

Requerimientos de diseño de la aplicación pysms.

Objetivo:

Desarrollo de una aplicación de envío de SMS de forma masiva que facilite a las personas su uso con un celular Android.

El usuario podrá manejar una agenda de números de celular y definir incluso la hora de envío de sms con reportes y estadísticas al finalizar cada actividad de envío de SMS.

Requerimientos no funcionales:

- Lenguaje de programación Python.
- Interfaz gráfica gtk o Qt.
- Base de datos sqlite3 con soporte de cifrado de datos.
- Manejo de colas con zeroMQ.
- Uso de thread para evitar el congelamiento de la interfaz gráfica.
- Utilizar el sdk de Android que viene en el repositorio de paquetes de Debian.
- Crear paquete debian de la aplicación con dependencias necesarias para facilitar su instalación y uso.
- La licencia de la Aplicación será la GPLv3.
- La licencia de la documentación será la creative common.
- Se creará un paquete fuente de instalación para Python.
- Se creará un paquete debian para wheezy y canaima.

Requerimientos funcionales:

- Envío de sms por medio de celulares android.
- Utilizar al librería SL4A
- Manejo de agenda de teléfonos (inserción, eliminación y edición de grupos de números de teléfonos celulares).
- Manejo de estadísticas y reportes de envío de SMS.
- Tener la aplicación en modo cliente (interfaz gráfica, consola) y servidor (demonio).

Aplicación servidora.

Se creará un demonio que atenderá las solicitudes de envío de sms, se le pasa la lista de números de teléfonos, el mensaje a enviar y la hora y fecha de envío. El servidor recibe la información y cataloga los envíos, estos se procesarán al cumplirse el momento de envío por medio de colas para buscar enviar el mayor números de sms en un instante dado.

Por el lado del servidor manejará la configuración de la conexión con el celular por medio de parámetros que maneje en archivos en el directorio /etc o en el caso del usuario en el directorio .config de cada usuario.

Aplicación cliente (Interfaz gráfica).

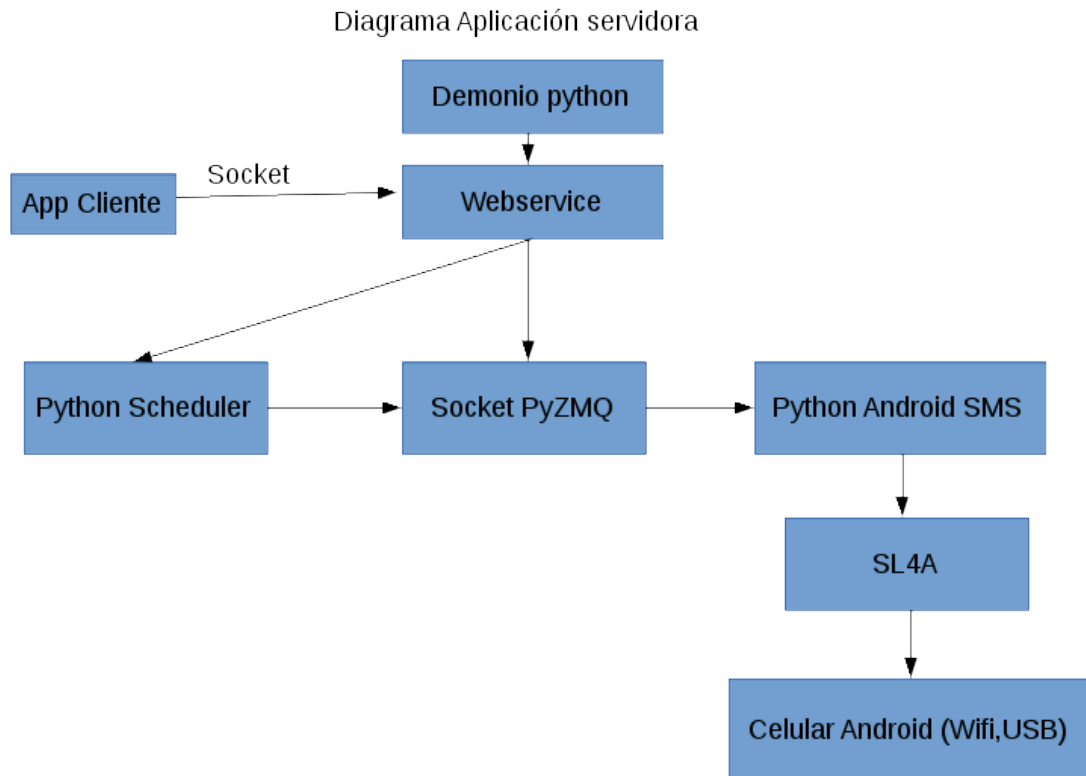
La aplicación gráfica facilitará la edición, inserción y eliminación de grupos y sus números de celular, podrá enviar sms de manera individual o de manera grupal.

Aplicación cliente (consola).

La aplicación tendrá las mismas funciones de la interfaz gráfica pero desde la línea de comandos o con interfaz de dialog.

Diagrama de la aplicación servidora

En la siguiente figura se muestra un diagrama de la funcionalidad de la aplicación servidora.



En el computador se inicia un servicio como demonio que estará escuchando un webservice de manera local, este webservice aceptará las peticiones de la aplicación cliente, la cual puede manejar un envío individual de SMS o grupos de SMS, además recibir si se requiere enviar los SMS en un momento determinado.

Si el sms es individual o de un grupo de sms se pasa la información al socket PyZMQ (manejo de colas) que procesa el envío al bloque Python Android SMS que procesa la cola y empieza a enviar los SMS por medio de SL4A al celular Android (conectado por wifi o por USB).

Si el envío de SMS individual o grupal se hace como un evento posterior, el webservice procesa la actividad enviando los datos a Python Scheduler quien es el que se encarga de revisar el momento que le toca enviar los SMS, cuando este momento ocurre, se le pasa toda la información (números de celular y mensaje) a la cola de PyZMQ quien se encarga de nuevo de pasarle los datos a la cola de Python Android SMS, de ahí a SL4A y al final al Celular.

Demonio

Es una aplicación en python junto con un script bash en /etc/init.d/ que levanta la aplicación

webservice o la detiene según las necesidades del usuario. Esto permite que cada vez que se inicie el equipo el webservice se levante y se apague de forma limpia.

WebService

Se utilizará la librería para webservice SOAP (python-soaplib o python-soappy) . Esta librería levantará un webservice local (127.0.0.1 en el puerto 8050) de manera segura (sólo escuchará localmente).

Como ya se mencionó este webservice está a la espera de peticiones del cliente (interfaz gráfica, consola o en un futuro interfaz web); de ahí se le pasará los datos a python-scheduler o a la cola PyZMQ. El webservice envía los datos de manera serializada (python-cjson).

Python Scheduler

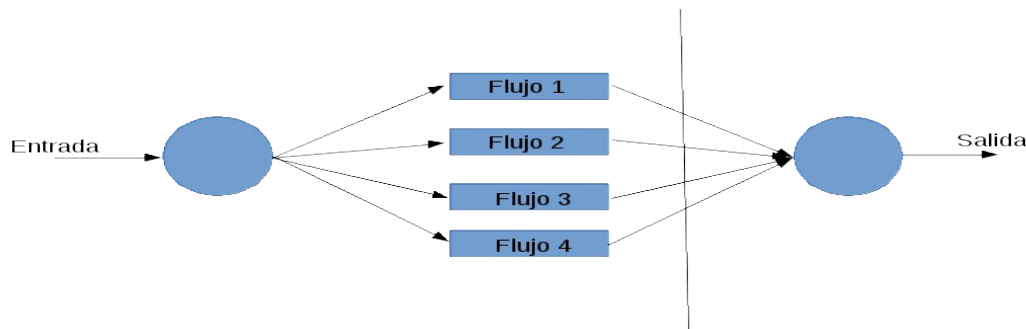
El scheduler es una librería de python que permite programar tareas de forma periódica o en un momento determinado, este módulo procesará los datos (números de celular, envío individual o grupal y el mensaje) colocandolo en una tarea o grupos de tareas que esperan el momento para pasar los datos a la cola PyZMQ.

Cola PyZMQ

La librería PyZMQ es una librería que facilita la creación de colas de mensajes, puede manejar patrones de mensajes de colas, el protocolo que se desea utilizar. La Cola PyZMQ recibirá los datos del scheduler o del webservice enviandolo luego al final de la cola en Python Android SMS.

Entre PyZMQ y Python Android SMS se maneja un alto flujo de datos más de los que pudiera procesar Python Android SMS se usará el modelo de cola.

Este puede manejar multiples flujos de datos entre la entrada y la salida (figura).



Python Android SMS

Este módulo maneja internamente la configuración que se necesita para comunicarse con SL4A y al celular (wifi o usb) y recibe los datos de la Cola PyZMQ, este le pasa los datos a SL4A para enviar

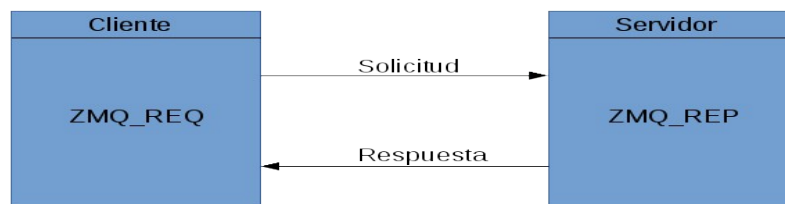
los SMS y procesar los posibles errores.

Los errores se guardarán en dos tipos de bitacora, una por medio de logs y por medio de una base de datos sqlite con el ORM sqlalchemy.

Es necesario incorporar un archivo desarrollado por la gente de SL4A que se llama android.py, este es el que facilita el uso de SL4A desde python.

Entre Python Android SMS y SL4A se usará la arquitectura Request-Reply (Solicitud-respuesta), es la arquitectura más sencilla que maneja ZeroMQ la cual permite que recibir el estatus del envío de SMS por medio de SL4A, de esa forma se conoce si el mensaje fue enviado o no (figura Arquitectura request-reply).

Arquitectura request-reply



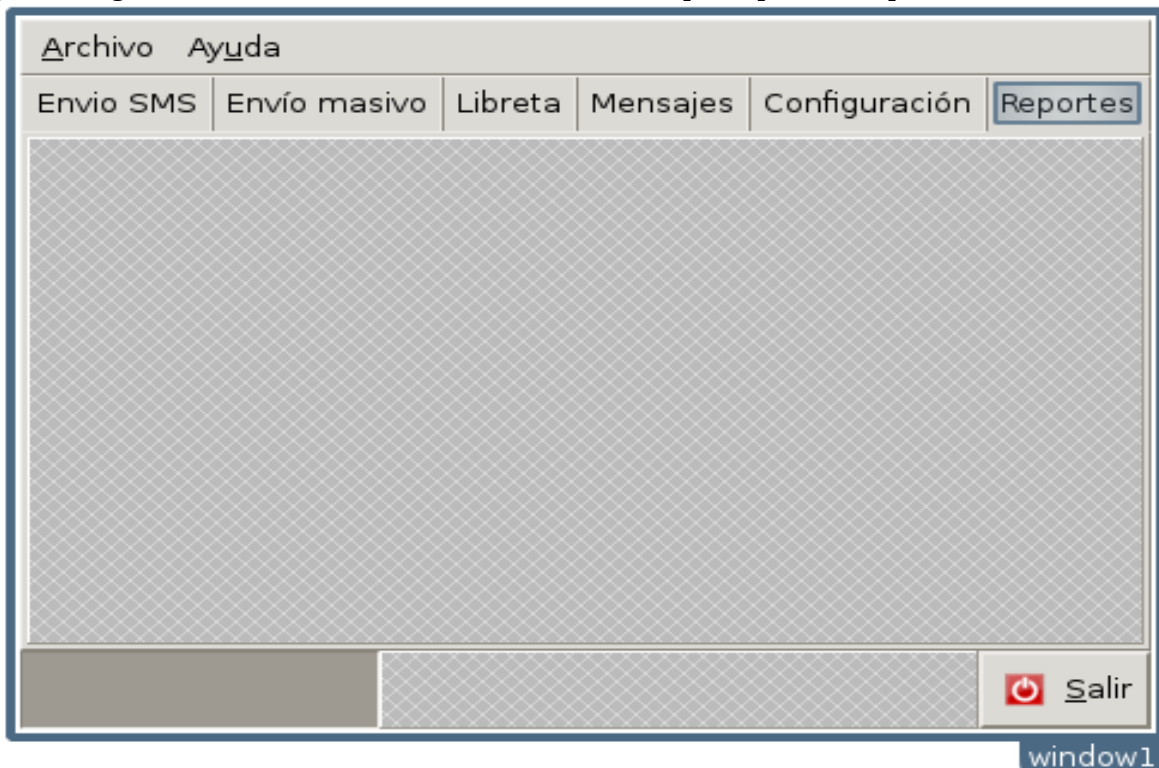
Aplicación Cliente

La aplicación cliente enviará al servidor las solicitudes y/o configuraciones que necesite enviar, la aplicación cliente sería el front-end del servidor. Para este caso se hará sólo una versión gráfica; pero se podría desarrollar una versión para la línea de comandos, una estilo cli o de dialogos de texto para la consola o una versión estilo web.

La aplicación cliente tendrá varias pestañas:

- Envío SMS: Permite enviar un mensaje a la vez.
- Envío masivo: Permite enviar varios mensajes por medio de la libreta de números de celular que se encuentren agrupados por grupos.
- Libreta: Permite crear, editar y eliminar grupos de números de celular.
- Mensajes: Permite guardar, editar o eliminar mensajes que podrán ser utilizados en el envío de un mensaje individual o grupal.
- Configuración: Permite configurar el nombre de la base de datos sqlite3 a utilizar, el puerto y host que se usará en la conexión con SL4A para enviar los SMS por medio de wifi o USB.
- Reportes: Permite definir y generar los reportes de los envíos realizados.

La siguiente figura muestra como se visualizará la ventana principal de la aplicación.



Envío SMS

Permite enviar un mensaje de texto a la vez, donde se puede seleccionar el número o seleccionarlo de una lista de grupos de números de celular. También se puede escribir el mensaje o seleccionar de una lista de mensajes, por último se tiene el envío inmediato o programado del mensaje.

Archivo Ayuda

Envío SMS Envío masivo Libreta Mensajes Configuración Reportes

Mensaje

☒ Mensaje ☐ Lista de Mensajes

Número Celular

☒ Celular ☐ Grupo Grupo Celular

Enviar

☒ Enviar ahora ☐ Programar Envío Fecha: Enviar Programar Salir

window1

Esta pestaña tendrá 3 marcos:

- Mensaje: Puede seleccionar escribir un mensaje nuevo o tomar un mensaje de la lista de mensajes.
- Número Celular: Puede seleccionar escribir un número de celular o tomar un número de celular de la lista de grupos de números de celular de la libreta de números.
- Enviar: Puede seleccionar enviar el mensaje de una vez o programar el mensaje para una fecha u hora en específico.

Envío masivo de SMS

Esta pestaña permitirá el envío masivo de mensajes de texto, donde se le puede pasar el mensaje o seleccionar un mensaje de la lista, se puede seleccionar un grupo de números de celular y por último programar el envío o enviar inmediatamente el mensaje de texto.

[Archivo](#) [Ayuda](#)

Envío SMS	Envío masivo	Libreta	Mensajes	Configuración	Reportes
-----------	---------------------	---------	----------	---------------	----------

Mensaje

☒ Mensaje

☐ Lista de Mensajes

Grupo envío Celular

Grupo:

Enviar

☒ Enviar ahora

☐ Programar envío Fecha:

Esta pestaña tendrá 3 marcos:

- Mensaje: Donde se escribe o selecciona el mensaje a enviar.
- Grupo envío celular: Donde se lista los distintos grupos de números de celular a quienes se les enviará el mensaje de texto.
- Enviar: Donde se selecciona enviar ahora o programar envío a una fecha particular.

Libreta de Teléfonos celular:

Esta pestaña permite gestionar grupos de números de celular; crear, editar y eliminar grupos; así como en la parte de editar y crear se puede agregar números celulares, editarlos o eliminarlos.

Esta pestaña tendrá 3 marcos:

- Agregar Grupo: Este contiene la entrada de datos para agregar el nombre de un grupo, se define si se requiere cargar los números celulares uno a uno o por medio de un archivo csv.
- Editar o eliminar Grupo: Aquí se permite seleccionar un grupo y se puede o editar o eliminar el grupo. En el caso de editar abrirá una ventana adicional donde se podrá o editar el nombre del grupo o, editar, agregar o eliminar números de celular. En el caso de eliminar se abre una ventana de confirmación de que quiere eliminar el grupo y se perderán los números de celular.
- Exportar a archivo csv: Se permite exportar a un archivo csv un grupo que se seleccione.

Mensajes:

Esta pestaña permite agregar mensajes, editarlos o eliminarlos.

Archivo Ayuda

Envio SMS Envío masivo Libreta **Mensajes** Configuración Reportes

Agregar

Mensaje:

Editar/Eliminar

Mensaje:

window1

Se tiene 2 marcos en la pestaña:

- Agregar: Permite agregar un mensaje a la lista de mensajes.
- Editar o Eliminar: Permite seleccionar la lista de mensajes, al darle al botón editar abrirá una ventana secundaria facilitando la edición del mensaje y si se le da clip al botón Eliminar se abre una ventana de confirmación de eliminación de mensaje.

Configuración:

Esta pestaña permite configurar si se usará el celular conectado por USB o por medio de wifi, además se define el responsable del envío de SMS, a esta persona es a quien le llegará un correo y/o un correo con el reporte de finalización de envío de SMS.

Archivo Ayuda

Envío SMS Envío masivo Libreta Mensajes **Configuración** Reportes

Android

Tipo de Conexión: ☐ Wifi ☒ USB

Host(IP):

Puerto:

Responsable

Nombre:

Celular:

Correo:

Guardar

Salir


window1

Esta pestaña tiene 3 marcos:

- Android: Permite definir el tipo de conexión al celular (USB o wifi), así como su IP o puerto para conectarse.
- Responsable: Define el responsable a quien se le enviará un correo y/o mensaje de texto con el resultado del envío de mensajes de texto.
- Guardar: Luego de realizar los cambios se le puede dar click al botón guardar, el cual guardará los cambios en la configuración.

Reportes:

La pestaña permite definir el formato del archivo a generar (pdf, csv, html o txt), luego se puede seleccionar si se requiere generar reporte de todos los envíos de sms o de un grupo en específico.

Archivo Ayuda					
Envío SMS	Envío masivo	Libreta	Mensajes	Configuración	Reportes
Formato Reporte					
Tipo de reporte:					
Grupos o todos					
<input checked="" type="radio"/> Grupo					
<input type="radio"/> Todos					
Generar reporte					
		Generar			
				 Salir	

Esta pestaña maneja 3 marcos:

- **Formato del Reporte:** Permite definir el tipo de formato al cual se generará el reporte (pdf, csv, html, ods).
- **Grupo o todos:** Permite definir a cual grupo se le generará el reporte o a todos los grupos.
- **Generar reporte:** Este contiene el botón para generar el reporte que se requiere.

Base de datos

La aplicación manejará como base de datos por defecto la base de datos sqlite3. Se utilizará como ORM a sqlalchemy.

Tablas:

- Mensajes(id,mensaje)
- Grupos(id,nombre,descripcion)
- Numeros(id,numero,nombre,grupoid)
- Responsable(id,nombre,celular,correo)
- Android(id,tipoconex,host,puerto,activo)
- Bitacora(id,mensajeid,numeroid,grupoid,fecha,enviado)

Mensajes:

Esta tabla maneja dos campos, un indentificador como clave primaria y el mensaje como texto.

Grupos:

Esta tabla maneja tres campos, un identificador del grupo como clave primaria, el nombre del grupo y la descripción como texto.

Números:

Esta tabla maneja 4 campos, un identificador del número como clave primaria, el número celular como texto de 11 caracteres, el nombre de la persona dueña del número celular y el identificador del grupo al que pertenece el número celular.

Responsable:

Esta tabla manejará 4 campos, un identificador del responsable como clave primaria, el nombre del responsable como texto, su celular como texto de 11 caracteres y el correo como texto.

Android:

Está tabla contendrá 5 campos, un identificador como clave primaria, el tipo de conexión como texto (usb,wifi), un host que contiene la IP por donde se conectará al celular, el puerto de la ip y si está activo o no (booleano).

Bitacora:

La tabla bitacora llevará un registro de los envíos de mensajes de texto, está contiene 6 campos, un identificador como clave primaria, mensajeid clave foranea de la tabla mensajes, numeroid como clave foranea de la tabla números, grupoid como clave foranea de la tabla grupos, la fecha de tipo fecha y hora y si fue enviado el mensaje de tipo booleano.

La tabla bitacora es muy probable que esté del lado servidor, quien es el que manejará los datos de envío de los sms, la tabla Android también se manejará del lado del servidor ya que es la que maneja la información que se necesita para Python Android SMS y SL4A.

Formato de envío de los datos de la aplicación cliente a la aplicación servidor.

Los datos que manejará la aplicación son los siguientes:

- Telefonos
- Mensajes
- hash md5 números celulares
- hash md5 mensaje
- Cantidad de teléfonos
- Inmediato (True/False)
- Fecha Envío

Los datos se manejarán como una estructura tipo diccionario:

```
Datos = {  
    telefonos: ['04125271515',...],  
    cantidad_celulares: 1,  
    mensaje: 'Mensaje',  
    hashmd5_numeros:'bcaa23',  
    hashmd5_mensaje:'01212c1b1dffe312',  
    inmediato:'True|False',  
    fecha_envio:'formatofecha',  
}
```

Manejar los datos entre la aplicación cliente y servidora como un diccionario facilita principalmente en manejarlo como un formato json. Entre los campos del diccionario se tienen 2 hash md5, uno para los números celulares y otro para el mensaje a enviar. De esta forma el webservice puede validar que la información que recibe es correcta.

Si los hash no son los mismos que se envían el servidor descarta todo el dato escribiendo en la tabla bitacora de la base de datos que los mensajes no se enviaron.

Referencias:

Python-Daemon:

<http://blog.crespo.org.ve/2014/02/crear-un-demonio-de-linux-con-python.html>

PyZMQ:

<http://blog.crespo.org.ve/search/label/ZeroMQ>

Python GTK:

<http://blog.crespo.org.ve/search/label/gtk>

Webservice SOAP:

<http://blog.crespo.org.ve/search/label/SOAP>

SQLite3 y Ssqlalchemy:

<http://blog.crespo.org.ve/search/label/Ssqlalchemy>

Python para Android:

<http://blog.crespo.org.ve/search/label/Android>