

# LibPyAfipWs: Biblioteca DLL/.so para acceder a Servicios Web de la AFIP/ANMAT/etc

- **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
  - ◆ PyFEPDE: 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 Traslado (ARBA, API, AGIP, DGR)
  - ◆ Ingresos Brutos: Consulta de alícuotas WS DFE IIBB ARBA
  - ◆ Retenciones A-122R: Ingreso de comprobantes de retenciones
- **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.)

Biblioteca compartida multiplataforma para Emisión y almacenamiento electrónico de comprobantes originales AFIP - Argentina. AFIP RG 1956/05, 1361/02, 1345/02, 2265/07, 2289/07, 2485/08, 2557/09, 2668/09, 2758/10, 2853/10, 2904/10, 2959/10, 2974/10, 3066/11, 3067/11, 3210/11, 3419/12 (servicios web de Factura Electrónica, Trazabilidad y Liquidación de Granos, Consultas de Operaciones cambiarias y otros) Código de Operaciones de Traslado (Remito Electrónico). Ley Provincial 13.405. Normativas 34/2011 y 45/2011 ARBA. Programa Nacional de Trazabilidad de Medicamentos. Resolución 435/2011 del Ministerio de Salud y Disposición 3683/2011 de ANMAT.

Para otros productos (herramientas, aplicativos, generación de PDF, etc.) ver el menú lateral.

- PyAfipWs: **Interfaz multiplataforma** (objetos COM) para VB, VFP, etc. o por **TXT, DBF o XML**
- PyRece: **Aplicativo independiente** (simil SIAP/RECE) por planilla de cálculo
- HerramientaFacturaElectronica (FE.py): **Herramienta integrada por bases de datos**
- pyafipws2k.mdb, pyafipws97.mdb: Ejemplos en MS Access 2000 (o sup.) y MS Access 97

## Índice

## Introducción

LibPyAfipWs es una biblioteca de software libre a los Servicios Web de la AFIP, desarrollado en Python compatible con C, C++, C#, Visual Basic, Visual Fox Pro, Cobol, Delphi, .Net, Java, etc. y cualquier lenguaje/aplicación que pueda crear utilizar bibliotecas compartidas de enlace dinámico DLL en Windows o .SO en linux.

La biblioteca utiliza interfaces y librerías que han sido inspirados en los ejemplos de la AFIP y han sido probadas con éxito por varias empresas.

Actualmente implementa el Webservice de Autenticación (WSAA) y métodos de Factura Electrónica (emisión electrónica de comprobantes originales), proximately se incluirán más webservices y métodos.

## Funciones exportadas

Funciones genéricas (similares a C.O.M. para crear y acceder internamente a los objetos python):

- `void * PYAFIPWS_CreateObject(char *module, char *name):` instanciar un objeto para acceder a sus métodos y atributos

- `void PYAFIPWS_DestroyObject(void *object)`: destruir el objeto
- `char * PYAFIPWS_Get(void *object, char *name)`: obtener el valor de un atributo de un objeto
- `bool PYAFIPWS_Set(void * object, char * name, char * value)`: establecer el valor de un atributo de un objeto
- `void PYAFIPWS_Free(char *psz)`: liberar recursos de valores devueltos por la librería

#### Autenticación y Auotrización (WSAA):

- `char * WSAA_CreateTRA(char *service, long ttl)`: generar un ticket de requerimiento de acceso
- `char * WSAA_SignTRA(char *tra, char *cert, char *privatekey)`: firmar digitalmente el requerimiento
- `char * WSAA_LoginCMS(char *cms)`: solicitar un ticket de acceso

#### Factura electrónica (WSFEv1):

- `bool WSFEv1_Conectar(void *object, char *cache, char *wsdl, char *proxy)`: conectar al webservice
- `bool WSFEv1_Dummy(void *object)`: obtener estados de los servidores
- `bool WSFEv1_SetTicketAcceso(void *object, char *ta)`: establecer el ticket de acceso (token y sign)
- `long WSFEv1_CompUltimoAutorizado(void *object, char *tipo_cbte, char *punto_vta)`: obtener el último número de comprobante autorizado.

Proximamente se adicionarán más funciones. Para una descripción completa de los métodos y su funcionamiento, ver [ManualPyAfipWs](#).

**Importante:** las declaraciones de las funciones han sido simplificadas, ver [libpyafipws.h](#) para el encabezado completo

#### Notas para Windows:

- Las funciones están exportadas usando `__stdcall` (la convención de llamadas estandar que usa Pascal y VB por ej.).
- Los strings son [BSTR \(OleAutomation\)](#) para soportar temas de encodings y compatibilidad con lenguajes como VB, VFP, etc.. En C, C++, C#, también pueden ser tratados como `char *` (Ansi).

Recordar liberar la memoria alojada para los string que devuelve con `PYAFIPWS_Free`.

## Ejemplos

Ejemplo básico en C:

```
#include "libpyafipws.h"

int main(int argc, char *argv[]) {
    BSTR tra, cms, ta;
    void *wsfev1;
    BSTR ret;
    bool ok;
    long nro;
```

```

/* Generar ticket de requerimiento de acceso */
tra = WSAA_CreateTRA("wsfe", 999);
printf("TRA:\n%s\n", tra);
/* Firmar criptograficamente el mensaje */
cms = WSAA_SignTRA((char*) tra, "reingart.crt", "reingart.key");
printf("CMS:\n%s\n", cms);
/* Llamar al webservice y obtener el ticket de acceso */
ta = WSAA_LoginCMS((char*) cms);
printf("TA:\n%s\n", ta);

/* Crear una objeto WSFEv1 (interfaz webservice factura electronica) */
wsfev1 = PYAFIPWS_CreateObject("wsfev1", "WSFEv1");
printf("crear wsfev1: %p\n", wsfev1); /* si funcionó ok, no debe ser NULL! */

/* conectar al webservice (para produccion cambiar URL) */
ok = WSFEv1_Conectar(wsfev1, "", "", "");
printf("conectar: %s\n", ok ? "true" : "false");

/* obtener datos genericos de la interfaz (version y ruta de instalación) */
ret = PYAFIPWS_Get(wsfev1, "Version");
printf("wsfev1 Version: %s\n", ret);
free(ret);
ret = PYAFIPWS_Get(wsfev1, "InstallDir");
printf("wsfev1 InstallDir: %s\n", ret);
free(ret);

/* obtener el estado de los servidores (llama al ws) */
ok = WSFEv1_Dummy(wsfev1);
printf("llamar a dummy: %s\n", ok ? "true" : "false");
/* obtener los atributos devueltos por AFIP */
ret = PYAFIPWS_Get(wsfev1, "AppServerStatus");
printf("dummy AppServerStatus: %s\n", ret);
free(ret);
ret = PYAFIPWS_Get(wsfev1, "DbServerStatus");
printf("dummy DbServerStatus: %s\n", ret);
free(ret);
ret = PYAFIPWS_Get(wsfev1, "AuthServerStatus");
printf("dummy AuthServerStatus: %s\n", ret);
free(ret);

/* establezco los datos para operar el webservice */
ok = PYAFIPWS_Set(wsfev1, "Cuit", "20267565393");
ok = WSFEv1_SetTicketAcceso(wsfev1, (char*) ta); /* devuelto por WSAA_LoginCMS */

/* obtengo el ultimo numero de comprobante generado */
nro = WSFEv1_CompUltimoAutorizado(wsfev1, "1", "1");
printf("ultimo comprobante: %ld\n", nro);

/* destruir el objeto */
PYAFIPWS_DestroyObject(wsfev1);

/* liberar la memoria adquirida para los valores devueltos de WSAA */
PYAFIPWS_Free(ta);
PYAFIPWS_Free(cms);
PYAFIPWS_Free(tra);
}

```

Ver [DIIFacturaElectronica](#) para más ejemplos (incluyendo LoadLibrary, GetProcAddress, DECLARE en VFP y Declare en VB)

## Licencia

El código fuente puede ser descargado y utilizado sin cargo (gratis) respetando la licencia [GPLv3](#) de software libre: sin garantías, sin soporte técnico dedicado y/o obligatorio, informar copyright, no incorporarlo ni distribuirlo junto con software propietario, mantener derivados como software libre y contribuir modificaciones, etc.

Se ofrece instalación y soporte técnico comercial pago, incluyendo atención prioritaria y autorización especial para incorporar y distribuir esta interfaz a sistemas propietarios que no sean software libre.

A su vez, al ser software libre de código abierto, permite proteger su inversión, al no depender de un componente cerrado del cual no puede tener acceso al código fuente, revisar su funcionamiento, realizar futuras actualizaciones, etc.

Por consultas sobre el lenguaje python y demás, dirigirse a [PyAr](#). Para más información ver [FacturaElectronica](#).

Consultar por desarrollos especiales, interfaces web, etc.

## Características

- DLL directa (online) de simple uso: autenticación y obtención de CAE en pocas líneas!
- No usa archivos temporales ni formatos especiales
- No usa servidores intermedios (conexión directa con WS de AFIP)
- No requiere programas residentes o batch (por lotes)
- No es necesario tener conocimientos de encriptación ni protocolos web
- Autoinstalable 2.5MB (Todo en uno)
- Sin dependencias ni librerías o runtimes externas (Php, .Net o Java)
- Sin licencia de uso propietarias ni límites por cada usuario final
- Código abierto: archivos fuentes publicados, revisados y modificables (Software Libre)
- Sin problemas de instalación de OCX ni ActiveX (ver [comparativa](#))
- No requiere formularios visuales

## Contacto

Para mayor información, consultar por mail a [facturaelectronica@sistemasagiles.com.ar](mailto:facturaelectronica@sistemasagiles.com.ar) o telefónicamente al 15-3048-9211

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

[PyAfipWs](#) Copyright 2008, 2009, 2010, 2011, 2012, 2013 por [MarianoReingart](#) [PyAfipWs](#) Copyright 2008, 2009, 2010, 2011, 2012, 2013 por [MarianoReingart](#)