

Manual de Uso PyAfipWs: Interfase y herramientas para servicios web AFIP (Factura Electrónica y otros)

- **Factura Electrónica: y Servicios Web AFIP**

- ◆ PyAfipWs: Interfase con otros lenguajes (VB, VFP, Cobol ...)
- ◆ Factura Electrónica en Python: Información Técnica (SOAP, XML)
- ◆ Bonos Fiscales Electrónicos: Bienes de Capital (RG2557)
- ◆ Factura Electrónica Exportación: Exportadores (RG2758)
- ◆ Código Trazabilidad de Granos: Transporte de granos (RG2806)
- ◆ PyRece: Aplicativo simil SIAP - RECE (CSV, PDF, Email)
- ◆ FE.py: Herramienta universal, unificada e integrada
- ◆ Manual de PyAfipWs: Documentación (Manual de Uso)

Emisión y almacenamiento electrónico de comprobantes originales AFIP - Argentina. RG 1956/05, 1361/02, 1345/02, 2265/07, 2289/07, 2557/09, 2758/10

Código de Trazabilidad de Granos (Carta de Porte). RG 2806/10

Depositorio Fiel. RG 2570/09

2008/2009/2010 © Mariano Reingart ? Versión 1.24 ? Julio 2010

Índice

Importante: leer primero antes de comenzar

Funcionamiento del campo ID

El ID es el Identificador del requerimiento requerido por AFIP, un número interno de secuencia controlado por el emisor, que permite identificar de manera única cada operación de autorización (solicitud de CAE). Este dato es de vital importancia para poder recuperar un CAE frente a problemas de comunicación o fallas del hardware/software. Sin el, es imposible recuperar un CAE y se puede llegar a bloquear todo el circuito de facturación electrónica en los servidores de AFIP. Por ello es recomendable que sea un dato propio del sistema de facturación, almacenado en un soporte permanente (base de datos en el disco rígido o similares)

El ID debe ser único para cada comprobante autorizado, y debe cambiar si hubo rechazo previo. Los ID no deben reutilizarse salvo para realizar reprocesos (recuperar un CAE ya emitido). La AFIP almacena las transacciones según su ID, a mismo ID, mismo resultado y CAE (independientemente de los otros datos de la operación de transacción).

En el caso de fallas, los webservices poseen métodos para recuperar el último ID informado (por ej., ver UltNro? o GetLastID)

Si bien los ejemplos (incluyendo los oficiales) utilizan generalmente el último ID + 1, esto es solo para ilustrar el método en general, no es recomendable su uso en producción, además de requerir comunicación extra (no es obligatorio llamar al método último ID), ya que puede generar inconvenientes en caso de múltiples equipos solicitando autorización simultáneamente, fallas de comunicación o problemas de almacenamiento. Sería recomendable, por ej, que el ID sea un campo autonumérico, autoincremental o serial secuencial que acompañe a la factura.

El ID es arbitrario y no tiene relación con el n° de factura. La única condición es que debe ser único, pero no es necesario que sea creciente o secuencial.

Validez de Comprobantes Electronicos emitidos

El uso incorrecto de los servicios web (no respetando el funcionamiento de los campos, en especial el ID, reproceso, motivos de observación y errores), puede ocasionar comportamientos erráticos (facturas no autorizadas o autorizadas con otros datos). Se recomienda revisar el funcionamiento general de su sistema en producción utilizando las validaciones que prevee la AFIP:

- Consulta interactiva de CAE: <http://www.afip.gov.ar/genericos/consultacae/>
- Por Clave Fiscal, servicio Verificación de validez de comprobantes emitidos (solo receptores)
- Por servicio web para Facturas E, A y B (Bono Fiscal): Método Recuperar comprobante (para facturas de exportación, solo emisores)

Mensajes XML

Los mensajes XML contienen la comunicación transmitida entre el sistema y la AFIP en el formato de intercambio de los servicios web:

- requerimiento.xml: incluye credenciales de acceso y la solicitud remota (método a utilizar, parámetros, etc.)
- respuesta.xml: incluye los datos devueltos por el proceso remoto, incluyendo mensajes de error (si corresponde)

Almacenar estos archivos es de vital importancia para el registro de las operaciones realizadas por webservice y su eventual verificación, depuración o solución de inconvenientes. La información contenida en estos archivos es solo relevante a los desarrolladores, no es necesario que sean enviados al usuario final.

Dependiendo de la interfaz, se pueden recuperar estos mensajes en forma de cadenas de texto para su posterior almacenamiento, o generar los archivos correspondientes.

Para más información ver atributos XmlRequest? y XmlResponse? o parámetros xml.

Documentación Oficial AFIP

Recomendamos revisar y seguir los lineamientos y normativa de AFIP:

- <http://www.afip.gov.ar/efactura/>
- <https://wshomo.afip.gov.ar/wsfedocs/>
- <http://wshomo.afip.gov.ar/fiscaldocs/>

Recordamos que la interfaz y bibliotecas ([PyAfipWs](#)) no realizan ningún tipo de chequeo o validación adicional (ya que esta tarea la realizan los servidores de AFIP), y tampoco convierte entre distintos tipos de datos (salvo conversión trivial a cadenas -strings- para ser enviadas por XML), por lo que es responsabilidad de la aplicación principal cumplir con las especificaciones y legislación vigente referidas a los servicios web de AFIP, proporcionando los datos correctos que correspondan.

Modo homologación

Por diseño, el instalador para Windows publicado de la interfaz COM (`pyafipws.py`) y herramientas por línea de comando (`rece.py`, `recex.py`, `receb.py`) funcionará inicialmente solo en modo homologación (testing), ya que su objetivo es simplemente facilitar las pruebas para que los usuarios que no concen Python puedan evaluar el producto antes de interiorizarse en el lenguaje y/o contratar el soporte comercial.

En este modo de homologación (pruebas), se ignorará las URLs proporcionadas a CallWSAA, Conectar o especificadas en `rece.ini`. Para activar modo producción (definitivo) es necesario ajustar el código fuente, ver instrucciones de [Instalación desde Código Fuente](#):

No es obligatorio usar el instalador ni realizar compilación, se puede bajar, revisar, modificar y ejecutar directamente el código fuente publicado en modo homologación o producción.

Ofrecemos soporte comercial, con el instalador ajustado para producción, para los clientes que así lo requieran.

Importante: Los CAE obtenidos en modo homologación no tienen validez fiscal

Advertencia: Teóricamente no debería ser posible conectarse a producción con certificados de homologación (y viceversa), pero en la práctica han sucedido casos (obteniendo CAE de homologación sin validez fiscal), por lo que recomendamos extremar las medidas de seguridad y control (verificando certificados y configuración de URL de servidores) para generar las facturas en el ambiente correcto. Ver [Verificar Validez CAE](#) y [Comprobación WSAA](#)

Disclaimer

Toda información es proporcionada a Título Informativo. El programa es software libre liberado bajo licencia GPLv3 y se entrega como está, sin garantías explícitas ni implícitas de ningún tipo, incluyendo sin limitación, pérdida de ganancias, interrupción de negocios, pérdida de programas u otros datos en sistemas de computación o cualquier otro reclamo. Al usarlo acepta hacerlo bajo su propia responsabilidad, conociendo la normativa y reglamentaciones existentes.

Instalación

Bajar y Ejecutar el autoinstalable:

- [instalador-pyafipws-v23b.exe](#) (elegir la última versión disponible)

Instrucciones: La interfase se instala automáticamente en `C:\PYAFIPWS`

- Instalación manual: `pyafipws.exe --register` o utilizar `regsvr32 pyafipws.dll`
- Desinstalación: `pyafipws.exe --unregister` o utilizar `regsvr32 /u`

pyafipws.dll

Un ejemplo completo del uso de la interfase en Visual Basic se puede descargar desde:

- [ejemplo-wsfe.zip](#)
- [ejemplo-wsbfe.zip](#)
- [ejemplo_wsfex.zip](#)

Para más información:

- Sitio del proyecto: <http://www.pyafipws.com.ar/>
- Grupo de noticias: <http://groups.google.com/group/pyafipws>
- Código Fuente: <http://code.google.com/p/pyafipws/>
- Información general: <http://www.sistemasagiles.com.ar/trac/wiki/PyAfipWs>
- Por mail: facturaelectronica@sistemasagiles.com.ar

Servicio Web de Autenticación y Autorización (WSAA)

Este servicio es necesario para acceder al resto de los servicios de negocios (ej. Factura Electrónica) ofrecidos por la AFIP.

Se debe generar un Ticket de Requerimiento de Acceso (TRA), firmarlo criptográficamente (generando un mensaje CMS) que es envía al servicio web para obtener un Ticket de Acceso (TA).

Esta interfase crea el mensaje en XML, lo firma criptográficamente y lo codifica en base-64 como lo requiere la AFIP, por lo que en general no es necesario realizar ninguno de estos pasos.

Para más información ver especificaciones técnicas del WSAA en el sitio de la AFIP:

https://wswhomo.afip.gov.ar/wsfedocs/Especificacion_Tecnica_WSAA_1.2.0.pdf

Objeto

- El objeto COM se crea invocando a `CreateObject("WSAA")`

Métodos

- `CreateTRA(service, ttl)`: recibe el nombre del servicio (wsfe, wsbfe, wsfex, wsctg, etc.) y el tiempo de vida (en segundos, por defecto 2400) y devuelve un ticket de requerimiento de acceso (string con formato xml). Se puede revisar o modificar el xml devuelto (en general no es necesario)
- `SignTRA(tra, certificado, clave_privada)`: recibe el ticket de requerimiento de acceso (string con formato xml, creado previamente) y las ubicaciones del certificado y clave privada. Firma el ticket y devuelve el mensaje CMS (string con formato xml).
- `CallWSAA(cms, url_webservice)`: Recibe el mensaje CMS (string con formato xml) y la URL del webservice. Se conecta con el servidor remoto, ejecuta el método remoto `LoginCMS` y devuelve el Ticket de Acceso (TA, string con formato xml) generado por la AFIP. Si no se especifica url, se utiliza servidores de homologación.

Atributos

- Token: el código de autorización generado por la AFIP, necesario para utilizar el resto de los servicios web.
- Sign: la firma de autorización generado por la AFIP, necesario para utilizar el resto de los servicios web.
- Version: versión de la interfase (ej. ?1.6?)

Ejemplo

```
Dim WSAA As Object

' Crear objeto interface Web Service Autenticación y Autorización
Set WSAA = CreateObject("WSAA")

' Generar un Ticket de Requerimiento de Acceso (TRA)
service = "wsfe" ' servicio a acceder
ttl = 2400      ' tiempo de vida en segundos
tra = WSAA.CreateTRA(service, ttl)
' Especificar la ubicación de los archivos certificado y claves
Path = CurDir() + "\"
Certificado = "reingart.crt" ' certificado es el firmado por la afip
ClavePrivada = "reingart.key" ' clave privada usada para crear el cert.

' Generar el mensaje firmado (CMS)
cms = WSAA.SignTRA(tra, Path + Certificado, Path + ClavePrivada)

' Llamar al web service para autenticar:
ta = WSAA.CallWSAA(cms, url_web_service)

' Obtener Token y Sign de autorización
Token = WSAA.Token
Sign = WSAA.Sign
```

El Ticket de Acceso contiene el Token (código) y Sign (firma) son los elementos que se utilizan en los otros servicios web para lograr acceso a los mismos.

Estos datos se pueden usar por un período de tiempo (5 horas, modificable), siendo necesario volver a invocar el WSAA solo al finalizar este período, para obtener nuevos Token y Sign.

Errores

El WSAA informa errores alfanuméricos (Err.Number ? vbObjectError = 0), debiendo verificar la descripción (Err.Description). Entre ellos los más comunes son:

- coe.notAuthorized, cms.cert.expired, cms.cert.invalid, cms.cert.untrusted: El certificado no fue firmado por la AFIP, expiró, es inválido, no se confía en su emisor, o no coincide la fuente. Se debe revisar o emitir correctamente los certificados.
- xml.generationTime.invalid, xml.expirationTime.expired, xml.expirationTime.invalid: El tiempo de generación/expiration es inválido. Debe sincronizar relojes con el servidor de la AFIP.
- wsn.unavailable, wsaa.unavailable: servicio no disponible momentáneamente

Servicio Web de Factura Electrónica (WSFE)

Este servicio es ofrecido por la AFIP para la facturación electrónica.

Para poder acceder, se debe obtener un Ticket de Acceso (TA) previamente (para utilizar el Token y Sign y enviarlos a los métodos de facturación electrónica).

Esta interfase recibe los valores y crea los mensaje en XML, lo envía y analiza la respuesta XML, por lo que no es necesario modificar XML. Se recomienda pasar los valores en formato string, números separados por punto y fechas YYYYMMDD.

Para más información ver especificaciones técnicas del WSFE en el sitio de la AFIP:

<http://wswhomo.afip.gov.ar/fiscaldocs/WSFE/WSFE%20-%20Manual%20para%20el%20desarrollador%20-%20200903>

Objeto

- El objeto COM se crea invocando a `CreateObject("WSFE")`

Métodos

- `Conectar(url)`: realiza la conexión a los servidores de la AFIP (primer paso esencial). Si no se especifica `ur`, se utiliza servidores de homologación.
- `Dummy()`: servicio de prueba para obtener el estado de los servidores de la AFIP.
- `UltNro()`: recuperar el último número de transacción (id). Devuelve el último valor de secuencia utilizado al pedir autorización. Uso opcional.
- `Aut(id, presta_serv, tipo_doc, nro_doc, tipo_cbte, punto_vta, cbt_desde, cbt_hasta, imp_total, imp_tot_conc, imp_net, impto_liq, impto_liq_rni, imp_op_ex, fecha_cbte, fecha_venc_pago, [fecha_serv_desde], [fecha_serv_hasta])`: autoriza la emisión de factura electrónica, recibe el id de secuencia (recuperado por `UltNro`), `presta_serv` en 1 si se esta facturando servicios (en este caso es obligatorio el uso de la fecha de servicio desde y hasta), los datos de la factura a emitir, y devuelve el Código de Autorización Electrónico (CAE). Ver ejemplo para el detalle de los parámetros.
- `RecuperaLastCMP(punto_venta, tipo_comprobante)`: recibe el punto de venta y tipo de comprobante (strings/enteros), devuelve el último número de comprobante autorizado. Uso opcional.
- `RecuperaQty()` recupera cantidad máxima de registros de detalle (por el momento no es de mayor utilidad ya que por simplicidad se envía 1 registro de detalle por autorización). Uso opcional.

Atributos

- `Token`: es el código de autorización generado por la AFIP (WSAA)
- `Sign`: es la firma de autorización generado por la AFIP (WSAA)
- `Cuit`: es el número de CUIT del emisor de facturas, formato string sin guiones.
- `AppServerStatus`, `DbServerStatus`, `AuthServerStatus`: estados de los servidores de AFIP (string ?OK? en caso de estar funcionales)
- `XmlRequest`, `XmlResponse`: requerimiento y respuesta XML (para depuración)
- `Respuesta`, `Motivo`, `Reproceso`: valores complementarios que retorna `Aut`
- `CbtDesde?`, `CbtHasta?`, `FechaCbte?`: números y fecha de comprobante autorizado que retorna `Aut`
- `ImpTotal?`, `ImpNeto?`, `ImptoLiq?`: importes total, neto e IVA que retorna `Aut`
- `CAE`, `Vencimiento`: CAE y Fecha de vencimiento autorización retorna `Aut`

- Version: versión de la interfase (ej. ?1.6?)

Errores

La interfase procesa los errores WSFE y los informa con códigos numéricos (Err.Number ? vbObjectError = nº de error de AFIP) y su correspondiente descripción (Err.Description). Entre ellos los más comunes son:

- 1000: Usuario no autorizado a realizar esta operación. Revisar Token y Sign obtenido del WSAA
- 1001: CUIT no autorizado. Revisar en el sitio de la AFIP si se asoció el CUIT del emisor al servicio de factura electrónica, con el correspondiente certificado.
- 1012,1014,1016,1017,1018,1019: Inconsistencia en los datos enviados. Revisar formatos (fechas, números, etc., usar fechas YYYYMMDD y números separados por puntos, ambos pasarlos como string para evitar errores). Revisar tipo de datos (tipo_cbte, tipo_doc) que correspondan a las tablas usadas por la AFIP. Revisar que los datos sean válidos para la operación a realizar (ej. Factura A solo para tipo_doc = 80 CUIT, Facturas B por lote hasta \$1000.-, etc.)

Ver especificaciones técnicas del WSFE para más información sobre el uso del servicio web, su formato, condiciones y errores.

Ejemplo

Autorizar una factura electrónica nacional a modo de ejemplo (ficticia):

```
' Crear objeto interface Web Service de Factura Electrónica

Set WSFE = CreateObject("WSFE")

' Setear token y sing de autorización (pasos previos)
WSFE.Token = WSAA.Token
WSFE.Sign = WSAA.Sign
' CUIT del emisor (debe estar registrado en la AFIP)
WSFE.Cuit = "20267565393"

' Conectar al Servicio Web de Facturación (servidor de homologación)
ok = WSFE.Conectar("https://wshomo.afip.gov.ar/wsfe/service.asmx")

' Llamo a un servicio ?tonto?, para obtener el estado (opcional)
WSFE.Dummy
Debug.Print "appserver status", WSFE.AppServerStatus
Debug.Print "dbserver status", WSFE.DbServerStatus
Debug.Print "authserver status", WSFE.AuthServerStatus

' Recupera cantidad máxima de registros (opcional)
qty = WSFE.RecuperarQty()

' Recupera último número de secuencia ID
LastId = WSFE.UltNro()

' Recupero último número de comprobante (opcional)
tipo_cbte = 1: punto_vta = 1
LastCBTE = WSFE.RecuperaLastCMP(punto_vta, tipo_cbte)

' Establezco los valores de la factura o lote a autorizar: "100.00"
impto_liq = "21.00": impto_liq_rni = "0.00": imp_op_ex = "0.00"
fecha_cbte = Fecha: fecha_venc_pago = Fecha
' Fechas del período del servicio facturado (solo si presta_serv = 1)
```

```

fecha_serv_desde = Fecha: fecha_serv_hasta = Fecha

' Llamo al Webservice de Autorización para obtener el CAE
cae = WSFE.Aut(id, presta_serv, _
  tipo_doc, nro_doc, tipo_cbte, punto_vta, _
  cbt_desde, cbt_hasta, imp_total, imp_tot_conc, imp_netto, _
  impto_liq, impto_liq_rni, imp_op_ex, fecha_cbte, fecha_venc_pago,
  fecha_serv_desde, fecha_serv_hasta) ' solo pasar si presta_serv=0
MsgBox "CAE: " & cae & " Vencimiento: " & WSFE.Vencimiento
Exit Sub

ManejoError:
' Si hubo error:
Debug.Print Err.Description          ' descripción error afip
Debug.Print Err.Number - vbObjectError ' código error afip
MsgBox Err.Description, vbCritical + vbRetryCancel, _
  "Error:" & (Err.Number - vbObjectError) & " en " & Err.Source

```

Nota: no es necesario (ni recomendado) recuperar el último ID y/o último número de comprobante. Dichos datos deberían estar almacenados de manera permanente en la aplicación que use la interfaz (ver aclaración).

Descripción del Método Aut (obtención de CAE)

Se debe llamar al método Aut con los siguiente parámetros:

- id: Número de identificación secuencial (debe almacenarse en el sistema local) No es obligatorio llamar a UltNro, puede utilizarse un dato local secuencial.
- presta_serv: 0 o 1 para indicar si es una factura de servicios
- tipo_doc, nro_doc: Tipo (80 CUIT, 96 DNI, etc.) y número de Documento
- tipo_cbte: Tipo de comprobante (1 Factura A , 6 Factura B, etc.)
- punto_vta: N° de punto de venta (debe estar autorizado)
- cbt_desde, cbt_hasta: N° de comprobante (si es un solo comprobante, repetir N°)
- imp_total: Importe total de la factura
- imp_tot_conc: Importe total de conceptos no gravados por el IVA
- imp_netto: Importe neto (gravado por el IVA) de la factura
- impto_liq: Importe del IVA liquidado (incluyendo percepciones de IVA)
- impto_liq_rni: Importe IVA RNI (no se utiliza mas, dejar 0.00)
- imp_op_ex: Importe de operaciones exentas
- fecha_cbte: Fecha del comprobante (no puede ser mayor o menor a 5 días)
- fecha_venc_pago: Fecha límite de vencimiento para el pago de la factura
- fecha_serv_desde, fecha_serv_hasta: Fechas del período de servicios prestado

Ver todos los tipos de comprobante: <https://wshomo.afip.gov.ar/wsfedocs/CodCbpte.txt>

Ver todos los tipos de documentos: <https://wshomo.afip.gov.ar/wsfedocs/CodDocID.txt>

Si no hubiere inconvenientes, la llamada debe devolver el CAE y se establece el atributo WSFE.Resultado = "A" (Aceptado) y WSFE.Vencimiento. Sino, devuelve:

- cae = "": La interfase no pudo procesar la respuesta del Webservice, o bien la conexión a internet esta caída, las direcciones de los servidores son incorrectas, o los servidores de la AFIP no funcionan o no devuelven la respuesta esperada.

- cae = "NULL": El servicio web rechazo los datos de la factura a generar. Se establece el atributo WSFE.Resultado = "R" (Rechazado) y WSFE.Motivo con los diversos motivos de rechazo proporcionados por el webservice:
 - ◆ WSFE.Motivo = "00" o "NULL": no hay error (solo como referencia)
 - ◆ WSFE.Motivo = "01": CUIT informada no es R.I.
 - ◆ WSFE.Motivo = "02": CUIT no autorizada a facturar electrónicamente
 - ◆ WSFE.Motivo = "03": CUIT registra inconvenientes con domicilio fiscal
 - ◆ WSFE.Motivo = "04": Punto de venta no se encuentra declarado
 - ◆ WSFE.Motivo = "05": Fecha del comprobante incorrecta
 - ◆ WSFE.Motivo = "06": CUIT no puede emitir comprobantes clase A
 - ◆ WSFE.Motivo = "07": para comprobantes clase A debe indicar CUIT
 - ◆ WSFE.Motivo = "08": CUIT informada es inválida
 - ◆ WSFE.Motivo = "09": CUIT informada no existe en el padrón
 - ◆ WSFE.Motivo = "10": CUIT informada no corresponde a un R.I.
 - ◆ WSFE.Motivo = "11": el nº de comprobante no es correlativo
 - ◆ WSFE.Motivo = "12": el rango informado se encuentra autorizado

Aún devolviendo el CAE y con el resultado Aceptado, puede haber un motivo (ej 01), que funcionan como advertencia para revisar la base de datos y corregir posibles irregularidades.

Advertencia: En el caso de que la conexión se interrumpa luego de que la AFIP registre la factura y antes de que la información del CAE llegue al equipo, la única alternativa es repetir la llamada a Aut con los mismos datos (incluido el ID original) para obtener el CAE generado anteriormente. En ese caso, se WSFE.Reproceso es ?S? para indicar que el CAE se ha recuperado. Por esto, es importante guardar el ID o utilizar un dato local.

Ver <http://wswhomo.afip.gov.ar/fiscaldocs/WSFE/WSFE-GuiaAdicionalParaElProgramador.pdf>

Servicio Web de Bono Fiscal Electrónico (WSBFE)

EL WSBFE (Bonos Fiscales Electrónicos) es un Servicio Web de la AFIP para Facturas Electrónicas de Bienes de Capital, correspondiente al Artículo 3 de la Resolución General 2557/2009, Este servicio es ofrecidos por la AFIP para la facturación electrónica.

Para poder acceder, se debe obtener un Ticket de Acceso (TA) previamente (para utilizar el Token y Sign y enviarlos a los métodos de facturación electrónica).

Esta interfase recibe los valores y crea los mensaje en XML, lo envía y analiza la respuesta XML, por lo que no es necesario modificar XML. Se recomienda pasar los valores en formato string, números separados por punto y fechas YYYYMMDD.

La operatoria es bastante similar al método de autorización del WSFE (con un ID secuencial, observaciones, reproceso, etc.), teniendo en cuenta esta mayor complejidad por tener que informar el detalle de cada item.

A su vez, el WSBFE devuelve mensajes de eventos (mantenimiento programado, advertencias, etc.), los que deben ser capturados e informados al usuario.

A diferencia del WSFE, este nuevo servicio funciona con tablas dinámicas de parámetros para los códigos de comprobante, moneda, alícuotas de iva, producto según NCM, zonas, unidades de medida. Estas tablas pueden sufrir modificaciones realizadas por la AFIP, con altas y bajas lógicas, por lo que tienen una fecha de vigencia (desde, hasta) y se proveen métodos para consultarlas por el mismo servicio web (a diferencia del

WSFE, que las tablas eran documentadas estáticamente en el sitio web).

Para más información ver especificaciones técnicas del WSBFE en el sitio de la AFIP:

<http://wswhomo.afip.gov.ar/fiscaldocs/WSBFE/WSBFE-ManualParaEIDesarrollador.pdf>

<http://wswhomo.afip.gov.ar/fiscaldocs/WSBFE/README.txt>

La programación de la interfase WSBFE es similar a WSFE, se recomienda revisarlo.

Objeto

- El objeto COM se crea invocando a `CreateObject("WSBFE")`

Métodos

- `Conectar(url)`: realiza la conexión a los servidores de la AFIP (primer paso esencial). Si no se especifica ur, se utiliza servidores de homologación.
- `Dummy()`: servicio de prueba para obtener el estado de los servidores de la AFIP.
- `CrearFactura(tipo_doc, nro_doc, zona, tipo_cbte, punto_vta, cbte_nro, fecha_cbte, imp_total, imp_netto, impto_liq, imp_tot_conc, impto_liq_rni, imp_op_ex, imp_perc, imp_iibb, imp_perc_mun, imp_internos, imp_moneda_id, Imp_moneda_ctz)`: crea internamente una factura para luego poder autorizarla, recibe los datos de la factura a emitir. Ver ejemplo para el detalle de los parámetros.
- `AgregarItem(ncm, sec, ds, qty, umed, precio, bonif, iva_id, imp_total)`: agrega internamente un ítem (línea de factura) a una factura para luego poder autorizarla, recibe los datos del ítem a factura a emitir. Ver ejemplo para el detalle de los parámetros.
- `Authorize(id)`: autoriza la emisión de factura electrónica para bono fiscal electrónico, recibe el id de secuencia (recuperado por `WSFE.UltNro`), y devuelve el Código de Autorización Electrónico (CAE). Ver ejemplo para el detalle de los parámetros.
- `GetCMP(tipo_cbte, punto_vta, cbte_nro)`: recupera los datos de una factura autorizada, recibe tipo de comprobante, punto de venta y número de comprobante original, y devuelve el Código de Autorización Electrónico (CAE) obtenido en su momento. A su vez, establece los datos de la factura (Cae, FechaCbte, ImpTotal, ImpNeto, ImptoLiq). Ver ejemplo para el detalle de los parámetros y valores devueltos.
- `GetLastCMP(tipo_cbte, punto_vta)`: recupera el último número de factura autorizada, recibe tipo de comprobante y punto de venta. Ver `WSFE.RecuperaLastCMP`
- `GetLastID()`: recupera el mayor número de secuencia (id) utilizado. Ver `WSFE.UltNro`
- `GetParamMon()`, `GetParamTipoCbte()`, `!GetParamTipoIVA()`, `GetParamUMed()`, `!GetParamNCM()`: recupera valores referenciales de códigos de las tablas de parámetros, devuelve una lista de strings con el id/código, descripción del parámetro y vigencia -si corresponde- (ver ejemplos). Más información en [Tablas de Parámetros](#)

Atributos

- `Token`: es el código de autorización generado por la AFIP (WSAA)
- `Sign`: es la firma de autorización generado por la AFIP (WSAA)
- `Cuit`: es el número de CUIT del emisor de facturas, formato string sin guiones.
- `AppServerStatus`, `DbServerStatus`, `AuthServerStatus`: estados de los servidores de AFIP (string ?OK? en caso de estar funcionales)
- `XmlRequest`, `XmlResponse`: requerimiento y respuesta XML (para depuración)

- Respuesta, Obs, Reproceso: valores complementarios que retorna Aut
- CAE, Vencimiento: CAE y Fecha de vencimiento autorización retorna Aut
- Version: versión de la interfase (ej. ?1.11?)
- FechaCbte: fecha del comprobante (del comprobante recuperado devuelto por GetCmp)
- ImpTotal: importe total del comprobante (del comprobante recuperado devuelto por GetCmp)
- ImpNeto: importe neto del comprobante (del comprobante recuperado devuelto por GetCmp)
- ImptoLiq: impuesto liquidado (IVA) del comprobante (del comprobante recuperado devuelto por GetCmp)

Errores

La interfase procesa los errores WSFE y los informa con códigos numéricos (Err.Number ? vbObjectError = nº de error de AFIP) y su correspondiente descripción (Err.Description). Entre ellos los más comunes son:

- 1000: Usuario no autorizado a realizar esta operación. Revisar Token y Sign obtenido del WSAA
- 1001: CUIT no autorizado. Revisar en el sitio de la AFIP si se asoció el CUIT del emisor al servicio de factura electrónica, con el correspondiente certificado.
- 1012,1014,1016,1017,1018,1019: Inconsistencia en los datos enviados. Revisar formatos (fechas, números, etc., usar fechas YYYYMMDD y números separados por puntos, ambos pasarlos como string para evitar errores). Revisar tipo de datos (tipo_cbte, tipo_doc) que correspondan a las tablas usadas por la AFIP. Revisar que los datos sean válidos para la operación a

Ver especificaciones técnicas del WSBFE para más información sobre el uso del servicio web, su formato, condiciones y errores.

Ejemplo

Autorizar una factura de bienes de capital (bono fiscal electrónico) a modo de ejemplo (ficticia):

```
Dim WSAA As Object, WSBFE As Object

' Crear objeto interface Web Service Autenticación y Autorización
Set WSAA = !CreateObject("WSAA")

' Generar un Ticket de Requerimiento de Acceso (TRA) para WSBFE
tra = WSAA.CreateTRA("wsbfe")

' Especificar la ubicacion de los archivos certificado y clave privada
Path = !CurDir() + "\"
' Certificado: certificado es el firmado por la AFIP
' ClavePrivada: la clave privada usada para crear el certificado
Certificado = "reingart.crt" ' certificado de prueba
ClavePrivada = "reingart.key" ' clave privada de prueba

' Generar el mensaje firmado (CMS)
cms = WSAA.SignTRA(tra, Path + Certificado, Path + !ClavePrivada)
' Llamar al web service para autenticar:
ta = WSAA.CallWSAA(cms, "!https://wsaahomo.afip.gov.ar/ws/services/LoginCms") ' Homologación
' Una vez obtenido, se puede usar el mismo token y sign por 24 horas

' Crear objeto interface Web Service de Factura Electrónica
Set WSBFE = !CreateObject("WSBFE")

' Setear token y sing de autorización (pasos previos)
WSBFE.Token = WSAA.Token
```

```

WSBFE.Sign = WSAA.Sign
' CUIT del emisor (debe estar registrado en la AFIP)
WSBFE.Cuit = "20267565393"

' Conectar al Servicio Web de Facturación
ok = WSBFE.Conectar("!http://wshomo.afip.gov.ar/wsbfe/service.asmx") ' homologación

' Establezco los valores de la factura a autorizar:
fecha = "20090530"
tipo_doc = 80: nro_doc = "23111111113"
zona = 0 ' Ver tabla de zonas
tipo_cbte = 1 ' Ver tabla de tipos de comprobante
punto_vta = 2: cbte_nro = 16
fecha_cbte = fecha
imp_total = "121.00": imp_tot_conc = "0.00": imp_netto = "100.00"
imppto_liq = "21.00": imppto_liq_rni = "0.00": imp_op_ex = "0.00"
imp_perc = "0.00": imp_iibb = "0.00": imp_perc_mun = "0.00"
imp_internos = "0.00"
imp_moneda_id = "10" ' Ver tabla de tipos de moneda
Imp_moneda_ctz = "1.0000" ' cotización de la moneda

' Creo una factura (internamente, no se llama al !WebService):
ok = WSBFE.!CrearFactura(tipo_doc, nro_doc, _
    zona, tipo_cbte, punto_vta, cbte_nro, fecha_cbte, _
    imp_total, imp_netto, imppto_liq, _
    imp_tot_conc, imppto_liq_rni, imp_op_ex, _
    imp_perc, imp_iibb, imp_perc_mun, imp_internos, _
    imp_moneda_id, Imp_moneda_ctz)

' Agrego un item:
ncm = "7308.10.00" ' Ver tabla de códigos habilitados del NCM
sec = "" ' Código de la Secretaría (no usado por el momento)
ds = "Prueba" ' Descripción completa del artículo (hasta 4000 car.)
umed = 1 ' kg, Ver tabla de unidades de medida
qty = "1.0" ' cantidad
precio = "105.0" ' precio neto (Fac.A), precio final (Fac..uras B)
bonif = "5.00" ' descuentos (en positivo)
iva_id = 5 ' 21%, ver tabla alícuota de iva
imp_total = "121.00" ' importe total final del artículo
' lo agrego a la factura (internamente, no se llama al !WebService):
ok = WSBFE.!AgregarItem(ncm, sec, ds, qty, umed, precio, bonif, iva_id, imp_total)

' Llamo al !WebService de Autorización para obtener el CAE
cae = WSBFE.Authorize(id)

If cae = "" Or WSBFE.Resultado <> "A" Then
    MsgBox "No se asignó CAE (Rechazado). Observación (motivos): " & WSBFE.Obs, _
        vbInformation + vbOKOnly
ElseIf WSBFE.Obs <> "" And WSBFE.Obs <> "00" Then
    MsgBox "Se asignó CAE pero con advertencias. Observación (motivos): " & WSBFE.Obs, _
        vbInformation + vbOKOnly
End If

MsgBox "!Resultado:" & WSBFE.Resultado & " CAE: " & cae & _
    " Reproceso: " & WSBFE.Reproceso & _
    " Obs: " & WSBFE.Obs, vbInformation + vbOKOnly

' Muestro los eventos (mantenimiento programados y otros mensajes de la AFIP)

For Each evento In WSBFE.Eventos
    If evento <> "0: " Then
        MsgBox "Evento: " & evento, vbInformation
    
```

```
End If
Next
```

Nota: no es necesario (ni recomendado) recuperar el último ID y/o último número de comprobante. Dichos datos deberían estar almacenados de manera permanente en la aplicación que use la interfaz([ver aclaración](#)).

Ejemplo para obtener códigos referenciales del Nomenclador Común Mercosur habilitados:

```
' recupero tabla del nomenclador común del mercosur
' ("codigo: descripción (vigencia desde - hasta)")
' Ej: "99.99.99.99: (item no incluido en el Beneficio Fiscal) (20070524 - 20091231)"
For Each x In WSBFE.GetParamNCM()
    Debug.Print x
Next
```

Descripción de la obtención de CAE

Como primer paso, se debe crear una factura (utilizada internamente por la interfase para contener los valores a autorizar), llamando al método CrearFactura con los siguiente parámetros:

- tipo_doc, nro_doc: Tipo (80 CUIT, 96 DNI, etc.) y número de Documento
- tipo_cbte: Tipo de comprobante (según tabla de parámetros)
- punto_vta: N° de punto de venta (debe estar autorizado para WSBFE)
- cbt_nro: N° de comprobante
- fecha_cbte: Fecha del comprobante (no puede ser mayor o menor a 10 días)
- zona_id: Zona (según tabla de parámetros) - Por el momento 0
- imp_total: Importe total de la factura
- imp_tot_conc: Importe total de conceptos no gravados por el IVA
- imp_netto: Importe neto (gravado por el IVA) de la factura
- impto_liq: Importe del IVA liquidado (incluyendo percepciones de IVA)
- impto_liq_rni: Importe IVA RNI (no se utiliza mas, dejar 0.00)
- imp_op_ex: Importe de operaciones exentas
- imp_moneda_id: Moneda de la factura (según tabla de parámetros)
- imp_moneda_ctz: Cotización de la moneda de la factura
- imp_perc: Importe de las percepciones
- imp_iibb: Importe de las percepciones de ingresos brutos
- imp_perc_mun: Importe de las percepciones municipales
- imp_internos: Importe de los impuestos internos

Luego, por cada artículo vendido (ítem), se debe llamar al método AgregarItem con los siguientes parámetros:

- ncm: código habilitado según el Nomenclador Común del Mercosur (NCM)
- sec: código de la secretaría (no utilizado por el momento, pasar ??)
- ds: Descripción completa
- precio: Precio Neto Unitario (facturas A) o precio final (facturas B)
- qty: Cantidad
- umed: Unidad de medida (según tabla de parámetros)
- bonif: Bonificación
- iva_id: Alícuota de IVA (según tabla de parámetros)
- imp_total: Importe total (incluyendo descuentos e IVA)

Como último paso, se debe llamar al método Authorize con los siguiente parámetros:

- id: Número de identificación secuencial (debe almacenarse en el sistema local) No es obligatorio llamar a WSFE.UltNro, puede utilizarse un dato local secuencial.

Si no hubiere inconvenientes, la llamada debe devolver el CAE y se establece el atributo WSBFE.Resultado = "A" (Aceptado) y WSBFE.Vencimiento. Sino, devuelve:

- cae = "": La interfase no pudo procesar la respuesta del Webservice, o bien la conexión a internet esta caída, las direcciones de los servidores son incorrectas, o el servicio web rechazo los datos de la factura a generar. Se establece el atributo WSBFE.Resultado = "R" (Rechazado) y WSBFE.Obs con los diversos motivos de rechazo proporcionados por el webservice (ver WSFE.Motivo):

Aún devolviendo el CAE y con el resultado Aceptado, puede haber un motivo (ej 01), que funcionan como advertencia para revisar la base de datos y corregir posibles irregularidades.

Advertencia: Revisar las consideraciones de reproceso al igual que con WSFE.

Servicio Web de Factura Electrónica Exportación (WSFEX)

EL WSFEX es un Servicio Web de la AFIP para Facturas Electrónicas de Exportación, correspondiente a la Resolución General 2758/2010.

Para poder acceder, se debe obtener un Ticket de Acceso (TA) previamente (para utilizar el Token y Sign y enviarlos a los métodos de facturación electrónica).

Esta interfase recibe los valores y crea los mensaje en XML, lo envía y analiza la respuesta XML, por lo que no es necesario modificar XML. Se recomienda pasar los valores en formato string, números separados por punto y fechas YYYYMMDD.

La operatoria es bastante similar al método de autorización del WSBFE (con un ID secuencial, observaciones, reproceso, etc.), teniendo en cuenta esta mayor complejidad por tener que informar los datos de exportación, permisos, comprobantes asociados y el detalle de cada item.

A su vez, el WSFEX devuelve mensajes de eventos (mantenimiento programado, advertencias, etc.), los que deben ser capturados e informados al usuario.

Al igual que el WSBFE, este nuevo servicio funciona con tablas dinámicas de parámetros para los códigos de comprobante, moneda, idiomas, terminos de comercio exterior, paises y cuits, unidades de medida, etc.. Estas tablas pueden sufrir modificaciones realizadas por la AFIP, con altas y bajas lógicas, por lo que tienen una fecha de vigencia (desde, hasta) y se proveen métodos para consultarlas por el mismo servicio web (a diferencia del WSFE, que las tablas eran documentadas estáticamente en el sitio web).

Para más información ver especificaciones técnicas del WSBFE en el sitio de la AFIP:

- [README.txt](#): información general
- [Manual para el desarrollador.pdf](#): Información importante sobre la operatoria
- [Anexo tablas referenciales.xls](#): parámetros vigentes (puede estar ligeramente desactualizado)

La programación de la interfase WSFEX es similar a WSBFE, se recomienda revisarlo.

Objeto

- El objeto COM se crea invocando a `CreateObject("WSFEX")`

Métodos

- `Conectar(url)`: realiza la conexión a los servidores de la AFIP (primer paso esencial). Si no se especifica `ur`, se utiliza servidores de homologación.
- `Dummy()`: servicio de prueba para obtener el estado de los servidores de la AFIP.
- `CrearFactura(tipo_cbte, punto_vta, cbte_nro, fecha_cbte, imp_total, tipo_expo, permiso_existente, dst_cmp, cliente, cuit_pais_cliente, domicilio_cliente, id_impositivo, moneda_id, moneda_ctz, obs_comerciales, obs, forma_pago, incoterms, idioma_cbte)`: crea internamente una factura para luego poder autorizarla, recibe los datos de la factura a emitir. Ver ejemplo para el detalle de los parámetros.
- `AgregarItem(ncodigo, ds, qty, umed, precio, imp_total)`: agrega internamente un ítem (línea de factura) a una factura para luego poder autorizarla, recibe los datos del ítem a factura a emitir. Ver ejemplo para el detalle de los parámetros.
- `AgregarPermiso(id, dst)`: agrega internamente un permiso de exportación a una factura para luego poder autorizarla, recibe identificación y país destino de la mercadería. Ver ejemplo para el detalle de los parámetros.
- `AgregarCmpAsoc(tipo_cbte_asoc, punto_vta_asoc, cbte_nro_asoc)`: agrega internamente un comprobante asociado a una factura para luego poder autorizarla, recibe tipo de comprobante, punto de venta y número. Ver ejemplo para el detalle de los parámetros.
- `Authorize(id)`: autoriza la emisión de factura electrónica para bono fiscal electrónico, recibe el id de secuencia (recuperado por `WSFE.UltNro`), y devuelve el Código de Autorización Electrónico (CAE). Ver ejemplo para el detalle de los parámetros.
- `GetCMP(tipo_cbte, punto_vta, cbte_nro)`: recupera los datos de una factura autorizada, recibe tipo de comprobante, punto de venta y número de comprobante original, y devuelve el Código de Autorización Electrónico (CAE) obtenido en su momento. A su vez, establece los datos de la factura (`Cae`, `FechaCbte`, `ImpTotal`, `ImpNeto`, `ImptoLiq`). Ver ejemplo para el detalle de los parámetros y valores devueltos.
- `GetLastCMP(tipo_cbte, punto_vta)`: recupera el último número de factura autorizada, recibe tipo de comprobante y punto de venta. Ver `WSFE.RecuperaLastCMP`
- `GetLastID()`: recupera el mayor número de secuencia (id) utilizado. Ver `WSFE.UltNro`
- `GetParamMon()`, `GetParamTipoCbte()`, `GetParamTipoExpo()`, `GetParamIdiomas()`, `GetParamUMed()`, `GetParamIncoterms()`, `GetParamDstPais()`, `GetParamDstCUIT()`: recupera valores referenciales de códigos de las tablas de parámetros, devuelve una lista de strings con el id/código o CUIT y descripción del parámetro (ver ejemplos). Más información en [Tablas de Parámetros](#)
- `GetParamCtz(moneda_id)`: devuelve cotización y fecha de la moneda indicada como parámetro

Atributos

- `Token`: es el código de autorización generado por la AFIP (WSAA)
- `Sign`: es la firma de autorización generado por la AFIP (WSAA)
- `Cuit`: es el número de CUIT del emisor de facturas, formato string sin guiones.
- `AppServerStatus`, `DbServerStatus`, `AuthServerStatus`: estados de los servidores de AFIP (string ?OK? en caso de estar funcionales)
- `XmlRequest`, `XmlResponse`: requerimiento y respuesta XML (para depuración)
- `Respuesta`, `Obs`, `Reproceso`: valores complementarios que retorna `Aut`
- `CAE`, `Vencimiento`: CAE y Fecha de vencimiento autorización retorna `Aut`
- `Vencimiento`: fecha devuelta junto con el CAE

- CbteNro: número de comprobante (del comprobante autorizado)
- FechaCbte: fecha del comprobante (del comprobante recuperado devuelto por GetCmp)
- ImpTotal: importe total del comprobante (del comprobante recuperado devuelto por GetCmp)
- Version: versión de la interfase (ej. ?1.11?)

Errores

La interfase procesa los errores WSFEX y los informa con códigos numéricos (Err.Number ? vbObjectError = nº de error de AFIP) y su correspondiente descripción (Err.Description). Entre ellos los más comunes son:

- 500: Error interno DB - FEXGetLadt_CMP - get_last: problemas internos de los servidores de AFIP (se debe esperar a que lo solucionen).
- 505: Error general (lockeo): problemas internos de los servidores de AFIP (se debe esperar a que lo solucionen).
- 1000: Usuario no autorizado a realizar esta operación. Revisar Token y Sign obtenido del WSAA
- 1001: CUIT no autorizado. Revisar en el sitio de la AFIP si se asoció el CUIT del emisor al servicio de factura electrónica, con el correspondiente certificado.
- 1012,1014,1016,1017,1018,1019: Inconsistencia en los datos enviados. Revisar formatos (fechas, números, etc., usar fechas YYYYMMDD y números separados por puntos, ambos pasarlos como string para evitar errores). Revisar tipo de datos (tipo_cbte, tipo_doc) que correspondan a las tablas usadas por la AFIP. Revisar que los datos sean válidos para la operación a

Ver especificaciones técnicas del WSFEX para más información sobre el uso del servicio web, su formato, condiciones y errores.

Ejemplo

Autorizar una factura electrónica de exportación a modo de ejemplo (ficticia):

```
Dim WSAA As Object, WSFEX As Object

' Crear objeto interface Web Service Autenticación y Autorización
Set WSAA = CreateObject("WSAA")

' Generar un Ticket de Requerimiento de Acceso (TRA) para WSFEX
tra = WSAA.CreateTRA("wsfex")

' Especificar la ubicacion de los archivos certificado y clave privada
Path = CurDir() + "\"
' Certificado: certificado es el firmado por la AFIP
' ClavePrivada: la clave privada usada para crear el certificado
Certificado = "..\reingart.crt" ' certificado de prueba
ClavePrivada = "..\reingart.key" ' clave privada de prueba

' Generar el mensaje firmado (CMS)
cms = WSAA.SignTRA(tra, Path + Certificado, Path + ClavePrivada)

' Llamar al web service para autenticar: (Homologación)
ta = WSAA.CallWSAA(cms, "https://wsaahomo.afip.gov.ar/ws/services/LoginCms")

' Una vez obtenido, se puede usar el mismo token y sign por 24 horas
' (este período se puede cambiar)

' Crear objeto interface Web Service de Factura Electrónica de Exportación
Set WSFEX = CreateObject("WSFEX")
```

```

' Setear token y sing de autorización (pasos previos)
WSFEX.Token = WSAA.Token
WSFEX.Sign = WSAA.Sign

' CUIT del emisor (debe estar registrado en la AFIP)
WSFEX.Cuit = "20267565393"

' Conectar al Servicio Web de Facturación (homologación)
ok = WSFEX.Conectar("http://wswhomo.afip.gov.ar/WSFEX/service.asmx")

' Llamo a un servicio nulo, para obtener el estado del servidor (opcional)
WSFEX.Dummy
Debug.Print "appserver status", WSFEX.AppServerStatus
Debug.Print "dbserver status", WSFEX.DbServerStatus
Debug.Print "authserver status", WSFEX.AuthServerStatus

' Establezco los valores de la factura a autorizar:
tipo_cbte = 19 ' FC Expo (ver tabla de parámetros)
punto_vta = 7
' Obtengo el último número de comprobante y le agrego 1
cbte_nro = WSFEX.GetLastCMP(tipo_cbte, punto_vta) + 1 '16
fecha_cbte = Format(Date, "yyyymmdd")
tipo_expo = 1 ' tipo de exportación (ver tabla de parámetros)
permiso_existente = "N"
dst_cmp = 203 ' país destino
cliente = "Joao Da Silva"
cuit_pais_cliente = "500000000016"
domicilio_cliente = "Rua 76 km 34.5 Alagoas"
id_impositivo = "PJ54482221-1"
moneda_id = "012" ' para reales, "DOL" o "PES" (ver tabla de parámetros)
moneda_ctz = 0.5
obs_comerciales = "Observaciones comerciales"
obs = "Sin observaciones"
forma_pago = "30 dias"
incoterms = "FOB" ' (ver tabla de parámetros)
idioma_cbte = 1 ' (ver tabla de parámetros)
imp_total = "250.00"

' Creo una factura (internamente, no se llama al Webservice):
ok = WSFEX.CrearFactura(tipo_cbte, punto_vta, cbte_nro, fecha_cbte, _
    imp_total, tipo_expo, permiso_existente, dst_cmp, _
    cliente, cuit_pais_cliente, domicilio_cliente, _
    id_impositivo, moneda_id, moneda_ctz, _
    obs_comerciales, obs, forma_pago, incoterms, _
    idioma_cbte)

' Agrego un item:
codigo = "PR01"
ds = "Producto Tipo 1 Exportacion MERCOSUR ISO 9001"
qty = 2
precio = "125.00"
umed = 1 ' Ver tabla de parámetros (unidades de medida)
imp_total = "250.00" ' importe total final del artículo
' lo agrego a la factura (internamente, no se llama al Webservice):
ok = WSFEX.AgregarItem(codigo, ds, qty, umed, precio, imp_total)

' Agrego un permiso (ver manual para el desarrollador)
id = "99999AAXX999999A"
dst = 225 ' país destino de la mercaderia
ok = WSFEX.AgregarPermiso(id, dst)

' Agrego un comprobante asociado (ver manual para el desarrollador)

```

```

tipo_cbte_asoc = 19
punto_vta_asoc = 2
cbte_nro_asoc = 1
ok = WSFEX.AgregarCmpAsoc(tipo_cbte_asoc, punto_vta_asoc, cbte_nro_asoc)

'id = "99000000000100" ' número propio de transacción
' obtengo el último ID y le adiciono 1
' (advertencia: evitar overflow y almacenar!)
id = CStr(CCur(WSFEX.GetLastID()) + 1)

' Llamo al Webservice de Autorización para obtener el CAE
cae = WSFEX.Authorize(id)

' Verifico que no haya rechazo o advertencia al generar el CAE
If cae = "" Or WSFEX.Resultado <> "A" Then
    MsgBox "No se asignó CAE (Rechazado). Observación (motivos): " & _
        WSFEX.obs, vbInformation + vbOKOnly
ElseIf WSFEX.obs <> "" And WSFEX.obs <> "00" Then
    MsgBox "Se asignó CAE pero con advertencias. Observación (motivos): " & _
        WSFEX.obs, vbInformation + vbOKOnly
End If

' Imprimo pedido y respuesta XML para depuración (errores de formato)
' si es posible almacenar para referencias futuras
Debug.Print WSFEX.XmlRequest
Debug.Print WSFEX.XmlResponse

MsgBox "Resultado:" & WSFEX.Resultado & " CAE: " & cae & _
    " Reproceso: " & WSFEX.Reproceso & _
    " Obs: " & WSFEX.obs, vbInformation + vbOKOnly

' Muestro los eventos (mantenimiento programados y otros mensajes de la AFIP)
For Each evento In WSFEX.Eventos
    If evento <> "0: " Then
        MsgBox "Evento: " & evento, vbInformation
    End If
Next

' Recuperar la factura
cae2 = WSFEX.GetCMP(tipo_cbte, punto_vta, cbte_nro)

Debug.Print "Fecha Comprobante:", WSFEX.FechaCbte
Debug.Print "Importe Total:", WSFEX.ImpTotal

If cae <> cae2 Then
    MsgBox "El CAE de la factura no concuerdan con el recuperado en la AFIP!"
Else
    MsgBox "El CAE de la factura concuerdan con el recuperado de la AFIP"
End If

```

Nota: no es necesario (ni recomendado) recuperar el último ID y/o último número de comprobante. Dichos datos deberían estar almacenados de manera permanente en la aplicación que use la interfaz([ver aclaración](#)).

Ejemplo para obtener códigos referenciales de países destino habilitados:

```

' recupero tabla de código de pais destino ("codigo: descripción")
' Ej: "203: BRASIL"
For Each x In WSFEX.GetParamDstPais()
    Debug.Print x
Next

```

Ejemplo para obtener la fecha y cotización de una moneda:

```
' busco la cotización del dolar (ver Parametro Mon)
' Ej: "20100708: 3.937"
moneda_id = "DOL"
ctz = WSFEX.GetParamCtz(moneda_id)
MsgBox "Cotización Dólar: " & ctz
```

Descripción de la obtención de CAE

Como primer paso, se debe crear una factura (utilizada internamente por la interfase para contener los valores a autorizar), llamando al método CrearFactura con los siguiente parámetros:

- tipo_cbte: código de comprobante (19: 'Facturas de Exportación', 20: 'Nota de Débito por Operaciones con el Exterior', 21: 'Nota de Crédito por Operaciones con el Exterior')
- punto_vta: N° de punto de venta (debe estar autorizado para WSFEX)
- cbte_nro: N° de comprobante
- fecha_cbte: Fecha del comprobante (no puede ser mayor o menor a 10 días)
- imp_total: Importe total de la factura
- tipo_expo: Tipo de exportacion (1: 'Exportación definitiva de Bienes', 2: 'Servicios', 4: 'Otros')
- permiso_existente: Indica si se posee documento aduanero de exportación (permiso de embarque). Posibles Valores: 'S', 'N', 'NULL' (vacío)
- dst_cmp: País de destino del comprobante (200: 'ARGENTINA', 203: 'BRASIL', 212: 'ESTADOS UNIDOS', etc.)
- cliente: Apellido y Nombre ó Razón Social del comprador
- cuit_pais_cliente: CUIT del país destino/Contribuyente (Ej: 50000000059: 'BRASIL - Persona F\xedsica', 51600000059: 'BRASIL - Otro tipo de Entidad', etc.)
- domicilio_cliente: Domicilio comercial cliente.
- id_impositivo: Clave de identificación tributaria del comprador.
- moneda_id: Moneda de la factura ('DOL': 'Dólar Estadounidense', 'PES': 'Pesos Argentinos', '012': 'Real', etc.)
- moneda_ctz: Cotización de la moneda de la factura
- obs_comerciales: observaciones comerciales (texto arbitrario)
- obs: observaciones (texto arbitrario)
- forma_pago: texto arbitrario (ej '30 días')
- incoterms: clausula de venta, terminos de comercio exterior ('DAF', 'DDP', 'CIF', 'FCA', 'FAS', 'DES', 'CPT', 'EXW', 'CIP', 'DDU', 'FOB', 'DEQ', 'CFR')
- idioma_cbte: idioma del comprobante {1: 'Español', 2: 'Inglés', 3: 'Portugués'}

Luego, por cada artículo vendido (ítem), se debe llamar al método AgregarItem con los siguientes parámetros:

- codigo: código del producto
- ds: Descripción completa
- precio: Precio Unitario
- qty: Cantidad
- umed: Unidad de medida (según tabla de parámetros)
- imp_total: Importe total

Adicionalmente se puede llamar al método AgregarPermiso para detallar los permisos de embarque y destinos de la mercadería, con los siguientes parámetros:

- id: Código de despacho ? Permiso de Embarque, formato 99999AAXX999999A (donde XX podrán ser números o letras)
- dst: País de destino de la mercadería

También se puede llamar al método AgregarCmpAsoc para detallar los permisos de embarque y destinaciones de la mercadería, con los siguientes parámetros:

- cbte_tipo: Código de tipo de comprobante.
- cbte_punto_venta: Punto de venta
- vbte_numero: Numero de comprobante

Como último paso, se debe llamar al método Authorize con los siguiente parámetros:

- id: Número de identificación secuencial (debe almacenarse en el sistema local) No es obligatorio llamar a WSFEX.GetLastID, puede utilizarse un dato local secuencial.

Si no hubiere inconvenientes, la llamada debe devolver el CAE y se establece el atributo WSFEX.Resultado = "A" (Aceptado) y WSFEX.Vencimiento. Sino, devuelve:

- cae = "": La interfase no pudo procesar la respuesta del Webservice, o bien la conexión a internet esta caída, las direcciones de los servidores son incorrectas, o el servicio web rechazo los datos de la factura a generar. Se establece el atributo WSFEX.Resultado = "R" (Rechazado) y WSFEX.Obs con los diversos motivos de rechazo proporcionados por el webservice (ver WSFE.Motivo):

Aún devolviendo el CAE y con el resultado Aceptado, puede haber un motivo (ej 01), que funcionan como advertencia para revisar la base de datos y corregir posibles irregularidades.

Advertencia: Revisar las consideraciones de reproceso al igual que con WSFE.

IMPORTANTE: Revisar las especificaciones técnicas de la AFIP en el [Manual para el desarrollador.pdf](#) por lo relativo a tablas de parámetros, métodos adicionales y operatoria en general (incluyendo obligatoriedad, formatos de los campos y validaciones)

Interfase por archivos de texto símil SIAP - RECE

Para lenguajes donde no es posible utilizar objetos COM, como en algunas versiones de COBOL, se desarrolló una interfase de texto para poder utilizar los Web Services de la AFIP, que funciona como un programa independiente, manteniendo las ventajas y características presentadas anteriormente.

Configuración

Editar el archivo RECE.INI en la carpeta de la intefase (C:\PYAFIPWS):

- CERT: ubicación del archivo certificado (ver WSAA)
- PRIVATEKEY: ubicación del archivo de la clave privada (ver WSAA)
- CUIT: CUIT del emisor
- ENTRADA: ubicación del archivo de texto de entrada
- SALIDA: ubicación del archivo de texto de salida
- URL: dirección de los servicios web de producción (ver WSAA y WSFE)

Ejemplo:

```
[WSAA]
CERT=C:\SISTEMA\empresa.crt
PRIVATEKEY= C:\SISTEMA\empresa.key
URL=https://wsaa.afip.gov.ar/ws/services/LoginCms

[WSFE]
CUIT=30000000000
ENTRADA= C:\SISTEMA\entrada.txt
SALIDA= C:\SISTEMA\salida.txt
URL=https://wsw.afip.gov.ar/wsfe/service.asmx

[WSBFE]
CUIT=30000000000
ENTRADA= C:\SISTEMA\entrada.txt
SALIDA= C:\SISTEMA\salida.txt
URL=https://wsw.afip.gov.ar/wsbfe/service.asmx
```

Formato de archivos de entrada y salida

Factura electrónica normal (WSFE):

El formato utilizado es similar a RECE pero extendido con algunos campos necesarios para utilizar los Servicios Web. A continuación se detalla nombre del campo, posición inicial, longitud y tipo.

- fecha_cbte: (2,8,A)
- tipo_cbte: (10,2,N)
- punto_vta: (13,4,N)
- cbt_desde: (17,8,N)
- cbt_hasta: (25,8,N)
- tipo_doc: (36,2,N)
- nro_doc: (38,11,N)
- imp_total: (79,15,I)
- imp_tot_conc: (94,15,I)
- imp_netto: (109,15,I)
- impto_liq: (124,15,I),
- impto_liq_rni: (139,15,I)
- imp_op_ex: (154,15,I)
- cae: (261,14,N)
- fecha_vto: (275,8,A)
- resultado: (291,1,A)
- motivo: (292,2,A)
- reproceso: (294,1,A)
- fecha_venc_pago: (295,8,A)
- presta_serv: (303,1,N)
- fecha_serv_desde: (304,8,A)
- fecha_serv_hasta: (312,8,A)
- id: (320,15,N)

Tipos de campo:

- A: Alfanumerico

- N: Numerico
- I: Importes con 2 decimales (sin incluir punto)

Los espacios no definidos se ignoran (completar con un espacio en blanco)

Para más información sobre los campos, ver parámetros y valores que retorna el método Aut en el presente manual

Como formato de entrada es posible utilizar el mismo archivo generado para el aplicativo SIAP/RECE, pero no se podrán facturar servicios ya que los campo presta_serv y fecha_serv_desde y fecha_serv_hasta no están contemplados en dicho formato.

Factura electrónica bienes de capital (WSBFE)

El formato es similar al de factura electrónica normal, pero varía el tipo de registro (0: Encabezado, 1: Detalle de cada Item):

Encabezado:

- Campo: tipo_reg Posición: 1 Longitud: 1 Tipo: Numerico Valor: 0
- Campo: fecha_cbte Posición: 2 Longitud: 8 Tipo: Alfanumerico
- Campo: tipo_cbte Posición: 10 Longitud: 2 Tipo: Numerico
- Campo: punto_vta Posición: 12 Longitud: 4 Tipo: Numerico
- Campo: cbte_nro Posición: 16 Longitud: 8 Tipo: Numerico
- Campo: tipo_doc Posición: 24 Longitud: 2 Tipo: Numerico
- Campo: nro_doc Posición: 26 Longitud: 11 Tipo: Numerico
- Campo: imp_total Posición: 37 Longitud: 15 Tipo: Importe
- Campo: imp_tot_conc Posición: 52 Longitud: 15 Tipo: Importe
- Campo: imp_netto Posición: 67 Longitud: 15 Tipo: Importe
- Campo: imppto_liq Posición: 82 Longitud: 15 Tipo: Importe
- Campo: imppto_liq_rni Posición: 97 Longitud: 15 Tipo: Importe
- Campo: imp_op_ex Posición: 112 Longitud: 15 Tipo: Importe
- Campo: imppto_perc Posición: 127 Longitud: 15 Tipo: Importe
- Campo: imp_iibb Posición: 142 Longitud: 15 Tipo: Importe
- Campo: imppto_perc_mun Posición: 157 Longitud: 15 Tipo: Importe
- Campo: imp_internos Posición: 172 Longitud: 15 Tipo: Importe
- Campo: imp_moneda_id Posición: 187 Longitud: 3 Tipo: Alfanumerico
- Campo: imp_moneda_ctz Posición: 190 Longitud: 10 Tipo: Importe
- Campo: zona Posición: 200 Longitud: 5 Tipo: Alfanumerico
- Campo: cae Posición: 205 Longitud: 14 Tipo: Numerico
- Campo: fecha_vto Posición: 219 Longitud: 8 Tipo: Alfanumerico
- Campo: resultado Posición: 227 Longitud: 1 Tipo: Alfanumerico
- Campo: obs Posición: 228 Longitud: 2 Tipo: Alfanumerico
- Campo: reproceso Posición: 230 Longitud: 1 Tipo: Alfanumerico
- Campo: id Posición: 231 Longitud: 15 Tipo: Numerico

Detalle (por cada item):

- Campo: tipo_reg Posición: 1 Longitud: 1 Tipo: Numerico Valor: 1
- Campo: ncm Posición: 2 Longitud: 15 Tipo: Alfanumerico
- Campo: sec Posición: 17 Longitud: 15 Tipo: Alfanumerico

Factura electrónica normal (WSFE):

- Campo: qty Posición: 32 Longitud: 15 Tipo: Importe
- Campo: umed Posición: 47 Longitud: 5 Tipo: Numerico
- Campo: precio Posición: 52 Longitud: 15 Tipo: Importe
- Campo: bonif Posición: 67 Longitud: 15 Tipo: Importe
- Campo: imp_total Posición: 82 Longitud: 15 Tipo: Importe
- Campo: iva_id Posición: 97 Longitud: 5 Tipo: Numerico
- Campo: ds Posición: 102 Longitud: 200 Tipo: Alfanumerico

Si se informa el campo ID, la factura se genera con dicho número de secuencia. De lo contrario, para evitar problemas y poder reprocesar el CAE, se genera con un número de secuencia del formato TPPPPNNNNNNNN (tipo de comprobante, punto de venta y número de comprobante).

Factura electrónica exportación (WSFEX)

El formato es similar al de factura electrónica bienes de capital, pero varía el tipo de registro (0: Encabezado, 1: Detalle de cada Item, 2: Permiso, 3: Comprobante asociado):

Formato:

Encabezado:

- Campo: tipo_reg Posición: 1 Longitud: 1 Tipo: Numerico Decimales:
- Campo: fecha_cbte Posición: 2 Longitud: 8 Tipo: Alfanumerico Decimales:
- Campo: tipo_cbte Posición: 10 Longitud: 2 Tipo: Numerico Decimales:
- Campo: punto_vta Posición: 12 Longitud: 4 Tipo: Numerico Decimales:
- Campo: cbte_nro Posición: 16 Longitud: 8 Tipo: Numerico Decimales:
- Campo: tipo_expo Posición: 24 Longitud: 1 Tipo: Numerico Decimales:
- Campo: permiso_existente Posición: 25 Longitud: 1 Tipo: Alfanumerico Decimales:
- Campo: dst_cmp Posición: 26 Longitud: 3 Tipo: Numerico Decimales:
- Campo: cliente Posición: 29 Longitud: 200 Tipo: Alfanumerico Decimales:
- Campo: cuit_pais_cliente Posición: 229 Longitud: 11 Tipo: Numerico Decimales:
- Campo: domicilio_cliente Posición: 240 Longitud: 300 Tipo: Alfanumerico Decimales:
- Campo: id_impositivo Posición: 540 Longitud: 50 Tipo: Alfanumerico Decimales:
- Campo: imp_total Posición: 590 Longitud: 15 Tipo: Importe Decimales: 3
- Campo: moneda_id Posición: 605 Longitud: 3 Tipo: Alfanumerico Decimales:
- Campo: moneda_ctz Posición: 608 Longitud: 10 Tipo: Importe Decimales: 6
- Campo: obs_comerciales Posición: 618 Longitud: 1000 Tipo: Alfanumerico Decimales:
- Campo: obs Posición: 1618 Longitud: 1000 Tipo: Alfanumerico Decimales:
- Campo: forma_pago Posición: 2618 Longitud: 50 Tipo: Alfanumerico Decimales:
- Campo: incoterms Posición: 2668 Longitud: 3 Tipo: Alfanumerico Decimales:
- Campo: incoterms_ds Posición: 2671 Longitud: 20 Tipo: Alfanumerico Decimales:
- Campo: idioma_cbte Posición: 2691 Longitud: 1 Tipo: Alfanumerico Decimales:
- Campo: cae Posición: 2692 Longitud: 14 Tipo: Numerico Decimales:
- Campo: fecha_vto Posición: 2706 Longitud: 8 Tipo: Alfanumerico Decimales:
- Campo: resultado Posición: 2714 Longitud: 1 Tipo: Alfanumerico Decimales:
- Campo: reproceso Posición: 2715 Longitud: 1 Tipo: Alfanumerico Decimales:
- Campo: motivos_obs Posición: 2716 Longitud: 40 Tipo: Alfanumerico Decimales:
- Campo: id Posición: 2756 Longitud: 15 Tipo: Numerico Decimales:
- Campo: fch_venc_cae Posición: 2771 Longitud: 8 Tipo: Alfanumerico Decimales:

Detalle:

- Campo: tipo_reg Posición: 1 Longitud: 1 Tipo: Numerico Decimales:
- Campo: codigo Posición: 2 Longitud: 30 Tipo: Alfanumerico Decimales:
- Campo: qty Posición: 32 Longitud: 12 Tipo: Importe Decimales:
- Campo: umed Posición: 44 Longitud: 2 Tipo: Numerico Decimales:
- Campo: precio Posición: 46 Longitud: 12 Tipo: Importe Decimales: 3
- Campo: imp_total Posición: 58 Longitud: 14 Tipo: Importe Decimales: 3
- Campo: ds Posición: 72 Longitud: 4000 Tipo: Alfanumerico Decimales:

Permiso:

- Campo: tipo_reg Posición: 1 Longitud: 1 Tipo: Numerico Decimales:
- Campo: id_permiso Posición: 2 Longitud: 16 Tipo: Alfanumerico Decimales:
- Campo: dst_merc Posición: 18 Longitud: 3 Tipo: Numerico Decimales:

Comprobante Asociado:

- Campo: tipo_reg Posición: 1 Longitud: 1 Tipo: Numerico Decimales:
- Campo: cbte_tipo Posición: 2 Longitud: 3 Tipo: Numerico Decimales:
- Campo: cbte_punto_vta Posición: 5 Longitud: 4 Tipo: Numerico Decimales:
- Campo: cbte_nro Posición: 9 Longitud: 8 Tipo: Numerico Decimales:

Nota: se utilizan 3 decimales para los importes y 6 para la cotización de la moneda

Forma de uso

Llamar al ejecutable RECE.EXE (factura electrónica normal, WSFE), RECEB.EXE (factura electrónica Bienes de Capital, WSBFE) o RECEX.EXE (factura electrónica Exportación) en la carpeta de la interfase (C:\PYAFIPWS)

En caso de ejecución correcta, informara por pantalla los ID y CAE obtenidos y el código de retorno es 0:

```
ID: 1000200000063 CAE: 59033567692234 Motivo: 00 Reproceso: N
ID: 1000200000064 CAE: 59033693441244 Motivo: 00 Reproceso: N
ID: 1000200000065 CAE: 59033880026543 Motivo: 00 Reproceso: N
```

El CAE obtenido, fecha de vencimiento y demás valores devueltos por WSFE (resultado, reproceso, motivo de rechazo u observación) son escritos en el archivo de salida, con la misma información de la factura que en el archivo de entrada.

En el caso de error, informa por pantalla el motivo y el código de retorno es distinto de 0:

ns1:coe.notAuthorized Computador no autorizado a acceder los servicio de AFIP

1000 FERecuperaLastCMPRequest@SRV_RA: Usuario no autorizado a realizar esta operacion.
ValidacionDeToken: No apareció CUIT en lista de relaciones: 30000000000

Parámetros

RECE.EXE, RECEB.EXE, RECEX.EXE reciben los siguientes argumentos por línea de comando:

- /ayuda: lista los parámetros habilitados
- /prueba: Teniendo los certificados instalados, se puede realizar una prueba donde la interface generará un archivo de entrada para las tres próximas facturas, obteniendo los últimos números de transacción y comprobante.
- /ult: Solicita Tipo de comprobante y Punto de Venta y devuelve el último numero de comprobante registrado
- /dummy: consulta estado de servidores (deberían ser OK los 3 servidores)
- /debug: modo depuración (detalla y confirma las operaciones)
- /formato: muestra el formato de los archivos de entrada/salida
- /get: recupera datos de un comprobante autorizado previamente (verificación, solo para WSFEX)
- /xml: almacena los requerimientos y respuestas XML (útil para depuración y registro)

Ejemplo:

```
C:\PYAFIPWS> RECEX.EXE /ult
Consultar ultimo numero:
Tipo de comprobante: 1
Punto de venta: 2
Ultimo numero: 62
```

Entorno

Algunas versiones de COBOL cambian el directorio u otras variables del entorno (codificación, zona horaria, etc.). En este caso se recomienda armar un archivo RECE.BAT:

```
@ECHO OFF
REM limpiar la zona horaria y cambiar al directorio de la interfase
SET TZ=
CD C:\PYAFIPWS
RECE.EXE
```

Régimen de Almacenamiento de Duplicados Digitales (RG1361)

Con la interfaz se provee un utilitario para generar los archivos requeridos por el aplicativo SIREDA (SIAP) de la Resolución General 1361/02, referente al almacenamiento digital de los comprobantes emitidos (Libro Ventas, Detalle y Cabeceras de Factura).

El formato de entrada es una planilla csv similar al [PyRece](#), respetando los tipos de datos y códigos requeridos por los webservices, con las siguientes columnas: tipo_cbte, punto_vta, cbt_numero, fecha_cbte, tipo_doc, nro_doc, imp_netto, impto_liq, imp_total, cae, fecha_vto, nombre.

Ejemplo de uso:

```
CD C:\PYAFIPWS
RG1361.EXE planilla.csv
```

Generará los archivos:

- VENTAS_200902.txt
- CABECERA_200902.txt

- DETALLE_200902.txt

Código de Trazabilidad de Granos

wsctg es la Interfaz para generar Código de Trazabilidad de Granos AFIP. Se invoca llamando a wsctg.py o wsctg.exe o a través de la interfaz COM PyAfipWs.

Opciones

Por línea de comando, es posible pasarle los siguientes parámetros:

```
--ayuda: este mensaje

--debug: modo depuración (detalla y confirma las operaciones)
--formato: muestra el formato de los archivos de entrada/salida
--prueba: genera y autoriza una CTG de prueba (no usar en producción!)
--xml: almacena los requerimientos y respuestas XML (depuración)

--dummy: consulta estado de servidores
--solicitar: obtiene el CTG
--confirmar: confirma el CTG

--provincias: obtiene el listado de provincias
--localidades: obtiene el listado de localidades por provincia
--especies: obtiene el listado de especies
--cosechas: obtiene el listado de cosechas
```

Formato de Archivos

Archivo de texto en formato csv con las siguientes columnas:

- codigoLocalidadDestino: entrada solicitar_ctg, ej. '3059'
- numeroCTG: salida solicitar_ctg, entrada confirmar_ctg, ej. '43816783'
- transaccion: ej. '10000001681'
- cuitDestino: entrada solicitar_ctg, ej. '20267565393'
- codigoLocalidadOrigen: entrada solicitar_ctg, ej. '3058'
- cuitRemitenteComercial: entrada solicitar_ctg, ej. '30640872566'
- observaciones: salida confirmar_ctg, ej. ''
- patenteVehiculo: entrada solicitar_ctg, ej. 'CZO985'
- codigoEspecie: entrada solicitar_ctg, ej. '23'
- cuitTransportista: entrada solicitar_ctg, entrada confirmar_ctg, ej. '20234455967'
- cuitDestinatario: entrada solicitar_ctg, ej. '20267565393'
- codigoCosecha: entrada solicitar_ctg, ej. '0910'
- pesoNetoCarga: entrada solicitar_ctg, entrada confirmar_ctg, ej. '1000'
- cantHoras: entrada solicitar_ctg, ej. '1'
- numeroCartaDePorte: entrada solicitar_ctg, entrada confirmar_ctg, ej. '512345678'

Configuración

Ver wsctg.ini para parámetros de configuración (URL, certificados, etc.). Ej:

```
[WSAA]
```

```

CERT=reingart.crt
PRIVATEKEY=reingart.key
#URL=https://wsaa.afip.gov.ar/ws/services/LoginCms

[WSCTG]
CUIT=20267565393
ENTRADA=entrada.txt
SALIDA=salida.txt
#URL=https://cereales.afip.gov.ar/wsctg/services/CTGService

```

Ejemplo Interfaz COM WSCTG

De igual manera que el resto de los servicios web, el web service del código de trazabilidad de granos puede ser invocado desde lenguajes visuales y no visuales (en windows) utilizando la interfaz COM:

Ejemplo en Visual Basic (similar para Visual FoxPro?, Delphi, etc.):

```

' Crear objeto interface Web Service de Código de Trazabilidad de Granos
Set WSCTG = CreateObject("WSCTG")
' Setear token y sing de autorización (pasos previos)
WSCTG.token = WSAA.token
WSCTG.sign = WSAA.sign

' CUIT (debe estar registrado en la AFIP)
WSCTG.cuit = "20267565393"

' Conectar al Servicio Web Código trazabilidad de granos (homologación)
ok = WSCTG.Conectar("https://fwshomo.afip.gov.ar/wsctg/services/CTGService")

' Llamo a un servicio nulo, para obtener el estado del servidor (opcional)
WSCTG.Dummy
Debug.Print "appserver status", WSCTG.AppServerStatus
Debug.Print "dbserver status", WSCTG.DbServerStatus
Debug.Print "authserver status", WSCTG.AuthServerStatus

numero_carta_de_porte = "512345678"
codigo_especie = 23
cuit_remitente_comercial = "20267565393"
cuit_destino = "20267565393"
cuit_destinatario = "20267565393"
codigo_localidad_origen = 3058
codigo_localidad_destino = 3059
codigo_cosecha = "0910"
peso_netto_carga = 1000
cant_horas = 1
patente_vehiculo = "AAA000"
cuit_transportista = "2026756539"

numero_CTG = WSCTG.SolicitarCTG(numero_carta_de_porte, codigo_especie, _
    cuit_remitente_comercial, cuit_destino, cuit_destinatario, codigo_localidad_origen, _
    codigo_localidad_destino, codigo_cosecha, peso_netto_carga, cant_horas, _
    patente_vehiculo, cuit_transportista)

' Guardar la respuesta XML
Debug.Print WSCTG.XmlResponse

MsgBox numero_CTG, vbInformation, "SolicitarCTG: número CTG:"

numero_CTG = "43816783"

```

```

transaccion = WSCTG.ConfirmarCTG(numero_carta_de_porte, numero_CTG, cuit_transportista, peso_neto_carga)
' Guardar la respuesta XML
Debug.Print WSCTG.XmlResponse

MsgBox WSCTG.Observaciones, vbInformation, "ConfirmarCTG: código transaccion:" & self.CodigoTransaccion

Debug.Assert transaccion = "10000001681"

```

wDigDepFiel: Depositario Fiel

Servicio Web de AFIP para Aviso de Recepción y Aceptación de la Digitalización de Documentos por parte de los Prestadores de Servicios de Archivos y Digitalización (PSAD) y Aviso de Digitalización de Documentos por parte de los Despachantes o de los Prestadores de Servicios de Archivos y Digitalización (PSAD) -

Sujetos Alcanzados: ver Resolución General 2570/2009

Actualmente la interfaz expone las funcionalidades de este webservice:

- Aviso de Digitalización
- Aviso de Recepción y Aceptación
- Método Dummy para verificación de funcionamiento

Ejemplo Interfaz COM wDigDepFiel

De igual manera que el resto de los servicios web, el web service de depositario fiel puede ser invocado desde lenguajes visuales y no visuales (en windows) utilizando la interfaz COM:

Ejemplo en Visual Basic (similar para Visual FoxPro?, Delphi, etc.):

```

' Crear objeto interface Web Service de Factura Electrónica de Exportación
Set wDigDepFiel = CreateObject("wDigDepFiel")
' Setear token y sing de autorización (pasos previos)
wDigDepFiel.token = WSAA.token
wDigDepFiel.sign = WSAA.sign

' CUIT del emisor (debe estar registrado en la AFIP)
wDigDepFiel.cuit = "20267565393"

' Conectar al Servicio Web de Depositario Fiel (homologación)
ok = wDigDepFiel.Conectar("https://testdia.afip.gov.ar/Dia/Ws/wDigDepFiel/wDigDepFiel.asmx")

' Llamo a un servicio nulo, para obtener el estado del servidor (opcional)
wDigDepFiel.Dummy
Debug.Print "appserver status", wDigDepFiel.AppServerStatus
Debug.Print "dbserver status", wDigDepFiel.DbServerStatus
Debug.Print "authserver status", wDigDepFiel.AuthServerStatus

tipo_agente = "PSAD" ' "DESP"
rol = "EXTE"
nro_legajo = "000000000000000000"
cuit_declarante = "20267565393"
cuit_psad = "20267565393"
cuit_ie = "20267565393"
codigo = "000" ' carpeta completa, "001" carpeta adicional

```

```

' Fecha/Hora en formato ISO: "2010-06-07T00:23:51.750000"
fecha_hora_acept = Format(Now(), "yyyy-MM-dd") & "T" & _
                    Format(Now(), "hh:mm:ss") & ".000000"
ticket = "1234"
errCode = wDigDepFiel.AvisoRecepAcept(tipo_agente, rol, _
                    nro_legajo, cuit_declarante, cuit_psad, cuit_ie, _
                    codigo, fecha_hora_acept, ticket)
Debug.Print wDigDepFiel.XmlResponse

MsgBox wDigDepFiel.DescError, vbInformation, _
        "AvisoRecepAcept Código Error: " & wDigDepFiel.CodError

tipo_agente = "PSAD" ' "DESP"
rol = "EXTE"
nro_legajo = "0000000000000000" ' "1234567890123456"
cuit_declarante = "20267565393"
cuit_psad = "20267565393"
cuit_ie = "20267565393"
cuit_ata = "20267565393"
codigo = "000" ' carpeta completa, "001" carpeta adicional
ticket = "1234"
url = "http://www.example.com"
hashing = "db1491eda47d78532cdfca19c62875aade941dc2"

' inicializo aviso: limpio datos (familias)
wDigDepFiel.IniciarAviso
codigo = "02"
cantidad = 1
wDigDepFiel.AgregarFamilia codigo, cantidad
codigo = "03"
cantidad = 3
wDigDepFiel.AgregarFamilia codigo, cantidad

cantidad_total = 4

errCode = wDigDepFiel.AvisoDigit(tipo_agente, rol, _
                    nro_legajo, cuit_declarante, cuit_psad, cuit_ie, cuit_ata, _
                    codigo, url, ticket, hashing, cantidad_total):

Debug.Print wDigDepFiel.XmlResponse

MsgBox wDigDepFiel.DescError, vbInformation, _
        "AvisoDigit Código: " & wDigDepFiel.CodError

```

Certificados

Generación

Para poder utilizar la interfase se deben tramitar y asociar los certificados de homologación/producción en la AFIP. Para mas información ver [Página principal de Factura Electrónica \(AFIP\)](#)

Pasos para crear el certificado (más información en [Instructivo AFIP](#)):

- Bajar e instalar [OpenSSL para windows](#) (en caso de inconvenientes, instalar [Redistributable de Visual C++](#))
- Ingresar por línea de comando al directorio de OpenSSL C:\OpenSSL\bin>
- Generar la clave privada:

```
openssl genrsa -out empresa.key 1024
```

- Generar el pedido (CSR: certificate signing request) por línea de comando:

```
openssl req -new -key empresa.key -subj "/C=AR/O=[empresa]/CN=[nombre]/serialNumber=CUIT [nro_cuit]" -out e
```

- ◆ Reemplazar [empresa] por el nombre de su empresa. Ej. *Nombre de la empresa tal cual figura en la consulta de inscripción, ej.: Empresa S A*
- ◆ Reemplazar [nombre] por su nombre o server hostname. *Nombre del servicio, aplicación u unidad operativa, ej: Interfaz PyAfipWs*
- ◆ Reemplazar [nro_cuit] por la CUIT sin guiones de la empresa o programador. Ej. **CUIT xxxxxxxxxxxx**
- Enviar el *empresa.csr* a la AFIP para que lo firmen y devuelvan el certificado *empresa.crt*.
 - ◆ Homologación: enviarlo por email a la AFIP (webservices@afip...).
 - ◆ Producción: Ingresar por clave fiscal al servicio online "Administración de Certificados Digitales", Ir a "Agregar Alias", escribir el alias (nombre relacionado al certificado), subir el archivo .CSR presionar "Agregar" y luego buscar y descargar el certificado CRT.
- Con esta interfase no es necesario convertir el certificado en formato pkcs12 ni importarlo al repositorio de Windows

Asociación

Para poder usar los servicios web se debe asociar el certificado al webservice y CUIT (empresa) que lo utilizará:

- Homologación: solicitando acceso a webservices@afip.gov.ar (indicar CUIT y nombre del servicio)
- Producción: utilizando el servicio de clave fiscal "ARFE - Gestionar Relaciones", ir a "Realizar una nueva Asociación" completando Empresa, Servicio y computador -certificado- que utilizara los webservices

Pre-requisitos para Producción (RECE)

Para poder usar Factura Electrónica en Producción (modo definitivo), además de los puntos mencionados anteriormente, dentro de Clave Fiscal, servicio online "Regimenes de facturación y registración (REAR/RECE/RFI)" es necesario:

- Adherirse al régimen (realizarlo con anterioridad, este trámite puede demorar varias semanas). Importante: en el "Rubro I. C) RECE / RCEL (Factura Electrónica y Factura Electrónica en Línea)" especificar *RECE* (no RCEL que es para "Comprobantes en línea" y Facturador Plus" no contemplados por los webservices)
- Dar de alta los nuevos puntos de ventas (uno por cada webservice a utilizar: factura nacional, exportación, bono fiscal, etc.)

Descargas

- [OpenSSL para windows](#)
- [Redistribuye VC++](#)

Comprobación de Autenticación (WSAA)

Para descartar problemas técnicos de acceso en producción u homologación (y verificar la validez del certificado generado), se puede utilizar la herramienta WSAA . EXE en modo depuración (sin necesidad de autorizar una factura, solo solicitar acceso) y/o el mensaje XML devuelto por WSAA . CallWSAA () :

```
C:\PYAFIP\ws>python wsaa.py reingart.crt reingart.key wsfex 2400
                https://wsaahomo.afip.gov.ar/ws/services/LoginCms --trace

Usando CERT=reingart.crt PRIVATEKEY=reingart.key
        URL=https://wsaa.afip.gov.ar/ws/services/LoginCms
        SERVICE=wsfex TTL=2400
Creando TRA...
-----
<?xml version="1.0" encoding="UTF-8"?>
<loginTicketRequest version="1.0">
  <header>
    <uniqueId>1279058341</uniqueId>
    <generationTime>2010-07-13T18:19:01</generationTime>
    <expirationTime>2010-07-13T19:39:01</expirationTime>
  </header>
  <service>wsfex</service>
</loginTicketRequest>
-----

Frimando TRA...
Llamando WSAA...
-----

POST https://wsaa.afip.gov.ar/ws/services/LoginCms
SOAPAction: "http://ar.gov.afip.dif.facturaelectronica/loginCms"
Content-length: 3369
Content-type: text/xml; charset="UTF-8"

...
=====
El archivo TA.xml se ha generado correctamente.
```

Para descartar cualquier problema técnico se puede utilizar la salida que es la traza de depuración de la comunicación con AFIP a la mesa de ayuda (requerimiento y respuesta xml).

Si se estaría conectando correctamente a producción, el certificado y las URL son correctas, etc., se debería constatar que:

- El ticket de requerimiento de acceso (TRA) correctamente solicita servicio correcto (wsfex):
<service>wsfex</service>
- Se está solicitando acceso a los servidores del ambiente correcto (wsaa_url): POST
https://wsaa.afip.gov.ar/ws/services/LoginCms (producción) o POST
http://wsaahomo.afip.gov.ar/ws/services/LoginCms (homologación)
- El equipo que responde sería del ambiente correcto: pereza.afip.gov.ar,
envidia.afip.gov.ar (producción) o avaricia.afip.gov.ar (homologación). Nota: los
nombres de los servidores pueden cambiar, y solo se informan en caso de error.

Si la autenticación es exitosa (El archivo TA.xml se ha generado correctamente., en el mensaje XML contenido en dicho archivo se puede verificar los siguientes datos:

- Fuente: AFIP producción (<source>CN=wsaa, O=AFIP, C=AR, SERIALNUMBER=CUIT

33693450239</source>), sinó diría CN=wsaahomo, O=AFIP, C=AR,
SERIALNUMBER=CUIT 33693450239

- Destino: datos del certificado de producción/homologación (C=ar, O=pyafipws-sistemas agiles, SERIALNUMBER=CUIT 20267565393, CN=mariano reingart)

Si el ambiente es el correcto (producción/homologación) y sigue devolviendo acceso denegado:
nsl:coe.notAuthorized: Computador no autorizado a acceder al servicio (gov.afip.desein.dvadac.sua.view.wsa.LoginFault), revisar el certificado, ARFE (asociación de servicio), RECE/REAR/RFI sobre régimen habilitado, puntos de venta, y demás trámites ante AFIP.

Errores Frecuentes

Los siguientes son posibles interpretaciones y soluciones a los errores que envían los servidores de AFIP.

Falla SOAP: ns1:coe.notAuthorized

En WSAA, descartando que se este usando el certificado del ambiente incorrecto (certificado de homologación en producción y viceversa), es necesario que el certificado y CUIT estén habilitados para acceder al servicio web especificado (wsfe, wsflex, wsbf, wsctg, wDigDepFiel, etc.). No es suficiente generar el certificado (en homologación o en producción), hay que solicitar explícitamente acceso al servicio requerido: ver [Asociación de Certificado](#)

Los servicios no relacionados a factura electrónica, en producción tienen otros métodos para asociar los certificados.

Ver [Comprobación de Autenticación WSAA](#) con información para determinar o depurar el problema de acceso.

1000 Usuario no autorizado a realizar esta operacion. ValidacionDeToken: No aparece CUIT en lista de relaciones:

En los web services de negocio (WSFE, WSFEX, WSBFE, etc.), se está accediendo con un ticket de acceso válido, pero se intenta operar con un CUIT incorrecto y/o que no se corresponde al certificado. Revisar que el CUIT sea correcto y que esté asociado al servicio y certificado (ver "ARFE" en punto anterior)

1000 Usuario no autorizado a realizar esta operacion. ValidacionDeToken: Error al verificar hash: VerificacionDeHash: No valida la firma digital.

En los web services de negocio (WSFE, WSFEX, WSBFE, etc.), se está accediendo con un ticket de acceso inválido para el ambiente requerido (por ej., ticket de acceso de homologación usado en producción), o, no se envió correctamente el token y sign.

Falla SOAP: Server was unable to read request. ---> There is an error in XML document (4, 1339). ---> Input string was not in a correct format.

Este error corresponde al protocolo de intercambio de datos y puede significar que:

- Un campo obligatorio no fue completado
- Se completó un campo con un tipo de datos inválido (float en lugar de int, coma en vez de punto, etc.)
- Se especificó un valor no esperado por el web service.

En cualquier caso, se puede revisar el mensaje XML de requerimiento (atributos xml_request o XmlRequest?) y buscar la etiqueta con datos incorrectos en la línea y caracter donde se informa el error (por ej., en este caso, línea 4, cerca del caracter 1339).

Error inesperado: no existe el archivo o directorio de certificado/clave privada

Este error corresponde a que la interfaz no encuentra los archivos del certificado/clave privada o la ruta no tiene algún directorio. Revisar la ruta completa y reintentar.

Mensaje de Error Original (certificado):

```
Unexpected Python Error: <type 'exceptions.IOError'>: [Errno 2] No such file or directory:
'reingart.crt'
```

Mensaje de Error Original (clave privada):

```
Unexpected Python Error: 936:error:02001002:system library:fopen:No such file or
directory:.\crypto\bio\bss_file.c:122:fopen('reingart.key','r') 936:error:2006D080:CRYPTO
routines:CRYPTO_new_file:no such file:.\crypto\bio\bss_file.c:125: : None
```

Error inesperado: La clave privada no coincide con el certificado

Este error corresponde a que el certificado no fue creado con la clave privada dada, por lo que no se puede realizar la encriptación necesaria. Revisar que la clave privada sea con la que se creó el pedido de certificado original (CSR) y reintentar. Si no se cuenta con la clave privada original, es imposible utilizar el certificado por lo que es necesario generar y asociar uno nuevo.

Mensaje de Error Original:

```
Unexpected Python Error: <class 'M2Crypto.SMIME.SMIME_Error'>:
936:error:0B080074:x509 certificate routines:X509_check_private_key:key values
mismatch:.\crypto\x509\x509_cmp.c:399: 936:error:2107407F:PKCS7
routines:PKCS7_sign:private key does not match certificate:.\crypto\pkcs7\pk7_smime.c:76:
```

Error Inesperado: La clave privada es inválida

Mensaje de Error Original:

```
Unexpected Python Error: 936:error:0906D06C:PEM routines:PEM_read_bio:no start
line:.\crypto\pem\pem_lib.c:647:Expecting: ANY PRIVATE KEY
```

Este error corresponde a que la clave privada no está en el formato correcto (texto plano: PEM), seguramente el archivo se encuentra en PKCS12 (binario: certificado x509 + clave privada con extensión .p12) lo que no es necesario para esta interfaz.

Falla SOAP: Server was unable to read request. ---> There is an error inXML document (4, 1339).33-> Input

Para convertir de pkcs12 a pem utilizar:

```
openssl pkcs12 -in empresa.p12 -out empresa.pem -nodes
```

Eso generará el archivo `empresa.pem` donde:

- La clave privada (`.key`) se encuentra entre: `-----BEGIN RSA PRIVATE KEY-----` y `-----END RSA PRIVATE KEY-----`
- El certificado (`.crt`) se encuentra entre: `-----BEGIN CERTIFICATE-----` y `-----END CERTIFICATE-----`

Copiar dichas líneas en archivos separados. Los archivos deben incluir el encabezado (`-----BEGIN . . .`) y pie (`-----END . . .`) para ser válidos.

En caso de ser necesario, para recorrer el camino inverso, generar archivo PKCS12 desde PEM (no necesario por esta interfaz), realizar:

```
openssl pkcs12 -export -inkey empresa.key -in empresa.crt -out empresa.p12
```

Error inesperado: Mala Descriptación

Este error corresponde a que el certificado fue creado usando una frase de seguridad (passphrase), por lo que en tiempo de ejecución solicita entrada dicha contraseña (bloqueándose si no se proporciona).

Mensaje de Error Original:

```
M2Crypto.EVP.EVPError: 3079395008:error:06065064:digital envelope routines:EVP_DecryptFinal_ex:bad decrypt:evp_enc.c:330:3079395008:error:0906A065:PEM routines:PEM_do_header:bad decrypt:pem_lib.c:428:
```

Para remover la passphrase (contraseña), por ej. del archivo `privada.pem` y grabarlo en `privada.key`, ejecutar el siguiente comando:

```
openssl rsa -in privada.pem -out privada.key
```

Error inesperado: Esperando certificado

El certificado proporcionado es inválido.

Es posible leer los datos de un certificado utilizando el comando `openssl`:

```
openssl x509 -in reingart.crt -noout -text
```

Debería mostrar la información completa del certificado:

Si no lo muestra, posiblemente el certificado esté en otro formato (ver como convertir desde pk7 en puntos anteriores)

Mensaje de Error Original:

M2Crypto.X509.X509Error: 3078817472:error:0906D06C:PEM routines:PEM_read_bio:no start line:pem_lib.c:650:Expecting: CERTIFICATE