

Modulo: “Python en Sistemas Embebidos”



Documento de Especificación de Requerimientos SRS.

Integrantes:

Correa, Carlos Maximiliano. Dni: 33.702.654

Galfioni, Dario Adrian. Dni: 37.177.650

1. Introducción

En el presente documento se explicarán y analizarán los requisitos que necesitará para su desarrollo el trabajo final evaluativo en el marco del módulo “Python en sistemas embebidos” , desarrollado por Correa, Carlos Maximiliano ; Galfioni, Adrián Darío.

1.1 Propósito

Este documento tiene como propósito dar a conocer el funcionamiento general del proyecto que está dirigido al equipo evaluador.

Docentes :

- Mg. Sebastián TOSCO
- Mg. Fernando CORTEGGIANO

1.2 Ámbito del Sistema.

- Nombre del Sistema: EEIA “ Envío de Email iterativo en Android”
- El sistema generará y enviará cada 1 minuto los estados de algunos sensores de un dispositivo móvil (tablet o smartphone).
- El Email debe tener la siguiente estructura:
 - Debe ser enviado desde la siguiente dirección:
pruebasdisse@gmail.com (pass: pruebas123).
 - Debe tener un asunto, y estar formado por la palabra “Report” y seguido la estampa de tiempo (fecha y hora).
 - El contenido debe ser una lista de estados de sensores del dispositivo (al menos 2 datos).
 - Además, debe incluirse como adjunto una imagen con una foto del dispositivo (debe ser sacada antes y luego adjuntada).

1.3 Referencias

- Python for Beginners:
 - <https://www.python.org/about/gettingstarted/>
- Documentación oficial Python:
 - <https://docs.python.org/2/library/email-examples.html>
- Stackoverflow, Attachment Image to send by email using Python:
 - <http://stackoverflow.com/questions/13070038/attachment-image-to-send-by-mail-using-python>
- Python Diario:
 - <http://www.pythondiario.com/2014/05/obtener-fecha-y-hora-actual-en-python.html>
- AndroidHelper
 - <https://kylelk.github.io/html-examples/androidhelper.html>

2. Análisis de Requerimientos

2.1 Perspectiva del Producto

El producto final permite emanar de manera iterativa cada un tiempo determinado información de los sensores del dispositivo en el que esté corriendo, vía email, en sistemas android de dispositivos móviles.

2.2 Requerimientos funcionales.

- Captar el estado de los sensores contenidos en un dispositivo móvil Android.
- Listar el estado de los sensores con formato texto.
- Levantar una archivo de imagen (*.jpeg), alojada en un directorio propio del sistema Android.
- Obtener la hora y el tiempo del momento de envío.
- Enviar email con la siguiente estructura:
 - Dirección de envío: pruebasdisse@gmail.com (pass: pruebas123).
 - Debe tener un asunto, y estar formado por la palabra "Report" y seguido la estampa de tiempo (fecha y hora).
 - El contenido debe ser una lista de estados de sensores del dispositivo (al menos 2 datos).
 - Además, debe incluirse como adjunto una imagen con una foto del dispositivo (debe ser sacada antes y luego adjuntada).

2.3 Requerimientos no funcionales.

- La APP debe correr en sistemas android superiores a la version 4.0
- El dispositivo móvil debe contar con sensores GPS, WIFI
 - Debe tener instalado un motor de secuencia de comandos que ejecute programas Python en los dispositivos Android (QPython).
 - Debe tener instalado una aplicación que permite configurar el dispositivo Android como servidor SSH. (SSH Server for Android)

2.4 Características de los Usuarios.

El sistema cuenta con un tipo de usuario final:

- Se conforma de usuarios avanzados con respecto a la usabilidad de sistemas embebidos; debe tener conocimientos básicos de redes, "Nivel técnico administrativo", con capacidad de manejo intermedio de comandos de sistemas linux por consola.

2.5 Restricciones de usabilidad.

Como mínimo, se debe hacer uso de los protocolos de intercambio de datos vía internet ya sea por wifi o 3G del dispositivo. El usuario final no ha especificado ninguna otra limitante, y algunos de las características se dejo al criterio de los desarrolladores.

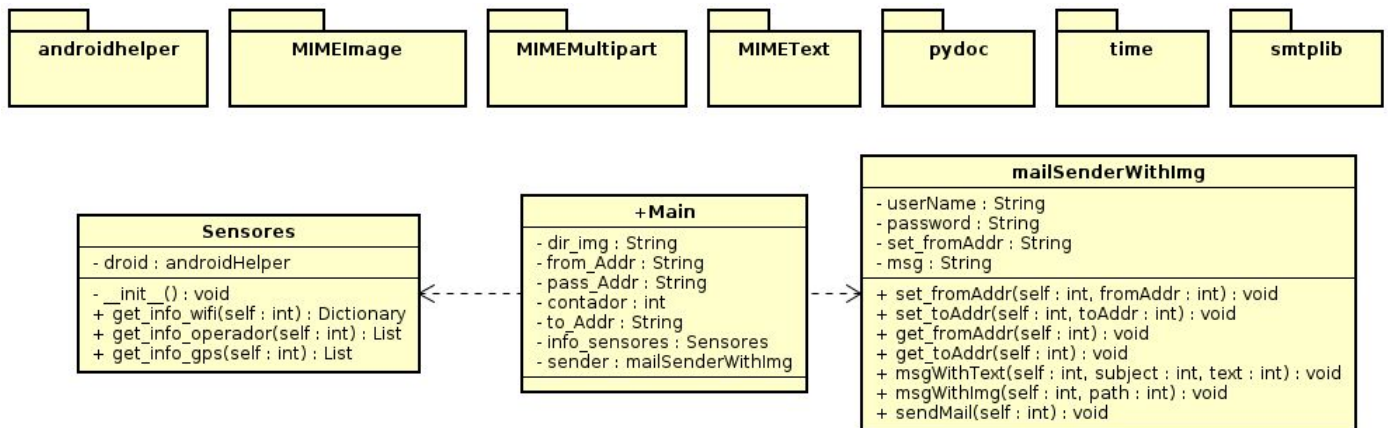
3. Diseño

3.1 Historias de usuario

Especificación de Requerimiento Funcional			
Código	Nombre	Fecha	Grado de necesidad
1	Estado de los sensores.	25-09-2015	Alto
Descripción	Captar el estado de los sensores contenidos en un dispositivo móvil Android, si es que el dispositivo contiene los utilizados en el Script		
Entradas	Fuente		
*****	sensor wifi, sensor GPS		
Salida	Información de estado de los sensores		
Restricciones	Tener los sensores habilitados.		
Efecto colateral	Consumo de 20 segundos para obtener datos.		

Especificación de Requerimiento Funcional			
Código	Nombre	Fecha	Grado de necesidad
3	Envío de email.	25-09-2015	Alto
Descripción	Permite enviar un email con los datos obtenidos de los sensores y la imagen del dispositivo. El asunto del mail está formado por la palabra "Report" y seguido de la estampa de tiempo.		
Entradas	Fuente		
Dirección y contraseña de origen.	Por defecto ' pruebasdisse@gmail.com ' y 'pruebas123', respectivamente.		
Dirección de destino	Entrada estándar.		
Nombre de la Imagen	Definida al momento de guardar la imagen.		
Salida	Mensaje de confirmación indicando si el email se envió correctamente o hubo algún error.		
Restricciones	Necesario implementar requerimientos funcionales 1 y 2.		
Efecto colateral	En caso de no ocurrir un error se enviará un email a la dirección de destino indicada.		

3.2 Diagrama de Clases



4. Implementación

Para llevar a cabo la implementación de nuestro sistema EEIA “Envío de Email iterativo en Android”, se utilizó el lenguaje python, que está provisto de varias librerías que nos permiten codificar sin problemas, dada la gran adaptabilidad y flexibilidad del lenguaje.

Entre las librerías que encontramos en este lenguaje tenemos:

- androidhelper
- time
- smtp
- MIMEImage
- MIMEMultipart
- MIMEText
- pydoc

Las cuales están provistas de la documentación necesaria para hacer un buen uso de las mismas.

Como entorno de desarrollo decidimos utilizar Eclipse IDE y Sublime Text. Estos entornos, nos permiten con respecto a los demás editores, el visualizar el sourceTree generado por el uso de las clases y el orden del código, así logrando una mayor mantención del mismo.

Utilizamos como herramienta de control de versiones GITHUB para alojar proyectos utilizando el sistema de control de versiones Git, para optimizar el trabajo en equipo en el desarrollo.

5. Prueba

Para gestionar las primeras pruebas de nuestra App tuvimos que generar un par de configuraciones antes de poder correr la misma en el dispositivo Android.

- Primero deberíamos en el dispositivo android instalar dos aplicaciones que se encuentran en el GOOGLE PLAY .
 - QPYTHON
 - SSH Server

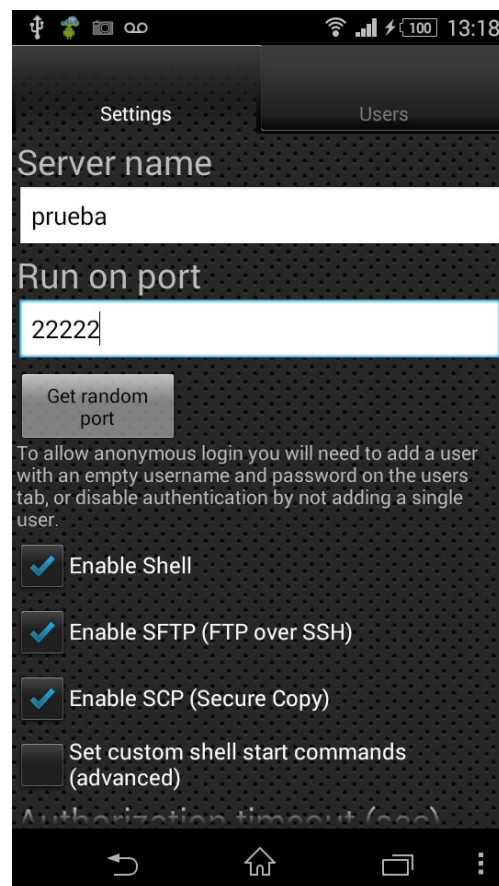
- Una vez que tengamos estas dos App instaladas, pasamos a configurar el SSH Server para lograr mediante este método SSH transferir nuestras primeras versiones del EEIA.

1ero) Ingresar a la App SSH Server

2do) Presionar en Add

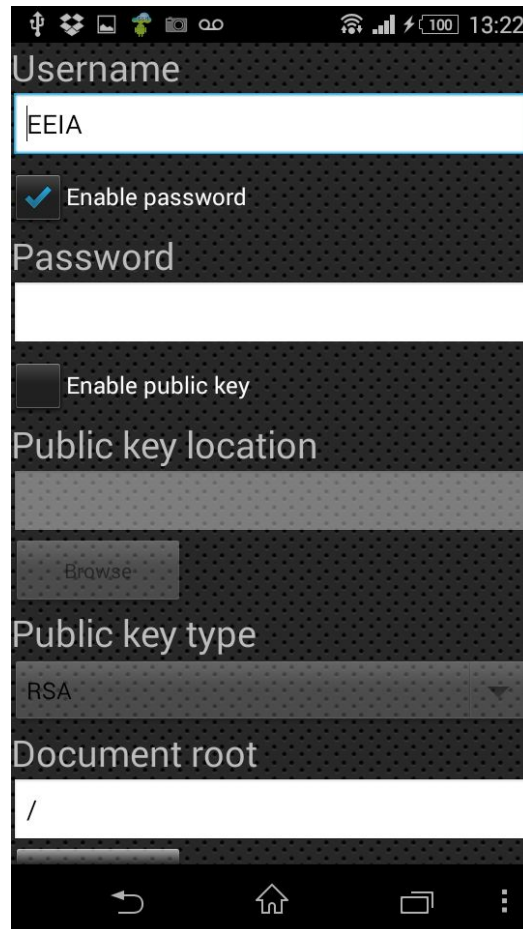
3ero) Aparecerá una nueva pantalla en nuestra App, en los campos **Server name** colocaremos el nombre que diferenciara a nuestro dispositivo como proveedor de servicio SSH. En el campo **Run on port** colocaremos el puerto de comunicación que utilizaremos..

Para nuestro Ejemplo:



4to) Ir a Users y presionar en Add

5to) Configurar un nombre de usuario y un pass . En nuestro caso no ponemos pass.



6to) una vez que tenemos todo configurado presionamos start en el home de la App **SSH Server**.

7mo) Verificamos la Dirección Ip de nuestro dispositivo móvil y hacemos pruebas de conexión con el servidor.

```
Terminal
asus@asus-K53SC ~ $ ping 192.168.1.66
PING 192.168.1.66 (192.168.1.66) 56(84) bytes of data.
64 bytes from 192.168.1.66: icmp_seq=1 ttl=64 time=46.8 ms
64 bytes from 192.168.1.66: icmp_seq=2 ttl=64 time=70.1 ms
64 bytes from 192.168.1.66: icmp_seq=3 ttl=64 time=91.4 ms
64 bytes from 192.168.1.66: icmp_seq=4 ttl=64 time=3.02 ms
64 bytes from 192.168.1.66: icmp_seq=5 ttl=64 time=34.4 ms
64 bytes from 192.168.1.66: icmp_seq=6 ttl=64 time=2.59 ms
64 bytes from 192.168.1.66: icmp_seq=7 ttl=64 time=81.2 ms
64 bytes from 192.168.1.66: icmp_seq=8 ttl=64 time=103 ms
64 bytes from 192.168.1.66: icmp_seq=9 ttl=64 time=23.2 ms

Terminal
asus@asus-K53SC ~ $ ssh -p 22222 EEIA@192.168.1.66
EEIA@192.168.1.66's password:
u0_a229@02306:/ $ ls
acct
cache
charger
config
crashlog
d
data
default.prop
```

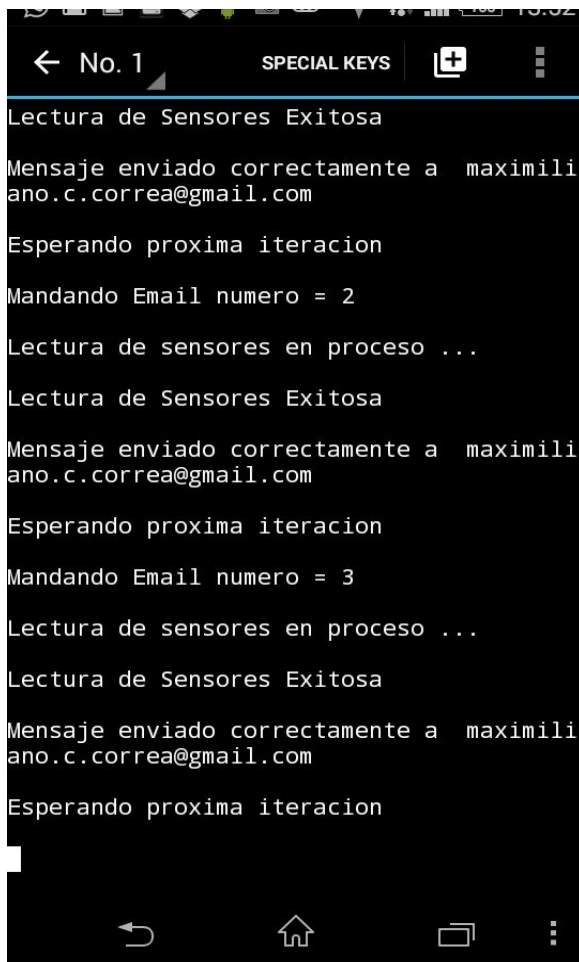
Como vemos en la imagen anterior, una vez dentro del dispositivo móvil podemos ver que estamos en el árbol de directorios de Android ..

8vo) Pasamos el script.py desde nuestro terminal remoto por terminal a nuestro dispositivo android para que la App Qpython pueda leerla y ejecutarla.

scp -P 22222 Final_Python.py

EEIA@192.168.1.66:/storage/emulated/0/com.hipipal.qpyplus/scripts

9no) probar nuestra aplicación con QPython, buscar nuestro script en la direccion que trae por defecto nuestra App.



```
No. 1 SPECIAL KEYS
Lectura de Sensores Exitosa
Mensaje enviado correctamente a maximiliano.c.correa@gmail.com
Esperando proxima iteracion
Mandando Email numero = 2
Lectura de sensores en proceso ...
Lectura de Sensores Exitosa
Mensaje enviado correctamente a maximiliano.c.correa@gmail.com
Esperando proxima iteracion
Mandando Email numero = 3
Lectura de sensores en proceso ...
Lectura de Sensores Exitosa
Mensaje enviado correctamente a maximiliano.c.correa@gmail.com
Esperando proxima iteracion
```

