DOCUMENTACIÓN MQTTS

Primero como root vamos a la carpeta de certificados de mosquitto y nos descargamos un codigo bash a traves de gihub

```
root@Stilgar:/etc/mosquitto/certs# wget https://raw.githubusercontent.com/owntra
cks/tools/master/TLS/generate-CA.sh
--2020-03-02 15:32:16--  https://raw.githubusercontent.com/owntracks/tools/maste
r/TLS/generate-CA.sh
Resolviendo raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.132
.133
Conectando con raw.githubusercontent.com (raw.githubusercontent.com)[151.101.132
.133]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 8723 (8,5K) [text/plain]
Grabando a: “generate-CA.sh”

generate-CA.sh      100%[===================>]   8,52K  --.-KB/s    in 0s

2020-03-02 15:32:21 (44,5 MB/s) - “generate-CA.sh” guardado [8723/8723]
```

le damos permisos de ejecución y lo ejecutamos

```
root@Stilgar:/etc/mosquitto/certs# HOSTNAME="stilgar.local" ./generate-CA.sh sti
lgar.local
Generating a RSA private key
.....................................................................................
..+++++
.........................+++++
writing new private key to './ca.key'
-----
Created CA certificate in ./ca.crt
subject=
    commonName                = An MQTT broker
    organizationName          = OwnTracks.org
    organizationalUnitName    = generate-CA
    emailAddress              = nobody@example.net
Warning: the CA key is not encrypted; store it safely!
--- Creating server key and signing request
Generating RSA private key, 2048 bit long modulus
...................+++++
..................+++++
e is 65537 (0x010001)
--- Creating and signing server certificate
Signature ok
subject=CN = stilgar.local, O = OwnTracks.org, OU = generate-CA, emailAddress =
nobody@example.net
```

Ahora abrimos el archivo de configuración de mosquitto y añadimos estas lineas

```
# MQTT over TLS/SSL
listener 8883
cafile /etc/mosquitto/certs/ca.crt
certfile /etc/mosquitto/certs/stilgar.local.crt
keyfile /etc/mosquitto/certs/stilgar.local.key
```

Por ultimo reiniciamos el mosquitto.

Ahora crearemos usuarios y contraseñas, primero creamos un fichero donde introduciremos un usuario y contraseña (usuari → contrasenya) y lo encriptamos

```
stilgar@Stilgar:/etc/mosquitto/certs$ sudo nano passwordfile
[sudo] password for stilgar:
stilgar@Stilgar:/etc/mosquitto/certs$ ls
ca.crt  certToArduino_01.py  README            stilgar.local.key
ca.key  generate-CA.sh       stilgar.local.crt
ca.srl  passwordfile         stilgar.local.csr
stilgar@Stilgar:/etc/mosquitto/certs$ sudo mosquitto_passwd -U passwordfile
stilgar@Stilgar:/etc/mosquitto/certs$ cat passwordfile
stilgar:$6$rYF8bYZ18iJ4CZdZ$jpkr09AdRR7ZUNLCjqZZAd1PbVYicT1gSOyVOrKf3jkoWthSgp8
TDFh36qfKxSoFs3bsdwmKNr+O/eMOND3H4w==
usuario:$6$uaHC/4M26vq+yP2m$ARuIjbRqgiC5ENWHeOUFXNsmTUNxH8gbJDA2b0GuH4/kM7N8Myz
tB5VETWpW9JaZWqfXRDvker5wSSDLS1Qj6w==
stilgar@Stilgar:/etc/mosquitto/certs$
```

Añades en el mosquitto.conf :

allow_anonymous false
password_file /etc/mosquitto/certs/passwordfiles

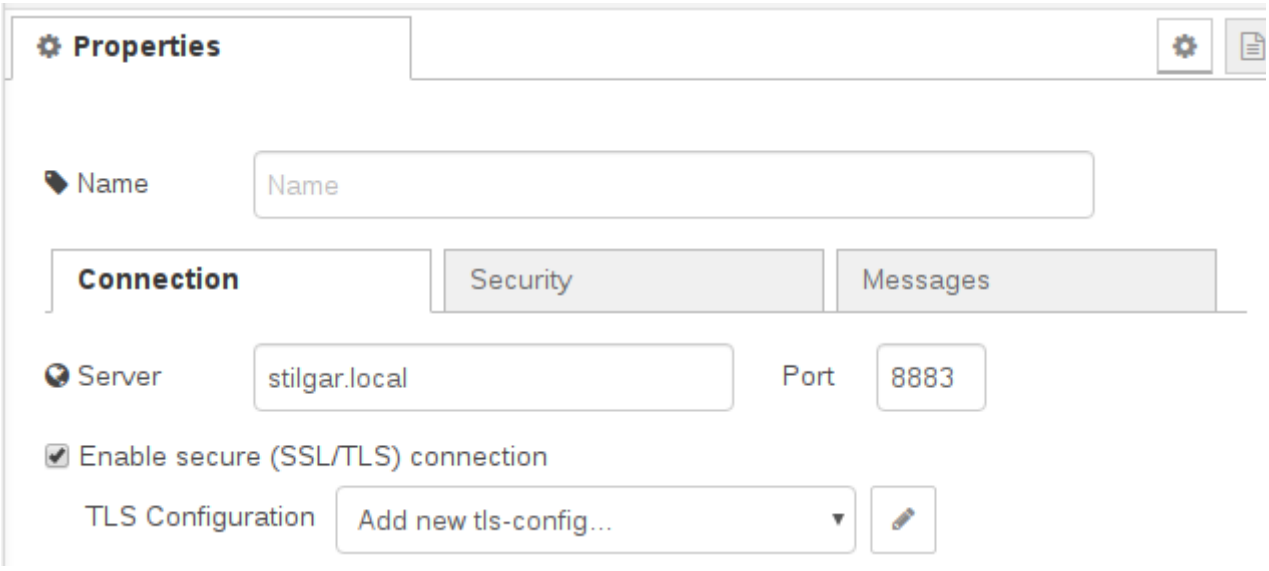
Y reinicias mosquitto

```
tilgar@Stilgar:/etc/mosquitto$ sudo nano mosquitto.conf
tilgar@Stilgar:/etc/mosquitto$ sudo systemctl mosquitto restart
nknown operation mosquitto.
tilgar@Stilgar:/etc/mosquitto$ sudo systemctl restart mosquitto
tilgar@Stilgar:/etc/mosquitto$
```

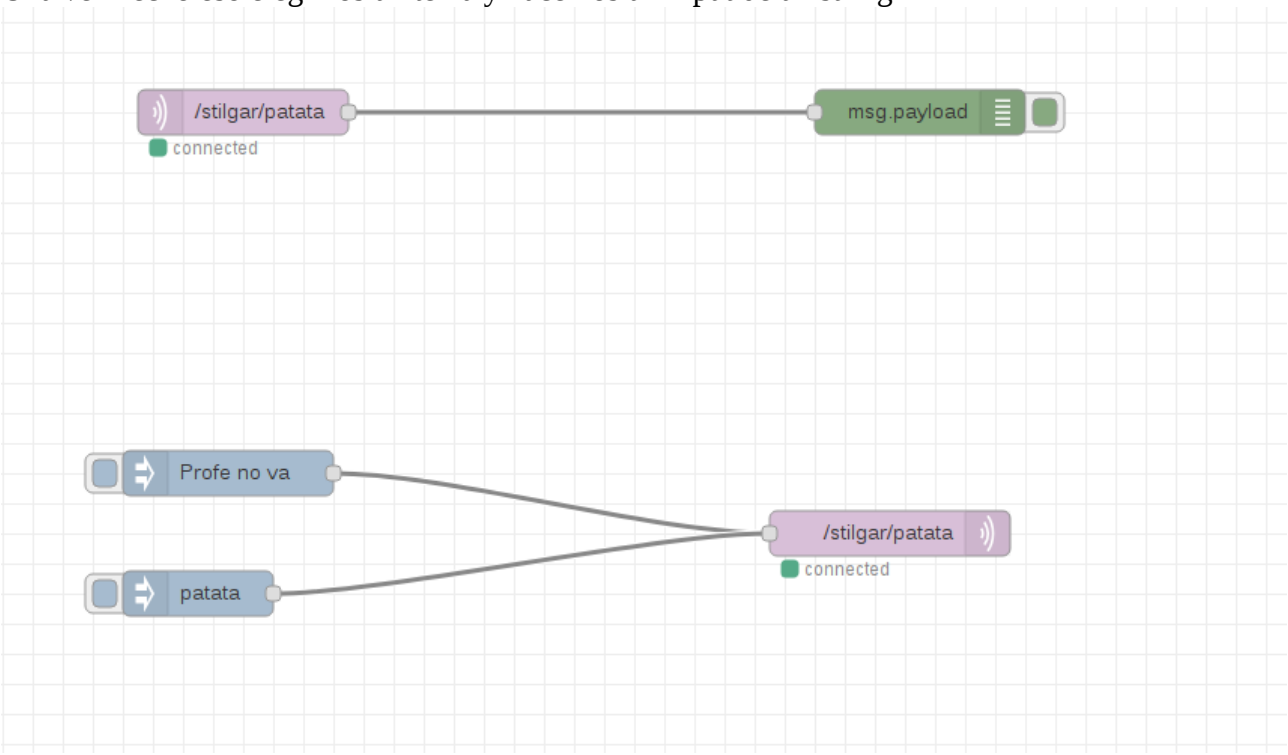
En la carpeta /etc/mosquitto/certs descargamos el código python y lo ejecutamos, su salida nos dará el codigo de certificado debidamente formatado y lo enganchamos en el archivo de arduino ESP32_mqtts_PubSub_00

```
stilgar@Stilgar:/etc/mosquitto/certs$ sudo python3 certToArduino_01.py
[sudo] password for stilgar:
"-----BEGIN CERTIFICATE-----\n" \
"MIIDqjCCApKgAwIBAgIJANxzJSXHmudgMA0GCSqGSIb3DQEBDQUAMGoxFzAVBgNV\n" \
"BAMMDkFuIE1RVFQgYnJva2VyMRYwFAYDVQQKDA1Pd25UcmFja3Mub3JnMRQwEgYD\n" \
"VQQLDAtnZW5lcmF0ZS1DQTEhMB8GCSqGSIb3DQEJARYSbm9ib2R5QGV4YW1wbGUu\n" \
"bmV0MB4XDTIwMDMwMjE0MzM0NFoXDTMyMDIyODE0MzM0NFowajEXMBUGA1UEAwwO\n" \
"QW4gTVFUVCBicm9rZXIxFjAUBgNVBAoMDU93blRyYWNrcy5vcmcxFDASBgNVBAsM\n" \
"C2dlbmVyYXRlLUNBMSEwHwYJKoZIhvcNAQkBFhJub2JvZHlAZXhhbXBsZS5uZXQw\n" \
"ggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCaB6mX9X1XSBljtYMWV3kk\n" \
"qpOJ6cxu4/dkiYmTDM4JXv1wVSuNHwRcc6rv2kjG/AuZbq4+MbmuPU40PveR5Fg8\n" \
"UCZErlizEPWVol3vhc+b4GjkR2GY7HJ9wj3dlSIjTl1nWBoaiDcqoB31kDyzMyjQ\n" \
"0FgywJxc8m+9FQpHKYa0toUCgvHysEjrcagQJRtK1d4kws3aHkIY0VBLLk59MNR8\n" \
"UxijYRAqUp7gqt/5S+yhck9thNrppIzzy6aHjLV62DazV4zdBT6+q94ElU83WSYC\n" \
"tChrjGYL4wHLGHeFoTkatidgl9ylMxIC6El110SUEERUmEEa9o1egP6WhZMPqlNb\n" \
"AgMBAAGjUzBRMB0GA1UdDgQWBBRsfjvjhVPMtqJuInCOXg2Im25NsjAfBgNVHSME\n" \
"GDAWgBRsfjvjhVPMtqJuInCOXg2Im25NsjAPBgNVHRMBAf8EBTADAQH/MA0GCSqG\n" \
"SIb3DQEBDQUAA4IBAQAB/tcIbG0/B0wg+ReHwTC3ZLSYggrXKziShUdy0iGza+VA\n" \
"eT3fX9Kbo1pXFs01F7r0HvIJ5VZziwjrCPmSY3s/5wJl7zG2OMPUbazTHGoA2Uw5\n" \
"9tsuzcIIoIjapVuxlVT6EELeNOgkdCpmf9RRWf6FTOnUtJoCKY6KHQaRdvZxqldx\n" \
"mw+fMa9gb31cL5mhNFsPZ1gUSA6ck/gvcC7feW6z9z+h6RqYdsdnAbawyHKxC73V\n" \
"zxFzOi7qa2gbocz3l70Hnig2fRnMMDpAuuQgHK/3HgOYDaF8GlQzAFg2n/nV0m1r\n" \
"cjqhBojHJ99nOMCuZ/RFErxgHs0BXT6CuJP1SJcn\n" \
"-----END CERTIFICATE-----\n";
```

Ahora vamos al node red, configuramos un nuevo borker en mqtt y le introducimos el nuevo que acabamos de crear



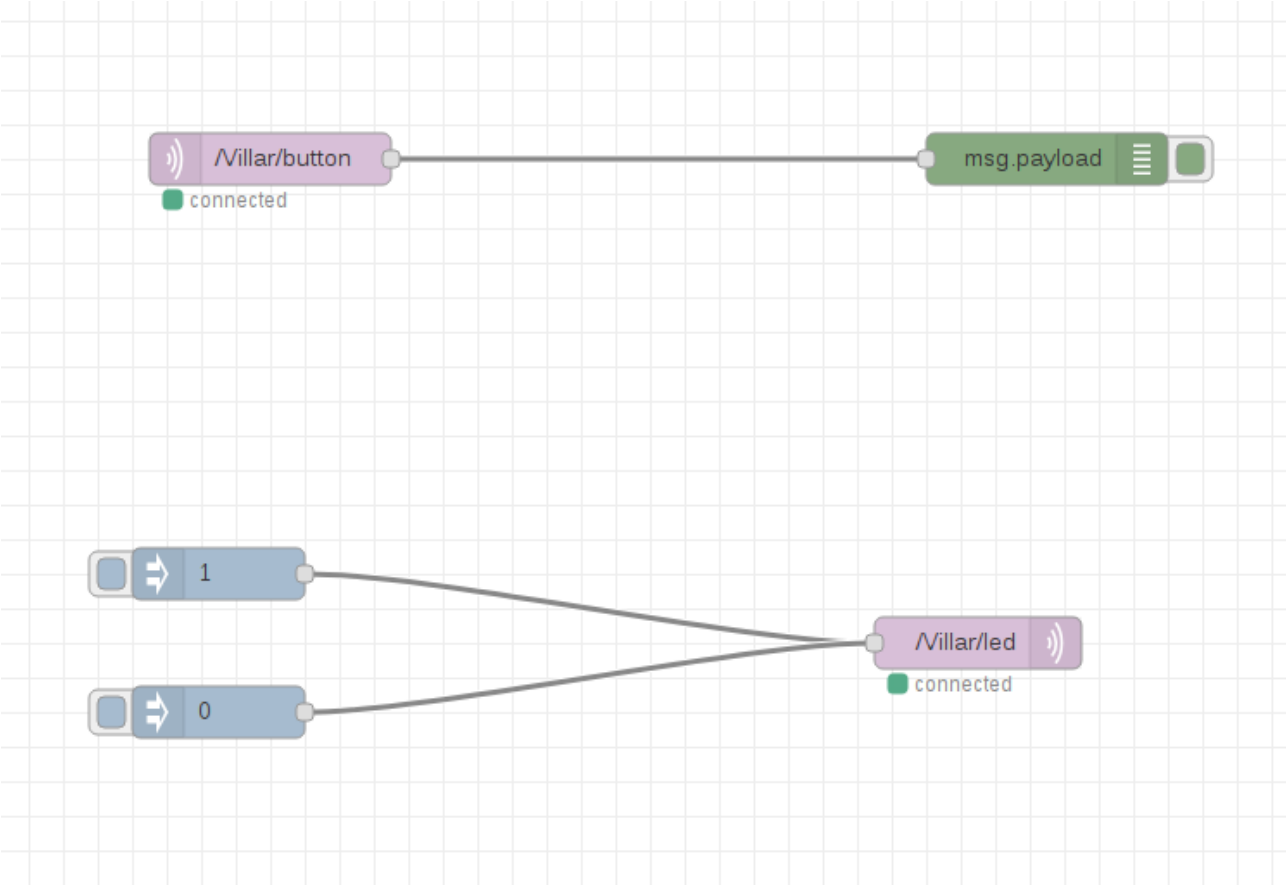
Una vez hecho eso elegimos un tema y hacemos un input de un string



Y el output:

Una vez probado eso en arduino cambiamos la ID de la maquina, el wifi, el usuario, la contraseña y los temas para que no hayan conflictos.

Y Retocamos el node red


/Villar/button
connected
msg.payload
1
0
/Villar/led
connected