

---

# **Pysafet Documentation**

***Release 1.0***

**jose sulbaran**

August 01, 2014



## CONTENTS

<b>1</b>	<b>Pyafet</b>	<b>3</b>
1.1	Sistema Automatizado de Firma Electrónica y Estampado Electrónico de Tiempo - SAFET . . . . .	3
1.2	SAFET - inflow . . . . .	3
1.3	Algunas Características . . . . .	3
<b>2</b>	<b>pyafet para Debian Wheezy 7.0 (64 bits)</b>	<b>5</b>
2.1	Instalacion Safet . . . . .	5
2.2	Funciones de pysafet . . . . .	6
<b>3</b>	<b>pysafet para Ubuntu</b>	<b>7</b>
3.1	Instalacion de las pysafet . . . . .	7
3.2	Pasos para la compilación de la librería Libsafet . . . . .	7
3.3	Verificar si la libreria Libsafet se instalo correctamente . . . . .	8
3.4	Funciones de pysafet . . . . .	8
<b>4</b>	<b>Sphinx</b>	<b>9</b>
4.1	Instalacion Sphinx . . . . .	9
4.2	tabla . . . . .	11



**Contenido:**



**PYSAFET**

biblioteca de Python para usar “Safet”, un flujo de trabajo y la biblioteca electrónica de la firma.

## **1.1 Sistema Automatizado de Firma Electrónica y Estampado Electrónico de Tiempo - SAFET**

Es una herramienta que permite desarrollar nuevas aplicaciones de software con flujos de trabajo, es decir, automatización de forma expedita de procesos de una organización, agregando firma Electrónica y estampillado de tiempo.

La firma electrónica tiene un soporte legal en la Ley sobre Mensajes de Datos y Firma Electrónica (2001) de la República Bolivariana de Venezuela. Esta ley brinda soporte legal a la firma electrónica como la firma autógrafa y especifica la creación de la Superintendencia de Servicios de Certificación Electrónica (SUSCERTE como organismo que gestiona la Infraestructura Nacional de Certificación Electrónica o PKI Nacional. El estampillado de tiempo es un servicio que permite adjuntar a documentos electrónicos la hora legal venezolana.

El sistema automatizado para la Firma Electrónica y Estampado de Tiempo (SAFET) surge como una herramienta que permite desarrollar nuevas aplicaciones de software que incluyan las funcionalidad de flujos de trabajo, firma Electrónica y estampillado de tiempo.

## **1.2 SAFET - inflow**

- Tener un software avanzado para gestión de proyectos basada en `trac`.
- Disponibilidad de Firma Electrónica mediante Tarjetas Inteligentes o archivos de claves.
- Crear un sistema completo de información de forma muy rápida.
- Modelar procesos de una organización y utilizarlos basado en un sencillo lenguaje de flujo de trabajos.
- Sistema de gestión de usuarios y roles.
- Ver información en tiempo real.
- Elaborar reportes, informes y gráficos avanzados.
- Realizar visualización inteligente.

## **1.3 Algunas Características**

1. Eficiente registro de documentacion.

2. Totalmente integrado con sistema de Firma electrónica y Estampillado de Tiempo.
3. Un lenguaje para hecer pequeñas aplicaciones basado en flujo de trabajo **Automatización muy rápida de procesos.**
4. Visualización a través de reportes **HTML** y **PDF**.
5. Visualización Gráfica de flujos de trabajo (**funciones: acercar, alejar,mover,etc**).
6. Sistema de usuario y auctorización.
7. Complementacion configurable y adactable.

## **1.4 Ver un archivo de safet**

### **Archivo de Safet**

- `libsafet-dev.install`
- `Parsegetopt.cpp`
- `main.cpp`



## PYSAFET PARA DEBIAN WHEEZY 7.0 (64 BITS)

biblioteca de Python para usar “Safet”, un flujo de trabajo y la biblioteca electrónica de la firma.

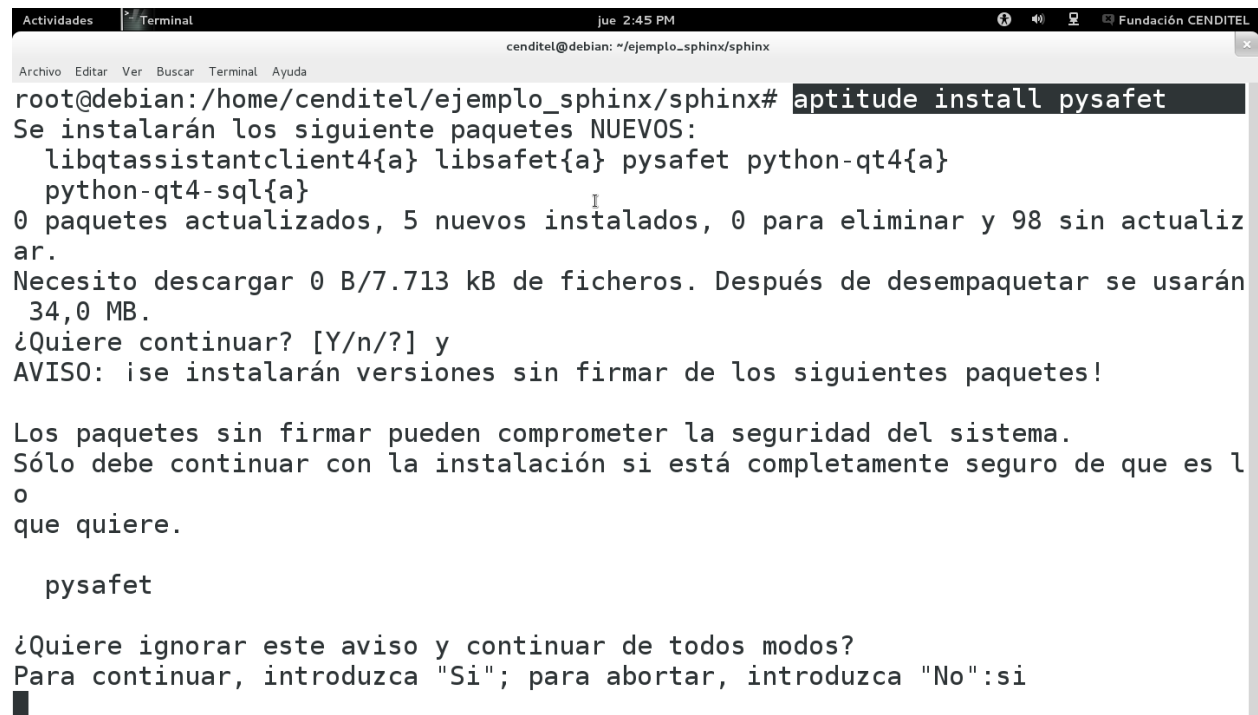
### 2.1 Instalacion Safet

Se Agrega el repositorio al archivo `sources.list` con la siguiente línea .. `productionlist`:

```
# cd /etc/apt/  
# gedit sources.list  
deb http://tibisay.cenditel.gob.ve/repositorio wheezy main
```

Luego ejecutamos .. `productionlist`:

```
# aptitude update  
# aptitude install pysafet
```



The screenshot shows a terminal window titled "Terminal" with the following content:

```
Actividades Terminal jue 2:45 PM Fundación CENDITEL  
cenditel@debian: ~/ejemplo_sphinx/sphinx  
Archivo Editar Ver Buscar Terminal Ayuda  
root@debian:/home/cenditel/ejemplo_sphinx/sphinx# aptitude install pysafet  
Se instalarán los siguiente paquetes NUEVOS:  
  libqtassistantclient4{a} libsafet{a} pysafet python-qt4{a}  
  python-qt4-sql{a}  
0 paquetes actualizados, 5 nuevos instalados, 0 para eliminar y 98 sin actualiz  
ar.  
Necesito descargar 0 B/7.713 kB de ficheros. Después de desempaquetar se usarán  
34,0 MB.  
¿Quiere continuar? [Y/n/?] y  
AVISO: ise instalarán versiones sin firmar de los siguientes paquetes!  
  
Los paquetes sin firmar pueden comprometer la seguridad del sistema.  
Sólo debe continuar con la instalación si está completamente seguro de que es l  
o  
que quiere.  
  
  pysafet  
  
¿Quiere ignorar este aviso y continuar de todos modos?  
Para continuar, introduzca "Si"; para abortar, introduzca "No":si  
■
```

## 2.2 Funciones de pysafet

**Importación** Dentro de la consola de python importamos safet para su uso

```
>>> import Safet
>>> dir(Safet)
['MainWindow', 'ParsedSqlToData', 'SafetDocument', 'SafetVariable', 'SafetWorkflow',
'SafetXmlObject', 'SafetYAWL', '__doc__', '__file__', '__name__', '__package__']
```

**Otra forma de mostrar las funciones**

```
>>> for i in range(len(dir(Safet))):
...     dir(Safet)[i]
...
'MainWindow'
'ParsedSqlToData'
'SafetDocument'
'SafetVariable'
'SafetWorkflow'
'SafetXmlObject'
'SafetYAWL'
'__doc__'
'__file__'
'__name__'
'__package__'
```

## PYSAFET PARA UBUNTU

biblioteca de Python para usar “Safet”, un flujo de trabajo y la biblioteca electrónica de la firma.

### 3.1 Instalacion de las pysafet

El código fuente de la librería Libsafet se encuentra alojada en una plataforma de desarrollo colaborativo de software (forja) llamada Github, esta plataforma utiliza como sistema de control de versión el software Git.

**Para la compilación se requiere que se instale los siguientes dependencias:**

```
libgraphviz-dev
libtar-dev
g++
libglib2.0-dev
x11proto-xext-dev
libqt4-dev
libssl-dev
make
python-qt4-sql
python-sip-dev
python-qt4-dev
libqt4-sql-sqlite
```

**Para instalar las dependencias desde la línea de comandos o terminal (puede hacerlo con synaptics):**

```
# aptitude install libgraphviz-dev libtar-dev g++ libglib2.0-dev x11proto-xext-dev
libqt4-dev libssl-dev make python-qt4-sql python-sip-dev python-qt4-
```

**Para clonar el repositorio de Libsafet debemos instalar el control de versiones git:**

```
$ sudo aptitude install git
```

**Luego ubicado en el directorio de trabajo (donde se va a descargar los fuentes) ejecuta el comando:**

```
$ git clone https://github.com/victorrbravo/pysafet.git pysafet
```

Dentro del directorio de trabajo se crea el directorio pysafet donde se tiene el clone de la librería Libsafet

### 3.2 Pasos para la compilación de la librería Libsafet

**Compilación de la librería Libsafet:**

```
$ cd pysafet/websafet
$ make clean
$ qmake-qt4
$ make
$ cd ../PySafet/
$ python configure.py
$ make
$ sudo make install
$ cd ../websafet
$ sudo make install
```

### 3.3 Verificar si la librería Libsafet se instaló correctamente

Se procede a arrancar un intérprete de Python, para lo mismo debe hacer los siguientes pasos:

```
$ python
cenditel@CENDITEL:~/proyecto/pysafet/websafet$ python
Python 2.7.3 (default, Jan  2 2013, 13:56:14)
[GCC 4.7.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Dentro del prompt de python (>>>), se importa la librería de Safet ejecutando el siguiente comando:

```
>>> import Safet
>>>
```

### 3.4 Funciones de pysafet

```
>>> import Safet
>>> dir(Safet)
['MainWindow', 'ParsedSqlToData', 'SafetDocument', 'SafetVariable', 'SafetWorkflow',
'SafetXmlObject', 'SafetYAWL', '__doc__', '__file__', '__name__', '__package__']
```

**SPHINX**

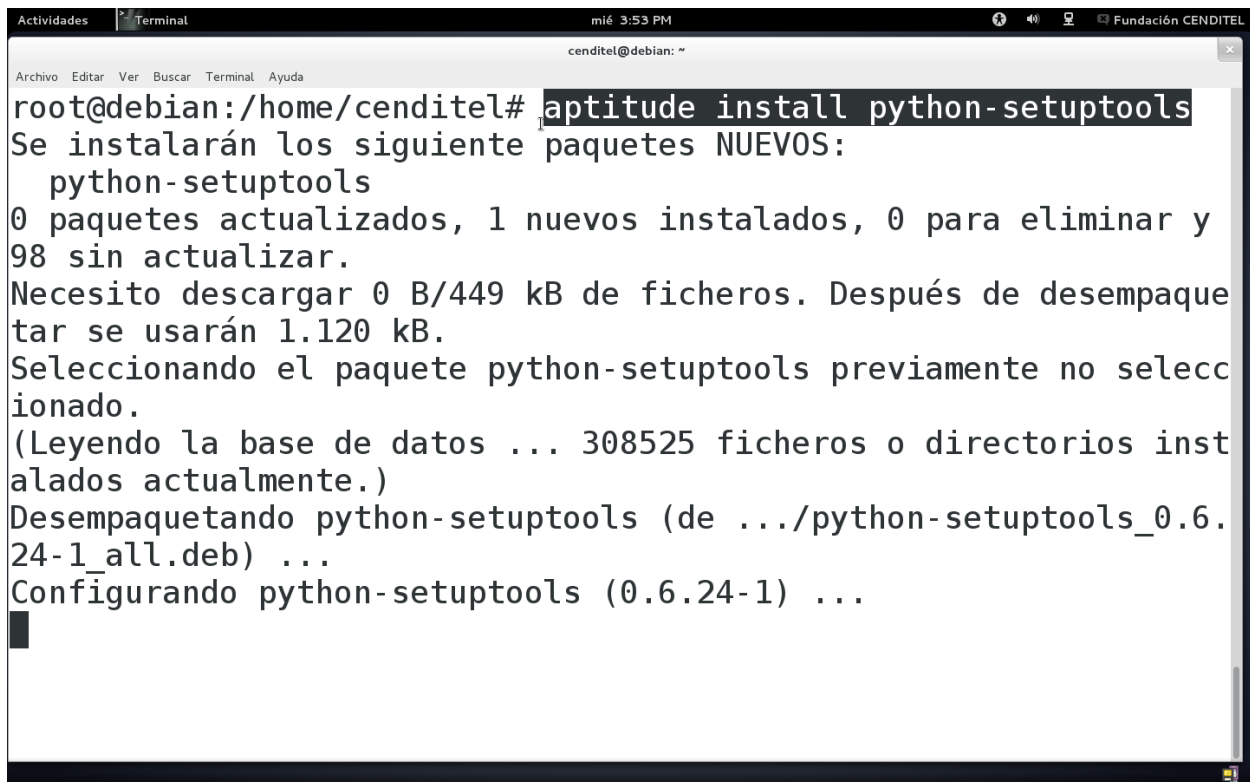
Sphinx es una herramienta que hace que sea fácil crear la documentación inteligente y hermosa, escrita por Georg Brandl y licenciado bajo la licencia BSD.

Fue creado originalmente para la nueva documentación de Python, y tiene excelentes instalaciones para la documentación de los proyectos de Python, pero C/C++ ya está apoyado también, y está previsto añadir soporte especial para otros idiomas. Por supuesto, este sitio también se crea a partir de fuentes reStructuredText utilizando Esfinge! Las siguientes funciones deben destacarse.

## 4.1 Instalacion Sphinx

### Primer paso:

```
#aptitude install python-setuptools
```

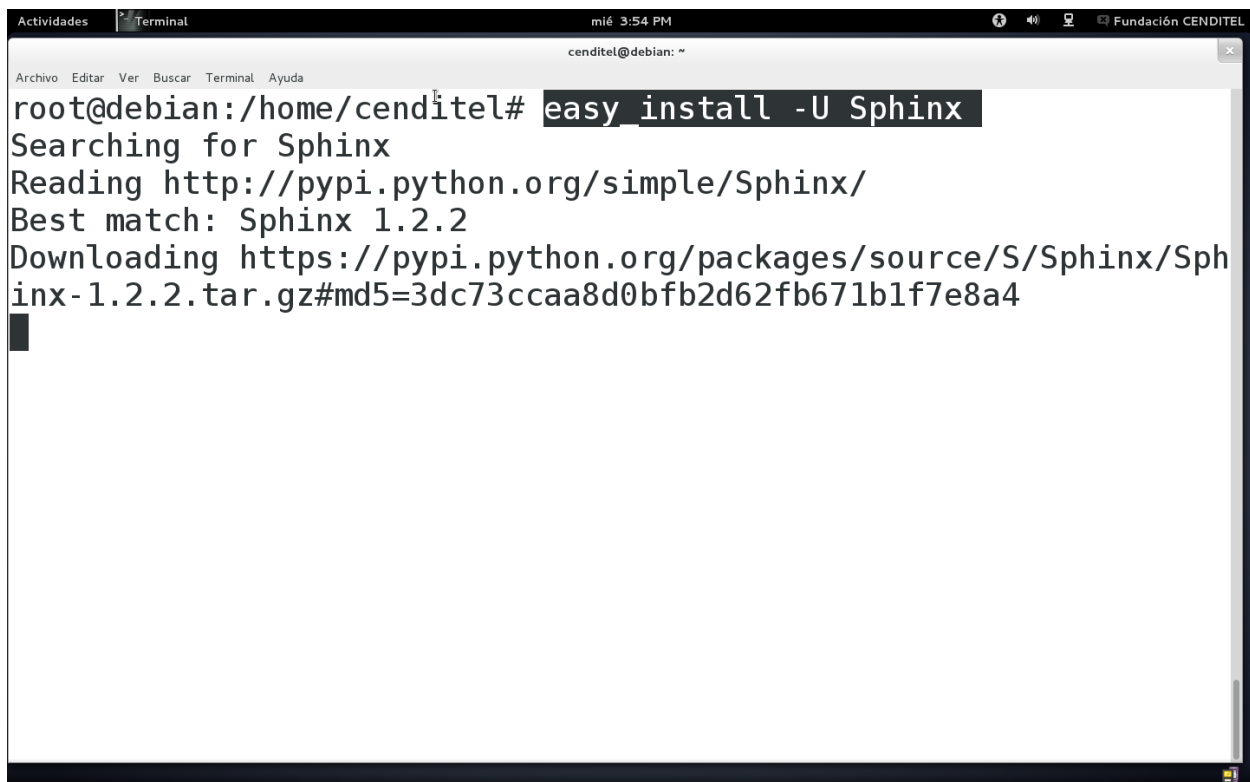


A terminal window titled 'Terminal' with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda) and a status bar (mié 3:53 PM, Fundación CENDITEL). The prompt is 'root@debian:/home/cenditel#'. The command 'aptitude install python-setuptools' is entered. The output shows the installation of python-setuptools, including package details, disk space requirements, and progress.

```
root@debian:/home/cenditel# aptitude install python-setuptools
Se instalarán los siguiente paquetes NUEVOS:
  python-setuptools
0 paquetes actualizados, 1 nuevos instalados, 0 para eliminar y
98 sin actualizar.
Necesito descargar 0 B/449 kB de ficheros. Después de desempaque
tar se usarán 1.120 kB.
Seleccionando el paquete python-setuptools previamente no selecc
ionado.
(Leyendo la base de datos ... 308525 ficheros o directorios inst
alados actualmente.)
Desempaquetando python-setuptools (de ../python-setuptools_0.6.
24-1_all.deb) ...
Configurando python-setuptools (0.6.24-1) ...
```

Segundo paso:

```
#easy_install -U Sphinx
```



A terminal window titled 'Terminal' with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda) and a status bar (mié 3:54 PM, Fundación CENDITEL). The prompt is 'root@debian:/home/cenditel#'. The command 'easy\_install -U Sphinx' is entered. The output shows the search for Sphinx, the best match (Sphinx 1.2.2), and the download link.

```
root@debian:/home/cenditel# easy_install -U Sphinx
Searching for Sphinx
Reading http://pypi.python.org/simple/Sphinx/
Best match: Sphinx 1.2.2
Downloading https://pypi.python.org/packages/source/S/Sphinx/Sph
inx-1.2.2.tar.gz#md5=3dc73ccaa8d0bfb2d62fb671b1f7e8a4
```

LICENCIA