Trazabilidad de Productos Médicos - WS TrazaProdMed ANMAT/PAMI SNT

- **Documentación Componentes y Herramientas Generales:**
  - PyAfipWs: Interfase simil OCX con otros lenguajes (VB, VFP, Cobol ...) Costos y Condiciones
  - Manual: Documentación, Información Importante, Certificados, Errores Frecuentes
  - Herramienta "universal": archivos de intercambio TXT Cobol, DBF dBase/FoxPro, JSON PHP/Java
  - PyFEPDF: Generador de Factura Electrónica en formato PDF
- **Factura Electrónica - Servicios Web AFIP:**
  - Mercado Interno: Factura Electrónica A/B/C/M WSFEv1+ (RG2485/3067/3668/3749/4004/4109/4367)
  - Matrix (codificación productos): Factura Electrónica A/B con detalle (RG2904/3536)
  - Bienes de Capital: Bonos Fiscales Electrónicos - Factura Electrónica A (RG2557)
  - Exportación: Factura Electrónica E Exportadores (RG2758 RG3689 RG4401)
  - Turismo: Comprobantes Factura Electrónica T WSCT CAE/CAEA (RG3971)
  - FE Cred (FCE): Factura Electrónica Crédito MiPyMEs (RG4367)
  - Código de Autorización Electrónico Anticipado CAEA
- **Agropecuario - Servicios web AFIP:**
  - Código Trazabilidad de Granos: Transporte de granos WSCTGv4 (RG2806 RG3113 RG3493)
  - Liquidación y Certificación de Granos: WSLPGv1.17 F. C1116 A / B / RT (RG3419 RG3690 RG3691)
  - Liquidación de Tabaco Verde: WSLTVv1.3
  - Liquidación Única Mensual Lechería: WSLUMv1.3
  - Liquidación Sector Pecuario: Hacienda, Compra directa, Carne WSLSPv1.3
  - Remito Electrónico Cárnico: WSRemCarne (RG4256/18 y RG4303/18)
  - Remito Electrónica Harinero: WSRemHarina - (RG 4519/19)
  - Remito Electrónico Azúcar y Derivados: WSRemAzucar - (RG 4519/19)
  - Carta de Porte: WSCPE - (RG 5017/21)
  - Carta de Porte Derivados Granarios: WSCPEv2 - (RG 5235/22) **Nuevo!**
- **Otros webservices y utilidades AFIP**
  - Constatación de Comprobantes CAI, CAE, CAEA (WSCDC)
  - Padron Contribuyentes AFIP: Constancia de Inscripción RG1817/2005 WS-SR-Padron
  - Consulta de Operaciones Cambiarias: Compra de Divisas (WSCOC)
  - SIRE_CertificadoRetencionElectronica: Certificado de retención electrónica del Impuesto al Valor Agregado (SIRE-WS) RG4523/19
- **Webservices provinciales: ARBA (Prov. Bs.As.), AGIP (C.A.B.A), API (Sta.Fe), DGR (Córdoba):**
  - Remito Electrónico: COT Código de Operaciones de Translado (ARBA, API, AGIP, DGR)
  - Ingresos Brutos: Consulta de alícuotas WS DFE IIBB ARBA
- **SNT: Sistema Nacional de Trazabilidad ANMAT, SEDRONAR, SENASA**
  - Trazabilidad de Medicamentos: ANMAT Disposición 3683/2011
  - Trazabilidad de Productos Médicos: ANMAT Disposición 2303/2014 y 2175/14
  - Trazabilidad de Precursores Químicos: RENPRE SEDRONAR Resolución 900/12
  - Trazabilidad de Productos Fitosanitarios: SENASA Resolución 369/13
  - Trazabilidad de Productos Fitosanitarios: SENASA Resolución 369/21 (WS_INFO_EMPRESAS / WS_DATOS_GENERALES)

- **Aplicativos Genéricos y Herramientas Avanzadas:**
    - ♦ PyRece: Aplicativo visual simil SIAP - RECE (CSV, PDF, Email)
    - ♦ FE.py: Herramienta universal, unificada e integrada
    - ♦ FacturaLibre: Aplicacion online (web2py)
    - ♦ PyFactura Aplicativo visual y simple (gui2py) para CAE y PDF factura electrónica
    - ♦ LibPyAfipWs: Biblioteca DLL para lenguajes C / C++ y similares
    - ♦ Factura Electrónica en Python: Información Técnica (SOAP, XML, PDF, DBF, etc.)

Interfaz para Servicio Web Código de Trazabilidad de Productos Médicos (SOAP) correspondiente a la Disposición Nº 2303/2014 y Disposición Nº 2175/14 del A.N.M.A.T. que deberán implementar las personas físicas o jurídicas que intervengan en la cadena de distribución, dispensación y aplicación de productos médicos registrados ante la Administración Nacional de Medicamentos, en los términos establecidos en el artículo 1º y siguientes de la Resolución del Ministerio de Salud Nº 2175/2013. B.O. 23 de abril de 2014.. SNT Especificación Técnica.

Índice

Introducción

Biblioteca para el web service de ANMAT/PAMI que permite automatizar la gestión de las trazabilidad de productos médicos:

- Interfaz COM: Simil DLL/OCX, embebible para aplicaciones programadas en lenguajes visuales bajo windows (Visual Basic, Visual Fox Pro, SAP, etc.)
- Interfaz por consola (linea de comandos / archivo de texto): similar a aplicativos SIAP para sistemas legados (ej. RM COBOL) multiplataforma (DOS, Windows, Unix)
- Interfaz por tablas DBF: compatible con dBase, FoxPro, Clipper -proximamente-
- Interfaz por base de datos: compatible con conectores ODBC (MS SQL Server), PostgreSQL y otros (Oracle, DB2) -proximamente-

Cubre totalmente el proceso, puede ser adaptado a programas existentes y no es requerido intervención del usuario.

Para más información ver PyAfipWs

Descargas

- Instalador: PyAfipWs-2.7.1920-32bit+trazaprodmed_1.01b-homo.exe (homologación versión inicial al 24/12/2015)
- Documentación Oficial: especificación técnica original (17/09/2015), set de datos (no disponible)
- Ejemplo en VB: TrazaProdMed.bas (Visual Basic "Classic")
- Ejemplo en VFP: TrazaProdMed.prg (Visual Fox Pro)
- Código Fuente (Python): trazaprodmed.py

Agentes de Prueba: Para buscar agentes de prueba se puede navegar el catálogo electrónico de GLNs/CUFEs?del entorno de entrenamiento, a través del siguiente link:
https://servicios.pami.org.ar/trazaenprodmed/consultaCatal.tz

Productos de Prueba: Para buscar productos de prueba se puede navegar el catálogo electrónico de GTINs del entorno de entrenamiento, a través del siguiente link:
https://servicios.pami.org.ar/trazaenprodmed/consultaCatalogoByGTIN.tz

URL:

- Entrenamiento: https://servicios.pami.org.ar/trazaenprodmed.WebService (WSDL)
- Producción: https://servicios.pami.org.ar/trazaprodmed.WebService (WSDL)

Métodos

- **`CrearTransaccion(f_evento, h_evento, gln_origen, gln_destino, n_remito, n_factura, vencimiento, gtin, lote, numero_serial, id_evento, cuit_medico, id_obra_social, apellido, nombres, tipo_documento, n_documento, sexo, calle, numero, piso, depto, localidad, provincia, n_postal, fecha_nacimiento, telefono, nro_afiliado, cod_diagnostico, cod_hiv, id_motivo_devolucion, otro_motivo_devolucion)`**: Crea internamente el registro de una transacción de Productos Médicos. Hasta id_evento son campos obligatorios (en general). Ver `SetParametro` para establecerlos antes de llamar a este método.
- **`InformarProducto(usuario, password)`**: Realiza el registro de una transacción de Productos Médicos
- **`SendCancelacTransacc(usuario, password, codigo_transaccion)`**: Realiza la cancelación de una transacción
- **`SendCancelacTransaccParcial(usuario, password, codigo_transaccion, gtin, numero_serial)`**: Realiza la cancelación parcial de una transacción, gtin_medicamento y numero serial son parámetros opcionales.
- **`GetTransaccionesWS(usuario, password, id_transaccion, gln_agente_origen, gln_agente_destino, gtin, lote, serie, id_evento, fecha_desde_op, fecha_hasta_op, fecha_desde_t, fecha_hasta_t, fecha_desde_v, fecha_hasta_v, n_remito, n_factura, id_provincia, id_estado, nro_pag=1, offset=100)`**: Obtiene los movimientos realizados y permite filtros de búsqueda
- **`GetCatalogoElectronicoByGTIN(usuario, password, gtin, gln, marca, modelo, cuit, id_nombre_generico, nro_pag=1, offset=100)`**: Obtiene el Catálogo Electrónico de Productos Médicos

Métodos generales / accesorios

- **`Conectar(cache, url_webservice_wsdl, proxy, wrapper, cacert)`**: Establece la conexión con el servidor remoto, recibe el directorio de archivos temporales, la URL del WSDL (descripción del webservice) y proxy en formato 'usuario:clave@servidor:puerto'. Si no se especifíca url, se utiliza servidores de homologación. Parametros adicionales optativos: wrapper es la librería HTTP a utilizar y cacert la ruta al certificado de la autoridad de certificante del servidor (CA)
- **`LeerError()`**: devuelve el error notificado por el webservice, o vacio si no hay más errores para informar.
- **`LeerTransaccion()`**: obtiene la próxima transacción a devuelta por el webservice al consultar (luego llamar a GetParametro para obtener los valores de cada campo), o enviada vía InformarProducto. Devuelve False si no hay más transacciones. NOTA: Utilizar este método luego de informar productos, para chequear y limpiar las transacciones procesadas (por ej., si es necesario

reutilizar el objeto para enviar nuevas transacciones).

- **`SetParametro(clave, valor)`**: establece el parámetro de entrada de nombre "clave" con el respectivo "valor", que será usado al llamar al webservice (útil para lenguajes de programación que solo soportan 27 parámetros como VFP)
- **`GetParametro(clave)`**: devuelve el parámetro de salida de nombre "clave", que fue establecido al llamar un método del webservice (útil para recorrer las transacciones devueltas de las consultas, previa llamada a `LeerTransaccion`)

Los métodos devuelven `True` (verdadero) en caso de que se ejecuten correctamente, y establecen los atributos `Resultado`, `Errores` y `CodigoTransaccion` según corresponda.

En caso de falla (por ej. error de comunicación), devuelven `False` (falso) y se debe revisar atributos `Excepcion` y `Traceback`

Atributos

- `Username`, `Password`: credenciales de seguridad para operar el webservice
- `CodigoTransaccion`: número otorgado por el método remoto (en caso correcto)
- `Errores`: lista de validaciones fallidas devueltas por el método remoto
- `Resultado`: verdadero (`True`) si fue procesado correctamente
- `TransaccionPlainWS`: lista de diccionarios (array clave->valor) de parámetros de salida de `GetTransaccionesNoConfirmadas` (consulta de transacciones)
- `XmlRequest`, `XmlResponse`: mensajes xml crudos sin procesar (para depuración)
- `Version`, `InstallDir`: datos para depuración
- `Traceback`, `Excepcion`: errores no esperados de comunicación o similar (por ej. SoapFault)

Campos para Informar Productos Médicos

El método SendProductos Médicos recibe los datos correspondientes a una transacción de Productos Médicos:

Nombre	Tipo	Longitud	Obligatorio	Descripción
f_evento	alfanumérico	10	si	Fecha en que ocurre el evento. Formato DD/MM/YYYY
h_evento	alfanumérico	5	si	Hora en la que ocurre el evento. Formato HH:MM
gln_origen	alfanumérico	13	si	Código GLN del agente origen.
gln_destino	alfanumérico	13	no	Código GLN del agente destino
cuit_destino	alfanumérico	11	no	Número de CUIT del agente destino. Numérico sin guiones.
n_remito	alfanumérico	20	no	Número de Remito
n_factura	alfanumérico	20	no	Número de Factura
vencimiento	alfanumérico	10	si	Fecha de Vencimiento del medicamento. Formato DD/MM/YYYY
gtin	alfanumérico	14	si	GTIN del medicamento
lote	alfanumérico	20	si	Número de lote
numero_serial	alfanumérico	20	si	Número de serie
id_evento	numerico	2	si	

				Identificador del evento. Ver tabla de Eventos.
cuit_medico	alfanumérico	11	si	Número de CUIT del Médico interviniente en la implantación. Numérico sin guiones.
apellido	alfanumérico	50	no	Apellido del paciente.
nombres	alfanumérico	100	no	Nombre /s del paciente
n_documento	alfanumérico	10	no	Número de Documento del paciente
sexo	alfanumérico	1	no	Sexo de la persona del paciente
tipo_documento	numerico		no	Tipo de Documento del paciente
calle	alfanumérico	100	no	Domicilio del paciente
localidad	alfanumérico	50	no	Localidad del paciente
numero	alfanumérico	10	no	Numero de calle del paciente
piso	alfanumérico	5	no	Piso del departamento del paciente
depto	alfanumérico	5	no	Departamento del paciente
n_postal	alfanumérico	8	no	Código postal del paciente
telefono	alfanumérico	30	no	Número de teléfono del paciente
id_obra_social	numerico	9	no	Numero de obra social que financia el producto que se implanta al paciente. Ver tabla de obras sociales en documento set_de_datos.pdf
nro_asociado	alfanumérico	30	no	Número de afiliado a la obra social que financia el producto que se implanta al paciente.
id_motivo_devolucion	numérico		no	Opcional. Enviar uno de los posibles valores: 1-No solicitado; 2-Producto no utilizado; 3-Producto próximo a vencer; 4-Producto retirado del mercado; 5-No coincide con la documentación fiscal remitida; 6-Otros; 7-Producto recibido sin cadena de frio,cuando sí lo requiere;
otro_motivo_devolucion	alfanumérico		no	Descripción del otro motivo de Solamente obligatorio devolución, solamente en caso de que el si se especifica el campo anterior se envíe como ?Otros?

Importante: si un campo no debe enviarse, completar con None, Null, Empty, ? o el valor equivalente para campos nulos. No usar cadena vacia o 0 ya que son datos válidos y se enviarán al webservice. No saltear u omitir parámetros, salvo que esten al final o sean pasados por nombre (keyword arguments), usar SetParametro en dicho caso.

Campos devueltos al consultar - !TransaccionDTO

El método !GetTransaccionesWS devuelve la lista de transacciones que coincicidan con los criterios de búsqueda especificados. Se puede acceder al atributo Transacciones para leer estos datos (diccionario clave - valor, dependiendo del lenguaje de programación). Es posible recorrer dichas transacciones una por una usando el método LeerTransaccion (devuelve True y establece los parámetros de salida, que pueden ser consultados con GetParametro?).

Para el listado de transacciones no confirmadas, los parámetros de salida definidos en la documentación de ANMAT son los siguientes:

- `idTransaccionGlobal`: Numero Global
- `fEvento`: Fecha del evento
- `razonSocialInformador`: Agente Informador
- `razonSocialOrigen`
- `glnDestino`: GLN Destino
- `razonSocialDestino`: Razon Social Destino
- `glnOrigen`: GLN Origen
- `nroRemito`: Numero de Remito
- `n_factura`: Numero de Factura
- `vencimiento`: Fecha de vencimiento de transaccion
- `gtin`: GTIN del producto médico
- `descProducto`: descripción del producto médico
- `lote`: Número de Lote
- `nroSerial`: Numero Serial
- `id_evento`: Numero de Evento (reestablecido pero no documentado por ANMAT al 03-02-2014)
- `dEvento`: Descripción del Evento
- `fTransaccion`: Fecha Transaccion*
- `idEstado`: Estado de la transaccion
- `descEstado`: Descripción del Estado de la transacción
- `idMotivoDevolucion`: Id Motivo Devolución

Importante: en la respuesta devuelta por ANMAT, pueden no estar todos los parámetros presentes (depende de la transacción). Si no se encuentra el parámetro, GetParametro devolverá nulo.

Los campos tachados fueron incluidos en la documentación preliminar pero omitidos por ANMAT en la versión definitiva.

Linea de Comandos

Para sistemas operativos legados (DOS bajo windows) y UNIX/Linux, es posible operar la herramienta de trazabilidad por consola. Recibe como parámetros los datos correspondientes a la llamada remota (ver métodos). Opcionalmente se puede especificar --testing para pruebas (usar xml de muestra como respuesta si no se tiene acceso a homologación) y --trace para imprimir por pantalla los datos enviados y recibidos.

Nota: dependiendo de la compilación y el instalador, el ejecutable puede ser **`TrazaProdMed_CLI.EXE`**

Ejemplo de uso para Informar un Medicamento (por defecto recibe los argumentos según el método SendMedicamento):

```
234" "A123400001234" "31/03/2015" "GTIN1" "2015" "12348" 134 "20267565393" "9999999999999" "20267565393" "Reinc
```

Si no se específican los campos por linea de comando, de manera predeterminada se utilizará el formato de texto

Para usar Tablas DBF agregar a la linea de comandos el parámetro `--dbf`. Para soporte de JSON

(JavaScript? Object Notation), agregar parametro `--json`. Se debe anteponer `--cargar` y `--grabar` para leer y escribir los datos en los archivos de intercambio.

Ejemplos para informar Productos Médicos usando archivos de texto, dbf y json respectivamente (*próximamente*):

```
C:\PYANMAT>TrazaProdMed.exe --cargar --grabar "pruebasws" "pruebasws"
C:\PYANMAT>TrazaProdMed.exe --cargar --grabar --dbf "pruebasws" "pruebasws"
C:\PYANMAT>TrazaProdMed.exe --cargar --grabar --json "pruebasws" "pruebasws"
```

Ejemplo de uso para cancelar una transacción (recibe usuario, password y número de transacción):

```
C:\PYANMAT>TrazaProdMed.exe --cancela "pruebasws" "pruebasws" "5114801"
|Resultado False|CodigoTransaccion       None|Errores|3: Transaccion NO encontrada, NO se puede anular.|
```

Ejemplo de uso para cancelar una transacción parcialmente (recibe usuario, password, número de transacción, gtin y numero de serie -estos dos últimos opcionales-):

```
C:\PYANMAT>TrazaProdMed.exe --cancela_parcial  pruebasws pruebasws 2224635 000000000GTIN2 998
|Resultado False|CodigoTransaccion       None|Errores|3: Transaccion NO encontrada, NO se puede anular.|
```

Ejemplo de uso para Consultar transacciones en ANMAT:

```
oSerial||descEstado||nroRemito||glnDestino||fTransaccion||idEstado||dEvento||descProducto||gtin||idTransaccion
0:00 || A1234 || Cargada || R0001-00000001 || 07791234567810 || 2016-03-10 12:36:07 || 5 || RECEPCIÓN DE PRODU
```

Para generar un archivo de intercambio con los datos de las transacciones, especificar `--grabar` (*proximamente*):

```
C:\PYANMAT>TrazaProdMed.exe --grabar --consulta "pruebasws" "pruebasws"
```

Archivos de Intercambio

La herramienta podrá soportar archivos de intercambio en DBF, JSON o archivos de texto fijo (COBOL). Consultar.

Ejemplos

Intefase COM en VB (5/6)

Envio de Productos Médicos (transacción de trazabilidad):

```
Dim TrazaProdMed As Object, ok As Variant

' Crear la interfaz COM
Set TrazaProdMed = CreateObject("TrazaProdMed")

' Establecer credenciales de seguridad
TrazaProdMed.Username = "testwservice"
TrazaProdMed.Password = "testwservicepsw"
```

```
' Conectar al servidor (pruebas)
ok = TrazaProdMed.Conectar()

' datos de prueba
usuario = "pruebasws"
password = "pruebasws"
f_evento = CStr(Date)            ' ej: "25/11/2011"
h_evento = Left(CStr(Time()), 5) ' ej:  "04:24"
gln_origen = "7791234567801"     ' Laboratorio
gln_destino = "7791234567801"    ' LABORATORIO (asociado al medicamento)
n_remito = "R0001-12341234"       ' formato 13 digitos!
n_factura = "A0001-12341234"      ' formato 13 digitos!
vencimiento = CStr(Date + 30)    ' ej. "27/03/2013"
gtin = "07791234567810"          ' código de medicamento de prueba
lote = Year(Date)                ' uso el año como número de lote
numero_serial = CDec(CDbl(Now()) * 86400)   ' número unico basado en la fecha
id_obra_social = "465667"
id_evento = 1  '
cuit_medico = "30711622507"
apellido = "Reingart": nombres = "Mariano"
tipo_docmento = "96": n_documento = "26756539": sexo = "M"
calle = "Saraza": numero = "1234": piso = "": depto = ""
localidad = "Hurlingham": provincia = "Buenos Aires"
n_postal = "1688": fecha_nacimiento = "01/01/2000"
telefono = "5555-5555"
nro_afiliado = "9999999999999"
cod_diagnostico = "B30"
cod_hiv = "NOAP31121970"
id_motivo_devolucion = 1
otro_motivo_devolucion = "producto fallado"

' Agregar Producto a Trazar:
ok = TrazaProdMed.CrearTransaccion( _
                    f_evento, h_evento, gln_origen, gln_destino, _
                    n_remito, n_factura, vencimiento, gtin, lote, _
                    numero_serial, id_evento, _
                    cuit_medico, id_obra_social, apellido, nombres, _
                    tipo_documento, n_documento, sexo, _
                    calle, numero, piso, depto, localidad, _
                    provincia, n_postal, fecha_nacimiento, telefono, _
                    nro_afiliado, cod_diagnostico, cod_hiv, _
                    id_motivo_devolucion, otro_motivo_devolucion)

' Enviar datos y procesar la respuesta:
ok = TrazaProdMed.InformarProducto(usuario, Password)

Debug.Print "Resultado:", TrazaProdMed.Resultado
Debug.Print "CodigoTransaccion:", TrazaProdMed.CodigoTransaccion

For Each er In TrazaProdMed.Errores
     MsgBox er, vbExclamation, "Error!"
Next

MsgBox "Resultado: " & TrazaProdMed.Resultado & vbCrLf & _
        "CodigoTransaccion: " & TrazaProdMed.CodigoTransaccion, _
        vbInformation, "Resultado"


' NOTA: extraer las transacciones enviadas (para poder enviar nuevas)
Do While TrazaProdMed.LeerTransaccion
   Debug.Print TrazaProdMed.GetParametro("nroSerial")
```

```
ENDDO
```

Ejemplo para anular una transacción de trazabilidad de Productos Médicos:

```
' Cancelo la transacción (anulación):
codigo_transaccion = TrazaProdMed.CodigoTransaccion
ok = TrazaProdMed.SendCancelacTransacc(usuario, password, codigo_transaccion)
MsgBox "Resultado: " & TrazaProdMed.Resultado & vbCrLf & _
       "CodigoTransaccion: " & TrazaProdMed.CodigoTransaccion, _
       vbInformation, "SendCancelacTransacc"
```

Intefase COM en VFP 9

```
*-- Crear objeto interface COM
TrazaProdMed = CREATEOBJECT("TrazaProdMed")

*-- Establecer credenciales de seguridad
TrazaProdMed.Username = "testwservice"
TrazaProdMed.Password = "testwservicepsw"

*-- Conectar al servidor (pruebas)
ok = TrazaProdMed.Conectar()

*-- datos de prueba
usuario = "pruebasws"
password = "pruebasws"
f_evento = "25/11/2011"
h_evento = "04;24"
gln_origen = "glnws"
gln_destino = "glnws"
n_remito = "1234"
n_factura = "1234"
vencimiento = "30/11/2011"
gtin = "GTIN1"
lote = "1111"
numero_serial = "12349"
id_obra_social = ""
id_evento = 133

*-- Establezco parámetros adicionales:

ok = TrazaProdMed.SetParametro("nro_afiliado", "9999999999999")

ok = TrazaProdMed.CrearTransaccion( ;
                    f_evento, h_evento, gln_origen, gln_destino, ;
                    n_remito, n_factura, vencimiento, gtin, lote, ;
                    numero_serial, id_evento ;

*-- Enviar datos y procesar la respuesta:
ok = TrazaProdMed.InformarProducto(usuario, password)

*-- Datos de la respuesta;

? TrazaProdMed.Resultado
? TrazaProdMed.CodigoTransaccion

*-- NOTA: extraer las transacciones enviadas (para poder enviar nuevas)
DO WHILE TrazaProdMed.LeerTransaccion()
   ? "Nro Serie:", TrazaProdMed.GetParametro("nroSerial")
ENDDO
```

Errores

Fallas SOAP (SoapFault) en atributo `Excepcion`:

- `soap:Server: La aplicacion usuario:"testwservice" intento ingresar con el password invalido:"testwservicepsw"`: verificar atributos `Username`, `Password` y url en `Conectar` (ambiente testing o producción)
- `soap:Server: La aplicacion usuario:"testwservice" no esta registrado en el sistema`: verificar atributos `Username`, `Password` y url en `Conectar` (ambiente testing o producción)
- `ns1:InvalidSecurity: An error was discovered processing the <wsse:Security> header`: ha proporcionado incorrectamente las credenciales de acceso o la biblioteca no soporta los encabezados de seguridad requeridos.
- `soap:Server: Error grave: null`: por lo visto, si no se especifica criterio de búsqueda o el criterio es muy amplio el webservice está devolviendo este error. En `GetTransaccionesWS` especificar al menos los parámetros de fechas (`fecha_desde_op` y `fecha_hasta_op`, ver ejemplo completo en VB)

Errores Internos del webservice en atributo `Errores` (lista):

- `99999999999: Acceso denegado.`: verificar usuario y contraseña pasada al método `SendProductos Médicos` (para testing es "pruebasws1" "pruebasws")
- `1000: El evento no se corresponde con los datos ingresados para GLN Origen y/o GLN Destino`: verifique los datos enviados (`numero_serial`), solo se puede enviar los requerimientos una vez.
- `3: Transaccion NO encontrada, NO se puede anular.`: en ocasiones sucedía por una falla interna de los servidores o está operando en homologación (testing/entrenamiento), si se ha informado la transacción con InformarProducto, consultar con los técnicos de PAMI/ANMAT: contactotrazabilidad@pami.org.ar

El atributo Errores contendrá una lista con el código y mensaje de error devuelto por el webservice, se puede recorrer uno a uno:

```
' recorro y muestro los errores
For Each er In TrazaProdMed.Errores
    MsgBox er, vbExclamation, "Error"
Next
```

Se puede recorrer la lista o llamar a LeerError() que irá devolviendo cada error uno por uno, retornando cadena vacía al finalizar, ejemplo:

```
' obtengo el error (si hay)
er = TrazaProdMed.LeerError()
Do While er <> ""
    MsgBox er, vbExclamation, "Error"
    er = TrazaProdMed.LeerError()   ' obtengo el proximo error (si hay)
Next
```

Novedades

Se recuerda que esta disponible el grupo de noticias (http://groups.google.com.ar/group/pyafipws) donde se publicarán futuras novedades sobre PyAfipWS: servicios web de factura electrónica y sus interfases (se recomienda suscribirse)

Historial de cambios:

- Diciembre de 2016: Versión inicial

IDL y TLB (libreria de tipos) para lenguajes estáticos

Disponemos de una versión del instalador para Trazabilidad de ANMAT con el archivo TrazaProdMed.TLB (Type Library) para usar en Clarion y otros lenguajes estáticos:

http://www.sistemasagiles.com.ar/soft/pyafipws/instalador-TrazaMed-1.05a-tlb-homo.exe

Luego de instalar, deben registrar el tlb ejecutando

```
C:\Archivos de Programa\TrazaMed\TrazaMed.EXE --register
```

El IDL fuente está publicado en:

https://github.com/reingart/pyafipws/blob/master/typelib

IMPORTANTE: el TLB no sirve para la versión normal previa, solo se debe usar con este instalador y viceversa.

Si lo usan con Visual Basic, deben ir a Referencias y agregar el archivo TLB, y luego cambiar la definición:

```
Dim TrazaProdMedObj As TrazaProdMed
```

La única diferencia con la versión estandard sin TLB, es que se activa el autocompletado y tips de llamadas (información de metodos y parámetros estáticos), pero no hay parámetros opcionales (deben completarlos con string vacios "") y se pierde las funcionalidades dinámicas (deben pasar practicamente todo como string)

Costos y Condiciones

Por soporte comercial consultar al (011) 15-3048-9211 o por mail a info@sistemasagiles.com.ar

Costos de soporte estimativos (puede variar dependiendo de las necesidades de cada implementación puntual):

- Soporte mínimo: $49.680.- (por 1 semana de cobertura), sólo acceso a instalador y soporte por temas de instalación únicamente, no incluye consultas generales o ajustes. (prefentemente para clientes actuales)
- Soporte básico: $131.100.- (hasta 6 hs en total por 1 mes máx.), incluye consultas particulares y ajustes menores, contemplando TLB (TypeLib? para lenguajes estáticos -solo TrazaMed?, consultar otros WS-)
- Soporte avanzado: $196.650.- (hasta 9hs en total por 3 meses máx.) adicional, incluyendo ajustes y desarrollo de ejemplos, documentación, pruebas, etc., contempla temas urgentes y/o grandes empresas/ciclos de desarrollo

- Soporte por actualización: desde $49.680 (1 semana máx., hasta 1 hs en total, solo instalación y acceso a actualizaciones por correcciones generales), aplica a la versión 2 para clientes previos.

Para soporte sin cargo de la comunidad, revisar la lista de temas y/o crear uno nuevo. Por novedades y consultas genereales, puede usar el Google Groups (Foro Público). Código fuente en GitHub.

Más información en PyAfipWs

MarianoReingart MarianoReingart MarianoReingart