



CEM | 44



Manual de uso Librería DataLink

VERSIONES Y ACTUALIZACIONES

Manual:	Modo de uso y Funciones DataLink Library Genérico
Revisión:	1.5
Preparó:	Guido Delgado (Departamento Desarrollo)
Aprobó:	Desarrollo Nivel II
Fecha de Creación:	07/10/2014
Última modificación:	02/12/2015

Revisión	Fecha	Descripción de los Cambios	Autor
1.0	07/10/2014	Versión Inicial	G. Delgado
1.1	16/11/2014	Agregado de modo DEMO e Índice	G. Delgado
1.2	19/11/2014	Corrección Documento Comando 3.7 – ObtenerUltimas10ventas	G. Delgado
1.3	06/02/2015	Agregado de Comandos Telemedición	G. Delgado
1.4	09/03/2015	Agregado Referencias de Librería	G. Delgado
1.5	06/05/2015	Agregado Licenciamiento de Librería Agregado Comando PagarVenta Reemplazo de Comando CantCaras por ObtenerConfig	G. Delgado
1.6	02/12/2015	Agregado de Método AcercaDe	G. Delgado

0. Índice

0. Índice	2
1. Introducción	4
2. Información Básica	4
2.1 Primeros Pasos	4
2.2 Glosario y Constantes	5
2.3 Parámetros Comunes	6
3. Funciones y Métodos de Comunicación	8
3.0 Crear Nueva Clase Controlador “NEW” o “Nuevo”	8
3.1 Conectarse al controlador. “ConectarControlador”	8
3.2 Controlador Conectado. “Conectado”	8
3.3 Estado del Sistema. “EstadoSistema”	9
3.4. Realizar Cierre. “CerrarTurno”	9
3.5. Obtener Configuración de la Estacion. “ObtenerConfig”	10
3.6. Estado de las caras. “EstadoCaras”	12
3.7. Ultimas 10 Ventas. “ObtenerUltimas10Ventas”	12
3.8. Obtener Aforadores. “ObtenerAforador”	14
3.9. Reporte de Cierre de Turno. “ObtenerReporteDeCierre”	15
3.9.1 Secciones	19
3.10. Obtener Reporte Pendientes por consultar. “ReportesPendientes”	28
3.11. Cambio de Precio. “CambioDePrecio”	29
3.12. Inventario de Tanques en Cierre. “ObtenerInventarioTLSTurno”	30
3.13. Pago de Venta. “PagarVenta”	31
4. Funciones y Métodos Internos de Librería	32
4.1. Obtener Último Error del sistema “ObtenerUltimoError”	32
4.2. Comprobar si la Licencia es válida “IsLicenseValid”	32
4.3. Setear Direccion Pump Service “SetearValores”	33
4.4. Setear Conexión TLS “TLSValues”	33
4.5. Información de Licencia “AcercaDe”	33
5. Modo DEMO	34
5.1. Descripción	34
5.2. Modo de uso y Respuestas	34
6. Referencia de Librería	43
6.1. Entorno Microsoft.Net	43

6.2. Entornos externos a Framework Microsoft.NET	44
7. Licenciamiento de Librería	49

1. Introducción

El propósito de este documento es proveer una descripción de las funciones y métodos existentes en la Clase Controller, de la Librería DataLinkLibrary.dll. DataLink Library es un producto de la empresa DataOil! S.R.L. homologado por la compañía Gilbarco Veeder-Root. Busque la última actualización de este documento en el siguiente enlace: [Latest Version Manual](#)

2. Información Básica

2.1 Primeros Pasos

Modo de operación básico:

Para dar uso a esta librería de clases es necesario referenciar el archivo DataLinkLibrary.dll a su proyecto, dependiendo del entorno de programación que utilice el modo de referenciarlo cambia. Para más detalles de como agregar esta librería según el lenguaje y entorno de desarrollo utilizado por favor vea el [Punto 6](#).

Existen algunas funciones que demandan Parámetros para su funcionalidad y otras que no son obligatorios. Durante este documento vamos a definir estos últimos como Parámetros Opcionales y aparecen encerrados entre corchetes. [*Parámetro*]

Formato de respuestas:

Existen dos respuestas posibles al llamar una función:

- Si la función procesó la información sin problemas y no existieron errores de ningún tipo, se recibirá la información descripta por cada Función en el [Punto 3](#).
- En caso que ocurra algún error, los valores posibles son los siguientes:
 - E5: Error en Parámetros, formato de parámetro incorrecto.
 - E6: El Controlador se encuentra Desconectada. Es necesario llamar la función [ConectarControlador\(\)](#)
 - E7: Error interno del sistema. En caso de persistir un error con valor E7, por favor comuníquese con el Soporte Técnico de DataOil S.R.L.
 - E8: Error de Licencia. La licencia de la Librería que intenta utilizar no es compatible con su Controlador.

Nota: Estos errores pueden ser recibidos también por la función [ObtenerUltimoError\(\)](#), en caso de que la función ejecutada devuelva únicamente un valor Booleano.

2.2 Glosario y Constantes

Mangueras

Manguera del surtidor	Código Manguera
1	'a' o ' ' o '0'
2	'b' o '1'
3	'c' o '2'
4	'd' o '3'

Estados de Surtidor

Código Estado	Descripción
1	Disponible
2	En Solicitud
3	Despachando
4	Autorizado
8	Error
9	Anulado
10	No Instalado
11	Stop

Tipo de Pago

Código de Pago	Descripción
00	Transacción impaga.
01	Pago contado automático
02	Prueba de dispensario con retorno
03	Prueba de dispensario sin retorno
04	Incobrable
05	Venta sin control
09	Bonificado
10	Pago CDF-Red
11	Pago CDF-Local
16	Pago contado
18	Efectivo en Crind
19	Tarjeta en Crind
32	Facturado pago contado
33	Facturado cuenta corriente
34	Facturado en pos NEXUS
43	Prepago en Platino
44	Venta en Platino
45	Efectivo en Crind Facturado
46	Tarjeta en Crind Facturado
47	Facturado en POS
48 .. 64	Facturado pago Tarjeta
80 .. 96	Pago Tarjeta
238	Pago invalido o inexistente

2.3 Parámetros Comunes

DirPumpSrv

Es la ubicación del servicio Pump Service. Depende también del tipo de controlador al que quiera conectarse. En caso de ser un controlador VOX, es necesario ingresar la dirección IP del mismo. Debe ser un valor del tipo Protocolo de Internet versión 4 (IPv4) válido. El controlador VOX trae de fábrica predefinida la dirección Ip: "192.168.0.100"

En caso de un Controlador CEM, existen dos formas posibles de acceder al servicio Pump Service, ya sea localmente, donde la dirección es un punto "." o desde otra PC, donde la dirección del Pump Service es el Ip de la Pc donde se encuentra instalado. Este último debe ser un valor del tipo Protocolo de Internet versión 4 (IPv4) válido.

IpTLS

Indica la dirección IP del Telemididor Veeder-Root. Debe ser un valor del tipo Protocolo de Internet versión 4 (IPv4) válido.

PortTCPTLS

Es el puerto TCP del Telemididor Veeder-Root, al que la librería debe conectarse. Este valor es un número entero. Por defecto su valor es el preseteado en las consolas de fábrica es PortTCPTLS: "10001".

Tanto la dirección IpTLS como el PortTCPTLS son configurables desde la consola del Telemididor Veeder-Root.

PortCOMTLS

Indica el Puerto Serie de la PC en el que se encuentra conectado el Telemididor Veeder-Root. Es necesario ingresar un valor entero correspondiente al Puerto. Ej: si el Telemididor se encuentra conectado en el Puerto Serie COM1, debe ingresar el valor 1.

Numero de Manguera

El número de manguera, son tres caracteres que representan el número de cara junto con la manguera correspondiente a esa cara. Está formado por dos caracteres que representan el número de cara y un tercer carácter que representa la manguera. "XXM".

Ejemplos: Cara 1 Manguera 1: " 1a" o " 1 " o " 10"

Cara 3 Manguera 4: " 3d" o " 33"

Cara 12 Manguera 2: "12b" o "121"

[Ver Mangueras](#)

Valor Volumen

Este valor representa la cantidad de Litros o Galones despachados. En la siguiente librería podrá encontrar el mismo en distintos reportes y tiene representada la parte decimal en los últimos dos caracteres.

Valor Monto

El siguiente representa los litros vendidos en la moneda que corresponda, según el precio a la hora del despacho. El punto decimal del mismo depende de la configuración establecida en el controlador. Desde la consola CEM44 o la consola Web (únicamente disponible en VOX) existe la posibilidad de setear este valor, en el Menú “Precios” -> “Opción Decimal” Importe de Venta.

En las respuestas DEMO, el valor devuelto tiene la configuración por defecto donde los últimos dos dígitos representan la parte decimal.

Valor PPU

Este valor representa el Precio Por Unidad. El punto decimal, también depende de la configuración establecida en el controlador. Desde la consola CEM44 o la consola Web (únicamente disponible en VOX) existe la posibilidad de setear este valor, en el Menú “Precios” -> “Opción Decimal” Precio Por Unidad.

En las respuestas DEMO, el valor devuelto posee la configuración de la mayoría de los surtidores donde los últimos dos dígitos representan la parte decimal (##,##).

3. Funciones y Métodos de comunicación Controlador

3.0. Crear Nueva Clase Controlador. “New” o “Nuevo”

Descripción: Crea un objeto del tipo Controller.

Parámetros: [DirPumpSrv](#) o “DEMO”

Respuesta: Ninguna.

Nota: Si se desea usar la librería en modo Demo, es necesario setear el parámetro *DirPumpSrv* con el valor string “DEMO”. Una vez seteado ese parámetro la clase va recibir respuestas predefinidas para todas las funciones y no podrá conectarse a un controlador.

[Ver Punto 5.](#)

3.1. Conectarse al controlador. “ConectarControlador”

Descripción: Método de conexión con el controlador. Este método es fundamental realizarlo al menos la primera vez, antes de utilizar cualquier otra Función de la clase.

Parámetros: [DirPumpSrv](#)

Respuesta:

- Verdadero: La conexión con el Controlador se realizó con éxito.
- Falso: Ocurrió algún error en el intento de conexión con el controlador. Para conocer el error llamar la función [ObtenerUltimoError\(\)](#).

3.2. Controlador Conectado. “Conectado”

Descripción: Devuelve el estado de conexión actual con el controlador.

Parámetros: Ninguno.

Respuesta:

- Verdadero: El controlador se encuentra conectado al sistema.
- Falso: El controlador está desconectado del Sistema. Es necesario llamar la función [ConectarControlador\(\)](#) para restablecer la comunicación.

3.3. Estado del Sistema. “EstadoSistema”

Descripción: Devuelve el estado del sistema del controlador de surtidores.

Parámetros: Ninguno.

Respuesta:

Campo	Tipo	Valor
Estado del Sistema	String	1 = Ok 4 = Cierre en progreso 8 = Ventas impagas 12 = Despachos en curso

3.4. Realizar Cierre. “CerrarTurno”

Descripción: Realiza un Cierre de Turno/Día/Mes según se solicite.

Parámetros: *TipoDeCierre*, [*CerrarArchivos*]

TipoDeCierre: Debe ser un Carácter, según el tipo de cierre que desea realizar:

T: Cierre de Turno;

D: Cierre de Día;

M: Cierre de Mes

CerrarArchivos: Valor Booleano que define si el cierre debe ser definitivo o es para consultar totales únicamente. Por defecto posee el valor Verdadero. (El valor recomendado para operatoria normal es el que viene por defecto, verdadero)

Respuesta:

- Verdadero: El comando de Cierre se recibió con éxito y comenzará el proceso de Cierre a continuación.
- Falso: Este caso puede deberse a que no había despachos o datos para mostrar en el cierre. En caso de que haya ocurrido algún error en el intento de cierre, llamar la función [ObtenerUltimoError\(\)](#) para obtener una descripción.

Nota: Cuando el valor devuelto es Verdadero no significa que el cierre se concretó, sino que el proceso se inició. Para comprobar si el Cierre se concretó correctamente utilice los comandos [EstadoSistema\(\)](#) y [ReportesPendientes\(\)](#).

3.5. Obtener Configuración de la Estación. “ObtenerConfig”

Descripción: Devuelve la configuración actual del controlador. Productos, Caras, Mangueras y Tanques.

Parámetros: Ninguno.

Respuesta: La respuesta es recibida como un String con saltos de línea por cada sección.

Campo	Valor
Sección	“CTP” - Cantidad de Caras, Tanques y Productos configurados en el controlador.
Separador	“~”
Cantidad de Caras	Integer (2)
Separador	“~”
Cantidad de Tanques	Integer (2)
Separador	“~”
Cantidad de Productos	Integer (2)
Salto de Línea	Char(Dh) & Char(Ah)

Campo	Valor
Sección	“ST” – Configuración de Surtidores con Tanques.
Separador	“~”
Numero de Cara 1	Integer (2)
Separador	“~”
Numero de Manguera 0	Integer (1) Ver Mangueras
Separador	“~”
Números de Tanques (Cara 1 Mang 0)	String (6) – Dos caracteres por cada tanque. Hasta 3 Tanques.
Separador	“~”
...	Se Repite por la cantidad de caras y mangueras configuradas
Numero de Cara n	Integer (2)
Separador	“~”
Numero de Manguera m	Integer (1) Ver Mangueras
Separador	“~”
Números de Tanques (Cara n Mang m)	String (6) – Dos caracteres por cada tanque. Hasta 3 Tanques.
Salto de Línea	Char(Dh) & Char(Ah)

Campo	Valor
Sección	"SP" – Configuración de cada surtidor con su Producto por manguera.
Separador	"~"
Numero de Cara 1	Integer (2)
Separador	"~"
Números de Productos (Cara 1)	String (8) – Dos caracteres por cada producto, correspondiente a las 4 mangueras. "00" si la manguera no está configurada
Separador	"~"
...	Se repite por la cantidad de caras configuradas
Numero de Cara n	Integer (2)
Separador	"~"
Números de Productos (Cara n)	String (8) – Dos caracteres por cada producto, correspondiente a las 4 mangueras. "00" si no está configurada
Salto de Línea	Char(Dh) & Char(Ah)

Campo	Valor
Sección	"PN" – Nombre de los Productos configurados.
Separador	"~"
Número del Producto 1	Integer (2)
Separador	"~"
Nombre del Producto 1	String (25)
Separador	"~"
...	Se repite por la cantidad de productos configurados
Número del Producto x	Integer (2)
Separador	"~"
Nombre del Producto x	String (25)
Salto de Línea	Char(Dh) & Char(Ah)

Campo	Valor
Sección	"TP" – Configuración de cada Tanque y su Producto.
Separador	"~"
Número del Tanque 1	Integer (2)
Separador	"~"
Número del Producto de Tanque 1	Integer (2)
Separador	"~"
...	Se repite por la cantidad de Tanques configurados
Número del Tanque y	Integer (2)
Separador	"~"
Número del Producto de Tanque y	Integer (2)

3.6. Estado de las caras. "EstadoCaras"

Descripción: Devuelve el estado actual de las 32 caras del controlador.

Parámetros: Ninguno.

Respuesta: Cada uno de estos campos se encuentra separado por el carácter "~"

Campo	Tipo
Estado de la Cara 1	Integer Ver Estados
...	Se repite 32 veces (Cantidad de caras máxima)
Estado de la Cara 32	Integer

3.7. Ultimas 10 Ventas. "ObtenerUltimas10Ventas"

Descripción: Devuelve las ultimas 10 ventas, de una de las caras, correspondientes al Turno actualmente abierto.

Parámetros: *NumCara*, [*VolumenDetalle*]

NumCara: Valor entero, correspondiente al Número de la cara que se desea consultar

VolumenDetalle: Valor Booleano que define si va a recibir o no el campo Volumen Venta Detalle, correspondiente al volumen de la venta definido con 3 dígitos. Por defecto posee el valor Falso, por lo tanto no se recibe ese campo.

Respuesta: Cada uno de estos campos se encuentra separado por el caracter “~”

Campo	Tipo
Id Venta 1	String(8)
Cara de Venta 1	String(2)
Manguera de Venta 1	String(1) Ver Mangueras
Monto de Venta 1	String(8) Ver Monto
Volumen de Venta 1	String(8) Ver Volumen
PPU de Venta 1	String (6) Ver PPU
Producto de Venta 1	String(2)
Tipo de Pago de la Venta 1	String(2) Ver Tipo de Pagos
[Volumen Venta Detalle 1]	String(8) [Opcional]
...	Se repite para las 10 últimas ventas
Id Venta 10	String(8)
Cara de Venta 10	String(2)
Manguera de Venta 10	String(1) Ver Mangueras
Monto de Venta 10	String(8) Ver Monto
Volumen de Venta 10	String(8) Ver Volumen
PPU de Venta 10	String (6) Ver PPU
Producto de Venta 10	String(2)
Tipo de Pago de la Venta 10	String(2) Ver Tipo de Pagos
[Volumen Venta Detalle 10]	String(8) [Opcional]

Nota: Si esta función se ejecuta luego de un cierre de turno, se obtendrán todos los campos en cero. A medida en que se realicen despachos, se completa el String de respuesta, ordenando las ventas de más reciente a más antiguo.

3.8. Obtener Aforadores. “ObtenerAforador”

Descripción: Obtiene los Aforadores electrónicos de una de las caras y puede especificarse de cuantas mangueras, siempre partiendo desde la manguera 1.

Parámetros: NumCara, [CantMangueras]

NumCara: Valor entero, correspondiente a la cara de la cual se desea obtener la información.

[CantManguera]: Valor Entero entre 1 y 4 correspondiente a la cantidad de mangueras que se desea recibir el aforador.

El valor por defecto es 4, por lo tanto devuelve el aforador de las cuatro mangueras, en caso de que no existan estos valores van a volver en 0.

Respuesta: Cada uno de estos campos se encuentra separado por el carácter “~”

Campo	Tipo
Monto Manguera 1	String(8) Ver Monto
...	Se repite según la cantidad de mangueras m que se envió
Monto Manguera m	String(8) Ver Monto
Volumen Manguera 1	String(8) Ver Volumen
...	Se repite según la cantidad de mangueras m que se envió
Volumen Manguera m	String(8) Ver Volumen
PPU Manguera 1	String(6) Ver PPU
...	Se repite según la cantidad de mangueras m que se envió
PPU Manguera m	String(6) Ver PPU

Nota: El aforador es el Totalizar electrónico de litros despachados que posee internamente el cabezal del surtidor. Este valor es independiente del Controlador. Es importante tener en cuenta que en la mayoría de los surtidores electrónicos este valor llega hasta 999,999 (nueve mil novecientos noventa y nueve) Litros y se inicializa nuevamente a 000000.

3.9. Reporte de Cierre de Turno. “ObtenerReporteDeCierre”

Descripción: Devuelve el reporte del Ultimo Cierre de Turno.

Parámetros: TipoDeDato

TipoDeDato: Debe ser un Carácter, según el tipo de dato que desee Recibir:

D: Si desea recibir Detalle del Cierre de Turno.

H: Si desea recibir la Cabecera del Cierre de Turno.

Respuesta: Cada uno de estos campos se encuentra separado por el carácter “~”

- **Respuesta de la Cabecera(‘H’):**

Campo	Tipo
Longitud del comando a recibir	String(8)
Encabezado ReportNr	“ReportNr”
Encabezado ShiftNr	“ShiftNr”
Encabezado DateOpen	“DateOpen”
Encabezado TimeOpen	“TimeOpen”
Encabezado DateClose	“DateClose”
Encabezado TimeClose	“TimeClose”
Encabezado StationNr	“StationNr”
Encabezado DispatchsQty	“DispatchsQty”
Encabezado ReportType	“ReportType”
Encabezado PPUMult	“PPUMult”
Encabezado RazonSocial	“RazonSocial”
Encabezado ExtraData	“ExtraData”
Encabezado Surtidor	“Surtidor”
Encabezado ReporteVer	“ReporteVer”
Encabezado AmountMult	“AmountMult”

Nueva Línea	Salto de Línea
Nro de Reporte (*)	Integer
Nro de Turno (**)	Integer
Día en que se inicio el turno	String(10)
Hora en que se inicio el Turno	String(8)
Día en que se cerró el turno	String(10)
Hora en que se cerró el Turno	String(8)
Nro Estación	Integer
Dispatchs Qty	Integer (Cantidad de despachos en el Turno)
Tipo de Reporte	"T"(Cierre de Turno), "D"(Cierre de Día), "M"(Cierre de Mes)
PPU Mult	Double
Razón Social	String
Extra Data	String
Surtidor	String
Versión del Reporte	String
Amount Mult	Double

Nota: (*) El Numero de Reporte es único en la vida útil de la memoria del controlador, este valor no se repite siempre y cuando no se Formatee la misma o cambie por otra. (**) El Número de Turno corresponde al número de turno en el día. Al enviar un cierre de Día, este valor vuelve a 1.

- **Respuesta del Detalle('D'):**

Campo	Tipo
Longitud del comando a recibir	String(8)
Encabezado Sección	"Sección"
Encabezado Grupo	"Grupo"
Encabezado OrderKey	"Order Key"

Encabezado Columna 1	"Col 1"
Encabezado Columna 2	"Col 2"
Encabezado Columna 3	"Col 3"
Encabezado Columna 4	"Col 4"
Encabezado Columna 5	"Col 5"
Encabezado Columna 6	"Col 6"
Encabezado Columna 7	"Col 7"
Nueva Línea	Salto de Línea
Numero de Sección	Integer
Numero de Grupo	Integer
OrderKey	Integer
Columna 1	String
Columna 2	String
Columna 3	String
Columna 4	String
Columna 5	String
Columna 6	String
Columna 7	String
Nueva Línea	Salto de Línea
...	Se repite por Secciones generadas en el turno

[Volver al Índice](#)

3.9.1 Secciones

Ventas Totales X Tipo de Pagos

Campo	Valor
Sección	"1"
Grupo	"1"=Contado, Tarjeta, Cobrado en POS e Incobrables "2"=Ventas s/ Control "3"=Prueba de Surtidor con y sin retorno
OrderKey	"0"
Col1	Descripción del Tipo de Pago (ej: "Incobrables", " Tarjeta", etc.)
Col2	Vacio
Col3	Vacio
Col4	"0"
Col5	"0"
Col6	Monto. Ver Monto
Col7	Volumen. Ver Volumen

[Volver al Índice](#)

Ventas Totales por Tipo de Producto

Campo	Valor
Sección	"2"
Grupo	Periodo de Precios. Valor="1".."10"
OrderKey	Nivel de Precios. Valor="1".."5"
Col1	Nombre del Producto
Col2	Vacio
Col3	Vacio
Col4	"0"
Col5	PPU. Ver PPU
Col6	Monto. Ver Monto
Col7	Volumen. Ver Volumen

Ventas Totales por Surtidor

Campo	Valor
Sección	"3"
Grupo	"1"
OrderKey	Vacio
Col1	Numero de Manguera Ver
Col2	Nombre del Producto
Col3	Vacio
Col4	"0"
Col5	"0"
Col6	Monto. Ver Monto
Col7	Volumen. Ver Volumen

Ventas Totales por Surtidor (Tipo de Venta)

Campo	Valor
Sección	"4"=Ventas Totales X Surtidor (Contado) "5"= Ventas Totales X Surtidor (Prueba de Surtidor) "6"= Ventas Totales X Surtidor (Prueba sin retorno) "7"= Ventas Totales X Surtidor (Pago Bonificado) "8"=Ventas Totales X Surtidor (Ventas Sin Control)
Grupo	"1"
OrderKey	Vacio
Col1	Numero de Manguera junto con el Nombre del Producto (*)
Col2	Vacio
Col3	Vacio
Col4	"0"
Col5	"0"
Col6	Monto. Ver Monto
Col7	Volumen. Ver Volumen

(*) El numero de manguera está formado por: "xx"(dos caracteres para la cara) y "x"(un caracter para la manguera) [Ver Numero de Manguera](#). Ej.: " 1 GASOLINA 93 SIN PLOMO "

[Volver al Índice](#)

Totalizadores Electrónicos por Surtidor

Campo	Valor
Sección	"9"
Grupo	Formado por el numero de surtidor y la Manguera. Ver Surtidor= 1..32 Manguera= 0..3
OrderKey	Vacio
Col1	Formado por el numero de surtidor y la Manguera. Ver Surtidor= 1..32 Manguera= a,b,c,d
Col2	Nombre del Producto
Col3	String sin importancia
Col4	Totalizador Inicial Monto
Col5	Totalizador Inicial Volumen
Col6	Totalizador Final Monto
Col7	Totalizador Final Volumen

[Volver al Índice](#)

Stock y Vacio por Tanque

Campo	Valor
Sección	"10"
Grupo	"1"
OrderKey	Vacio
Col1	Numero del Tanque
Col2	Nombre del Producto
Col3	Vacio
Col4	Capacidad del Tanque. Últimos 2 dígitos representan a la parte decimal
Col5	Cantidad de Producto dentro del Tanque. Últimos 2 dígitos representan a la parte decimal
Col6	Cantidad de Agua dentro del Tanque. Últimos 2 dígitos representan a la parte decimal
Col7	Cantidad de litros Vacios. Últimos 2 dígitos representan a la parte decimal

[Volver al Índice](#)

Stock y Vacio agrupado por Producto

Campo	Valor
Sección	"12"
Grupo	Numero del Producto
OrderKey	Vacio
Col1	Nombre del Producto
Col2	Vacio
Col3	Vacio
Col4	Capacidad disponible para ese producto. Últimos 2 dígitos representan a la parte decimal
Col5	Cantidad de Producto disponible. Últimos 2 dígitos representan a la parte decimal
Col6	Cantidad de Agua para ese Producto. Últimos 2 dígitos representan a la parte decimal
Col7	Cantidad de litros Vacios para ese Producto. Últimos 2 dígitos representan a la parte decimal

[Volver al Índice](#)

Movimientos de Recepción y Baja por Tanque

Campo	Valor
Sección	"13"
Grupo	"1"
OrderKey	Vacio
Col1	Numero del Tanque
Col2	Nombre del Producto
Col3	"0"
Col4	"0"
Col5	Recepción de combustible. Últimos 2 dígitos representan a la parte decimal
Col6	Baja de Combustible. Últimos 2 dígitos representan a la parte decimal
Col7	Vacio

[Volver al Índice](#)

Movimientos de Recepción y Baja por Producto

Campo	Valor
Sección	"14"
Grupo	Numero del Producto
OrderKey	Vacio
Col1	Nombre del Producto
Col2	Vacio
Col3	"0"
Col4	"0"
Col5	Recepción de combustible de ese Producto. Últimos 2 dígitos representan a la parte decimal
Col6	Baja de ese Producto. Últimos 2 dígitos representan a la parte decimal
Col7	Vacio

[Volver al Índice](#)

Reporte de Tanques

Campo	Valor
Sección	"15"
Grupo	"1"
OrderKey	Vacio
Col1	Numero del Tanque
Col2	Nombre del Producto
Col3	Vacio
Col4	Ventas según Surtidor. Últimos 2 dígitos representan a la parte decimal
Col5	Salidas según Medición. Últimos 2 dígitos representan a la parte decimal
Col6	Diferencia. Últimos 2 dígitos representan a la parte decimal
Col7	Variación por Mil. Últimos 2 dígitos representan a la parte decimal

[Volver al Índice](#)

Reporte de Producto

Campo	Valor
Sección	"16"
Grupo	Numero del Producto
OrderKey	Vacio
Col1	Nombre del Producto
Col2	Vacio
Col3	Vacio
Col4	Ventas según Surtidor de ese Producto. Últimos 2 dígitos representan a la parte decimal
Col5	Salidas según Medición de ese Producto. Últimos 2 dígitos representan a la parte decimal
Col6	Diferencia. Últimos 2 dígitos representan a la parte decimal
Col7	Variación por Mil. Últimos 2 dígitos representan a la parte decimal

[Volver al Índice](#)

3.10. Obtener Reporte Pendientes por consultar. “ReportesPendientes”

Descripción: Informa si existen reportes pendientes de informar y que aún no se han consultado.

Respuesta:

Campo	Tipo	Valor
REPORTE TURNO PENDIENTE	Char(1)	T o ` `
REPORTE DIA PENDIENTE	Char(1)	D o ` `
REPORTE MES PENDIENTE	Char(1)	M o ` `
PRECIOS PENDIENTES A IMPRIMIR	Char(1)	P o ` `
REPORTE PRODUCTO PENDIENTE	Char(1)	Z o ` `
REPORTE CENTRAL PENDIENTE	Char(1)	R o ` `
DINERO EN PLAYA	Char(1)	W o ` `
SMS DE DISPOSITIVO ACTIVO PENDIENTE	Char(1)	S o ` `
TICKET A IMPRIMIR	Char(1)	C o ` `
REPORTE DE RECEPCION EN TANKE PEND.	Char(1)	A o ` `
CAMBIO DE PRECIO TERMINADO	Char(1)	F o ` `
MOSTRAR BANCO DE DEBITO	Char(1)	B o ` `
MOSTRAR NOTIFICACION NLP	Char(1)	N o ` `
MOSTRAR NOTIFICACION BD	Char(1)	E o ` `
REPORTE PARCIAL PENDIENTE	Char(1)	G o ` `
BANDERAS DE AVISO	Char(1)	H o ` `

Nota Importante: Si la consola CEM se encuentra abierta al finalizar algún cierre, por defecto va a consultar el ultimo cierre y este comando va a devolver Vacio ` ` el campo correspondiente al reporte, ya que para el controlador ese reporte fue consultado.

3.11. Cambio de Precio. “CambioDePrecio”

Descripción: Permite establecer un cambio de precio Inmediato o Temporizado.

Parámetros: *FechaYHora*, *Inmediato*, *NroProducto*, *NivelPrecio*, *PPU*

FechaYHora: Valor DateTime, correspondiente a la fecha y hora que se desea realizar el cambio de precio en caso de que el cambio no sea inmediato.

Inmediato: Valor Booleano, indica si el cambio de precio debe ser de forma Inmediata (Verdadero) o si es temporizado (Falso).

NroProducto: Valor entero, entre 1 y 10 (Número máximo de productos configurables). Corresponde al Número de producto que se desea cambiar el precio.

NivelPrecio: Valor entero, entre 1 y 5, correspondiente al Nivel de Precio que se desea cambiar.

PPU: Valor del nuevo Precio Por Unidad del producto. Debe ser un String de 10 caracteres, el precio se setea según la opción decimal configurada en el controlador, se completa con 0 delante y detrás.

Ejemplo: CambioDePrecio(“11/12/2014 17:30”,False,2,1,“0000013230”)

En el ejemplo se realizará un cambio de Precio Temporizado para el día 11 de Diciembre del 2014 a la hora 17:30, en el producto 2, correspondiente al nivel de precios 1 y el Precio por unidad del producto es de 13,23 debido a que la configuración de la opción decimal en el controlador ejemplo esta seteada en “##.##”.

Respuesta:

- Verdadero: El cambio de precio se realizó con éxito.
- Falso: Ocurrió algún error en el intento de cambio de precio. Para conocer el error llamar la función [ObtenerUltimoError\(\)](#).

Nota Importante: El cambio o consulta de configuración de la *Opción Decimal* se puede realizar según el controlador. Desde la consola Web que proporciona el controlador VOX, en el Tab *Precios* -> *Opción Decimal*. Desde la consola CEM44 que se conecta al servicio Pump Service, desde el Tab *Precios* -> *Opción Decimal*.

3.12. Inventario de Tanques en el Cierre. “ObtenerInventarioTLSTurno”

Descripción: Devuelve el Inventario de todos los tanques en el último Turno Cerrado, siempre que la Telemedición haya sido configurada previo al cierre.

Respuesta:

- En caso de ocurrir algún inconveniente:
La Respuesta comenzará con las letras “Err~” y seguido una breve descripción del inconveniente sucedido. Llame la función [ObtenerUltimoError\(\)](#) para más detalles.
- En caso de recepción en proceso se recibirá un String que comienza con las letras “Pro~” y seguido el proceso que se está realizando.
- En caso de recibir correctamente el inventario:

La Respuesta va a estar dada por estos campos, separados por el caracter “~”

Campo	Tipo
Inv	String(3) (“Inv”)
Fecha y Hora	String(16) (“DD-MM-AAAA HH:MM”)
Numero de Tanque 1	“T” + Entero(2) (“T01”)
Delivery en Proceso 1	Integer(1) (1: Verdadero – 0: Falso)
Volumen de Combustible 1	Decimal(cantidad de decimales variable)
Volumen de Agua 1	Decimal(cantidad de decimales variable)
...	Se repite para Todos los Tanques existentes conectados (n)
Numero de Tanque n	“T” + Entero(2) (“Tnn”)
Delivery en Proceso n	Integer(1) (1: Verdadero – 0: Falso)
Volumen de Combustible n	Decimal(cantidad de decimales variable)
Volumen de Agua n	Decimal(cantidad de decimales variable)

Nota 1: Es necesario configurar los parámetros del Telemedidor Veeder-Root con la función [TLSTValues\(\)](#) antes de enviar un Cierre de Turno. Si la función [CerrarTurno\(\)](#) devuelve un valor verdadero, entonces la librería intentará conectarse al Telemedidor y obtener los valores.

Nota 2: En caso que alguno de los tanques no se encuentre conectado al Telemedidor, en el momento del cierre, no se recibirá los datos del mismo.

3.13. Pago de Venta. “PagarVenta”

Descripción: Permite pagar una venta específica a través de su Id de Venta con un Tipo de Pago específico.

Parámetros: *IdVenta*, *TipoDePago*, [*SobreEscribirPago*], [*TanqueRetorno*]

IdVenta: Valor String de hasta 8 caracteres, correspondiente al valor Id de la Venta que desea pagar. Este valor puede ser obtenido de la Función [ObtenerUltimas10Ventas\(\)](#).

TipoDePago: Valor String de dos caracteres, indica la forma de pago de la venta seleccionada. [Ver Tipos de Pago](#)

SobreEscribirPago: Valor Booleano, en el caso que desee cambiar el Tipo de Pago de una venta debe enviarse un valor Verdadero. Por defecto el valor es Falso, se recomienda enviarlo en Falso para no sobre escribir una venta en caso de que haya más de un sistema operando con el controlador, de esta manera solo afectará si la venta está Impaga.

TanqueRetorno: Valor entero, correspondiente al tanque al que se está retornando el combustible, solo valido cuando la forma de pago es Prueba de Dispensario con Retorno “02”.

Respuesta:

- Verdadero: El pago de la venta se realizó con éxito.
- Falso: Ocurrió algún error en el intento de pago de la Venta. Para conocer el error llamar la función [ObtenerUltimoError\(\)](#).

Nota Importante: El pago de una venta puede realizarse mientras el turno en el que se realizó la misma se encuentre abierto, una vez cerrado el mismo es imposible cambiar la forma de pago de una venta.

4. Funciones y Métodos Internos de Librería.

4.1. Obtener Último Error del sistema “ObtenerUltimoError”

Descripción: Permite obtener el ultimo error producido en algún procedimiento.

Parámetros: [*Limpiar*]

Limpiar: Valor Booleano, establece si la variable interna Ultimo Error debe borrarse para no confundirlo con otro error anterior. Por defecto es Falso

Respuesta:

String de tamaño indefinido, con la descripción del último error generado por un procedimiento de la librería. También puede recibir alguno de los errores descritos en el inicio de este documento, [ver errores](#).

Nota: Si decide evaluar una función y obtiene un valor no deseado, antes de ejecutar dicha función llame la función `ObtenerUltimoError(True)`, con el valor `limpiar` en Verdadero, para poder evaluar el error interno.

4.2. Comprobar si la Licencia es válida “IsLicenseValid”

Descripción: Permite saber si la Licencia de la librería coincide con la del Controlador.

Parámetros: Ninguno

Respuesta:

- Verdadero: La licencia coincide con la del Controlador.
- Falso: Este valor puede deberse a que la Licencia efectivamente no coincide con la del Controlador o puede haber ocurrido algún error. Para conocer cuál de estas dos opciones sucedió, llame la función [ObtenerUltimoError\(\)](#).

Nota: En el Modo DEMO, el valor de respuesta va a ser siempre Falso.

4.3. Setear Dirección Pump Service “SetearValores”

Descripción: Permite establecer la dirección del Servicio Pump Service.

Parámetros: [DirPumpSrv](#)

Respuesta:

- Verdadero: La Dirección fue seteada correctamente.
- Falso: Ocurrió un error en el intento de seteo de la Dirección. Para conocer el error llamar la función [ObtenerUltimoError\(\)](#).

4.4. Setear Conexión TLS “TLSValues”

Descripción: Permite establecer la forma de conexión con el Telemetro Veeder-Root.

Parámetros: [IpTLS](#), [\[PortTCPTLS\]](#) o [PortCOMTLS](#)

Respuesta:

- Verdadero: La Ip o Puerto COM fue seteado correctamente.
- Falso: Ocurrió un error en el intento de seteo de conexión. Para conocer el error llamar la función [ObtenerUltimoError\(\)](#).

Nota 1: Esta es una función sobrecargada, por lo tanto admite uno de los dos parámetros para setear el modo de comunicación con el Telemetro.

Nota 2: En el Modo DEMO, el valor de respuesta va a ser siempre Verdadero.

4.5. Información de Librería “AcercaDe”

Descripción: Este método le permite conocer información acerca del archivo de librería que esta ejecutando.

Parámetros: Ninguno.

Respuesta: Cada uno de estos campos se encuentra separado por el carácter “;”

Campo	Tipo
Version: ‘VersionLibrería’	Entre comillas simples indica la versión del archivo
LicenciaLocal: ‘TipoDeLicenciaLcoal’	‘DEMO’ o ‘Licenciado’
LicenciaExterna: ‘TipoDeLicenciaExterna’	‘DEMO’ o ‘Licenciado’
FechaExpiracion: ‘IndicadorDeFecha’	‘NO’ en caso de no tener o la fecha de Expiracion

[Volver al Índice](#)

5. Modo DEMO.

5.1. Descripción

El Modo Demo es una opción en la librería que le permitirá utilizar todas las funciones de la clase y conocer una respuesta tipo de cada una.

Las respuestas obtenidas en el modo demo por cada función van a ser siempre iguales, ya que el único objetivo es obtener un valor de referencia para facilitar el desarrollo de una aplicación que intenta comunicarse con un Controlador y no poseen uno para conectar la librería.

5.2. Modo de uso y Respuestas

A continuación se describirán cuales son las respuestas predefinidas y sus condicionales. En algunas funciones es posible recibir un valor de error como fueron descriptos al principio de este documento. [Ver Errores](#).

En el Modo Demo, se simula una conexión con un controlador, el cual se encuentra conectado a un surtidor doble, es decir dos caras con una manguera cada una.

5.1.0. Inicial.

Indicado en el [Punto 3.0](#), para utilizar el modo Demo, es necesario setear en el parámetro DirPumpSrv de la función New, el valor string "DEMO".

`New Controller("DEMO")`

Una vez creada la clase con ese parámetro, la misma no podrá ser reutilizada con un controlador, por lo tanto todas las respuestas serán en Modo Demo.

5.1.1. ConectarControlador

Esta función devolverá el valor Verdadero en el Modo Demo siempre que la Dirección ingresada sea correcta o no haya ingresado una.

Es necesario llamarla para poder utilizar el resto de las funciones, aun en Modo Demo.

[Ver Punto 3.1](#)

5.1.2. Conectado

La respuesta en el Modo Demo, dependerá si se llama o no a la función ConectarControlador al menos una vez. De ser así el valor será Verdadero.

[Ver Punto 3.2](#)

5.1.3. EstadoSistema

En este modo, la respuesta va a ser siempre el valor “1”, el cual indica que el estado del sistema es OK.

[Ver Punto 3.3](#)

5.1.4. CerrarTurno

Si los parámetros enviados en la función son correctos, se recibirá un valor Verdadero.

[Ver Punto 3.4](#)

5.1.5. ObtenerConfig

El valor obtenido en el modo Demo corresponde al de una estación con la siguiente configuración:

- Sección CTP: 2 Caras, 2 Productos y 2 Tanques instalados.
- Sección ST: La Manguera 0 de la Cara 1 se encuentra conectada al Tanque 1.
La Manguera 0 de la Cara 2 se encuentra conectada al Tanque 2.
- Sección SP: La Cara 1 tiene configurado el Producto 1 en la Manguera 0.
La Cara 2 tiene configurado el Producto 2 en la Manguera 0.
- Sección PN: El nombre del Producto 1 es GASOIL
El nombre del Producto 2 es NAFTA.
- Sección TP: El Tanque 1 aloja el producto 1
El Tanque 2 aloja el producto 2

[Ver Punto 3.5](#)

5.1.6. EstadoCaras

El valor devuelto de esta función será siempre que la Cara1, está en estado Disponible y la Cara2 se encuentra en estado Defectuoso. El resto de las caras se encuentra en estado No Instalado.

[Ver Punto 3.6](#)

5.1.7. ObtenerUltimas10Ventas

Esta función va a devolver la cantidad de ventas que se hicieron durante el turno actualmente abierto, que corresponden 9 Ventas a la Cara1 y 1 Venta de la Cara2.

A continuación se muestra una tabla de la última venta realizada por la Cara1 con los valores correspondientes, la primer carga del turno para esa cara y la 10ma carga que en este caso contiene todos sus campos en ceros ya que en la simulación, se realizaron 9 despachos desde el ultimo cierre de turno.

[Ver Punto 3.7](#)

Campo	Dato
Id Venta 1	225
Cara de Venta 1	1
Manguera de Venta 1	1 o A
Monto de Venta 1	50.04
Volumen de Venta 1	3.85
PPU de Venta 1	13.00
Producto de Venta 1	1
Tipo de Venta de la Venta 1	1
[Volumen Venta Detalle 1]	3.849
...	Se repite para las 10 últimas ventas
Id Venta 9	1
Cara de Venta 9	1
Manguera de Venta 9	1 o A
Monto de Venta 9	0.32
Volumen de Venta 9	0.32
PPU de Venta 9	1.00
Producto de Venta 9	1
Tipo de Venta de la Venta 9	1
[Volumen Venta Detalle 9]	0.322
Id Venta 10	0
Cara de Venta 10	0
Manguera de Venta 10	0
Monto de Venta 10	0
Volumen de Venta 10	0
PPU de Venta 10	0
Producto de Venta 10	0
Tipo de Venta de la Venta 10	0
[Volumen Venta Detalle 10]	0.322

Nota: En el valor Volumen Venta Detalle 10 se recibió un valor que no corresponde a esa venta, este comportamiento va a suceder si una venta tiene todos sus campos en ceros y se solicitó el campo Volumen Venta Detalle, desde el parámetro *VolumenDetalle* seteado en Verdadero.

5.1.8. ObtenerAforador

En este modo, la respuesta va ser el valor del aforador de cada una de las caras junto con su monto y el precio por unidad actual. [Ver Punto 3.8](#)

Cara1:

- Monto: 9202.72
- Volumen: 4560.12
- PPU: 13.00

Cara2:

- Monto: 1807.19
- Volumen: 973.30
- PPU: 15.00

5.1.9. ObtenerReporteDeCierre

En este modo, la respuesta va a ser siempre la misma para los tipos de reporte.

[Ver Punto 3.9](#)

Reporte Cabecera:

Campo	Dato
Nro de Reporte	9
Nro de Turno	1
Día en que se inicio el turno	12/11/2014
Hora en que se inicio el Turno	17:23:14
Día en que se cerró el turno	17/11/2014
Hora en que se cerró el Turno	12:14:42
Nro Estación	1
Dispatchs Qty	10
Tipo de Reporte	"T "
PPU Mult	100.000000
Razón Social	" EESS 00001"
Extra Data	" "
Surtidor	"Surtidor"
Versión del Reporte	"repo2.rpt"
Amount Mult	100.000000

Reporte Detalle:

Cara 1, Manguera 1 o A	Valor
Total Monto Despachado	2202.43
Total Volumen Despachado	169.72
Totalizador Inicial Monto	7401.95
Totalizador Inicial Volumen	4421.31
Totalizador Final Monto	9604.38
Totalizador Final Volumen	4591.02
Monto Contado Despachado	1800.77
Volumen Contado Despachado	138.82
Monto Sin Control Despachado	401.66
Volumen Sin Control Despachado	30.90

Cara 2, Manguera 1 o A	Valor
Total Monto Despachado	601.93
Total Volumen Despachado	40.12
Totalizador Inicial Monto	1407.15
Totalizador Inicial Volumen	946.63
Totalizador Final Monto	2009.08
Totalizador Final Volumen	986.76
Monto Contado Despachado	400.04
Volumen Contado Despachado	26.66
Monto Sin Control Despachado	201.89
Volumen Sin Control Despachado	13.46

Totales por Tipos De Pago	Valor
Total Monto Contado Despachado	2200.81
Total Volumen Contado Despachado	165.48
Total Monto Sin Control Despachado	603.55
Total Volumen Sin Control Despachado	44.36

Totales por Producto	Valor
Monto Gasoil Periodo 1, Nivel 1	0.32
Volumen Gasoil Periodo 1, Nivel 1	0.32
PPU Gasoil Periodo 1, Nivel 1	1.00
Monto Gasoil Periodo 2, Nivel 1	2202.11
Volumen Gasoil Periodo 2, Nivel 1	169.40
PPU Gasoil Periodo 2, Nivel 1	13.00
Monto Nafta Periodo 1, Nivel 1	601.93
Volumen Nafta Periodo 1, Nivel 1	40.12
PPU Nafta Periodo 1, Nivel 1	15.00

Tanques	Valor
Capacidad Tanque 1	20000.00
Cantidad de Producto Tanque 1	0
Cantidad de Agua Tanque 1	0
Capacidad Disponible Tanque 1	20000.00
Capacidad Tanque 2	20000.00
Cantidad de Producto Tanque 2	0
Cantidad de Agua Tanque 2	0
Capacidad Disponible Tanque 2	20000.00

Productos	Valor
Capacidad Total Producto 1	20000.00
Cantidad de Producto Disponible Producto 1	0
Cantidad de Agua Producto 1	0
Cantidad Disponible Producto 1	20000.00
Capacidad Total Producto 2	20000.00
Cantidad de Producto Disponible Producto 2	0
Cantidad de Agua Producto 2	0
Cantidad Disponible Producto 2	20000.00

Ventas Surtidor Vs Tanque	Valor
Ventas Según Surtidor Tanque 1	169.72
Salida Según Medición Tanque 1	169.72
Ventas Según Surtidor Tanque 2	40.12
Salida Según Medición Tanque 2	40.12
Ventas Según Surtidor Producto 1	169.72
Salida Según Medición Producto 1	169.72
Ventas Según Surtidor Producto 2	40.12
Salida Según Medición Producto 2	40.12

[Volver al Índice](#)

5.1.10. ReportesPendientes

En el modo Demo, se recibirá siempre el valor indicando que existe un reporte de Cierre de Turno y un reporte de Cierre de Mes pendiente por consultar.

El string tiene la forma: "T M "

[Ver Punto 3.10](#)

5.1.11. CamioDePrecio

Si los parámetros enviados en la función son correctos, se recibirá siempre un valor Verdadero.

[VerPunto 3.11](#)

5.1.12. ObtenerInventarioTLSTurno

Esta función devuelve el Inventario de dos tanques, en modo DEMO. Para lograr obtener el inventario es necesario setear los valores del Telemedidor (no es necesario que sean físicamente valores reales de un Telemedidor existente), luego enviar un cierre de turno, y recién ahí se obtendrá el valor de inventario llamando esta función. De no respetarse este orden se obtendrá un valor de error que describe el problema.

Los valores de Inventario que recibirá, si se respeta el proceso descrito anteriormente, son:

[Ver Punto 3.12](#)

Campo	Dato
Inv	"Inv"
Fecha	"06-02-2015 11:27"
Numero de Tanque 1	"T01"
Delivery en Proceso 1	1
Volumen de Combustible 1	7542.479
Volumen de Agua 1	106.238
Numero de Tanque 2	"T02"
Delivery en Proceso 2	0
Volumen de Combustible 2	19458.5
Volumen de Agua 2	0

Nota: En este caso la función devuelve que existe un Delivery en Proceso en el Tanque 1, y hay 106.238 Litros de agua en el mismo.

5.1.13. PagarVenta

Si los parámetros enviados en la función son correctos, se recibirá siempre un valor Verdadero.

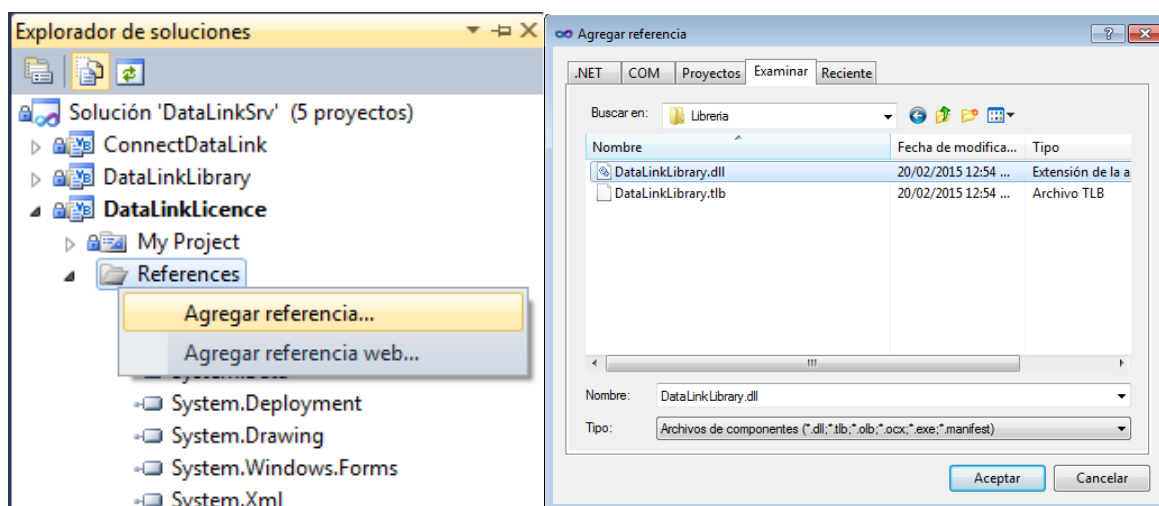
[Ver Punto 3.13](#)

6. Referencia de Librería.

6.1. Entorno Microsoft.NET

En caso que el entorno de trabajo sea .NET, indistintamente del lenguaje, el agregado de la librería al proyecto es relativamente sencillo y no es necesario ninguna registración de la misma. Siempre y cuando la versión del Framework sea 2.0 o superior.

En primer lugar es necesario agregar la librería DataLinkLibrary.dll como Referencia del proyecto. Luego de localizar la librería en una dirección fija seleccionamos aceptar y la misma ya pasa a ser parte de nuestro proyecto



Finalmente es necesario importarla desde el código donde se utilizará la clase interna para poder acceder a todas las funcionalidades de la misma.

Para establecer una comunicación con el controlador es necesario crear una variable en la aplicación a la cual se le asigne una nueva Clase **Controller**, se recomienda utilizar siempre la misma variable global para toda la aplicación y mantener la conexión al controlador abierta para evitar reconexiones de clientes innecesarios.

Código VB.NET

```
Imports DataLinkLibrary
```

```
Public Contoller As New Controller 'o Controller("DEMO")
```

Código C#.NET

```
using DataLinkLibrary;
```

```
public Controller Controlador = new Controller();  
//o Controller("DEMO")
```

6.2. Entornos externos a Framework Microsoft.NET

En caso de trabajar con entornos de este tipo es necesario utilizar una interfaz propia de la librería, para poder acceder a las clases desde la interoperabilidad COM.

6.2.1. Registrado de Librería.

6.2.1.1. Agregado de Librería al GAC (Global Assembly Cache).

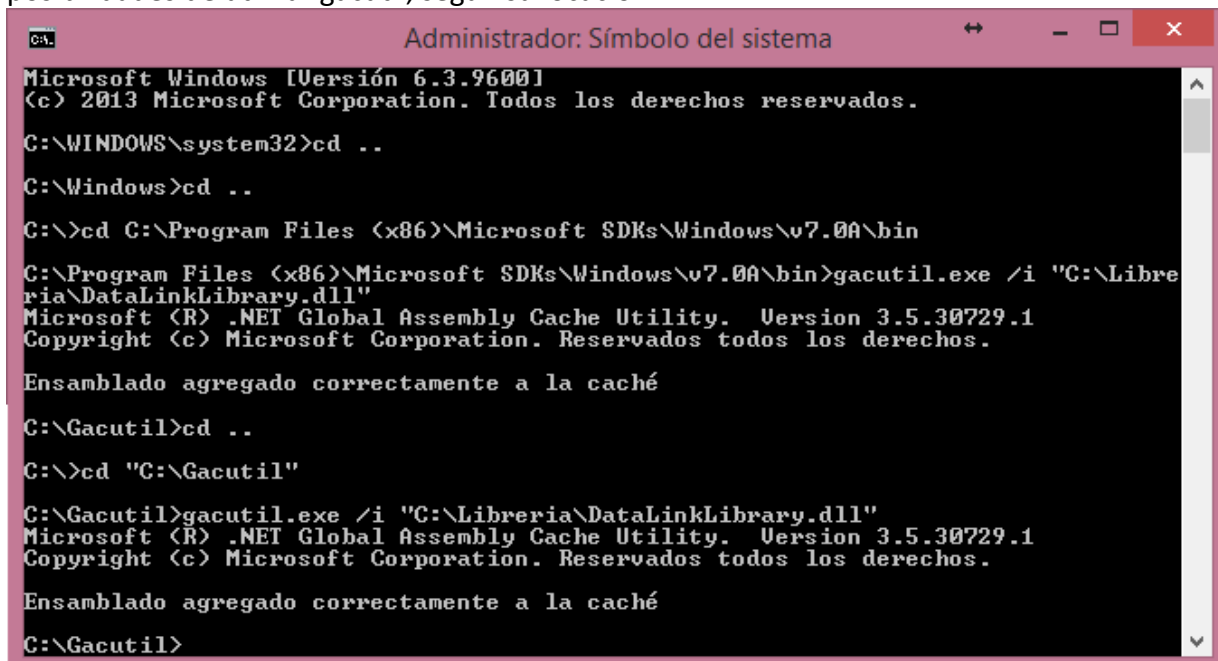
Este proceso es necesario realizarlo únicamente en las pc que se realice el desarrollo, no así en las que tendrán instalado el sistema (Pc cliente).

Existen diversas formas de agregar la librería al GAC, pero la forma más sencilla es utilizando un pequeño aplicativo llamado gacutil.exe.

Para comprobar si la PC de desarrollo posee este aplicativo, por defecto el mismo se aloja en la dirección "C:\Program Files\Microsoft SDKs\Windows\v7.0A\bin\gacutil.exe". La ruta puede variar según las versiones instaladas, y en vez de v7.0A, puede ser v6.0A.

En caso de no tener la carpeta Microsoft SDKs en la PC, entonces será necesario descargar los archivos del siguiente enlace [Archivos Gacutil](#) y dejarlo en su PC desarrollo, para la utilización. Recuerde que es necesario mantener los archivos descargados en una misma carpeta para poder completar el proceso.

Una vez localizado el ejecutable, procedemos a abrir el símbolo del sistema (cmd) y lo ejecutamos como administrador. Desde la línea de comandos nos dirigimos hasta la dirección donde se encuentra el ejecutable gacutil.exe e ingresamos la siguiente línea: `>gacutil.exe /i "DireccionDeLibrería\DataLinkLibrary.dll"`, donde DireccionDeLibrería es la ubicación del archivo DataLinkLibrary.dll. La imagen debajo es un ejemplo de las dos posibilidades de utilizar gacutil, según su locación.



```
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>cd ..
C:\Windows>cd ..
C:\>cd C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\bin
C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\bin>gacutil.exe /i "C:\Libreria\DataLinkLibrary.dll"
Microsoft (R) .NET Global Assembly Cache Utility. Version 3.5.30729.1
Copyright (c) Microsoft Corporation. Reservados todos los derechos.

Ensamblado agregado correctamente a la caché
C:\Gacutil>cd ..
C:\>cd "C:\Gacutil"
C:\Gacutil>gacutil.exe /i "C:\Libreria\DataLinkLibrary.dll"
Microsoft (R) .NET Global Assembly Cache Utility. Version 3.5.30729.1
Copyright (c) Microsoft Corporation. Reservados todos los derechos.

Ensamblado agregado correctamente a la caché
C:\Gacutil>
```

Una vez ejecutada la línea aparecerá debajo una nueva línea indicando que el Ensamblado fue agregado correctamente al caché.

En la imagen superior se realizó dos veces solo a modo de ejemplo.

Este proceso debe realizarse por primera vez para poder utilizar la librería y cada vez que el archivo DataLinkLibrary.dll sea modificado.

6.2.1.2. Registración de DLL en .NET y uso de archivo TLB.

Para poder acceder a todas las cualidades de la dll es necesario que en la pc se encuentre instalado alguna versión de Microsoft .NET, este Framework viene por defecto en la mayoría de las PC con sistema operativo Windows 7 o superior y Algunos Windows XP.

Existen algunas formas para corroborar si en la Pc tenemos instalado alguna versión .NET Framework. Una forma es localizando el directorio:

"C:\Windows\Microsoft.NET\Framework" y otra directamente localizar desde el Panel de Control > Programas > Programas y Características, en el listado de los programas instalados debería aparecer "Microsoft .NET Framework ...". En caso de no tener instalada ninguna versión de .NET Framework puede descargar la versión 2.0 desde el siguiente enlace: [Link .NET Framwork](#). Si el sistema operativo es Windows XP, y no posee el Windows installer, en ese mismo enlace podrá encontrar los instaladores para poder implementar esa versión de .NET Framework en la PC.

Accedemos al símbolo del sistema (cmd), ejecutando el mismo como administrador.

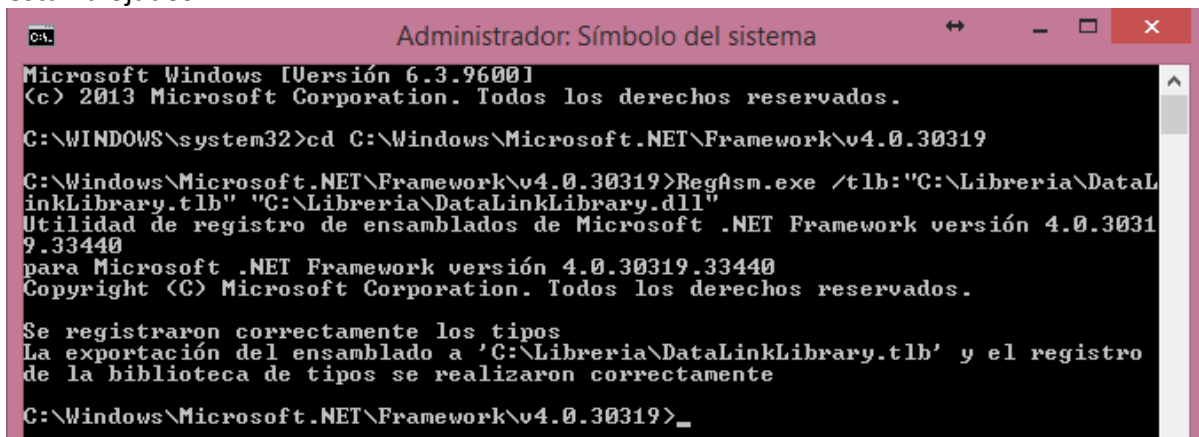
Desde la línea de comandos nos dirigimos al directorio:

>"C:\Windows\Microsoft.NET\Framework\v(Ultima Versión)", es necesario acceder a la carpeta de la última versión instalada para encontrar el ejecutable RegAsm.exe.

Una vez en la carpeta de la última versión ejecutamos la siguiente línea:

```
>RegAsm.exe /tlb:"DireccionArchivos\DataLinkLibrary.tlb"  
"DireccionArchivos\DataLinkLibrary.dll"
```

Se recomienda que ambos archivos se encuentren en la misma dirección, tanto el archivo *.tlb, como el *.dll. En el ejemplo se utiliza *DireccionArchivos* como el directorio donde están alojados.



```
Microsoft Windows [Versión 6.3.9600]  
(c) 2013 Microsoft Corporation. Todos los derechos reservados.  
C:\WINDOWS\system32>cd C:\Windows\Microsoft.NET\Framework\v4.0.30319  
C:\Windows\Microsoft.NET\Framework\v4.0.30319>RegAsm.exe /tlb:"C:\Libreria\DataLinkLibrary.tlb" "C:\Libreria\DataLinkLibrary.dll"  
Utilidad de registro de ensamblados de Microsoft .NET Framework versión 4.0.30319.33440  
para Microsoft .NET Framework versión 4.0.30319.33440  
Copyright (C) Microsoft Corporation. Todos los derechos reservados.  
  
Se registraron correctamente los tipos  
La exportación del ensamblado a 'C:\Libreria\DataLinkLibrary.tlb' y el registro de la biblioteca de tipos se realizaron correctamente  
C:\Windows\Microsoft.NET\Framework\v4.0.30319>_
```

Este proceso es necesario realizarlo tanto en la PC de desarrollo del proyecto como en la Pc donde se instalará el sistema que utiliza la dll, del cliente.

Nota: Si la última versión de .NET Framework que posee instalada es 3.0 o 3.5, el ejecutable *RegAsm.exe* se encuentra en la versión 2.0, ya que las dos anteriores son reversiones de la 2.0 y el ejecutable no se encuentra en esas carpetas.

6.2.1.3. Actualización de Archivo DLL.

En caso de tener que actualizar el archivo dll por una nueva versión, es necesario realizar tres pasos que ya fueron descriptos anteriormente.

1. En primer lugar es necesario anular el registro desde el ejecutable RegAsm.

Para lograrlo es necesario ejecutar la misma línea descripta en el Paso 6.2.2, utilizando los archivos actuales, agregando el apartado */unregister*.

```
>RegAsm.exe /tlb:"DireccionArchivos\DataLinkLibrary.tlb"  
"DireccionArchivos\DataLinkLibrary.dll" /unregister
```

2. Actualizar ambos archivos por la nueva versión (*.dll y *.tlb), y ejecutar nuevamente el ejecutable gacutil, igual que se describe en el Punto 6.2.1.
3. Volver a registrar la dll desde el RegAsm, siguiendo las indicaciones del Punto 6.2.2

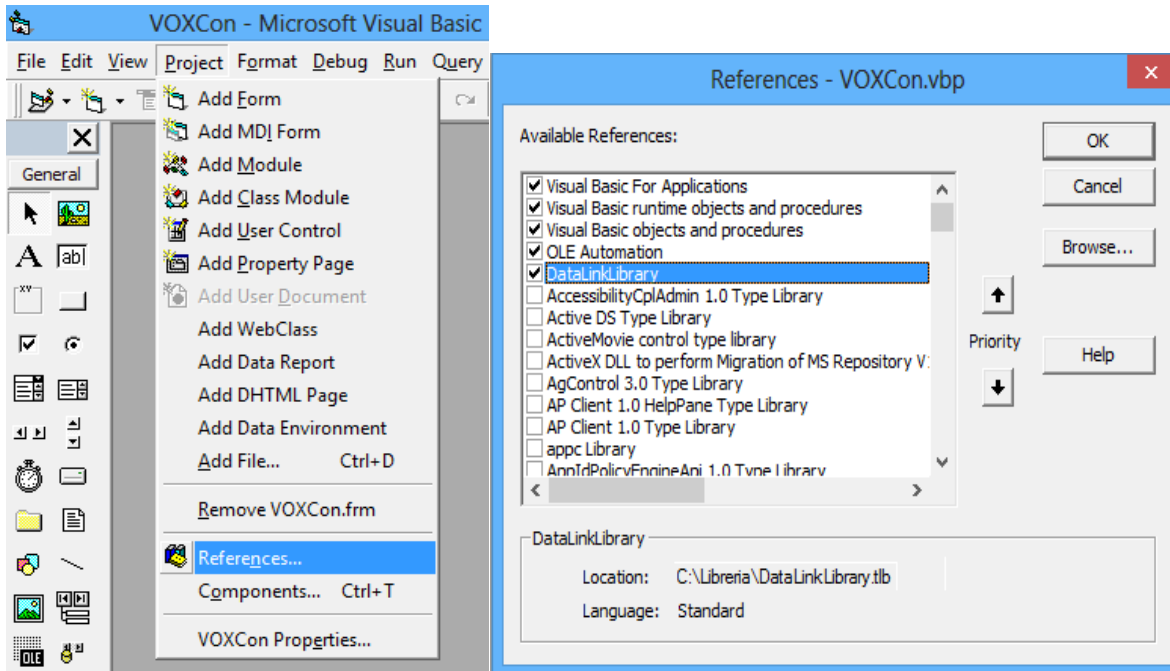
[Volver al Índice](#)

6.2.2. Implementación de librería en entornos externos a .NET.

Antes de realizar cualquiera de las indicaciones debajo, asegúrese de haber cumplido todas las indicaciones descritas en los Puntos 6.2.1.

6.2.2.1. Visual Studio 6.

Para este entorno de trabajo es necesario en primer lugar agregar la librería como referencia a su proyecto.



Si en la registración de la librería no hubo inconvenientes debería aparecer automáticamente entre las referencias disponibles, en caso de que no aparezca puede hacer click sobre el botón Browse... y localizar el archivo DataLinkLibrary.tlb e importarlo. Una vez encontrado el archivo seleccione la casilla correspondiente y apreté el botón Ok.

Si la referencia fue agregada a su proyecto, solo queda importarla desde el código donde se utilizará la clase interna para poder acceder a todas las funcionalidades de la misma.

Para establecer una comunicación con el controlador es necesario crear una variable en la aplicación a la cual se le asigne la Interface [InterfaceControlador](#), y luego setear esa variable asignándole una nueva clase [IController](#). Se recomienda utilizar siempre la misma variable global para toda la aplicación y mantener la conexión al controlador abierta para evitar reconexiones de clientes innecesarios.

Para comenzar a utilizar la variable referenciada a la clase, es necesario llamar a la subrutina [Nuevo\(\)](#).

Código Visual Basic 6

```
Public Controlador As DataLinkLibrary.InterfaceControlador
Set Controlador = New DataLinkLibrary.InterfaceController
Controlador.Nuevo() 'o Controlador.Nuevo("DEMO")
```

6.2.2.2. Delphi.

Para este entorno de trabajo existen varias formas de utilizar una librería de clases. En el siguiente caso vamos a crear un acceso a la misma en forma dinámica, desde el código. Recuerde que como estamos accediendo dinámicamente, puede que las funciones del objeto Controlador no aparezcan listadas, pero con la ayuda de este manual podrán disponer de la información necesaria para una correcta utilización de las funciones.

Código Delphi 7:

```
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, OleServer, OleAuto;
var
  Controlador: Variant;
begin
  Controlador := CreateOLEObject('DataLinkLibrary.IController'); // NameSpace.ClassName
  Controlador.Nuevo(); // o Controlador.Nuevo('DEMO');
end;
```

6.2.2.3. Visual Fox.

En el siguiente entorno de trabajo vamos a crear un acceso a la librería en forma dinámica, desde el código. Recuerde que como estamos accediendo dinámicamente, puede que las funciones del objeto Controlador no aparezcan listadas, pero con la ayuda de este manual podrán disponer de la información necesaria para una correcta utilización de las funciones.

Código Visual FOX Pro 9:

```
Public Controlador

Controlador = CREATEOBJECT("DataLinkLibrary.IController")

Controlador.Nuevo() && o Controlador.Nuevo("DEMO")
```

[Volver al Índice](#)

7. Licenciamiento de Librería.

Para poder utilizar una Librería DataLink y conectarse a un controlador Gilbarco, es necesario poseer la librería licenciada para ese controlador específico.

En el siguiente apartado describiremos como obtener el código de licencia de su controlador, tanto para CEM como para VOX.

Para facilitar la obtención del código de licencia, definiremos que la librería siempre se conecta al servicio Pump Service. En el caso de un controlador VOX, el mismo corre internamente en el equipo, por lo tanto la dirección del Servicio Pump es el Ip del controlador VOX. En cambio para el controlador CEM, el servicio Pump Service, es un servicio de Windows que corre en una PC física. En este caso puede haber dos formas de acceder al servicio:

- Localmente, si el aplicativo que usa la librería corre en la misma PC que se encuentra instalado el servicio Pump Service. Entonces la dirección se accede con un punto “.”.
- Externamente, en este caso el aplicativo que usa la librería se ejecuta en una PC distinta a la que se encuentra instalado el servicio. La dirección es ahora la Ip de la PC donde se encuentra instalado el servicio.

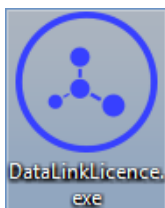
[Ver DirPumpSrv.](#)

Nota: Si el controlador que intenta conectar es un CEM, se recomienda solicitar la librería licenciada facilitando el código de licencia Local y Externo a su proveedor. Recuerde que con una misma librería puede conectarse tanto local como externamente al mismo controlador.

Definimos ahora los pasos a seguir para obtener la/s licencia/s.

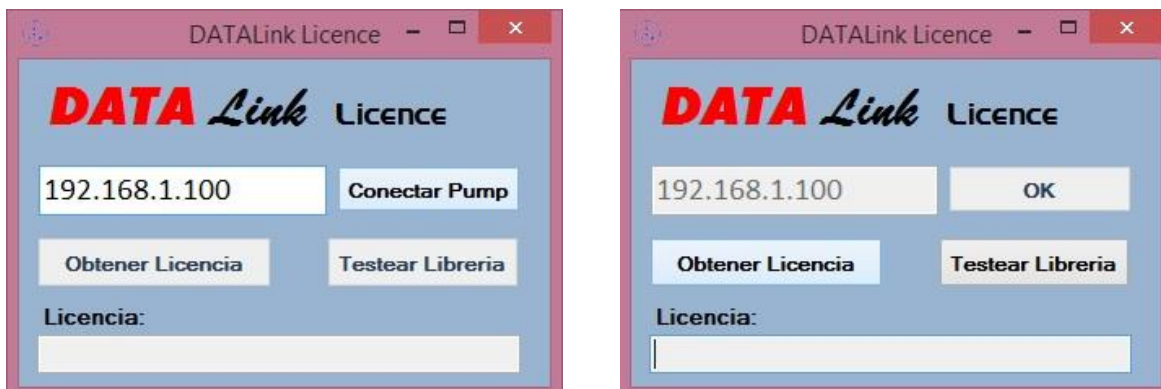
En primer lugar debemos tener en nuestro poder el ejecutable del aplicativo DataLink Licence.

Ejecutamos el archivo DataLinkLicence.exe



Seguido a esto se abrirá la ventana principal del aplicativo. En la misma podremos encontrar una caja de texto con la leyenda *Dirección Pump*, haciendo referencia a la dirección Ip donde corre el servicio Pump Service descripto anteriormente.

Una vez ingresada la dirección, procedemos al intento de conexión con el servicio, presionando el botón Conectar Pump. Si el aplicativo logra conectarse exitosamente el botón recién presionado se tornará gris y cambiará su contenido por la palabra OK. En caso de que algún error ocurra aparecerá un mensaje describiendo el mismo.



Nota: Si está intentando obtener una licencia Local de un servicio Pump Service, recuerde que es necesario ejecutar este aplicativo en la PC que tiene instalado el mismo.

Ya conectado al servicio aparecerán habilitados dos botones debajo. Para obtener la licencia es necesario presionar el botón Obtener Licencia. En el recuadro debajo aparecerá el código de licencia correspondiente a la IP ingresada.

Entregar el código obtenido a su proveedor para recibir la Librería Licenciada.

En el caso que ya posea una librería y quiere corroborar si la misma está o no licenciada para el controlador al que quiere conectarse, coloque el ejecutable en la misma dirección de la librería y conecte el mismo al servicio igual al caso descripto anteriormente.

Una vez conectado presione el botón Testear Librería, en caso de que la librería sea la adecuada el botón se tornará verde y una leyenda aparecerá debajo.

En caso de que la librería no corresponda a la licencia de su controlador el botón se tornará rojo y otra leyenda describirá esto.

