**MANUAL DE USUARIO**

**Sistema de Detección de Incendios**

**Índice**

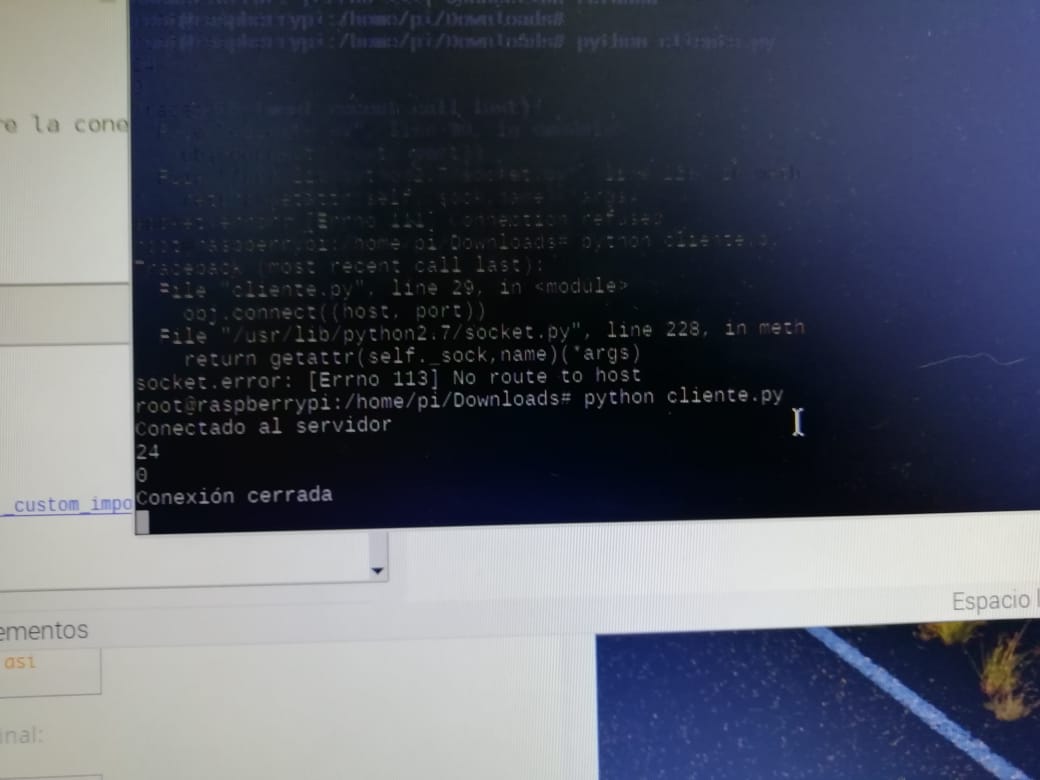
[Conexión del cliente Raspberry con el servidor 1](#_Toc535757697)

[Conexión del servidor CentOS 7 con diversos clientes 2](#_Toc535757698)

[Menú de Administración de Base de Datos 3](#_Toc535757699)

El presente manual tiene como finalidad facilitar el uso del sistema de detección de incendios en lo que respecta a conexión desde un cliente hacia el servidor y viceversa, tratando lo concerniente a cómo y qué scripts usar, además de cómo usar el menú de administración de base de datos, el cual permitirá editar, consultar o eliminar datos de la BD, así como también generar gráficos estadísticos de incendios sensados por cada nodo sensor.

**Conexión del Cliente con el Servidor**

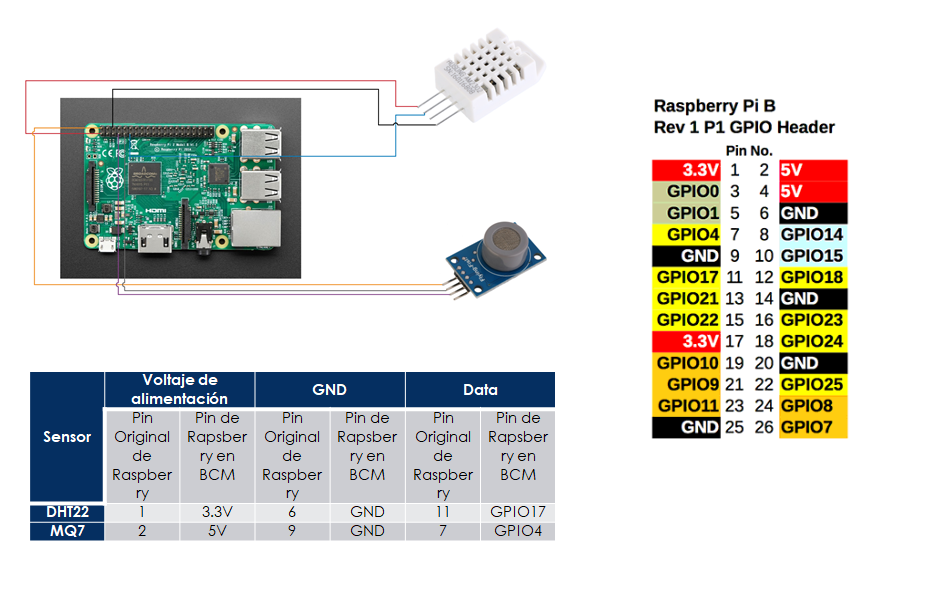
****

*Ilustración 1. Resultado de ejecución de script cliente.py*

Para levantar la conexión desde el cliente hacia el servidor, es necesario contar con el script ‘cliente.py’ ubicado en un directorio determinado. Después, para ejecutarlo, se usa el comando ‘python cliente.py’. Se mostrará el nivel de temperatura y de gas sensado a través de la consola de comandos; en la Ilustración 1 se puede apreciar que al momento de la captura la temperatura sensada fue de 24 ̊C, y el nivel de concentración de gas es de 0.

Con el uso de la librería ‘timer’, se establece un tiempo para una nueva sesión de sensado y posterior envío de datos. Esto puede ser configurado asignando valor de tiempo en la función ‘time.sleep(t)’, donde t está en segundos.

Cabe destacar que previo a la ejecución del script, se debe realizar el cableado respectivo de forma correcta, siguiente como esquema el diagrama circuital del Proyecto (Ilustración 2).

****

*Ilustración 2. Diagrama que ilustra la forma de conector los sensores*

**Conexión del Servidor con el Cliente**

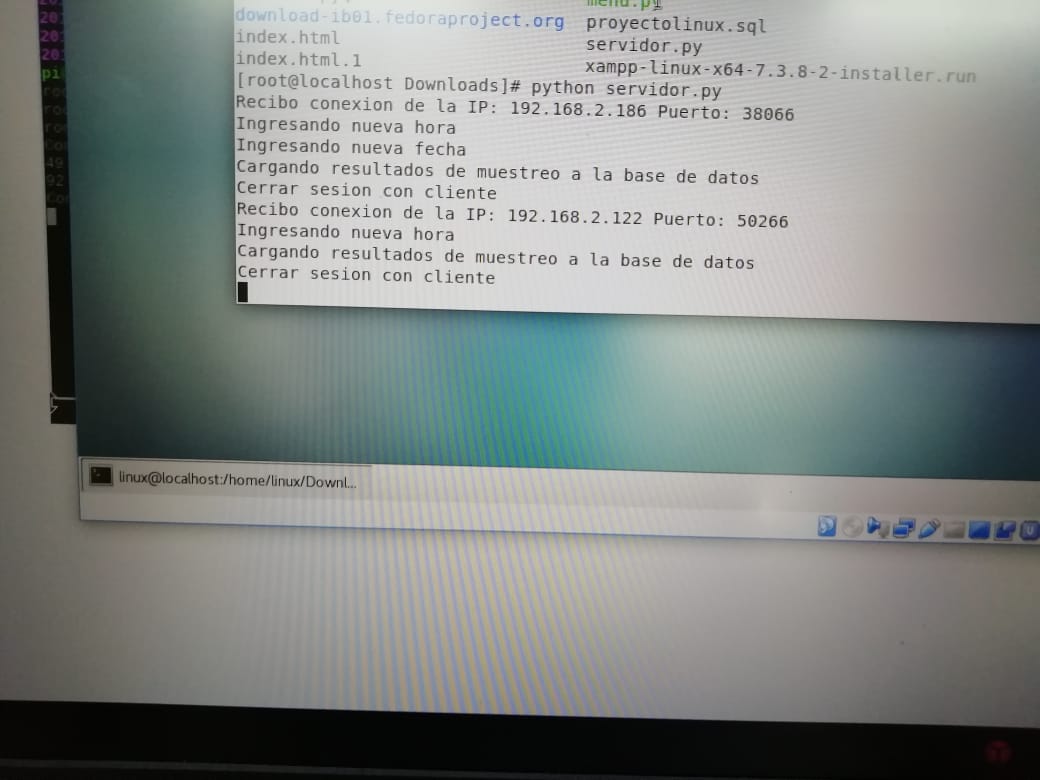
De manera similar a como se procedió con el cliente, del lado del servidor se debe contar con el script ‘servidor.py’ en un determinado directorio, y desde el mismo se debe ejecutar el script mencionado. Un detalle de suma importancia radica en que el servidor debe estar levantado en primera instancia, antes de ejecutar el script ‘cliente.py’ del lado del cliente; si no se respeta este punto ocurrirá un error de conexión de sockets.

La conexión a través de los sockets se sostiene mediante una red, la cual engloba tanto al cliente como al servidor; por ello, el despliegue de la red es otro aspecto importante a considerar previo a la ejecución del script de servidor, o de cliente, si no se ven a través de una red, la conexión mediante sockets no será exitosa.

Los procesos de servidor y de cliente se ejecutan en segundo plano, transparentado dichos algoritmos y permitiendo que el administrador del servidor visualice únicamente el menú de administración de la base de datos.

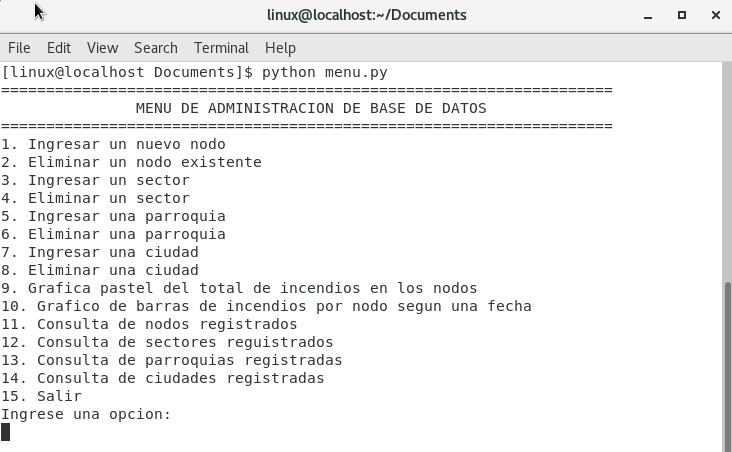
Nuevamente, se ejecuta el script con el comando ‘python servidor.py’, dando como resultado lo mostrado en la Ilustración 3.

El proceso de inserción de datos en la base de datos es transparente para el usuario, y se lo confirma con un mensaje a través de la pantalla “Cargando resultados de muestreo a la base de datos”. Como resultado, el usuario no tiene que preocuparse del ingreso de datos en la BD porque el proceso se lo realiza de forma correcta y abstraída para brindar comodidad.

****

*Ilustración 3. Resultado de ejecución del script servidor.py*

**Menú de Administración de Base de Datos**



*Ilustración 4. Ejecución del script para menú de administración*

Del lado del servidor, mientras se ejecuta el script ‘servidor.py’ en segundo plano, se usa nuevamente el comando ‘python’ para correr el archivo de menú de administración. Al ejecutar dicho archivo se abre el menú, el cual está constituido por 15 opciones.

Entre las opciones se tienen operaciones de ingreso, eliminación y consulta de información almacenada en la base de datos, así como también 2 funcionalidades para generar gráficas estadísticas relacionadas con los incendios suscitados.