



Manual de programación

API Aruak

Versión 1.7.0 [Publicado]

Bematech International

26/10/2006

Contenido

Contenido	5
Introducción.....	8
Equipos apoyados	9
Descripción.....	10
Control Aruak.....	11
Propiedades.....	11
Coupons	11
Family.....	11
WorkingPath.....	11
SafeInsertion.....	12
Métodos	12
StartServer	12
StopServer	12
AddTerminal.....	13
RemoveTerminal	13
GetLoggedCoupons	13
ClearLogs	14
AddProductEntry	15
AddAssemblyItemEntry	18
UpdateProductEntry	19
AddClientEntry.....	22
AddOperatorEntry	24
AddSalesmanEntry	26
AddDepartmentEntry	27
AddDepartmentGroupEntry	28
AddMessageEntry	29
AddTaxEntry	30
AddTenderingTypeEntry	31
InitializeUpdate	31
InsertTable	32
UpdateTable	33
VoidInsertTable	34
ReadTable.....	34
VoidReadTable.....	35
VoidTransaction	36
WaitForOperation.....	36
GetOperationStatus.....	36
DoGenericOperation.....	37
Eventos	38
OnSalesArrive	38
OnTerminalUpdate	38
OnTableReceived	39
OnOperationComplete.....	39
OnTransactionComplete.....	40
Interfaz ICoupons	40

API Aruak

Propiedades.....	40
Count	40
Item.....	40
Interfaz ICoupon.....	41
Propiedades.....	41
ClientId	41
OperatorId.....	41
SalesmanId	41
TerminalId	41
Date.....	41
Time.....	41
Discount.....	41
Increase	42
SubTotal	42
Iva	42
Number	42
Status	42
TicketType.....	42
Items	42
Payments.....	42
Interfaz IRegisteredItems	43
Propiedades.....	43
Count	43
Item.....	43
Interfaz IRegisteredItem	43
Propiedades.....	43
Code.....	43
Description	43
Tax	44
TaxType.....	44
Department	44
Quantity	44
Status	44
Discount.....	44
Increase.....	44
UnitValue	44
TotalValue	44
Interfaz IPayments	45
Propiedades.....	45
Count	45
Item.....	45
Interfaz IPayment	45
Propiedades.....	45
Index	45
Amount	45
Ejemplos de Programación.....	46
Visual Basic.....	46

API Aruak

Anexo I	49
Archivos do BemaPos	49
Archivos del SB3010.....	52

Introducción

El API (*Application Programming Interfaz*) Aruak simplifica la integración y la interoperabilidad de la aplicación de *backoffice* - para Microsoft Windows - con las CPUs Bematech ejecutando el software de punto de venta.

Ella tiene funcionalidades de registro y lectura de los datos (tablas) permitiendo la gerencia y operación *online* de diversas CPUs conectadas en una red. Los datos són presentados de forma transparente y independiente del modelo y tipo de la CPU.

El API Aruak es distribuida en la forma de un control ActiveX invisible en tiempo de ejecución (*runtime*). Este componente de software puede ser fácilmente integrado en una variedad de plataformas de desarrollo. Abajo, sigue algunas de las plataformas apoyadas por este componente:

- Visual Basic
- Visual C++
- Access
- Excel
- Visual InterDev
- PowerBuilder
- Delphi
- C++ Builder
- .NET
- Visual FoxPro

Equipos apoyados

CPU	MODELO	VERSIÓN
Microterminal	SmartBox 3010	Versión V5.0 o superior. Versiones inferiores solamente registro e lecturas de tablas – log de venta no es soportado.
Microterminal	SmartBox 2030 (BemaPos)	Version 1.0.2 o superior

Descripción

El API se estructura de tal manera para ofrecer un modelo único de objetos a la presentación de los datos contenidos en el “log de ventas”, independiente del modelo del terminal.

La exposición de estos objetos es hecha por el control *ActiveX* que se incorpora en el ambiente del desarrollo, a través de sus métodos y sus propiedades. **Eventos** son disparados por el control cuando ha llegado el log de ventas y también para acusar el resultado de las otras funcionalidades existentes en el control, como las operaciones de lectura/actualización/inserción de las tablas en el banco de datos del PC x Microterminal.

Los siguientes tipos son utilizados y/o definidos por la API.

Tipo	Descripción
BOOL	Un entero con valores TRUE (non-zero) y FALSE (zero).
BSTR	String del sistema.
BSTR*	Un apuntador a una string de sistema.
SHORT	Un entero con 16 bits.
LONG	Un entero con 32 bits.
LONG*	Un apuntador a un entero.
CURRENCY ou CY	Un valor monetario. Es un entero con 64 bits. Tiene 4 casas decimales. Por ejemplo, si el entero es “1234567”, entonces su valor es “123.4567”.
CURRENCY*	Un apuntador a un CURRENCY.
ICoupons	Un tipo nuevo que representa una interfaz a una colección de cupones de venta.
ICoupon	Un tipo nuevo que representa una interfaz a un cupón de venta.
IRegisteredItems	Un tipo nuevo que representa una interfaz a una colección de artículos vendidos en el cupón de venta.
IRegisteredItem	Un tipo nuevo que representa una interfaz para un artículo registrado en el cupón
IPayments	Un tipo nuevo que representa una interfaz para una colección de pagos efectuados en el cupón.
IPayment	Un tipo nuevo que representa una interfaz para un pago efectuado en el cupón

Control Aruak

El control expone sus funcionalidades a través de propiedades, métodos y eventos que estan descritos abajo.

Propiedades

Coupons

Tipo Lectura ;

Sintaxis ICoupons Coupons (LONG terminalId);

Comentario Esta propiedad indexada retorna una colección de los cupones que representa el log de ventas corriente para el terminal especificado. Después de la Lectura de estos datos la información es reducida a cero, y solo será nuevamente creado cuando ocurra el recibimiento de una nueva operación de venta por la terminal. Para recuperar cupones después de esta operación, utilizar el método *GetLoggedCoupons*.

Esta propiedad debe ser llamada de dentro del *handler* para el evento *OnSalesArrive*.

Vea también Evento *OnSalesArrive* y método *GetLoggedCoupons*.

Family

Tipo Lectura/Escritura;

Sintaxis SHORT Family;

Comentario Esta propiedad configura el modelo/tipo del terminal que será gerenciado por el componente.

Valor	Descripción
1	Microterminal SB2000
2	Microterminal SB3010
3	Microterminal Jakarta
4	SB2030 - BemaPos

WorkingPath

Tipo Lectura/Escritura;

Sintaxis BSTR Family;

Comentario String identificando el path donde serán salvos y creados los archivos de trabajo del componente. Directório de sistema (%systemdir%) es el default.

SafeInsertion

Tipo Lectura/Escritura;

Sintaxis BOOL SafeInsertion;

Comentario Cuando TRUE, la inserción de tablas son hechas de forma que los datos sensibles a las mismas (contadores de venta bruta y líquida de la tabla producto del SB3010) son preservados durante la inserción de una nueva carga de datos. Como consecuencia la actualización es más lenta pues es necesario el download de la tabla específica juntamente con la actualización de los contadores en los datos finales.

Operación afectada	Dispositivo
Upload de la tabla de productos	Microterminal SB3010

Métodos

StartServer

Sintaxis BOOL StartServer (void);

Comentario Empieza el gerenciamiento de la red de terminales. Debe ser llamado después de la configuración del modelo del dispositivo, a través de la propiedad *Family*. Los terminales que serán monitorados deben ser autorizados por el método *AddTerminal*.

Retorno

Valor	Descripción
TRUE	Exito
FALSE	Error

Vea también Propiedad *Family* y método *AddTerminal*.

StopServer

Sintaxis BOOL StopServer (void);

Comentario Interrompe el gerenciamiento de la red.

Retorno

Valor	Descripción
TRUE	Exito
FALSE	Error

Vea también Método *StartServer*.

AddTerminal

Sintaxis **BOOL AddTerminal (LONG terminalId, BSTR address);**
Comentario Adiciona un terminal específico a lista de dispositivos monitorados pelo servidor.

Parámetros	Descripción
<i>terminalId</i>	Identificador para el terminal.
<i>address</i>	Específico para el tipo de red. En el caso red Ethernet (SB3010 o SB2030E-BemaPos) debe identificar la dirección IP del dispositivo. Para el SB2030-BemaPos sin Ethernet debe identificar el puerto serial en lo cual la terminal esté conectado (“COM1”, “COM2”,..).

Retorno

Valor	Descripción
TRUE	Exito en la inserción.
FALSE	Error (id o address incorrecto – ip inválido)

Vea también Método *RemoveTerminal*.

RemoveTerminal

Sintaxis **BOOL RemoveTerminal (LONG terminalId);**
Comentario Remove un terminal, o todos, de la lista de dispositivos monitorados pelo servidor.

Parámetros	Descripción
<i>terminalId</i>	Identificador para el terminal. Caso sea pasado -1, todos los terminales son removidos de la lista.

Retorno

Valor	Descripción
TRUE	Exito en la remoción.
FALSE	Error (terminal inexistente)

Vea también Método *AddTerminal*.

GetLoggedCoupons

Sintaxis **ICoupons* GetLoggedCoupons (LONG terminalId, LONG initialCoupon, LONG finalCoupon);**

Comentario Retorna una colección de cupones a través del archivo de log de ventas salvo en el directorio previamente especificado por la propiedad *WorkingFolder*. Ese archivo es constantemente concatenado con las informaciones del log recibidas en cada una de las ventas. Se hace entonces necesario limpiarlo periódicamente con la función *ClearLogs*.

Parámetros	Descripción
<i>terminalId</i>	Identificador para la terminal que se desea recuperar el log
<i>initialCoupon</i>	Número del cupón inicial para efectuar una búsqueda en el log. Debe ser mayor que cero.
<i>finalCoupon</i>	Número del cupón final para efectuar una búsqueda en el log. Caso sea pasado (-1) serán recuperados todos los cupones con números mayores que <i>initialCoupon</i> .

Retorno Una colección de cupones recuperados del log. Podendo ser una colección vacía caso no exista los registros pedidos.

Vea también Método *ClearLogs*.

ClearLogs

Sintaxis LONG ClearLogs (LONG terminalId, LONG initialCoupon, LONG finalCoupon);

Comentario Remove la faja especificada de cupones del log faja de terminal de la lista de dispositivos monitorados por el servidor.

Parámetros	Descripción
<i>terminalId</i>	Identificador para la terminal.
<i>initialCoupon</i>	Número del cupón inicial para efectuar una búsqueda en el log
<i>finalCoupon</i>	Número del cupón final para efectuar una búsqueda en el log. Caso sea pasado (-1) serán removidos todos los cupones con números mayores que <i>initialCoupon</i> .

Retorno

Valor	Descripción
1	Registros removidos con éxito
-1	Registro no encontrado
-2	Registro corrompido
0	Servidor no inicializado

Vea también Método *GetLoggedCoupons*.

AddProductEntry

Sintaxis LONG AddProductEntry (BSTR code, BSTR description, CY price, SHORT taxIndex, SHORT department, LONG StockAmount, SHORT flags, SHORT keyGroup, CY costPrice, LONG margin, SHORT groupDepartment, LONG comission, SHORT remoteNumber, LONG minimumStock, LONG idealStock).

Comentario Adiciona un registro a la tabla de productos para posterior actualización del dispositivo con el método *InsertTable*. Debe ser llamada para cada producto que se desee insertar en el dispositivo. Es necesario llamar el método *InitializeUpdate* antes de la primera llamada a ese método.

IMPORTANTE: Para el **SB3010** es necesario y obligatorio que los registros sean inseridos en orden creciente de código de producto (PLU – código de barras). La propia función les ordenan en un archivo temporario. Ese proceso puede ser más despacio que la ordenación hecha por el banco de datos a través de un “SELECT .. ORDER BY”. Así, para deshabilitar la ordenación se debe utilizar la función *DoGenericOperation* pasandose el código 3. En ese caso, es de responsabilidad de la aplicación ejecutar correctamente la ordenación de los registros previamente, en caso contrario la integridad del SB3010 estará comprometida.

Parámetros	Descripción				
<i>code</i>	string conteniendo el código del producto.				
	<table border="1"> <tr> <td>SB3010</td> <td>Hasta 12 caracteres</td> </tr> <tr> <td>BemaPos</td> <td>Hasta 13 caracteres</td> </tr> </table>	SB3010	Hasta 12 caracteres	BemaPos	Hasta 13 caracteres
SB3010	Hasta 12 caracteres				
BemaPos	Hasta 13 caracteres				
<i>description</i>	string conteniendo la Descripción del producto.				
	<table border="1"> <tr> <td>SB3010</td> <td>Hasta 20 caracteres</td> </tr> <tr> <td>SB3010</td> <td>Hasta 20 caracteres</td> </tr> </table>	SB3010	Hasta 20 caracteres	SB3010	Hasta 20 caracteres
SB3010	Hasta 20 caracteres				
SB3010	Hasta 20 caracteres				
<i>price</i>	tipo CURRENCY conteniendo precio del producto				
<i>taxIndex</i>	índice para la situación tributaria del producto				
	<table border="1"> <tr> <td>SB3010 Brasil</td> <td>Índices 14,15 y 16 son reservados para FF, II y NN, respectivamente.</td> </tr> <tr> <td>BemaPos</td> <td>> 0</td> </tr> </table>	SB3010 Brasil	Índices 14,15 y 16 son reservados para FF, II y NN, respectivamente.	BemaPos	> 0
SB3010 Brasil	Índices 14,15 y 16 son reservados para FF, II y NN, respectivamente.				
BemaPos	> 0				
<i>department</i>	índice para el departamento al cual ese producto esta vinculado				
	<table border="1"> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> </table>				

API Aruak

stockAmount

cantidad en stock

SB3010	Con 3 casas decimales.
BemaPos	idem

flags

Flags específicos al dispositivo

SB3010	Combinación (OU lógico) de las siguientes opciones : item pesado (0x01), item litrado (0x02) e item tres casas decimales (0x04); o 0x00 para ninguna de las opciones arriba
BemaPos	TIPO DE UNIDAD ÉS EXCLUSIVO 12 - Unidad 11 -Litro 10 - Metro 9 - kilo 8 – paquete Mapa de bits: 7 Precio Fijo 6 Producto Mixto (tecla de pedido de precio habilitada) 5 Producto Compuesto 4 Pedido Remoto 3 Producto Comisionado 2 Cupón Unitario 1 Controla estoque.

keyGroup

Grupo de teclas de acceso rápido al cual ese ítem está asociado.

SB3010	0 a 5. Cuando 0, producto es vendido normalmente.
BemaPos	No se aplica - Parámetro Ignorado

costPrice

tipo CURRENCY conteniendo precio de costo del producto

SB3010	No se aplica - Parámetro Ignorado
BemaPos	Parámetro obligatorio

margin

Contiene la margen de venta del producto

SB3010	No se aplica - Parámetro Ignorado
BemaPos	Margin percentual. Ex: "1250" representa margem de 12,5 %

API Aruak

groupDepartment

índice para el grupo de departamento al cual ese producto esta vinculado

SB3010	No se aplica - Parámetro Ignorado
BemaPos	Representa el conjunto “departamentos” que contiene “setores”

commission

Contiene el valor de comisión atribuido a este producto

SB3010	No se aplica - Parámetro Ignorado
BemaPos	valor porcentual. Ej: “550” representa comissão de 5,5 %

remoteNumber

Indica si el producto tendrá una impresión especial nombreando su destino final

SB3010	No se aplica - Parámetro Ignorado
BemaPos	0 – Sin impresión especial; Mayor que cero - debe contener el índice del pedido remoto registrado en el PDV(Ex: “cozinha(01), bar(02), etc”);

minimumStock

cantidad minima en el stock para ese articulo

SB3010	No se aplica - Parámetro Ignorado
BemaPos	Con 3 casas decimales Ex: “55500” representa 55,5

idealStock

cantidad ideal en el stock para ese articulo

SB3010	No se aplica - Parámetro Ignorado
BemaPos	Con 3 casas decimales Ex: “55500” representa 55,5

Retorno

Valor	Descripción
1	Registros inserido con exito
-2	Parámetro inválido
-1	Error de ejecución
0	Servidor no inicializado

Vea también Métodos *InsertTable* y *InitializeUpdate*.

AddAssemblyItemEntry

Sintaxe LONG AddAssemblyItemEntry (BSTR code, BSTR partCode, LONG quantity);

Comentário Añada un registro a la tabla de ficha técnica – articulos compuestos - para posterior actualización/inserción de los mismos a través del método *InsertTable*. Debe ser llamada para cada componente que se desee insertar en la composicion del producto. Es necesario llamar el método *InitializeUpdate* antes de la primera llamada a ese método.

IMPORTANTE: No es sपोर्टado em el SB3010.

Parámetros	Descripción				
<i>code</i>	string conteniendo el código del producto.				
	<table border="1"> <tr> <td></td> <td></td> </tr> <tr> <td>BemaPos</td> <td>Hasta 13 caracteres</td> </tr> </table>			BemaPos	Hasta 13 caracteres
BemaPos	Hasta 13 caracteres				
<i>partCode</i>	string conteniendo el código del producto componente.				
	<table border="1"> <tr> <td></td> <td></td> </tr> <tr> <td>BemaPos</td> <td>Hasta 13 caracteres</td> </tr> </table>			BemaPos	Hasta 13 caracteres
BemaPos	Hasta 13 caracteres				
<i>quantity</i>	Cantidad de componente en la composición				
	<table border="1"> <tr> <td></td> <td></td> </tr> <tr> <td>BemaPos</td> <td>3 casas decimales. Ex: “55500” = 55,5</td> </tr> </table>			BemaPos	3 casas decimales. Ex: “55500” = 55,5
BemaPos	3 casas decimales. Ex: “55500” = 55,5				

Retorno

Valor	Descripción
1	Registros insertados con éxito
-2	Parámetro inválido
-1	Error de ejecución
0	Servidor no inicializado

Vea también Métodos *UpdateTable* e *InitializeUpdate*.

UpdateProductEntry

Sintaxis LONG UpdateProductEntry (SHORT action, BSTR code, BSTR description, CY price, SHORT taxIndex, SHORT department, LONG StockAmount, SHORT flags, SHORT keyGroup, CY costPrice, LONG margin, SHORT groupDepartment, LONG comisión, SHORT remoteNumber, LONG minimumStock, LONG idealStock).

Comentario Añada un registro a la tabla de actualización de productos para posterior actualización/inserción o remoción de los mismos a través del método *UpdateTable*. Debe ser llamada para cada producto que se desee actualizar/insertar o borrar en el dispositivo. Es necesario llamar el método *InitializeUpdate* antes de la primera llamada a ese método.

Parámetros	Descripción				
<i>action</i>	SHORT especificando el tipo de la operación .				
	<table border="1"> <tr> <td>SB3010</td> <td>1 = remoción, 3 = inserción o actualización, 4 cambio sin modificar la provisión en el equipo</td> </tr> <tr> <td></td> <td></td> </tr> </table>	SB3010	1 = remoción, 3 = inserción o actualización, 4 cambio sin modificar la provisión en el equipo		
SB3010	1 = remoción, 3 = inserción o actualización, 4 cambio sin modificar la provisión en el equipo				
<i>code</i>	string conteniendo el código del producto.				
	<table border="1"> <tr> <td>SB3010</td> <td>Hasta 12 caracteres</td> </tr> <tr> <td></td> <td></td> </tr> </table>	SB3010	Hasta 12 caracteres		
SB3010	Hasta 12 caracteres				
<i>description</i>	string conteniendo la Descripción del producto.				
	<table border="1"> <tr> <td>SB3010</td> <td>Hasta 20 caracteres</td> </tr> <tr> <td></td> <td></td> </tr> </table>	SB3010	Hasta 20 caracteres		
SB3010	Hasta 20 caracteres				

API Aruak

price tipo CURRENCY conteniendo precio del producto

taxIndex índice para la situación tributaria del producto

SB3010 Brasil	Índices 14,15 y 16 son reservados para FF, II y NN, respectivamente.

department índice para el departamento al cual ese producto esta vinculado

stockAmount cantidad en stock

SB3010	Con 3 casas decimales.

flags Flags específicos al dispositivo

SB3010	Combinación (OU lógico) de las siguientes opciones : ítem pesado (0x01), ítem litrado (0x02) e item três casas decimales (0x04); o 0x00 para ninguna de las opciones de arriba.

keyGroup Grupo de teclas de acceso rápido al cual ese ítem está asociado.

SB3010	0 a 5. Cuando 0, producto es vendido normalmente.

costPrice tipo CURRENCY conteniendo precio de costo del producto

SB3010	No se aplica - Parámetro Ignorado
BemaPos	Parámetro obligatorio

margin Contiene la margen de venta del producto

SB3010	No se aplica - Parámetro Ignorado
BemaPos	Margen percentual. Ej: "1250" representa margen de 12,5 %

groupDepartment índice para el grupo de departamento a cual ese producto esta vinculado

API Aruak

SB3010	No se aplica - Parámetro Ignorado
BemaPos	Representa el conjunto “departamentos” que contiene “setores”

commission

Contiene el valor de comisión atribuido a este producto

SB3010	No se aplica - Parámetro Ignorado
BemaPos	valor percentual. Ej: “550” representa comisión de 5,5 %

remoteNumber

Indica si el producto tendrá una impresión especial nombrando su destino final

SB3010	No se aplica - Parámetro Ignorado
BemaPos	0 – Sin impresión especial; Mayor que cero - debe contener el índice del pedido remoto registrado en el PDV(Ex: “cozinha(01), bar(02), etc”);

minimumStock

cantidad minima en el stock para ese articulo

SB3010	No se aplica - Parámetro Ignorado
BemaPos	Con 3 casas decimales Ex: “55500” representa 55,5

idealStock

cantidad ideal en el stock para ese articulo

SB3010	No se aplica - Parámetro Ignorado
BemaPos	Con 3 casas decimales Ex: “55500” representa 55,5

Retorno

Valor	Descripción
1	Registros insertados con éxito
-2	Parámetro inválido
-1	Error de ejecución
0	Servidor no inicializado

Veá también Métodos *UpdateTable* e *InitializeUpdate*.

AddClientEntry

Sintaxis LONG AddClientEntry (LONG key, BSTR clientId, BSTR name, BSTR address1, BSTR complement, SHORT status, BSTR localArea, BSTR city, BSTR state, BSTR zip, BSTR fone1, BSTR fone2, BSTR contactName, BSTR email, CY currentDebt, CY debtLimit, BOOL hasDebtLimit).

Comentario Adiciona un registro a la tabla de clientes para posterior actualización del dispositivo con el método *InsertTable*. Debe ser llamada para cada cliente que desee insertar en la tabla del dispositivo. Es necesario llamar el método *InitializeUpdate* antes de la primera llamada a ese método.

Parámetros	Descripción				
<i>key</i>	código del cliente				
<i>clientId</i>	string conteniendo documento del cliente				
	<table border="1"> <tr> <td>SB3010</td> <td>String numérica conteniendo cpf/cgc con hasta 14 caracteres</td> </tr> <tr> <td></td> <td></td> </tr> </table>	SB3010	String numérica conteniendo cpf/cgc con hasta 14 caracteres		
SB3010	String numérica conteniendo cpf/cgc con hasta 14 caracteres				
<i>name</i>	string con el nombre del cliente				
	<table border="1"> <tr> <td>SB3010</td> <td>String con hasta 30 caracteres</td> </tr> <tr> <td></td> <td></td> </tr> </table>	SB3010	String con hasta 30 caracteres		
SB3010	String con hasta 30 caracteres				
<i>address1</i>	string conteniendo primera línea con informaciones de dirección				
	<table border="1"> <tr> <td>SB3010</td> <td>String con hasta 40 caracteres</td> </tr> <tr> <td></td> <td></td> </tr> </table>	SB3010	String con hasta 40 caracteres		
SB3010	String con hasta 40 caracteres				
<i>complement</i>	string conteniendo segunda línea con información de dirección				

API Aruak

SB3010	String con hasta 40 caracteres

status *información de status del cliente especifica al dispositivo*

SB3010	0, 1 o 2 para VERDE, AMARILLO y ROJO respectivamente.
BemaPos	No se aplica - Parámetro Ignorado

localArea *string conteniendo nombre del bairrio*

SB3010	No se aplica - Parámetro Ignorado
BemaPos	String con hasta 15 caracteres

city *string conteniendo nombre de la ciudad.*

SB3010	No se aplica - Parámetro Ignorado
BemaPos	String con hasta 20 caracteres

state *string conteniendo el nombre del estado*

SB3010	No se aplica - Parámetro Ignorado
BemaPos	String con hasta 2 caracteres

zip *string conteniendo el Código Postal*

SB3010	No se aplica - Parámetro Ignorado
BemaPos	String con hasta 8 caracteres

fone1 *string conteniendo numero de teléfono principal para contacto*

SB3010	No se aplica - Parámetro Ignorado
BemaPos	String con hasta 10 caracteres

fone2 *string conteniendo numero de teléfono secundario para contacto*

SB3010	No se aplica - Parámetro Ignorado
BemaPos	String con hasta 10 caracteres

contactName *string conteniendo nombre de la persona para contacto*

API Aruak

<i>email</i>	SB3010	No se aplica - Parámetro Ignorado
	BemaPos	String con hasta 20 caracteres
string conteniendo email de la persona		
	SB3010	No se aplica - Parámetro Ignorado
	BemaPos	String con hasta 40 caracteres
<i>currentDebt</i>	tipo CURRENCY conteniendo el débito actual del cliente	
	SB3010	No se aplica - Parámetro Ignorado
<i>debtLimit</i>	tipo CURRENCY conteniendo el débito límite del cliente	
	SB3010	No se aplica - Parámetro Ignorado
<i>hasDebtLimit</i>	tipo BOOL indicando si el cliente tiene límite de crédito o no	
	SB3010	No se aplica - Parámetro Ignorado
	BemaPos	Si <i>true</i> utiliza el límite pasado em el parámetro <i>currentDebt</i>

Retorno

Valor	Descripción
1	Registros insertado con exito
-2	Parámetro inválido
-1	Error de ejecución
0	Servidor no inicializado

Vea también Métodos *InsertTable* e *InitializeUpdate*.

AddOperatorEntry

Sintaxis LONG AddOperatorEntry (BSTR id, BSTR name, BSTR password, SHORT profile, LONG permissions);

Comentario Adiciona un registro a la tabla de operadores para posterior actualización del dispositivo con el método *InsertTable*. Debe ser llamada para cada

API Aruak

operador que desee insertar en la tabla del dispositivo. Es necesario llamar el método *InitializeUpdate* antes de la primera llamada a ese método.

Parámetros	Descripción				
<i>id</i>	string conteniendo el código del operador.				
	<table border="1"> <tr> <td style="text-align: center;">SB3010</td> <td>String con hasta 4 caracteres</td> </tr> <tr> <td> </td> <td> </td> </tr> </table>	SB3010	String con hasta 4 caracteres		
SB3010	String con hasta 4 caracteres				
<i>name</i>	string con el nombre del operador				
	<table border="1"> <tr> <td style="text-align: center;">SB3010</td> <td>String con hasta 15 caracteres</td> </tr> <tr> <td> </td> <td> </td> </tr> </table>	SB3010	String con hasta 15 caracteres		
SB3010	String con hasta 15 caracteres				
<i>password</i>	string con la clave del operador				
	<table border="1"> <tr> <td style="text-align: center;">SB3010</td> <td>String con hasta 4 caracteres</td> </tr> <tr> <td style="text-align: center;">BemaPos</td> <td>String con hasta 6 caracteres</td> </tr> </table>	SB3010	String con hasta 4 caracteres	BemaPos	String con hasta 6 caracteres
	SB3010	String con hasta 4 caracteres			
BemaPos	String con hasta 6 caracteres				
<i>profile</i>	perfil del operador				
	<table border="1"> <tr> <td style="text-align: center;">SB3010</td> <td>1 – para gerente y 2 – para normal</td> </tr> <tr> <td style="text-align: center;">BemaPos</td> <td>1-Caixa, 2-Supervisor, 3-Gerente, 4-Diretor, 5-Proprietário.</td> </tr> </table>	SB3010	1 – para gerente y 2 – para normal	BemaPos	1-Caixa, 2-Supervisor, 3-Gerente, 4-Diretor, 5-Proprietário.
	SB3010	1 – para gerente y 2 – para normal			
BemaPos	1-Caixa, 2-Supervisor, 3-Gerente, 4-Diretor, 5-Proprietário.				
<i>permissions</i>	LONG conteniendo combinación (mascara de bits) con las permisiones del operador				
	<table border="1"> <tr> <td style="text-align: center;">SB3010</td> <td>0x000001: desconto item 0x000002: desconto subtotal 0x000004: cancelamento item 0x000008: cancelamento cupom 0x000010: Lectura x 0x000020: reducción z 0x000040: sangria 0x000080: suprimiento 0x000100: limit 0x000200: informes 0x000400: incremento item 0x000800: incremento subtotal</td> </tr> <tr> <td style="text-align: center;">BemaPos</td> <td>No se aplica - Parámetro Ignorado</td> </tr> </table>	SB3010	0x000001: desconto item 0x000002: desconto subtotal 0x000004: cancelamento item 0x000008: cancelamento cupom 0x000010: Lectura x 0x000020: reducción z 0x000040: sangria 0x000080: suprimiento 0x000100: limit 0x000200: informes 0x000400: incremento item 0x000800: incremento subtotal	BemaPos	No se aplica - Parámetro Ignorado
SB3010	0x000001: desconto item 0x000002: desconto subtotal 0x000004: cancelamento item 0x000008: cancelamento cupom 0x000010: Lectura x 0x000020: reducción z 0x000040: sangria 0x000080: suprimiento 0x000100: limit 0x000200: informes 0x000400: incremento item 0x000800: incremento subtotal				
BemaPos	No se aplica - Parámetro Ignorado				

Retorno

Valor	Descripción
1	Registros insertados con éxito
-2	Parámetro inválido
-1	Error de ejecución
0	Servidor no inicializado

Vea también Métodos *InsertTable* e *InitializeUpdate*.

AddSalesmanEntry

Sintaxis LONG AddSalemanEntry (BSTR id, BSTR name, LONG commission);

Comentario Adiciona un registro a la tabla de vendedores para posterior actualización del dispositivo con el método *InsertTable*. Debe ser llamada para cada vendedor que desee insertar en la tabla del dispositivo. Es necesario llamar el método *InitializeUpdate* antes de la primera llamada a ese método.

Parámetros	Descripción				
id	string conteniendo el código del operador.				
	<table border="1"> <tr> <td>SB3010</td> <td>String con hasta 4 caracteres</td> </tr> <tr> <td>BemaPos</td> <td>String com hasta 4 caracteres</td> </tr> </table>	SB3010	String con hasta 4 caracteres	BemaPos	String com hasta 4 caracteres
	SB3010	String con hasta 4 caracteres			
BemaPos	String com hasta 4 caracteres				
name	string con el nombre del operador				
	<table border="1"> <tr> <td>SB3010</td> <td>String con hasta 15 caracteres</td> </tr> <tr> <td>BemaPos</td> <td>String com hasta 15 caracteres</td> </tr> </table>	SB3010	String con hasta 15 caracteres	BemaPos	String com hasta 15 caracteres
	SB3010	String con hasta 15 caracteres			
BemaPos	String com hasta 15 caracteres				
commission	LONG con la comisión del vendedor				
	<table border="1"> <tr> <td>SB3010</td> <td>No se aplica - Parámetro Ignorado</td> </tr> <tr> <td>BemaPos</td> <td>valor porcentual. Ej: "550" representa comisión de 5,5 %</td> </tr> </table>	SB3010	No se aplica - Parámetro Ignorado	BemaPos	valor porcentual. Ej: "550" representa comisión de 5,5 %
	SB3010	No se aplica - Parámetro Ignorado			
BemaPos	valor porcentual. Ej: "550" representa comisión de 5,5 %				

Retorno

Valor	Descripción
1	Registros insertados con éxito
-2	Parámetro inválido
-1	Error de ejecución
0	Servidor no inicializado

Vea también Métodos *InsertTable* e *InitializeUpdate*.

AddDepartmentEntry

Sintaxis LONG AddDepartmentEntry (SHORT number, BSTR description, LONG flags, SHORT taxIndex);

Comentario Adiciona un registro a la tabla de departamentos [setores] para posterior actualización del dispositivo con el método *InsertTable*. Debe ser llamada para cada departamento [setor] que desee insertar en la tabla del dispositivo. Es necesario llamar el método *InitializeUpdate* antes de la primera llamada a ese método.

Parámetros	Descripción				
<i>number</i>	número del departamento				
	<table border="1"> <tbody> <tr> <td>SB3010</td> <td>1 hasta 99</td> </tr> <tr> <td>BemaPos</td> <td>1 hasta 99</td> </tr> </tbody> </table>	SB3010	1 hasta 99	BemaPos	1 hasta 99
SB3010	1 hasta 99				
BemaPos	1 hasta 99				
<i>description</i>	string con la Descripción del departamento				
	<table border="1"> <tbody> <tr> <td>SB3010</td> <td>String con hasta 15 caracteres</td> </tr> <tr> <td>BemaPos</td> <td>String con hasta 15 caracteres</td> </tr> </tbody> </table>	SB3010	String con hasta 15 caracteres	BemaPos	String con hasta 15 caracteres
SB3010	String con hasta 15 caracteres				
BemaPos	String con hasta 15 caracteres				
<i>flags</i>	Flags específicas al dispositivo				
	<table border="1"> <tbody> <tr> <td>SB3010</td> <td>Grupo del departamento de 1 hasta 99</td> </tr> <tr> <td>BemaPos</td> <td>No se aplica – parámetro ignorado</td> </tr> </tbody> </table>	SB3010	Grupo del departamento de 1 hasta 99	BemaPos	No se aplica – parámetro ignorado
SB3010	Grupo del departamento de 1 hasta 99				
BemaPos	No se aplica – parámetro ignorado				

API Aruak

taxIndex

Índice para el IVA

SB3010	No se aplica – parámetro ignorado
BemaPos	Índice del IVA(alicuota) vinculado a esse departamento

Retorno

Valor	Descripción
1	Registros insertado con éxito
-2	Parámetro inválido
-1	Error de ejecución
0	Servidor no inicializado

Vea también Métodos *InsertTable* e *InitializeUpdate*.

AddDepartmentGroupEntry

Sintaxis LONG AddDepartmentGroupEntry (SHORT number, BSTR description);

Comentario Adiciona un registro a la tabla del grupo de departamentos [departamento] para posterior actualización del dispositivo con el método *InsertTable*. Debe ser llamada para cada grupo [departamento] que desee insertar en la tabla del dispositivo. Es necesario llamar el método *InitializeUpdate* antes de la primera llamada a ese método.

Parámetros	Descripción
<i>number</i>	número del departamento
SB3010	1 hasta 99
BemaPos	1 hasta 99
<i>description</i>	string con la descripción del grupo de departamentos [departamento]
SB3010	String con hasta 20 caracteres
BemaPos	String con hasta 15 caracteres

Retorno

Valor	Descripción
1	Registros insertados con éxito
-2	Parámetro inválido
-1	Error de ejecución
0	Servidor no inicializado

Vea también Métodos *InsertTable* e *InitializeUpdate*.

AddMessageEntry

Sintaxe LONG AddMessageEntry (BSTR text, SHORT type, LONG flags);

Comentario Adiciona un registro a la tabla de mensajes [encabezados y notas de pié de pagina] para posterior actualización del dispositivo con el método *InsertTable*. Debe ser llamada para cada mensaje que desee insertar en la tabla del dispositivo. Es necesario llamar el método *InitializeUpdate* antes de la primera llamada a ese método.

IMPORTANTE: No es soportado por el SB3010.

Parámetros	Descripción				
<i>text</i>	texto de la mensaje				
	<table border="1"> <tr> <td></td> <td></td> </tr> <tr> <td>BemaPos</td> <td>hasta 48 caracteres</td> </tr> </table>			BemaPos	hasta 48 caracteres
BemaPos	hasta 48 caracteres				
<i>type</i>	identifica el tipo de la mensaje				
	<table border="1"> <tr> <td></td> <td></td> </tr> <tr> <td>BemaPos</td> <td>0 – encabezado, 1 –nota de pié</td> </tr> </table>			BemaPos	0 – encabezado, 1 –nota de pié
BemaPos	0 – encabezado, 1 –nota de pié				
<i>flags</i>	Opciones para la mensaje				
	<table border="1"> <tr> <td></td> <td></td> </tr> <tr> <td>BemaPos</td> <td> Mapa de bits: 0x000001: expandido 0x000002: itálico 0x000004: negrito Exclusivo (normal o condensado): 0x000010: NORMAL 0x000040: CONDENSADO </td> </tr> </table>			BemaPos	Mapa de bits: 0x000001: expandido 0x000002: itálico 0x000004: negrito Exclusivo (normal o condensado): 0x000010: NORMAL 0x000040: CONDENSADO
BemaPos	Mapa de bits: 0x000001: expandido 0x000002: itálico 0x000004: negrito Exclusivo (normal o condensado): 0x000010: NORMAL 0x000040: CONDENSADO				

Retorno

Valor	Descripción
1	Registros insertados con éxito
-2	Parámetro inválido
-1	Error de ejecución
0	Servidor no inicializado

Vea también Métodos *InsertTable* e *InitializeUpdate*.

AddTaxEntry

Sintaxe LONG AddTaxEntry (LONT taxValue, SHORT index, SHORT type);

Comentário Adiciona un registro a la tabla de IVAS o alícuotas para posterior actualización del dispositivo con el método *InsertTable*. Debe ser llamada para cada IVA que desee insertar en la tabla del dispositivo. Es necesario llamar el método *InitializeUpdate* antes de la primera llamada a ese método.

IMPORTANTE: No es soportado por el SB3010.

Parámetros	Descripción				
<i>taxValue</i>	Valor porcentual del IVA				
	<table border="1"> <tbody> <tr> <td></td> <td></td> </tr> <tr> <td>BemaPos</td> <td>2 casas decimales. de 0% hasta 99,99%. Ex: "2755" = 27,55%</td> </tr> </tbody> </table>			BemaPos	2 casas decimales. de 0% hasta 99,99%. Ex: "2755" = 27,55%
BemaPos	2 casas decimales. de 0% hasta 99,99%. Ex: "2755" = 27,55%				
<i>Index</i>	índice que está gravado el IVA em el PDV				
	<table border="1"> <tbody> <tr> <td></td> <td></td> </tr> <tr> <td>BemaPos</td> <td>1 a 5</td> </tr> </tbody> </table>			BemaPos	1 a 5
BemaPos	1 a 5				
<i>type</i>	Tipo del IVA (ALICUOTA)				
	<table border="1"> <tbody> <tr> <td></td> <td></td> </tr> <tr> <td>BemaPos</td> <td>Ignorado</td> </tr> </tbody> </table>			BemaPos	Ignorado
BemaPos	Ignorado				

Retorno

Valor	Descripción
1	Registros insertados con éxito
-2	Parámetro inválido
-1	Error de ejecución
0	Servidor no inicializado

Vea también Métodos *InsertTable* e *InitializeUpdate*.

AddTenderingTypeEntry

Sintaxe LONG AddTenderingTypeEntry (BSTR description, LONG flags);

Comentario Adiciona un registro a la tabla de formas de pago para posterior actualización del dispositivo con el método *InsertTable*. Debe ser llamada para cada forma que desee insertar en la tabla del dispositivo. Es necesario llamar el método *InitializeUpdate* antes de la primera llamada a ese método.

IMPORTANTE: No es soportado por el BemaPos.

Parámetros	Descripción				
<i>description</i>	nombre de la forma				
	<table border="1"> <tr> <td>SB3010</td> <td>hasta 15 caracteres</td> </tr> <tr> <td></td> <td></td> </tr> </table>	SB3010	hasta 15 caracteres		
SB3010	hasta 15 caracteres				
<i>flags</i>	opciones para la forma				
	<table border="1"> <tr> <td>SB3010</td> <td>Parametro ignorado</td> </tr> <tr> <td></td> <td></td> </tr> </table>	SB3010	Parametro ignorado		
SB3010	Parametro ignorado				

Retorno Valor	Descripción
1	Registros insertados con éxito
-2	Parámetro inválido
-1	Error de ejecución
0	Servidor no inicializado

Vea también Métodos *InsertTable* e *InitializeUpdate*.

InitializeUpdate

Sintaxis void InitializeUpdate (SHORT tableId);

Comentario Prepara una actualización/inserción de una tabla específica. Debe ser llamada antes de empezar al inserción de los registros.

Parámetros	Descripción																														
<i>tableId</i>	Identificador de la tabla a ser cargada.																														
	<table border="1"> <thead> <tr> <th>Valor</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Tabla de productos</td> </tr> <tr> <td>2</td> <td>Tabla de Clientes</td> </tr> <tr> <td>3</td> <td>Tabla de operadores</td> </tr> <tr> <td>4</td> <td>Tabla de departamentos (setores)</td> </tr> <tr> <td>5</td> <td>Tabla de tributos - Solo BemaPos</td> </tr> <tr> <td>6</td> <td>Tabla de vendedores</td> </tr> <tr> <td>7</td> <td>Tabla de grupos de departamentos (departamentos)</td> </tr> <tr> <td>8</td> <td>Tabla de finalizadores - solo SB3010</td> </tr> <tr> <td>9</td> <td>Tabla de información del sistema [solamente Lectura]</td> </tr> <tr> <td>10</td> <td>Tabla de mensajes [solamente BemaPos]</td> </tr> <tr> <td>11</td> <td>Tabla de ficha tecnica [solamente BemaPos]</td> </tr> <tr> <td>12</td> <td>Tabla de Logo [solamente BemaPos]</td> </tr> <tr> <td>13</td> <td>Tabla de Backup [solamente BemaPos]</td> </tr> <tr> <td>14</td> <td>Tabla de Stock [solamente BemaPos]</td> </tr> </tbody> </table>	Valor	Descripción	1	Tabla de productos	2	Tabla de Clientes	3	Tabla de operadores	4	Tabla de departamentos (setores)	5	Tabla de tributos - Solo BemaPos	6	Tabla de vendedores	7	Tabla de grupos de departamentos (departamentos)	8	Tabla de finalizadores - solo SB3010	9	Tabla de información del sistema [solamente Lectura]	10	Tabla de mensajes [solamente BemaPos]	11	Tabla de ficha tecnica [solamente BemaPos]	12	Tabla de Logo [solamente BemaPos]	13	Tabla de Backup [solamente BemaPos]	14	Tabla de Stock [solamente BemaPos]
Valor	Descripción																														
1	Tabla de productos																														
2	Tabla de Clientes																														
3	Tabla de operadores																														
4	Tabla de departamentos (setores)																														
5	Tabla de tributos - Solo BemaPos																														
6	Tabla de vendedores																														
7	Tabla de grupos de departamentos (departamentos)																														
8	Tabla de finalizadores - solo SB3010																														
9	Tabla de información del sistema [solamente Lectura]																														
10	Tabla de mensajes [solamente BemaPos]																														
11	Tabla de ficha tecnica [solamente BemaPos]																														
12	Tabla de Logo [solamente BemaPos]																														
13	Tabla de Backup [solamente BemaPos]																														
14	Tabla de Stock [solamente BemaPos]																														

Vea también Método *InsertTable*.

InsertTable

Tipo Assíncrono

Sintaxis LONG InsertTable (LONG terminalId, SHORT tableId);

Comentario Inserta un pedido de inserción/carga de la tabla pasada en la cola de pedidos del terminal especificado. Ese comando será ejecutado en la próxima conexión del terminal con el servidor. Debe ser llamado después de la adición del último registro para actualización a través de los métodos *AddXXXXEntry*. Será disparado el evento *OnTerminalUpdate* después del término de la inserción. Para espera síncrona del resultado de la operación, debe se utilizar los métodos *WaitForOperation* y *GetOperationStatus*.

Ese método retorna un id único, incrementado a cada nueva requisición, que puede ser utilizado posteriormente para identificar esa transacción tanto en el evento *OnTransactionComplete* o para cancelamiento posterior de esa operación con el comando *VoidTransaction*. Ese id es valido encuanto el servidor estuvier activo (aplicativo rodando) y será reiniciado a partir del valor uno (1) para cada reinicio del servidor.

IMPORTANTE: Esa operación hace con que todos los registros de la tabla de productos sean borrados antes de la nueva inserción; por esto debe ser utilizada solamente para la carga inicial del device (Carga más rápida). Para actualización de la tabla utilizar el método *UpdateTable*.

Parámetros	Descripción
<i>terminalId</i>	Identificador para la terminal.
<i>tableId</i>	identificador de la tabla. Ver <i>InitializeUpdate</i>

Retorno

Valor	Descripción
> 0	Id único para la operación si pedido insertado con éxito (<i>TransactionId</i>)
0	Servidor no inicializado
-3	Terminal inválido
-4	Operación no soportada

Evento asociado *OnTerminalUpdate*, *OnTransactionComplete*.

Vea también Métodos *InitializeUpdate*, *WaitForOperation*, *VoidInsertTable*, *VoidTransaction* e *GetOperationStatus* e métodos *AddXXXXEntry*.

UpdateTable

Tipo Assíncrono

Sintaxis LONG UpdateTable (LONG terminalId, SHORT tableId);

Comentario Inserta un pedido de actualización de la tabla (remoción/inserción o cambio de registros) pasada en la cola de pedidos del terminal especificado. Ese comando será ejecutado en la próxima conexión del terminal con el servidor. Debe ser llamado después de la adición del último registro para actualización a través de los métodos *UpdateXXXXEntry*. Será disparado el evento *OnTerminalUpdate* después del término de la inserción. Para espera síncrona del resultado de la operación, debe se utilizar los métodos *WaitForOperation* y *GetOperationStatus*.

Ese método retorna un id único, incrementado a cada nueva requisición, que puede ser utilizado posteriormente para identificar esa transacción tanto en el evento *OnTransactionComplete* o para cancelamiento posterior de esa operación con el comando *VoidTransaction*. Ese id es valido encuanto el servidor estuvier activo (aplicativo rodando) y será reiniciado a partir del valor uno (1) para cada reinicio del servidor.

Parámetros	Descripción
<i>terminalId</i>	Identificador para la terminal.
<i>tableId</i>	identificador de la tabla. Ver <i>InitializeUpdate</i>

Retorno

Valor	Descripción
> 0	Id único para la operación si pedido insertado con éxito.
0	Servidor no inicializado
-3	Terminal inválido
-4	Operação no suportada

Evento asociado *OnTerminalUpdate, OnTransactionComplete.*

Vea también Métodos *InitializeUpdate, WaitForOperation, VoidInsertTable, VoidTransaction* e *GetOperationStatus* e métodos *UpdateXXXXEntry*.

VoidInsertTable

Tipo Síncrono

Sintaxis LONG VoidInsertTable (LONG terminalId, SHORT tableId);

Comentario Remove un pedido de inserción, carga o actualización de la tabla pasada de la cola de pedidos del terminal especificado, evitando así que la operación ocurra en la próxima conexión del dispositivo.

Parámetros	Descripción
<i>terminalId</i>	Identificador para la terminal.
<i>tableId</i>	identificador de la tabla. Ver <i>InitializeUpdate</i>

Retorno

Valor	Descripción
1	Pedido removido con éxito
0	Servidor no inicializado
-3	Terminal inválido
-4	Operación no soportada

Evento asociado *Ningun.*

ReadTable

Tipo Asíncrono

Sintaxis LONG ReadTable (LONG terminalId, SHORT tableId);

Comentario Inserta un pedido de Lectura de la tabla en la cola de pedidos del terminal especificado, cancelando así la operación de Lectura pedida.

Ese método retorna un id único, incrementado a cada nueva requisición, que puede ser utilizado posteriormente para identificar esa transacción tanto en el evento

OnTransactionComplete o para cancelamiento posterior de esa operación con el comando *VoidTransaction*. Ese id es valido encuanto el servidor estuvier activo (aplicativo rodando) y será reiniciado a partir del valor uno (1) para cada reinicio del servidor.

IMPORTANTE: Feature todavia **no** soportada por el **BemaPos**.

Parámetros	Descripción
<i>terminalId</i>	Identificador para la terminal.
<i>tableId</i>	identificador de la tabla. Ver <i>InitializeUpdate</i>

Retorno

Valor	Descripción
> 0	Id único para la operación si pedido insertado con exito.
0	Servidor no inicializado
-3	Terminal inválido
-4	Operación no soportada

Evento asociado *OnTableReceived, OnTransactionComplete*.

Vea también Métodos *WaitForOperation, VoidInsertTable, VoidTransaction* e *GetOperationStatus*.

VoidReadTable

Tipo Síncrono

Sintaxis LONG VoidReadTable (LONG terminalId, SHORT tableId);

Comentario Remove un pedido de Lectura de la tabla de la cola de pedidos del terminal especificado.

Parámetros	Descripción
<i>terminalId</i>	Identificador para la terminal.
<i>tableId</i>	Identificador de la tabla. Ver <i>InitializeUpdate</i>

Retorno

Valor	Descripción
1	Pedido removido con exito
0	Servidor no inicializado
-3	Terminal inválido
-4	Operación no soportada

Evento asociado *Ningun*.

VoidTransaction

Tipo Síncrono
 Sintaxis LONG VoidReadTable (LONG transaction);
 Comentario Anula una transacción (pedido de Lectura, carga o actualización) de la cola de pedidos del servidor

Parámetros	Descripción
<i>transactionId</i>	Identificador para la transacción, obtenido en el retorno de las funciones <i>InsertTable</i> , <i>ReadTable</i> , <i>UpdateTable</i> o <i>DoGenericOperation</i> ..

Retorno

Valor	Descripción
1	Transacción anulada con éxito
0	Servidor no inicializado
-3	Transacción inexistente
-4	Operación no soportada

Evento asociado *Ningun.*

WaitForOperation

Sintaxis LONG WaitForOperation (LONG timeOut);
 Comentario Aguarda por la finalización de una operación de inserción (*InsertTable*) o Lectura (*ReadTable*).

Parámetros	Descripción
<i>timeOut</i>	time out en mili segundos.

Retorno

Valor	Descripción
[1-100]	Percentual completo de la operación, si finalizada debe ser 100
0	time out ocurrió
-1	Falla em la ejecución

Vea también Método *GetOperationStatus*.

GetOperationStatus

Sintaxis LONG GetOperationStatus (void);

API Aruak

Comentario Retorna el percentual completo de la operación de inserción y Lectura previamente llamada.

Retorno

Valor	Descripción
[0-100]	Percentual completo de la operación
-1	Falla na ejecución

Vea también Método *WaitForOperation*.

DoGenericOperation

Tipo Assíncrono/Síncrono

Sintaxis LONG DoGenericOperation (LONG terminalId, LONG operation);

Comentario Inserta una requisición genérica em la cola de pedidos del terminal especificado. Ese comando será ejecutado em la próxima conexión del terminal con el servidor. Será ejecutado el evento *OnOperationComplete* después del término de la ejecución del pedido. Para espera síncrona del resultado de la operación, debe se utilizar los métodos *WaitForOperation* y *GetOperationStatus*.

Ese método retorna un id único, incrementado a cada nueva requisición, que puede ser utilizado posteriormente para identificar esa transacción tanto en el evento *OnTransactionComplete* o para cancelamiento posterior de esa operación con el comando *VoidTransaction*. Ese id es valido encuanto el servidor estuvier activo (aplicativo rodando) y será reiniciado a partir del valor uno (1) para cada reinicio del servidor.

Parámetros	Descripción
<i>terminalId</i>	Identificador para la terminal.
<i>operationId</i>	identificador del tipo. Específico a cada dispositivo
SB3010 Brasil	0: Genera registros 60A, 60D, 60R, 60M, 75 mês actual; (Async) 1: Genera registros 60A, 60D, 60R, 60M, 75 mês anterior;(Asyn) 3: Deshabilita ordenación em la función AddProductEntry (Sync) 4: Habilita ordenación em la función AddProductEntry; (Sync)
BemaPos	5 : Inicializa un bloco de envío con tablas distintas 6 : Finaliza un bloco de envío con tablas distintas 7: Anula inicio de bloco de envío de tablas

Retorno

Valor	Descripción
> 0	Id único para la operación si pedido insertado con éxito
0	Servidor no inicializado
-3	Terminal inválido
-4	Operación no soportada

Evento asociado *OnOperationComplete*, *OnTransactionComplete* (cuando a operação for assíncrona).

Vea también Métodos *WaitForOperation* e *GetOperationStatus*.

Eventos

OnSalesArrive

Sintaxis `void OnSalesArrive (LONG terminalId);`

Comentario Este evento es ejecutado para cada log de venta recibido por el servidor. Se puede llamar la propiedad indexada *Cupones* con la *terminalId* pasado por el evento.

IMPORTANTE: Para el SB3010, los datos del log de ventas solamente son válidas para la versión 5.0 o superior del microterminal.

Parámetros	Descripción
<i>terminalId</i>	id del terminal que acaba de enviar el log de ventas

Vea también Propiedad *Cupones*.

OnTerminalUpdate

Sintaxis `void OnTerminalUpdate (LONG terminalId, SHORT table, BOOL failed);`

Comentario Este evento es ejecutado al término de las operaciones de inserción de tabla (*InsertTable*).

Parámetros	Descripción
<i>terminalId</i>	id del terminal target de la operación de inserción
<i>table</i>	tabla requisitada en la operación
<i>failed</i>	senaliza ERROR en la inserción si fuera TRUE

Vea también Método *InsertTable*.

OnTableReceived

Sintaxis void OnTableReceived (LONG terminalId, SHORT table, BOOL failed);

Comentario Este evento es ejecutado al termino de las operaciones de Lectura de la tabla (*ReadTable*). Los datos recibidos son salvos en archivo conforme la tabla dEscritura abajo.

Parámetros	Descripción
<i>terminalId</i>	id del terminal target de la operación de Lectura
<i>table</i>	tabla requisitada en la operación
<i>failed</i>	senaliza ERROR en la Lectura de la tabla si fuera TRUE

Nombres de los archivos creados:

Productos	"PRODUCTS_TABLE_[ID_TERMINAL].rec"
Clientes	"CLIENTS_TABLE_[ID_TERMINAL].rec"
Operadores	"%OPERATORS_TABLE_[ID_TERMINAL].rec"
Departamentos	"%DEPARTMENTS_TABLE_[ID_TERMINAL].rec"
Grupo Departamentos	"GRPDEPARTMENTS_TABLE_[ID_TERMINAL].rec"
Vendedores	"SALESMEN_TABLE_[ID_TERMINAL].rec"
Finalizadoras	"TENDERINGTYPES_TABLE_[ID_TERMINAL].rec"
Info Terminal	"TERMINALINFO_TABLE_[ID_TERMINAL].rec"
Mensajes	"MESSAGES_TABLE_[ID_TERMINAL].rec"
Ficha técnica	"ASSEMBLY_ITEMS_TABLE_[ID_TERMINAL].rec"
BackUp	"BEMA_BKUP_TABLE_[ID_TERMINAL].rec"
Stocks	"STOCK_TABLE_[ID_TERMINAL].rec"
Ivas	"TRIBUTES_TABLE_[ID_TERMINAL].rec"

Vea también Método *ReadTable*.

OnOperationComplete

Sintaxis void OnOperationComplete (LONG terminalId, LONG operationId, BOOL failed);

Comentario Este evento es ejecutado al termino de las operaciones.

Parámetros	Descripción
<i>terminalId</i>	id del terminal target de la operación
<i>operationId</i>	id de operación requisitada
<i>failed</i>	sinaliza ERROR en la operación si fuera TRUE

Vea también Método *DoGenericOperation*.

OnTransactionComplete

Sintaxis void OnTransactionComplete (LONG transationId, BOOL failed);

Comentario Este evento es ejecutado al termino de las operaciones *ReadTable*, *InsertTable*, *UpdateTable*, *DoGenericOperatio*.

Parámetros	Descripción
<i>transactionId</i>	Identificador para la transacción, obtenido en el retorno de las funciones <i>InsertTable</i> , <i>ReadTable</i> , <i>UpdateTable</i> ou <i>DoGenericOperation</i>
<i>failed</i>	sinaliza ERROR en la operación si fuera TRUE

Vea también Método *ReadTable*, *InsertTable*, *UpdateTable*, *DoGenericOperation*.

Interfaz ICoupons

Representa una colección de cupones contenida en el log de venta. Puede ser interada a través de un loop *FOR*, utilizando las propiedades dEscrituras abajo o, si utilizada en la plataforma Visual Basic, a través de un *FOREACH*.

Propiedades

Count

Tipo Lectura ;
Sintaxis LONG Count;
Comentario Esta propiedad indexada retorna el número de cupones en la colección.

Item

Tipo Lectura ;
Sintaxis ICoupon* Item (SHORT index);
Comentario Esta propiedad indexada retorna el cupón (*ICoupon*) especificado en el parámetro. Utilizada durante la iteración de la colección a través de un loop *FOR*. A colección es iniciada a partir del índice 1.

Parámetros	Descripción
<i>index</i>	de 1 hasta Count
Retorno	interfaz ICoupon representando un cupón individual de la colección

Interfaz ICoupon

Representa un cupón individual presente en el log de venta.

Propiedades

ClientId

Tipo	Lectura ;
Sintaxis	BSTR clientId;
Comentario	String con el id del cliente vinculado a ese cupón.

OperatorId

Tipo	Lectura ;
Sintaxis	LONG OperatorId;
Comentario	Identificador del operador que efectuó la venta.

SalesmanId

Tipo	Lectura ;
Sintaxis	LONG SalesmanId;
Comentario	Identificador del vendedor que efectuó la venta.

TerminalId

Tipo	Lectura ;
Sintaxis	LONG TerminalId;
Comentario	Identificador del terminal que efectuó la venta.

Date

Tipo	Lectura ;
Sintaxis	BSTR Date;
Comentario	String con la fecha del cupón en el formato <i>ddmmaaaa</i> .

Time

Tipo	Lectura ;
Sintaxis	BSTR Time;
Comentario	String con la hora del cupón en el formato <i>hhmm</i> .

Discount

Tipo	Lectura ;
Sintaxis	CURRENCY Discount;

API Aruak

Comentario Valor absoluto del descuento efectuado en el subtotal del cupón.

Increase

Tipo Lectura ;
Sintaxis CURRENCY Increase;
Comentario Valor absoluto del incremento efectuado en subtotal del cupón.

SubTotal

Tipo Lectura ;
Sintaxis CURRENCY SubTotal;
Comentario Subtotal del cupón.

Iva

Tipo Lectura ;
Sintaxis CURRENCY Iva;
Comentario Valor del Iva total del cupón no incluso en la venta (solamente para BemaPos con Impresora Fiscal)

Number

Tipo Lectura ;
Sintaxis LONG Number;
Comentario Número del cupón.

Status

Tipo Lectura ;
Sintaxis SHORT Status;
Comentario Status del cupón (1 – confirmado, 2 – cancelado).

TicketType

Tipo Lectura ;
Sintaxis SHORT TicketType;
Comentario Tipo del ticket (0 – comprobante de venta, 1- Fondo de caja, 2- Recebimiento Crédito, 3-Recebimiento Diversos, 4-Retirada, 5-Pago Diverso).

Items

Tipo Lectura ;
Sintaxis IRegisteredItems * Items;
Comentario Colección de itens registrados en el cupón

Payments

Tipo Lectura ;
Sintaxis IPayments * Payments;
Comentario Colección de pagos efectuados en el cupón

Interfaz *IRegisteredItems*

Representa una colección de itens registrados en el cupón. Puede ser iterada a través de un loop *FOR*, utilizando las propiedades dEcrituras abajo o, si utilizada en la plataforma Visual Basic, a través de un *FOREACH*.

Propiedades

Count

Tipo	Lectura ;
Sintaxis	LONG Count;
Comentario	Esta propiedad indexada retorna el número de itens en la colección.

Item

Tipo	Lectura ;
Sintaxis	IRegisteredItem* Item (SHORT index);
Comentario	Esta propiedad indexada retorna el ítem (IRegisteredItem) especificado en el parámetro. Utilizada durante la iteración de la colección a través de un loop <i>FOR</i> . La colección es iniciada a partir del índice 1.

Parámetros	Descripción
<i>index</i>	de 1 hasta Count
Retorno	interfaz IRegisteredItem representando un ítem de la colección

Interfaz *IRegisteredItem*

Representa un ítem individual registrado en el cupón.

Propiedades

Code

Tipo	Lectura ;
Sintaxis	BSTR Code;
Comentario	String con el código del ítem.

Description

Tipo	Lectura ;
Sintaxis	BSTR Description;
Comentario	String con la Descripción del ítem.

Tax

Tipo Lectura ;
 Sintaxis BSTR Tax;
 Comentario String con el valor percentual de la situación tributaria.

TaxType

Tipo Lectura ;
 Sintaxis SHORT TaxType;
 Comentario Identifica el tipo del tributo (ICMS=1, ISS=2, II=3, FF=4, NN=5, SI=6, SF=7, SN=8; cero para no aplicado)

Department

Tipo Lectura ;
 Sintaxis SHORT Department;
 Comentario Identifica el departamento al cual ese ítem está vinculado.

Quantity

Tipo Lectura ;
 Sintaxis LONG Quantity;
 Comentario Cantidad con 3 casas decimales.

Status

Tipo Lectura ;
 Sintaxis SHORT Status;
 Comentario Status del ítem (1 = confirmado, 2 = cancelado, 3 = devolución).

Discount

Tipo Lectura ;
 Sintaxis CURRENCY Discount;
 Comentario Valor absoluto del descuento efectuado en el ítem.

Increase

Tipo Lectura ;
 Sintaxis CURRENCY Increase;
 Comentario Valor absoluto del incremento efectuado en el ítem.

UnitValue

Tipo Lectura ;
 Sintaxis CURRENCY UnitValue;
 Comentario Valor unitario del ítem.

TotalValue

Tipo Lectura ;

Sintaxis CURRENCY TotalValue;
 Comentario Valor total del ítem (cantidad + incrementos – descuentos).

Interfaz IPayments

Representa una colección de pagos efectuados en el cupón. Puede ser iterada a través de un loop *FOR*, utilizando las propiedades dEscrituras abajo o, si utilizada en la plataforma Visual Basic, a través de un *FOREACH*.

Propiedades

Count

Tipo Lectura ;
 Sintaxis LONG Count;
 Comentario Esta propiedad indexada retorna el número de ítems en la colección.

Item

Tipo Lectura ;
 Sintaxis IPayment* Item (SHORT index);
 Comentario Esta propiedad indexada retorna el pago (IPayment) especificado en el parámetro. Utilizada durante la iteración de la colección a través de un loop *FOR*. La colección es iniciada a partir del índice 1.

Parámetros	Descripción
<i>index</i>	de 1 hasta Count
Retorno	interfaz IPayment representando un pago de la colección

Interfaz IPayment

Representa un pago individual efectuado en el cupón.

Propiedades

Index

Tipo Lectura ;
 Sintaxis SHORT Index;
 Comentario Índice de la forma de pago.

Amount

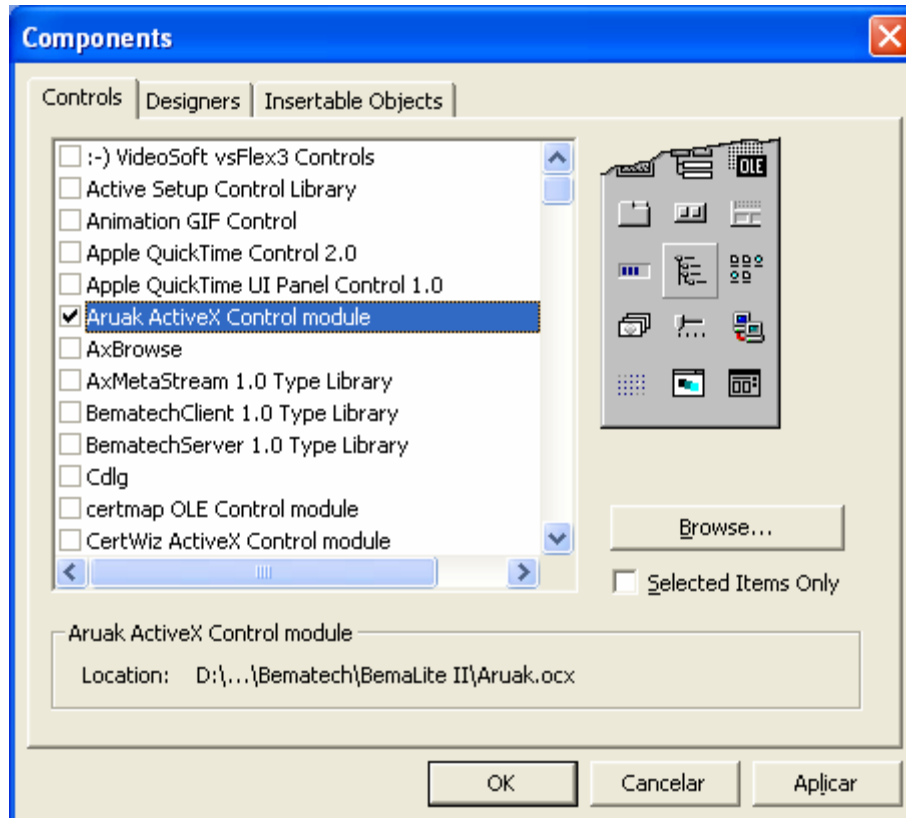
Tipo Lectura ;
 Sintaxis CURRENCY Amount;
 Comentario Valor del pago efectuado.

Ejemplos de Programación

El primer paso es registrar el control activeX a través del comando `regsvr32 aruak.ocx`. El dll `bemasb32.dll` debe estar contenido en la misma carpeta donde se localiza el control, pero esta no necesita ser registrada.

Visual Basic

Seleccione el control a través del menú *Project->Components* en el Visual Basic.



Después de seleccionarlo, el control estará disponible en la caja de herramientas, listo para ser utilizado, bastando para eso arrastrarlo a su *Form* principal.

1 -Inicialización

'En el Load del form o en otra función de inicialización cualquier
'configure y inicialice el componente

```
Private Sub Form_Load()
...
    Aruak1.Family = 2 'PDV SB3010
    Aruak1.StartServer '
...
End Sub
```

API Aruak

2 - Configurar los terminales

`Despues de iniciar el componente, añáde los terminales que serán
`supervisionados

```
Private Sub CommandAddTerm_Click()  
...  
    id = 1  
    ip = "10.10.100.110"  
    Aruak1.AddTerminal id, ip  
...  
End Sub
```

3 - Trabajar con el log de ventas

`Añáde un *handler* para el evento OnSalesArrive que es disparado por
`el componente cuando es enviado el log de ventas

```
Private Sub Aruak1_OnSalesArrive(ByVal TerminalId As Long)
```

```
    Dim coupons As AruakLib.coupons  
    Dim coupon As AruakLib.coupon  
    Dim Item As AruakLib.RegisteredItem  
  
    Set coupons = Aruak1.coupons(TerminalId)  
    For Each coupon In coupons  
        ...  
        `coupon.Number  
        `coupon.OperatorId  
        `coupon.Time  
        `coupon.Discount  
        `coupon.Subtotal  
        `coupon.TerminalId  
        ...  
        For Each Item In coupon.Items  
            ...  
            `item.description  
            `item.Code  
            `item.Status  
            `item.Discount  
            ...  
        Next Item  
        For Each Payment In coupon.Payments  
            ...  
            `Payment.Index  
            `Payment.amount  
            ...  
        Next Payment  
    Next coupon  
    ...  
End Sub
```

4 - Enviando tablas al PDV

`Los Departamentos

```
Private Sub InsertDepartments()  
Dim codigo as long  
Dim description As String  
Dim tax As Long  
  
` configura update para tabla de departamentos (4)  
FormPrincipal.Aruak1.InitializeUpdate (4)  
  
Adodcl.Recordset.MoveFirst  
`Busca todos los departamentos de un recordSet cualquier
```

API Aruak

```
'Agrega los registros para cada departamento existente
Do While Adodcl.Recordset.EOF = False
  codigo = Adodcl.Recordset("DEP_CODIGO")
  description = Adodcl.Recordset("DEP_DESCR")
  tax = Adodcl.Recordset("TRIBUTE")

  ret = FormPrincipal.Aruak1.AddDepartmentEntry(plu, description, tax)
  If ret <> 1 Then
    MsgBox ("Erro no addentry: " + CStr(ret) + "= " + description)
    Exit Do
  End If
  Adodcl.Recordset.MoveNext
Loop
'dispara inserción de la tabla de departamentos (4) para la terminal (1)
'el retorno de la función es un Id único = pedido asíncrono
DepartmentsTransactionId = Aruak1.InsertTable(1, 4)
End Sub

5 - Tratando respostas de los requests (InsertTable ou ReadTable)
'Añade un Handler para el evento OnTransactionComplete
\
Private Sub Aruak1_OnTransactionComplete(ByVal transactionId As Long, ByVal
failed As Boolean)
...

If failed Then
  MsgBox ("Transaction :" + CStr(transactionId) + " failed ")
Else
  MsgBox ("Transaction :" + CStr(transactionId) + " succeeded ")
End If
If transactionId = DepartmentsTransactionsId then
  ...
  'Actualiza status del request de envio de la tabla de departamentos
  ...
End if
End Sub
```

Anexo I

Formato de los archivos generados pela API Aruak para cada tabla leída del PDV.

Archivos do BemaPos

Productos

Cada registro es compuesto pelos campos descritos abajo:

Código	-	07	BCD*
Grupo	-	01	BCD
Depto	-	01	BCD
Nombre	-	20	(ASC/alfa)
Num. Iva	-	02	(ASC/NUMERICO)
Unidad	-	02	(ASC/alfa)
Porc. Comis	-	02	BCD
Num. Pedido Rem	-	02	(ASC/NUMERICO)
Precio de Costo	-	08	BCD
Margen	-	03	BCD*****
Precio de Venta	-	08	BCD
Flags	-	02	INT**
SALDO ACTUAL	-	04	BCD (****)
SALDO MÍNIMO	-	04	BCD (****)
SALDO IDEAL	-	04	BCD (****)

(*) Son 13 dígitos que pueden ser EAN13

No es utilizado cálculo del dígito verificador, cero a la izquierda para el dígito 14.

(**) (LSB+MSB) Los bits indican:

Tipo unidad medida (exclusivo) ***:

12	UN
11	LT
10	MT
9	KG
8	PC

Combinación posible:

7	Precio Fijo
6	Producto Mixto (tecla de pedido de precio habilitada)
5	Producto Compuesto
4	Pedido Remoto
3	Producto Comisionado
2	Cupón Unitario
1	Controla estoque.

(****) Todos saldos con 3 casas decimales

(*****) Margen del producto até 999,99%, 3 bytes BCDs en la forma 0XXX,XX

Cientes

API Aruak

Cada registro es compuesto pelos campos descritos abajo:

Código	-	04	(ASC/NUMERICO)
Nombre	-	40	(ASC/alfa)
Dirección	-	40	(ASC/alfa)
Complemento	-	40	(ASC/alfa)
Telefone1	-	10	(ASC/alfa)
Telefone2	-	10	(ASC/alfa)
Contacto	-	20	(ASC/alfa)
(CPF/NIF/RIF)	-	15	(ASC/alfa)
Saldo deuda	-	08	BCD
Limite Crédito-	-	08	BCD
Email	-	40	(ASC/alfa)
Flag de LC	-	01	ASC***
Flag de id	-	01	ASC****

(***) 0 – No permite la venta a crédito a ese cliente; 1 – venta a crédito permitida hasta el limite especificado, y 2–venta a crédito sin limite.

(****) 1 –Cliente, 2 – proveedor, 3 – funcionario.

Operadores

Hasta 99 registros. Cada registro es compuesto pelos campos descritos abajo:

Código	2	(ASC/NUMERICO)
Nombre	15	(ASC/alfa)
Clave	6	(ASC/alfa)
Acceso	2	(ASC/NUMERICO)*

(*) 01-Caja, 02-Supervisor, 03-Gerente, 04-Director, 05-Propietario.

Vendedores

Hasta 99 registros. Cada registro es compuesto pelos campos descritos abajo:

Código	2	(ASC/NUMERICO)
Nombre	15	(ASC/alfa)
Comisión	2	BCD (*)

(*) Porcentual de Comisión (NN,NN%)

Grupos de departamentos (o departamentos)

Hasta 99 registros. Cada registro es compuesto pelos campos descritos abajo:

Código	2	(ASC/NUMERICO)
Nombre	15	(ASC/alfa)

Departamentos (ou sectores)

Hasta 99 registros. Cada registro es compuesto pelos campos descritos abajo:

Código	2	(ASC/NUMERICO)
Nombre	15	(ASC/alfa)
Num. Iva	2	(ASC/NUMERICO)
Id do grupo	2	(ASC/NUMERICO)

Mensajes

Cada registro es compuesto pelos campos descritos abajo:

Texto	48	(ASC/alfa)
Tipo	1	(ASC/NUMERICO)*
Flags	02	INT**

* - '0' para encabezado e '1' para pie de página

** - mapa de bits con la combinación de las fuentes posibles: (LSB +MSB)

FORMATOS CHARACTER (combinación)

TIPO	MAPA
Expandido	0x01
Itálico	0x02
Negrita	0x04

TAMAÑOS CHARACTER (exclusivo)

TIPO	MAPA
Condensado	0x10
Normal	0x20

Até 5 encabezados e 3 pies de página.

Ficha técnica

Registros de tamaño variable que describen los productos compuestos, segundo el formato abajo:

PLU	7	BCD
Nuns componentes	1	INT
Componente1	7	BCD
Quant1	4	BCD
...		
Componente15	7	BCD
Quant15	4	BCD

Un producto pode ser compuesto por hasta 15 otros PLUs.

<7 bytes con código producto> + <1 byte con el numero de productos agregados> + <N pacotes con 11 bytes>, donde 0 <= N <= 15

Stocks

Cada registro es compuesto pelos campos descritos abajo:

Nombre	Tam.	Tipo
Grupo	-	01 bcd
Depto	-	01 bcd
Código	-	07 BCD*
SALDO ACTUAL	-	04 BCD(****)
SALDO MÍNIMO	-	04 BCD(****)
SALDO IDEAL	-	04 BCD(****)

Archivos del SB3010

Productos

Cada registro es compuesto pelos campos descritos abajo:

06 bytes	-	Código do PLU (BCD)
20 bytes	-	Descripción do PLU (ALPHA)
04 bytes	-	Precio (BCD)
01 byte	-	índice IVA (BCD)
01 byte	-	Departamento (BCD) - número
06 bytes	-	Cantidad Vendida
06 bytes	-	Cantidad Vendida R60 *
06 bytes	-	Venta Bruta R60 *
06 bytes	-	Venta Líquida R60 *
06 bytes	-	Cantidad Vendida R60 *- Mes Anterior
06 bytes	-	Venta Bruta R60 *- Mes Anterior
06 bytes	-	Venta Líquida R60 *- Mes Anterior
04 bytes	-	Estoque (BCD)
01 byte	-	Flags (Binario)
01 byte	-	Grupo de Teclas (ALPHA)
06 bytes	-	Venta Líquida

El byte Flags es compuesto de:

Bit_0 (0x01):	Flag Pesado
Bit_1(0x02):	No utilizado
Bit_2(0x04):	No utilizado
Bit_3(0x08):	Flag Litrazo
Bit_4(0x10):	No utilizado
Bit_5(0x20):	No utilizado
Bit_6(0x40):	Flag tres casas decimales
Bit_7(0x80):	No utilizado

Clientes

Hasta 200 registros. Cada registro es compuesto pelos campos descritos abajo:

02 bytes	-	Código (BCD)
07 bytes	-	RIF/NIF/ID (BCD)
30 bytes	-	Nombre (ALPHA)
40 bytes	-	Dirección (ALPHA)
40 bytes	-	Complemento Dirección (ALPHA)
01 byte	-	Status
08 bytes	-	Espacios en blanco

Operadores

Hasta 20 registros. Cada registro es compuesto pelos campos descritos abajo:

04 bytes	-	Código (ASC/NUMERICO)
15 bytes	-	Nombre (ASC/ALPHA)
04 bytes	-	Clave (ASC/NUMERICO)
16 bytes	-	Permisos* (ASC/ALPHA)

API Aruak

*String con 16 caracteres "S" o "N"; cada un configura una permisión:

- 1° - Descuento en ítem
- 2° - Descuento en sub. Total
- 3° - Anulación de ítem
- 4° - Anulación de cupón
- 5° - lectura x
- 6° - Reducción Z
- 7° - Sangría
- 8° - Fondo de caja
- 9° - Limite
- 10° - Informes
- 11° - Incremento en ítem
- 12° - Incremento en subtotal.
- 13° - Perfil Gerente
- 14° - Reservado
- 15° - Reservado
- 16° - Reservado

Vendedores

Hasta 20 registros. Cada registro es compuesto pelos campos descritos abajo:

04 bytes	-	Código (ASC/NUMERICO)
15 bytes	-	Nombre (ASC/ALPHA)

Grupos de departamentos (o departamentos)

Hasta 20 registros. Cada registro es compuesto pelos campos descritos abajo:

02 bytes	-	Número (ASC/NUMERICO)
20 bytes	-	Nombre (ASC/ALPHA)

Departamentos (ou sectores)

Hasta 99 registros. Cada registro es compuesto pelos campos descritos abajo:

02 bytes	-	Número (ASC/NUMERICO)
02 bytes	-	Número do grupo (ASC/NUMERICO)
15 bytes	-	Nombre (ASC/ALPHA)

Formas de pago

Até 20 registros. Cada registro é compuesto pelos campos descritos abajo:

20 bytes	-	Nombre (ASC/ALPHA)
----------	---	--------------------

Info Terminal

Un registro compuesto pelos campos descritos abajo:

20 bytes	-	Numero serial (ASC/ALPHA)
05 bytes	-	Versión del software (ASC/ALPHA)