

py **Afí**  **p**
WS

Factura Electrónica Libre

Mariano Reingart

reingart@gmail.com

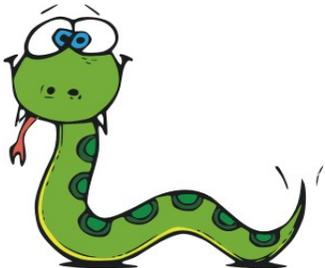
41° JAIIO 2012



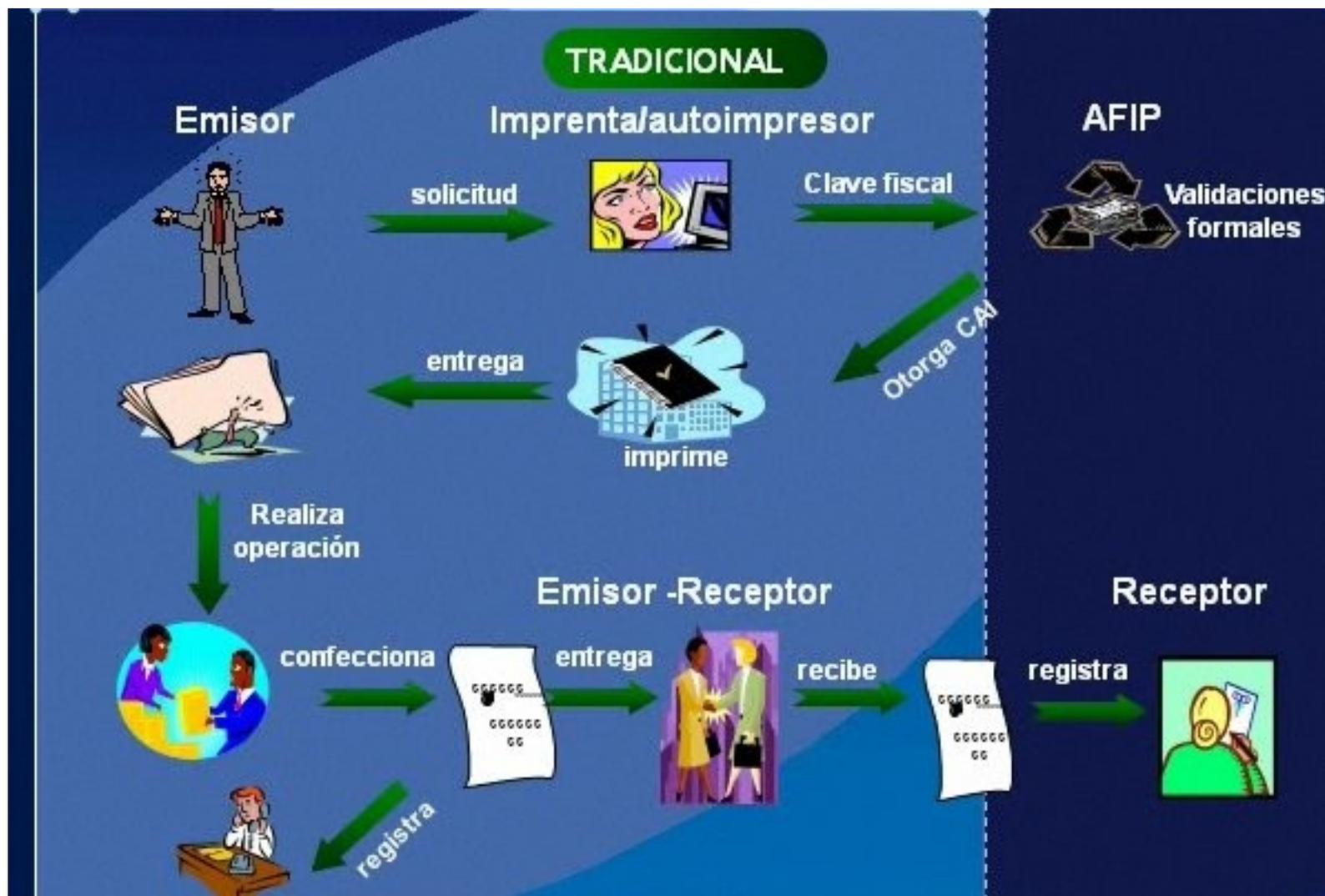
Factura Electrónica Libre

Temario:

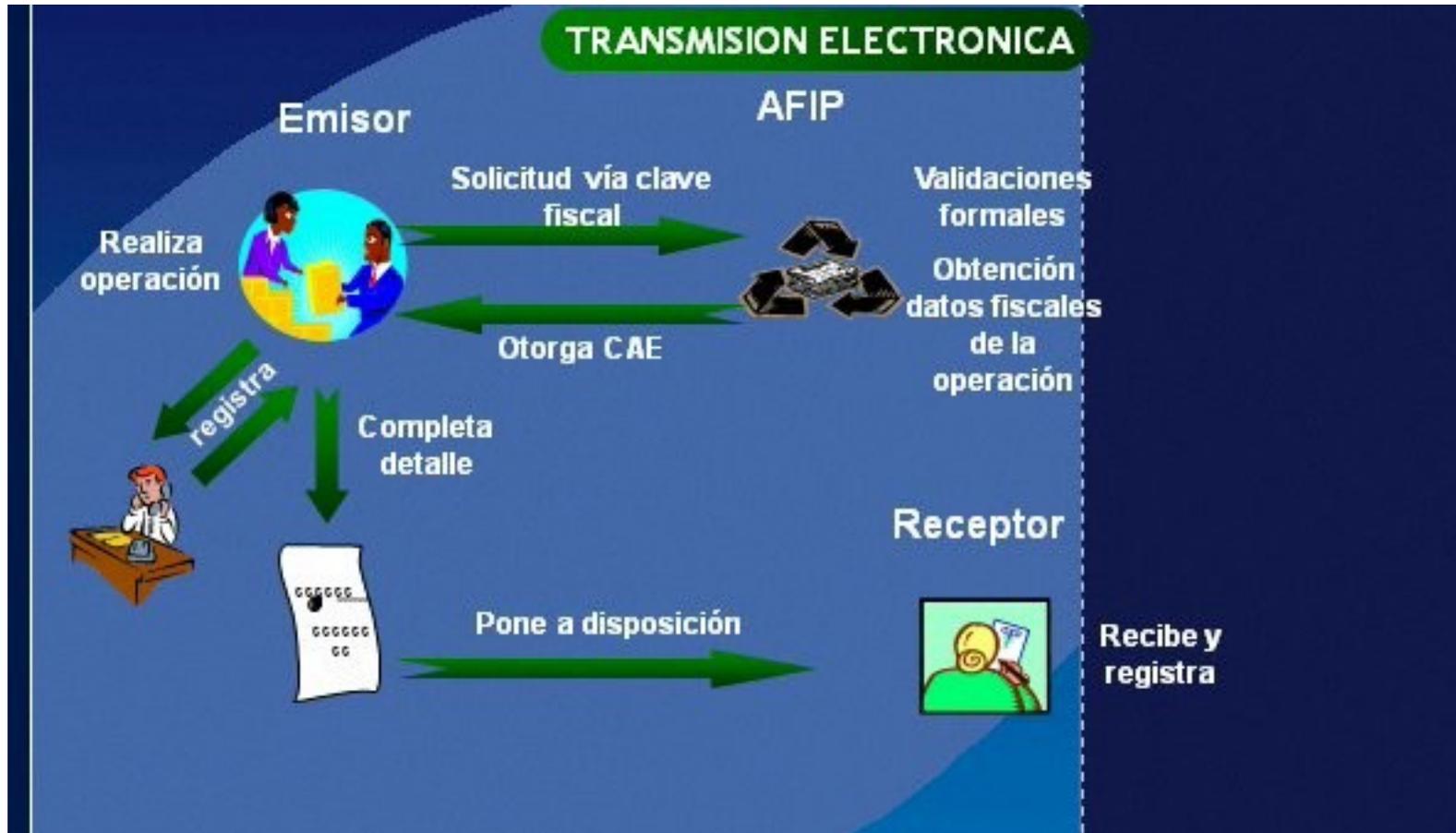
- **Introducción a Factura Electrónica: RECE y RCEL**
- **Webservices, Generación de PDF e Instaladores**
- **Comparativa Python vs Java vs PHP vs .NET**
- **Autenticación (WSAA) Gestión de CAE (WSFE)**
- **El proyecto y la comunidad: lecciones aprendidas**
- **El modelo de negocios: Soporte Comercial**
- **Aplicativo PyRece: prueba de concepto “SIAP LIBRE”**
- **El futuro: FacturaLibre → GestiónLibre: mini ERP**



Factura impresa tradicional



Factura electrónica



Regímenes para solicitud de autorización de facturas electrónicas:

Régimen de Emisión de Comprobantes Electrónicos (RECE)

- Programa aplicativo SIAP - Factura Electrónica (RECE)
- Web services (WSAA, WSFE, WSBFE, WSSEG, WSFEX, etc.)

Régimen de Emisión de Comprobantes Electrónicos en Línea (RCEL)

- Servicio con clave fiscal denominado "Comprobantes en línea"
- Servicio con clave fiscal denominado "Facturador Plus"

Facturador en línea por Clave Fiscal (y facturador plus):



AFIP
SITIO SEGURO

Usuario: GUARINO SEBASTIAN
Representando a: Asociados SA



Regímenes de Facturación y Registración
Salir

Asociados SA

CUIT: 330000000006
Dependencia: AGENCIA NRO 63

RCEL - GENERACIÓN DE COMPROBANTES - FACTURA A

DATOS DE LA OPERACIÓN (PASO 3 DE 4)

●●●●○

Código	Producto/Servicio	Cant.	U. Medida	Prec. Unitario	% Bon.	Importe Bon.	Subtotal	Alícuota IVA	Importe IVA	Subtotal	Eliminar
0005	Software de Gestión	1	unidades	5000	10	500.00	4500.00	21%	945.00	5445.00	X
0009	Mantenimiento	1	unidades	500	0		500.00	21%	105.00	605.00	X

Detalle	Base Imponible	Alícuota %	Importe
Per./Ret. de Imp. a las Ganancias:	<input type="text"/>	<input type="text"/>	\$ <input type="text"/>
Per./Ret. de IVA:	<input type="text"/>	<input type="text"/>	\$ <input type="text"/>
Per./Ret. Ingresos Brutos:	<input type="text"/>	<input type="text"/>	\$ <input type="text"/>
Impuestos Internos:	<input type="text"/>	<input type="text"/>	\$ <input type="text"/>
Impuestos Municipales:	<input type="text"/>	<input type="text"/>	\$ <input type="text"/>
Importe Otros Tributos:			\$ <input type="text"/>

Importe Neto Gravado:	\$	<input type="text" value="0.00"/>
Importe Exento:	\$	<input type="text" value="0.00"/>
Importe Neto Gravado:	\$	<input type="text" value="5000.00"/>
IVA 27%:	\$	<input type="text" value="0.00"/>
IVA 21%:	\$	<input type="text" value="1050.00"/>
IVA 10,5%:	\$	<input type="text" value="0.00"/>
Importe Otros Tributos:	\$	<input type="text"/>
Importe Total:	\$	<input type="text" value="6050.00"/>



Servicios Web:

- Pensado para integrar Sistemas de facturación existentes:
 - WSAA (Webservice de Autorización y Autenticación)
 - WSFE (Webservices de Factura Electrónica)
 - WSFEX (" Factura electrónica exportación)
 - WSBFE (" Bonos Fiscales Electrónicos)
 - WSSEG (" Pólizas de Seguros de Caución)
- Autorización (CAE) on-line en el momento

Aplicativo SIAP RECE:

- Programa independiente bajo Windows
- Alta de facturas individual o importación archivo de texto
- Resultados por Clave Fiscal (Ventanilla Electrónica)
- Automatización vía archivos de texto y emails



Implicancias

- Optimización de recursos y procesos*
- Reducción de costos*
- Problemas de compatibilidad:
 - SIAP solo Windows
 - WS realizado con Java (vs) y .NET
- Adaptación de sistemas actuales o adquirir nuevos
- Bibliotecas, componentes OCX, etc. propietarias
- Software como Servicio

* Nota: Dependiendo de la solución y tecnología elegida



Motivación PyAfipWS

Objetivos:

- Software Libre
- Dependencias mínimas
- Mantenimiento simple (código KISS)
- Multiplataforma (Windows/Linux)
- Embebible en otros lenguajes (VB, VFP, etc.)
- Posibilidad de Interfase Gráfica (ventanas)
- Posibilidad de Interfase Web
- Instalación simple todo-en-uno (EXE/DLL)



¿Por que Python?

Fundamentos:

- Software Libre. Código Maduro. Comunidad establecida. Aprox. 20 años de desarrollo sostenido. [\[ohloh\]](#)
- Baterias Incluidas (ej: xml, email) y extensiones (pdf)
- Código compacto, simple y claro: desarrollo rápido de fácil mantenimiento
- Multiplataforma (Win, *nix, Mac) y Embebible: COM (Win32), Jython (Java), IronPython (.NET)
- Interfases Gráficas: WX, GTK, QT, Win32, etc.
- Interfases Web: web2py / django
- Instalación todo-en-uno: Py2Exe, Py2App, etc.



Tabla Comparativa

<u>Entorno:</u>	<u>JAVA</u>	<u>.NET</u>	<u>PHP</u>	<u>Python</u>
WSAA (LOC aprox.)	300	500***	100**	100
WSFE (LOC aprox.)	N/A	300***	140**	150
SDK (Windows)	73 MB	350 MB	17 MB	10 MB
Runtime	15 MB	22 MB	N/A	N/A
Instalador todo-en-uno	N/A	N/A	posible	2.5 MB
Fácil de Embeber	limitado*	limitado*	no*	sí
Multiplataforma	sí	no	sí	sí
Software Libre	sí	no	sí	sí

* Comparado con Python (COM, cantidad código, tamaño instalador, runtime, etc.)

** No tiene prácticamente líneas en blanco, comentarios, documentación ni licencia

*** Incluye breve ayuda de uso para el usuario dentro del código. Cantidades para C#

Temas Resueltos y bibliotecas desarrolladas:

PySimpleSOAP:

- Resolver incompatibilidades (.NET; Java: Axis 1/2, Jboss)
- Cliente mínimo adaptable, servidor *simulaciones* (web2py)

SimpleXMLElement:

- Simplificar el uso de XML
- Facilitar la serialización / tipos de datos

PyFPDF:

- PDF dinámicos y sencillos (celdas fluidas)
- Plantillas en PDF

Puro python, sin código generado ni configuración



WebServices: PySimpleSOAP

SOAP (Simple Object Access Protocol) permite realizar llamadas a procedimientos remotos utilizando.

Se desarrolló una librería basada en idas tomadas de extensiones de PHP:

- XML: Manipulación de datos (Simple XML Element)
- HTTP/HTTPS: Transmisión de datos (urllib, httplib2, libcurl) y soporte para servidores proxy (ej. MS ISA) y validación de certificados SSL (cliente y servidor)
- WSDL: Definición (análisis dinámico vs artefactos), compatibles con distintos “dialectos” y espacios de nombres



Generación de PDF: PyFPDF

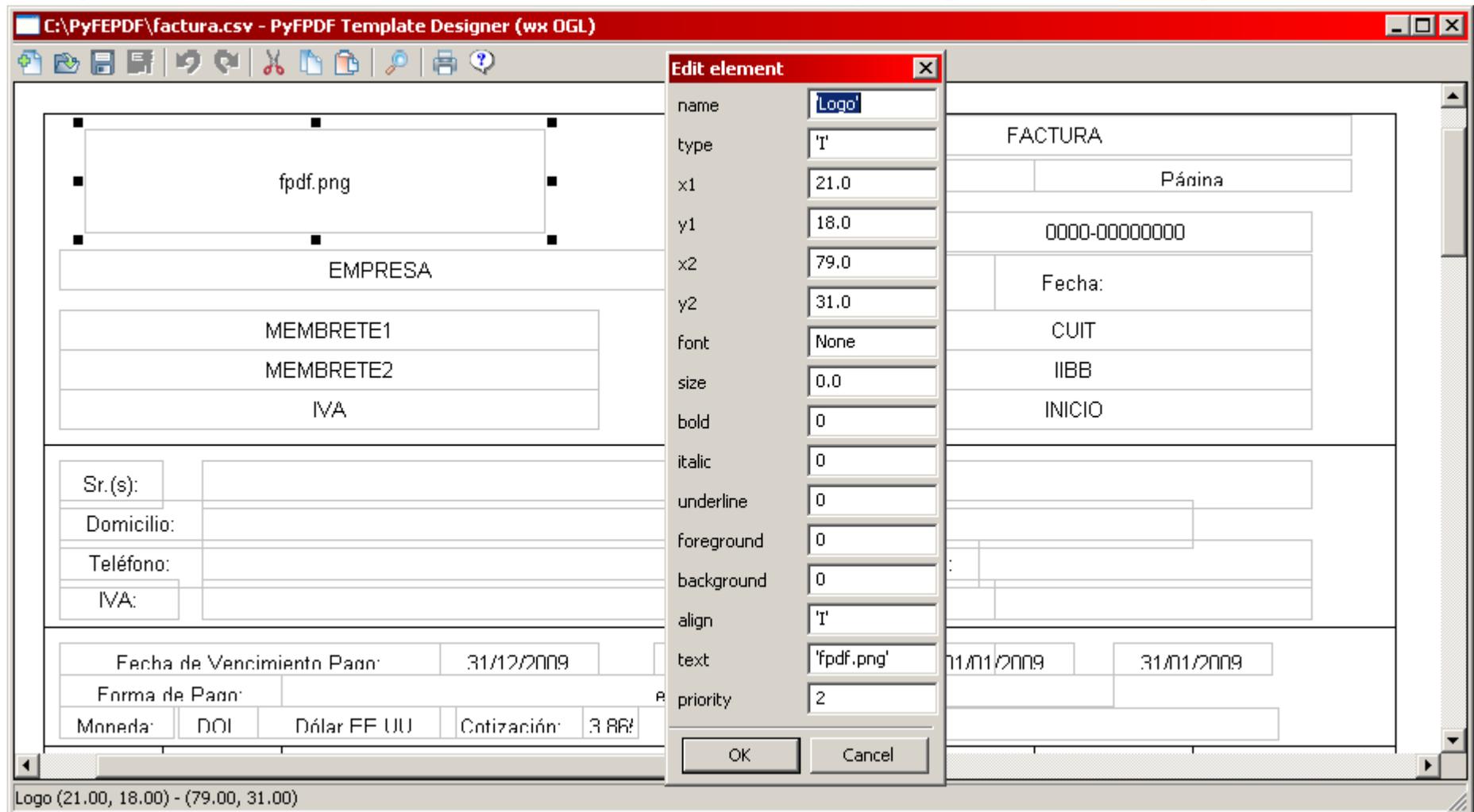
FPDF permite generar documentos portables.

Se desarrolló una librería basada en ideas tomadas de FPDF de PHP:

- Simple y de fácil uso, con primitivas avanzadas
- Celdas fluidas (ideal para plantillas tipo “Factura”)
- Extensible (por ej. códigos de barra) y Flexible (hooks)
- Ejemplos y documentación extensiva (varios lenguajes)

A diferencia de ReportLab es más compacta sin layout engine

Se desarrolló un diseñador para ser utilizado por el usuario final (no requiere conocimientos de programación):





Generación de Instaladores

Se extendió el sistema de generación de instaladores de python para:

- Automatizar “compilación en ejecutables”
- Empaquetar en archivos autoextraíbles
- Soportar Nullsoft Scriptable Install System (NSIS)

Ejemplo en 1 línea:

```
c:\python25\python.exe setup_wsfev1.py py2exe
```

La función del WSAA es devolver el "Ticket de Acceso" compuesto por un token y un sign, que se utilizarán para poder operar con el resto de los servicios web de negocio.

- Generar un **"Ticket de Requerimiento de Acceso"** (TRA) en formato XML, con los datos del origen, destino, tiempos de generación y expiración, identificador único, servicio a acceder.
- Firmar el TRA con nuestro certificado y clave privada, generando una **"Solicitud de Acceso Firmada"** (SAF) en formato CMS
- Llamar remotamente al WSAA con el SAF para obtener el **"Ticket de Acceso"** (TA)



Ejemplo Python WSAA

Ejemplo de autenticación (WSAA):

```
from pyafip.ws import wsaa

# crear ticket acceso, firmarlo y llamar al ws
tra = wsaa.create_tra()
cms = wsaa.sign_tra(tra, "homo.crt", "homo.key")
ta_xml = call_wsaa(cms, wsaa.WSAAURL)

# procesar el xml
ta = wsaa.SimpleXMLElement(ta_xml)
token = str(ta.credentials.token)
sign = str(ta.credentials.sign)
```

Ejemplo de autorización (WSFE):

```
from pyafip.ws import wsfe

# crear cliente SOAP
client = wsfe.SoapClient(wsfe.WSFEURL,
                        action = wsfe.SOAP_ACTION,
                        namespace = wsfe.SOAP_NS)

# autorizar factura electronica (obtener CAE)
res = wsfe.aut(client, token, sign,
              CUIT =20267565393, tipo_cbte=1, punto_vta=1,
              cbt_desde=1, cbt_hasta=1,
              tipo_doc=80, nro_doc=23111111113,
              imp_total=121, imp_netto=100, impto_liq=21)

print res['cae'], res['motivo']
```

Ejemplo en VB WSFE (aplicaciones legadas)

Ejemplo de uso interfase COM y Visual Basic 5:

```
Set WSAA = CreateObject("WSAA")  
tra = WSAA.CreateTRA()  
cms = WSAA.SignTRA(tra, "homo.crt", "homo.key")  
ta = WSAA.CallWSAA(cms, url)
```

```
Set WSFE = CreateObject("WSFE")  
WSFE.Token = WSAA.Token ' setear token y sign de wsa  
WSFE.Sign = WSAA.Sign  
WSFE.Cuit = "30000000000" ' CUIT del emisor
```

```
ok = WSFE.Conectar(url)
```

```
cae = WSFE.Aut(id, presta_serv, tipo_doc, ...  
             imp_tot_conc, imp_netto, impto_liq, ...)
```

Proyectos de ejemplo:

- [ej_vb.zip](#): Visual Basic 5/6
- [ej_vfp.zip](#): Visual Fox Pro 5 o posterior
- [ej_php.zip](#): PHP 5 o posterior
- [ej_delphi.zip](#): Turbo Delphi 2006

Otros lenguajes/entornos compatibles:

-
- Power Builder, Genexus, etc.
- Cobol (WOW), ABAP (SAP)
- Excel (VBA), Access (VBA)
- Java (vía COM o vía Jython)
- .Net (vía COM o IronPhyton)



COT: Remito Electrónico

Servicio Web de ARBA para Código de Operación de Traslado (COT) "Remito Electrónico" según Artículo 34 bis del Código Fiscal (T.O. 2004) incorporado por la Ley 13.405 (Provincia de Buenos Aires).

Funcionalidades:

- **Presentar Remitos:**
 - **Archivo de texto (Formulario)**
 - **Respuesta XML**



Trazabilidad de Medicamentos:

Interfaz para Servicio Web Código de Trazabilidad de Medicamentos (SOAP) correspondiente a la Resolución 435/2011 del Ministerio de Salud y Disposición 3683/2011 de A.N.M.A.T.:

Funcionalidades:

- **SendMedicamentos**
- **SendCancelacTransac**

PyRece: Aplicativo Ad-hoc

- Aplicación multiplataforma
- Interfaces por línea de comando y gráfica (visual)
- Alternativa mejorada a SIAP / RECE
 - Proceso automatizado online (en el momento)
 - Generar Facturas PDF y enviarlas por Email
 - Interfaz con planilla de cálculo simple (CSV) y otros formatos (XML, DBF, json, etc.)
- No requerir un sistema de facturación
- Fácil de programar y mantener (500 líneas aprox.)
- Prueba de Concepto para SIAP Libre

DEMO de Factura Electrónica (RECE)

Archivo:

Facturas:

Tipo Cbte	Punto Vta	Cbt Num...	Fecha C...	Tipo Doc	Nro Doc	Imp N...	Impto Liq	Imp
1	3	1	20090129	80	23111111...	100.00	21.00	121
1	3	2	20090129	80	23111111...	100.00	21.00	121
1	3	3	20090129	80	23111111...	100.00	21.00	121
1	3	4	20090129	80	23111111...	100.00	21.00	121
1	3	5	20090129	80	23111111...	100.00	21.00	121

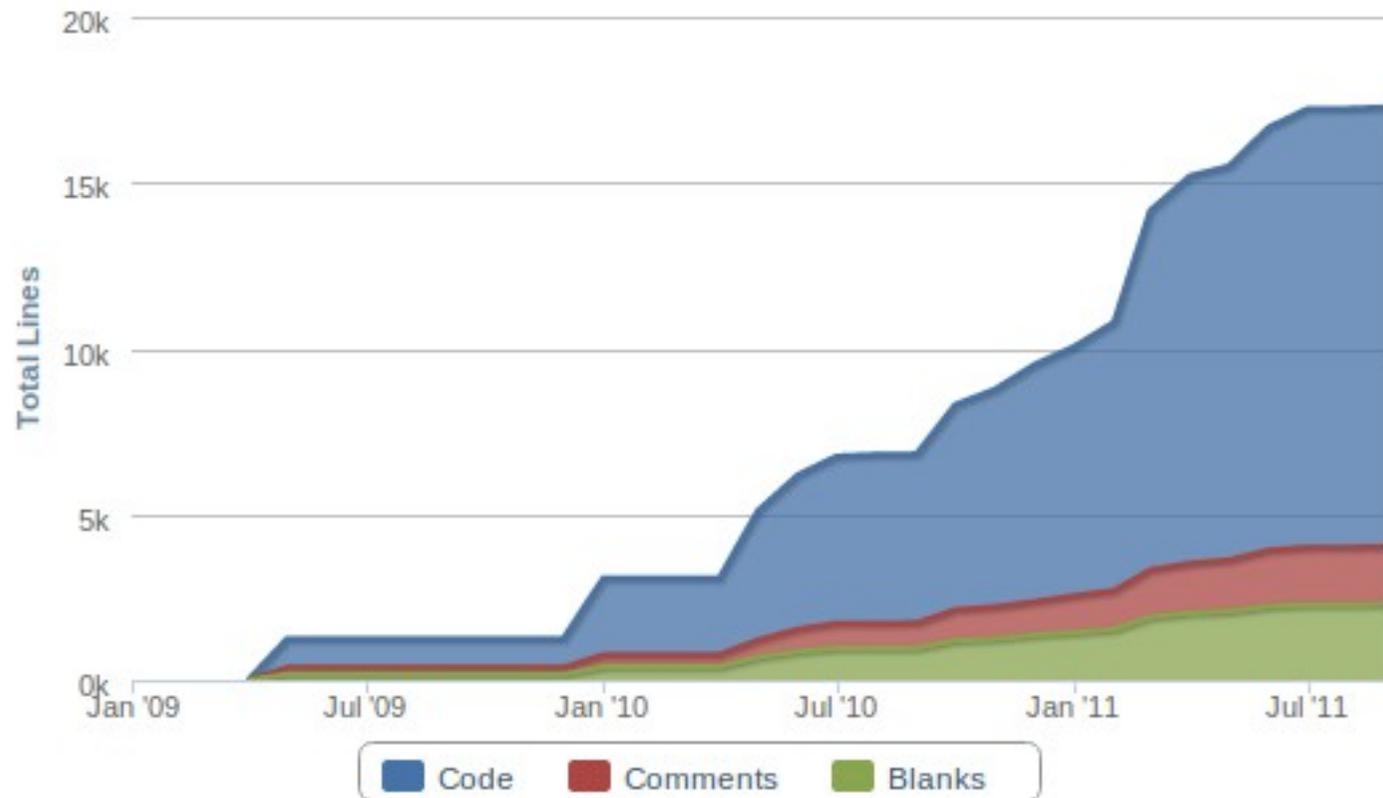
Progreso:

Estado:

Enviando email: Factura Electronica 3 a mariano@nsis.com.ar
 Enviando email: Factura Electronica 2 a mariano@nsis.com.ar
 Enviando email: Factura Electronica 1 a mariano@nsis.com.ar
 ID: 2000400000022 CAE: NULL Motivo: 11 Reproceso: S
 ID: 2000400000021 CAE: NULL Motivo: 11 Reproceso: S
 ID: 2000400000018 CAE: 59053558622276 Motivo: 00 Reproceso: S
 ID: 2000400000017 CAE: 59053644931062 Motivo: 00 Reproceso: S
 ID: 2000400000016 CAE: 59053639299624 Motivo: 00 Reproceso: S
 ID: 2000400000015 CAE: 59053233908335 Motivo: 00 Reproceso: S
 ID: 2000400000014 CAE: 59053205909320 Motivo: 00 Reproceso: S
 ID: 2000400000013 CAE: 59053875684369 Motivo: 00 Reproceso: S
 ID: 2000400000012 CAE: 59053965793493 Motivo: 00 Reproceso: S

Evolución del código

[Análisis https://www.ohloh.net/p/pyafipws/:](https://www.ohloh.net/p/pyafipws/)





El Proyecto

Estadísticas <https://www.ohloh.net/p/pyafipws/>:

- o +3 años, 13K SLOC, **Costo Estimado: US\$ 165.823**
- o 635 cambios a 137 archivos (solo en 2 años googlecode)
- o ~10 colaboradores (inicialmente 2)
- o >100 usuarios en el grupo de noticias

Más de 100 clientes:

The Coca-Cola Company, La Hispano Argentina Curtiembre SA, Boehringer Ingelheim S.A., Alliance One International, TUPPERWARE Brands Argentina, Grupo Solunet S.A. (ISP), Diario El Litoral SRL, Scienza Argentina, E-Payment SRL (DineroMail), Canal 4 Carlos Pelegrini S.R.L ...

Temas a tener en cuenta:

- Indiferencia de las comunidades relacionadas (python)
- Falta de masa crítica (colaboradores)
- Desconocimiento de Google Code (wiki, issue), Repositorio (hg) y grupos de discusión
- Violaciones a la GPL3 (mejoras de 3ros. sin fuentes)

Temas positivos:

- El proyecto nació de la lista de PyAr
- Lista de correo (grupo de usuarios) autonomo
- Ejemplos contribuidos y algunos parches
- La mayor participación fue del exterior (librerías SOAP)

Soporte Comunitario Gratuito:

- Instalador de homologación para **evaluación**
- Código Fuente publicado bajo GPL3 (sin garantía)
- Documentación y Video publicados bajo FDL

Soporte Comercial Pago:

- Acceso a instaladores para producción
- Atención prioritaria, Correcciones y Mejoras. Niveles (hs):
 - **Mínimo:** consultas puntuales
 - **Actualización:** nuevos ws
 - **Básico:** consultas y ajustes
 - **Avanzado:** grandes empresas

Factura Libre (facturador):

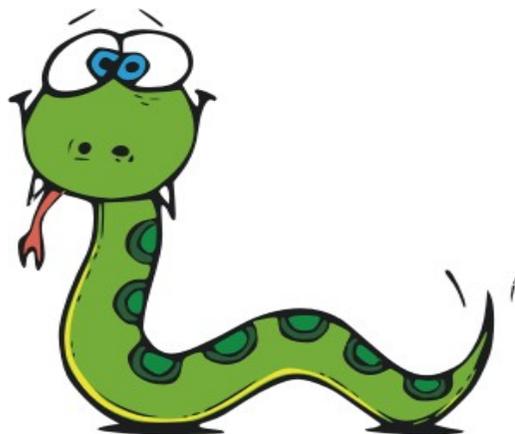
- Aplicación web2py mejorando a RCEL (clientes, productos, sesión más persistente, diseño del pdf)
- Analisis del desarrollo necesario del framework
- Estado: terminado

Gestión Libre (mini ERP/SCM/CRM):

- Sistema de gestión Administrativo/Contable
- Basado en contabilidad por partida doble
- Simple, fácil de instalar, usar y mantener
- Interfaz dual: Visual (wxPython) y Web (web2py)
- Estado: iniciado

Direcciones Útiles:

- Soporte comercial: [Sistemas Ágiles](#)
- Sitio Web: www.pyafipws.com.ar info@pyafipws.com.ar
- Noticias: <http://groups.google.com.ar/group/pyafipws>
- Código Fuente: <http://code.google.com/p/pyafipws/>
- AFIP: www.afip.gov.ar/fe



py Afip WS

GRACIAS!

2009-2012 Copyright

Marcelo Alaniz, Mariano Reingart et. al.

Esta presentación se publica bajo licencia GNU FDLv3.

