

2021

Eric García Carrizo

[DOCUMENTACIÓN TECNICA DE LA SOLUCIÓN]

INDICE

1.-Gateway MQTT	4
1.1.- instalación y configuración básica del Bróker Mosquitto	4
1.2.- Configuración Bridge.....	5
1.3.-conexión bluetooth con el modulo de sensores.....	5
1.4.-Sistema de alarmas Off-Line	8
2.-Plataforma de procesamiento Cloud	11
2.1.-Creacion de la maquina virtual.....	11
2.2.-Conexión a la maquina virtual.	15
2.3.-Instalación del bróker MQTT	15
2.4.- Base de datos	16
2.5.-Guardado de datos recibidos desde MQTT en la base de Datos	18
2.6.-Comunicación App-Servidor	20
2.7.-Envio de archivos por SCP mediante WinSCP	21
3.-Visualizacion de datos	23
4.-Anexos.....	23

1.-Gateway MQTT

1.1.- instalación y configuración básica del Bróker Mosquitto

Se procede a instalar el bróker MQTT en la raspberry pi. Para ello se introducen los siguientes comandos en la línea de comandos:

Actualizar la raspberry

```
sudo apt update
```

Instalar el Broker

```
sudo apt install -y mosquitto mosquitto-clients
```

Hacer que Mosquitto se inicie automáticamente cuando se inicie el sistema

```
sudo systemctl enable mosquitto
```

Iniciar Mosquitto

```
sudo systemctl start mosquitto.
```

Detener Mosquitto

```
sudo systemctl stop mosquitto.
```

Subscripción a un topic

```
Mosquitto_sub -h ipBroker -t "tópicoA/topicA1 -v
```

Publicación del topic

```
Mosquitto_pub -h ipBroker -t "tópicoA/topicA1 -m Datos
```

Para que esta conexión sea segura se establece un usuario, una contraseña. Esto es posible hacerlo siguiendo los siguientes pasos:

Usando *Mosquitto_passwd* se crea un archivo en la ruta especificada. El sistema solicitará que se introduzca una contraseña (**pass1**) para el usuario introducido (**user1**).

```
sudo mosquitto_passwd -c /etc/mosquitto/nombreArchivo nombreUsuario
```

Se abre el archivo de configuración usando un editor de texto. Por ejemplo, *nano*.

```
sudo nano /etc/mosquitto/conf.d/default.conf
```

Se añaden las dos líneas mostradas para, por un lado, impedir el acceso a usuarios anónimos, y por otro, indicar cual es el archivo donde se encuentran estos usuarios y contraseñas.

```
allow_anonymous false
```

```
password_file /etc/mosquitto/passwordFile
```

A partir de ahora se indicará con **-u** y **-P** el usuario y la contraseña, respectivamente, a la hora de suscribirnos y publicar.

1.2.- Configuración Bridge

Como se ha explicado en la documentación funcional nuestra solución requiere de dos brokers MQTT, uno en la red local y otro en el servidor en la nube. Para poder realizar este tipo de conexión necesitamos configurar en cualquiera de los dos brokers un bridge. Este se encarga de que los mensajes que lleguen a uno de los dos brokers, se envíen automáticamente al otro, en este caso los vamos a configurar para que los mensajes que lleguen al bróker local se envíen automáticamente al segundo.

Para una explicación en del funcionamiento en detalle de cómo funciona el bridge, ir al apartado "configuración del bridge de los anexos

Para configurar el bridge tenemos que editar el archivo **mosquitto.conf** en la ruta **etc/mosquitto/** y añadir al archivo las siguientes líneas

```
#configuracion del "bridge para la conexion del broker
#local con un broker en uuna red externa

#nombre del bridge
connection aws_bridge

#direccion del servisor
address 34.240.3.254:1883

#indicamos el comportamiento del bridge para los mensajes, en este ord
#indicando los topics que queremos enviar y, o recibir
#si publicamos desde el broker (out) o recibimos desde el remoto (in)
#QoS
#prefijo local /prefijo remoto (opcionales)
```

De este modo el bróker remoto podrá recibir los mensajes que se intercambien en la red local.

1.3.-conexión bluetooth con el modulo de sensores

Hay que tener en cuenta si el dispositivo al que nos vamos a conectar es bluetooth o BLE (bluetooth Low Energy), ya que los pasos que hay que seguir para establecer la conexión son distintos.

En el caso de la sensortilebox de st la comunicación es BLE, por lo que para conseguir conectarnos al dispositivo y emparejarnos con este, se necesita realizar los siguientes cambios en el funcionamiento de bluetooth en el sistema operativo:

Abrir el archivo en la siguiente ruta

```
CD etc/bluetooth/main.conf
```

Cambiar los siguientes parámetros como se indica

```
privacy = off
ControllerMode = ble
```

Abrir el archivo en la siguiente ruta

```
etc/systemd/system/bluetooth.target.wants/bluetooth.service
```

Y en el apartado "service" completar la línea de ExecStart del siguiente modo

```
ExecStart=/usr/lib/bluetooth/bluetoothd --compat --noplugin=sap
```

Instalar los siguientes paquetes

```
sudo pip3 install bluepy  
sudo pip3 install blue-st-sdk
```

Activamos la búsqueda de dispositivos bluetooth

```
sudo bluetoothctl  
scan on  
pair <address>
```

Introducimos la contraseña del dispositivo y comprobamos que nos conectamos.

Para comprender los siguientes pasos se recomienda conocer, el funcionamiento del protocolo GATT (características, servicios, UUIDs, etc.) esta explicación se puede encontrar en el link del anexo "explicación del protocolo GATT.

Ahora que podemos conectarnos a los dispositivos de st, comprobamos que podemos suscribirnos a los servicios y características y recibir la información de estos, antes de realizar el programa:

Accedemos a las instrucciones para el control de bluetooth

```
Bluetoothctl
```

Buscamos los dispositivos en la red

```
Scan on
```

Nos conectamos a nuestro dispositivo introduciendo su mac con el siguiente comando

```
Connec <MAC>
```

Accedemos a los comandos gatt

```
Menú gatt
```

Buscamos los atributos disponibles

```
List-attributes
```

Seleccionamos el atributo del que queramos obtener información

```
Select-atibute <attributeUUID>
```

Nos suscribimos para recibir la información

```
Notify on
```

Nos desuscribimos

```
Notify off
```

A continuación instalaremos un dongle usb Bluetooth 5.0, esto es debido a que tras poner en funcionamiento el programa, el cual se va a explicar a continuación. Se ha comprobado, que de forma recurrente el adaptador bluetooth incorporado en la propia raspberry se desactiva, y no hay ningún modo de volver a reactivarlo por software, solamente quitando la alimentación al adaptador, es decir, reiniciando la raspberry, razón por la cual se ha decidido usar un adaptador externo para evitar estos constantes fallos.

Para hacer funcionar el adaptador hay que instalar los drivers necesarios, para ello hacemos lo siguiente.

Averiguamos información sobre el modulo usb para ello escribimos el siguiente comando

Lusb

```
pi@raspberrypi:~$ lsusb
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0bda:8771 Realtek Semiconductor Corp.
Bus 001 Device 002: ID 2109:3451 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Observamos entre otros datos el identificador del adaptador, en este caso Obda:8771, que corresponde al chip bluetooth "RTL8761B".

De este chip habrá que descargar los drivers correspondientes, un ".fw" y un ".config", y ubicarlos en una ruta concreta.

Vamos a la siguiente carpeta (donde están los drivers de los dispositivos bluetooth).

usr/lib/firmware/rtl_bt

Con wget indicamos la url de descarga y con -O indicamos la ruta donde lo vamos a guardar y el nombre con el que lo vamos a guardar (notese que se deben guardar como ".bin")

```
Wget https://raw.githubusercontent.com/Realtek-OpenSource/android_hardware_realtek/rtk1395/bt/rtkbt/Firmware/BT/rtl8761b_config -O rtl8761b_config.bin
```

```
Wget https://raw.githubusercontent.com/Realtek-OpenSource/android_hardware_realtek/rtk1395/bt/rtkbt/Firmware/BT/rtl8761b_fw -O rtl8761b_fw.bin
```

Finalmente reiniciamos la raspberry.

Comprobamos que se puede recibir información de forma constante si que el sistema se caiga.

Una vez comprobado que podemos recibir información realizamos el programa que crea un cliente GATT que se puede suscribir a una serie de características y servicios. Este necesita las siguientes librerías

Para interactuar con bluetooth

sudo pip3 install pygatt

```
sudo pip3 install pygatt[GATTTOOL]
```

Para poder usar la librería mqtt

```
sudo pip3 install paho-mqtt
```

Para poder interactuar con los pines de la raspberry

```
sudo apt-get install python-dev
```

```
sudo apt-get install python-rpi.gpio
```

otras librerías necesarias para el proyecto

```
sudo pip3 install numpy
```

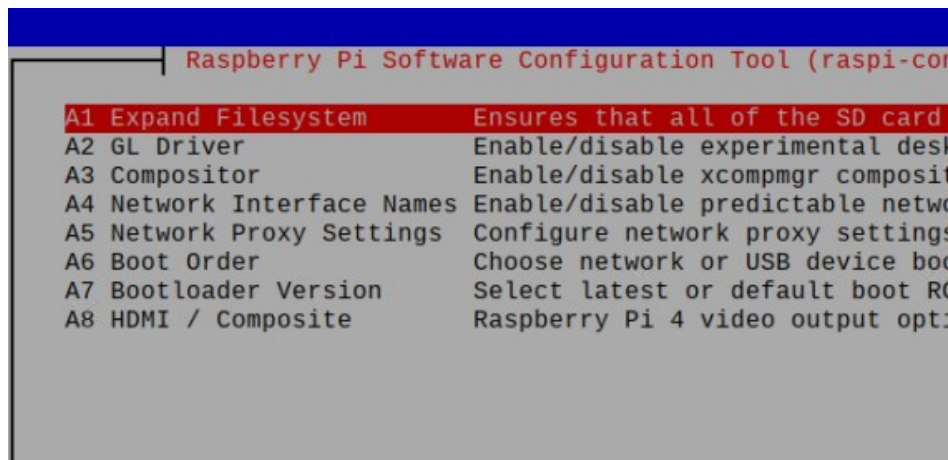
El funcionamiento del sistema de comunicación esta explicado en el propio código, no obstante una explicación superficial del funcionamiento sería el siguiente.

- Leemos que dispositivos hay, y que atributos tienen estos dispositivos (todos tienen los mismos)
- Guardamos los dispositivos y los atributos de estos cada, uno en una lista
- Creamos una matriz de timers que establece cuando se lee que dispositivo y que atributo, hay tantos timers como el numero de atributos por el numero de dispositivos.
- Chequeamos cada dispositivo uno a uno
- En cada dispositivo chequeamos sus atributos uno a uno, si aun no es momento de mandar a la información no se hace nada
- Si se ha alcanzado el momento estipulado, nos suscribimos al atributo del dispositivo seleccionado y recogemos su valor, lo estructuramos de acuerdo a la estructura de datos, especificada en el documento **"gatt implementation st"**, que se encuentra en la ruta **documentacion proyecto\documentacion tecnica\bibliografia** y lo mandamos a bróker, por ultimo reseteamos el contador.

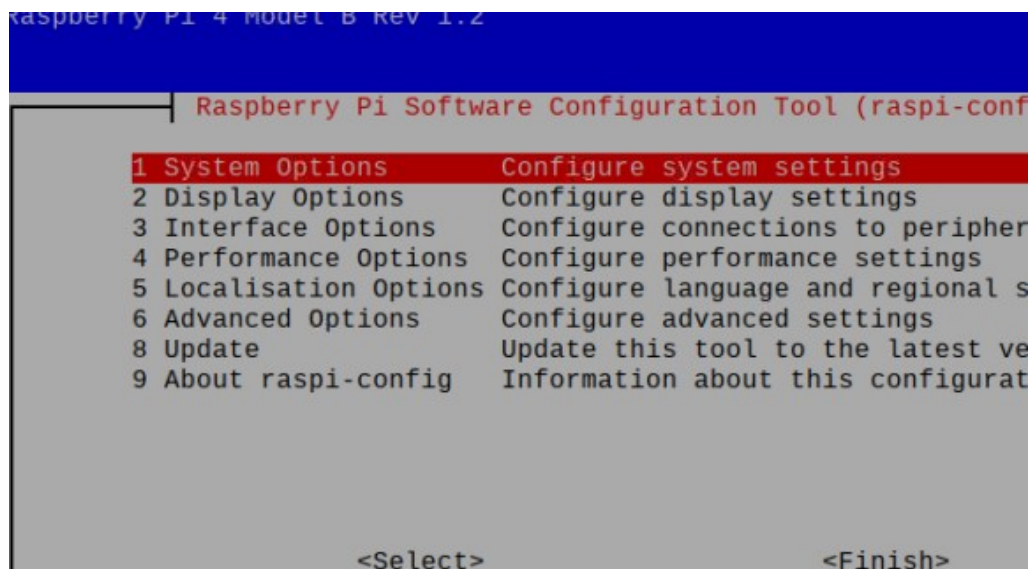
1.4.-Sistema de alarmas Off-Line

Como según la documentación funcional se establece que debe de haber algún sistema de alarmas que no dependa de la conexión a un servidor cloud, se va a explicar el procedimiento para la poder utilizar un sistema de megafonía con la raspberry.

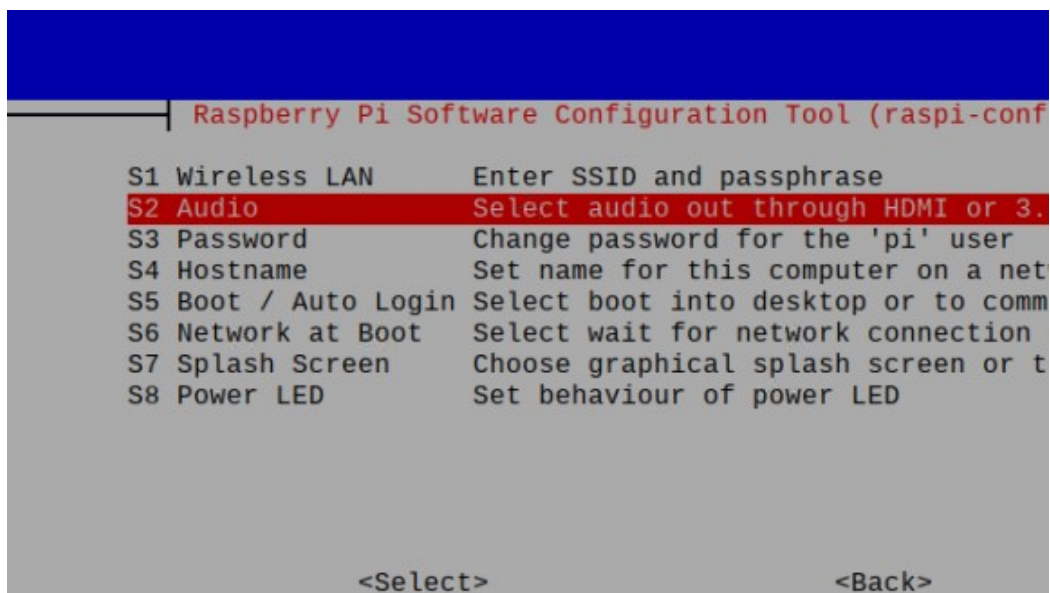
Entramos a la configuración y elegimos la opción **A1 Expand Filesistem**



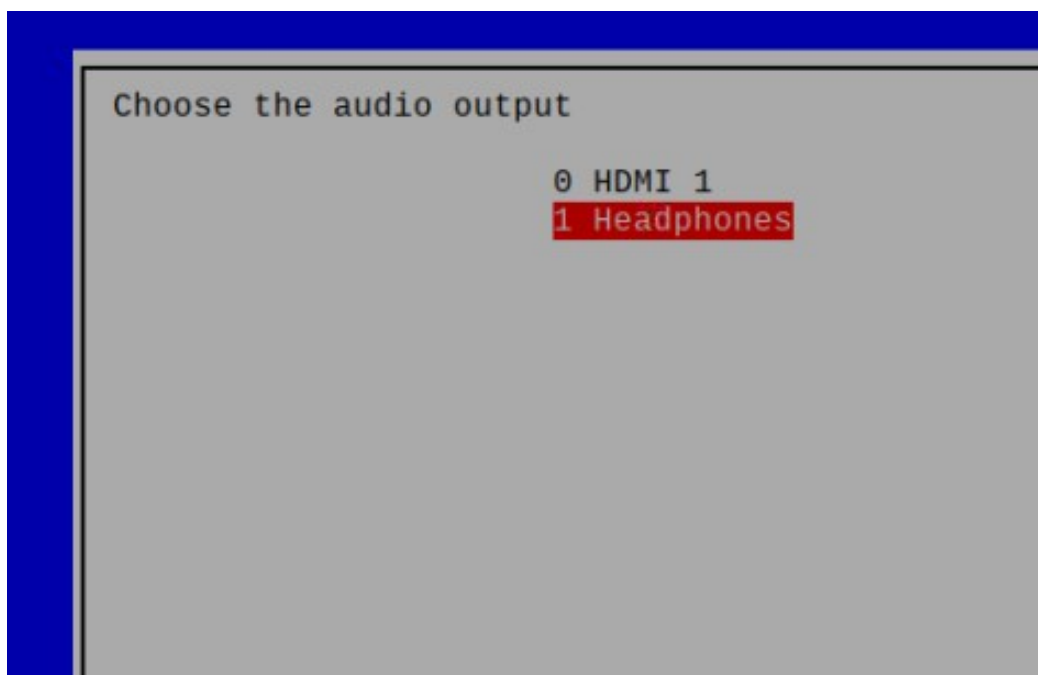
Dentro, seleccionamos **system options**.



Seleccionamos la opción S2 audio para seleccionar la fuente de audio predeterminada.



Seleccionamos **headphones** y reiniciamos la raspberry.



A continuación en el del programa de python que recoge los datos del bluetooth introducimos una función que reproduzca un mensaje de audio previamente guardado por el altavoz.

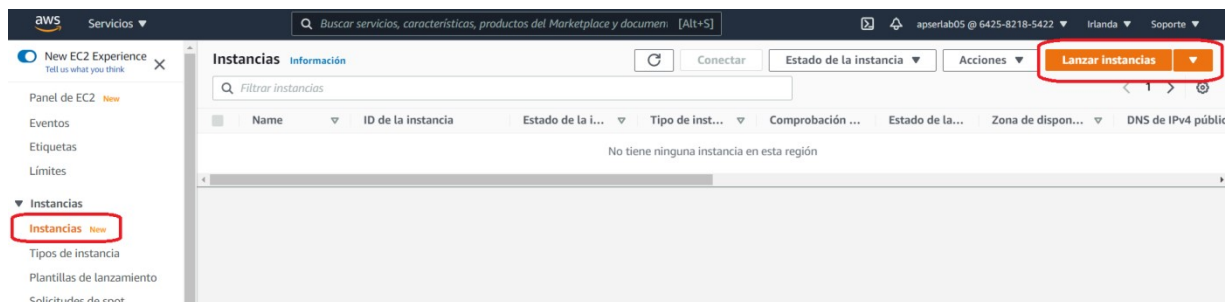
Tener en cuenta que el altavoz tiene que tener una impedancia apropiada para poder funcionar con la raspberry (4 Ohmios) o tener un amplificador adecuado.

2.-Plataforma de procesamiento Cloud

2.1.-Creacion de la maquina virtual

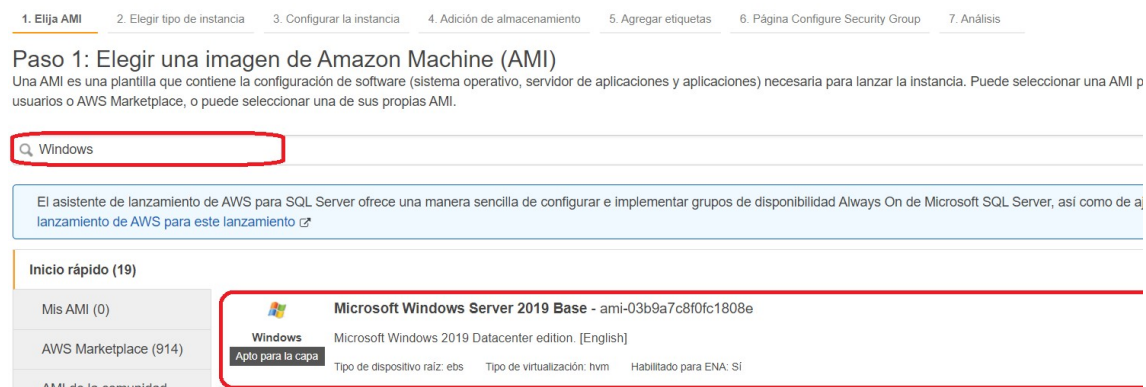
En función de los criterios de desarrollo se ha decidido construir la solución cloud sobre AWS usando el servicio EC2 (maquina virtual en la nube), para crearla seguimos los siguientes pasos.

Creamos una cuenta en AWS, dentro de los servicios que nos ofrece seleccionamos EC2, hacemos clic en instancia en la barra lateral izquierda y seleccionamos lanzar instancia.



Luego seleccionamos el sistema operativo de la maquina virtual.

Para ello escribimos el sistema operativo que deseemos en la parte superior, en nuestro caso debían, seleccionamos la versión de mas reciente y hacemos clic en siguiente.



A continuación escogeremos el tipo de máquina que queremos y el desempeño que esperamos obtener de ella, en este caso seleccionaremos el tipo t2.micro, enfocada al uso general y poca capacidad de computo, si luego viésemos que son pocos recursos, cambiaríamos a una versión con mayor rendimiento.

Paso 2: Página Choose an Instance Type

Amazon EC2 proporciona una amplia selección de tipos de instancias optimizados para adaptarse a diferentes casos de uso. Las instancias son servidores virtuales que pueden ejecutar aplicaciones, memoria, almacenamiento y capacidad de red, lo que proporciona una gran flexibilidad para elegir la combinación de recursos adecuada para las aplicaciones. [Más información](#) acerca de los tipos de instancias y sus necesidades de computación.

Filtrar por: Todas las familias de instancias Generación actual Mostrar/ocultar columnas

Seleccionada actualmente: t2.micro (- ECU, 1 vCPU, 2.5 GHz, -, 1 GiB memoria, EBS solo)

	Familia	Tipo	vCPU	Memoria (GiB)	Almacenamiento de la instancia (GB)	Optimizado para EBS disponible	De
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS solo	-	
<input checked="" type="checkbox"/>	t2	t2.micro <small>Apto para la capa gratuita</small>	1	1	EBS solo	-	

Pulsamos siguiente para continuar con la configuración.

Configuramos otros detalles de la instancia, podemos dejar todos como vienen por defecto, pero nos aseguraremos de que en la sección, **comportamiento de cierre** esta seleccionado **detener**, de otro modo cuando se apague la instancia, se perderá toda la información en ella almacenada.

Hacemos clic en siguiente para continuar con la configuración.

Paso 3: Página Configuración de los detalles de la instancia

Configure la instancia adecuada a sus requisitos. Puede lanzar varias instancias desde la misma AMI, solicitar instancias de spot para aprovechar precios más bajos, o configurar una instancia para que se ejecute en un grupo de Auto Scaling.

Número de instancias: 1 [Lanzar en grupo de Auto Scaling](#)

Opción de compra: ☐ Solicitar instancias de spot

Red: vpc-62877e1b (predeterminada) [Crear nueva VPC](#)

Subred: Sin preferencia (subred predeterminada de cualquier VPC) [Crear nueva subred](#)

Asignar automáticamente IP pública: Usar configuración de subred (habilitar)

Grupo de ubicación: ☐ Agregue la instancia a un grupo de ubicación.

Reserva de capacidad: [Abrir](#)

Directorio de unión al dominio: Ningún directorio [Crear nuevo directorio](#)

Rol de IAM: Ninguno [Crear un nuevo rol de IAM](#)

Comportamiento de cierre: **Detener**

Detener: comportamiento de hibernación ☐ Habilitar la hibernación como un comportamiento de cierre adicional

Habilitar la protección de terminación ☐ Protegerse contra la terminación accidental

Monitorización: ☐ Habilitar monitorización detallada de CloudWatch [Se aplican cargos adicionales.](#)

Propiedad: Compartida: ejecutar una instancia de hardware o [Se aplicarán cargos adicionales por la tenencia dedicada.](#)

Elastic Graphics: ☐ Añadir la aceleración de gráficos [Se aplican cargos adicionales.](#)

Especificación de crédito: ☐ Sin límite [Podrían aplicarse cargos adicionales](#)

Detalles avanzados

Enclave: ☐ Habilitar

Metadatos accesibles: [Habilitado](#)

Versión de metadatos: V1 y V2 (token opcional)

Límite de saltos de respuesta de token de metadatos: 1

Datos de usuario: ☒ Como texto ☐ Como archivo ☐ La entrada ya está codificada en base64

(Opcional)

Pasamos a configurar el almacenamiento seleccionamos las opciones como se muestran en la imagen y pasamos a la siguiente pestaña de configuración.

Paso 4: Adición de almacenamiento

Su instancia se lanzará con la siguiente configuración de dispositivo de almacenamiento. Puede asociar volúmenes de EBS y volúmenes del almacén de instancias adicionales a la instancia o editar la configuración del volumen raíz. También puede asociar volúmenes de EBS adicionales después de lanzar una instancia, pero no volúmenes del almacén de instancias. [Obtenga más información](#) acerca de las opciones de almacenamiento de Amazon EC2.

Tipo de volumen	Dispositivo	Snapshot	Tamaño (GiB)	Tipo de volumen	IOPS	Velocidad (MB/s)	Eliminar
Raíz	/dev/sda1	snap-066aef2270a99947a	30	SSD de uso general (gp2)	100/3000	N/D	<input checked="" type="checkbox"/>

Omitimos la sección de configuración de etiquetas, pues no queremos configurar ninguna y hacemos clic en siguiente.

Pasamos a configurar los puertos que abriremos en nuestra máquina virtual, para ello seleccionamos, crear nuevo grupo (**1**), introducimos un nombre para la configuración de los puertos, y opcionalmente, una descripción (**2**), finalmente seleccionamos los puertos que queremos abrir, en este caso los siguientes puertos:

- 3389 para RDP para poder conectarnos a la máquina virtual.
- 8883 y 1883 para mqtt y mqtt respectivamente.
- 443 y 80 para https y http respectivamente.
- 2909 para el intercambio de datos con la aplicación.

Paso 6: Página Configure Security Group

Un grupo de seguridad es un conjunto de reglas del firewall que controlan el tráfico de la instancia. En esta página, puede agregar reglas para permitir que determinado tráfico llegue a la instancia web y permitir que el tráfico de Internet llegue a la instancia, agregue reglas que permitan el acceso sin restricción a los puertos HTTP y HTTPS. Puede crear un nuevo grupo de seguridad o se [información](#) sobre los grupos de seguridad de Amazon EC2.

Asignar un grupo de seguridad: ☒ Crear un nuevo grupo de seguridad **1**
☐ Seleccionar un grupo de seguridad existente

Nombre del grupo de seguridad: **2**
 Descripción:

Tipo	Protocolo	Rango de puertos	Origen	
RDP	TCP	3389	Personaliz...	0.0.0.0/0
Regla TCP pe	TCP	8883	Personaliz...	0.0.0.0/0
Regla TCP pe	TCP	1883	Personaliz...	0.0.0.0/0
HTTP	TCP	80	Personaliz...	0.0.0.0/0

Hacemos clic en siguiente para pasar a revisar la configuración y a continuación en lanzar, para lanzar la instancia.

Se abrirá un menú para crear un par de claves para acceder a la máquina virtual, usaremos una clave ya existente o crearemos una nueva (la clave de datos actual se encuentra en la documentación en la siguiente ruta "**documentacion proyecto\documentacion tecnica\servidor**").

A continuación configuraremos una IP elástica, para evitar que nuestra maquina cambie su DNS, asi como su IP publica cada vez que reiniciemos.

En la barra izquierda vamos al apartado direcciones IP elásticas, dentro del apartado Red y seguridad.

▼ Red y seguridad

Security Groups New

Direcciones IP elásticas New

Grupos de ubicación

Pares de claves

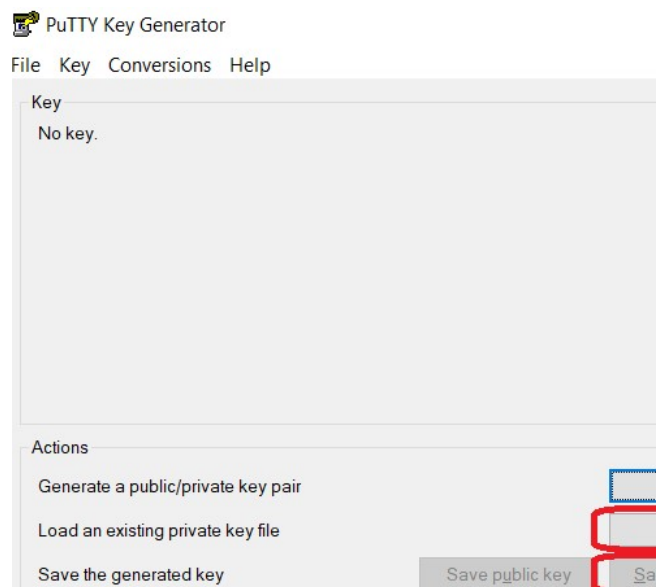
Se abrirá un menú, dentro de él hacemos clic en asignar dirección IP elástica (1), se abrirá otro menú, hacemos clic nuevamente en asignar, esto hara que se cree una dirección IP elástica (3), para asignarla a nuestra maquina virtual hacemos clic en acciones, asociar IP elastica (2).

Se abrirá el menú que se observa a continuación, simplemente indicamos que queremos asociar la ip a una instancia (1), indicamos la instancia (2) y la dirección privada de la instancia a la que le queremos asignar la ip (3), por comodidad indicamos que podemos volver a asociar esta ip a otras instancias (4) finalmente hacemos clic en asociar (5).

Finalmente volvemos al apartado de instancias y lanzamos la maquina.

2.2.-Conexión a la maquina virtual.

Descargar "putty" e instalarlo, primero se generan un par de claves para poder conectarse a la maquina virtual por ssh (ya que el par de claves que nos proporciona amazon tiene una extensión que putty no maneja), para ello escribimos puttygen en la barra de búsqueda de Windows, abrimos el programa, y seleccionamos load. Buscamos el par de claves anteriormente generado en amazon, las seleccionamos, hacemos clic en save private key y las guardamos (Estan guardadas con el nombre de **aws iot machine key** en la ruta "**documentacion proyecto\documentacion tecnica\servidor**").



Para conectarnos a la máquina de AWS escribimos en el campo hostName el DNS publico de la maquina o bien su ip publica.

Después, en las opciones a la izquierda extendemos la opción SSH, hacemos clic en Auth, buscamos las claves que hemos generado antes, las seleccionamos y hacemos clic en open.

Se abrirá una consola de comandos con la cual trabajaremos.



2.3.-Instalación del bróker MQTT

Actualizamos la maquina con los comandos

```
sudo apt update  
sudo apt install -y mosquitto mosquitto-clients
```

Instalamos el Broker

```
sudo apt install -y mosquitto mosquitto-clients
```

Hacemos que mosquitto se inicie automáticamente cuando se inicie el sistema

```
sudo systemctl enable mosquitto
```

Para iniciar mosquitto usamos el siguiente comando

```
sudo systemctl start mosquitto.
```

Para parar mosquitto usamos el siguiente comando

```
sudo systemctl stop mosquitto.
```

Para suscribirnos a un topic usamos el siguiente comando.

```
Mosquitto_sub -h ipBroker -t "tópica/topicA1 -v
```


Para publicar en un topic

```
Mosquitto_pub -h ipBroker -t "tópica/topicA1 -m Datos
```

2.4.- Base de datos

La información que nos llegue la almacenaremos en una base de datos, esta constara de cuatro tablas, una para los datos de los usuarios, otra de los datos de los dispositivos, otra de los datos de constantes vitales, y una para las anomalías que se puedan producir.

La relación entre las tablas queda definida del siguiente modo

usuarios		
	ID_Usuario	int unsigned
	ID_Dispositivo	varchar(12)
	Nombre	varchar(50)
	Apellidos	varchar(50)
	Edad	tinyint

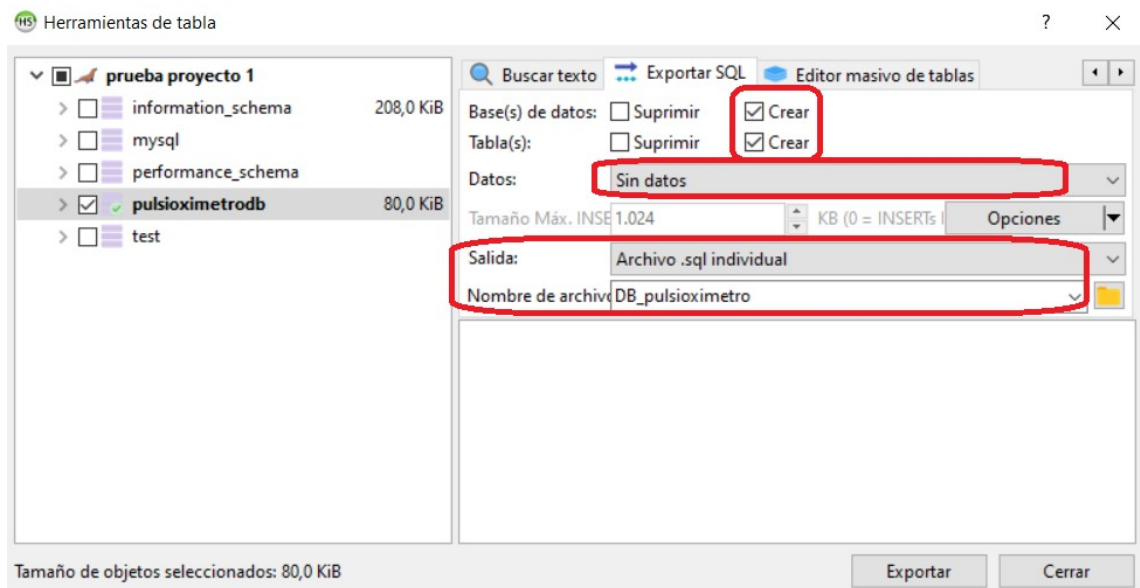
pulso		
	ID_Dispositivo	varchar(12)
	Pulso	float unsigned
	O2	tinyint
	Tiempo	datetime

anomalias		
-----------	--	--

baterias		
----------	--	--

Para crear la base de datos usaremos HeidiSQL, definiremos las tablas como se ve arriba y haciendo botón derecho sobre la base de datos, elegimos exportar como código sql y marcamos los campos crear base de datos y crear tabla, y elegimos como formato de exportación ".sql", finalmente ponemos un nombre al archivo y exportamos.

La base de datos se encuentra en la ruta
"proyecto\documentacion tecnica\servidor\"



A continuación subiremos el script para la generación de la base de datos a la maquina virtual, para ello usando el programa WinSCP (si estamos trabajando desde Windows) tal y como se indica en el apartado 2.8.

2.5.-Guardado de datos recibidos desde MQTT en la base de Datos

Para el almacenaje de datos se realizara un programa en python que recibirá mensajes de tipo JSON enviados por el bróker de la red local al bróker en aws y los guardara en una base de datos, con los criterios especificados en la documentación funcional, el programa suscriptor se encuentra en la ruta "**documentacion proyecto\documentacion tecnica\programas\MqttSuscriber_DbInserter**" con el nombre **MqttSuscriber_DbInserter.py**

Para que el script funcione necesitaremos tener los siguientes programas instalados en la propia maquina

- MariaDB nuestro gestor de base de datos.
- Python3: para poder ejecutar el propio script
- Pip: para poder instalar paquetes de Python3

Ademas, necesitaremos instalar las siguientes librerías, las cuales son requeridas para el funcionamiento del scrip con el gestor de paquetes de python

- paho: para poder interactuar con un bróker por mqtt
- mariadb: para poder interactuar con una base de datos de MariaDB

Para ello ejecutamos los siguientes comandos respectivamente

```
Sudo apt install mariadb-server
Sudo apt install python3
Sudo apt-get install libmariadbclient -dev
Sudo pip3 install mariadb
Sudo pip3 install paho-mqtt
```

Procedemos a configurar mariadb:

Hacemos una instalación segura

```
sudo /usr/bin/mysql_secure_installation
```

Introducimos la contraseña para el usuario root, en este caso **"root"**

Realizamos la instalación del siguiente modo:

- Eliminar usuarios anónimos: SI
- Deshabilitar el login como root de forma remota: SI
- Eliminar la tabla TEST: SI
- Recargar la tabla de privilegios: SI

A continuación creamos la base de datos con el fichero que hemos traído a la maquina.

Para ello primero entramos en el Shell de mariaDB

```
Sudo mysql -u root -p <root>
```

Creamos una base de datos

```
CREATE DATABASE pulsioximetro_db;
```

Salimos del Shell de mariaDB e importamos la base de datos que hemos traído a la base de datos recién creada.

```
Sudo mysql -u username -p pulsioximetro_db < DB_pulsioximetro.sql
```

Volvemos a entrar en el Shell y seleccionamos la base de datos

```
USE pulsioximetro_db;
```

Mostramos las tablas para comprobar que estas están

```
SHOW TABLES;
```

Dado que en la maquina solo nos deja acceder con el usuario root, si usamos a la vez el comando sudo (cosa que no podemos hacer en el script) crearemos un nuevo usuario, que tenga todos los privilegios y el cual usara el script.

Creamos el usuario

```
CREATE USER 'admin'@localhost IDENTIFIED BY 'admin';
```

Le damos al usuario todos los permisos

```
GRANT ALL PRIVILEGES ON *.* TO 'user1'@localhost IDENTIFIED BY 'password1';
```

El funcionamiento del programa se encuentra descrito en el propio código, pero el funcionamiento básico es el siguiente.

- Lee el fichero config.txt ubicado en el mismo directorio y asigna a cada elemento de configuración una variable.

- Con las variables del fichero de configuración se establece una conexión a la base de datos y al bróker
- Se realiza la suscripción a todos los topics
- Cuando llega un mensaje se llama a un handler que lee la información recibida y la decodifica de JSON a una lista de python.
- Inserta en una tabla u otra dependiendo del tipo de información recibida.

```
[DB]
user= root
password= root
host= 127.0.0.1
port= 3306
database= pulsioximetrodb
```

```
[BROKER]
broker_address = 192.168.250.59
borker_port = 1883
topic = #
```

2.6.-Comunicación App-Servidor

Para enviar la información a la aplicación final del cliente se va a establecer una comunicación con el dispositivo en el que este ejecutando la aplicación final.

El funcionamiento de este programa esta explicado en el propio código, pero aquí hay una explicación básica de su funcionamiento.

- Leemos un archivo de configuración, en el que se indica por un lado el puerto por el que se va a establecer el socket de comunicación y el tamaño de la cabecera del mensaje, y por otro los parámetros de conexión a la base de datos local.
- Enlazamos con el dispositivo y esperamos a que nos envíe un mensaje (este contendrá el identificador del dispositivo del que se quiere conocer la información)

- Se hace una consulta en la que se extrae el último dato de la base de datos del dispositivo solicitado.
- Se envía la información por el canal establecido.
- Se cierra el socket de comunicación

Para en funcionamiento del programa hay que instalar las siguientes librerías.


Para poder interactuar con la base de datos (se usa en este caso mysql en lugar de mariadb porque este paquete da problemas an algunas funciones necesarias para el funcionamiento del programa)

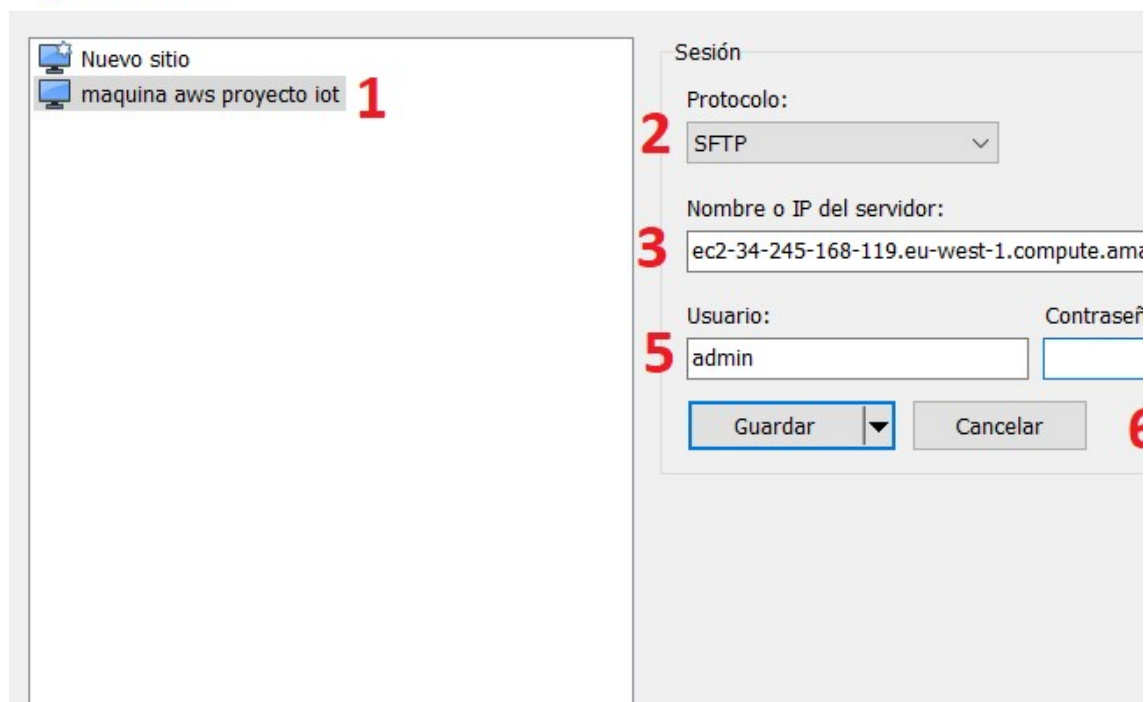
Pip3 install mysqlclient

Pip3 install mysql-connector-python-rf

2.7.-Envio de archivos por SCP mediante WinSCP

Abrimos el programa y le damos un nombre a la conexión (*1) seleccionamos el protocolo SFTP (*2). En el apartado nombre, escribimos el dns o la ip del servidor, seleccionamos el puerto 22 (el ordenador al que queremos acceder tiene que tener habilitada la conexión por ssh) y en el campo usuario escribimos admin, por ultimo clicamos en avanzado.

 Iniciar sesión



The image shows the WinSCP 'Iniciar sesión' (Login) dialog box. It has a left pane for site management and a right pane for session configuration. Red numbers 1 through 5 are overlaid on the interface to indicate the steps for creating a new session:

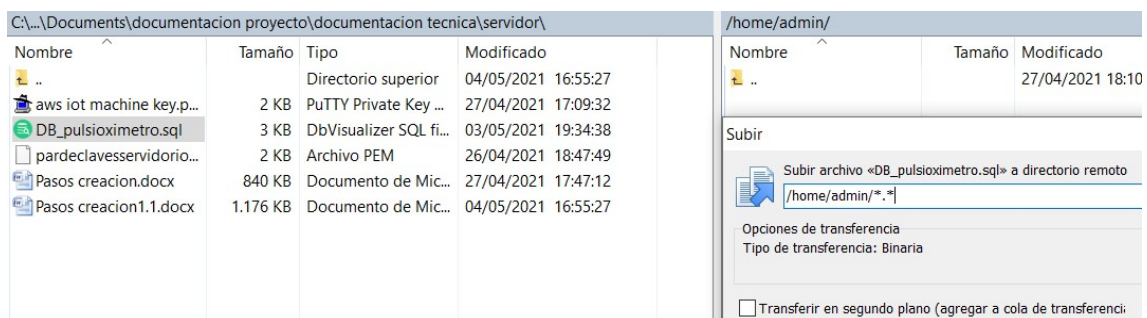
- 1**: Points to the 'maquina aws proyecto iot' entry in the 'Nuevo sitio' list.
- 2**: Points to the 'Protocolo:' dropdown menu, which is set to 'SFTP'.
- 3**: Points to the 'Nombre o IP del servidor:' text field, containing the address 'ec2-34-245-168-119.eu-west-1.compute.amazonaws.com'.
- 5**: Points to the 'Usuario:' text field, containing the username 'admin'.

Other visible elements include a 'Contraseña:' field, a 'Guardar' button, and a 'Cancelar' button.

Dentro del menú vamos al apartado autenticación y buscamos la clave privada de nuestra maquina, hacemos clic en aceptar.

Configuración avanzada de sitio

Se nos abrirá una consola con dos secciones, a la izquierda nuestra maquina local y a la derecha la maquina virtual, arrastramos el archivo que queramos copiar, al soltar no pedirá que escribamos la ruta en la que guardaremos el archivo.



3.-Visualizacion de datos

Para visualizar los datos se va a realizar un programa en python que solicitara al servidor la información del dispositivo del cual desea los datos de pulso, luego representara esta información en una grafica de forma dinámica.

Para el funcionamiento de este programa además de python y pip es necesario instalar los siguientes paquetes.

Kivy, que nos permite hacer interfaces graficas en python

```
python -m pip install kivy[full]
```

Garden, que es un framework para integrar graficas en kivy la aplicación móvil

```
sudo pip install kivy-garden
```

Paquete para trasladar datos a una grafica

```
garden install --upgrade graph
```

Aunque el funcionamiento del código se encuentra descrito en el propio código aquí se hace una explicación genérica del funcionamiento de este.

- Leemos el fichero de configuración, en este establecemos la dirección ip del servidor así como el puerto de comunicación y el tamaño del header del mensaje.
- Leemos la información introducida por el usuario (nombre de usuario del paciente del que se desea información).
- Recibimos la información de pulso saturación y tiempo del dispositivo asociado a ese paciente
- Colocamos en respectivas listas el valor de pulso y tiempo enviados por el servidor.
- Dibujamos una grafica mostrando en el eje "y" los valores de pulso y saturación de oxígeno y en el eje "x" el valor de tiempo en el cual fueron capturados los datos.

4.-Anexos

- Guía configuración bridge:

[Mosquitto MQTT Bridge-Usage and Configuration \(steves-internet-guide.com\)](https://steves-internet-guide.com/Mosquitto-MQTT-Bridge-Usage-and-Configuration)

- Explicación del protocolo GATT

[The Practical Guide to Hacking Bluetooth Low Energy \(attify.com\)](https://attify.com/The-Practical-Guide-to-Hacking-Bluetooth-Low-Energy)

-Servicios y características BLE de la sensortilebox de st (user manual del siguiente enlace)

www.st.com/en/embedded-software/bluest-sdk

