MANUAL DE USUARIO

1. INTRODUCCIÓN

La programación de dispositivos electrónicos es muy importante al momento de realizar estudios e investigación. Las herramientas informáticas que se utilizan para este fin deben tener lineamientos de operación y funcionamiento muy bien definidos, para optimizar recursos y aprovechar sus ventajas.

Los programas desarrollados en el proyecto de titulación están realizados en el Sistema Operativo Raspbian Jessie, una versión de Linux (de distribución gratuita) utilizada específicamente para trabajar en tarjetas Raspberry Pi.

2. OBJETIVO

Conocer el funcionamiento de los diferentes *scripts* desarrollados, entender su estructura, aplicarlos correctamente, e implementar el sistema aéreo móvil para adquisición de datos meteorológicos georreferenciados junto con el sistema fijo en tierra para el almacenamiento de dicha información trabajando con el sistema operativo Raspbian Jessie.

3. CONSIDERACIONES PREVIAS

- La Raspberry Pi 2B necesita una tarjeta microSD externa donde se encuentra el sistema operativo (SO) para arrancar y trabajar con normalidad.
- Raspbian Jessie es un sistema operativo Linux basado en Debian, de distribución libre y gratuita, especialmente desarrollado para trabajar en placas Raspberry Pi 2 o superiores.
- El sistema operativo Raspbian Jessie puede ser descargado directamente de la página web de Raspberry Pi (https://www.raspberrypi.org/).

- El lenguaje de programación con el cual se desarrollan los diferentes *scripts* es *Python*, un lenguaje interpretado de alto nivel orientado a objetos que permite realizar una programación ordenada y muy bien estructurada.
- El ambiente gráfico de *Python* se compone de dos partes principales: la primera se destina a la programación de códigos, programas y *scripts*, el lenguaje que se utiliza es fácil de entender ya que se parece al inglés técnico, la segunda parte representa el intérprete de *Python*, una herramienta que permite ejecutar un programa sin necesidad de ser compilado.
- Para la configuración de los dispositivos para comunicación inalámbrica se utiliza el *software* XCTU desarrollado por la empresa DIGI.
- La Raspberry Pi 2B necesita de tres periféricos para su funcionamiento: Monitor (HDMI o VGA), teclado y ratón.

4. INSTALACIÓN DE RASPBIAN JESSIE

Desde la página web de Raspberry Pi se descarga gratuitamente el sistema operativo Raspbian en formato .zip, luego de descomprimirlo se obtiene un archivo en formato .img (imagen de disco) con un peso aproximado de 4 GB, como se muestra en la Figura 1.

Figura 1.- Archivo .img descargado.

El archivo *.img* debe ser cargado en una tarjeta MicroSD mediante un *software* adecuado para dicha función, el *software* recomendado es el *Win32DiskImager*, un programa que trabaja en sistemas operativos Windows que puede escribir imágenes de disco en varios tipos de memoria extraíble además de realizar copias de seguridad para recuperar información en caso de daño o errores en los archivos.

Luego de insertar una memoria MicroSD vacía en un computador con Windows, se ingresa en el programa Win32DiskImager el cual pide que se elija una imagen de disco y un dispositivo para escribirla, se procede a seleccionar la imagen de Raspbian Jessie descargada previamente junto con la memoria MicroSD respectiva, posteriormente se selecciona la opción "*Write"* y el proceso de escritura continúa automáticamente durante varios minutos hasta que aparece un mensaje indicando que el proceso ha concluido con éxito.

La Figura 2 muestra una captura de pantalla del proceso de escritura con Win32DiskImager.

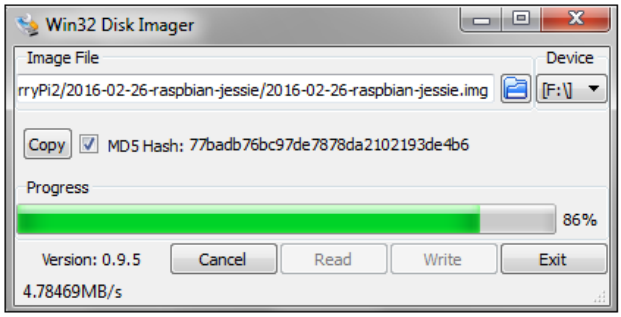

Figura 2.- Escritura del Sistema Operativo en la memoria MicroSD utilizando Win32DiskImager.

La memoria MicroSD debe ser de al menos 8GB, puesto que va a trabajar como un disco duro, en el cual solo el sistema operativo ocupa 4GB y se necesita de memoria adicional para el almacenamiento de otro tipo de información que la Raspberry Pi 2B requiere como aplicaciones o archivos.

5. CONFIGURACIÓN INICIAL

5.1. Autenticación

Para ingresar al ambiente gráfico que ofrece la Raspberry Pi 2B es necesario realizar una autenticación de usuario, ya que generalmente aparece una pantalla como la mostrada en la Figura 3.

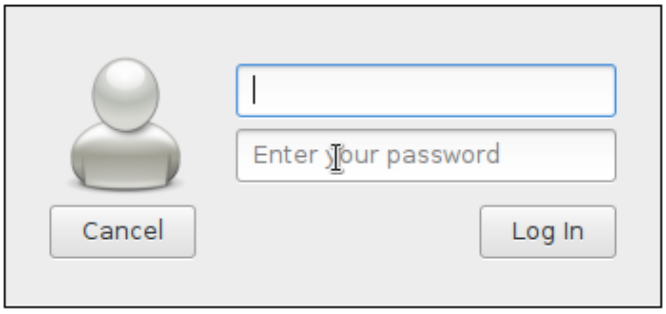

Figura 3.- Pantalla de Autenticación de Usuario.

La configuración predeterminada establece como nombre de usuario: "*pi*" y como contraseña: "*raspberry*". En caso de realizar este paso desde la consola, también es necesario ejecutar el comando "*startx*".

5.2. Habilitar los pines GPIO

Para que los pines *GPIO* de la Raspberry Pi 2B puedan ser habilitados se necesita ejecutar desde la línea de comandos una instrucción que activa al "*demonio*" *pigpiod*. Dicha instrucción es la siguiente:

"sudo pigpiod"

El comando sudo permite a un usuario (*pi*) ejecutar programas con privilegios de seguridad de administrador (*root*) convirtiéndose temporalmente en superusuario de forma segura.

5.3. Fecha y Hora en Raspberry Pi 2B

En vista que la tarjeta Raspberry Pi 2B no cuenta con un reloj interno que permita igualarse automáticamente, su actualización es posible de dos maneras: la primera es conectando el dispositivo a internet, ya que cuenta con un servidor *NTP* que actualiza la fecha y hora al realizar dicha conexión, la segunda opción es manualmente mediante el siguiente comando:

“*sudo date --set “AAA-MM-DD HH:MI”* ”

Donde:

- *AAAA-MM-DD*: representa la fecha en formato *año-mes-día*.
- *HH:MI*: Representa la hora en formato *hora:minutos* (24 horas).

La Figura 4 muestra el comando utilizado para la actualización horaria y su comprobación al ser ejecutado.

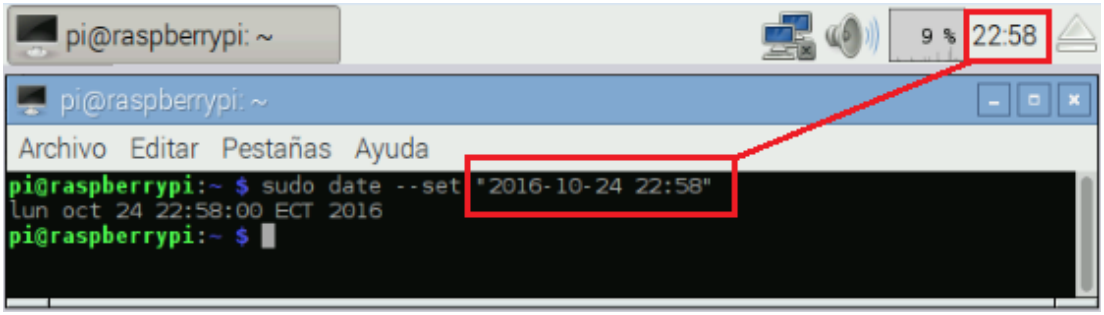

Figura 4.- Actualización de Fecha y Hora.

6. PROGRAMA XCTU

XCTU es el *software* que se utiliza para realizar la configuración y pruebas de funcionamiento de los módulos XBee Pro S1, es una aplicación gratuita multiplataforma compatible con Windows, Linux y MacOS, distribuida libremente por Digi a través de su página web (https://www.digi.com/).

Su interfaz gráfica permite realizar el diseño de una red ZigBee de manera fácil con la configuración de:

- *Parámetros de comunicación*: identificador de red, canal, velocidad, direcciones MAC, número de puerto.
- *Tipo de dispositivos XBee*: controlador, router, terminal e
- Representación gráfica de la red con la intensidad de señal en cada conexión. La Figura 5 muestra el ambiente gráfico de XCTU.

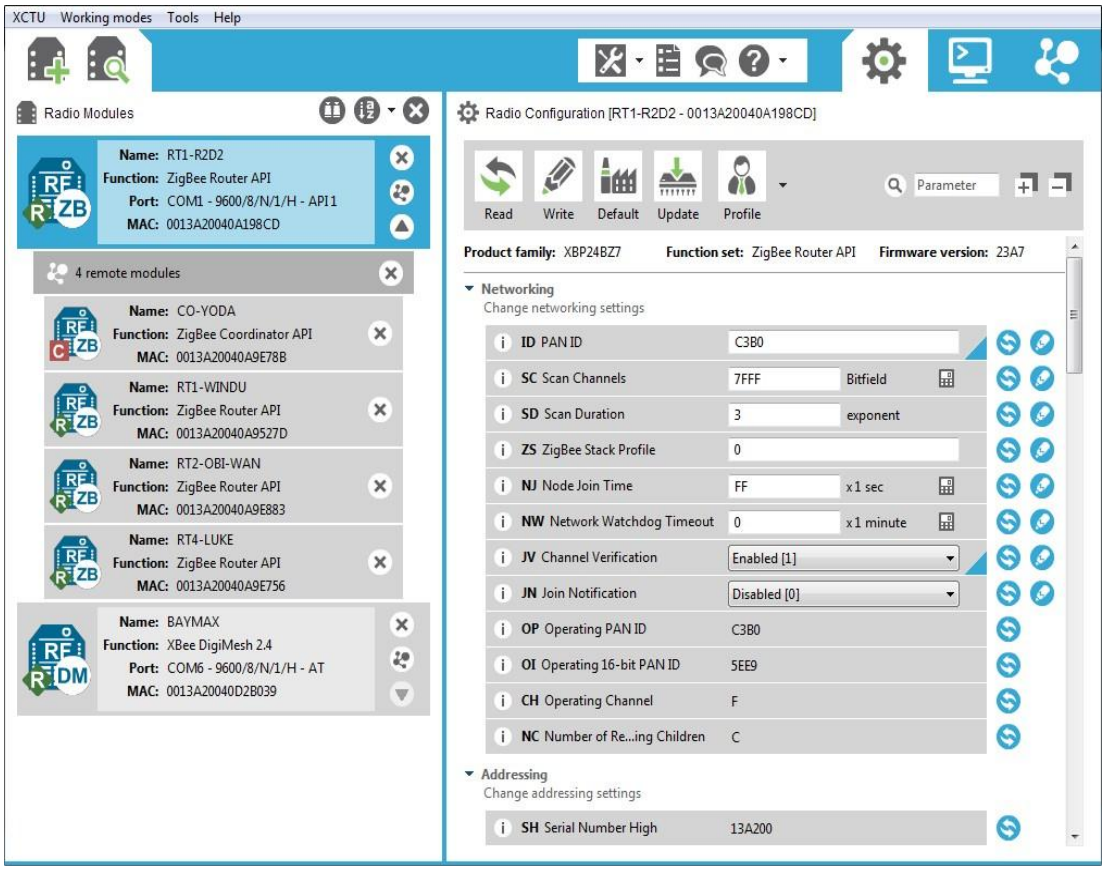

Figura 5.- Interfaz gráfica XCTU.

6.1. Configuración de Dispositivos XBee con XCTU

Luego de conectar los dispositivos XBee a un computador mediante cable USB, se procede a configurar sus principales parámetros de comunicación, además de la especificación de la red que forman ambos equipos.

El *software* XCTU permite elegir los puertos seriales donde se realiza la búsqueda de los dispositivos XBee, luego se escogen los parámetros del puerto (velocidad, bits de datos, bits de parada, paridad y control), una vez determinados tales datos, el *software* trabaja y genera información de los dispositivos encontrados.

La Figura 6 muestra la información necesaria para realizar la búsqueda de los dispositivos XBee.

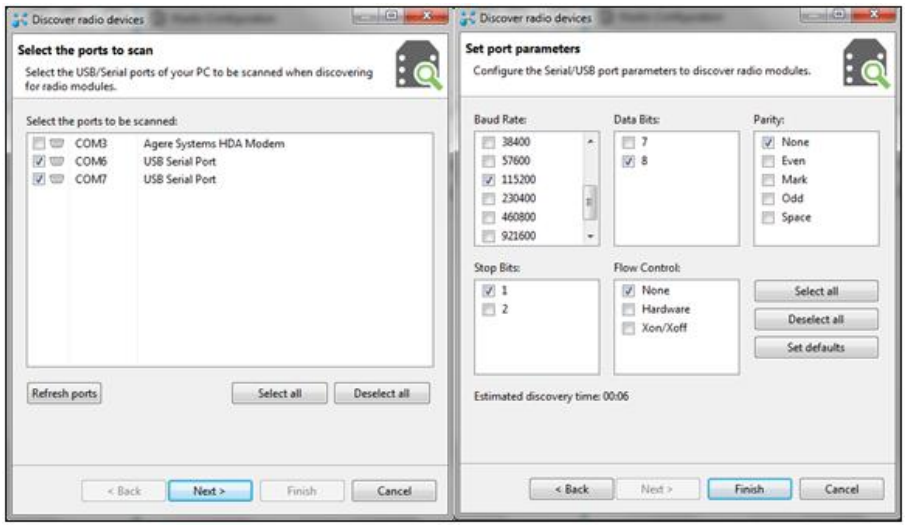

Figura 6.- Información requerida por XCTU.

7. PROGRAMAS EJECUTABLES

7.1. Sistema Aéreo Móvil

Existen varios *scripts* desarrollados en *Python* que permiten al sistema aéreo la adquisición, almacenamiento y transmisión de parámetros meteorológicos, mediante la programación de sus dispositivos electrónicos.

7.1.1. Scripts Desarrollados

Los cinco *scripts* desarrollados son los siguientes:

- *Script_SensorBMP180*.- permite realizar mediciones de temperatura y presión atmosférica con el sensor BMP180.
- *Script_SensorDHT22*.- permite realizar mediciones de humedad relativa y temperatura con el sensor DHT22.
- *Script_VectorMeteorologico*.- genera un vector de tres parámetros (temperatura, humedad,presión) cada minuto.
- *Script_RecepcionGPS*.- permite recibir la señal GPS y crear un vector de cuatro parámetros con la posición geográfica del sistema móvil.

- *Script_SistemaMovil*.- reúne varias funciones para adquirir datos meteorológicos y enviarlos inalámbricamente al sistema ubicado en tierra.

La Figura 7 muestra los *scripts* desarrollados en lenguaje *Python* (formato *.py*).

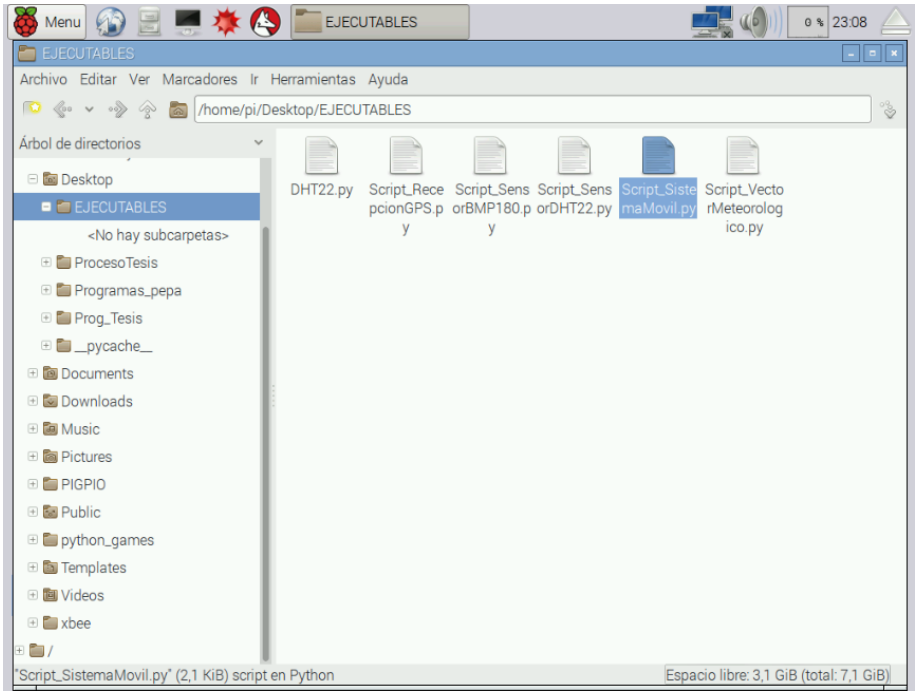

Figura 7.- *Scripts* del sistema aéreo.

La figura anterior muestra el archivo "*DHT22.py*" dentro de la carpeta "*EJECUTABLES*", dicho archivo es el controlador del sensor DHT22 desarrollado por la empresa *Adafruit,* que contiene las funciones necesarias para su configuración. Dicho archivo debe permanecer en la misma carpeta donde se ejecuten *scripts* que involucren al sensor DHT22.

7.1.2. Ejecución de Programas

La ejecución de los programas desarrollados es sencilla, para ello es necesario abrir el *script* en el ambiente de configuración, posteriormente en la barra de herramientas ubicada en su parte superior, se selecciona la opción "*Run*" como se muestra en la Figura 8.

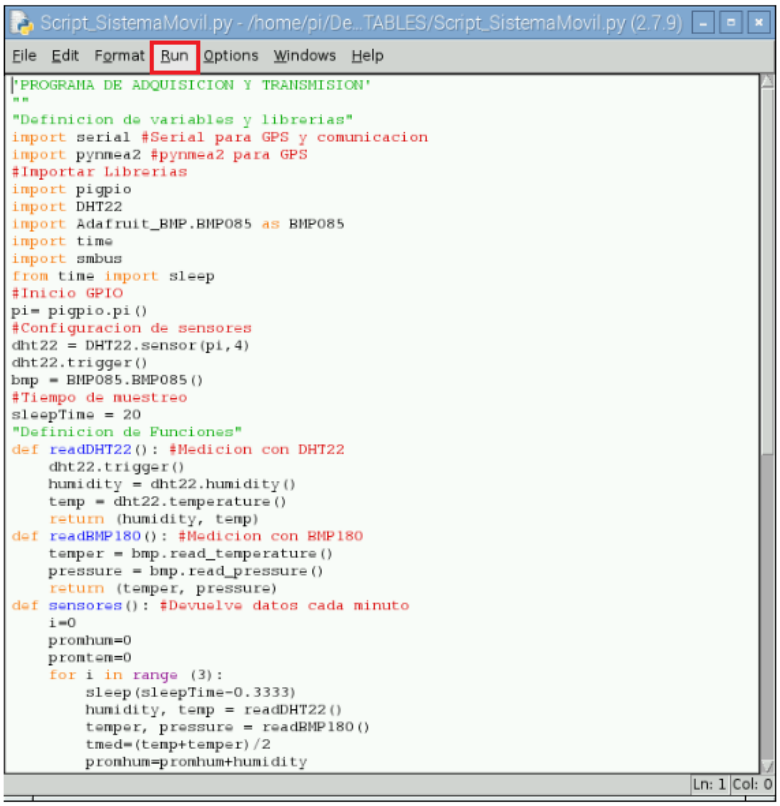

Figura 8.- Ejecución de un *script* para el sistema aéreo.

Finalmente se selecciona la opción "*Run Module*", de esta forma la ejecución se completa y se puede apreciar los resultados en el Intérprete de *Python*.

7.2. Sistema en Tierra

El sistema en tierra permite la recepción inalámbrica de información meteorológica y su posterior almacenamiento en archivos de texto.

7.2.1. Scripts Desarrollados

Para la implementación del sistema en tierra se han desarrollado dos *scripts* dependiendo del hemisferio donde se realizan las mediciones:

- *Script_FijoNorte*.- permite recibir la información y almacenarla, cuando las mediciones se realizan en el hemisferio Norte.

- *Script_FijoSur*.- permite recibir la información y almacenarla, cuando las mediciones se realizan en el hemisferio Sur.

La Figura 9 muestra los *scripts* desarrollados en lenguaje *Python* (formato *.py*).

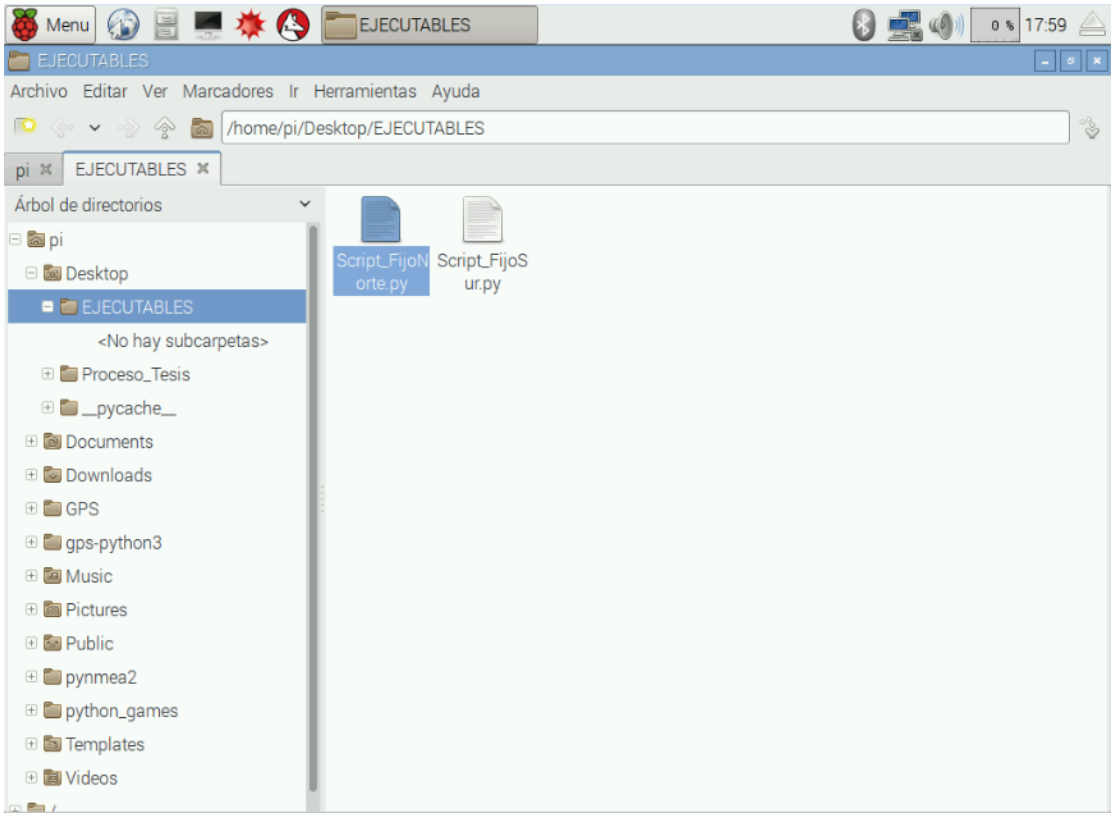

Figura 9.- *Scripts* del sistema en tierra.

7.2.2. Ejecución de Programas

La ejecución de los s*cripts* desarrollados se realiza de la misma manera que el sistema aéreo, luego de abrir el código en *Python*, se selecciona la opción "*Run*" de la barra de herramientas, posteriormente se selecciona "*Run Module*". La ejecución es exitosa cuando el intérprete de *Python* no genera alertas ni errores, como se muestra en la Figura 10.

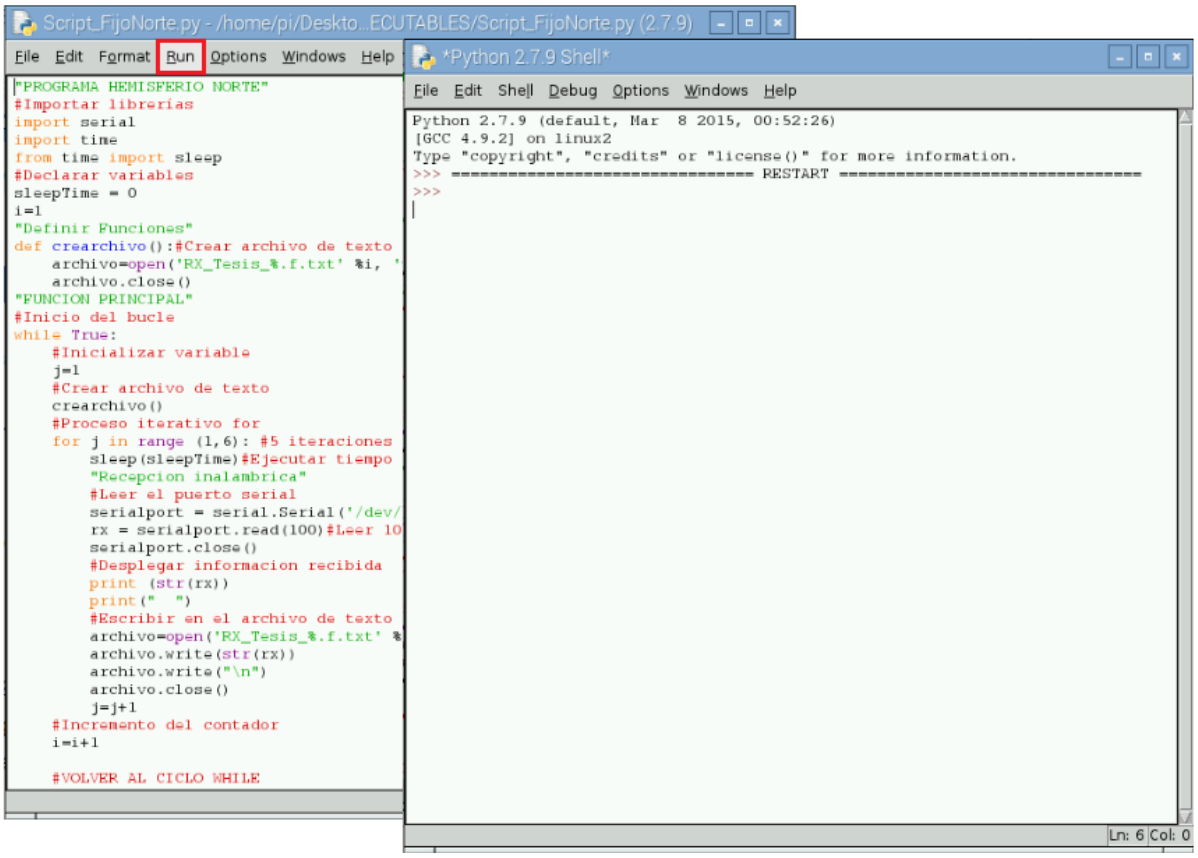

Figura 10.- Ejecución de un *script* para el sistema en tierra.