dado que se usan pesos estáticos y que el producto introducido en los sacos se encuentra en unidades discretas (paquetes) de

tamaño significativo frente al valor total no fue entonces necesario utilizar un valor de histéresis para este umbral de detección.

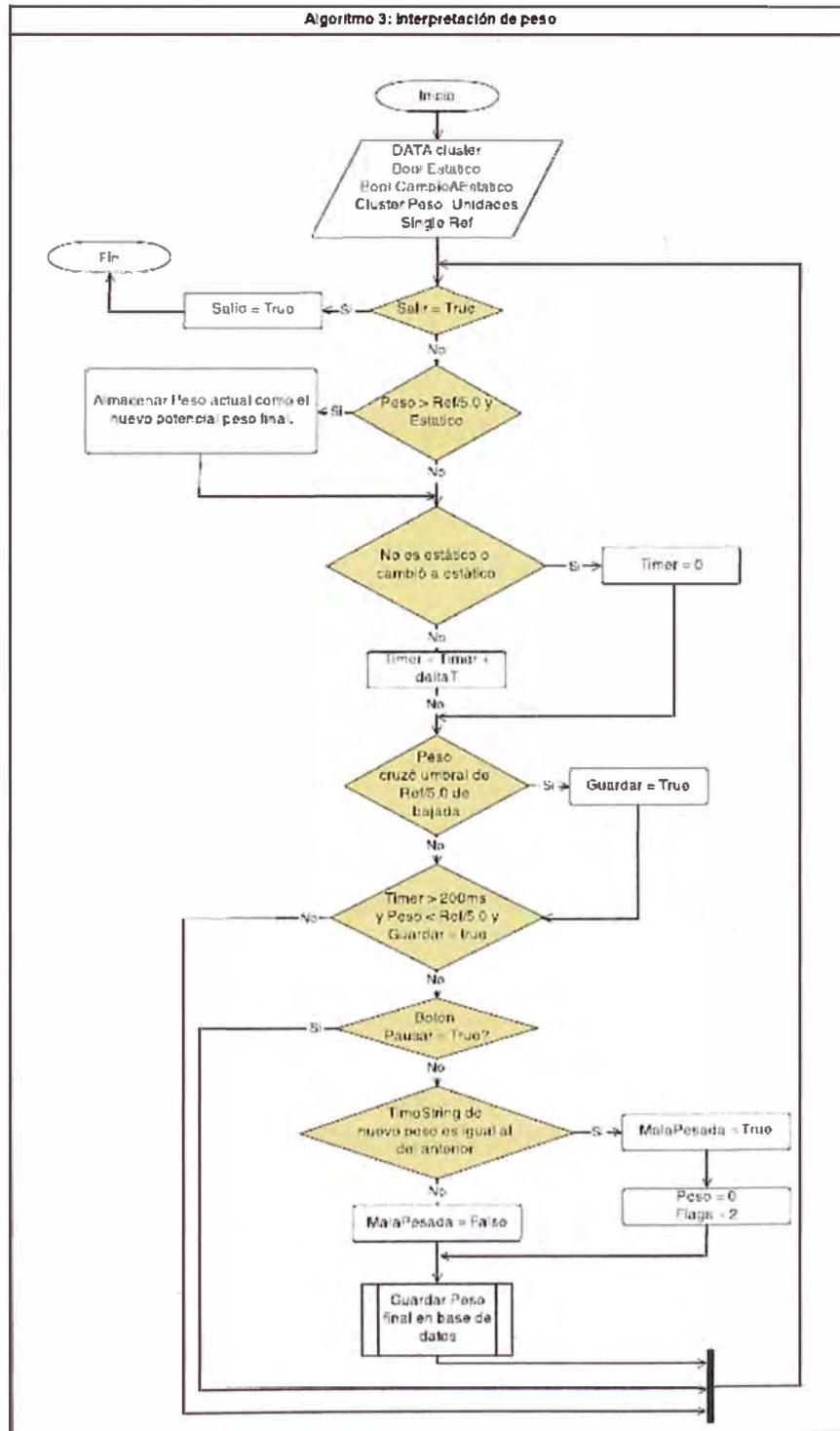


Figura 3.7 –Diagrama de flujo para interpretación y clasificación de pesos.

Se cuenta con un botón de pausa que permite detener la grabación de datos para cualquier de los terminales de pesaje, permitiendo realizar pruebas, calibraciones y otros sin distorsionar la data de la base de datos con valores no relacionados a la operación normal.

3.2. Registro en base de datos

La aplicación cliente registra los pesos de cada saco en los servidores que contienen la base de datos corporativa de la empresa cliente. La plataforma de esta base de datos es Microsoft SQL Server 2003.

3.2.1 Campos de la tabla en la base de datos

A pedido del cliente, toda la data es guardada en una sola tabla en la base de datos; el cliente pidió no usar un modelo relacionar sino solo una tabla por simplicidad. Los datos son entonces guardados en la tabla de nombre BALANZAS con los campos mostrados en la figura 3.8.



Name
Balanza (int, null)
FECHA_HORA (datetime, null)
PESO (real, null)
UNIDADES (varchar(50), null)
FLAGS (int, null)
OBSERVACION (varchar(50), null)
LIMITE_INF (real, null)
LIMITE_SUP (real, null)
SET_POINT (real, null)

Figura 3.8 – Campos de la tabla BALANZAS en la base de datos.

Cada uno de los campos de la tabla tienen el significado siguiente:

- **Balanza:** dato de tipo entero que identifica al terminal de pesaje del cual provienen los datos.
- **FECHA_HORA:** campo de fecha y hora en la cual se registró el peso en la aplicación cliente.
- **PESO:** valor numérico del peso para el saco registrado.
- **UNIDADES:** unidades físicas para el valor numérico del peso (por ej. Kg).
- **FLAGS:** número entero sin signo, para el cual el valor cero indica que el peso ha sido obtenido correctamente y está dentro de rango, el valor uno indica que el peso fue obtenido correctamente pero está fuera de los límites superior o inferior, y el valor dos indica que la operación de pesaje fue realizada incorrectamente.
- **OBSERVACION:** campo de texto de hasta cincuenta caracteres para introducir cualquier comentario y observación referente a este pesaje. Fue posteriormente extendido a doscientos caracteres a pedido del cliente.
- **LIMITE_INF:** Límite inferior de peso para el saco de producto que se está envasando. Este dato es configurado para la referencia de saco que se está envasando actualmente.
- **LIMITE SUP:** Límite superior de peso para el saco de producto que se está envasando. Este dato es configurado para la referencia de saco que se está envasando actualmente.
- **SET_POINT:** referencia de valor nominal de peso del saco de producto que se está envasando actualmente.

3.2.2 Escritura de pesos en la base de datos

La conexión con la base de datos se realiza a través de ODBC. La aplicación cliente tiene para esto definida la información de conexión y parámetros de la base de datos en un DSN (Nombre de fuente de datos).

La apertura de la conexión es trivial como se muestra en el algoritmo cuatro, mostrado en la figura 3.9. El algoritmo no consulta una reconexión ya que en otra etapa de la aplicación cliente se intentará periódicamente la reconexión en caso no estuviéramos conectados a la base de datos. El cliente pidió este intento de reconexión sea automática.

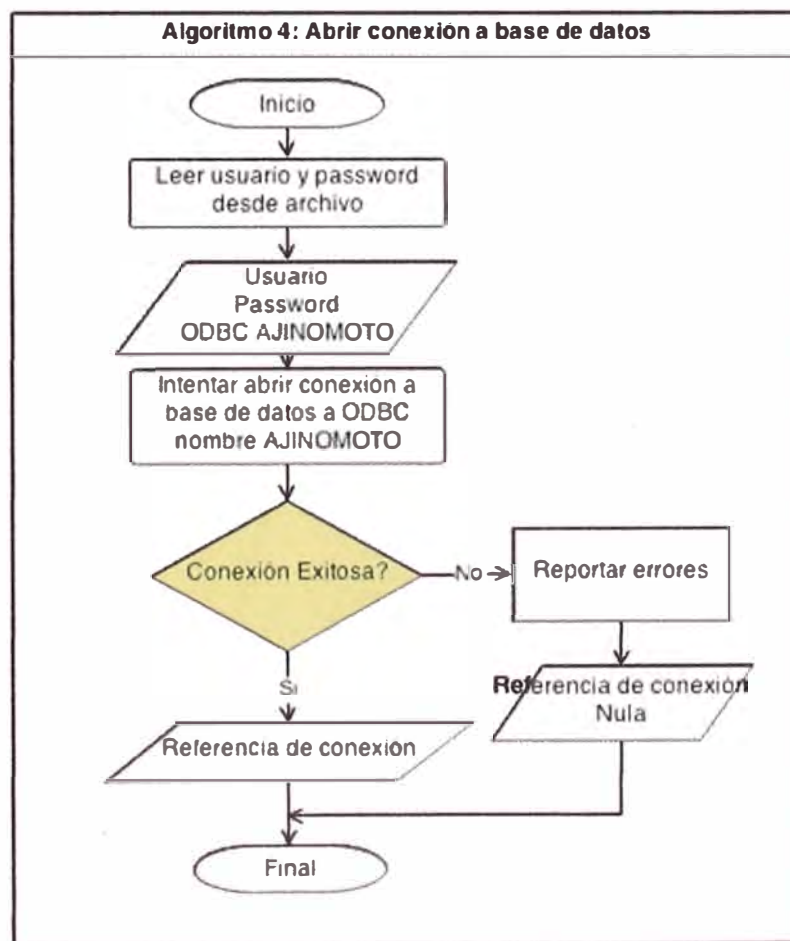


Figura 3.9 – Diagrama de flujo para la apertura de conexión a la base de datos

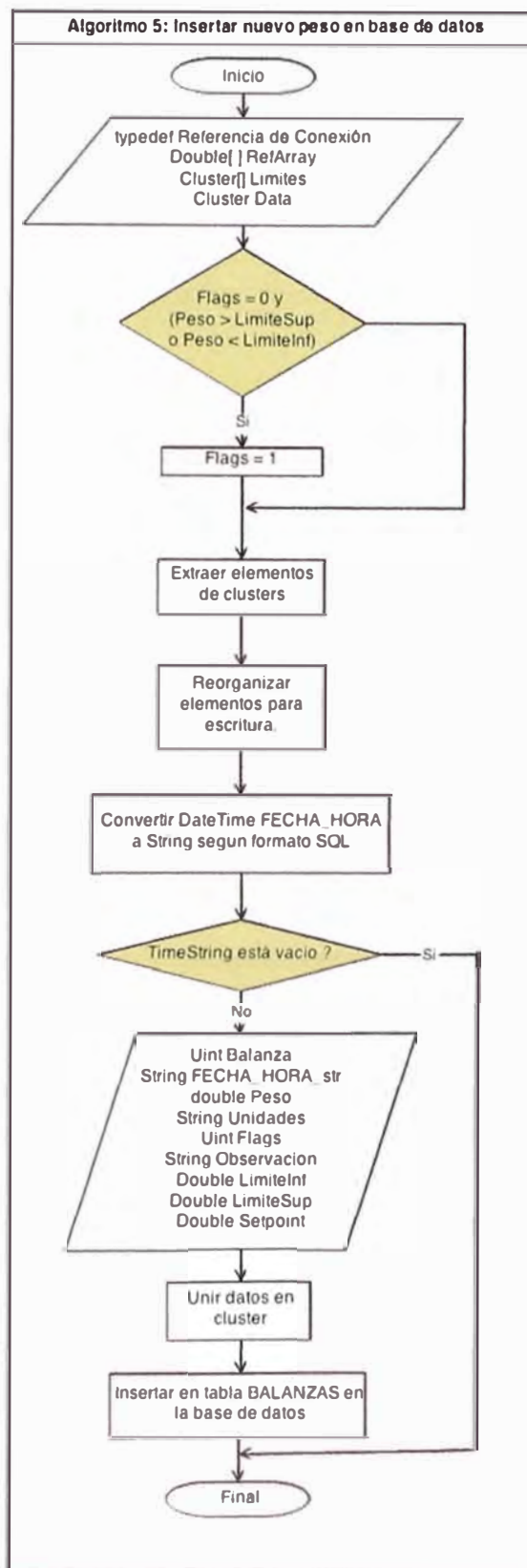


Figura 3.10 – Diagrama de flujo para la inserción de una nueva entrada de pesaje en la base de datos

Para la escritura en la base de datos se extrae primeramente los datos necesarios de los cluster de datos y de límites, así como la referencia de peso desde el array de referencias y se reorganizan de acuerdo a los campos definidos en la tabla BALANZAS de la base de datos. Vale la pena mencionar que el flag que identifica un pesaje fuera de rango es activado en este bloque.

3.2.3 Lectura de pesos desde la base de datos

Los nuevos pesos registrados por el sistema son grabados en la base de datos y a su vez la aplicación muestra el contenido de datos grabados. De esta manera, se asegura que en todo momento la data que se muestra en pantalla corresponde efectivamente a la data grabada en la base de datos y no a ningún archivo o memoria temporal de la aplicación.

3.2.4 Agregar observaciones a los pesos

La aplicación permite igualmente utilizar la interfaz gráfica para mostrar datos del día de hoy (valor por defecto), o de cualquier otro día. La alta eficiencia que proveen los motores de base de datos permite que la aplicación pueda proveer esa flexibilidad sin deteriorar su desempeño.

La aplicación permite agregar observaciones en la base de datos de hasta cincuenta caracteres. Para esto se utiliza el comando UPDATE de SQL. La figura 3.12 muestra el algoritmo para la inserción de un comentario a un peso en la base de datos.

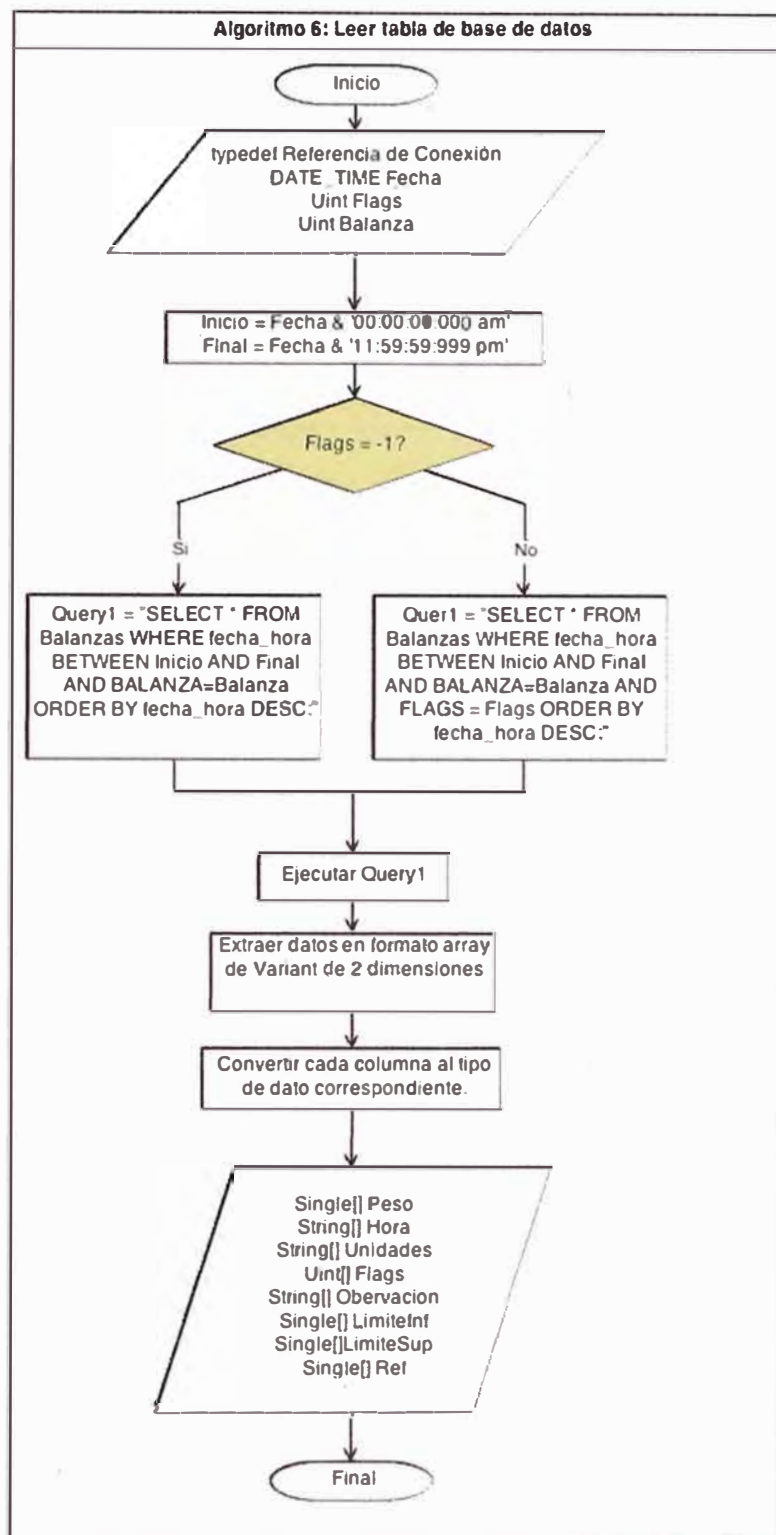


Figura 3.11 – Diagrama de flujo para la lectura de la tabla de pesajes en la base de datos.

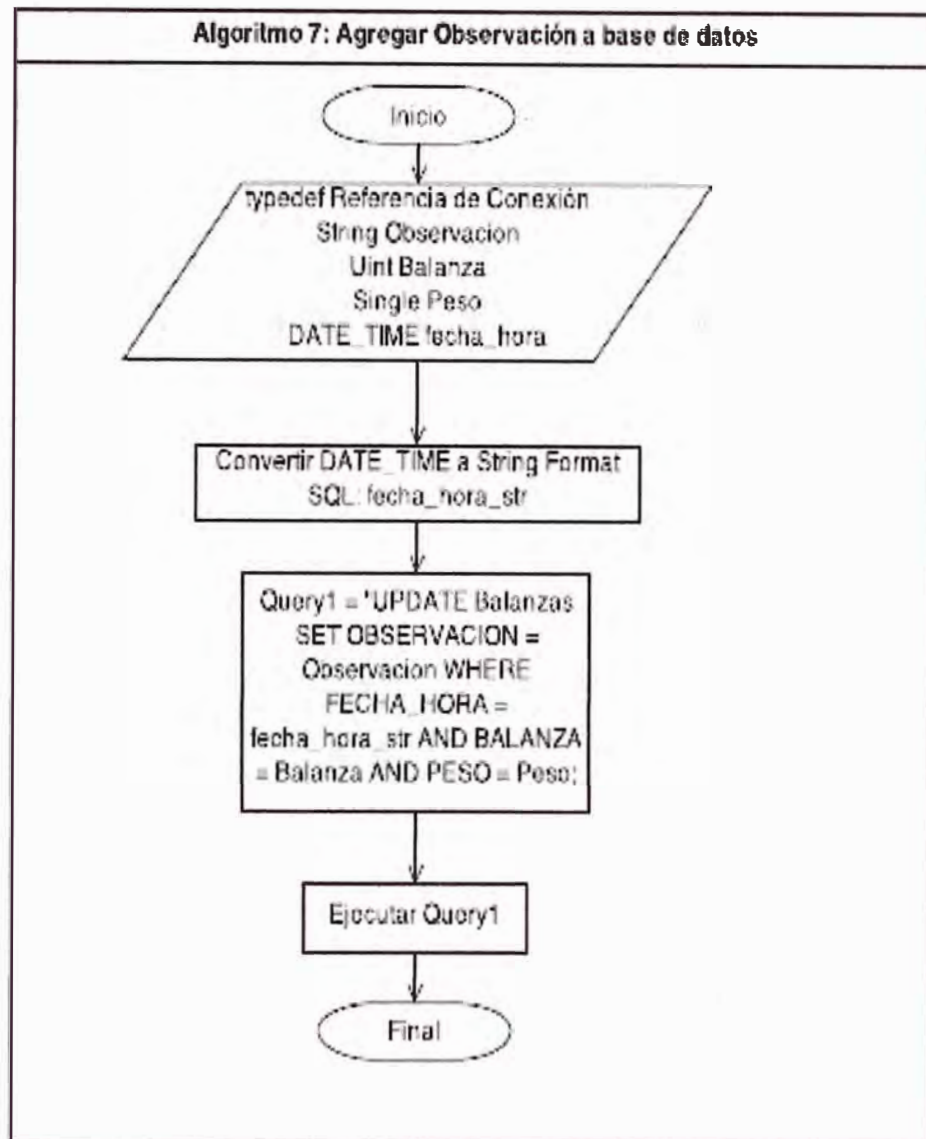


Figura 3.12 – Diagrama de flujo para el ingreso de observación a base de datos.

3.2.5 Eliminación de pesos en la base de datos

A pedido del cliente, la aplicación permite eliminar valores de la base de datos. Para esto se utiliza el comando DELETE de SQL. El algoritmo para la eliminación de pesos de la base de datos es esquematiza en la figura 3.13.

Vale la pena notar que esta parte del código es llamada como respuesta al evento KeyDown de las Tablas que contienen los datos.

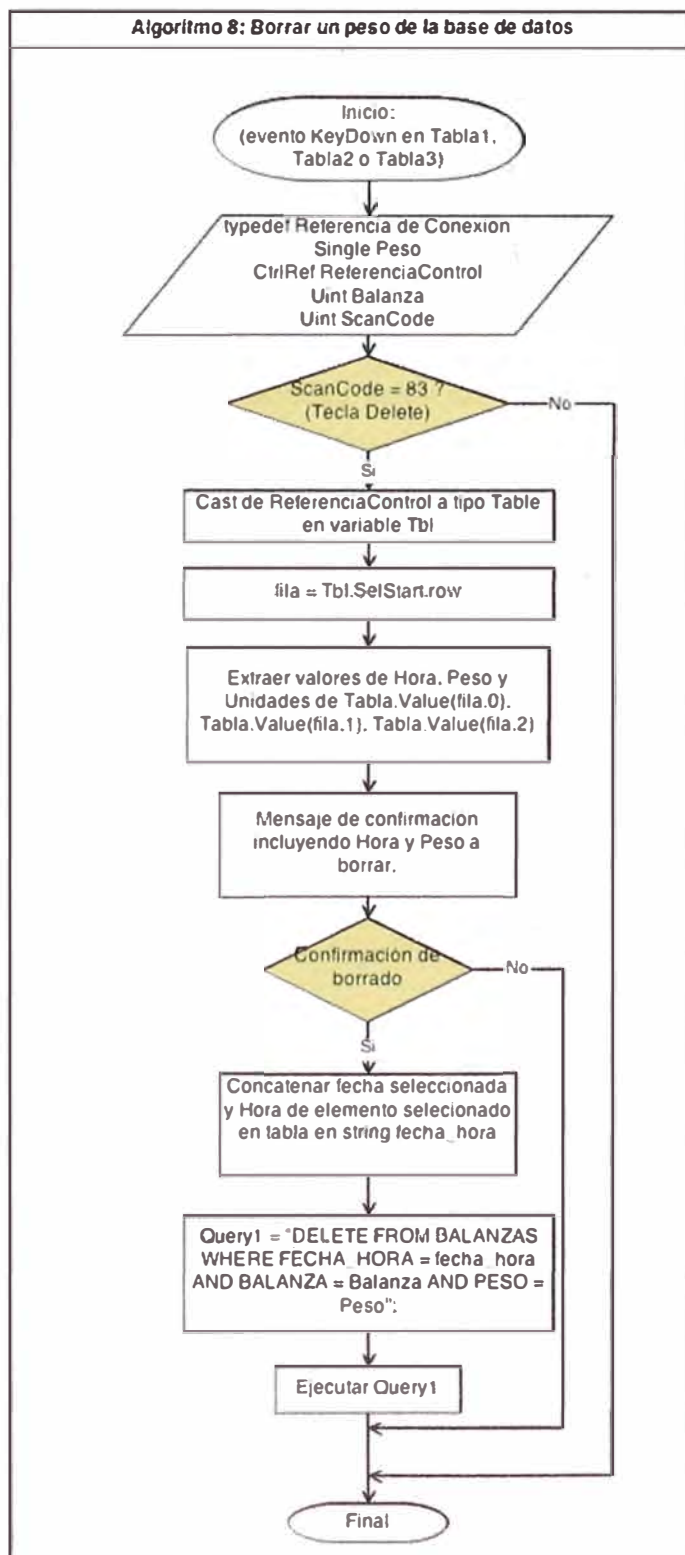


Figura 3.13 – Diagrama de flujo para eliminación de pesos en la base de datos

3.3. Diseño de la interfaz gráfica

3.3.1 Panel principal de la aplicación

El software consta de un panel principal en el cual se visualiza el peso medido por cada una de las balanzas en tiempo real. Además se muestra una tabla por cada balanza, en la cual se agregan el peso de cada saco colocado sobre la balanza una vez que el saco sea retirado de la balanza (esto para permitir al operador de la balanza colocar o retirar el contenido en el saco hasta alcanzar un valor correcto), los valores de peso que estuvieron fuera del rango establecido en el software de configuración son agregados a la tabla y la respectiva fila es coloreada de amarillo para indicar que este valor estuvo fuera de rango. Además en caso de que el operario coloque el saco en la balanza pero lo retire antes de que esta llegue a la condición de peso estable el software registrará este proceso como una operación de mal pesaje y coloreará dicha fila de rojo. (Para que el sistema detecte que el saco fue pesado en la balanza es necesario que el operador luego de colocar el saco en la balanza espere que ésta alcance la condición de equilibrio tal como se especifica en el manual “instrucciones de manejo” de la Terminal de pesada Id3s de Mettler Toledo).

La figura 3.14 muestra el panel principal del software, cada uno de los ítems numerados allí son descritos a continuación:

1. Nombre dado a la balanza para la identificación por parte del usuario.



Figura 3.14 – Panel principal de la aplicación cliente

2. Casilla que permite pausar la grabación de los pesos en la base de datos.
3. Indicadores de los límites y el valor de referencia para el peso de los sacos en la balanza correspondiente. Estos valores determinan si el valor medido por la balanza es considerado un valor fuera de rango o no.
4. Indicador para el valor del peso. El símbolo ~ a la izquierda del valor corresponde a un valor de peso dinámico (no estático).
5. Tabla que muestra el contenido de la base de datos para la balanza correspondiente y para la fecha que se indique en el control de fecha (10). Los valores fuera de rango son coloreados de amarillo, los malos pesajes de rojo y los pesajes correctos mantienen el color gris.
6. Casilla que permite activar la emisión de sonido al realizarse un mal pesaje o un pesaje fuera de rango.

7. Botón para acceder al diálogo de configuración del software.
8. Botón para acceder al diálogo de generación de reportes.
9. Botón para salir del software.
10. Selector de fecha, los datos visualizados en la tabla

corresponden a esa fecha. El botón hoy permite volver a visualizar la fecha actual dándole clic.

11. Indicadores luminosos de valores fuera de rango y malos pesajes, están encendidos mientras exista un pesaje fuera de rango o un mal pesaje, además muestra el número total de cada una de estas.

12. Contiene el total de valores en la tabla.

13. Permite exportar los datos contenidos en la tabla a Excel.

3.3.2 Pantalla de configuración

En esta ventana se ingresan algunos parámetros de configuración para el software. Este archivo de configuración se encuentra protegido por una contraseña, una vez ingresado el usuario y la contraseña se podrá ver y modificar los parámetros de configuración.

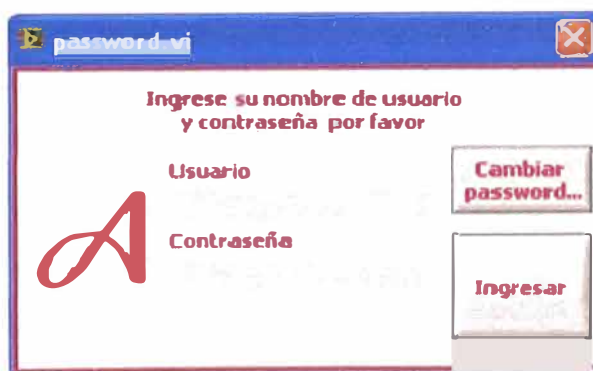


Figura 3.15 – Diálogo para autenticación de usuario y cambio de password.

Botón Ingresar: Una vez colocado un usuario y contraseña correcto permite ingresar al archivo de configuración.

Botón Cambiar password: Este botón abre un nuevo diálogo en el cual se ingresa la nueva contraseña, es necesario colocar previamente el nombre de usuario y la contraseña anterior para abrir la ventana de ingreso de nueva contraseña.

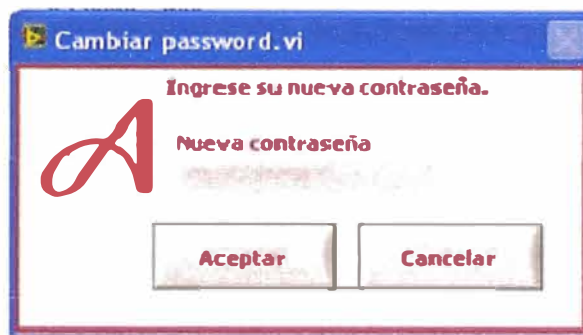


Figura 3.16 – Diálogo para cambio de contraseña

En la ventana de diálogo de configuración tenemos las siguientes opciones:

Balanza seleccionada: Indica a cuál de las tres balanzas corresponden los parámetros que se están visualizando y/o editando. Las balanzas están etiquetadas con los nombres Balanza0, Balanza1 y Balanza2, corresponden a las balanzas que están conectadas a los tres primeros puertos del Nport 5410 instalado en el tablero. El cuarto puerto del Nport no es utilizado.

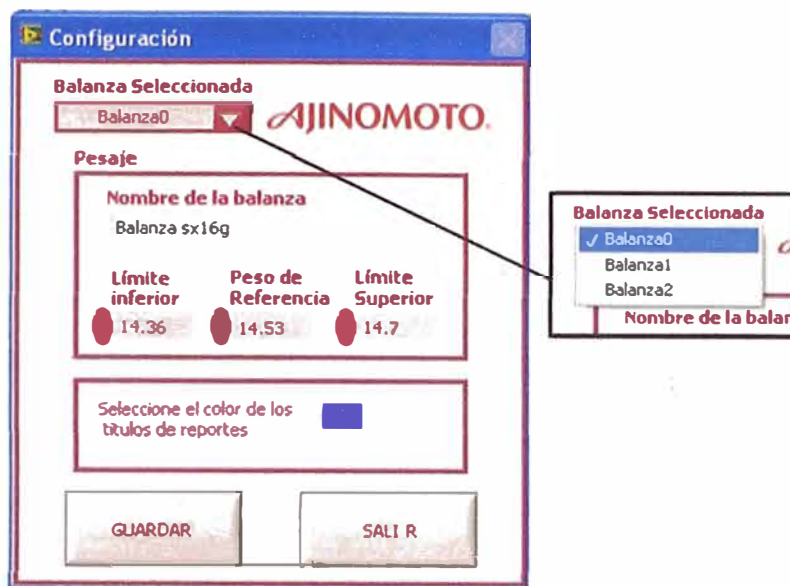


Figura 3.17 – Diálogo para cambios de configuración

Nombre de la balanza: Contiene el texto que será mostrado en el software para identificar a esta balanza. Internamente el software siempre las identificará como Balanza0, Balanza1 y Balanza2. Este texto solo sirve como una referencia para el usuario.

Límite inferior: indica el valor de límite inferior permisible para el peso de saco.

Peso de referencia: Peso de referencia del saco correspondiente a la balanza seleccionada.

Límite superior: indica el valor del límite superior permisible para el peso del saco.

Seleccione el color de los títulos de reportes: Permite seleccionar el color de dichos títulos, permitiendo realizar los reportes en blanco y negro o con títulos del color seleccionado.

Botón guardar: Guardar los parámetros ingresados y los actualiza luego de salir. No es necesario reiniciar el software.

Botón salir: cierra la ventana y actualiza los parámetros en caso hayan sido modificados.

3.3.3 Pantalla de generación de reportes

El diálogo de generación de reportes permite imprimir informes con la información adquirida dentro de las fechas establecidas. Se generará el reporte para cada una de las balanzas seleccionadas en la parte inferior. La casilla de selección "Incluir tablas de información detallada por día" generará en el reporte tablas conteniendo la información de cada uno de los sacos pesados en cada una de las tres balanzas. El reporte será impreso en la impresora configurada por defecto en Windows.

La aplicación permite la generación de reportes para el día especificado, para siete días consecutivos, para un mes o para un rango de días continuos indicados y para las balanzas indicadas. Adicionalmente una casilla de verificación permite activar la generación de reportes separados por día adicionalmente al reporte total.

3.3.4 Manipulación de datos en las tablas

Si desea pausar la grabación en la base de datos basta con activar la casilla "pausar grabación en base de datos" que se encuentra en el panel principal. El peso medido por la balanza seguirá mostrándose en la pantalla pero no se realizará la grabación de estos datos.

Figura 3.18 – Diálogo para generación de reportes

Para eliminar un datos basta con seleccionarlo en la tabla y presionar el botón suprimir (Delete). Se visualizará un mensaje solicitando la confirmación de la eliminación de este dato en la base de datos. De responder afirmativamente el dato será eliminado.

HORA	PESO	Unid	FLAG	OBSERVACION	IMPORTE
09:25:58 a.m.	16.12	kg	Fuera de rango		14.36
09:08:08 a.m.	14.60	kg	Valor correcto		14.36
09:07:56 a.m.	14.55	kg	Valor correcto		14.36
09:07:12 a.m.	14.71	kg	Fuera de rango		14.36
09:06:55 a.m.	14.59	kg	Valor correcto		14.36
09:06:43 a.m.	14.37	kg	Valor correcto		14.36

Figura 3.19 – Eliminación de datos en las tablas

Es posible exportar a Excel la información contenida en la tabla de visualización de datos del software presionando el botón de la parte inferior derecha de la tabla.

09:08:08 a.m.	14.60	kg	Valor correcto	14.36
09:07:56 a.m.	14.55	kg	Valor correcto	14.36
09:07:12 a.m.	14.71	kg	Fuera de rango	14.36
09:06:55 a.m.	14.59	kg	Valor correcto	14.36
09:06:43 a.m.	14.37	kg	Valor correcto	14.36
09:06:27 a.m.	14.52	kg	Valor correcto	14.36

Total de valores pesados: 7



Figura 3.19 – Exportación directa a MS Excel ®

CAPÍTULO IV

IMPLEMENTACIÓN DEL SISTEMA

4.1 Implementación de los algoritmos

La implementación de los algoritmos se realizó utilizando la plataforma de desarrollo Labview® de la empresa National Instruments. El lenguaje utilizado por esta plataforma es un lenguaje gráfico, en el cual las funciones, constantes, variables y otros son representados como bloques.

La principal razón por la cual se seleccionó esta plataforma de desarrollo frente a otras posibles fue el hecho de contar con las licencias para esta y todos sus kits de herramientas de desarrollo de manera gratuita, al ser representantes en el país de la empresa National Instruments, propietarios de esta herramienta de desarrollo. Adicionalmente, esta plataforma provee herramientas y funciones de alto nivel, las cuales permiten realizar una implementación rápida de soluciones que pueden tomar un tiempo mucho mayor en otras plataformas orientadas a programación de más bajo nivel. Otro punto a favor reside en la facilidad de depurar el programa y detectar errores de diversos tipos, debido a que la programación es gráfica, las herramientas de depuración permiten visualizar paso a paso como los datos son transferidos de un bloque a otro, lo cual permite depurar el programa de manera semejante a como uno lo ve en un diagrama de flujo.



Figura 4.1 – Proyecto en Labview

4.1.1 Proyecto en Labview

Para facilitar el orden y organización en la implementación del software se manejó este como un proyecto en la interfaz de programación de Labview. Esto permite tener una visión ordenada de todos los componentes utilizados y facilita también luego la utilización de la herramienta de Labview que permite la creación de archivos de instalación en base al contenido del proyecto. El contenido de archivos del proyecto en Labview se muestra en la figura 4.1.

4.1.2 Inicialización de parámetros

Uno de los primeros pasos ejecutados en el programa consiste en la inicialización de los valores de diversas variables y formatos utilizados en el programa principal, llamado "principal.vi". Algunos de estos parámetros, como los límites de pesos, son leídos del archivo cnfg.ini y otros son constantes. La figura 4.2 muestra algunos ejemplos de inicialización en Labview, en la parte superior para los colores en las tablas utilizadas para mostrar los pesos históricos, y en la segunda para la inicialización de los indicadores de límites de peso y de referencia de peso.

4.1.3 Conexión de datos hacia los terminales de pesaje

La figura 4.3 muestra el código en Labview para la apertura de conexión hacia el conversor seria-Ethernet, el cual permitirá al programa comunicarse de manera transparente con los terminales de pesaje. El primer paso que realizamos para la conexión hacia los terminales de pesajes es leer la dirección IP del conversor, la cual está grabada en el archivo IP.txt. El bloque "Read Character from file" de Labview simplifica esta tarea.

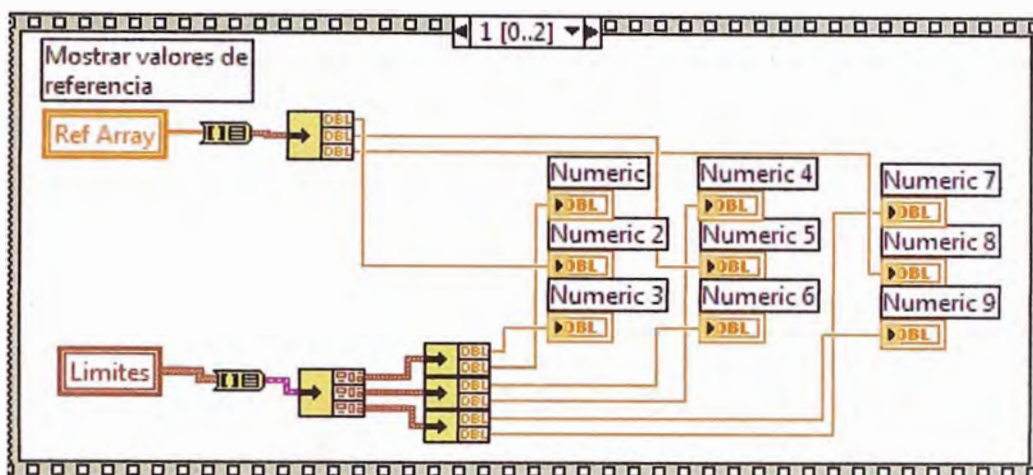
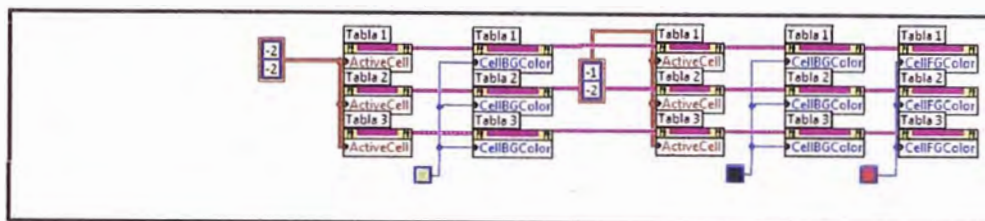


Figura 4.2 – Ejemplos de inicializaciones de variables en el programa en Labview

A continuación se intenta la conexión a cada una de las tres balanzas al IP leído previamente y a los puertos TCP número 4001, 4002 y 4003 utilizando el bloque “TCP Open Connection”, con un timeout de 5000 ms. El código de error obtenido en la salida “error out” de este bloque es luego evaluado; si el código es igual a 56 significa se venció el tiempo límite y el software manda el mensaje correspondiente y pregunta al usuario si desea reintentar la conexión, de ser negativa la respuesta se aborta la aplicación, caso contrario se intentará nuevamente.

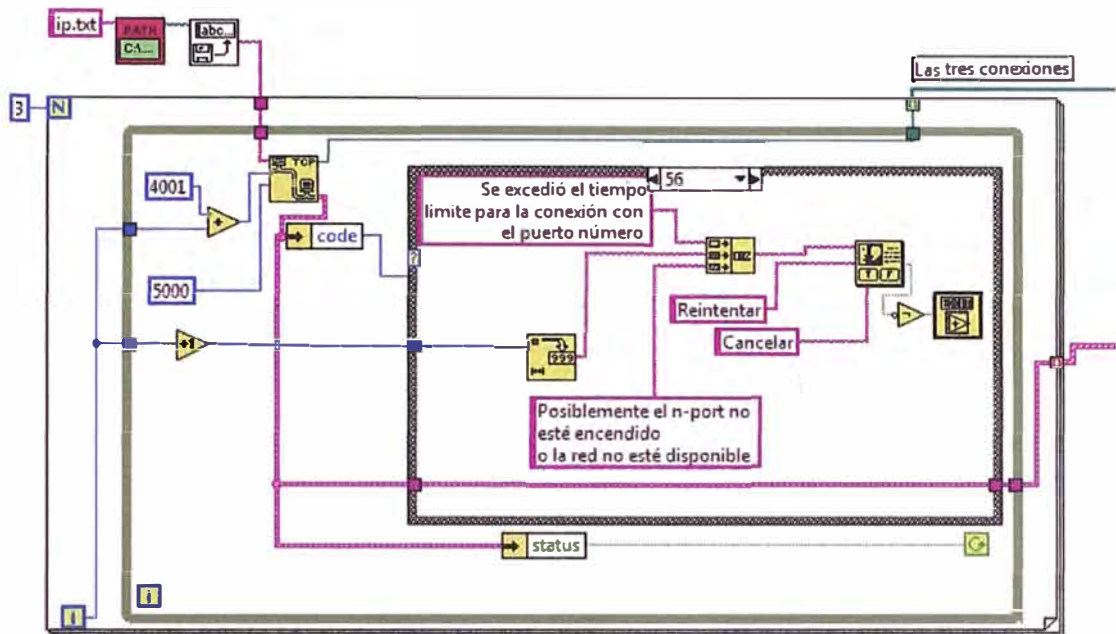


Figura 4.3 – Conexiones TCP a equipo conversor Serial-Ethernet

Si el conversor serial-Ethernet está operando correctamente y es alcanzable en la red la conexión a los tres puertos de software debería ser exitosa independientemente de si la balanza está físicamente conectada o no.

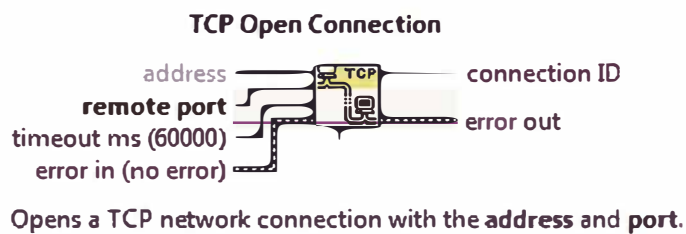


Figura 4.4 – Bloque “TCP Open Connection” de Labview

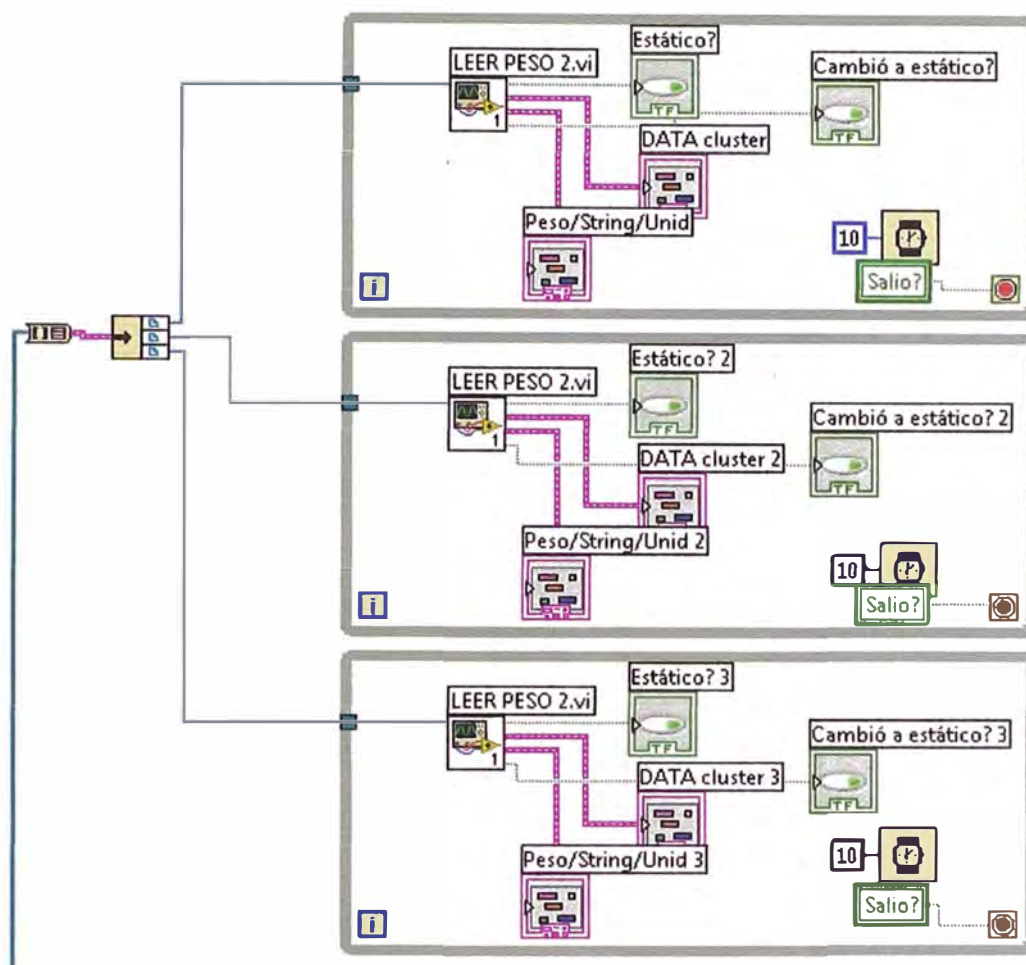


Figura 4.5 – Lazos while para lectura de datos a cada una de las tres balanzas.

Una vez las conexiones TCP al convertidor serial-Ethernet han sido abiertas, los identificadores de conexión son enviados a otras etapas del programa para la lectura de datos.

4.1.4 Lectura de pesos desde las terminales de pesaje

La rutina “Leer peso 2.vi” es utilizada dentro de los lazos while de la figura 4.5 para leer los pesos de las balanzas. La figura 4.6 muestra la implementación en Labview de esta rutina “Leer peso 2.vi”.

Esta rutina escribe el comando SI seguido por los caracteres “carrier return” y “line feed”, en la conexión TCP previamente abierta hacia el conversor serial-Ethernet. Este comando será enviado hacia el puerto serial correspondiente al puerto TCP de esta conexión el cual está conectado a uno de los tres terminales de pesaje. Como se explicó en el subcapítulo 3.1.1, el terminal de pesaje responderá al comando “SI” enviando el valor de peso actual, ya sea este estático o dinámico según el formato mostrado en la tabla 3.3.

Luego de enviar el comando SI se leerá la respuesta TCP la cual según el formato de respuesta del comando SI está marcado por los caracteres CR (carriage return) y LF (line feed). Por este motivo el bloque “TCP read” es usado en modo CRLF.

Los primeros 3 caracteres de la trama de respuesta son utilizados en un lazo Case en Labview, como se muestra en la figura 4.6. Si estos caracteres corresponden a “S “ o “SD “el terminal de pesaje nos está indicando que prosigue un peso estático y uno dinámico respectivamente, y la variable booleana “Estático?” será asignada correspondientemente.

Los datos de la trama serán seguidamente segmentados, el peso convertido a formato de doble precisión, y los campos unidos en dos clusters de datos de Labview. El primero de ellos llamado “DATA cluster” conteniendo la fecha, hora, peso y unidades, y otro llamado “Peso/String/Unid” conteniendo el peso en formato número, el peso en formato string incluyendo

el símbolo de indicación de peso dinámico y las unidades. Los cluster de datos permiten encapsular datos de distintos tipos en una sola estructura para luego extraer estos según sea necesario en otra parte del programa.

4.1.5 Generación de reportes

Para la generación de reportes se creó una aplicación separada, de tal manera que esta pueda ser ejecutada sin necesidad de abrir la aplicación que realiza la adquisición de datos de los terminales de pesaje

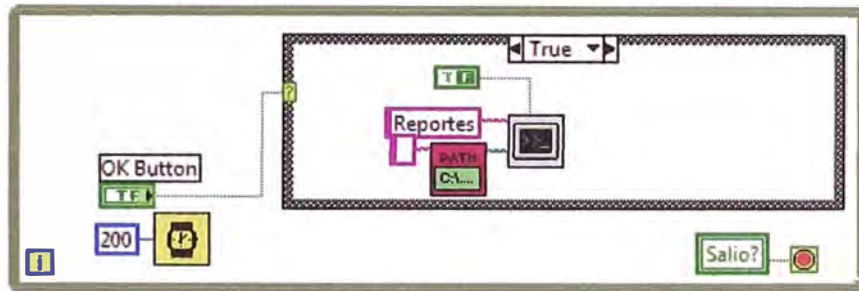


Figura 4.7 – Llamada a aplicación de reportes

4.1.6 Generación de mensajes y sonidos de alerta.

De ser identificado un mal pesajes o un pesaje fuera de rango, la aplicación mostrará los mensajes correspondientes y ejecutará un sonido previamente establecido cada 1.5 segundos. El sonido permanecerá hasta que el usuario reconozca el mensaje dando clic en el botón OK de la ventana de diálogo que contiene dichos mensajes. La figura 4.8 Muestra tres imágenes de lazos while en Labview, las dos superiores corresponden al mismo lazo pero muestran el contenido de los dos "stacked sequence" con valores 0 y 1. Las variables booleanas "FDR?" y "MP?" estarán en true si ha sucedido un pesaje fuera de rango o un mal pesaje y este aún no ha sido reconocido por el usuario. Por otro lado, el lazo while mostrado en la parte

inferior esta misma figura reproduce los sonidos en formato .wav según corresponda, para un mal pesaje o un pesaje fuera de rango.

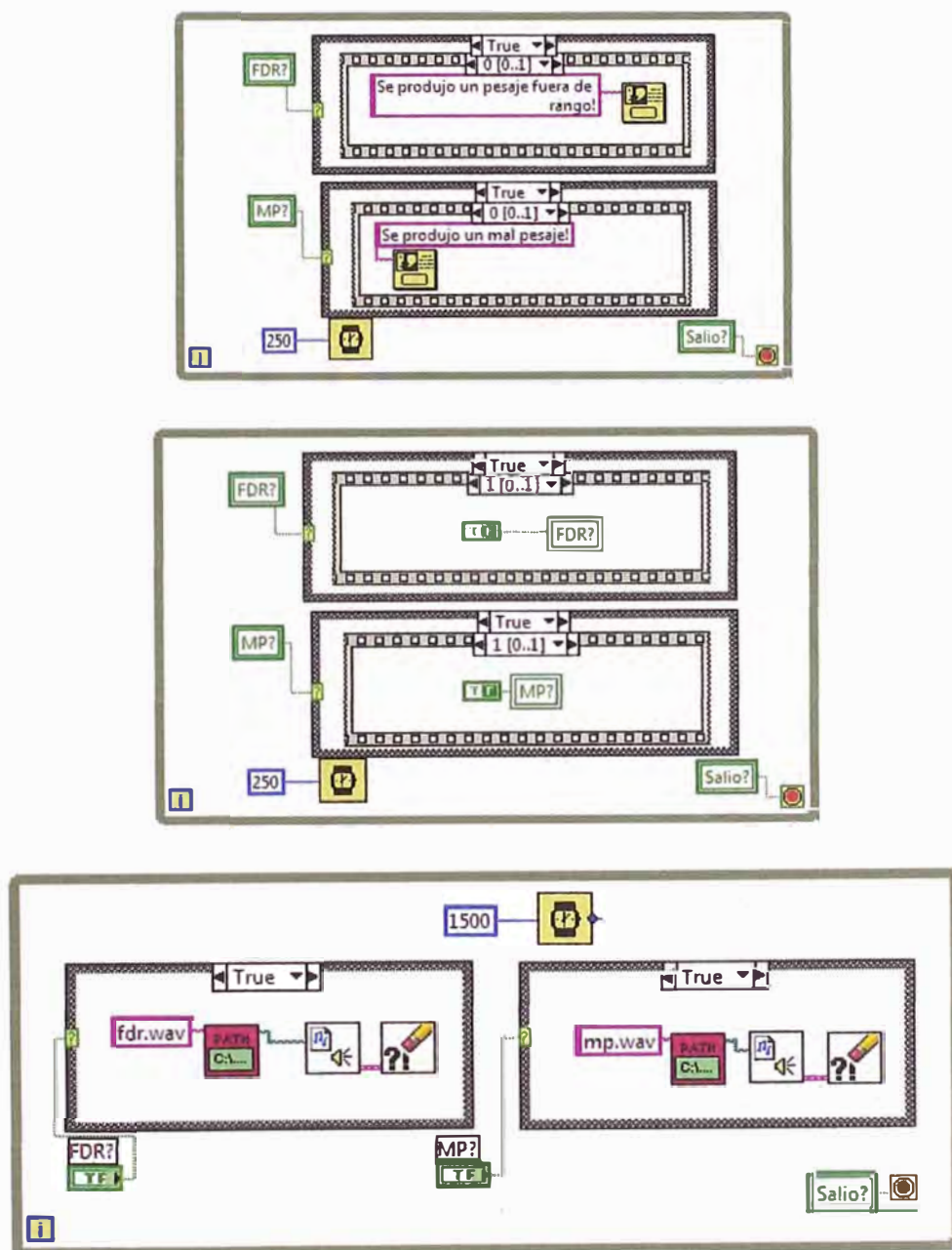


Figura 4.8 – programación de mensajes y sonido de alerta para pesajes fuera de rango y malos pesajes.

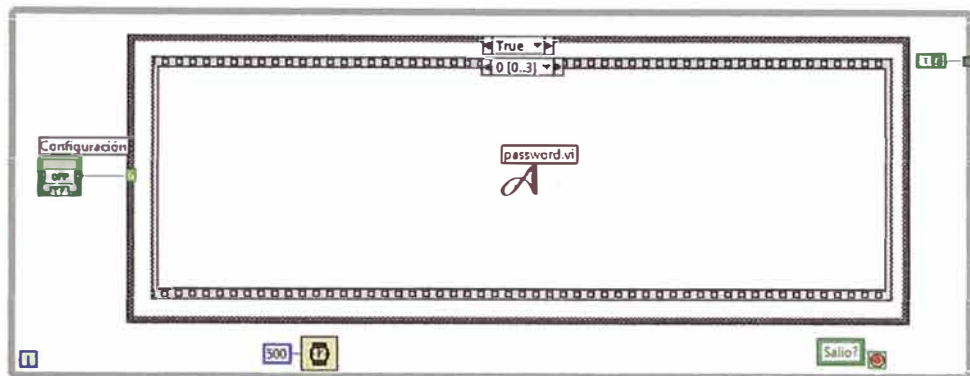


Figura 4.9 – programa de acceso de configuración; llamada a password.vi.

4.1.7 Configuración del sistema.

El diálogo de acceso para la configuración del sistema se obtiene dando clic en el botón 7 de la figura 3.14. Este botón es el marcado en el software con la variable booleana "Configuración". Sin embargo, previamente el código contenido en el programa "password.vi" es llamado al ejecutar esta opción, como muestra la figura 4.9. El programa principal responderá a la activación del botón "Configuración" llamando a una estructura "stacked sequence" la cual primeramente ejecutará el programa password.vi, seguidamente el programa Leer Config Bal.vi, posteriormente cargará los valores de referencias y límites en las variable e indicadores correspondientes en el programa principal, y finalmente cargará los nombres de cada producto envasado en cada una de las tres balanzas, según se muestra en las figuras 4.9 y 4.1, las cuales permiten apreciar el contenido de cada una de las cuatro etapas del "stacked sequence" de Labview.

En el primer paso del "stack sequence" el programa password.vi ejecuta el diálogo para el registro de usuario y password previo al acceso al programa de configuración.

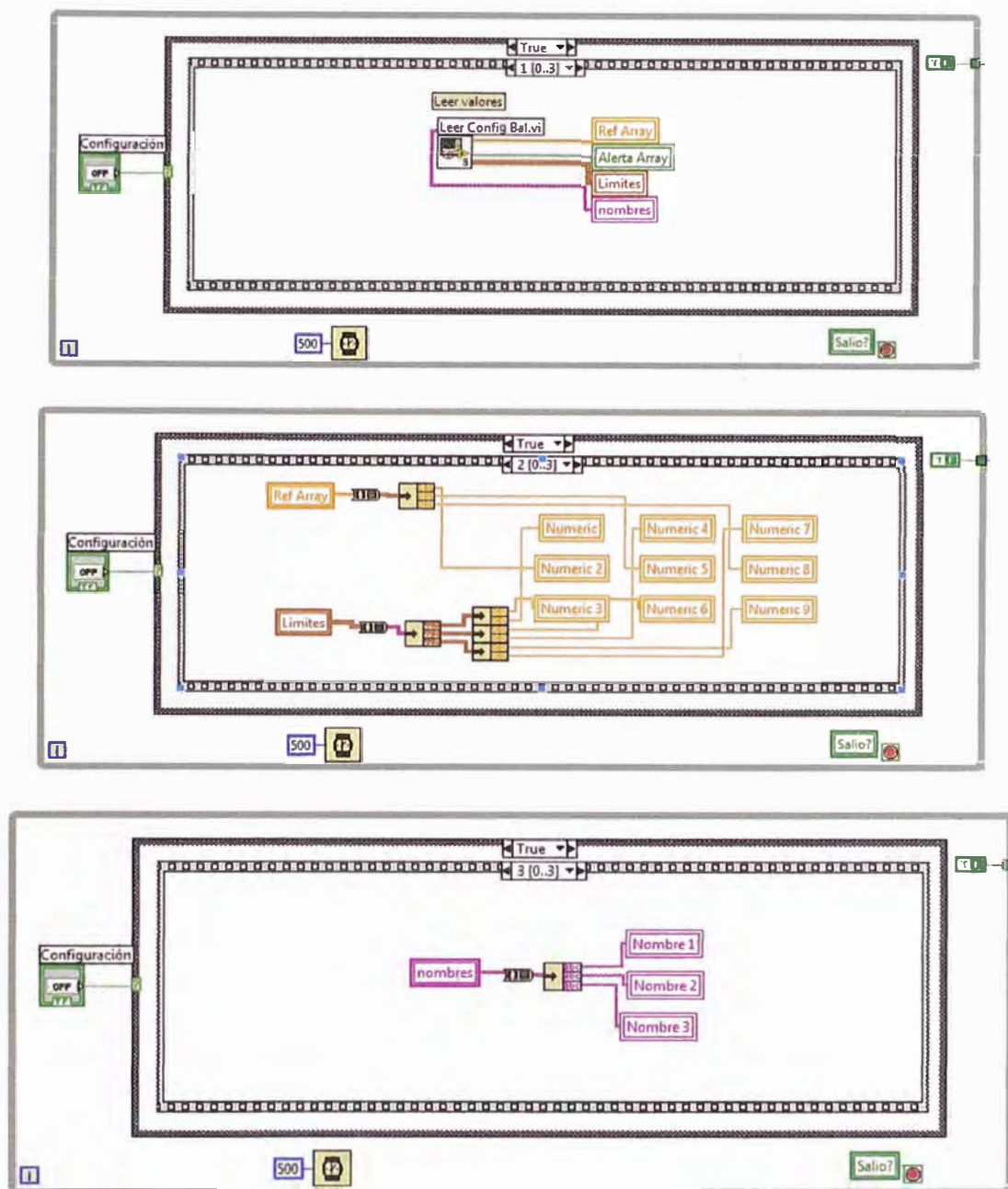


Figura 4.10 – Ejecuciones posteriores a llamada a password.vi.

El usuario y clave de acceso no tuvieron como fin principal proveer de un alto nivel de seguridad para provenir algún tipo de acceso malicioso al sistema, si no que su principal objetivo fue evitar que algún usuario, por falta de conocimiento, pudiera alterar parámetros erróneamente; por este motivo, y a pedido del cliente, el usuario y la clave de acceso fueron guardados en un

archivo de texto plano llamado “up.upx” en la carpeta de instalación del programa sin mayor encriptación, lo cual facilitaría al personal de soporte de tecnologías de la información la recuperación de la clave si fuera necesario.

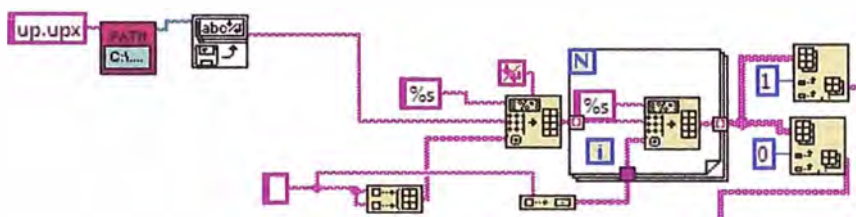


Figura 4.11 – Lectura de usuario y password del acceso a la configuración del sistema.

El programa password.vi permitirá el acceso al programa “Configuración 2.vi” sólo al tener éxito en el ingreso del usuario y password correcto. La ventana de diálogo de configuración es la mostrada en la figura 3.17.

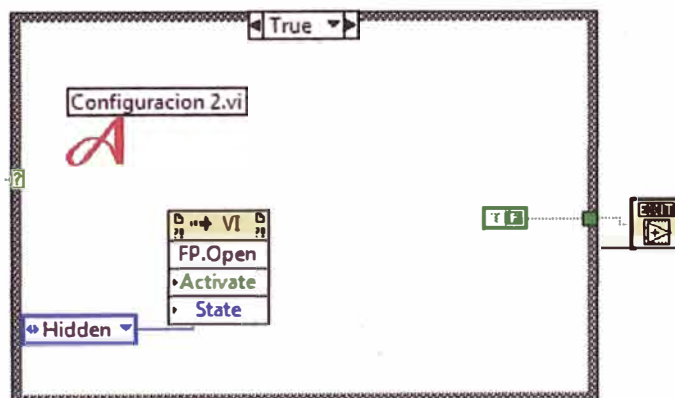
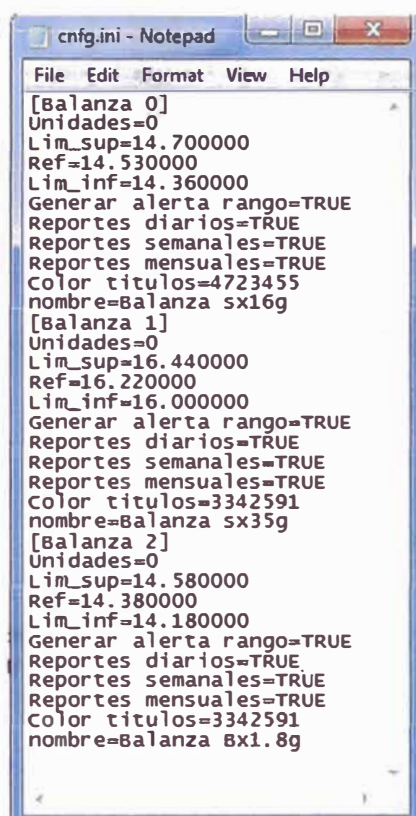


Figura 4.12 – Llamada a programa Configuración.vi por password.vi

El programa “Configuración 2.vi” lee la configuración actual guardada en el archivo cnfg.ini, el cual contiene los parámetros de configuración como se

muestra en la figura 4.13, en el formato standard de Windows para archivos .ini,



```
cnfg.ini - Notepad
File Edit Format View Help
[Balanza 0]
Unidades=0
Lim_sup=14.700000
Ref=14.530000
Lim_inf=14.360000
Generar alerta rango=TRUE
Reportes diarios=TRUE
Reportes semanales=TRUE
Reportes mensuales=TRUE
Color titulos=4723455
nombre=Balanza sx16g
[Balanza 1]
Unidades=0
Lim_sup=16.440000
Ref=16.220000
Lim_inf=16.000000
Generar alerta rango=TRUE
Reportes diarios=TRUE
Reportes semanales=TRUE
Reportes mensuales=TRUE
Color titulos=3342591
nombre=Balanza sx35g
[Balanza 2]
Unidades=0
Lim_sup=14.580000
Ref=14.380000
Lim_inf=14.180000
Generar alerta rango=TRUE
Reportes diarios=TRUE
Reportes semanales=TRUE
Reportes mensuales=TRUE
Color titulos=3342591
nombre=Balanza Bx1.8g
```

Figura 4.13 – Contenido del archivo de configuración cnfg.ini

Para la lectura y escritura de datos en el archivo cnfg.ini se crearon los programas “LEER CONFIG.vi” y “Guardar config.vi”. Los cuales son llamados dentro del programa “Configuración 2.vi” para la lectura y escritura de datos, los cuales con encapsulados en este programa en un array unidimensional de clusters. Estos cluster contienen los datos mostrados en la figura 4.14.

Si el archivo no existe o no puede ser leído el programa cargará unos valores por defecto los cuales fueron provistos por el cliente y se muestran en la figura 4.15.

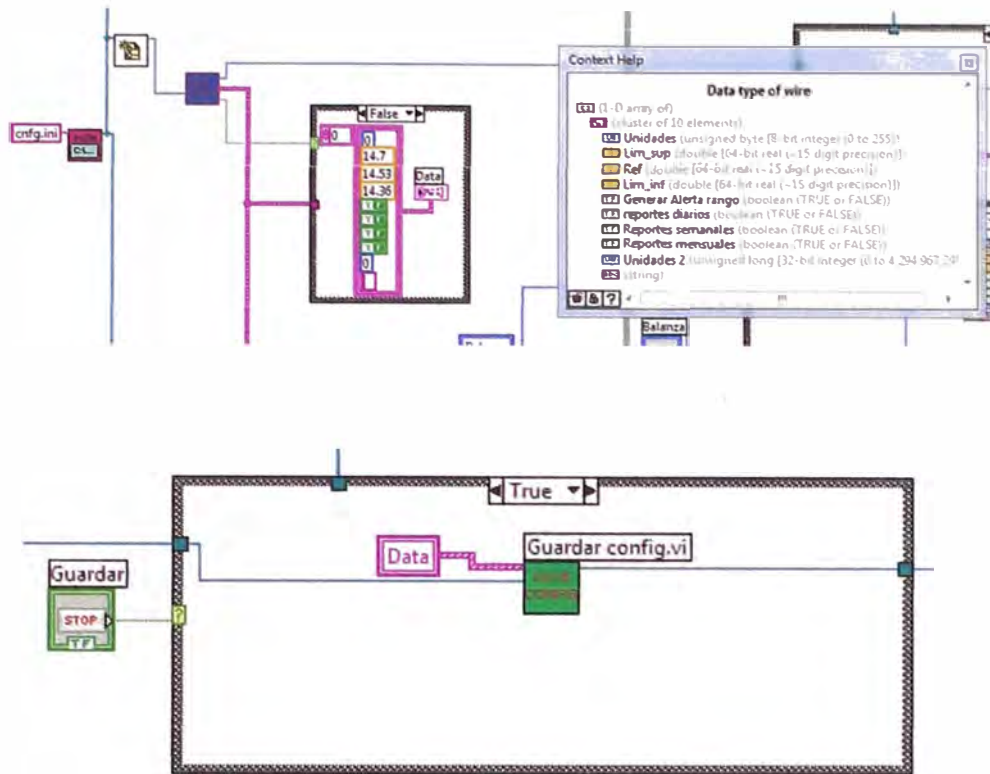


Figura 4.14 –Contenido de cluster Data y llamada a programa “guardar config.vi”

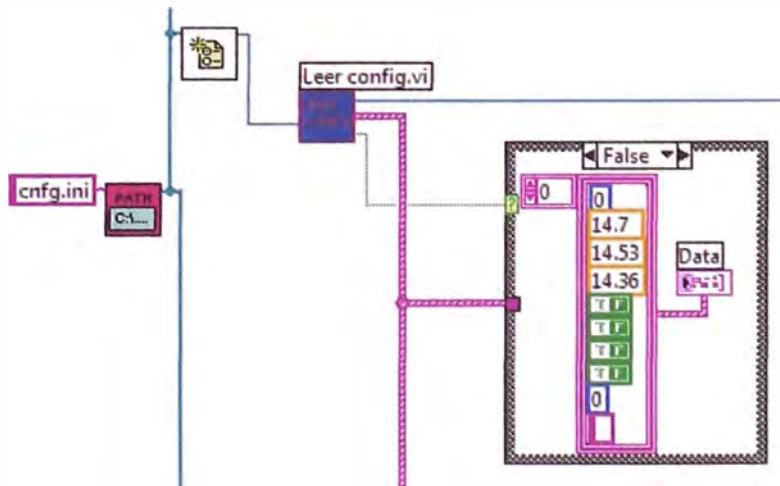


Figura 4.15 – inicialización por defecto de configuración

Los programas “LEER CONFIG.vi” y “Guardar config.vi” realizan las manipulaciones y conversiones necesarias usando los bloques de lectura y escritura de archivos .ini de Labview y otros bloques de manipulación de

datos, para convertir el contenido del archivo .ini en un array unidimensional de clusters. Cada elemento de este array es un cluster que contiene los datos correspondientes a una de las tres balanzas. La figuras 4.16 y 4.17 muestran el código de estos programas.

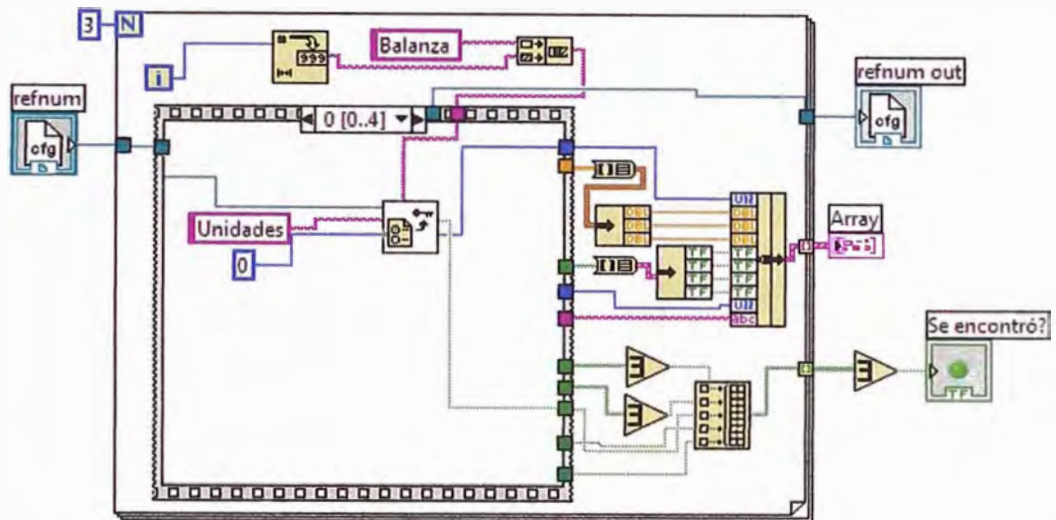


Figura 4.16 – programa “leer config.vi”

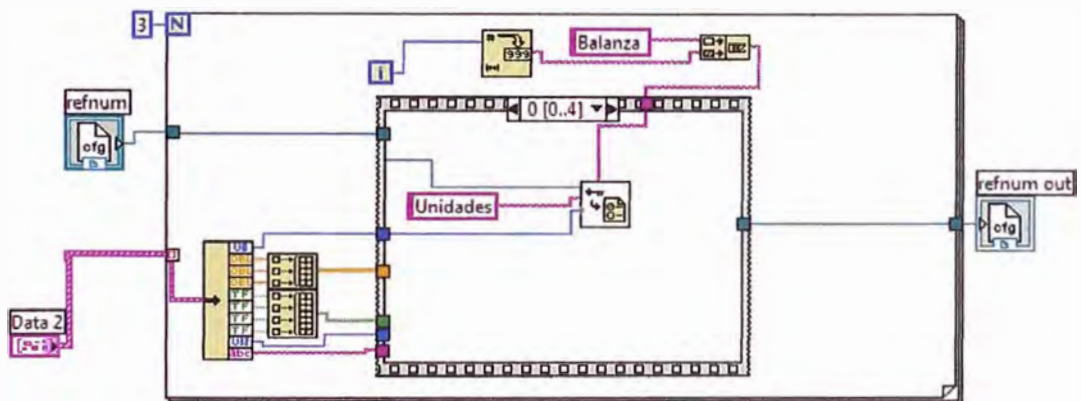


Figura 4.17 – programa “guardar config.vi”

4.1.8 Grabar datos en base de datos

El programa “GUARDAR BASE DATOS.vi” se encargará de la grabación de datos en la base de datos. Este bloque de programa implementa el

4.1.9 Actualización de datos en las tablas

Un lazo en el programa principal tiene como función principal leer los datos de la base de datos y actualizarlos en las tablas mostradas en la interfaz de usuario. Este lazo se muestra en la Figura 4.20. El programa fue desarrollado de tal manera que los valores que se muestran en pantalla sean los valores que han sido ya grabados en la base de datos

El lazo previamente mencionado evalúa primero si hubo un cambio en la fecha seleccionada, de ser así, inicializa los colores de fondo de cada una de las tablas y de sus cabeceras al valor iniciales. La selección de fecha se efectúa con el control marcado con el número 10 en la Figura 3.15. Posteriormente el software realiza un lazo for para cada una de las tres balanzas, en cada iteración llama al programa “LEER TABLA.vi” el cual fue creado para devolver el conjunto de datos de un solo día para una balanza especificada. Las salidas y entradas del bloque “LEER TABLA.vi” son mostradas en la Figura 4.19. Las salidas son luego convertidas en un array de strings en dos dimensiones el cual será cargado a la tabla correspondiente.

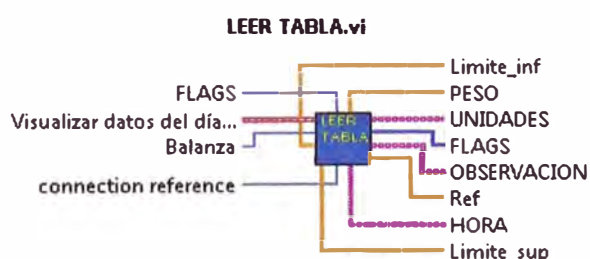


Figura 4.19 – entradas y salidas del bloque de programa “LEER TABLA.vi”

Otra tarea que se realiza en este lazo es evaluar si la conexión con la base de datos se ha perdido. Esto se efectúa llamado al bloque “List Tables” que listará las tablas incluidas en la base de datos (en este caso solo una) y

evaluando el código de error de respuesta. Si el status de error está en true, se mostrará una indicación al usuario.

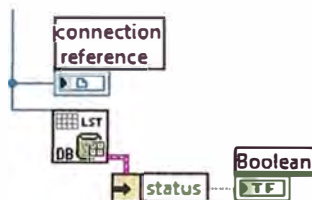


Figura 4.21 – Estado de conexión a base de datos.

Existe además otro lazo for internamente que indexa los arrays devueltos por el bloque "LEER_TABLA.vi". En esta lazo se realiza el conteo del total de pesos fuera de rango, del total de malos pesajes, y el total de bueno pesajes, en las variables Tot_fdr1, Tot_mp1 y Tot_vc1 respectivamente para la balanza 1, Tot_fdr2, Tot_mp2 y Tot_vc2 para la balanza 2, y Tot_fdr3, Tot_mp3 y Tot_vc3 respectivamente para la balanza 3. Adicionalmente las variables booleanas fdr1 y mdr1 son usadas para mostrar una indicación en pantalla cuando se tiene algún mal pesaje o pesaje fuera de rango. Estos indicadores están señalados con el número 11 en la Figura 3.14. La otra función realizada dentro de este lazo for es el coloreo de las filas en función a si estas contienen un buen pesaje (plomo), un pesaje fuera de rango (amarillo) o un mal pesaje (rojo). Esto se realiza utilizando los métodos ActiveCell y CellBGColor para seleccionar y colorear el fondo de las celdas, respectivamente. Todo esto se muestra igualmente en la misma Figura 4.20.

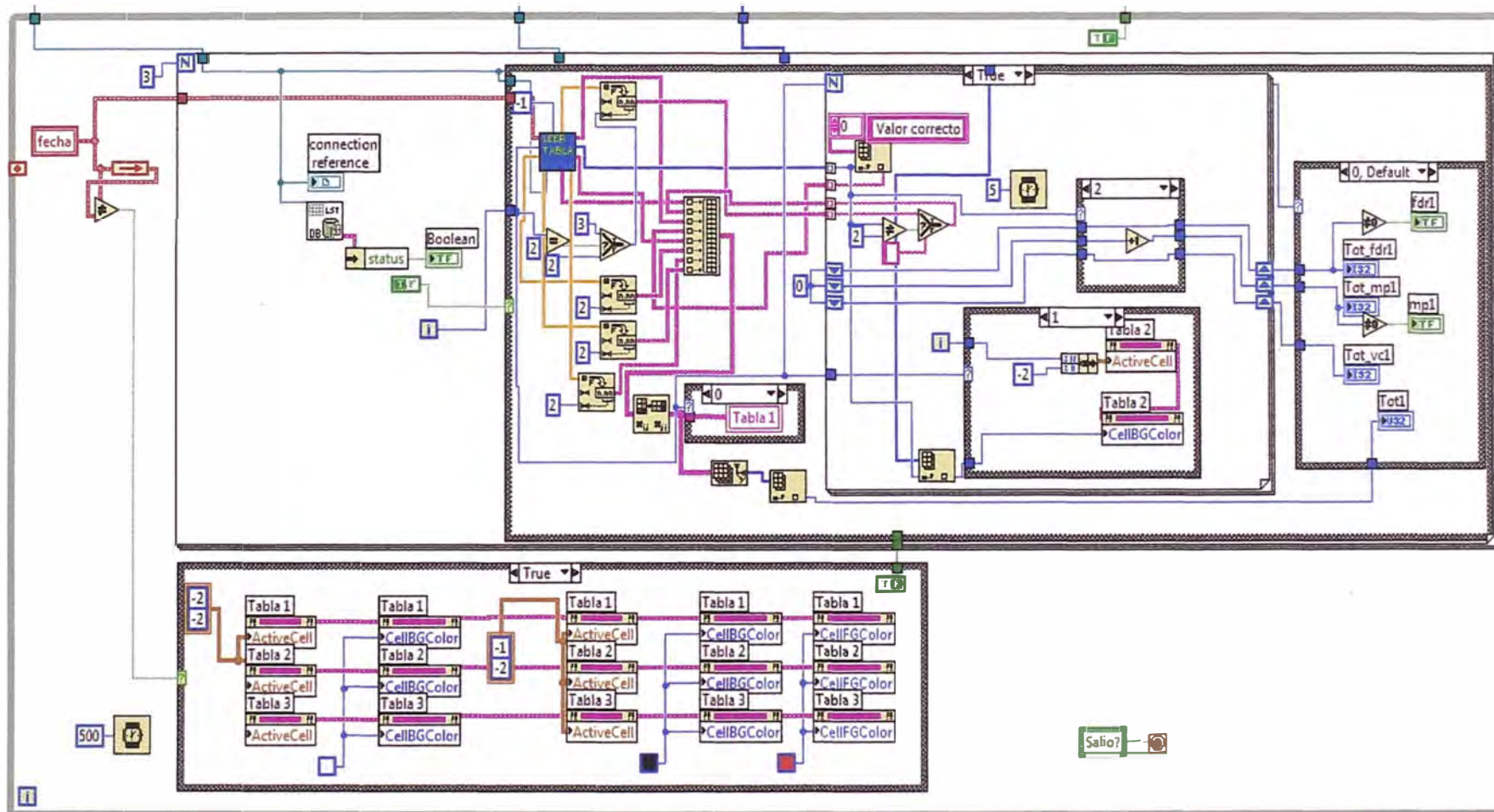


Figura 4.20 – Lazo de actualización de datos en las tablas de cada balanza

4.1.10 Lectura de pesos desde la base de datos

El programa "LEER TABLA.vi", mostrado en la Figura 4.22 implementa la lectura de datos desde la base de datos SQL. Este programa implementa el algoritmo seis descrito en el capítulo III. En este programa se envía el query SQL a la base de datos mediante los bloques de Labview "Execute Query" y se recibe la data usando el bloque "Fetch Recordset Data". La data es recibida en el formato Variant de Labview, para luego ser convertida a los tipos de datos correspondientes.

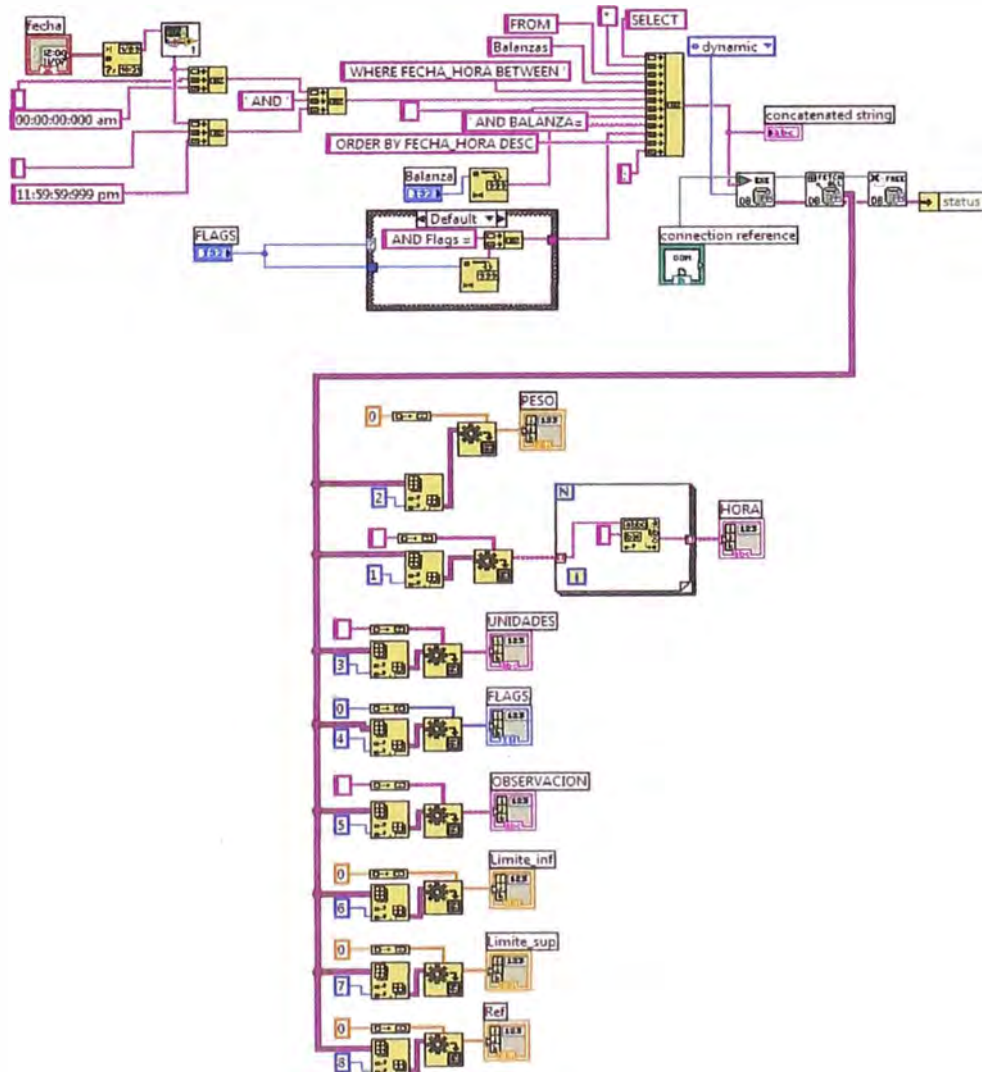


Figura 4.22 – programa para leer tabla de base de datos.

4.1.11 Reacción a eventos de teclado en las tablas

Algunas funciones están disponibles al presionar algunas teclas mientras se selecciona alguna fila en las tablas de la pantalla principal de la interfaz gráfica del software. El lazo mostrado en la Figura 4.23 contiene una estructura de manejo de eventos de Labview, el cual está configurado para atender eventos “keydown” enviados por el sistema operativo cuando estos suceden sobre las tablas previamente mencionadas. La variable CtlRef contendrá la referencia del indicador o control del cual proviene el evento, y el valor ScanCode contendrá el código único de la tecla presionada.

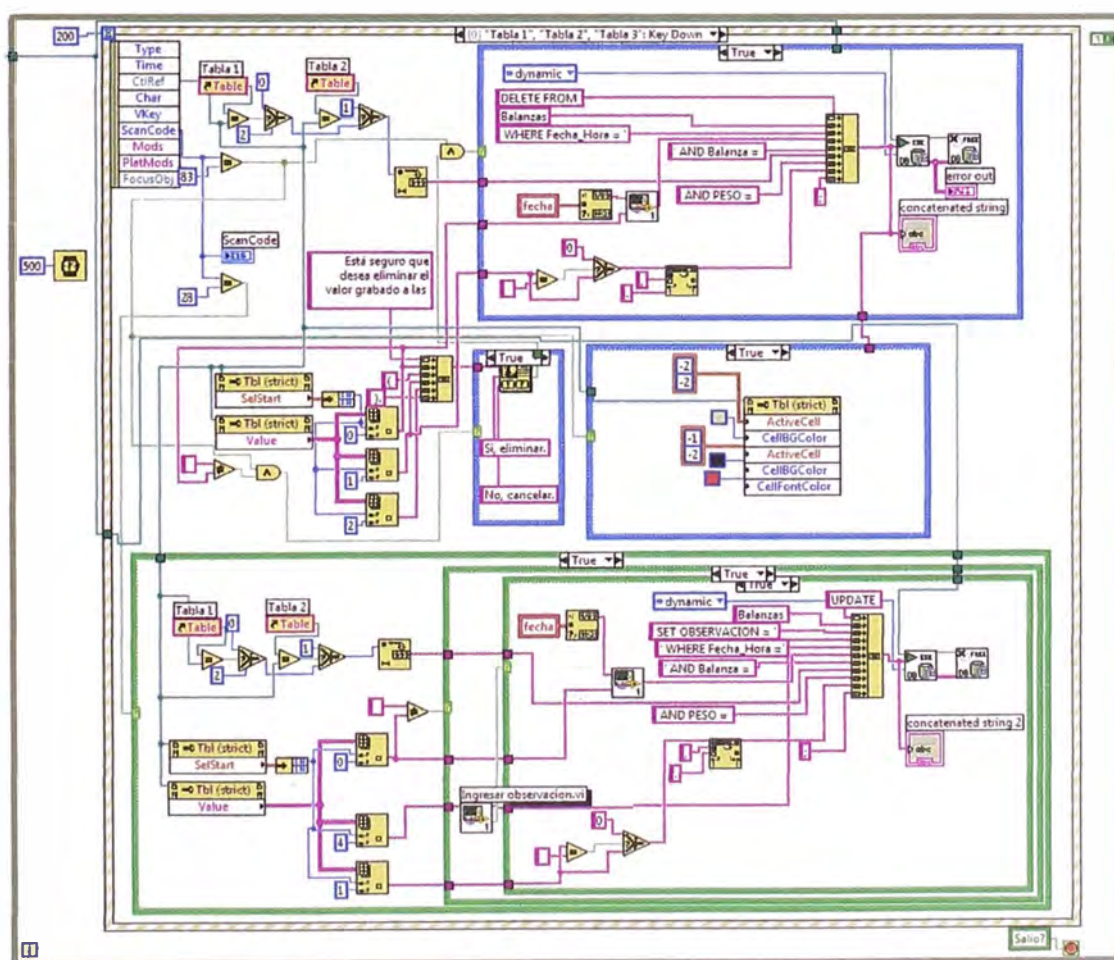


Figura 4.23 – Lazo de atención a eventos “key down” en tablas

Se evalúan dos posibilidades: presionar la tecla delete que corresponde al ScanCode 83, y presionar la tecla Enter que corresponde al ScanCode 28.

4.1.11.1 Ingreso de comentario u observaciones relacionados a un pesaje

La respuesta a la tecla enter está manejada por los tres lazos if marcados con color verde en la Figura 4.23. En esta parte del código se le permite al usuario introducir un comentario asociado al peso seleccionado en la tabla. Para esto se evalúa que el contenido de la línea seleccionada no sea nulo, es decir que contenga datos relacionados a un peso realizado. Si el contenido no es nulo se llama al bloque "Ingresar Observacion.vi", el cual muestra el diálogo de ingreso de observación mostrado en la Figura 4.24 y devuelve el texto correspondiente así como un valor booleano que indica si se debe actualizar la observación en la base de datos. De ser necesario actualizarlo se creará el query correspondiente para realizar la actualización en la base de datos usando el comando SQL Update.



The image shows a dialog box with a red border. At the top, it contains the text "Ingresar observación (máximo 200 caracteres):" in red. Below this text is a large, empty text input field. To the right of the input field are two buttons: "OK" and "CANCELAR", both in red text on a light background.

Figura 4.24 – Diálogo para ingreso de observaciones

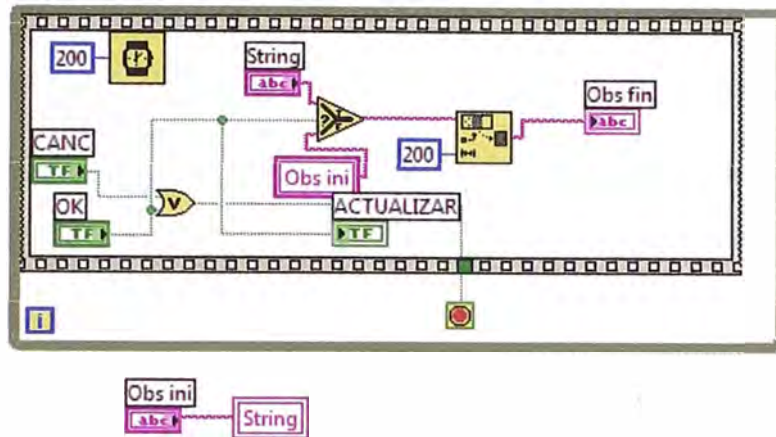


Figura 4.25 – Ingreso o actualización de observaciones.

4.1.11.2 Eliminación de pesajes de la base de datos.

La respuesta a la tecla delete es manejada por los lazos marcados en color azul en la Figura 4.23. En este caso, siempre y cuando el contenido no sea nulo, se pedirá al usuario confirmar que desea eliminar el dato que se tiene seleccionado mediante un mensaje, el cual incluirá los valores de peso y hora que para evitar errores. Si la respuesta es afirmativa se construirá el query adecuado para eliminar este dato de la base de datos usando el comando SQL delete. A continuación se reinicializará el color de fondo de la tabla al gris por defecto y la cabecera a los colores por defecto. En la siguiente lectura de datos de la base de datos el software refrescará el contenido de la tabla y coloreará nuevamente todos los malos pesajes o pesajes fuera de rango según corresponda.

4.1.12 Clasificación de pesajes

El lazo mostrado en la Figura 4.26 se encarga de analizar los pesajes y clasificarlos en relación a si fueron un mal pesaje, un pesaje correcto, o un pesaje incorrecto. Previamente a la ejecución del lazo se abre una conexión a

la base de datos, para esto lee el contenido del archivo `usr_pwd_ODBC.txt`, el cual contiene el usuario y password para la conexión a la base de datos. Seguidamente usa esta información para abrir la conexión a la base de datos y de haber algún error muestra el mensaje de error recibido al usuario. Esta parte inicial de apertura de la conexión implementa el algoritmo cuatro, descrito en el capítulo III de este informe.

Como es usual en el código, se utiliza un lazo `for` internamente para actuar sobre cada una de las variables relacionadas a cada una de los tres terminales de pesaje. Las variables "Peso/String/Unid", "Cambió a Estático?", "Estático?", y "DATA cluster", así como sus correspondientes para las balanzas 2 y 3, contienen la información relacionada al último pesaje recibido de cada balanza y leído por el bloque "LEER PESO 2 " mostrado en la Figura 4.6, como se explicó ya previamente en la Sección 4.1.3 de este informe. Se usa el lazo `case` marcado de color naranja para acceder al grupo de variables correspondiente a cada una de las tres balanzas. Otras variables son guardadas como arrays con tres elementos, cada uno relacionado a una de las tres balanzas.

En este lazo se utilizan varios "feedback node" los cuales funcionan en Labview como un buffer que permite acceder fácilmente al valor de una variable en la iteración previa.

El valor de "Ref Array" contiene la referencia de peso para el producto actualmente empacado, leído por el programa "Leer Config Bal.vi" como se mostró previamente en la Figura 4.10. En primera instancia este valor dividido entre cinco es usado en este lazo para obtener un umbral de referencia, como se explicó previamente en la Sección 3.1.5 de este informe. Si el cluster que contiene el peso no está vacío, el valor de peso obtenido es

mayor a este umbral, y es además un peso estático, el lazo if de color rojo mostrado en la parte inferior izquierda de la Figura 4.26 guardará este valor para la siguiente iteración usando un “feedback node”.

Mientras el peso se mantenga estático el software contará el tiempo en el cual se mantuvo el peso estático; si el tiempo fue menor a 200ms este dato no será considerado válido.

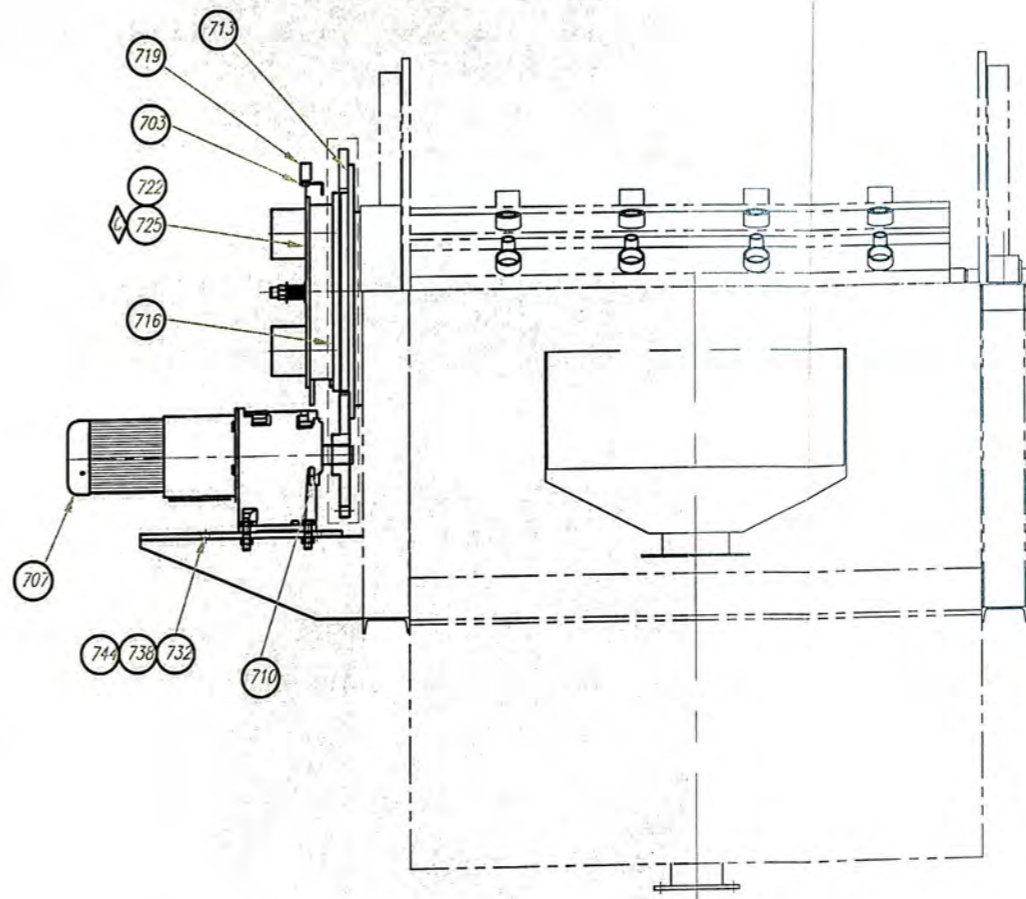
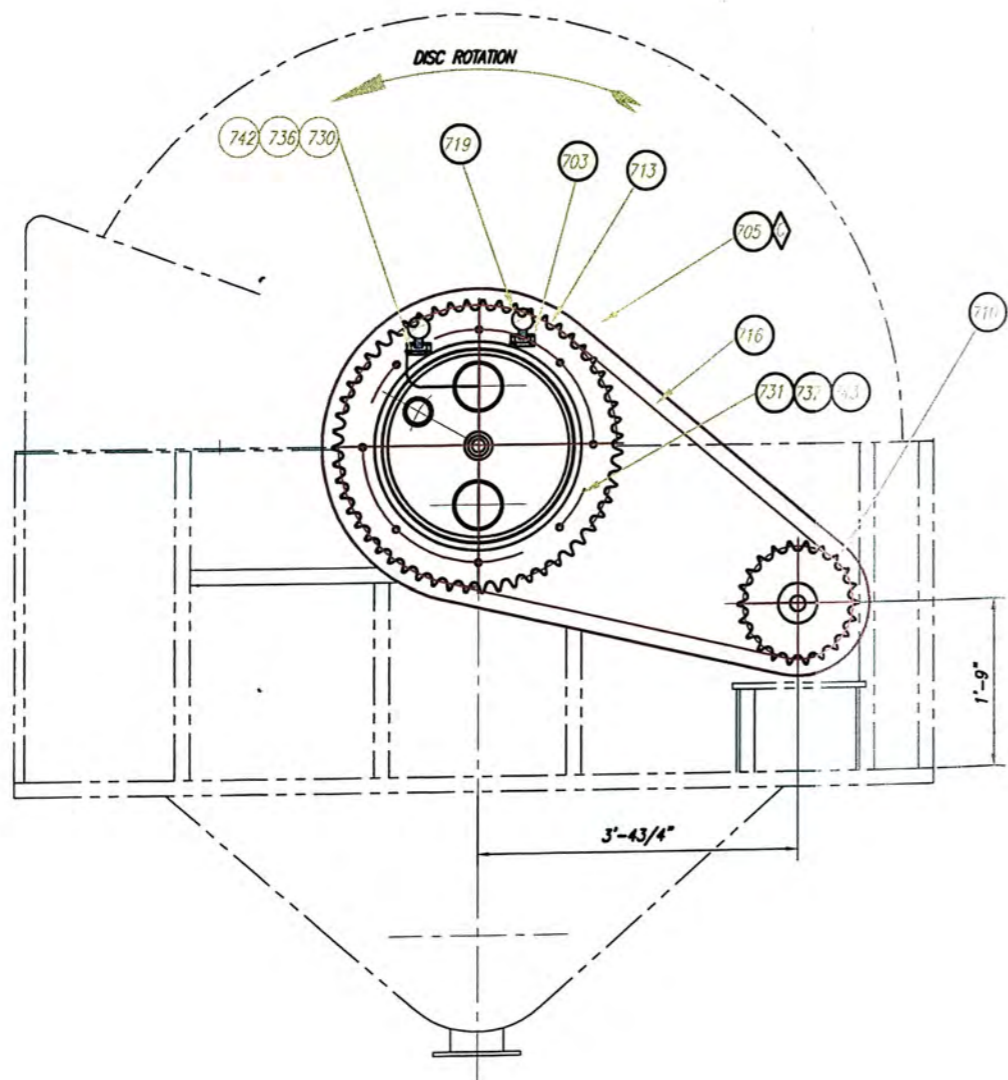
La variable booleana grabar correspondiente a la balanza en evaluación es activada cuando la lectura de peso pasa de ser superior al umbral a ser inferior a este, con pesos estáticos, como se muestra en el lazo if de color naranja. Vale la pena notar que debido a que se usan pesos estáticos no fue necesario usar histéresis para evitar la sensibilidad a ruido. Si la variable grabar está en true, el tiempo en el cual el peso permaneció estático fue mayor a 200ms, y el peso es menor de la referencia, se considera que el saco fue retirado del terminal del pesaje. El código contenido en los lazos if de color azul se encargará de grabar el último pesaje estático el cual fue pasado de iteración en iteración por el software. Si el operador realizó la operación de pesaje según el procedimiento establecido, este último peso estático del saco correspondería al peso final del mismo.

Existe sin embargo la posibilidad de que en todo este proceso el sistema no haya registrado ningún peso estático, por ejemplo, si el operador coloca el producto rápidamente y retira el saco de producto sin permitir a la balanza estabilizar su lectura. En este caso la información pasada al bloque de grabación sería la misma del peso previamente grabado, ya que en ningún momento un peso estático nuevo fue aceptado. El software compara entonces la información recibida con la del último peso grabado y, dado que es imposible que dos operaciones de pesaje se completen en un mismo

segundo, el campo "time string", que contiene la hora, minutos y segundos en que fue grabado el dato es comparado con el del nuevo peso a grabar, si ambos son iguales, se identifica este peso como una mala pesada, usando el flag correspondiente, se utiliza el valor de peso cero, la fecha y hora del momento en que se idéntica el mal pesaje.

Este lazo también permite pausar la grabación de datos utilizando tres casillas de verificación en la interfaz gráfica, tal como se muestran en la Figura 3.14, marcadas como el ítem número 2.

Esta parte del programa también actualiza los indicadores del panel frontal mostrando los pesos y unidades recibidos por el sistema. También se implementó una gráfica de histórico de peso lo cual quedó finalmente deshabilitado.



ITEM	QTY.	DESCRIPTION/MATERIAL	WEIGHT
703	2	GAGE MOUNTING BRACKET (SOPORTE ABRAZADERA DE MANOMETRO)	.
705	1	CHAIN GUARD (BUILD TO SUIT) ~ (GUARDA DE CADENA)	.
707	1	FILTER DRIVE ~ (CONDUCTOR DEL FILTRO)	.
710	1	DRIVE SPROCKET 14T, 2" P. W/TL ~ (PIÑON CONDUCTOR)	.
713	1	DRIVEN SPROCKET 84T, 2" P. Type A (PIÑON CONDUCTIDO)	.
716	1	CHAIN ROLLER ANSI 160 ,2" P. SINGLE (CADENA DE RODILLOS)	.
719	1	GAUGE VACUUM 30" Hg, 3 1/2 DIA. (VACUOMETRO)	.
722	2	CONNECTOR, MALE N° 68-F, 1/4" ~ (CONECTOR MACHO)	.
725	1	TUBING 1/4" O.D. x 4' - 0" Lg. (CAÑERIA)	.
727	1	ADJUSTABLE FREQUENCY AC DRIVE ~ (VARIADOR DE VELOCIDAD)	.

CLIENTE	...		
PROYECTO	SISTEMA DE FILTRADO DE RELAVES		
TÍTULO	FILTRO DE DISCOS 9'x4 FILTER DRIVE ASSEMBLY		
DESIGNADO	...	REVISADO	J.MEDINA
ELABORADO	...	APROBADO	W.SILVA
FECHA	03/11/04	ESTADO	S/E
UNIVERSIDAD	CMC-DFA-701		OTRO

UNI
FIM

4.2 Validación de algoritmos por simulación

Para acelerar el desarrollo de la solución en el área de software se desarrollaron las siguientes herramientas en software:

- Una herramienta que simula los mensajes enviados por el terminal de pesaje para un buen pesaje, mal pesaje y pesaje fuera de rango. Esta herramienta está implementada en el programe “emu_bal.vi”, incluida en el proyecto de Labview.
- Una base de datos conteniendo una tabla con la misma estructura de la tabla contenida en la base de datos de la empresa cliente.

Estas herramientas permitieron simular la operación del sistema durante el desarrollo del programa, minimizando la necesidad de visitas a planta para pruebas del sistema. La mayor parte de los errores de software fueron depurados utilizando estas herramientas.

CAPÍTULO V

PUESTA EN OPERACIÓN DEL SISTEMA EN UNA PLANTA DE ENVASADO

5.1 Descripción de la planta de envasado

El sistema objeto del presente informe se instaló y puso en operación en el área de envasado de una planta de elaboración de condimentos alimenticios ubicada en la ciudad de Lima, Perú. Esta área de envase recibe desde un piso superior el producto, el cual fluye de manera natural por gravedad a la máquina de embolsado. Esta máquina entrega el producto envasado en pequeños empaques llamados por el fabricante "sobres, los que a su vez están empacados en bolsas que contienen decenas de estos usualmente. Finalmente estas bolsas son entregadas por la máquina a una faja transportadora la cual transporta estas bolsas a otra zona en la cual un operador realiza el empaque manualmente, usando un terminal de pesaje para verificar si el peso total de saco que está empacando cumple con las especificaciones de peso requeridas. Los sacos de producto son luego entregados por una tolva que los envía hacia el piso inferior. La Figura 5.1 esquematiza el flujo previamente mencionado.

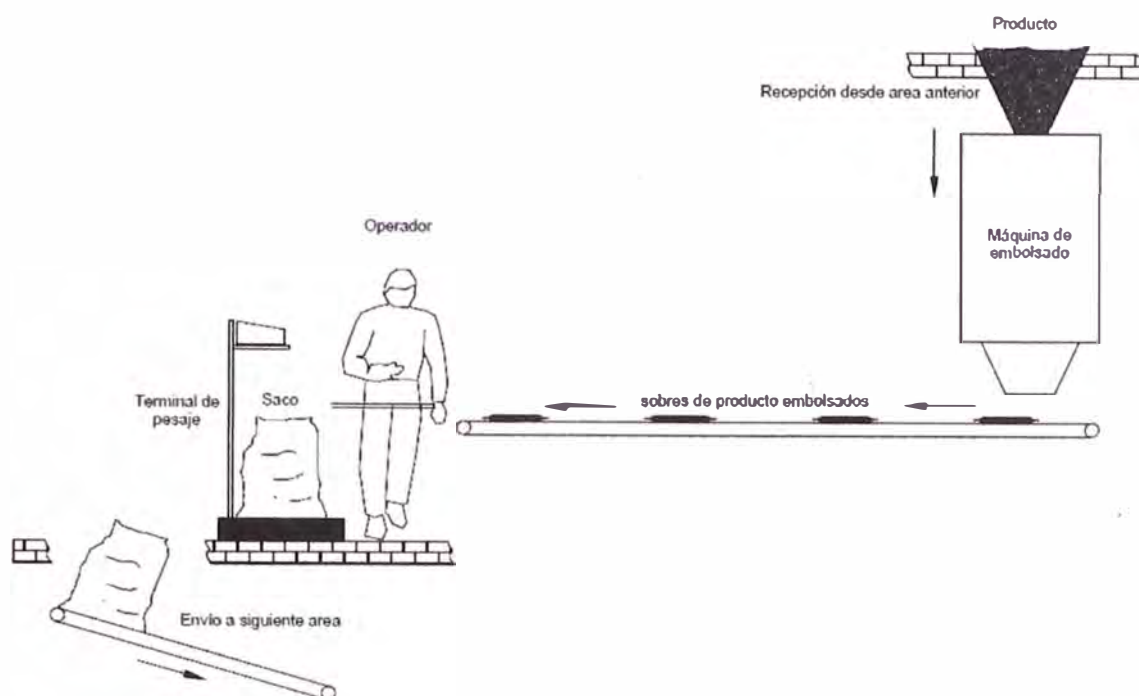


Figura 5.1 – Esquema del proceso de envase de sacos de producto

5.2 Instalación del sistema

La instalación del sistema puede ser separada en dos partes: instalación de hardware e instalación de software.

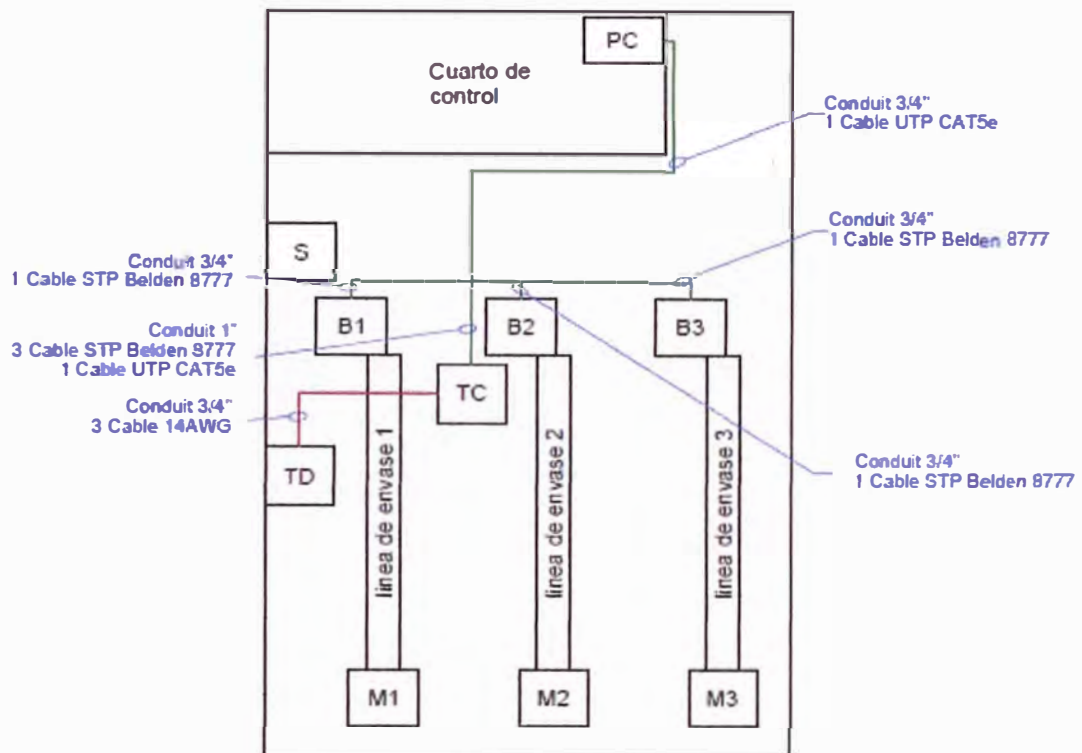
5.2.1 Instalación de hardware

Los siguientes componentes físicos conforman del sistema: el gabinete de comunicación, la infraestructura eléctrica y cableado, el computador, y los terminales de pesaje. Las características de su instalación se describen a continuación.

5.2.1.1 Gabinete de comunicación.

El gabinete de comunicaciones se instaló cerca de los terminales de pesaje en el área de producción. La instalación se realizó fue subcontratada

a un contratista especialista en instalaciones de este tipo. En este gabinete se instaló el conversor Serial-Ethernet, su fuente de alimentación, y el switch Ethernet.



Bn: Terminal de pesaje de la línea de envase número "n"
 Mn: Máquina de embolsado de la línea de envase número "n"
 TC: Tablero de comunicaciones.
 TD: Tablero de distribución eléctrica.
 S: Área de envío al piso inferior.
 PC: computador en el cual se instaló el program de software"

Figura 5.2 – Esquema de instalación de cableado e infraestructura eléctrica

La Figura 5.2 Muestra un esquema básico de la instalación del cableado e infraestructura del sistema.

5.2.1.2 Infraestructura eléctrica y cableado.

La instalación de la infraestructura eléctrica, y cableado fueron subcontratadas a una empresa especializada en este tipo de trabajos. Se anexa al presente trabajo, como referencia, el esquema de conexión de los puertos seriales del conversor Ethernet-serial modelo NPort 5410.

5.2.1.3 Terminales de pesaje.

Se trabajó con los terminales de pesaje Mettler-Toledo modelo ID3S existentes en la planta; ningún terminal de pesaje nuevo fue instalado como parte de este proyecto.

5.2.1.4 Computador

Se trabajó con un computador proveído por el cliente y ya instalado en el área de envase de la planta, por lo que no existió un proceso de instalación del computador como parte de la instalación del sistema descrito en el presente informe.

Uno de los objetivos para el cliente era realizar una implementación de bajo costo, este fue el principal motivo para usar un computador existente así como para usar un computador en vez de un equipo de hardware industrial. Sin embargo, se dejó claro desde un inicio al cliente que Labview permite transportar el código ya programado hacia otras arquitecturas tales como los PACs Compact Fieldpoint de National Instruments, los cuales si contienen hardware diseñado para ambientes industriales así como ejecución en tiempo real, manteniendo la flexibilidad y amplia gama de librerías existentes en Labview.

5.2.2 Instalación de Software

La plataforma de desarrollo de NI Labview contiene una herramienta de desarrollo de instaladores, la cual facilita y automatiza gran parte de esta labor. Esta herramienta fue usada para crear los instaladores del sistema descrito en el presente informe.

La herramienta de creación de instaladores incluye automáticamente todas las librerías, controles u otros objetos de software necesarios para la ejecución de todos los componentes de la aplicación.

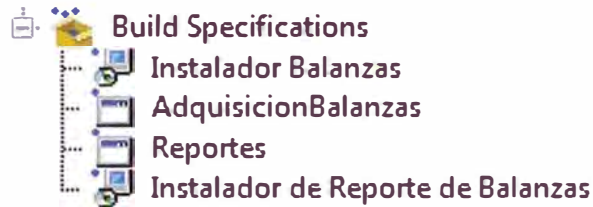


Figura 5.3 – Herramienta para creación de instaladores de Labview.

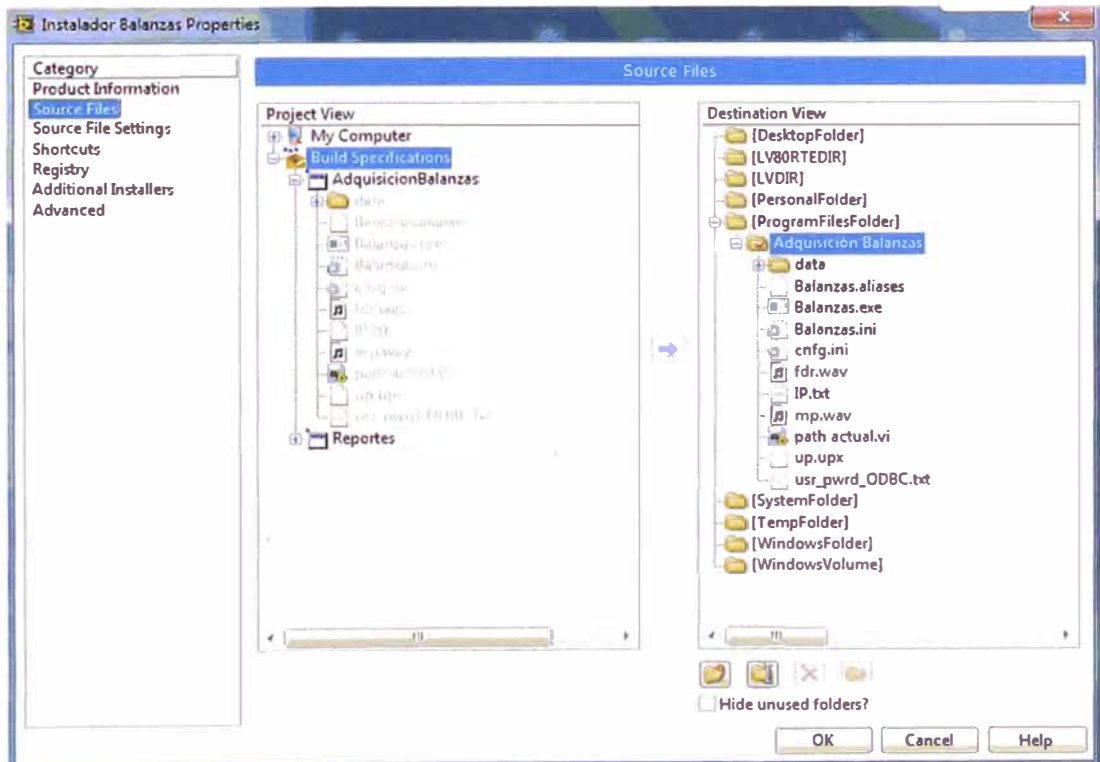


Figura 5.4 – Configuración de archivos fuente del instalador "Balanzas".

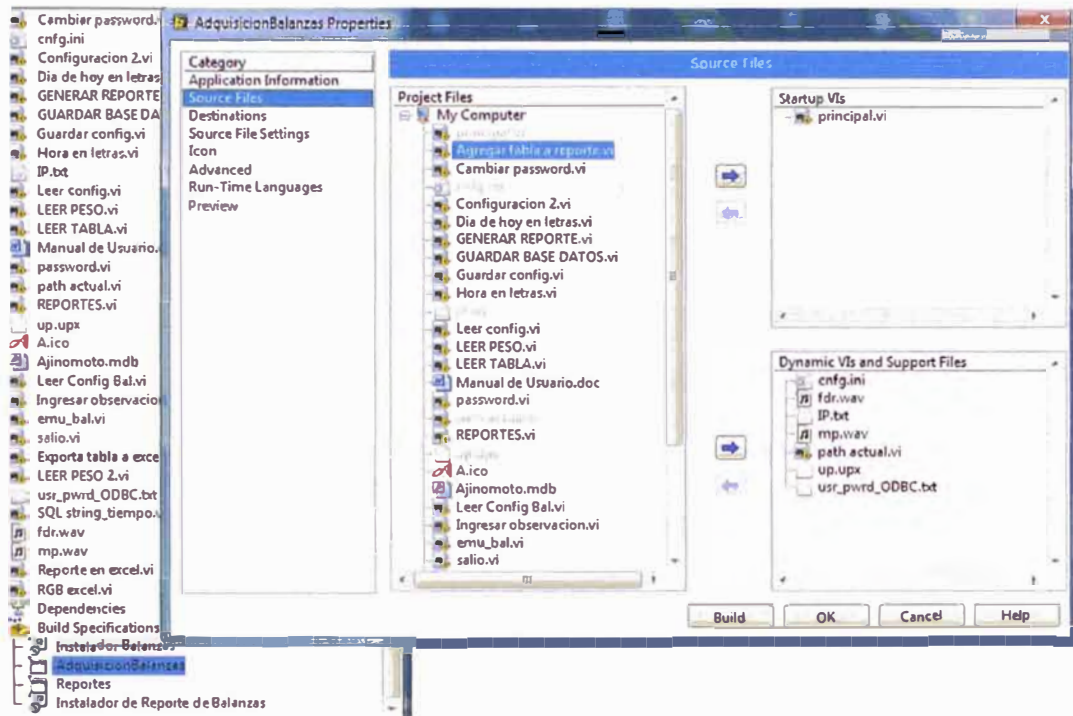


Figura 5.5 – Configuración de archivos fuente de programa ejecutable “AdquisicionBalanzas”.

5.3 Puesta en operación

5.3.1 Versión preliminar

Luego de completar una primera versión beta, se instaló esta versión preliminar como piloto en la planta de envase con el objeto de que el cliente pueda interactuar y verificar las distintas funcionalidades.

5.3.2 Levantamiento de observaciones del cliente

La mayor parte de las observaciones del cliente fueron levantadas entre la instalación del piloto y el lanzamiento final. Entre ellas se tuvieron las siguientes:

- Incluir un mensaje de advertencia cuando se pierda la conexión con la base de datos SQL durante la operación.
- Cambio de diversas características de la interfaz gráfica.
- Agregar sonido al mensaje de advertencia de mal pesaje y pesaje incorrecto.
- Invertir orden de datos en las tablas, manteniendo ahora los nuevos datos en la parte superior de la tabla.
- Mantenimiento a la base de datos: se acordó será realizado por el área de TI.

5.3.3 Lanzamiento final

Dado que se instaló una versión preliminar y la mayor parte de las observaciones fueron ya corregidas en esa etapa, el lanzamiento final consistió básicamente en los siguientes puntos:

- Entrenamiento a los operadores por parte del supervisor en relación al procedimiento correcto de operación de los terminales de pesaje.
- Entrenamiento al supervisor del área de envase sobre el sistema a nivel usuario.
- Entrenamiento al supervisor del área de mantenimiento eléctrico sobre el sistema a nivel eléctrico y de software.
- Informar a todos los interesados en planta.

CONCLUSIONES

Considerando que se realizó lo siguiente:

- El desarrollo e implementación de los algoritmos para la comunicación entre la base de datos corporativa y la aplicación cliente.
- El desarrollo e implementación de los algoritmos de comunicación entre la aplicación cliente y los terminales de pesaje modelo ID3S de la marca Mettler-Toledo.
- El desarrollo e implementación de la interfaz gráfica conforme a las especificaciones del cliente.
- La puesta en operación de la solución desarrollada en una planta de envase y el levantamiento de todas las observaciones del cliente.

Se concluye lo siguiente:

Se desarrolló, implementó y puso en operación un sistema de adquisición, registro y monitoreo de datos de terminales de pesaje de la marca Mettler-Toledo, modelo ID3S.

El enunciado previamente mencionado constituye la tesis, materia del presente informe de suficiencia.

RECOMENDACIONES

El desarrollo del presente trabajo motiva las siguientes recomendaciones:

- Dado que la solución del presente trabajo tuvo como plataforma de hardware un computador debido a las restricciones de costos establecidas por el cliente, una recomendación resaltante es la futura exportación de este programa a una plataforma industrial. El programa puede ser exportado con relativa facilidad a la plataforma Compact Fieldpoint Real-time de National Instruments, la cual también se programa en el entorno de Labview.
- La solución fue desarrollada para tres balanzas a pedido del cliente; sin embargo, esta puede ser fácilmente ampliada para el resto de balanzas ubicadas en la misma área de envase en la cual se instaló el sistema. Esto demandaría modificaciones menores en el software y la instalación de un conversor Ethernet-Serial con una mayor cantidad de puertos.

BIBLIOGRAFÍA

1. Sitio electrónico oficial para soporte en productos de National Instruments en *http://www.ni.com/support/*
2. Oscar Alvarado. Nava (2006) *Diseño y administración de redes de computadoras Encapsulamiento Ethernet e IEEE 802.2/802.3*
3. Compañía Mettler-Toledo (Albstadt) GmbH (1997). *Operating Instructions. Mettler-Toledo Multirange ID3S weighting terminal.*
4. Compañía Moxa Technologies Co., Ltd. (2005). *Nport 5400 Series user's manual.*
5. Luis Miguel Crespo Martínez, Francisco A. Candelas Herías (1998). *Introducción a TCP/IP. Sistemas de transporte de datos.*
6. Página oficial de ayuda en base de datos del sitio web *http://technet.microsoft.com/en-US/*

ANEXOS

ESQUEMA DE CONEXIÓN SERIAL PARA LOS PUERTOS DB9 DEL
CONVERSOR ETHERNET-SERIAL NPORT 5410 PROVEHIDO POR EL
FABRICANTE.

DB9 Female to DB9 Male for NPort 5410

