

# **UNIVERSIDAD NACIONAL DE INGENIERÍA**

**FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**



**OPTIMIZACIÓN DEL SISTEMA DE DESPACHO DE UNA PLANTA DE  
ALMACENAMIENTO DE GLP UTILIZANDO SOFTWARE SCADA  
PROFICY CIMPPLICITY Y PLC'S Y MONITOREO DE DATOS VÍA WEB**

**INFORME DE SUFICIENCIA**

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

**PRESENTADO POR:  
JHON ROBERT MILLA PAJUELO**

**PROMOCIÓN**

**2009-II**

**LIMA-PERÚ**

**2013**

**OPTIMIZACIÓN DEL SISTEMA DE DESPACHO DE UNA PLANTA DE  
ALMACENAMIENTO DE GLP UTILIZANDO SOFTWARE SCADA  
PROFICY CIMPLICITY Y PLC'S Y MONITOREO DE DATOS VÍA WEB**

A mis padres, hermanos y esposa  
por su apoyo incondicional y por ser mi  
razón para seguir adelante.

A todo el personal de PPAL-REPSOL  
por su apoyo en el desarrollo de esta  
aplicación

## SUMARIO

En el presente informe se describe el desarrollo de un sistema que facilita las operaciones y almacenamiento de la información del despacho de GLP en la planta de almacenamiento Repsol – Ventanilla, la implementación de este sistema permite optimizar las operaciones de despacho se realizan en 3 estaciones diferentes en el que se hacen varias impresiones manuales de los pesos de las unidades y posteriormente se ingresan estos datos a la computadora del operador, para llevar un registro del GLP cargado y despachado en la planta así como también para la emisión de facturas y guías para el transporte de GLP.

Para plantear la implementación de un sistema SCADA que se encargue de estas operaciones el primer paso será realizar la integración de la Balanza de pesaje de cisternas, la cual tiene un display con un puerto serial, para dicha integración se utilizará un PLC en el que se interprete la información de pesos recibida por la Balanza; luego se realizará el desarrollo de una aplicación SCADA que realice las operaciones de registro, pesaje y despacho.

Para facilitar el manejo de información se utiliza una base de datos en SQL, la cual almacena la información de las operaciones y los datos de las unidades a despachar, de modo que se puede consultar cuando se requiera realizar un nuevo registro, una consulta de los despachos realizados o la emisión de una guía.

Finalmente se utiliza el servidor web del software SCADA utilizado para la publicación de datos en Internet y se permite el monitoreo de datos en una estación remota.

## ÍNDICE

INTRODUCCIÓN.....	1
<b>CAPITULO I</b>	
<b>PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>2</b>
<b>1.1 Descripción del Problema .....</b>	<b>2</b>
<b>1.2 Objetivos .....</b>	<b>4</b>
<b>1.2.1 Objetivo General .....</b>	<b>4</b>
<b>1.2.2 Objetivos Especificos.....</b>	<b>5</b>
<b>1.3 Limitaciones.....</b>	<b>5</b>
<b>CAPITULO II</b>	
<b>MARCO TEÓRICO CONCEPTUAL .....</b>	<b>6</b>
<b>2.1 Planta de Almacenamiento – REPSOL .....</b>	<b>6</b>
<b>2.1.1 Gas Licuado de Petroleo (GLP) .....</b>	<b>6</b>
<b>2.2 Controlador Lógico Programable (PLC).....</b>	<b>7</b>
<b>2.2.1 Partes principales de un PLC .....</b>	<b>8</b>
<b>2.2.3 Estructura de un PLC .....</b>	<b>9</b>
<b>2.2.4 Controladores GE-IP.....</b>	<b>10</b>
<b>2.3 Comunicación Serial RS-232 .....</b>	<b>11</b>
<b>2.3.1 Código ASCII.....</b>	<b>13</b>
<b>2.4 Sistemas SCADA .....</b>	<b>13</b>
<b>2.4.1 Procedimiento para la implementación de un sistema SCADA.....</b>	<b>15</b>
<b>2.5 Bases de Datos .....</b>	<b>18</b>
<b>CAPÍTULO III</b>	
<b>METODOLOGÍA PARA LA SOLUCIÓN DEL PROBLEMA .....</b>	<b>20</b>
<b>3.1 Planteamiento de la Solución .....</b>	<b>20</b>
<b>3.2 Captura de datos en el PLC .....</b>	<b>20</b>
<b>3.3 Programación del SCADA .....</b>	<b>22</b>
<b>3.3.1 Comunicación con el Controlador.....</b>	<b>22</b>
<b>3.3.3 Desarrollo de Entorno Gráfico.....</b>	<b>28</b>
<b>CAPITULO IV</b>	
<b>ANALISIS Y PRESENTACIÓN DE RESULTADOS .....</b>	<b>64</b>

<b>4.1</b>	<b>Presentación de Resultados .....</b>	<b>64</b>
<b>4.2</b>	<b>Presupuesto de la Implementación .....</b>	<b>69</b>
	<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>70</b>
	<b>ANEXOS .....</b>	<b>72</b>
	<b>BIBLIOGRAFIA.....</b>	<b>79</b>

## INTRODUCCIÓN

La elaboración del presente informe surge de la necesidad de implementar un sistema que optimice las operaciones relacionadas al despacho de GLP en la planta de almacenamiento de Repsol – Ventanilla, de manera que se facilite las operaciones de registro, despacho y organización de la información de forma que permita contabilizar el flujo de ingreso y salida de GLP de la planta, elaborar guías de despacho y realizar reportes sobre las operaciones realizadas.

A su vez este sistema plantea una forma de integrar equipos con comunicación no estándar pero de trama conocida como lo son las balanzas de camiones utilizadas para este tipo de aplicaciones. La presente implementación integra la utilización de tecnología basada en PLCs y sistemas SCADA con el modo clásico de operación de despacho de GLP, el sistema a implementar además de permitir las operaciones y manejo de datos necesarios, debe ser fácil de utilizar de modo que no involucre que los operadores necesiten un tiempo prolongado de preparación para la utilización de este sistema.

Para poder cumplir con el propósito planteado, nuestro informe consta de cuatro capítulos, conclusiones y anexos.

En el primer capítulo, se describe y evalúa el problema, se establecen las limitaciones del trabajo y se detalla el objetivo del mismo, objetivo que guía su ejecución y que está presente desde la estructura hasta las conclusiones de este informe.

En el segundo capítulo marco teórico conceptual, se refiere a los conceptos teóricos con los que se basa el presente informe, desarrollando principalmente lo que es la programación de PLC's y sistemas basados en Visual Basic.

En el tercer capítulo, se presenta la propuesta de solución al problema planteado, así como la metodología a utilizar. El desarrollo de la propuesta, consiste en la programación de Controlador, desarrollo de la aplicación SCADA y su publicación a través de un servidor WEB incorporado.

En el cuarto capítulo, se realiza la presentación de los resultados, en la que se muestra las ventajas y recomendaciones asociadas al manejo del sistema implementado en el presente informe para asegurar su correcto funcionamiento.

Finalmente, se presentan las conclusiones finales y recomendaciones del informe.

## **CAPITULO I PLANTEAMIENTO DEL PROBLEMA**

### **1.1 Descripción del Problema**

En la planta de almacenamiento de GLP se realizan a diario aproximadamente 70 operaciones de despacho o carga de GLP a través de camiones cisterna, para contabilizar estos despachos se realiza el registro de datos de unidad, conductor, etc. en una estación llamada Caseta de despacho, posteriormente el operador de sala de control realiza la captura del peso a través de un display de Balanza, la cual realiza una impresión del peso registrado; posteriormente la unidad se dirige a las islas de carga en las que el operador de caseta controla el llenado del producto, verificando los niveles de presión y temperatura del producto en la cisterna. Luego la unidad se dirige nuevamente a la Balanza, para que el operador de sala capture su segundo peso y verifique que el nivel de llenado se encuentre dentro de los valores permitidos, luego de ello realiza la impresión del segundo peso. Con estos datos en la caseta de despacho se ingresa los datos de pesos, presión y temperatura en una plantilla excel y por diferencia de pesos se sabe la cantidad de producto que se esta despachando y con ello se elaboran las guías de remisión, guías de pesaje ó factura de ser el caso.

#### **Limitaciones en la operación actual de despacho de glp a granel:**

En la operación relatada anteriormente se realizan tareas repetidas de registro en cada una de las estaciones de operación por no estar estos datos disponibles en un sistema integrado, adicionalmente al ser en su mayoría operaciones de llenado de datos manual esta sujeto a errores de tipeo, y la impresión de numerosos tickets por cada operación y la cantidad de operaciones al día hacen mas tediosa la contabilización de operaciones y producto despachado. Durante el proceso descrito anteriormente se tiene que ingresar manualmente datos, a las plantillas de excel utilizadas, para la impresión de guías y tickets de peso; así como también para realizar el recuento de el total de producto despachado a cada cliente para su facturación y para el registro de despachos de la planta. Adicionalmente al despacho, los operadores de sala de control deben llenar un inventario llamado INVENCODOPAUDGLP, en el que se lleva un registro de todos los despachos realizados por clientes y se contrasta el total de despacho realizado con el



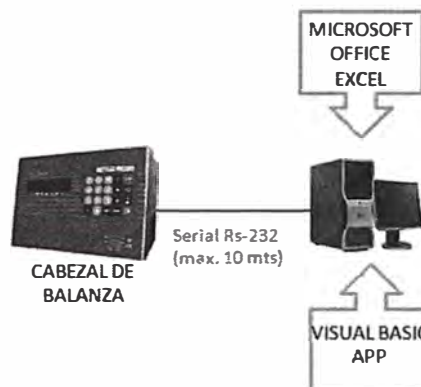
total de producto recibido y la cantidad de GLP almacenado en la planta (Esferas e Islas de Carga). La figura 1.1 muestra las características deficientes y sus respectivas consecuencias en la operación actual.



**Figura 1.1** Deficiencias del proceso de despacho con el sistema actual

### Soluciones alternativas:

En el mercado existen algunas soluciones alternativas, las cuales son aplicaciones de visual Basic que permiten la lectura y escritura de datos sobre un puerto serial, sin embargo tienen la deficiencia de no tener un desarrollo basado en base de datos, lo cual es importante para la contabilización de operaciones y al estar basados en comunicación RS-232 presentan limitaciones de distancia entre la PC y la fuente de datos serial. La figura 1.2 muestra el esquema de estas soluciones existentes en el mercado.

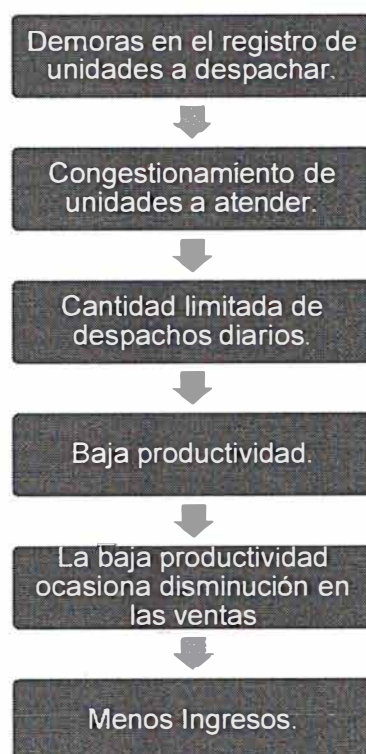


**Figura 1.2** Alternativas de solución existentes en el mercado

El tipo de operación que realizan se podría facilitar si la información referente a los pesos, estuviera organizada y disponible para su extracción cuando se necesite realizar alguna contabilización o consultar algún despacho realizado anteriormente.

La planta cuenta con una sola Balanza de cisternas y 6 islas de carga para despacho, por lo que es importante que el tiempo de pesaje y despacho sea lo mas óptimo posible.

La figura 1.3 muestra el flujo de consecuencias de las demoras en la operación de despacho debido al modo de operación actual.



**Figura 1.3** Consecuencias de la deficiencia en el despacho de GLP a granel.

## 1.2 Objetivos

Los objetivos del presente trabajo estan divididos en un objetivo general y tres objetivos específicos, los cuales señalamos a continuación.

### 1.2.1 Objetivo General

Implementar un sistema automatizado que permita facilitar las operaciones de despacho de GLP a granel para la planta de almacenamiento de GLP PPAL-REPSOL, a fin de reducir el tiempo empleado y permitir un mayor número de operaciones diarias. El sistema deberá permitir ser operado desde las 3 estaciones de operación que utilizan

en PPAL pero deberá ser del tipo integrado, es decir se concentrarán los datos en un servidor de Base de datos.

### **1.2.2 Objetivos Especificos**

Integrar las señales de pesaje a un sistema SCADA a través de un PLC.

Diseñar una base de datos en la cual se pueda almacenar la información relacionada a los despachos de GLP a granel.

Desarrollar una aplicación SCADA que permita realizar las operaciones de despacho de GLP a granel, integrando las señales de peso y enlazada a una base de datos SQL server, de forma que permita la impresión de tickets, guías y reportes de operaciones.

### **1.3 Limitaciones**

La Implementación de este sistema es resultado del analisis del modo de operación de la planta de almacenamiento PPAL Repsol – Ventanilla, este sistema es perfectamente adaptable a plantas similares, sin embargo para ello se necesitaría hacer algunos cambios para su adaptación, a nivel de Hardware a Utilizar, Programación, adaptación de reportes de despacho, etc.

En el presente informe se desarrolla la implementación del sistema de administración del despacho de GLP, pero no contempla el sistema de control de planta el cual se encarga monitorear las variables de presión, temperatura, flujo, etc. de la planta y controla la instrumentación en general de las planta de Almacenamiento.

## CAPITULO II MARCO TEÓRICO CONCEPTUAL

### 2.1 Planta de Almacenamiento – REPSOL

La planta de almacenamiento PPAL-REPSOL es una planta encargada de la recepción, almacenamiento y despacho a granel de GLP (Gas licuado de petróleo).

El producto es recepcionado principalmente de un Buque proveniente de la refinería de Talara y por camiones cisterna, provenientes de la Refinería de la Pampilla, la planta cuenta con 2 esferas T1 y T2, las cuales son dedicadas al almacenamiento y 6 islas de carga, dedicadas principalmente al despacho del producto a granel a camiones cisterna y despacho a través de tuberías subterráneas a la planta de envasado, para la venta de del producto en balones.



**Figura 2.1** Vista de la planta de almacenamiento PPAL-REPSOL

#### 2.1.1 Gas Licuado de Petróleo (GLP)

El GLP es una denominación aplicada a diversas mezclas de propano y butano que alcanzan el estado gaseoso a presión y temperatura atmosférica. El gas al ser comprimido y enfriado se condensa hasta convertirse en líquido, a presiones bajas, lo que se aprovecha para su almacenamiento y transporte. El gas es inodoro e incoloro por lo que normalmente se le agrega un odorizante llamado EtilMercaptano (EM), para que pueda ser identificado principalmente en caso de fugas.

Excepcionalmente es despachado producto inodoro para aplicaciones específicas como uso en aerosoles.

## 2.2 Controlador Lógico Programable (PLC)

Un PLC (Controlador Lógico Programable, por sus siglas en inglés) es una computadora industrial (dispositivo electrónico de estado sólido) que acepta entradas de switches y sensores, evaluándolas de acuerdo a un programa almacenado y genera salidas para controlar equipos o procesos.

Todos los procesos de producción experimentan una secuencia repetitiva fija de operaciones que involucra una serie de pasos y decisiones lógicas. Un PLC es usado para controlar los tiempos y regular las secuencias. Un PLC tiene las siguientes características básicas:

- Diseñados para soportar diversas condiciones ambientales; ruido eléctrico e interferencias electromagnéticas, arranque de motores, caídas de tensión, vibraciones, polvo y humedad.
- Realizan procesos complejos a pesar de sus dimensiones pequeñas.
- Soportan varios protocolos de comunicación que permite el monitoreo de los datos.
- Conectores y nivel de señales normalizados utilizados en la industria.
- Trabaja con un registro de fallas, permitiendo la identificación de la fecha, hora y tipo de falla.
- Desarrollo de la lógica en diversos lenguajes de programación, no todos los PLC poseen diversidad de lenguajes.
- Facilidad de programar y reprogramar en planta.

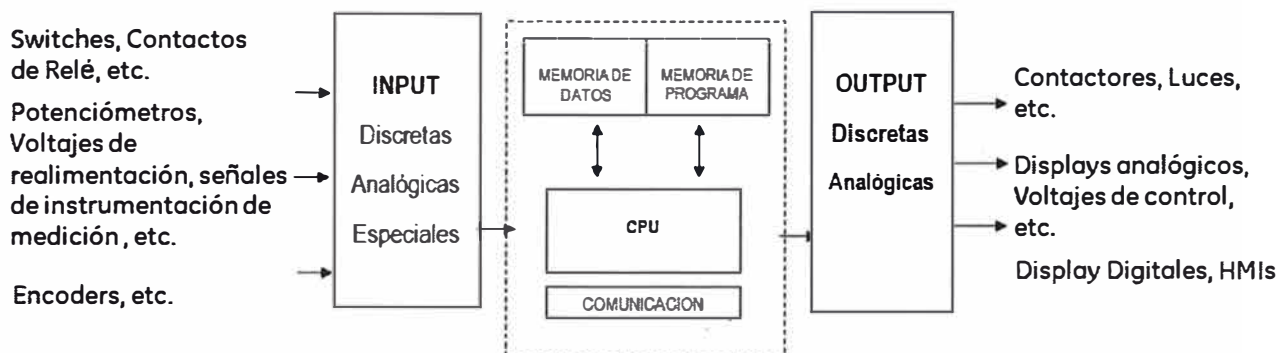
Los usos típicos de un PLC, son los siguientes:

- Concentrador de señales para integración a un sistema de control superior.
- Cambio de protocolo de comunicación, integración de equipos a un sistema.
- Automatización de una máquinas o planta
- Control remoto
- Monitoreo de señales y generación de alarmas.

Por lo tanto un PLC (Controlador Lógico Programable) es un aparato que reemplaza circuitos secuenciales de relés utilizados en el control de máquinas.

### 2.2.1 Partes principales de un PLC

El PLC de unidad de control de procesos o CPU sobre el cual la lógica es descrita, memoria de programa; además cuenta con una variedad de periféricos, módulos de entrada/salida, módulos de comunicación e interfaces hombre-máquina.



**Figura 2.2** Partes de un PLC

### 2.2.2 Señales en un PLC

Existen dos tipos de señales bien definidas en un PLC, las señales discretas y las analógicas. Las señales discretas pueden ser pulsadores, interruptores, etc, y la analógicas pueden ser parámetros físicos como presión, temperatura, etc.

Toda la información que el PLC recepciona ya sea en forma discreta o análoga, la representa en forma binaria, por ello es común encontrar 3 términos importantes que son el Bit, el Byte y la Palabra.

- **BIT:** El bit es la unidad elemental de información donde sólo puede tomar dos valores un "1" ó un "0", es decir, un bit es suficiente para representar una señal binaria.
- **BYTE:** El byte es una unidad compuesta por una agrupación ordenada de 8 bits, es decir, ocho dígitos binarios. Los bits se agrupan de derecha a izquierda tomando como número de bit del 0 al 7. En un byte se puede representar el estado de hasta ocho señales binarias, puede usarse para almacenar un número cuya magnitud como máximo sería:

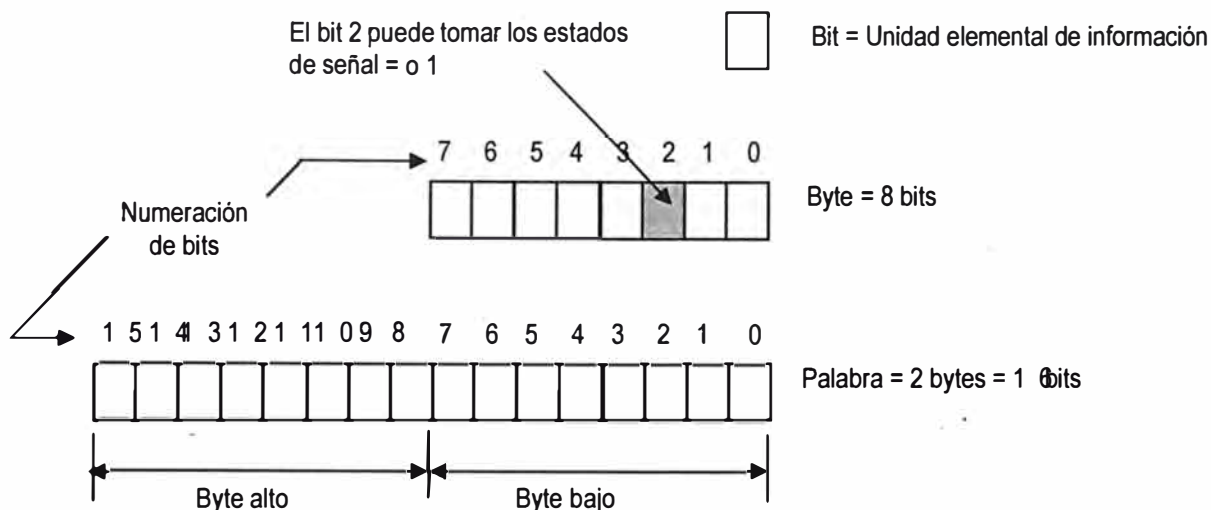
$$\text{Número máximo de un byte} = 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 = 2^8 - 1 = 255$$

- **PALABRA:** Para obtener mayor capacidad de procesamiento a veces se agrupan los bytes formando lo que se denomina las palabras. La palabra es una unidad mayor compuesta de 16 bits = 2 bytes. Los bits de una palabra se agrupan de derecha a izquierda tomando como número de bit del 0 al 15. En una palabra se pueden



representar hasta 16 señales binarias, puede usarse para almacenar un número cuya magnitud como máximo sería:

$$\text{Número máximo en una Palabra} = 2^{16} - 1 = 65535$$



**Figura 2.3** Representación gráfica de Bit, Byte y Palabra.

### 2.2.3 Estructura de un PLC

Los controladores PLC se componen principalmente por Módulo de CPU, fuentes de alimentación, módulos I/O y módulos de comunicación.

- **Módulo de CPU:** Contiene el CPU (Unidad Central de procesamiento), la cual se encarga de ejecutar las instrucciones para la operación del controlador y las memorias de Programa que pueden ser del tipo Volatil (RAM) ó no Volatil (FLASH – EPROM) generalmente es la que contiene almacenada la lógica de programación.
- **Fuentes de Alimentación:** genera la tensión para los módulos eléctricos de los dispositivos de automatización. El nivel de esta tensión puede ser de 24 VDC, 125 VDC, 120 VAC o 220 VAC.
- **Módulos I/O:** Módulos de entradas/salidas llamados así por sus iniciales en inglés (Input/Output), son los equipos que permiten al plc leer señales discretas o analógicas que pueden ser sensores de tipo RTD, 4-20 mA, switches, etc. y comandar actuadores como electroválvulas, válvulas proporcionales, lámparas indicadoras, etc.
- **Módulos de comunicación:** módulos que intercambian datos por medio de protocolos de comunicación, un protocolo de comunicación es un conjunto de reglas que indican como se debe llevar a cabo un intercambio de datos o información, se asocia a un

medio físico llamado puerto de comunicación. Por lo general los módulos de CPU tienen puertos de comunicación embebidos.

#### 2.2.4 Controladores GE-IP

La siguiente clasificación de equipos se realiza por capacidad de señales y tamaño máximo de lógica, en la parte superior los de gama alta o de mayor capacidad y del mismo modo de menor capacidad conforme va descendiendo de la lista.



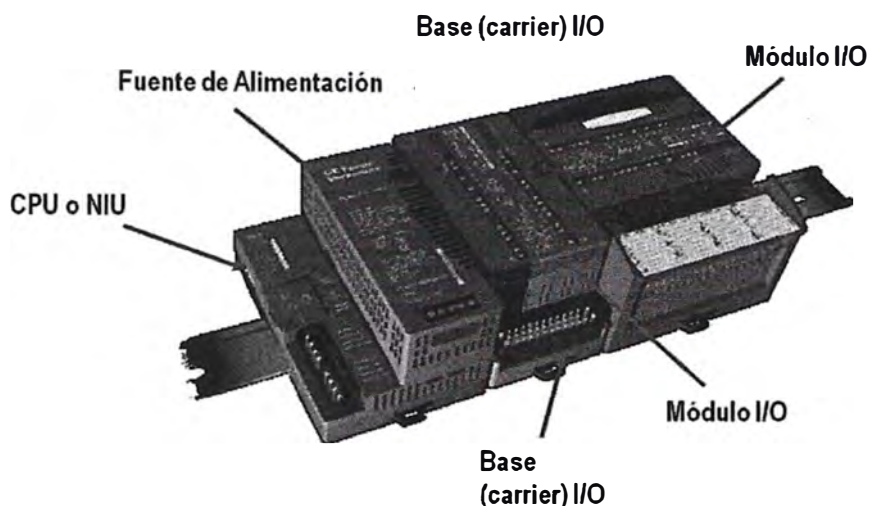
**Figura 2.4** Familia de controladores GE-IP

La elección del controlador más adecuado depende del tipo de aplicación a implementar, en este caso describiremos el controlador Versamax modular, por ser el de menor Gama y con las prestaciones suficientes para utilizarse como concentrador de señales, sus características principales son las siguientes:

- PLC totalmente modular, es decir que la fuente, CPU, módulos de I/O se piden por separado.
- Permite realizar un control intermedio, con la capacidad de realizar funciones más elaboradas.
- Mayor capacidad de memoria lógica, es ideal para concentrar señales y control de sistemas.
- Posee dos lenguajes de programación.
- Los VersaMax PLC por ser modulares son de tamaño variable que permiten adaptarse al espacio requerido.



- Posibilidad de comunicación :Modbus RTU Maestro/esclavo, SNP esclavo, Serial I/O (en todos los modelos de CPU), EGD, SRTTP – Maestro/esclavo (con CPU IC200CPUE05), Profibus esclavo, DeviceNet maestro, AS-i maestro, (con módulos adicionales), Profinet esclavo, Devicenet esclavo, EGD, Modbus TCP esclavo, Genius esclavo, Profibus esclavo (como I/O remoto).



**Figura 2.5** Componentes de un Controlador Versamax

### 2.3 Comunicación Serial RS-232

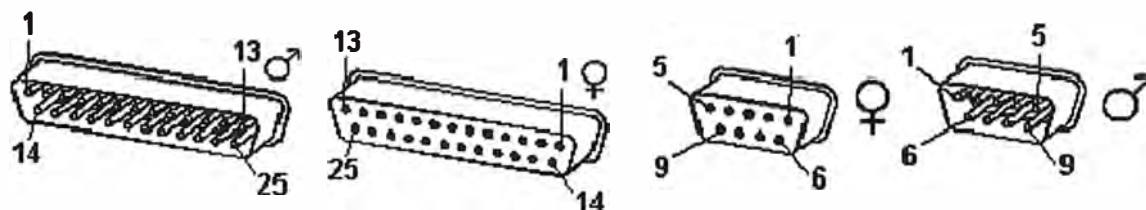
El estándar de comunicación RS-232 se define según la ANSI como intercambio de datos en forma de serie de datos binarios, generalmente entre 2 equipos uno de ellos del tipo DTE (Equipo terminal de datos) y otro DCE (Equipo de comunicación de datos), sin embargo también es posible vía la comunicación entre 2 equipos de tipo DTE ó 2 equipos de tipo DCE, invirtiendo los terminales de transmisión y recepción en el cable de conexión entre ellos.



**Figura 2.6** Comunicación serial RS-232

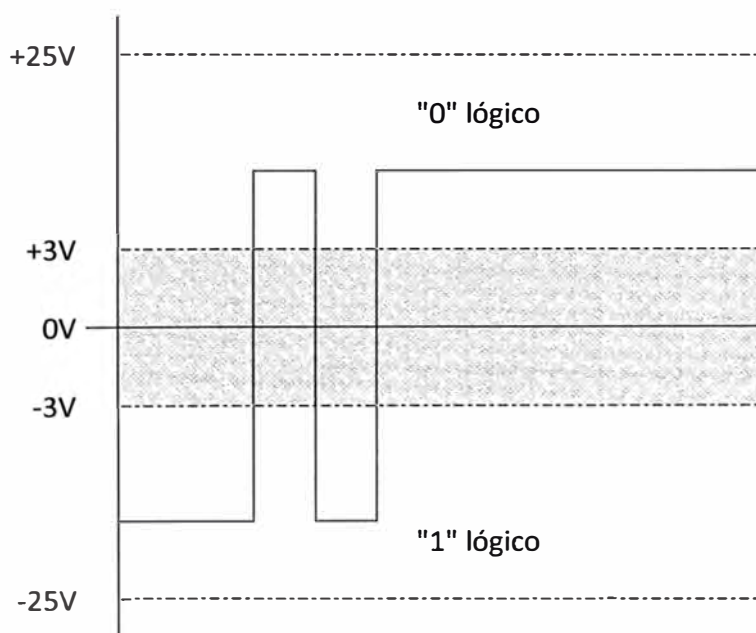
Un ejemplo de equipos del Tipo DTE, son los puertos que antiguamente venían en todas las computadoras o los de los convertidores serial-usb, mientras que ejemplos de equipos tipo DCE serían medidores eléctricos, PLCs, Cabezales de Balanza, etc.

Las características del RS-232 son el tipo de conector a utilizar, en la figura siguiente se muestran los conectores DB25 macho, DB25 hembra, DB9 hembra y DB9 macho de izquierda a derecha.



**Figura 2.7** Conectores RS232

Los niveles de tensión el "0" lógico corresponde a valores entre +3 y +25 Volts, mientras que un "1" lógico estará entre -3 y -25 Volts, la región entre -3 a +3 volts, se denomina banda muerta. Las velocidades de transmisión de datos pueden ser de 4800 hasta 20Kbits, sin embargo existen aplicaciones que pueden llegar hasta 115.2 Kbps. La longitud máxima para permitir la comunicación entre dos equipos vía RS-232 es de 15 a 20 mts, debido a la capacidad de carga, para distancias mayores se debe utilizar otro estandar como el RS-485, RS-422, etc.



**Figura 2.8** Regiones "0" lógico, indefinida y "1" lógico

Configuración de comunicación serial RS-232, para permitir una conexión válida entre 2 dispositivos, se debe configurar en ambos los mismos valores de Velocidad de transmisión de datos, paridad, número de bits de datos y de bits de parada. Adicionalmente se debe elaborar adecuadamente el cable de comunicación entre los dos dispositivos, en el caso de esta aplicación se tiene el puerto de PLC de tipo DCE, conector DB9 hembra y el puerto del Cabezal de Balanza de tipo DCE DB25 hembra, entonces el cable de conexión debería ser DB25 Macho por un extremo y DB9 macho por el otro extremo.

**Tabla 2.1** Pines de conexión de conectores DB9 y DB25

DB-25	DB-9	Nombre	Descripción
2	3	TD	Transmitted Data, TxD
3	2	RD	Received Data, RxD
4	7	RTS	Request To Send
5	8	CTS	Clear To Send
6	6	DSR	Data Set Ready
7	5	SG	Signal Ground, GND
8	1	DCD	Data Carrier Detect
20	4	DTR	Data Terminal Ready

### 2.3.1 Código ASCII

El código estándar americano para intercambio de información, llamado ASCII por sus iniciales en inglés. Es un conjunto alfanumérico universal aceptado y utilizado por la mayoría de equipos electrónicos, como los teclados de computadora por ejemplo, al pulsar una tecla del teclado se envía los datos en ASCII al CPU, en hexadecimal es un código de 8 bits que va desde 00xH a 7FxH.

## 2.4 Sistemas SCADA

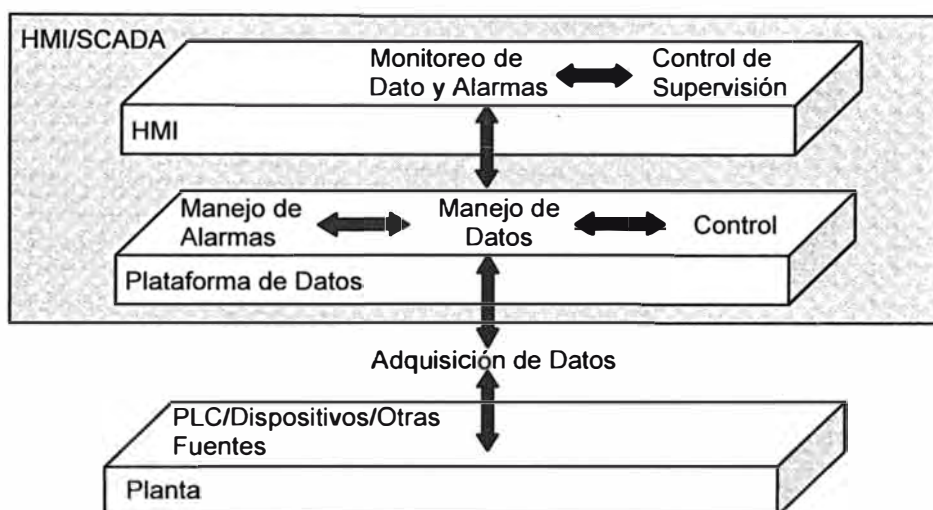
Los sistemas SCADA (Supervisory Control And data Acquisition), son aplicaciones de software de control que permiten administrar aplicaciones de diversos tipos desde un ordenador, un sistema SCADA es un sistema diseñado para controlar dispositivos de campo automáticamente desde la pantalla del ordenador. Hoy en día los sistemas SCADA combinan una gran cantidad de tecnologías para ofrecer en forma segura la adquisición de datos y el control del proceso, todo ellos sobre una gran cantidad de medios de comunicación disponibles.

Un sistema SCADA es capaz de mantener una base de datos amplia, conteniendo los últimos valores conocidos de los dispositivos de campo presentándolos al operador en

forma amigable y adecuada al proceso. Todo esto incrementa la productividad y la calidad de nuestro proceso así como la reducción de costos operativos.

Prestaciones de un software SCADA:

- Monitoreo de datos (Data Monitoring)
- Distribución de la información de datos (Data Distribution and Sharing)
- Manejo de Alarmas (Alarming)
- Manejo de Reportes (Reporting)
- Tendencias actuales (Real Time Trending)
- Tendencias Históricas (Historical Trending)
- Protección de la información y seguridad (Data Protection and Security)



**Figura 2.9** Diagrama de Bloques de funcionalidades SCADA

Un sistema SCADA debe cumplir varios objetivos para que su instalación sea mas eficiente:

- Deben ser sistemas de arquitectura abierta y escalable para poder adaptarse a las necesidades de la planta, escalables a nivel de adquisición de datos, de modo que se pueda seleccionar el tamaño de licencia mas adecuada para la aplicación.
- De comunicación abierta, deben tener facilidad de comunicación transparente con los dispositivos externos, tener la mayor cantidad de drivers de comunicación permite a los software SCADA ser mas adaptables a la industria.
- Deben tener un desarrollo que permita a los usuarios operar facilmente y reconocer lo que se esta manejando de manera intuitiva y graficamente amigable.
- Debe tener facilidad para la administración de datos históricos y conexión con base de datos relacionales.

### 2.4.1 Procedimiento para la implementación de un sistema SCADA

- Instalación del Software en servidor y clientes si los hubiera.
- Configuración Básica del sistema, definición de arquitectura, entre otros.
- Configuración de drivers o canales de comunicación.
- Creación de Tags o Variables de Proceso.
- Creación de Gráficos (Pantallas).
- Incorporación de funcionalidades específicas: Alarmas, seguridad, tendencias, reportes, enlaces a base de datos, etc.



**Figura 2.10** Diagrama de Bloques del desarrollo de un sistema SCADA

### 2.4.2 SCADA Proficy Cimplicity

El software SCADA es una solución versátil, potente y escalable, que cuenta con diversas aplicaciones destinadas a facilitar la implementación de sistemas SCADA sobre su plataforma.

Características:

- Visualización de variables del proceso en forma gráfica.
- Control para modificar el comportamiento de los procesos.
- Generación de tendencias e históricos de variables.



- Comunicación con dispositivos de campo a través de protocolos de comunicación con drivers embebidos y OPC.
- Niveles de acceso (Operador, Supervisor, Gerente, etc.).
- Almacenamiento de información para gestión con eficiencia.
- Posibilidad de programación en lenguaje estándar (como VBA o C) para cálculos y funciones específicas.

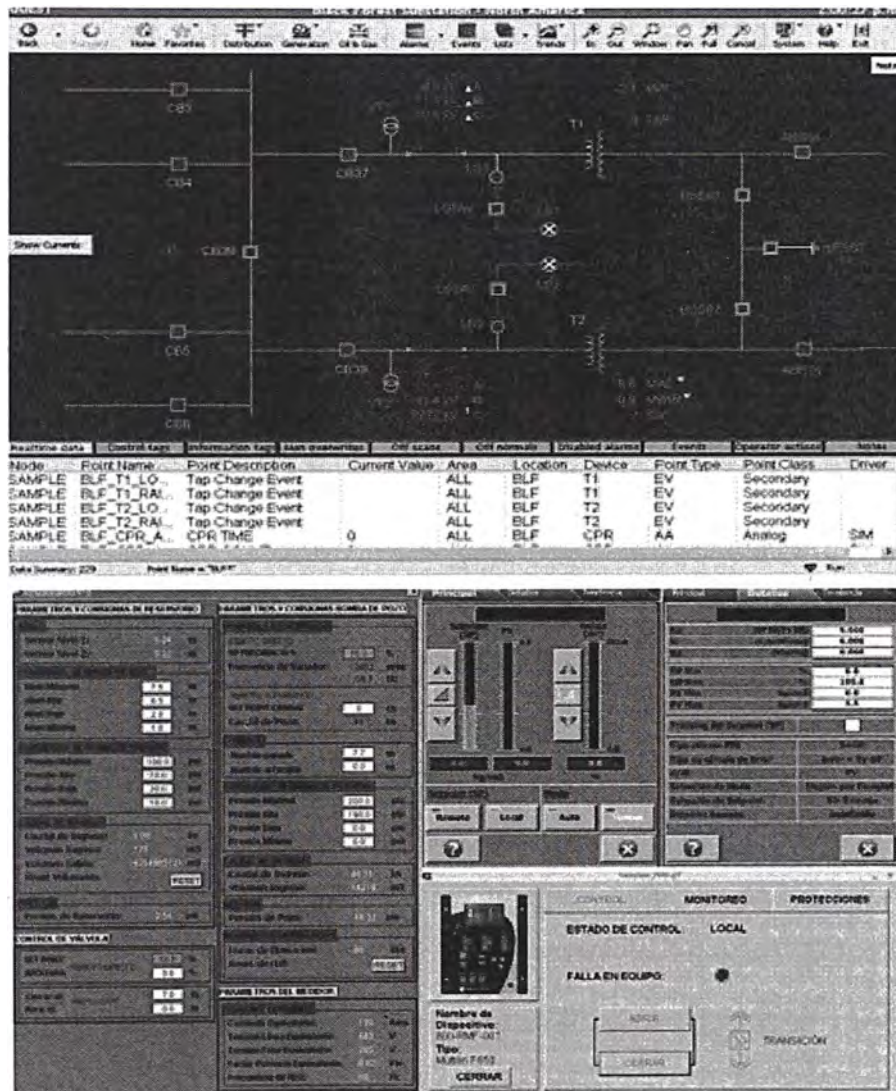


Figura 2.11 Imágenes Software SCADA Proficy Cimplicity

**Descripción del entorno de trabajo:** El entorno de trabajo para aplicaciones con proficy Cimplicity consta de 3 áreas bien diferenciadas, la primera "Project" contiene las utilidades de programación de la aplicación específica que se está desarrollando, la segunda contiene las utilidades para la configuración del servidor y por último "Runtime" en la que se ubican los utilitarios de apoyo en el modo Runtime, como configuración de alarmas, monitoreo de tags, usuarios, etc.

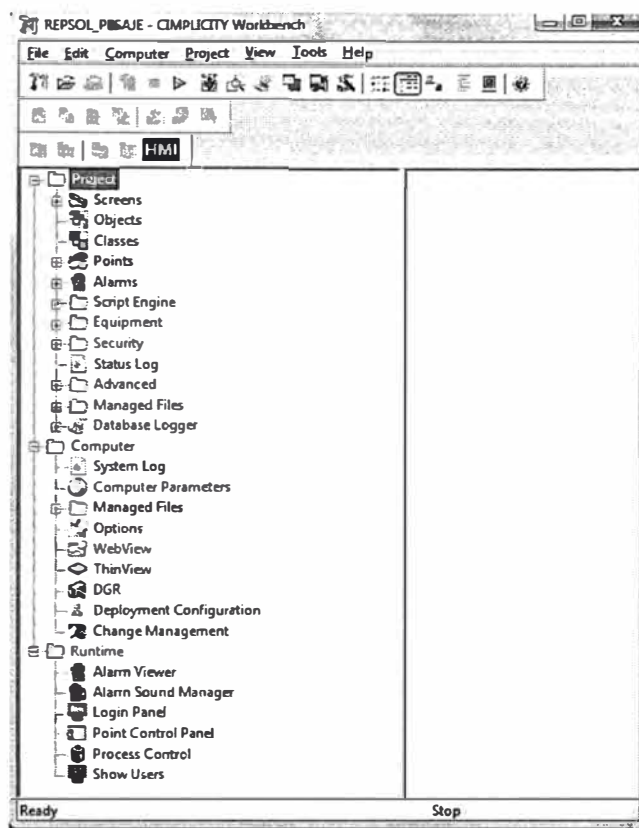


Figura 2.12 Entorno de trabajo Proficy Cimplicity

Las principales herramientas para la edición de esta aplicación son:

- Screens: Sección dedicada a la edición de pantallas y faceplates.
- Classes: Aquí se puede definir elementos que se pueden reutilizar.
- Objects: Aquí se pueden definir objetos basados en las clases ya definidas.
- Points: Variables definidas en la aplicación.
- Script Engine: Definición de Eventos globales y scripts basados en Visual C.
- Equipment: Definición de Hardware para comunicación, puertos y equipos.

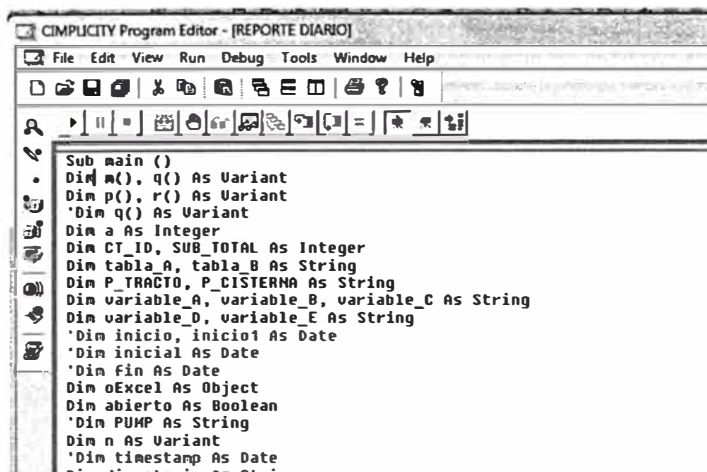


Figura 2.13 Script desarrollado en Proficy Cimplicity

## 2.5 Bases de Datos

Una base de datos es una colección de información organizada de manera que se pueda extraer de ella los datos que se requiera, es decir un es un sistemas de archivo electrónico. Las bases de datos tradicionales se agrupan en campos, registros y archivos, un campo es una pieza unica de información, por ejemplo el apellido, un registro es un sistema completo de campos, por ejemplo todos los datos de un empleado de una empresa y un archivo, sería un tabla con el registro de datos de todos los empleados de una empresa. Existen Software dedicados a la organización de bases de datos como Microsoft Acces, Microsoft SQL Server, Oracle, etc.



**Figura 2.14** Esquema Organización de base de Datos

Una instancia de Base de datos es un enlace a un conjunto de bases de datos, asu vez una base de datos es un conjunto de Tablas o registros en los que se tiene organizada la información.

### 2.5.1 Lenguaje SQL

El lenguaje de consulta estructurado (SQL por sus siglas en ingles), es un lenguaje de datos normalizado utilizado por diferentes motores de base de datos para realizar diferentes operaciones sobre las mismas. Existen 2 tipos de comandos en el lenguaje, las DLL que permiten crear y definir nuevas bases de datos, campos e indices y las DML que permiten generar consulatas para extraer datos, agregar nuevos datos y actualizar los existentes. Adicionalmente a las consultas básicas se les debe agregar clausulas, las cuales indican las codiciones a las que se les aplicará la consulta.



The screenshot displays the Microsoft SQL Server Management Studio interface. The main window shows a SQL query in the query editor:

```
SELECT * from documento
select ct_id, PESO_INICIAL, PESO_FINAL from DOCUMENTO o
```

The Object Explorer on the left shows the database structure for 'CRILLON (SQL Server 10.0.1600 - sa)'. The 'Tables' folder is expanded, listing various tables such as 'dbo.ALARM\_LOG', 'dbo.BATCH\_DATA', 'dbo.CHOFERES', 'dbo.CLAVES', 'dbo.CLIENTE\_TRANSPORTE', 'dbo.CLIENTES', 'dbo.CLIENTES\_SCOP', 'dbo.CORRIENTE MES', 'dbo.DATA\_LOG', 'dbo.DESTINOS', 'dbo.DOCUMENTO', 'dbo.EM\_LOG', 'dbo.EVENT\_LOG', 'dbo.GRAVEDADES', and 'dbo.LUGARES\_CARGA'.

The Results pane shows the output of the query, displaying two tables of data:

	DOCUMENTO_ID	DOCUMENTO_SERIE	DOCUMENTO_CORR	DOCUMENTO_1
1	1	001	000001	
2	2	001	000002	
3	3	001	000003	
4	4	001	000004	
5	5	001	000005	

	ct_id	PESO_INICIAL	PESO_FINAL
1	306	24720.0000	48270.0000
2	291	26080.0000	48040.0000
3	281	47990.0000	47990.0000
4	302	24880.0000	47790.0000
5	308	24200.0000	47750.0000
6	312	23400.0000	47650.0000
7	282	47580.0000	47580.0000

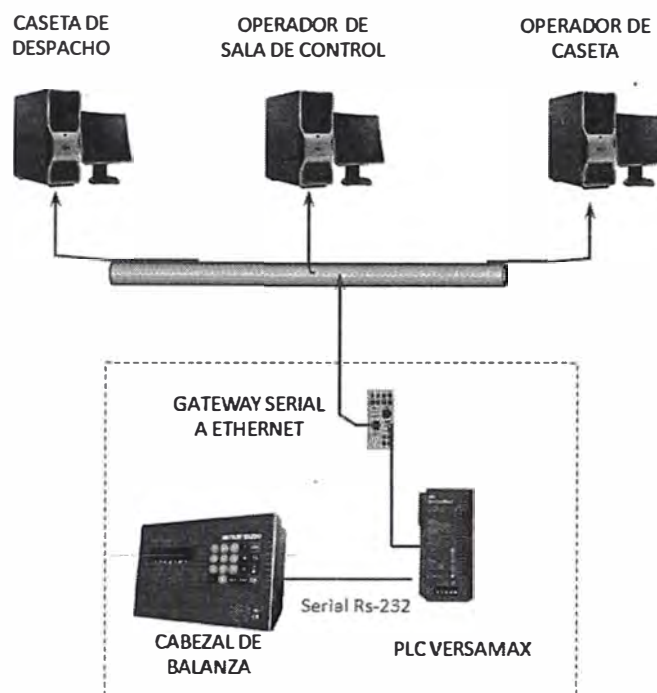
The status bar at the bottom indicates 'Query executed successfully.' and 'CRILLON (10.0 RTM) sa'.

Figura 2.15 Consultas en SQL

## CAPÍTULO III METODOLOGÍA PARA LA SOLUCIÓN DEL PROBLEMA

### 3.1 Planteamiento de la Solución

Se plantea una arquitectura distribuida de acuerdo a la metodología de operación con una base de datos compartida, de acuerdo a la arquitectura indicada en la figura 3.1.



**Figura 3.1** Arquitectura del sistema a Implementar

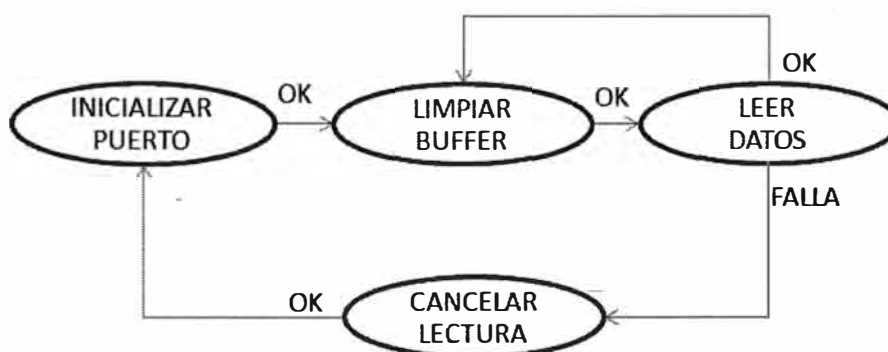
Para Implementar esta solución empezaremos por dividir nuestro problema por etapas, la primera de ellas será desarrollar el programa que le permita al PLC leer los valores del peso, luego la programación del SCADA que permita grabar los valores y mostrarlos cuando se requiera apoyandose en una base de datos SQL.

### 3.2 Captura de datos en el PLC

El cabezal de Balanza tiene un modo de envío continuo de pesos vía ASCII Serial, en la cual el envío se da de acuerdo a la Figura 3.2. Para su programación en el controlador se realizará, según lo indicado en el Manual de sistema del Fabricante, por etapas de modo que tengamos una lectura continua de los datos, como se muestra en la figura 3.3.

Character	1	Status			Indicated Weight						Tare Weight					17	18	
		2	3	4	5	6	7	8	9	10	11	12	13	14	15			16
Data	S T X	S W A	S W B	S W C	M S D	-	-	-	-	L S D	M S D	-	-	-	-	L S D	C R	C H K
Notes	A	B			C						D					E	F	

**Figura 3.2** Trama de envío de datos en ASCII



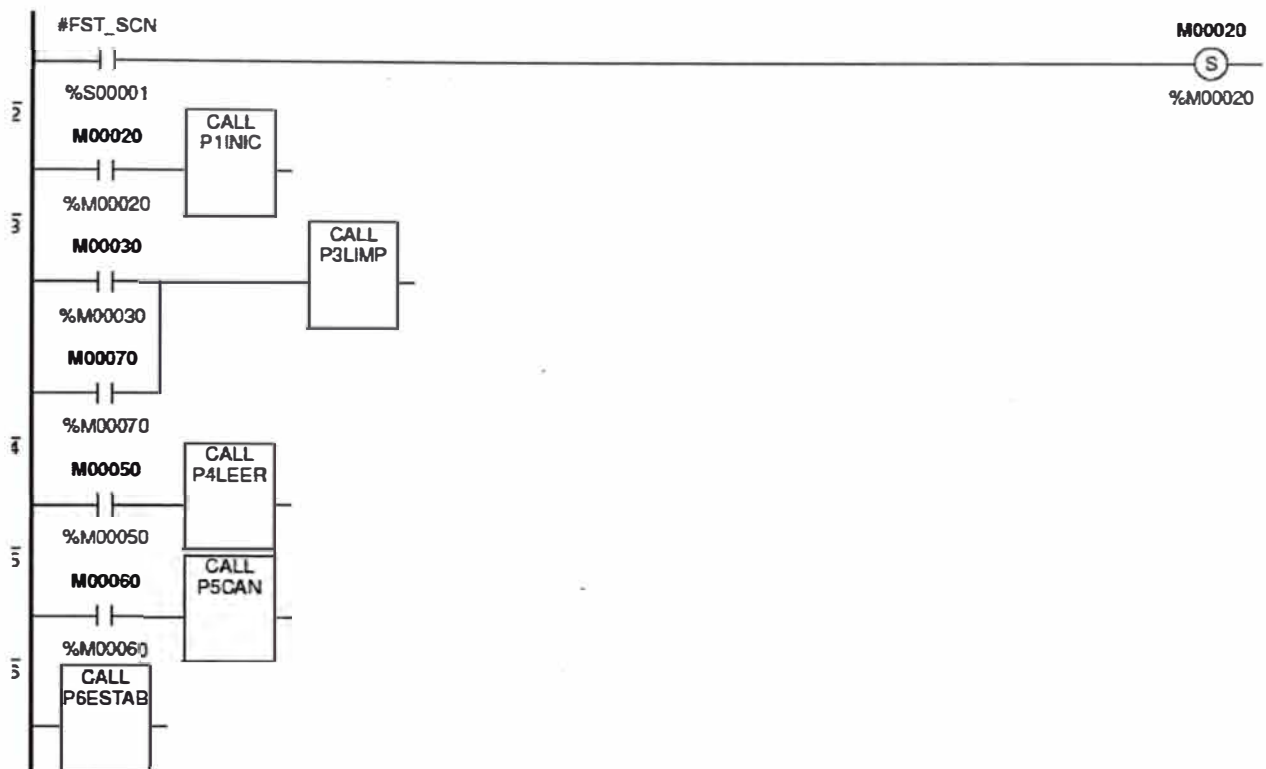
**Figura 3.3** Secuencia de operaciones en el PLC

De este modo tendremos una lectura continua y en caso de falla se reiniciará la secuencia por sí sola. Adicionalmente en el controlador una rutina adicional interpretará los datos leídos y guardará el dato de lectura actual en un registro que pueda ser leído por el SCADA.

Esta lógica ha sido desarrollada en base a subrutinas, las cuales realizan las siguientes funciones:

- P1INIC: Inicializa el puerto con los parámetros de comunicación necesarios.
- P3LIMP: Limpia el Buffer de Bytes no leídos.
- P4LEER: Envía comando para leer cadena de caracteres.
- P5CAN: Cancela las operaciones cuando hay un comando no válido.
- P6ESTAB: Separa los Bytes y convierte los caracteres ASCII a un número real.

## Programa en PLC



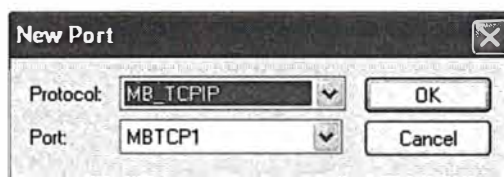
**Figura 3.4** Programa principal en PLC

### 3.3 Programación del SCADA

El desarrollo de la aplicación SCADA se divide en 4 etapas configuración de la comunicación, desarrollo del entorno gráfico, de la interfaz con base de datos y publicación de aplicación en web.

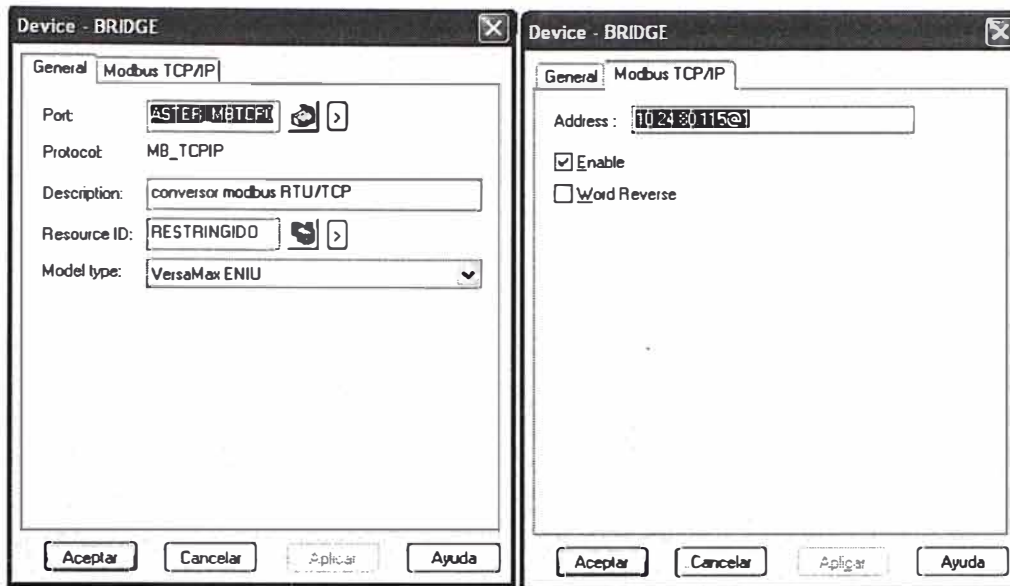
#### 3.3.1 Comunicación con el Controlador

Para permitir la comunicación entre el controlador y el sistema SCADA, se debe de configurar un puerto de comunicación, el cual esta relacionado a un protocolo, la figura 3.5 muestra el modo de configurar el puerto de comunicación.



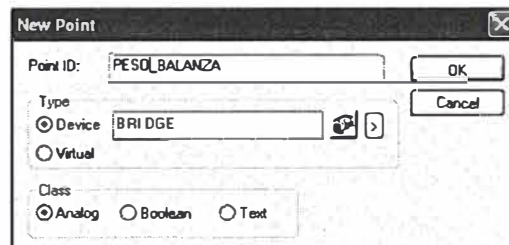
**Figura 3.5** Configuración de puerto de comunicación

Posteriormente se debe definir el equipo que se comunicará a través de este puerto

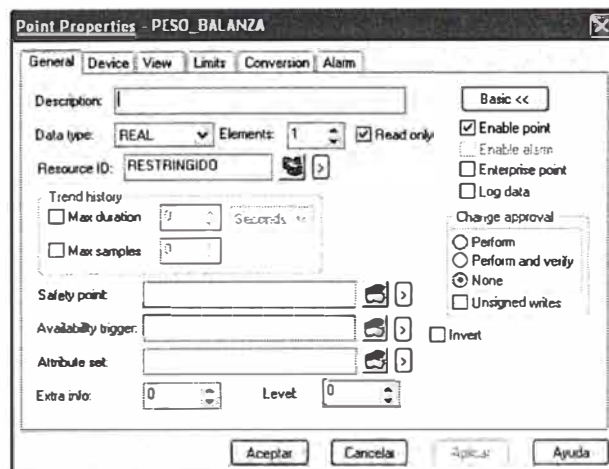


**Figura 3.6** Detalle de configuración de dispositivo

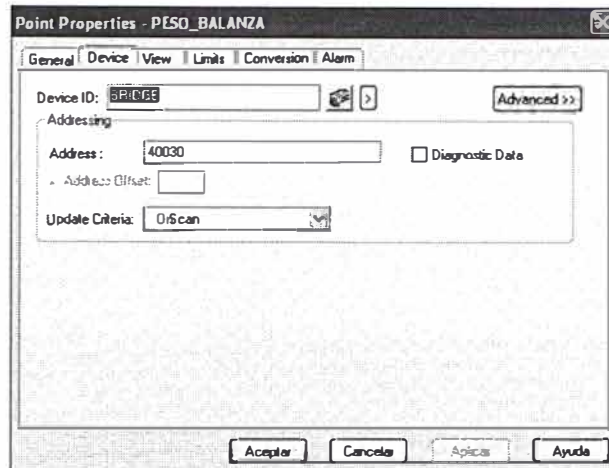
Luego se puede agregar las variables que se obtienen del controlador, del mismo modo se agregarán las variables internas del SCADA, la cuales se utilizarán para mostrar los datos en las operaciones de la aplicación.



**Figura 3.7** Agregando una nueva Variable



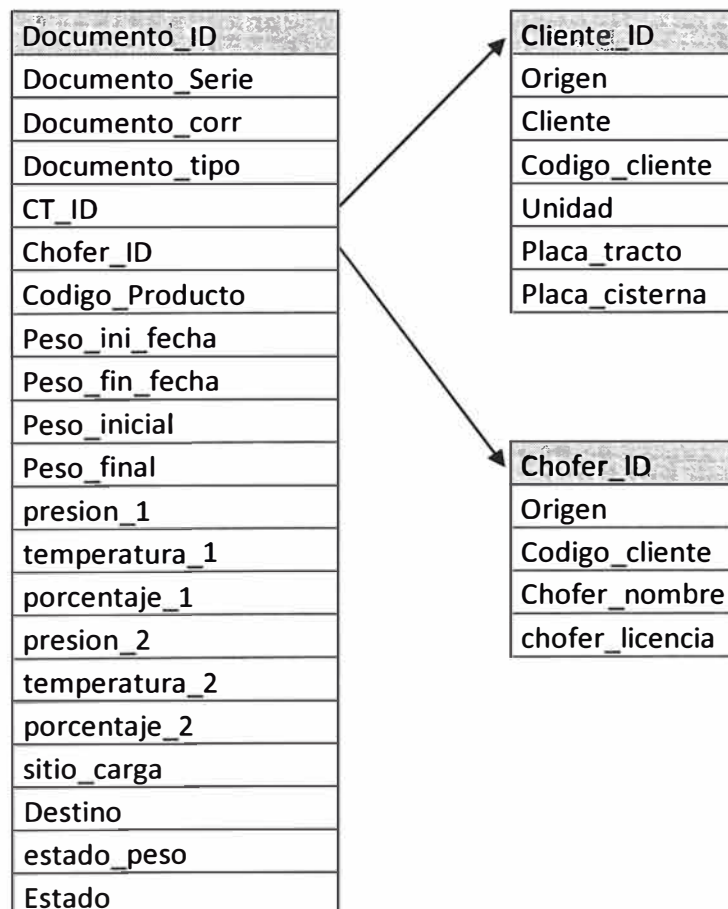
**Figura 3.8** Propiedades de la nueva variable



**Figura 3.9** Dirección física de la nueva variable (Solo para variables externas)

### 3.3.2 Configuración de la Base de Datos

De acuerdo al tipo de operación se establecerán las siguientes tablas, las cuales podrán ser escritas/leídas de acuerdo a la consulta/operación que se esté realizando. Tomaremos como base una tabla llamada documento, la cual tendrá los datos principales de cada operación y estas tablas tendrán datos en común entre ellos, como son los choferes y clientes, para ellos se crearán tablas adicionales.



**Figura 3.10** Detalle relación de tablas de datos

A continuación, el script de creación de la tabla documento, como se vé en ella se define el nombre de la tabla, nombre de las columnas y tipo de variable

```
USE [CIMPPLICITY]
GO

CREATETABLE [dbo].[DOCUMENTO] (
    [DOCUMENTO_ID] [int] NOTNULL,
    [DOCUMENTO_SERIE] [varchar](50)NOTNULL,
    [DOCUMENTO_CORR] [varchar](50)NOTNULL,
    [DOCUMENTO_TIPO] [varchar](50)NOTNULL,
    [CT_ID] [int] NOTNULL,
    [CHOFER_ID] [int] NOTNULL,
    [CODIGO_PRODUCTO] [varchar](20)NOTNULL,
    [PESO_INI_FECHA] [datetime] NOTNULL,
    [PESO_FIN_FECHA] [datetime] NOTNULL,
    [PESO_INICIAL] [numeric](17, 4)NOTNULL,
    [PESO_FINAL] [numeric](17, 4)NOTNULL,
    [GRAVEDAD_ESP] [numeric](17, 4)NOTNULL,
    [PRESION_1] [numeric](17, 4)NOTNULL,
    [TEMPERATURA_1] [numeric](17, 4)NOTNULL,
    [PORCENTAJE_1] [varchar](50)NOTNULL,
    [PRESION_2] [numeric](17, 4)NOTNULL,
    [TEMPERATURA_2] [numeric](17, 4)NOTNULL,
    [PORCENTAJE_2] [varchar](50)NOTNULL,
    [SITIO_CARGA] [varchar](600)NOTNULL,
    [ESTADÓ] [varchar](50)NOTNULL,
    [ESTADO_PESO] [varchar](10)NOTNULL,
    [DESTINÓ] [varchar](50)NOTNULL,
    CONSTRAINT [PK_DOCUMENTO] PRIMARYKEYCLUSTERED
    (
        [DOCUMENTO_ID] ASC
    )WITH
    (PAD_INDEX=OFF,STATISTICS_NORECOMPUTE=OFF,IGNORE_DUP_KEY=OFF,ALLOW_ROW_LO
CKS=ON,ALLOW_PAGE_LOCKS=ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

	DOCUMENTO_ID	DOCUMENTO_SERIE	DOCUMENTO_CORR	CT_ID	CHOFER_ID	CODIGO_PRODUCTO	PESO_INI_FECHA	PES
1	83	001	000083	341	222	000010	2010-09-02 15:54:15.000	20
2	82	001	000082	340	148	000010	2010-09-02 13:50:07.000	20
3	81	001	000081	339	224	000010	2010-09-02 11:28:29.000	20
4	80	001	000080	338	137	000010	2010-09-02 11:04:22.000	20
5	79	001	000079	337	115	000010	1900-01-01 00:00:00.000	19
6	78	001	000078	336	42	000010	2010-09-02 10:46:22.000	20
7	77	001	000077	335	240	000010	2010-09-01 18:09:59.000	20
8	76	001	000076	335	240	000010	2010-09-01 18:09:56.000	20
9	75	001	000075	334	39	000010	2010-08-27 12:32:59.000	20
10	74	001	000074	316	260	000010	2009-12-10 17:40:39.000	20
11	73	001	000073	331	137	000010	2010-08-25 15:34:49.000	20
12	72	001	000072	333	146	000030	2010-08-25 15:12:29.000	20
13	71	001	000071	331	136	000010	1900-01-01 00:00:00.000	19

Figura 3.11 Detalle de la tabla llamada "DOCUMENTO" en la base de datos

A continuación, el script de creación de la tabla choferes:



```

USE [SIMPLICITY]
GO

CREATETABLE [dbo].[CHOFERES] (
    [CHOFER_ID] [int] IDENTITY(1,1) NOTNULL,
    [ORIGEN] [varchar] (50) NOTNULL,
    [CODIGO_CLIENTE] [varchar] (80) NOTNULL,
    [CHOFER_NOMBRE] [varchar] (80) NOTNULL,
    [CHOFER_LICENCIA] [varchar] (20) NOTNULL,
    CONSTRAINT [PK_CHOFERES] PRIMARYKEYCLUSTERED
    (
        [CHOFER_ID] ASC
    )
) WITH
(PAD_INDEX=OFF, STATISTICS_NORECOMPUTE=OFF, IGNORE_DUP_KEY=OFF, ALLOW_ROW_LO
CKS=ON, ALLOW_PAGE_LOCKS=ON) ON [PRIMARY]
) ON [PRIMARY]

GO

```

	CHOFER_ID	ORIGEN	CODIGO_CLIENTE	CHOFER_NOMBRE	CHOFER_LICENCIA
1	1	LIMAGAS		Adolfo Cruzado	D17944734
2	2	LIMAGAS		Alejandro Salazar	Q15999042
3	3	LIMAGAS		Antenor Morillas	Q18055198
4	4	LIMAGAS		Antero Fernandez	D18012241
5	5	LIMAGAS		Aparicio Encarnación	Q10881106
6	6	LIMAGAS		Carlos Lecca	D18088245
7	7	LIMAGAS		Cesar Baldeos	Q15722151
8	8	LIMAGAS		Cesar Zelada	Q25532422
9	9	LIMAGAS		Cohed Valdeirama	D17954279
10	10	LIMAGAS		Miguel del Aguila	Q10258358
11	11	LIMAGAS		Eduardo De La Torre	Q09713640
12	12	LIMAGAS		Felipe Lecca	D18103665
13	13	LIMAGAS		Felix Gandolfo	Q07991482
14	14	LIMAGAS		Felix Paucar	Q29593318
15	15	LIMAGAS		Fernando Angulo	D17894837

**Figura 3.12** Detalle de la tabla llamada “choferes” en la base de datos

A continuación, el script de creación de la tabla clientes:

```

USE [SIMPLICITY]
GO

CREATETABLE [dbo].[CLIENTES] (
    [CLIENTE_ID] [int] IDENTITY(1,1) NOTNULL,
    [ORIGEN] [varchar] (50) NOTNULL,
    [CLIENTE] [varchar] (80) NOTNULL,
    [CODIGO_CLIENTE] [varchar] (80) NOTNULL,
    [UNIDAD] [varchar] (80) NOTNULL,
    [PLACA_TRACTO] [varchar] (20) NOTNULL,
    [PLACA_CISTERNA] [varchar] (20) NOTNULL,
    CONSTRAINT [PK_CLIENTES] PRIMARYKEYCLUSTERED
    (
        [CLIENTE_ID] ASC
    )
)

```



```

) WITH
(PAD_INDEX=OFF, STATISTICS_NORECOMPUTE=OFF, IGNORE_DUP_KEY=OFF, ALLOW_ROW_LO
CKS=ON, ALLOW_PAGE_LOCKS=ON) ON [PRIMARY]
) ON [PRIMARY]

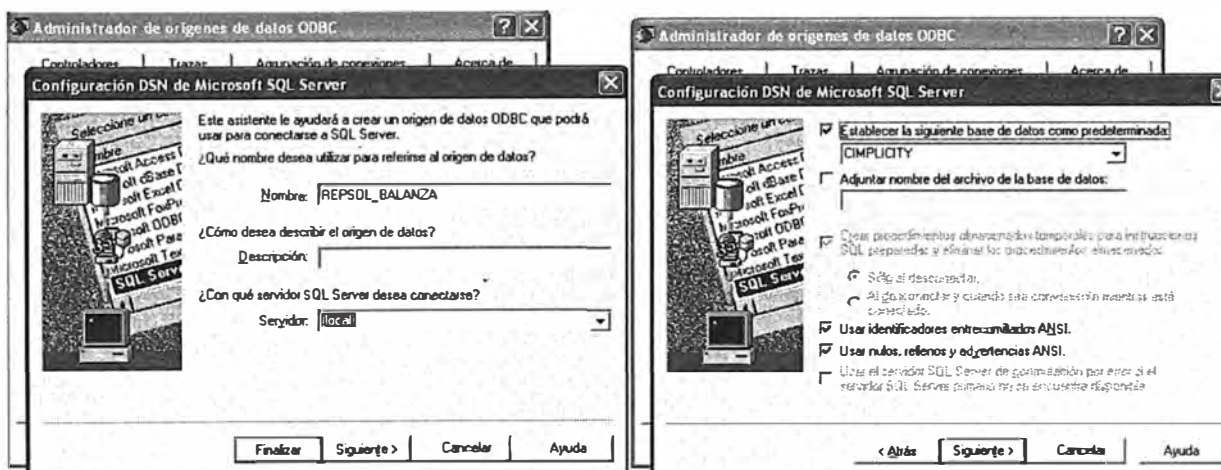
```

GO

CUENTE_ID	ORIGEN	CUENTE	CODIGO_CUENTE	UNIDAD	PLACA_TRACTO	PLACA_CISTERNA
90	PLUSPETROL	PLUSPETROL	Energigas		YQ-1700	ZQ-1122
91	PLUSPETROL	PLUSPETROL	Energigas			XP-3645
92	PLUSPETROL	PLUSPETROL	Energigas		YI-5619	ZI-7452
93	PLUSPETROL	PLUSPETROL	Esta Gas Peru			XI-7360
94	PLUSPETROL	PLUSPETROL	Exacto Gas		YI-2129	ZK-1083
95	PLUSPETROL	PLUSPETROL	Exacto Gas		YI-3148	ZG-8973
96	PLUSPETROL	PLUSPETROL	Flama Gas		YG-7286	ZG-8985
97	PLUSPETROL	PLUSPETROL	Flama Gas		YI-7253	ZI-5044
98	PLUSPETROL	PLUSPETROL	Flama Gas		YG-9183	ZI-3313
99	PLUSPETROL	PLUSPETROL	Flama Gas		YI-4251	ZI-3642
100	PLUSPETROL	PLUSPETROL	Flama Gas		YI-4269	ZI-4893
101	PLUSPETROL	PLUSPETROL	Flama Gas		YG-9183	ZG-8986
102	PLUSPETROL	PLUSPETROL	Fulgas		YC-1677	ZI-5029
103	PLUSPETROL	PLUSPETROL	Fulgas		YI-5957	ZI-5027
104	PLUSPETROL	PLUSPETROL	Fulgas			ZI-5027

**Figura 3.13** Detalle de la tabla llamada “clientes” en la base de datos

Estas son las principales tablas de datos, sin embargo a lo largo de la implementación se generarán tablas que nos ayudarán a identificar o diferenciar entre los productos a cargar o descargar, lugares de destino, etc. Ahora se configurará en el administrador de origen de base de datos (ODBC) un DSN de sistema que nos permite acceder a la base de datos CIMPLICITY, la cual contiene las tablas de datos que usaremos en esta aplicación, las figuras 3.14 y 3.15 muestran las imágenes de configuración y pruebas de conectividad de la base de datos creada.



**Figura 3.14** Detalle de la configuración del ODBC



**Figura 3.15** Detalle del resultado de las pruebas de conectividad con base de datos

Con esto a partir de ahora se podrá acceder a la base de datos con una consulta de este tipo:

```
I& = SQLRequest("dsn=repsof_balanza;uid=sa;PWD=;",CADENA_B,p,,4,True)
```

Donde CADENA\_B, representa el requerimiento a la base de datos, ejemplo: "select ... from ...", "inset into ... values .." , etc.

### 3.3.3 Desarrollo de Entorno Gráfico

Antes de desarrollar las pantallas de operación se evaluará de acuerdo al procedimiento de operación realizado para el despacho.

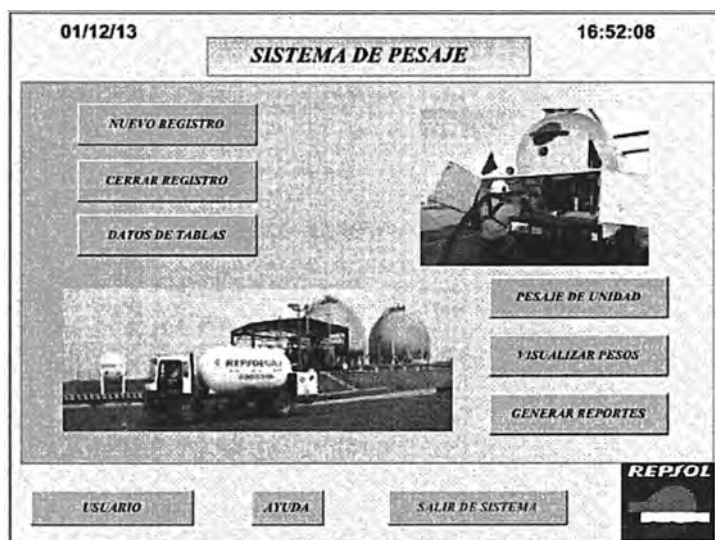
Primeramente se establecerá los usuarios y permisos respectivos:

- **SYSMGR:** Administrador de sistema, es el único que puede abrir la aplicación en modo edición y realizar cambios a nivel de programación y configuración.
- **SUPERVISOR:** Tiene todos los permisos para operar el sistema y agregar o quitar usuarios.
- **OPERADOR:** Tiene solo los permisos necesarios para operar el sistema.

User ID	Role ID	User Name
ADMINISTRATOR	SYSMGR	
PU10675	SUPERVISOR	PU10675
PU10040	SUPERVISOR	PU10040
RY48914	OPERADOR	RY48914
RY06241	OPERADOR	RY06241
RY06029	OPERADOR	RY06029
PU30436	OPERADOR	PU30436
PU10949	OPERADOR	PU10949
PU10926	OPERADOR	PU10926

**Figura 3.16** Listado de usuarios configurados en el sistema

Definimos una pantalla inicial llamada MENÚ, desde la cual accederemos a las diferentes pantallas. Antes de desarrollar las pantallas de operación, a las cuales se accede según la necesidad de operación.



**Figura 3.17** Imagen de la Pantalla "MENU"

La pantalla inicial contiene los siguientes accesos directos:

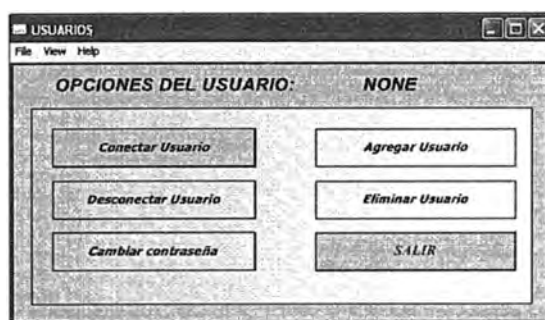
- **USUARIO:** Acceso directo a la pantalla de administración de usuarios, en necesario iniciar sesión para poder ver los datos de la aplicación y poder operarlo.
- **AYUDA:** Acceso directo al manual de operación del sistema.
- **NUEVO RESISTRO:** Accede a la pantalla de Aperturo de registros, en ella se ingresan los datos de las unidades a despachar antes de ser pesadas.
- **CERRAR REGISTRO:** Acceso a la pantalla en la que se completan los datos de registro y se realiza la impresión de tickets de pesaje, impresión de guías, etc.
- **DATOS DE TABLAS:** Acceso a la pantalla de selección para corregir o actualizar datos de clientes, transportistas, lugares de destino, etc.
- **PESAJE DE UNIDAD:** Acceso a pantalla en la que se realiza la captura de pesos, inicial y final.
- **VISUALIZAR PESOS:** Acceso a pantalla de ultimos pesos, utilizada mayormente por el operador de Caseta, permite localizar rapidamente los ultimos pesos tomados, para informar el peso a los choferes.
- **GENERAR REPORTES:** Permite seleccionar el tipo de reporte que sequiere obtener.
- **SALIR:** Cierra la pantalla de SCADA para salir del sistema.

Para el desarrollo de esta pantalla realizamos el dibujo con las herramientas del entorno cimedit del software, luego a cada boton se le asignará una acción de modo que abra la ventana correspondiente o el archivo en caso de el manual, la figura 3.18 muestra una acción que abre un archivo al pulsar un boton.



**Figura 3.18** Detalle acción asignada a un botón

Diseñamos la pantalla emergente que nos permita administrar los usuarios, en ella, los botones habilitados dependerán de el usuario que se encuentre conectado. En las imágenes a continuación se vé el estado de la pantalla sin ningún usuario conectado, con un usuario del tipo supervisor y con un usuario del tipo operador logeado respectivamente.



**Figura 3.19** Pantalla “USUARIOS” sin ningún ususario logeado

Para permitir iniciar sesión en el proyecto, debemos asignar al boton iniciar sesión, el siguiente Script, la penúltima instrucción es la que permite iniciar sesión en el proyecto actual.

```

Sub OnMouseDown1(x As Long, y As Long, flags As Long)
Dim directorio As String
Dim project As CimProject
Set project = CreateObject("CimProject")
directorio = Environ("SITE_Root")
project.OpenLocalProject directorio & "REPSOL_PESAJE.gef"
CimLogin project.projectname
End Sub

```

Y para crear un nuevo usuario en el sistema, ejecutaremos el siguiente script, el cual envía una ventana emergente que pregunta el nombre, contraseña y rol del nuevo usuario, a partir de la etiqueta "NOMBRE" y a partir de la etiqueta "GUARDAR", se agrega al proyecto.

```

Sub OnMouseDown(x As Long, y As Long, flags As Long)
Dim new_user As String
Dim new_password As String
Dim re_new_password As String
Dim role As String
NOMBRE:
TITULO$="NUEVO USUARIO"
Begin Dialog UserDialog 81,64,128,40,TITULO$
CancelButton 80,24,40,14
OKButton 80,8,40,14
TextBox 4,8,68,14,.TextBox1,0
End Dialog
Dim d1 As UserDialog
d1.TextBox1 = ""
rc1% = Dialog(d1)
new_user = UCase$(d1.TextBox1)
If rc1% <> -1 Then 'boton diferente de OK
Exit Sub
Else
If d1.TextBox1 = "" Then
msgbox "DEBE INGRESAR UN NOMBRE PARA EL NUEVO USUARIO"

```

```

GoTo NOMBRE
Else
MsgBox "EL NUEVO USUARIO ES: " & new_user
End If
End If
new_password = AskPassword$("INGRESE PASSWORD:")
reingreso:
re_new_password = AskPassword$("CONFIRME PASSWORD:")
If new_password <> re_new_password Then
msgbox "ERROR EN CONFIRMACION DE PASSWORD"
GoTo reingreso
End If
ROLE_ASSIGNADO:
Dim Roles$(2)
Roles$(0) = "SUPERVISOR"
' Roles$(1) = "USER"
Roles$(1) = "OPERADOR"
Begin Dialog FindTemplate 16,32,220,48,"Find"
Text 4,8,80,8,"ROLES DISPONIBLES:"
DropListBox 90,6,64,80,Roles,.WhichField
OKButton 160,7,40,14
CancelButton 160,27,40,14
End Dialog
Dim FindDialog As FindTemplate
FindDialog.WhichField = 0
Dialog FindDialog
Select Case respuesta%
Case 6 'boton YES
GUARDAR:
Dim project As CimProject
Dim directorio As String
directorio = Environ("SITE_Root")

Set project = CreateObject("CimProject")
project.OpenLocalProject directorio & "REPSOL_PESAJE.gef"
project.DynamicMode = True

```



```

Dim users As CimUserList
Dim user As CimUser
Set users = project.Users
Set user = users.New(new_user)
user.RoleID = Roles(FindDialog.WhichField)
user.PasswordNeeded = True
user.Password = new_password
user.Enabled = True
user.PasswordExpiresDays = 0
user.UserName = new_user
users.Save user, True
Case 7 'boton NO
GoTo nombre
End Select

```

End Sub

Luego según el nivel de acceso daremos a los botones las propiedades de visibilidad y ejecución, como se muestra en la figura 3.20.

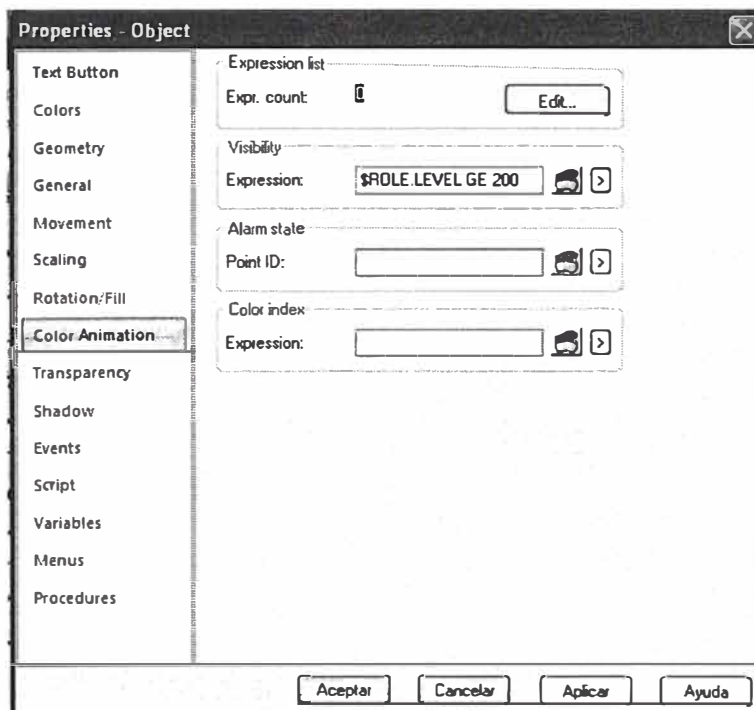
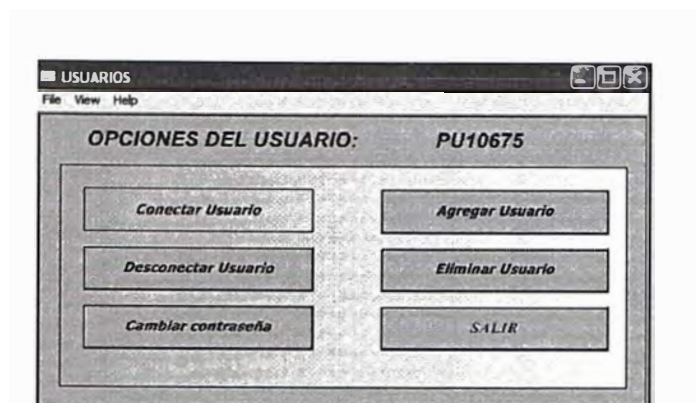
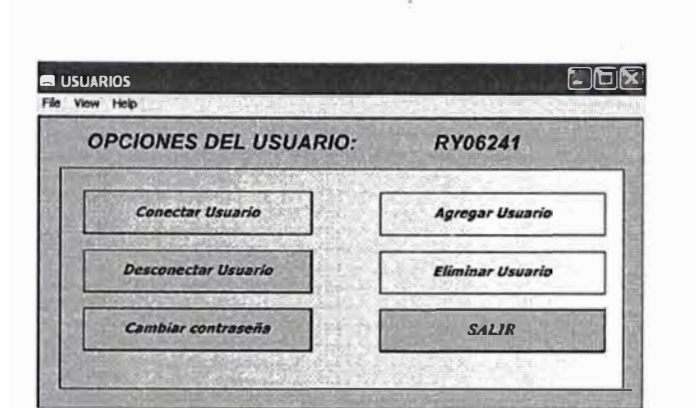


Figura 3.20 Detalle de animación de visibilidad por usuario



**Figura 3.21** Pantalla "USUARIOS" con un Supervisor logeado



**Figura 3.22** Pantalla "USUARIOS" con un operador logeado

Según la metodología de operación definimos la pantalla NUEVO REGISTRO, en la cual se genera el número de registro y se completan los datos del tipo de operación a realizar y el tipo de unidad a despachar. Para generar un nuevo registro previamente limpiamos todos los datos de modo que se tenga llenar nuevamente los datos antes de grabar. En el código se inicia declarando las variables auxiliares a utilizar y luego inicializa o borra las demás variables.

*Sub LIMPIAR()*

*Dim LISTA() As Variant*

*Dim ELECCION As Integer*

*Dim DOC\_ID As Variant*

*Dim M(), q() As Variant*

*Dim CADENA As String*

*Dim project As CimProject*

*Dim directorio As String*

*directorio = Environ("SITE\_Root")*

*Set project = CreateObject("CimProject")*

*project.OpenLocalProject directorio & "REPSOL\_PESAJE.gef"*



```

CLAVE = ""
POINTSET "CLIENTE_NOMBRE", ""
POINTSET "TRANSPORTISTA_NOMBRE", ""
POINTSET "PRODUCTO_CODIGO", "000033"
POINTSET "PRODUCTO_DESCRIPCION", "GAS LICUADO DE PETROLEO"
POINTSET "UNIDAD", ""
POINTSET "PLACA_CISTERNA", ""
POINTSET "PLACA_TRACTO", ""
POINTSET "CHOFER_NOMBRE", ""
POINTSET "ESTADO_REGISTRO", ""
POINTSET "ORIGEN", ""
PointSet "DESTINO_DIRECCION", ""
PointSet "DESTINO_CODIGO", 0
PointSet "DESTINO_FACTOR", 0.0
PointSet "doc_base_id", 0
Next
End Sub

```

Luego de ello habilitaremos los botones de la parte inferior en la que se escogerá el tipo de venta y de acuerdo a ello se llenarán sugerencias en los campos obligatorios los cuales estarán filtrados por el tipo de venta.

Al crear un nuevo registro se debe escoger el tipo de operación (Venta directa, Limagas, Secundarios, etc., las cuales son los tipos de operaciones y nombres utilizados en la planta de almacenamiento PPAL Repsol), de acuerdo a la elección arriba se podrá ir filtrando los datos en las opciones cliente, transportista, unidad, etc. Al escoger datos de una unidad no registrada anteriormente, automáticamente se agregará a la base de datos. Por ejemplo al seleccionar el tipo venta directa se ejecutará el siguiente script, el cual consultará en la base de datos las unidades y choferes relacionados a este tipo de despacho.

```

Sub OnMouseDown()
Dim LISTA() As Variant
Dim ELECCION As Integer
Dim DOC_ID As Variant
Dim M(), q() As Variant

```

```

Dim CADENA As String
Dim project As CimProject
Dim directorio As String
directorio = Environ("SITE_Root")
Set project = CreateObject("CimProject")
project.OpenLocalProject directorio & "REPSOL_PESAJE.gef"

```

Apartir de aquí se inicializan las variables de acuerdo al tipo de operación escogida, en este caso "Venta Directa", luego consulta a la base de datos los clientes registrados en ese tipo de operación y los transportistas ligados a este tipo de operación

```

operacion = 2
PointSet "OPERACION",operacion
PointSet "origen","VENTA DIRECTA"
PointSet "CLIENTE_NOMBRE","PRIMAX"
'esta parte le da un nombre de transportista cuando escoge le tipo de registro
ORIGEN = PointGet("ORIGEN")
cliente = PointGet("cliente_nombre")
cadena = "Select distinct(codigo_cliente) From clientes where origen = '" & origen & "' and
cliente = '" & cliente & "' ORDER BY CODIGO_CLIENTE "
I& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;pwd=" & clave,cadena,m,,4,True)
If UBOUND(M)>0 Then
    PointSet "TRANSPORTISTA_NOMBRE",m(1,0)
cadena = "SELECT DISTINCT(TRANSPORTISTA_RUC) FROM CLIENTES WHERE
CODIGO_CLIENTE = '" & m(1,0) & "' AND LEN(TRANSPORTISTA_RUC)=11"
I& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;pwd=" & clave,cadena,m,,4,True)
nombretransp = PointGet("TRANSPORTISTA_NOMBRE")
' con el transportista le da una unidad, asi como tambien un chofer todo esto de manera
alfabetica, es el unico criterio
Cadena = "select distinct unidad,placa_tracto,placa_cisterna,MTC_TRACTO,
MTC_BOTELLA from clientes WHERE cliente ='" & cliente & "' and CODIGO_CLIENTE
=''" & nombretransp & "' And Len(UNIDAD)>0 And UNIDAD <> '-ORDER BY UNIDAD"

I& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;pwd=" & clave,cadena,m,,4,True)

```

Apartir de aquí luego de consultar en la base de datos, establece los datos de Unidad, Placa, Tracto y chofer.

```

If UBOUND(M)>0 Then
    If M(1,0) = "-" Or M(1,0) = "" Or M(1,0) = " " Then
        PointSet "UNIDAD","--"
        cliente = PointGet("TRANSPORTISTA_NOMBRE")
        ' parte que escoge el tracto
        cadena = "SELECT DISTINCT PLACA_TRACTO, mtc_tracto FROM CLIENTES
where len(placa_tracto)>2 And (cliente ="" & cliente & """" & " OR CODIGO_cliente ="" &
cliente & "")"
I& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;pwd=" & clave,cadena,m,,4,True)
        If UBOUND(M)>0 Then
            PointSet "PLACA_TRACTO",m(1,0)
        End If
        'parte q escoge la cisterna
        cadena = "SELECT DISTINCT PLACA_cisterna, mtc_botella FROM CLIENTES
where len(placa_cisterna)>2 And ( cliente ="" & cliente & """" & " OR CODIGO_cliente ="" &
cliente & "")"
I& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;pwd=" & clave,cadena,m,,4,True)
        If UBOUND(M)>0 Then
            PointSet "PLACA_CISTERNA",M(1,0)
        End If
    Else
        PointSet "UNIDAD",m(1,0)
        PointSet "placa_tracto",m(1,1)
        PointSet "placa_cisterna",m(1,2)
    End If
    ' parte q escoge el chofer y luego sus documentos
    cadena = "Select distinct chofer_nombre from choferes where
CODIGO_CLIENTE ="" & nombretransp & """"
I& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;pwd=" & clave,cadena,m,,4,True)
    If UBOUND(M)>0 Then
        PointSet "CHOFER_NOMBRE",M(1,0)
        ' con el chofer escoge el documento

```

```

cadena2 = "Select distinct chofer_licencia from choferes where
chofer_nombre =" & M(1,0) & ""
I& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;pwd=" &
clave,cadena2,q,,4,True)

If ubound(q)>0 Then
    PointSet "CHOFER_DOC",q(1,0)
    PointSet "CHOFER_tipo_DOC","LICENCIA"
End If
End If
Else
    Exit Sub
End If
Else
    Exit Sub
End If
End Sub

```

**Figura 3.23** Pantalla "NUEVO REGISTRO"

Este modo de ingreso de datos, permite reducir el tiempo de ingreso de datos para unidades que ya existan en la base de datos y en el caso de unidades nuevas, a la vez

que permite digitar para que se guarden nuevos datos automáticamente, los campos de unidad, placa y tracto contarán con una lista desplegable que permitirá escribir también para agregar y predecir cuando se trate de unidades ya guardadas, por ejemplo mostraremos el script realizado para agregar dato de placa, la primera parte declara las variables a utilizar y hace la consulta de unidades a la tabla clientes de la base de datos.

```

Sub OnMouseDown ()
Dim LISTA() As Variant
Dim M() As Variant
Dim project As CimProject
Dim directorio As String
directorio = Environ("SITE_Root")
constancia = PointGet("CONSTANCIA_INSCRIPCION")
cliente = PointGet("TRANSPORTISTA_NOMBRE")
cadena = "SELECT DISTINCT PLACA_cisterna, mtc_botella FROM CLIENTES where
len(placa_cisterna)>2 And ( cliente ="" & cliente & "" & " OR CODIGO_cliente ="" & cliente
& "")"
I& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;pwd=" & clave,cadena,m,,4,True)
If UBOUND(M)>0 Then
    ReDim Preserve lista(1 To ubound(m))
Else
    Exit Sub
End If

For i= 1 To ubound(m)
    lista(i) = m(i,0)
Next i

A partir de aquí ya con los datos llenados por la instrucción "For", mostrará un cuadro
de diálogo en la que se podrá escoger o escribir la placa de la cisterna.
ELEGIR:
Begin Dialog Lista 76,44,232,104,"Elija una Cisterna"
    Text 4,8,60,8,"Lista de Cisternas:"
    ComboBox 68,6,96,102,lista,.WhichField
    OKButton 179,7,40,14
    CancelButton 179,28,40,14
End Dialog

```

```

Dim PRODUCTOS As Lista
PRODUCTOS.WhichField = 0
acepta = Dialog(PRODUCTOS)
eleccion = PRODUCTOS.WhichField

```

```

codigo=      eleccion
If acepta = -1 Then 'ok
    GoTo continuacion
Else
    If acepta = 0 Then 'cancel
        Exit Sub
    End If
End If

```

```
End If
```

CONTINUACION:

```
PointSet "PLACA_CISTERNA",CODIGO
```

```
cadena = "SELECT DISTINCT PLACA_cisterna, mtc_botella FROM CLIENTES where
PLACA_cisterna="" & CODIGO & "" and len(placa_cisterna)>2 And ( cliente ="" & cliente &
"" & " OR CODIGO_cliente ="" & cliente & "")"
```

```
I& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;pwd=" & clave,cadena,m,,4,True)
```

```
End Sub
```

El siguiente paso en la operación es grabar para tener este dato disponible para las otras estaciones antes de grabar el sistema verifica que la unidad seleccionada exista en el sistema, de no existir la agrega para tenerla disponible para un registro posterior.

*Sub guardar*

```
Dim cadena1 As String
```

```
Dim CADENA2 As String
```

```
Dim CADENA As String
```

```
Dim CADENA3 As String
```

```
Dim m() As Variant
```

```
Dim p() As Variant
```

```
Dim ct_ID As Integer
```

```
Dim choferid As Integer
```

```
Dim project As CimProject
```

```
Dim directorio As String
```

```

USUARIO = PointGet("$USER")
PointSet "USUARIO",USUARIO
PointSet "DOC_BASE_ID","0"
Dim prod_desc As String
directorio = Environ("SITE_Root")
Set project = CreateObject("CimProject")
project.OpenLocalProject directorio & "REPSOL_PESAJE.gef"
CLAVE = project.ProjectPassword

```

Al momento de grabar consulta el último número de documento y asigna el siguiente número al nuevo registro guardado.

```

CADENA = "SELECT DISTINCT MAX(DOCUMENTO_ID) FROM DOCUMENTO"
k& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;pwd=" & clave,cadena,M,,4,True)
If UBOUND(M)> 0 Then 'EXISTE AL MENOS UN DATO
    If ISNULL(m(1,0)) Then
        NUMREG = 1
    Else
        NUMREG = VAL(M(1,0)) + 1
    End If
Else
    NUMREG = 1
End If
PointSet "DOC_ID",FORMAT(NUMREG,"000000")
CADENA = "SELECT DISTINCT MAX(DOCUMENTO_SERIE) FROM DOCUMENTO
WHERE DOCUMENTO_SERIE <>""
k& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;pwd=" & clave,cadena,M,;4,True)
If UBOUND(M)> 0 Then 'EXISTE AL MENOS UN DATO
    If ISNULL(m(1,0)) Then
        SERIE = 1
    Else
        SERIE = VAL(M(1,0)) '+ 1 EN EL CASO DE LA SERIE, MANTENEMOS EL
VALOR. YA QUE NORMALMENTE EL QUE VARIA ES EL CORRELATIVO
    End If
Else
    SERIE = 1

```



End If

PointSet "DOCUMENTO\_SERIE",FORMAT(SERIE,"000")

CADENA = "SELECT DISTINCT MAX(DOCUMENTO\_CORR) FROM DOCUMENTO  
WHERE DOCUMENTO\_CORR <>""

k& = SQLRequest("dsn=REPSOL\_BALANZA;uid=sa;pwd=" & clave,cadena,M,,4,True)

If UBOUND(M)> 0 Then 'EXISTE AL MENOS UN DATO

    If ISNULL(m(1,0)) Then

        CORRE = 1

    Else

        CORRE = VAL(M(1,0)) + 1

    End If

Else

    CORRE = 1

End If

PointSet "DOCUMENTO\_CORR",FORMAT(CORRE,"000000")

corre = PointGet("DOCUMENTO\_CORR")

serie = PointGet("DOCUMENTO\_SERIE")

ORIGEN = PointGet("ORIGEN")

nombrecliente = PointGet("CLIENTE\_NOMBRE")

nombretransp = PointGet("TRANSPORTISTA\_NOMBRE")

codigoprod = PointGet("PRODUCTO\_CODIGO")

unidad = PointGet("UNIDAD")

cisterna = PointGet("PLACA\_CISTERNA")

tracto = PointGet("PLACA\_TRACTO")

nombrechofer = PointGet("CHOFER\_NOMBRE")

NUMDOC = PointGet("DOC\_ID")

DOCID = VAL(NUMDOC)

sitio = PointGet("sitio\_carga")

MOTIVO = PointGet("MOTIVO")

operacion = PointGet("operacion")

doc\_base1 = PointGet("DOC\_BASE\_ID")

doc\_base = VAL(DOC\_BASE1)

FECHA\_INI = PointGet("PESO\_INIC\_FECHA")

FECHA\_FIN = PointGet("PESO\_FIN\_FECHA")

HORA\_INI = PointGet("PESO\_INIC\_HORA")



```

HORA_FIN = PointGet("PESO_FIN_HORA")
PESO_INI = PointGet("PESO_INICIAL")
PESO_FIN = PointGet("PESO_FINAL")
cad1 = fecha_ini & " " & hora_ini
cad2 = fecha_fin & " " & hora_fin

```

Antes de grabar verifica que el cliente exista en la base de datos, de no existir agrega este datos a la tabla clientes, de igual manera con el dato de chofer y luego de esto graba el registro en la tabla.

```

CADENA = "SELECT * FROM clientes WHERE cliente = '" & nombrecliente & "' and
CODIGO_CLIENTE ='" & nombretransp & "' and placa_cisterna = '" & cisterna & "' and
placa_tracto = '" & tracto & "' AND UNIDAD = '" & UNIDAD & "'"
k& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;PWD=" & clave,cadena,M,,4,True)
'msgbox ubound(m)
If UBOUND(M) > 0 Then ' EXISTE ALGUN DATO EN LA TABLA
    GoTo consulta_chofer
Else
    CADENA1 = "INSERT INTO CLIENTES
(CLIENTE,CODIGO_CLIENTE,PLACA_CISTERNA,PLACA_TRACTO,UNIDAD,ORIGEN)
"
    CADENA2 = " VALUES ('" & NOMBRECLIENTE & "','" & nombretransp & "','" &
CISTERNA & "','" & TRACTO & "','" & UNIDAD & "','" & ORIGEN & "','" & "'"
    CADENA = CADENA1 & CADENA2
k& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;PWD=" & clave,cadena,p,,4,True)
End If
nombretransp = PointGet("TRANSPORTISTA_NOMBRE")
cadena = "select * from choferes where origen = '" & origen & "' and Chofer_nombre ='" &
nombrechofer & "' and chofer_licencia = '" & datolic & "' and codigo_cliente = '" &
nombretransp & "'"
k& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;PWD=" & clave,cadena,M,,4,True)
'msgbox ubound(m)
If UBOUND(M) > 0 Then ' EXISTE ALGUN DATO EN LA TABLA
    GoTo DOCUMENTO
Else

```

```

CADENA          =          "INSERT          INTO          Choferes
(ORIGEN,CHOFER_NOMBRE,CHOFER_LICENCIA,CODIGO_CLIENTE) VALUES (" &
origen & "," & nombrechofer & "," & datolic & "," & nombretresp & ")"
```

```

k& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;PWD=" & clave,cadena,p,,4,True)
End If
```

DOCUMENTO:

```

CADENA = "SELECT * FROM clientes WHERE cliente = " & nombrecliente & " and
CODIGO_CLIENTE =" & nombretresp & " and placa_cisterna = " & cisterna & " and
placa_tracto = " & tracto & " AND UNIDAD = " & UNIDAD & ""
```

```

k& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;PWD=" & clave,cadena,M,,4,True)
```

```

If UBOUND(M) > 0 Then ' EXISTE ALGUN DATO EN LA TABLA
```

```

    ct_id = cint(m(1,0))
```

```

End If
```

```

cadena = "select * from choferes where origen = " & origen & " and Chofer_nombre =" &
nombrechofer & " and chofer_licencia = " & datolic & ""
```

```

k& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;PWD=" & clave,cadena,M,,4,True)
```

```

If UBOUND(M) > 0 Then ' EXISTE ALGUN DATO EN LA TABLA
```

```

    choferid = cint(m(1,0))
```

```

End If
```

```

CADENA1 = "SELECT Documento_id FROM documento WHERE documento_ID -"
&NUMDOC & ""
```

```

k& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;PWD=" & clave,CADENA1,p,,4,True)
```

```

If UBOUND(P) > 0 Then
```

```

    NUMDOC = val(NUMDOC) + 1
```

```

    DOCID = VAL(NUMDOC)
```

```

    PointSet "DOC_ID",numdoc
```

```

End If
```

```

CADENA1          =          "insert          into          documento
(Documento_id,ct_id,CHOFER_ID, CODIGO_PRODUCTO,doc_INI_FECHA,estado,CLIE
NTE_FACTURA,CLIENTE_GUIA,sitio_carga,motivo,OPERACION,gravedad_Esp,scop,sc
op_1,DESTINO,USUARIO,PESO_INICIAL,PESO_INI_FECHA,documento_base_id,"
cadena2 = "documento_serie,documento_corr) "
```

```

cadena3 = "VALUES (" &NUMDOC & "," & CT_ID & "," & CHOFERID & "," &
CODIGOPROD & "," & fecha1 & "," & 'ABIERTO' & "," & FACTCLIENTE & "," & GUIACLIENTE
& "," & sitio & "," & motivo & "," & OPERACION & "," & grav & "," & scop & "," & scop_1
& "," & "
```

```
CADENA4 = DESTINO_BREVE & "," & USUARIO & "," & PESO_INI & " " &
fecha_ini_cad & "," & doc_base & "," & serie & "," & corre & ")")
```

```
cadena = cadena1 & cadena2 & CADENA3 & cadena4
```

```
k& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;PWD=;";CADENA,p,,4,True)
```

```
Msg.Open "Datos grabado correctamente",0,True,True
```

```
Sleep 2000
```

```
Msg.Close
```

```
PointSet "TIPO_DATOS_PRODUCTO", ""
```

```
Exit Sub
```

```
evadir:
```

```
msgbox "No se grabaron los datos, revisar la conexion a la Base de Datos"
```

```
End Sub
```

Luego de completado el registro y grabar, se puede realizar un nuevo registro o realizar cualquier otra operación en otra pantalla, del mismo modo una vez grabado el registro en la base de datos, ya se encuentra disponible para ser visto o grabar información adicional en el a través de cualquiera de las estaciones de operación.

El siguiente paso en las operaciones es la captura del peso inicial y posteriormente la captura del peso final, para que por medio de una sustracción puede obtener la cantidad de producto que se esta despachando en la unidad.

Figura 3.24 Imagen de la Pantalla "PESAJE DE UNIDAD"

El sistema tiene una alarma visual y sonora, la cual se activa cuando el peso de la balanza empieza a aumentar desde un valor de 0 hasta un valor mayor a 100 Kg, para no tomar en cuenta fallas de calibración de la Balanza de pesaje. En esta pantalla se pueden realizar 3 operaciones distintas:

- Captura de pesos: Para ello se debe seleccionar el registro de la unidad a la cual se quiere capturar el peso, esto se realiza a través del boton pendientes, el cual por programación hace una consulta a la base de datos del siguiente tipo:

En la primera parte se defina una grilla en la que se mostrarán los datos que consultaremos en este caso los registros pendientes, en la segunda parte carga los valores a mostrar.

```

Private cimOleObj As CIMPGRIDLib.ICCimpGrid 'DEFINIR LA VARIABLE TIPO GRILLA
Private scrn As GefObjectModel.DGefScreen 'DEFINIR LA VARIABLE TIPO VENTANA
Private a() As Variant 'MATRIZ PARA ALMACENAR DATOS DE LA CONSULTA sql
Private column(410) As Integer 'MATRIZ PARA LLENAR LA GRILLA
Private cadena As String
Public clave As String
Sub scrn_init
Dim project As CimProject
Dim directorio As String
directorio = Environ("SITE_Root")
Set project = CreateObject("CimProject")
project.OpenLocalProject directorio & "REPSOL_PESAJE.gef"
Set scm = CimGetScreen
scm.refresh TRUE
cimoleobj.Remove_Column -1 'LIMPIO LA GRILLA
cimoleobj.Add_Column 1, 70, "REGISTRO"
cimoleobj.Add_Column 2, 70, "PLACA"
cimoleobj.Add_Column 3, 60, "ESTADO"
cimoleobj.Add_Column 4, 60, "UNIDAD"
cimoleobj.Add_Column 5, 190, "TRANSPORTISTA"
cimoleobj.Delete_allrows 'LIMPIAR DATOS DE LA GRILLA
cadena1 = "SELECT DOCUMENTO_ID, PLACA_CISTERNA, ESTADO_PESO, UNIDAD,
CLIENTES.CODIGO_CLIENTE "

```

```
cadena2 = "FROM DOCUMENTO,CLIENTES WHERE DOCUMENTO.CT_ID =
CLIENTES.CLIENTE_ID and estado = 'ABIERTO' ORDER BY DOCUMENTO_ID DESC"
CADENA = CADENA1 & CADENA2
```

```
I&= SQLRequest("dsn=REPSOL_BALANZA;UID=sa;PWD=" & clave, CADENA, a,, 3, TRUE)
```

```
load_grid
```

```
End Sub
```

```
Sub load_grid
```

```
  If ubound(a) > 0 Then
```

```
    For i=1 To ubound(a)
```

```
      a(i,0) = FORMAT(a(i,0),"000000")
```

```
    Next
```

```
    For i = 0 To ubound(a,2) 'otra manera de obtener la cantidad de columnas
```

```
      column(i) = i ' i+1 (sumo 1 para correr los datos una columna hacia la
derecha)
```

```
    Next
```

```
    For i = 1 To ubound(a) 'select the row
```

```
      cimOleObj.Add_Row i-1, str(i)
```

```
      For j = 0 To ubound(a,2)
```

```
        If column(j) >= 0 Then
```

```
          cimOleObj.Set_rowitem i-1,column(j),a(i,j)
```

```
        End If
```

```
      Next
```

```
    Next
```

```
  End If
```

```
End Sub
```

```
Sub OnDbfClick()
```

```
Dim popup As GefObjectModel.DGefScreen
```

```
  'get the row selected
```

```
row = cimoleobj.Row_Selected
```

```
If row < 0 Then
```

```
  MsgBox "No ha seleccionado ningun item!","Advertencia"
```

```
  Exit Sub
```

```
End If
```

```
PointSet "doc_id", CimOleObj.Retrieve_RowItem(row,0)
```

```
Dim LISTA() As Variant
```

*Dim M() As Variant*

*Dim punto As New point*

*clave = ""*

*sleep 1000*

*doc\_id = PointGet("DOC\_ID")*

*If doc\_id = "" Then*

*Exit Sub*

*End If*

*CONTINUACION:*

*PointSet "CONFIRMA\_GRAB",0*

*PointSet "TIPO\_DATOS\_PESAJE","M"*

*cadena1= "SELECT CLIENTE, CODIGO\_CLIENTE, PLACA\_CISTERNA,  
PLACA\_TRACTO, UNIDAD, PESO\_INI\_FECHA,PESO\_FIN\_FECHA, PESO\_INICIAL,  
PESO\_FINAL,OBSERVACION\_1,OBSERVACION\_2,ESTADO,ESTADO\_PESO,SITIO\_C  
ARGA , ORIGEN "*

*cadena2= "FROM CLIENTES,documento where documento\_id = " & doc\_id & " and  
documento.ct\_id = clientes.cliente\_id"*

*CADENA=CADENA1 & CADENA2*

*I& = SQLRequest("dsn=REPSOL\_BALANZA;uid=sa;pwd=" & clave,cadena,m,,4,True)*

*VALOR1=format(M(1,5),"dd/mm/yyyy hh:mm:ss")*

*VALOR2=format(M(1,6),"dd/mm/yyyy hh:mm:ss")*

*If left(VALOR1,10) = "01/01/1900" Then*

*fecha1=""*

*hora1=""*

*Else*

*fecha1=left(VALOR1,10)*

*hora1=Mid(valor1,12,8)*

*End If*

*If left(VALOR2,10) = "01/01/1900" Then*

*fecha2=""*

*hora2=""*

*Else*

*fecha2=left(VALOR2,10)*

*hora2=Mid(valor2,12,8)*

*End If*



```

If ubound(m) = 1 Then
    PointSet "CLIENTE_NOMBRE",m(1,0)
    PointSet "transportista_NOMBRE",m(1,1)
    PointSet "placa_cisterna",m(1,2)
    PointSet "placa_tracto",m(1,3)
    PointSet "unidad",m(1,4)
    PointSet "PESO_INIC_FECHA",fecha1
    PointSet "PESO_FIN_FECHA",fecha2
    PointSet "PESO_INIC_hora",hora1
    PointSet "PESO_FIN_hora",hora2
    PointSet "PESO_inicial",val(M(1,7))
    PointSet "PESO_final",val(M(1,8))
    If M(1,12) = "" Then
        POINTSET "PESOS_GRABADOS",0.0
    Else
        If M(1,12) = "I" Then
            POINTSET "PESOS_GRABADOS",1.0
        Else
            POINTSET "PESOS_GRABADOS",2.0
        End If
    End If

    PointSet "ESTADO_REGISTRO",M(1,11)
    PointSet "SITIO_CARGA",M(1,13)
    PointSet "ORIGEN",M(1,14)

    If Len(TRIM(M(1,12))) > 0 Then
        If M(1,12) = "I" Then
            PointSet "TIPO_PESAJE","SALIDA"
        Else
            PointSet "TIPO_PESAJE","SALIDA"
        End If
    Else
        PointSet "TIPO_PESAJE","INGRESO"
    End If
    PointSet "READY",TRUE

```

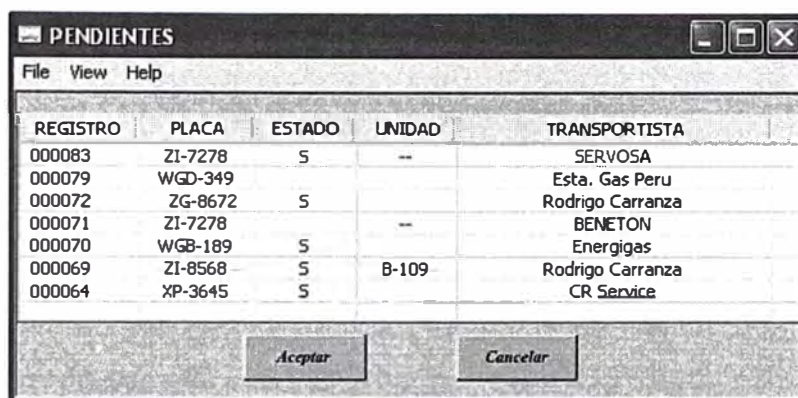
*Else*

*msgbox "Error en la consulta"*

*End If*

*End Sub*

Como resultado de esta consulta se llenan los datos de la grilla de esta forma, en ella se escoge el registro en el que se va trabajar para capturar alguno de sus pesos, en la figura 3.25 se muestra la grilla con los datos consultados.

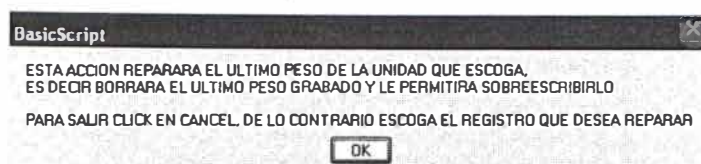


REGISTRO	PLACA	ESTADO	UNIDAD	TRANSPORTISTA
000083	ZI-7278	S	--	SERVOSA
000079	WGD-349			Esta. Gas Peru
000072	ZG-8672	S		Rodrigo Carranza
000071	ZI-7278		--	BENETON
000070	WGB-189	S		Energigas
000069	ZI-8568	S	B-109	Rodrigo Carranza
000064	XP-3645	S		CR Service

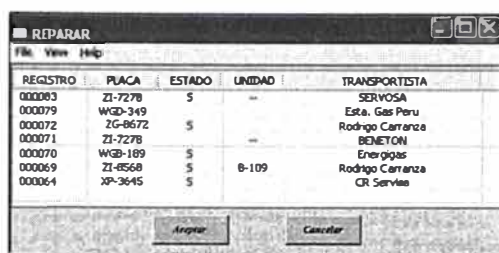
**Figura 3.25** Imagen de la Ventana emergente de "Pendientes"

El sistema reconoce y permite que se pueda grabar solo un peso a la vez es decir si el registro necesita el peso inicial, permite solo esta opción y de la misma forma si se requiere capturar el segundo peso.

- **Reparar pesos:** Por el modo de captura de pesos eventualmente podría haber errores en la captura de pesos, los cuales se pueden dar por cometer un error al momento de escoger el registro y se este pesando otra unidad, también podría darse si el operador captura el peso de la unidad antes de que el chofer baje de la unidad; en estos casos se puede realizar la operación reparar pesos, con la cual se puede borrar el ultimo peso capturado de alguna unidad, con lo cual se puede corregir el error cometido y continuar las operaciones con normalidad. Todo registro reparado es guardado con los datos de peso, fecha en que se capturó el peso, fecha en la que se hizo la reparación y usuario que la realizó, esto se almacena en una tabla de la base de datos llamada REPARACIONES, y posteriormente aparece esta información en el reporte diario de registros, con ello el operador de turno debe justificar porque se realizó dicha operación. Antes de realizar esta acción envía un mensaje emergente y notifica las acciones a realizar.



**Figura 3.26** Aviso de información antes de ejecutar la acción “REPARAR”

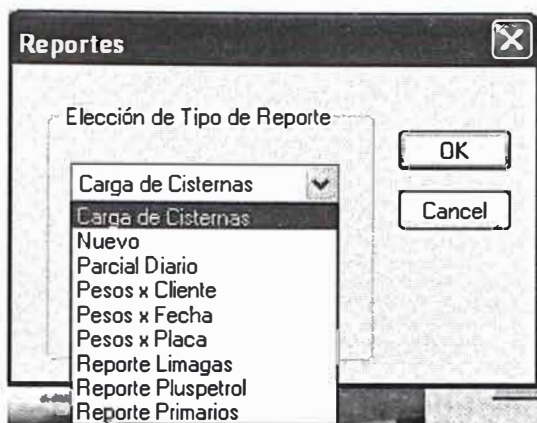


**Figura 3.27** Ventana emergente llamada “REPARAR”

- Registrar: Si bien el sistema es normalmente operado por 3 personas (Sala de despacho, operador de sala de control y operador de caseta), los días domingo, la planta es operada por una sola persona, por ello en la misma pantalla de pesaje se ha habilitado la opción de que se pueda realizar un registro rápido de unidades, para agilizar esta operación, sin embargo por esta modalidad solo se puede registrar unidades ya existentes en la base de datos.

Por último se implementó una pantalla en la que se pueda acceder para agregar, modificar o corregir datos de las principales tablas de base de datos, esto se realizó con el fin de dar acceso al personal de operaciones de actualizar los datos de tablas de clientes, unidades, etc. Provee un acceso a la revisión de la información almacenada, pero protegiendo el no tener acceso a las tablas de documentos.

Durante todas las etapas anteriores se va realizando consultas de lectura y escritura a la base de datos, sin embargo adicionalmente a la operación al final de cada turno, en el caso de los operadores de sala de control, en la contabilización del mes en el caso de los operadores de caseta de despacho, entre otras ocasiones se hace necesario realizar una consulta a modo de administración en la que se reuna la cantidad total de registros para poder reportar o cuantificar los despachos de la planta de almacenamiento, por ello se vió la necesidad de desarrollar los reportes de sistema, se habilitó la opción “GENERAR REPORTE”.



**Figura 3.28** Ventana emergente de los Reportes del sistema

Estos reportes han sido diseñados de acuerdo al formato solicitado por el personal de operaciones, para su ejecución se asignó el siguiente script al botón.

*Sub excel\_Nuevo*

*Dim m(), q() As Variant*

*Dim p(), r() As Variant*

*Dim a As Integer*

*Dim CT\_ID, SUB\_TOTAL As Integer*

*Dim tabla\_A, tabla\_B As String*

*Dim P\_TRACTO, P\_CISTERNA As String*

*Dim variable\_A, variable\_B, variable\_C As String*

*Dim variable\_D, variable\_E As String*

*Dim oExcel As Object*

*Dim abierto As Boolean*

*Dim n As Variant*

*Dim directorio As String*

*fecha = format(now, "Short Date")*

*fecha\_formateada = format(now, "dd/mm/yyyy hh:mm:ss")*

*dia=left(cstr(fecha\_formateada),2)*

*dia\_actual = dia*

*mes=Mid(cstr(fecha\_formateada), 4,2)*

*anho=Mid(cstr(fecha\_formateada),7,4)*

*hra=Mid(cstr(fecha\_formateada),12,2)*

*mnto=Mid(cstr(fecha\_formateada), 15,2)*

*fecha\_formateada\_anterior = format(now, "dd/mm/yyyy hh:mm:ss")*

En esta parte del código se hace llamado a un cuadro de diálogo en el que se escoge las características que tendrá el reporte, luego se consulta que columnas se han escogido para el reporte.

*caracteristicas:*

*Begin Dialog Reportes\_nuevo ,,316,172,"Reportes"*

*GroupBox 16,12,92,52,"Rango de Fechas",.GroupBox1*

*Text 24,24,32,8,"Desde",.Text1*

*Text 24,40,32,8,"Hasta",.Text2*

*TextBox 56,24,44,12,.fecha1,&h0000*

*TextBox 56,40,44,12,.fecha2,&h0000*

*GroupBox 120,12,192,128,"Columnas a Considerar",.GroupBox3*

*CheckBox 128,26,47,8,"Cliente",.chkcliente*

*CheckBox 128,39,56,8,"Transportista",.chktransp*

*CheckBox 128,53,92,8,"Unidad",.chkunidad*

*CheckBox 128,96,92,8,"Nro Documento",.chkdoc*

*CheckBox 128,111,92,8,"Peso",.chkpeso*

*CheckBox 237,39,56,8,"Hora ingreso",.chkingreso*

*CheckBox 237,53,71,8,"Hora Salida",.chksalida*

*PushButton 4,148,66,20,"Imprimir Reporte",.imprimir*

*PushButton 80,148,66,20,"Crear Archivo .XLSX",.archivo*

*CancelButton 260,148,52,20*

*Text 44,54,44,8,"(dd/mm/yyyy)",.Text3*

*End Dialog*

*Dim reportes\_NUEVO As Reportes\_NUEVO*

*reportes\_nuevo.fecha1 = dia & "/" & mes & "/" & anho*

*reportes\_nuevo.fecha2 = dia & "/" & mes & "/" & anho*

*respuesta = Dialog(reportes\_NUEVO)*

*directorio = Environ("SITE\_Root")*

*Begin Dialog nombre\_reporte ,,220,80,"Nombre del reporte",,,1*

*OKButton 173,24,40,14*

*TextBox 12,24,148,40,.name\_report,&h0001*

*Text 12,8,148,14,"Ingrese un nombre para este reporte",.string,,9,ebBold*

*End Dialog*

*Dim nombre\_reporte As nombre\_reporte*

*Dialog nombre\_reporte*

```

    nombre = directorio &"REPORTES\Nuevos\"& nombre_reporte.name_report
campos = "" 'es una cadena que indica que columnas se van a leer
suma = 0
suma1 = 0
If reportes_NUEVO.chkingresso = 1 Then
    If Len(CAMPOS) > 0 Then
        campos = campos & ", peso_ini_fecha"
    Else
        campos = campos & "peso_ini_fecha"
    End If
    suma = suma + 1
    suma1 = suma1 + 1
    ReDim Preserve titulos(1 To suma1)
    titulos(suma1) = "Ingreso"
    ReDim Preserve formatos(1 To suma1)
    formatos(suma1) = "General"
End If
If reportes_NUEVO.chkcliente = 1 Then
    If Len(CAMPOS) > 0 Then
        campos = campos & ", cliente"
    Else
        campos = campos & "cliente"
    End If
    suma = suma + 1
    suma1 = suma1 + 1
    ReDim Preserve titulos(1 To suma1)
    titulos(suma1) = "Cliente"
    ReDim Preserve formatos(1 To suma1)
    formatos(suma1) = "General"
End If
If reportes_NUEVO.chktransp = 1 Then
    If Len(CAMPOS) > 0 Then
        campos = campos & ", codigo_cliente"
    Else
        campos = campos & "codigo_cliente"
    End If

```



```

suma = suma +1
suma1 = suma1 +1
ReDim Preserve titulos(1 To suma1)
titulos(suma1) = "Transportista"
ReDim Preserve formatos(1 To suma1)
formatos(suma1) = "General"

```

End If

If reportes\_NUEVO.chkdoc = 1 Then

```

If Len(CAMPOS) > 0 Then
    campos = campos & ", documento_id"

```

Else

```

    campos = campos & "documento_id"

```

End If

```

suma = suma +1

```

```

suma1 = suma1 + 1

```

```

ReDim Preserve titulos(1 To suma1)

```

```

titulos(suma1) = "Nro Documento"

```

```

ReDim Preserve formatos(1 To suma1)

```

```

formatos(suma1) = "General"

```

End If

If reportes\_NUEVO.chkunidad = 1 Then

```

If Len(CAMPOS) > 0 Then

```

```

    campos = campos & ", unidad"

```

Else

```

    campos = campos & "unidad"

```

End If

```

suma = suma +1

```

```

suma1 = suma1 + 1

```

```

ReDim Preserve titulos(1 To suma1)

```

```

titulos(suma1) = "Unidad"

```

```

ReDim Preserve formatos(1 To suma1)

```

```

formatos(suma1) = "General"

```

End If

If reportes\_NUEVO.chktracto = 1 Then

```

If Len(CAMPOS) > 0 Then

```

```

    campos = campos & ",placa_tracto"

```

*Else*

*campos = campos & "placa\_tracto"*

*End If*

*suma = suma +1*

*suma1 = suma1 + 1*

*ReDim Preserve titulos(1 To suma1)*

*titulos(suma1) = "Tracto"*

*ReDim Preserve formatos(1 To suma1)*

*formatos(suma1) = "General"*

*End If*

*If reportes\_nuevo.chkpeso = 1 Then*

*If Len(CAMPOS) > 0 Then*

*campos = campos & ", peso\_inicial, peso\_final, peso\_final-peso\_inicial"*

*Else*

*campos = campos & ", peso\_inicial, peso\_final,peso\_final-peso\_inicial"*

*End If*

*suma = suma +1*

*suma1 = suma1 + 1*

*ReDim Preserve titulos(1 To suma1)*

*titulos(suma1) = "Peso Inicial"*

*ReDim Preserve formatos(1 To suma1)*

*formatos(suma1) = "0.00"*

*suma = suma +1*

*suma1 = suma1 + 1*

*ReDim Preserve titulos(1 To suma1)*

*titulos(suma1) = "Peso Final"*

*ReDim Preserve formatos(1 To suma1)*

*formatos(suma1) = "0.00"*

*suma = suma +1*

*suma1 = suma1 + 1*

*ReDim Preserve titulos(1 To suma1)*

*titulos(suma1) = "Neto"*

*ReDim Preserve formatos(1 To suma1)*

*formatos(suma1) = "0.00"*

*End If*

*Select Case respuesta*

*Case 1 'imprimir archiv*

*If suma = 0 Then*

*msgbox "Debe elegir al menos una columna para mostrar en el reporte"*

*GoTo características*

*Else*

*If isdate(reportes\_nuevo.fecha1) Then*

*ini\_sql = format(cdate(reportes\_nuevo.fecha1 & " 00:00:00"), "dd/mm/yyyy hh:mm:ss")*

*If isdate(reportes\_nuevo.fecha2) Then*

*fin\_sql = format(cdate(reportes\_nuevo.fecha2 & " 23:59:59"), "dd/mm/yyyy hh:mm:ss")*

*GoTo imprimir*

*Else*

*msgbox "Error en los datos de fecha y hora finales para el reporte"*

*GoTo características*

*End If*

*Else*

*msgbox "Error en los datos de fecha y hora iniciales para el reporte"*

*GoTo características*

*End If*

*End If*

*Case 2 'guardar archivo excel*

*If suma = 0 Then*

*msgbox "Debe elegir al menos una columna para mostrar en el reporte"*

*GoTo características*

*Else*

*If isdate(reportes\_nuevo.fecha1) Then*

*ini\_sql = format(cdate(reportes\_nuevo.fecha1 & " 00:00:00"), "dd/mm/yyyy hh:mm:ss")*

*If isdate(reportes\_nuevo.fecha2) Then*

*fin\_sql = format(cdate(reportes\_nuevo.fecha2 & " 23:59:59"), "dd/mm/yyyy hh:mm:ss")*

*GoTo generararchivo*

*Else*

*msgbox "Error en los datos de fecha y hora finales para el reporte"*

*GoTo características*

*End If*

*Else*

*msgbox "Error en los datos de fecha y hora iniciales para el reporte"*

```

        GoTo características
    End If
End If
Case 0
    Exit Sub
End Select
Imprimir:
    GoTo generararchivo
    sleep 3000
    oexcel.webpagePreview
    oexcel.Sheets(1).Range("a1:aF500").delete
Exit Sub
GenerarArchivo:
cadena1 = "select origen, " & campos & " from documento,clientes "
cadena2 = " where documento.ct_id = clientes.cliente_id And (peso_fin_fecha between ""
& ini_sql & "" and "" & fin_sql & "") order by peso_fin_fecha"
cadena = cadena1 & cadena2
k& = SQLRequest("dsn=REPSOL_BALANZA;uid=sa;pwd=" & clave,cadena,m,,4,True)
longitud= ubound(m)

```

A partir de aquí se listan las instrucciones para crear el archivo excel que tendrá los datos que se han consultado en el reporte.

```

Set oExcel = CreateObject("Excel.Application")
oExcel.Application.Visible = true
oExcel.Workbooks.add
oExcel.Workbooks(1).Sheets(1).Range("A2") = "REPSOL COMERCIAL DEL PERU S.A."
oExcel.Workbooks(1).Sheets(1).Range("A2:D2").merge
oExcel.Workbooks(1).Sheets(1).Range("A2:D2").HorizontalAlignment = 1
oExcel.Workbooks(1).Sheets(1).Range("A1:" & Chr$(65+ suma) &
longitud+200).font.Name = "arial"
oExcel.Workbooks(1).Sheets(1).Range("A1:" & Chr$(65+ suma) & 8).font.bold = true
oExcel.Workbooks(1).Sheets(1).Range("A2") = "Carretera a Ventanilla KM. 24 - Callao"
oExcel.Workbooks(1).Sheets(1).Range("A5") = nombre_reporte.name_report
oExcel.Workbooks(1).Sheets(1).Range("A5:B5").merge
oExcel.Workbooks(1).Sheets(1).Range("A5:B5").HorizontalAlignment = 1

```

```

oExcel.Workbooks(1).Sheets(1).Range("A6") = "Desde el " & format(ini_sql,"dd/mm/yyyy")
& " hasta el " & format(fin_sql,"dd/mm/yyyy")
oExcel.Workbooks(1).Sheets(1).Range(Chr$(64+ suma)&"3") = "Fecha:"
oExcel.Workbooks(1).Sheets(1).Range(Chr$(64+ suma)&"4") = "Hora:"
oExcel.Workbooks(1).Sheets(1).Range(Chr$(65+ suma)&"3") =
left(format(now,"dd/mm/yyyy hh:mm:ss"),10)
oExcel.Workbooks(1).Sheets(1).Range(Chr$(65+ suma)&"4") = Mid(format(now,"yyyy-
mm-dd hh:mm:ss"),12,8)
oExcel.Workbooks(1).Sheets(1).Range(Chr$(65+ suma)&"4").HorizontalAlignment = 2
oExcel.Workbooks(1).Sheets(1).Range(Chr$(65)&"8") = "Origen"
    j=0
    Do While j < suma +1
    If j=0 Then
    oExcel.Workbooks(1).Sheets(1).Range(Chr$(65)&"8") = "Origen"
    Else
    oExcel.Workbooks(1).Sheets(1).Range(Chr$(65+j)&"8") = titulos(j)
    If formatos(j) = "general" Then
    oExcel.Workbooks(1).Sheets(1).Range(Chr$(65+j)&"9:" & Chr$(65)& longitud +
200).numberFormat = formatos(j)
    End If
oExcel.Workbooks(1).Sheets(1).Range(Chr$(65+j)&"8").HorizontalAlignment = 3
    End If
    k=0
    Do While k < longitud
    oExcel.Workbooks(1).Sheets(1).Range(Chr$(65+j)&k+9) = m(k+1,j)
    k=k+1
    Loop
    If j=0 Then
oExcel.Workbooks(1).Sheets(1).Columns(Chr$(65)&":"&Chr$(65)).columnwidth = 25.0
    End If
    j=j+1
    Loop
oExcel.Workbooks(1).Sheets(1).Range("A8:" & Chr$(65+suma) & "8").AutoFilter
oExcel.Workbooks(1).Sheets(1).Rows("9:9").select
oExcel.ActiveWindow.FreezePanes = True
j=1

```

*Do While j < suma +1*

*oExcel.Workbooks(1).Sheets(1).Columns(Chr\$(65+j)&":"&Chr\$(65+j)).EntireColumn.Auto Fit*

*j=j+1*

*Loop*

*oExcel.Workbooks(1).SaveAs filename:= nombre & ".xlsx"*

*msgbox "REPORTE COMPLETADO"*

*End Sub*

### 3.3.4 PUBLICACIÓN DE LA APLICACIÓN EN WEB

La aplicación Cimplicity Plant edition Web Browser, provee 4 niveles de acceso:

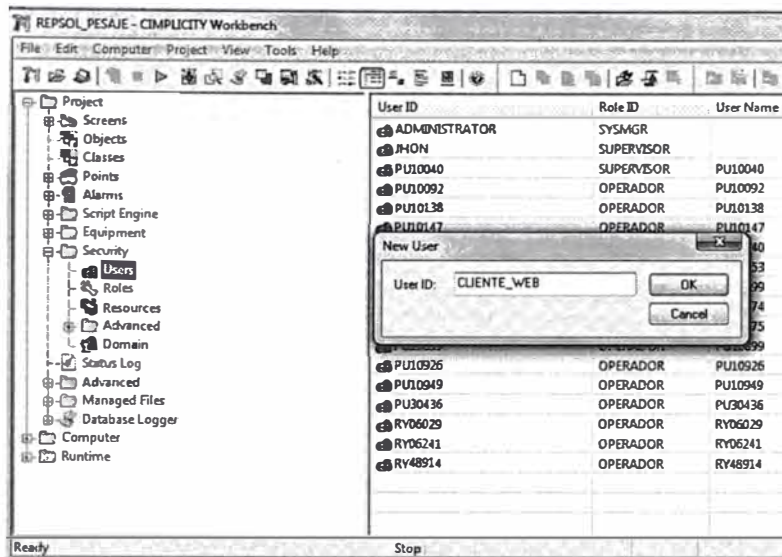
- **Network Project Data:** Habilita a los usuarios en modo solo lectura los datos de todos los proyectos que están siendo administrados por el servidor WEB.
- **Local Project Data:** Permite el acceso a los usuarios a los proyectos que están corriendo sobre el servidor, permite explorar a través del proyecto seleccionado, para ver estado de variables, navegar por las pantallas, etc.
- **WebView Screens:** Permite a los usuarios solamente el acceso a las pantallas y la navegación entre ellas, se accede mediante una dirección URL.
- **Terminal Services:** Permite de acuerdo al nivel de acceso la edición de las aplicaciones a través del Workbench, con todos los accesos que se tengan para el usuario configurado, se requiere una licencia adicional.

Las comunicación entre WebServer y Webviewer es del tipo broadcast y utiliza el protocolo UDP, puerto 32000, es decir cuando el servidor está activo no espera una respuesta del cliente luego de enviada la información.

El acceso WEB está sujeto a problemas de retardo a nivel de transmisión, tanto dentro de las redes locales como son por la codificación, conmutación, retardo por señalización y también por propagación en la red por lo que no se debe dedicar para controles críticos ni que requieran tener respuestas en tiempo real, para nuestro caso solo será del tipo monitoreo y no se permitirá agregar ni modificar los registros existentes.

Para la publicación de la aplicación en WEB y permitir el acceso de la aplicación, primeramente empezaremos por crear un usuario adicional, el cual tendrá un rol al que se limitará los accesos, para permitirle solamente monitorear datos, pero no realizar ninguna operación adicional.



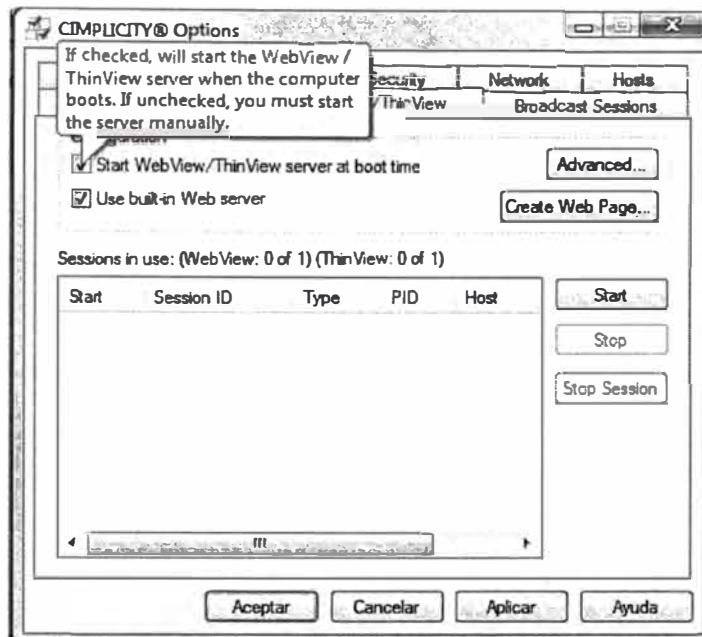


**Figura 3.29** Generación del nuevo usuario “CLIENTE\_WEB” en la aplicación



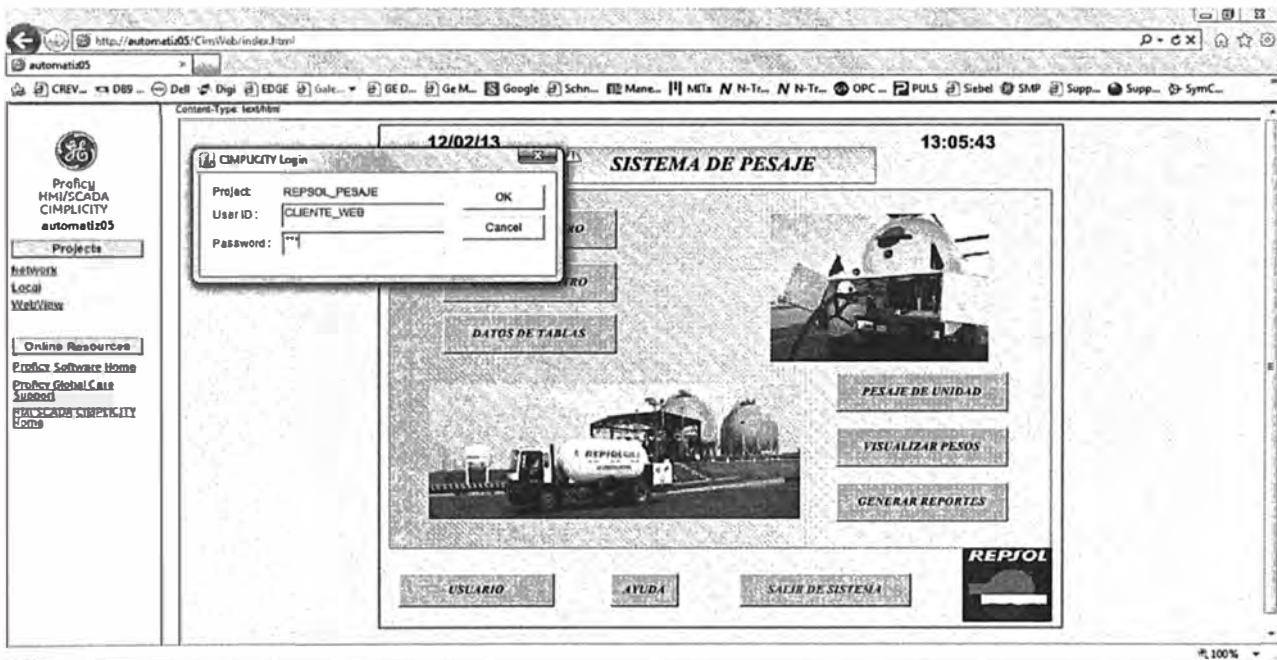
**Figura 3.30** Propiedades del usuario “CLIENTE\_WEB”

Ahora procederemos a configurar el servidor WEB y se modificarán las pantallas para no permitir el acceso a áreas no permitidas a los clientes WEB, para ello nos apoyaremos en el aplicativo del Software SCADA en el que configuraremos las opciones de WEB server. Escogeremos la opción WebView/ThinView y en el habilitaremos el inicio automático del servidor Web, otra forma es habilitarlo manualmente.



**Figura 3.31** Habilitando el inicio automático del servidor Web

Luego para acceder Escogeremos la opción WebView la aplicación se requiere tener instalado JAVA y se accede a ella a través del nombre del Host o la dirección IP del servidor, cuando es dentro de la misma red o a través de una IP pública para acceder vía Internet.



**Figura 3.32** Acceso a la aplicación vía WEB

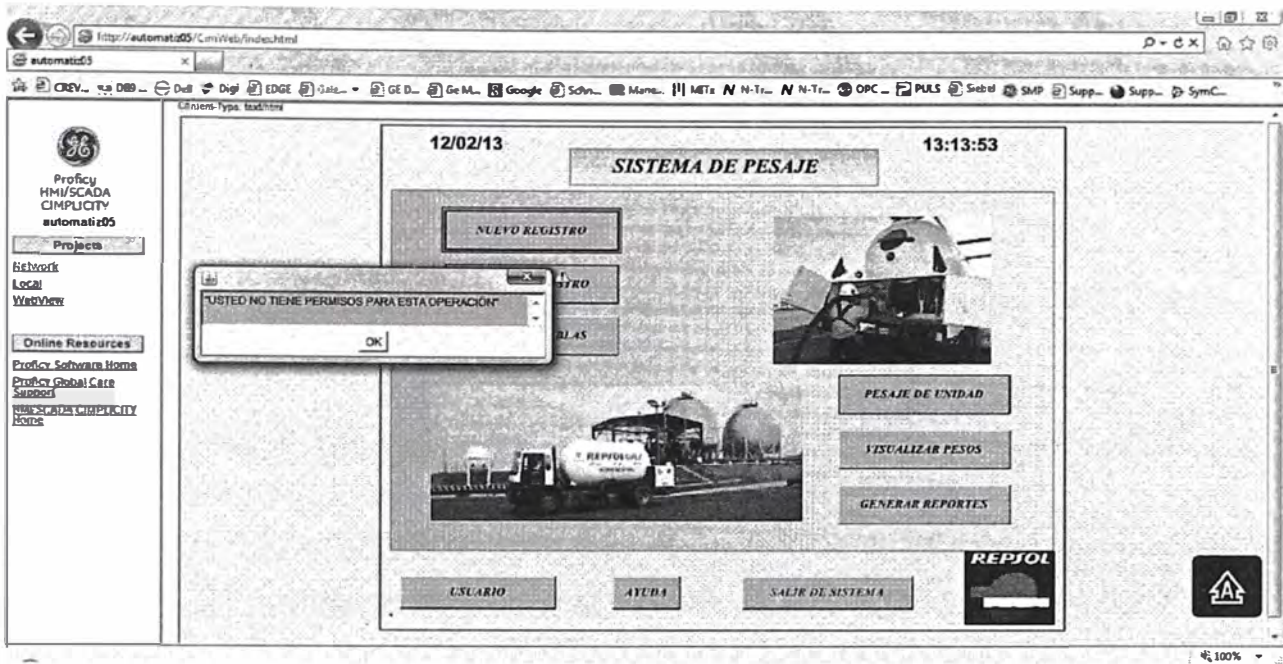


Figura 3.33 Restricción de usuario por WEB

Las propiedades y permisos del usuario Web solo dependen de los permisos otorgados, en este caso solo se le ha dado permisos de monitoreo de los registros.

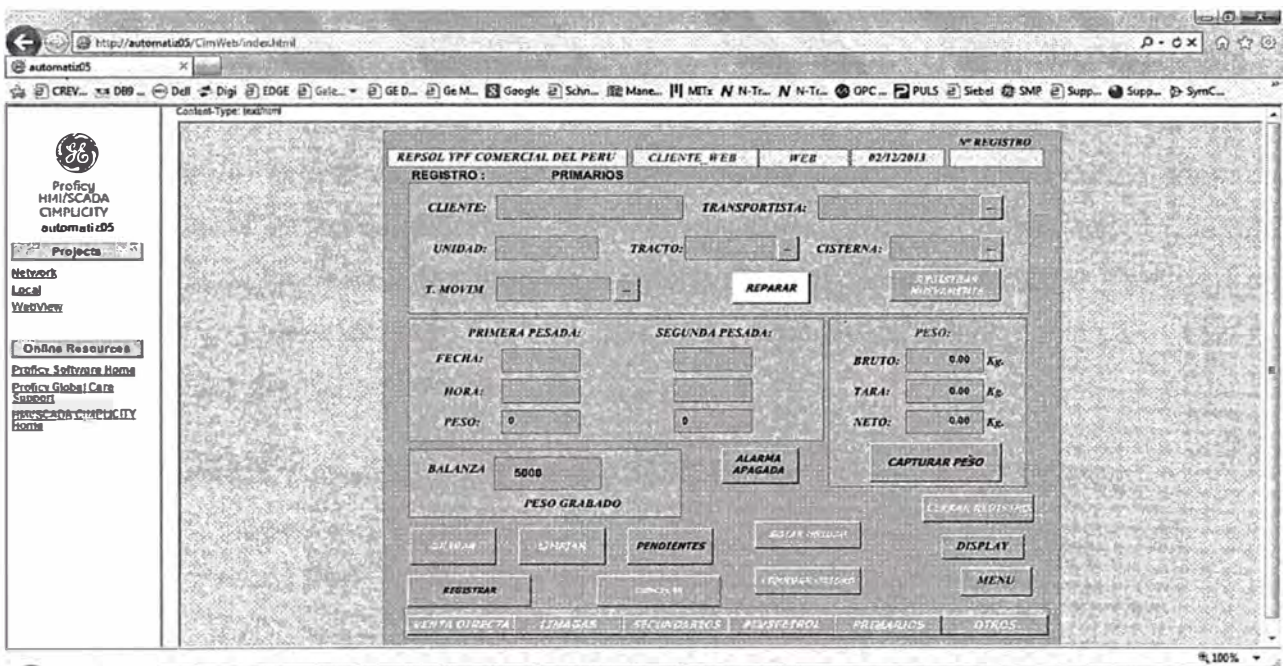


Figura 3.34 Zona permitida de acceso a cliente Web



## CAPITULO IV ANALISIS Y PRESENTACIÓN DE RESULTADOS

### 4.1 Presentación de Resultados

El presente informe mostró el desarrollo de un sistema de SCADA para optimizar las operaciones de despacho de GLP a granel en la planta de almacenamiento PPAL-Repsol, como aportes principales de este desarrollo, podemos describir los siguientes:

- **Precisión de la información:** En el presente desarrollo aporta los valores instantáneos de peso registrados en el cabezal de Balanza, al ser interpretados los datos mediante un controlador, se omite datos no válidos recibidos a causa de ruidos e interferencias, al convertirse los datos a una red ethernet permitió tener los datos disponibles para cualquier punto de la red, dejando de ser un problema la distancia entre la PC del operador y el cabezal de Balanza.
- **Seguridad de la información:** Al estar los datos dentro de una base de datos protegida, no esta al alcance la modificación de los mismos por parte del personal de operación, ni personas externas, permite conservar datos de un periodo de tiempo mayor ocupando menor cantidad de memoria de almacenamiento. En la Tabla 4.1 se muestra una comparación entre la operación antes y despues de implementar el sistema que hemos planteado.

**Tabla 4.1** Comparación antes y despues de implementar el sistema

<b>Antes de la implementación</b>	<b>Después de la implementación</b>
No se puede generar alarmas de variación de peso	Se puede generar alarmas que indiquen la variación de peso
Distancia entre operador y cabezal de Balanza max. 10m	La PC de operador puede estar en cualquier punto de la red LAN
Posibilidad de datos no válidos al momento de la consulta	Los datos son procesados constantemente por el controlador
Almacenamiento de datos en archivos Excel solamente	Almacenamiento de datos en una base de datos

- Optimización del tiempo: Al estar la información organizada y disponible en las tres estaciones de operación a la vez y por las facilidades del registro de datos en la aplicación SCADA, se reduce el tiempo dedicado a las operaciones de registro, pesaje y despacho de las unidades, lo que permite a los operadores dedicar mas tiempo al monitoreo y control de la planta en general, ya que por ser una planta que trabaja con hidrocarburos es crítico el constante monitoreo de la instrumentación de presión y temperatura principalmente de las esferas de carga y las islas de despacho.
- Control de Operaciones: Al estar registrados los despachos, y ser posible la extracción de reportes de diversos tipos es menor el tiempo que toman los operadores de turno en completar el reporte de movimientos del día lo cual se utiliza para controlar la cantidad de producto que ingresa, se despacha y almacena en la planta.
- Facturación y emisión de guías: Al estar integrado el sistema para pesaje y despacho permite realizar la emisión de guías directamente del sistema y no es necesario volver a digitar los valores para la elaboración de guías. La facturación se puede controlar ya que con los reportes “Pesos x cliente” permite saber las cantidades totales despachadas a un mismo cliente y es consecuencia una facturación exacta.
- Disminución de la congestión: Al ser menor el tiempo de las operaciones, disminuye la posibilidad de congestionamiento de unidades en horas punta.
- Aumento de la capacidad de la planta: La implementación de la aplicación permitió ampliar la capacidad de la planta de un promedio de 65 despachos a 95 a mas, aunque normalmente la demanda esta entre 80 y 85 despachos. La figura 4.1 muestra las características integradas que tiene el sistema implementado, con lo que se cumple uno de los objetivos del proyecto.



**Figura 4.1** Integración de Funciones en el sistema implementado.

## Reportes Obtenidos del sistema:

Luego del desarrollo se ha podido exportar a través del sistema los tickets de pesaje, guías y reportes de acuerdo a lo requerido para el tipo de operación llevada en la planta, en la figuras 4.2, 4.3 y 4.4 se muestran un ticket de pesaje, una guía de remisión de Primarios y una guía de remisión Limagas respectivamente, obtenidos por el sistema, con lo que se cumple el objetivo de poder exportar los datos guardados en el sistema y reducir el tiempo de despacho por incluir automáticamente los datos de placas, pesos, fechas, cliente, etc.

A	B	C	D	E	F	G	H	I	J	K	L	M	NO	P	QR
1	REPSOL YPF COMERCIAL DEL PERU S.A.												RUC :	20100176450	
2	Carretera a Ventanilla KM. 24 - Callao												Fecha :	01/12/2013	
3													Hora :	08:39:42 p.m.	
4															
5	=====														
6	TICKET DE PESAJE N° 005488														
7	=====														
8															
9	TANQUE											:	A4S-983		
10	CHOFER											:	JUAN VELASQUEZ		
11	PRODUCTO											:	GAS LICUADO DE PETROLEO		
12	TIPO MOV.											:	SALIDA		
13	O/PEDIDO											:			
14	CLIENTE											:	REPSOL		
15	OPERADOR											:	PU10598		
16	OBSERVACIONES											:			
17															
18													RECIBIDO CONFORME		
19															
20	=====														
21	Primera Pesada				Segunda Pesada				Peso Final						
22	Fecha :	22/07/2012				Fecha :	22/07/2012				Bruto :	48700 Kg.			
23	Hora :	05:39:35 a.m.				Hora :	08:39:02 a.m.				Tara :	23370 Kg.			
24	Peso :	23370 Kg.				Peso :	48700 Kg.				Neto :	23330 Kg.			
25															
26	REPSOL BALANZAS Y GRUPO BSC - PROFICY COMPUTY														
27															

Figura 4.2 Formato de TICKET de pesaje

Carretera Ventanilla Km. 24 Lote 1,2 - Z.1. - Ventanilla - Callao - Lima		20100176450		01/12/2013		Empresa de Transportes Majhoma S.R.L	
Repsol YPF Comercial del Peru S.A		20222419311		R-80299349		Volvo	
CAR. CARRETERA CENTRAL KM. 7.5 JUNIN HUANCAYO SAN AGUSTIN		150815183		YI 8190 / ZI 8611			
6							
<div style="border: 1px solid black; width: 50px; height: 20px; margin: 0 auto;"></div>							
G 03	23430	KGS.	Gas Licuado de Petroleo				
N° Scopy:		N° Transporte		' 01/12/2013 '		20:43	
G.E. 0.531							
Paredes Hugo		23.43					
01/12/2013		1813.25					

Figura 4.3 Guia de Remisión de PRIMARIOS

H21      90

Fecha : 01-dic-13

Producto      GAS LICUADO DE PETROLEO      Hora Salida : 20:44

Guia Remision      LIMA GAS      509-10577      Factura : 1176-0007412

Cliente :      LIMAGAS      Chofer :      JORGE CUBAS

Placa Cisterna : ZG-3919      Placa Tracto : YG-1710

Precintos :      256405 / 256406      Gravedad Esp. =      0.528

	Pesada	P (PSI)	T (° F)	%
Inicial	22,540 Kg.	50	79	0
Final	40,230 Kg.	75	68	90
Neto	17,690 Kg.			

Scop N° : 60280767550

DNI Q10380074      Recibi Conforme

REPSOL\_BALANZA BY GRUPO BSC - PROFICY SIMPLICITY

**Figura 4.4** Guia de remisión LIMAGAS

Reporte Nuevo: Permite generar un reporte seleccionando los campos que se requiere y el rango de fechas, la figura 4.5 muestra las distintas opciones para creación de reportes personalizados que se puede obtener del sistema.

**Reportes**

Rango de Fechas:

Desde:

Hasta:

Columnas a Considerar:

Cliente       Motivo

Transportista       Hora ingreso

Unidad       Hora Salida

Tracto       Sitio de Carga

Cisterna       Gravedad

Nro Documento       Destino

Peso

Scop

Imprimir Reporte      Crear Archivo .XLS      Guardar Configuración      Cancel

**Figura 4.5** Selección de campos de reporte nuevo



La figura 4.6 muestra un reporte personalizado obtenido del sistema con lo que se cumple el objetivo de facilitar la contabilización de las operaciones efectuadas en el tiempo.

Carretera a Ventanilla KM. 24 - Callao

Fecha: 12/0  
Hora: 11:29:13

REPORTE PERSONALIZADO  
Desde el 02/12/2010 hasta el 02/12/2013

Origen	Ingreso	Transportista	Nro Documento	Cisterna	Peso Inicial	Peso Final	Neto	Salida
LIMAGAS	23/02/2010 09:12	ANDRADE	53	ZG-3745	19800	19800	0	23/02/2010
VENTA DIRECTA	08/04/2010 05:55	SERVOSA	66	XO-6132	23900	23900	0	08/04/2010
VENTA DIRECTA	03/07/2010 01:43	SERVOSA	67	ZG-8356	23900	23900	0	03/07/2010
PRIMARIOS	03/08/2010 15:29	Empresa de Transportes Majhorna S.R.L	68	ZQ-2410	23900	23900	0	03/08/2010
PRIMARIOS	23/08/2010 16:47	Rodngo Carranza	69	ZI-8568	23900	23900	0	23/08/2010
PRIMARIOS	25/08/2010 15:12	Rodngo Carranza	72	ZG-8672	23900	23900	0	25/08/2010
PRIMARIOS	25/08/2010 15:34	BENETON	73	ZI-7278	23900	23900	0	25/08/2010
PLUSPETROL	27/08/2010 12:32	CostaGas	75	ZG-8672	23900	23900	0	27/08/2010
PLUSPETROL	01/09/2010 18:09	Flama Gas	76	WGB-520	23900	23900	0	01/09/2010
PLUSPETROL	01/09/2010 18:09	Flama Gas	77	WGB-520	23900	23900	0	01/09/2010

Figura 4.6 Reporte nuevo obtenido

Reporte Pesos x Fecha: contiene todos los movimientos en un rango de fechas, incluido los datos de pesos reparados por los operadores, los formatos Pesos x placa y Pesos x cliente, son de formato similar, con la diferencia de que están filtrados desde el momento de extraer el reporte. La figura 4.7 muestra el reporte básico utilizado por el personal de operaciones de PPAL, pero obtenido del sistema con lo que se facilita la operación de los mismos ya que se obtiene automáticamente y no estar sujeto a errores de tipo.

Pesos x Fechas [Modo de compatibilidad]

REPSOL YPF COMERCIAL DEL PERU S.A.  
Carretera a Ventanilla KM. 24 - Callao

Fecha: 11/12/2009  
Hora: 18.02.20

Reporte de pesos por Fechas  
Desde el 05/12/2009 hasta el 05/12/2009

Fecha	Nro Reg	T oque	Cliente	Ingreso	Hora Ingreso	Salida	Hora Salida	Peso Neto
05 - Dic	33	ZG-3998	PLUSPETROL-LLamaGas	19510	05/12/2009 01:10:33	31020	05/12/2009 01:41:42	11510
	36	ZG-3745	LIMAGAS-ANDRADE	25430	05/12/2009 07:10:56	42770	05/12/2009 07:47:35	17340
	35	ZQ-1612	REPSOL-Peru Gas	24750	05/12/2009 03:47:23	47820	05/12/2009 08:18:51	23070
	37	ZI-8093	REPSOL-Transportes Rodr	24200	05/12/2009 07:19:35	47750	05/12/2009 08:53:05	23550
	38	WGB-247	PLUSPETROL-Energigas	13940	05/12/2009 08:02:34	24450	05/12/2009 08:57:45	10510
	39	ZG-3919	LIMAGAS-ANDRADE	21920	05/12/2009 10:11:56	39650	05/12/2009 10:58:41	17730
	40	ZG-3808	SERVOSA-SERVOSA	19830	05/12/2009 11:31:16	42050	05/12/2009 12:55:19	22220

Figura 4.7 Reporte Pesos x Fecha

## 4.2 Presupuesto de la Implementación

La Tabla 4.2 presenta el cuadro de costos de la implementación del sistema detallado en este informe, en el se incluyen los costos de suministros y servicios necesarios para su desarrollo.

**Tabla 4.2** Resumen de costos de sistema a Implementar

ITEM	CANTIDAD	DESCRIPCIÓN	P. UNIT.	TOTAL
1	1	IC200CPU001 - CPU 34K user configurable user memory, 1.80MSEC/K Boolean one RS232 and one RS485 serial ports ,VersaMax	\$218.00	\$218.00
2	1	IC200PWR102 - power supply with expanded 3.3vdc 120/240vac input ,versamax	\$81.00	\$81.00
3	1	ML30.241 -Power Supply 30 W, 1-1.3 Amp. PULS	\$90.00	\$90.00
4	1	IC200SET001 - Bridge converter RS232/485 to Ethernet	\$400.00	\$400.00
5	1	Accesorios de instalación y montaje	\$30.00	\$30.00
6	1	Servicio de instalación y montaje	\$200.00	\$200.00
7	3	IC647TRT050 - CIMPLICITY HMI Server 50 I/O Runtime System	\$529.00	\$1,587.00
8	1	Servicio de Programación de PLC y SCADA, Incluye: - Programación y configuración de PLC Versamax. - Configuración de Bridge Serial to Ethernet. - Instalación de Software, configuración de Sistema SCADA en 3 estaciones de operación. - Desarrollo de Aplicación SCADA de sistema de balanza, incluye desarrollo de hasta 8 reportes en Excel. - Entrega de manual de Operación y Dossier de calidad.	\$4,500.00	\$4,500.00
			<b>SUBTOTAL</b>	<b>\$7,106.00</b>
			<b>IGV (18%)</b>	<b>\$1,279.08</b>
			<b>TOTAL</b>	<b>\$8,385.08</b>

## CONCLUSIONES Y RECOMENDACIONES

En el presente informe se ha desarrollado la implementación de un sistema SCADA para la administración del despacho de GLP a granel en la planta de almacenamiento PPAL-Repsol, el objetivo principal para el desarrollo ha sido optimizar el tiempo empleado para el despacho, desarrollar un entorno gráfico amigable y funcional que sea fácil de entender y operar, a la vez que garantizar la confiabilidad y seguridad de la información.

En el desarrollo de nuestro sistema SCADA, se han utilizado en gran parte los conocimientos de Electrónica, Comunicación Industrial, Base de Datos, en conjunto con software de programación de controladores PLC, sistemas SCADA basado en Visual Basic y programación para comunicación con Base de datos.

Al mejorar el tiempo para registro, pesaje y despacho, permitió mejorar la capacidad de la planta, con lo cual los despachos son más eficientes y la atención de las unidades dependerá básicamente de la capacidad de llenado/descarga de unidades a partir de sus 6 islas de carga.

Al existir un sistema de control en paralelo el cual se encarga del control de Bombas, válvulas y monitoreo de la instrumentación en general no se pudo integrar los valores de presión y temperatura directamente, es necesario agregarla como dato de ingreso previo a la impresión de guías. La planta al contar con 6 islas de carga y una sola Balanza, ubicada en una zona diferente a las islas de carga, realiza despachos simultáneos, por lo que no se podría realizar un monitoreo constante de la presión mientras se realiza el pesaje.

La implementación de este sistema permitió facilitar el despacho incluso para ser operado por una sola persona como es el caso de los domingos, en el que un solo operador realiza estas funciones, sin embargo normalmente es operado por tres personas.

El sistema Implementado utiliza una versión express de Microsoft SQL server, por lo que la capacidad de almacenamiento está limitada por su licencia, por ello ha sido necesario realizar un mantenimiento automático a la base de datos vía programación en el SCADA, según la cantidad de operaciones se ha visto que la base de datos puede soportar poco más de 2 años seguidos de operaciones, el migrar este sistema a una

versión licenciada de Microsoft SQL server permitiría ampliar la capacidad histórica de almacenamiento y que esta sea limitada solo por la capacidad de disco duro.

Para el buen funcionamiento de nuestro sistema, es necesario que las consolas de operador sean dedicadas al sistema, ya que al operarlo se realizan numerosas consultas a base de datos y el correr al mismo tiempo otros programas que consuman muchos recursos podría volver mas lento el sistema; a la vez que disminuye la posibilidad de corrupción del sistema debido a programas nocivos o Virus.

El sistema de pesaje de Balanza fue implementado con éxito, se cumplió con el objetivo de tener un sistema integrado y que pueda ser operado por 3 estaciones de operación, y de facil exportación de datos, y se tiene en proyecto ampliaciones adicionales como la integración de displays de pesaje, entre otros.

## **ANEXOS**

**ANEXO A**  
**BROCHURE PROFICY CIMPLICITY**

GE Fanuc  
Intelligent Platforms



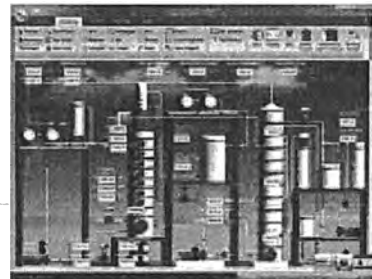
## Proficy® HMI/SCADA – CIMPPLICITY® 8.0

Maximizing the power of your information with the latest features, CIMPPLICITY 8.0 delivers a new user experience and drives results:

- Decreased development costs
- Lower total cost of ownership
- Increased productivity
- Increased usability and ease of use
- Powerful analytics for enhanced decision making

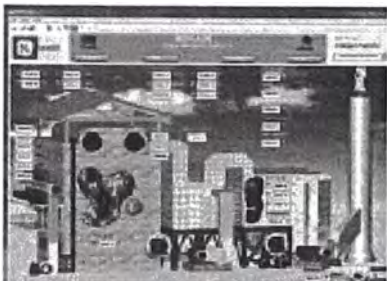
The latest release of Proficy® HMI/SCADA- CIMPPLICITY® is a revolutionary software solution that goes beyond the superior advanced control and visualization you've come to expect from CIMPPLICITY. Version 8.0 is more powerful, more flexible and more feature-rich than ever before. Here's a look at what's new:

**New User Interface.** CIMPPLICITY 8.0 introduces a new way for developers to interact with their HMI/SCADA application. With a new look and feel for the CIMPEdit development environment, based on Microsoft®-fluent User Interface (UI) technology, users will instantly realize time savings and reduced costs. The new UI follows the Ribbon Bar concept, which enables quicker access to galleries and the ability to create custom menus therefore increasing productivity.



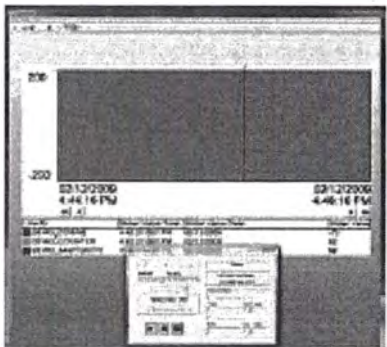


## Proficy HMI/SCADA – CIMPLICITY 8.0



**All new Graphics Engine.** CIMPLICITY 8.0 introduces a new graphics-rendering engine, including larger development canvas functionality for runtime and development, transparency abilities, improved gradient shading, anti-aliasing, native picture support, and a pipe object tool. With a built-in CIMView rubber band zoom tool that enables dynamic zooming from the toolbar, users have enhanced visual accessibility.

**Dynamic Time Handling.** CIMPLICITY 8.0 features a dynamic time handling capability—changing time zones “on the fly” is now native to CIMPLICITY 8.0. With applications being accessed globally, this capability is key to handling multiple global clients and clients progressively moving through time zones.



**Digital Graphical Replay 2.0.** CIMPLICITY 8.0 introduces Digital Graphical Replay (DGR) 2.0, which lets you replay and analyze past events in slow motion, real time or up to 10x the speed—enabling you to identify and troubleshoot issues and prevent repeat occurrences. DGR 2.0 provides the ability to replay trend data as well, a powerful feature for more intelligent decision-making and accelerated issue resolution. With instant access to historical data with little to no development effort, you can easily enhance performance and drive results.

**Enhanced Security Protection.** CIMPLICITY 8.0 only requires a Microsoft Windows® Vista™/Server 2008 standard user account to develop and run your SCADA application. It has been thoroughly tested to ensure full operation while using a standard user account with no User Account Protection (UAP) security prompts—thus reducing the risk of unauthorized file/program access and core system issues.

**Proficy Historian Integration (100 Tags Included).** Forming a powerful combination with GE Fanuc's Proficy Historian and the included Historian license, CIMPLICITY 8.0 enables you to revolutionize your application with true information—not just data—for enhanced decision making.

**Enabling Success Today and Tomorrow.** CIMPLICITY 8.0 offers tight integration with our Proficy software suite for a complete solution and leverages an open system design to protect your current and future investments. Keeping you ahead with the most advanced technologies, CIMPLICITY 8.0 is a powerful and scalable solution that delivers unprecedented insight into your operations—enabling increased productivity and lower total cost of ownership for a sustainable advantage.

CIMPLICITY 8.0 supports Microsoft Vista SP1, Windows Server 2008, Microsoft Windows XP & Windows Server 2003.

### GE Fanuc Intelligent Platforms Information Centers

Headquarters:  
1 800 GEFANUC  
1 900 322 3616  
1 434 978 5100

Global Regional phone numbers  
are available on our web site  
[www.gefanuc.com](http://www.gefanuc.com)

### Additional Resources

For more information, please visit the  
GE Fanuc Intelligent Platforms web site at:

[www.gefanuc.com](http://www.gefanuc.com)



©2009 GE Fanuc Intelligent Platforms, Inc. All Rights Reserved.  
\*Trademark of GE Fanuc Intelligent Platforms, Inc.  
All other brands or names are property of their respective holders.

02/09 GIP-1233

**ANEXO B**  
**DATASHEET DEL CONVERTIDOR IC200SET001**



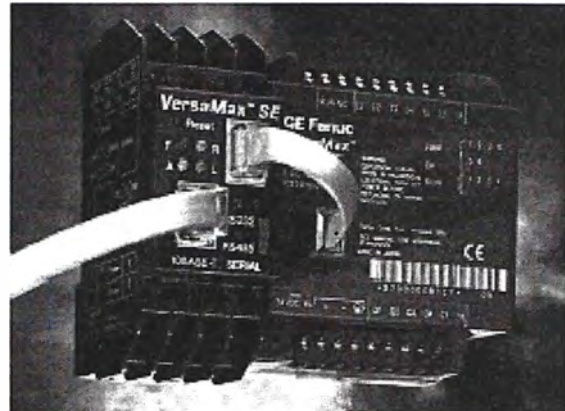
## GE Fanuc Automation

### VersaMax® SE Ethernet Bridge

The new VersaMax SE is a versatile, cost-effective bridge from the serial port on a device to an Ethernet network (10/100Mbps). The VersaMax SE provides the VersaMax Nano and VersaMax Micro PLC easy access to Ethernet for exchanging information, transferring programs, monitoring and controlling.

Now it is possible to connect a wide variety of serial devices, such as PLCs, VFDs, bar code readers, and scales to Ethernet LAN. Thus eliminating the complexity of serial connectivity and protocol mismatches. Serial devices that support Modbus RTU Slave or custom protocols can be easily connected to the VersaMax SE as a bridge to Ethernet LAN. Custom protocols can be created using the software tools provided.

The VersaMax SE is compact, measuring just 3.5 in. (90mm) high x 2.25 in. (60mm) deep x 1.37 in. (36mm) wide. DIN rail mounting provides fast, easy installation and is powered by 0–24VDC and 24VAC.



*VersaMax SE - Bridging the gap between a VersaMax Micro and Nano to a Ethernet LAN*

#### VersaMax SE Product Features:

- Serial Port connection to controller:
  - One RS-232 (RJ45 or screw terminals) or RS-422/RS-485 (screw terminals) connection.
  - Supports SNP, RTU Slave and ASCII serial protocols.
  - RS-422/RS-485 (up to 32 devices multi-dropped. SNP devices not supported)
- 10/100BASE-T/TX connection to Ethernet LAN:
  - One RJ45 connection.
  - 10/100Mbps communication speed.
  - Supports GE Fanuc SRTP, Modbus Ethernet, UDP, TCP and Telnet.
- Rugged construction for industrial applications.
- Compatible with GE Fanuc VersaMax Nano, Micro, VersaMax modular, and Series 90-30 PLCs using the SNP protocol.
- Diagnostic LEDs show:
  - Ready or Connection to host established
  - Network traffic activity
  - Link – good Ethernet link
  - Serial transmit and receive activity
  - Fault or Configuration error
- Reset switch to reset power and initialize
- IP address is stored in Flash Memory

#### VersaMax SE Application Benefits:

- The VersaMax SE is preloaded with the GE Fanuc SNP and SRTP protocols. Data from the plant floor can be accessed by an HMI, such as SIMPLICITY® HMI and other high level devices without costly customization.
- Programming and debugging of GE Fanuc PLCs can easily be done using the VersaPro™ PLC and SIMPLICITY Machine Edition programming software over Ethernet. This reduces downtime and increases efficiency.
- Software tools are provided that allow protocol customization of the VersaMax SE for serial communication to ASCII devices such as bar code readers, scales, robots, variable frequency drives, solid state starters and other controllers over an Ethernet network.
- The industrial design of the VersaMax SE enables it to work almost anywhere on the plant floor.
- VersaMax SE can be set up completely over Ethernet. No need to use serial cables to a PC for configuration.
- Modbus Ethernet connectivity can be achieved by configuring the GE Fanuc PLC in the Modbus RTU mode thus enabling data transfers between the Ethernet LAN and the PLC's.
- Network costs can be greatly reduced by using off-the-shelf hardware and tools.



## Specifications and Ordering Information



Compatibility Matrix for GE Fanuc PLCs		
Model	Protocol	Firmware Version
Series 90 Micro	RTU Slave	3.10 or greater
VersaMax Nano and Micro	SNP and RTU Slave	1.0 or greater
VersaMax modular	SNP and RTU Slave	1.0 or greater
Series 90-30	SNP and RTU Slave	8.2 or greater
Series One, Three, Five and Six	RTU Slave	
LogiMaster Ethernet	SNP	Ver. 9.02 or greater
VersaPro	SNP	Ver. 1.1 or greater
Control	SNP	Ver. 2.3 or greater
CIMPLICITY HMI	SNP or RTU Slave	Ver. 4.01 or Greater

Above have been tested as of 7/18/00

Note: Release 1.0 supports SRTP to SNP, Modbus Ethernet to Modbus Slave or custom protocols only.

<b>Processor and Memory</b>	
Flash System Software Tools	Downloadable from a TCP/IP host (TFTP download) or serial port
IP Address	The VersaMax SE is shipped with a default IP address of 0.0.0.0 which automatically enables the DHCP within the VersaMax SE
Reset Button	Reset button on the front enables user to reset power and re-initialize unit.
<b>Status LED's</b>	
<b>Ethernet Status</b>	
"F" LED	Red on steady - configuration error Red light flashing - configuration mode/ during reset/ or during power cycle Red light off - normal operation
"A" LED	Yellow Flashing during normal operation. Unit seeing transmission packets and broad cast traffic
"R" LED	Green on steady - connection to network host established
"L" LED	Green on steady - link is good to Ethernet
<b>Serial Status</b>	
Transmit LED	Flashing during normal transmissions
Receiving LED	Flashing during normal receptions
<b>Communications</b>	
<b>Ethernet</b>	
Network Interface	Integrated 10/100 Base T/IX port (RJ45 connector)
Communications Speeds	10/100Mbps
Compatibility	Ethernet version 2/IEEE 802.3
Protocols Supported	- SRTP (Default, GE Fanuc) - Modbus TCP/IP - UDP - TCP - Telnet
<b>Serial</b>	
Serial Interface (Only one active)	RJ45 connector - RS-232 Screw terminal - RS-232 or RS-422/485 (Up to 32 drops except SNP)
Communications Speeds	300 to 38.4 Kbps
Serial Selector Switch	Selectable setting for either RS-232 or RS-422/RS-485
Modem Control	RTS and CTS
Serial Line Formats	- Characters: 7 or 8 data bits - Stop bits: 1 or 2 - Parity: odd, even or none
Flow Control	XON/XOFF hardware flow control, and None
Protocols Supported	- SNP/SNPX (Default) - Modbus RTU - ASCII
<b>DC Power Source</b>	
Primary Power Range	9 to 30VDC @ 3 watts power consumption 24VAC @ 3 watts power consumption
<b>Dimensions</b>	
Dimensions	3.5 in. (90mm) High x 2.25 in. (60mm) deep x 1.37 in. (35mm) wide
Mounting	DIN Rail (DIN50 022 rail)
<b>Environmental</b>	
Operating Temperature	0 to +50 Degrees C
Humidity	20% to 90% non-condensing
Agency Approvals	UL, C-UL, FCC, TUV and CE

Ordering Information	
Part Number	Description
IC200SET001	VersaMax SE with IC200CBL504 (4.0 inch RJ45 to RJ45 cable to connect the VersaMax Micro or Nano to VersaMax SE), Manual (GFK-1852) and CD (Configuration software)
IC200TBX210	Ethernet Starter Kit - VersaMax SE (IC200SET001) and VersaMax Nano (IC200NDR001) with VersaPro (IC640VPS002) and CIMPLICITY Machine Edition (IC646MPM101)
IC200TBX214	Ethernet Starter Kit - VersaMax SE (IC200SET001) and VersaMax Micro (IC200UDR001) with VersaPro (IC640VPS002) and CIMPLICITY Machine Edition (IC646MPM101)
IC200TBX223	Ethernet Starter Kit - VersaMax SE (IC200SET001) and VersaMax Micro (IC200UAL006) with VersaPro (IC640VPS002) and CIMPLICITY Machine Edition (IC646MPM101)
IC200TBX228	Ethernet Starter Kit - VersaMax SE (IC200SET001) and VersaMax Micro (IC200UDR005) with VersaPro (IC640VPS002) and CIMPLICITY Machine Edition (IC646MPM101)



## GE Fanuc Automation

GE Fanuc Automation Information Centers

USA and the Americas 1 800 GE FANUC or (434) 876 5100

Europe and Middle East (33) 21 727979-1

Asia Pacific 65-21-3222-4559

GFA-226B

© 2002 GE Fanuc Automation Americas, Inc.  
All Rights Reserved. VersaMax and CIMPLICITY are registered trademarks and Series 90 and VersaPro are trademarks of GE Fanuc Automation Americas, Inc.

### Additional Resources

For additional information on our products and services, please visit our website at:

[www.gefanuc.com](http://www.gefanuc.com)

## BIBLIOGRAFIA

- [1] Lucelly Reyes, "Estandar de comunicaciones RS-232C", Curso de Instrumentación Universidad de Antioquia
- [2] Escuela politecnica Superior UAM, "CODIGO ASCII".
- [3] D.J. Toncich, "Data Communications and Networking for Manufacturing Industries"
- [4] Eduardo Diaz Rosales, "Curso Fundamentos de HMI/SCADA – Introducción al Proficy Ifix".
- [5] Silvia Abrahao, "Curso Manipulacion de bases de datos con SQL".
- [6] Carlos Rodrigo Blanco, "SQL Básico", Primera Escuela de la red temática SVO – Madrid- 2006.
- [7] <http://www.automatas.org/redes/scadas.htm>
- [8] Mettler Toledo, "8530 Digital INDICATOR – Technical Manual"
- [9] GE FANUC Automation, "GFK-1852A Versamax Serial to Ethernet Adapter, User's Manual" – 2003.
- [10] GE FANUC Automation, "GFK-1503C-SP – PLC Versamax, Manual de Usuario", 2001.
- [11] L. A. Bryan / E.A. Bryan, "Programmable Controllers, Theory and Implementation" – 1997.
- [12] <http://bibdigital.epn.edu.ec/bitstream/15000/10020/2/PARTE%202.pdf>
- [13] <http://www.ing.unlp.edu.ar/electrotecnia/procesos/apuntes/ApuntePLC.pdf>
- [14] <http://www.marcombo.com/Descargas/8426714188-scada/cap%c3%8dtulo%20l.pdf>