

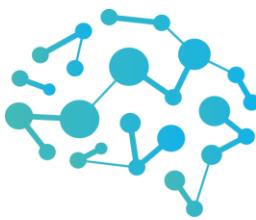


GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®



LANITED  
LABORATORIO NACIONAL DE INVESTIGACIÓN EN TECNOLOGÍAS DIGITALES

Programas Nacionales Estratégicos de Ciencia, Tecnología y Vinculación con los  
Sectores Social, Público y Privado

## E4. Reporte técnico del desarrollo del dispositivo IoT Edge

### 4.1 Como crear un servidor IoT con MQTT y el bróker EMQX

Institución

Centro de Ingeniería y Desarrollo Industrial (CIDESI)

Responsable Técnico

Nombre: Leonardo Barriga Rodríguez

Cargo: Profesor Investigador

CVU 271835

Responsable del Entregable

Nombre: Mateo Álvarez Ruiz

Cargo: Ingeniero de Proyecto

Equipo de trabajo

Mateo Álvarez Ruiz

Israel Uribe Hernández

Aldair Isaac Mercado García

Número de Convenio:

Pp F003 6/V E/2022, 000000000321074,

Laboratorio Nacional de Investigación en Tecnologías Digitales  
(LANITED) 2023-2024



<b>1 UN PLANETA MAS INTELIGENTE</b>	<b>4</b>
<b>2 TELEMETRIA E INTERNET</b>	<b>4</b>
<b>3 Sección 2: CONEXIÓN MQTT [1]</b>	<b>5</b>
3.1 PUBLICADOR	6
3.2 SUSCRIPCION	6
3.3 TOPICOS y COMODINES	7
3.3.1 Como crear un TOPICO	8
3.4 CALIDAD DE SERVICIOS	9
3.4.1 QUE SERVICIO UTILIZAR	10
<b>4 Sección 3: EMQX V4 TODO SOBRE EL BROKER [2]</b>	<b>11</b>
4.1 INSTALACION DEL BROKER EMQX	11
4.2 CONEXIÓN POR WEBSOCKETS	18
4.3 CLIENTE EMQTTX	21
4.4 CONEXIÓN CON CLIENTE MQTTX	24
4.5 INSTALAR OPENSSL	27
4.6 GENERAR CERTIFICADOS AUTOFIRMADOS	32
4.7 CONEXIÓN SEGURA HTTPS EN EL DASHBOARD	41
4.8 AUTENTIFICACION MySQL con Laragon	47
4.9 LISTAS DE CONTROL DE ACCESO (ACL)	59
4.10 PHP MyAdmin EN LARAGON	62
4.11 INSTALACION EN LINUX	64
4.12 CONEXIÓN POR WEBSOCKETS	76
<b>5 Sección 6: EMQX MODO DE PRODUCCION (NUBE DE ORACLE)</b>	<b>76</b>
5.1 ¿Qué es VPS?	76
5.2 Producción VPS de Oracle	76
5.3 Iniciar sesión con Putty en la instancia	83
5.4 Instalar Hestia Panel [14]	89
5.5 Registro de un dominio	98
5.6 Certificados SSL al Dominio	101
5.7 Instalar EMQX en el VPS	105

<b>5.8 Agregar certificados SSL al bróker EMQX</b>	<b>127</b>
<b>5.9 Certificados SSL al Dashboard EMQX</b>	<b>130</b>
<b>6 SECCION 7: CLIENTE ESP32</b>	<b>142</b>
<b>6.1 MQTTY ESP32 por conexión segura</b>	<b>151</b>
<b>7 SECCION 9: CLIENTE PHP</b>	<b>172</b>
<b>7.1 Cliente PHP con ESP32 con envío de temperatura por MQTT</b>	<b>176</b>
<b>7.2 Almacenar datos de Temperatura en BD desde MQTT</b>	<b>183</b>
<b>7.3 Registro de un dominio</b>	<b>Error! Bookmark not defined.</b>
<b>7.4 Registro de un dominio</b>	<b>192</b>
<b>7.5 Registro de un dominio</b>	<b>192</b>
<b>8 Referencias</b>	<b>193</b>



## 1 UN PLANETA MAS INTELIGENTE

El concepto de IBM Smarter Planet se basa en un conjunto de pilares llamado las 3i que son:

**Instrumental:** La información se captura dondequiera que exista como por ejemplo mediante el uso de sensores remotos

**Interconectado:** La información se mueve desde el punto de recopilación a cualquier lugar donde se pueda consumir de manera útil

**Inteligente:** La información se procesa, analiza y actúa para obtener el máximo valor y conocimiento



## 2 TELEMETRIA E INTERNET

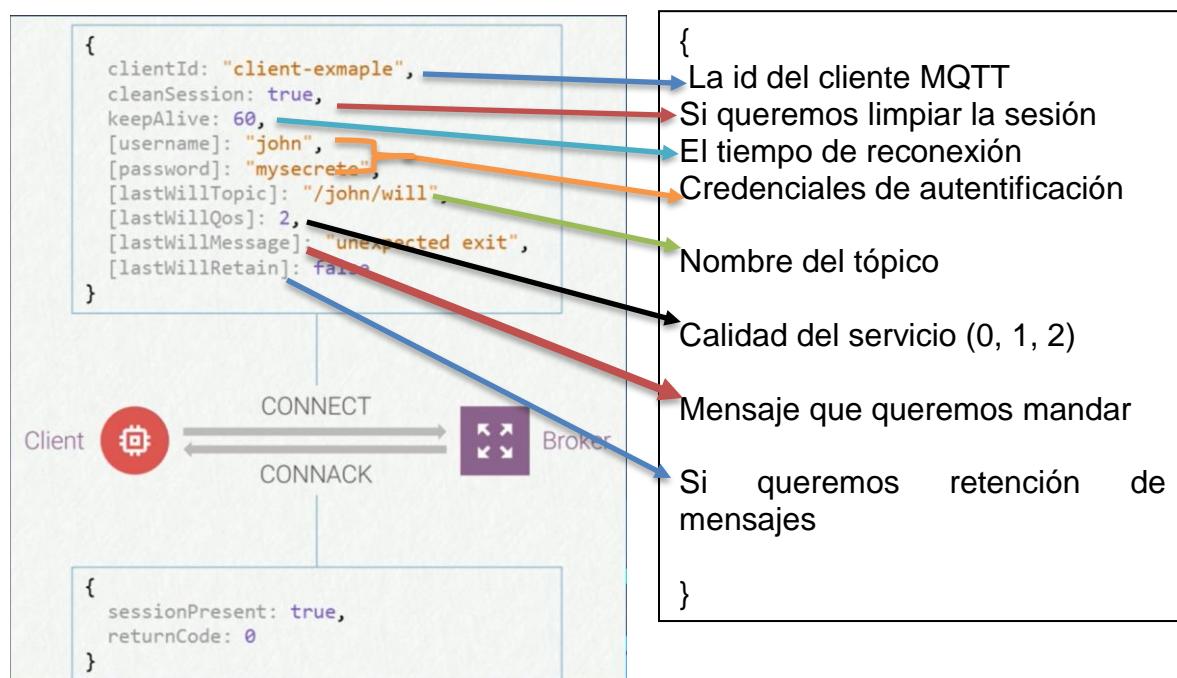
Las tecnologías de telemetría permite medir o monitorear cosas a distancia, también las nuevas tecnologías de telemetría permiten interconectar la medición y monitorear dispositivos en diferentes ubicaciones y reducir el costo de crear publicaciones que puedan ejecutarse en dispositivos inteligentes





### 3 Sección 2: CONEXIÓN MQTT [1]

Tanto el cliente como el bróker tienen que tener la pila TCP/IP por ello los clientes no se pueden conectar directamente a los clientes así que se tienen que realizar mediante el bróker



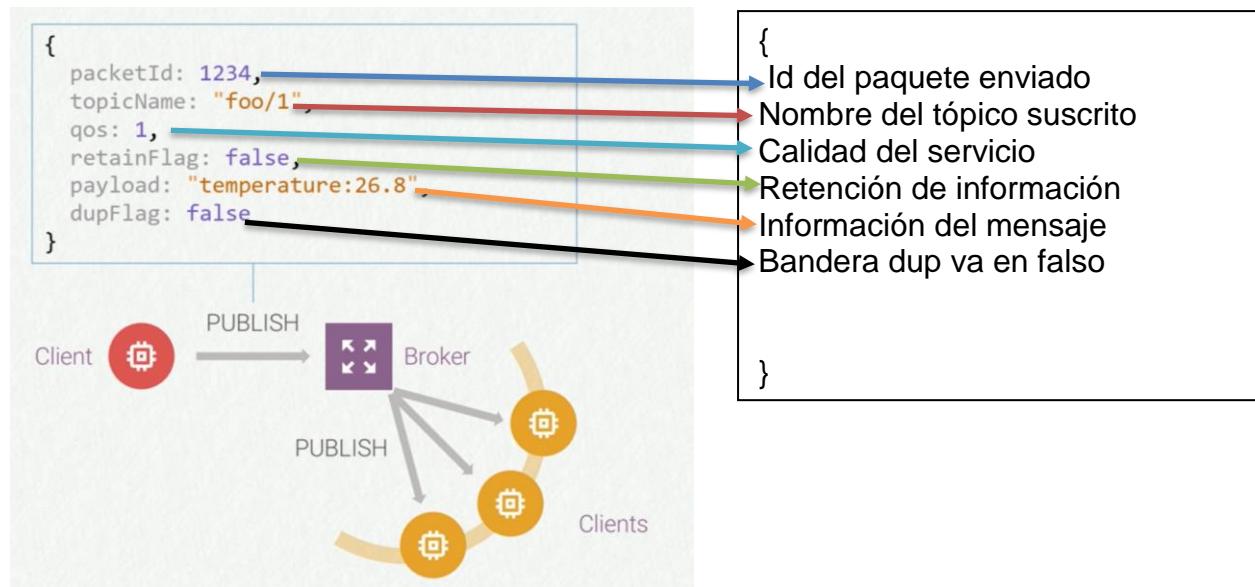
#### Return CODE

Return Code	Return Code Response
0	Connection Accepted
1	Connection refused, unacceptable protocol version
2	Connection refused, identifier rejected
3	Connection refused, server unavailable
4	Connection refused, bad username or password
5	Connection refused, not authorized



### 3.1 PUBLICADOR

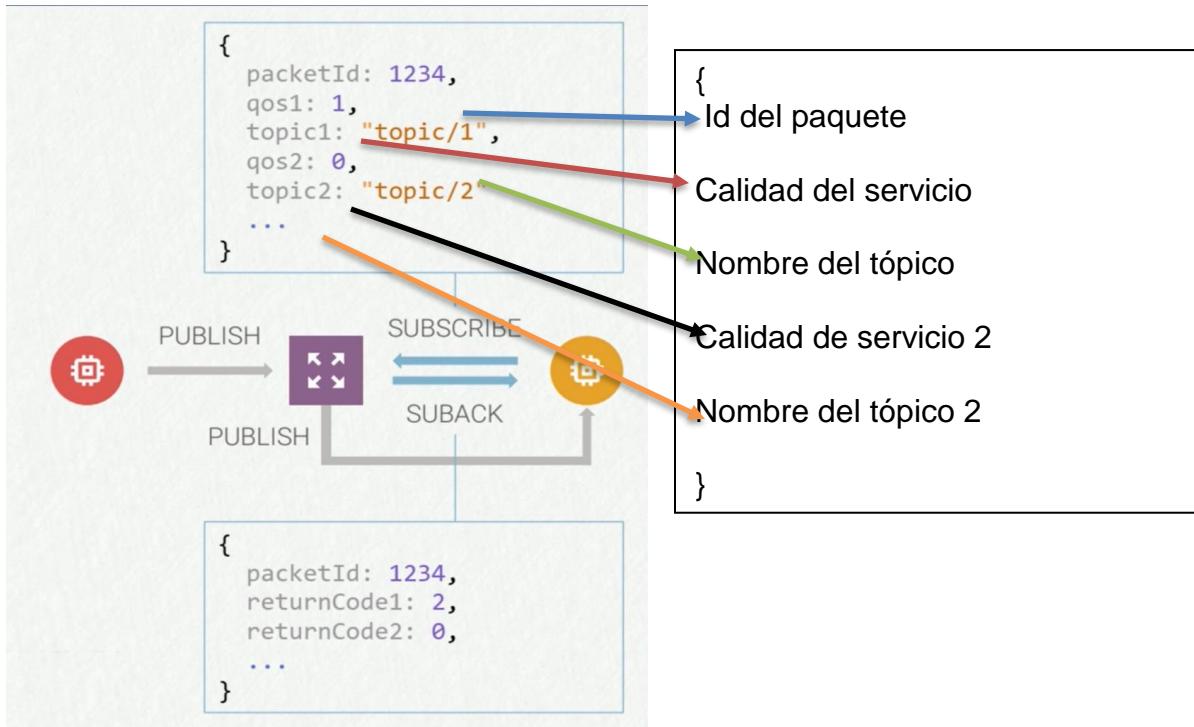
El cliente envía mensajes al tópico + su carga útil (Información de temperatura, etc.) por ello cada mensaje tiene un tópico asignado



### 3.2 SUSCRIPCION

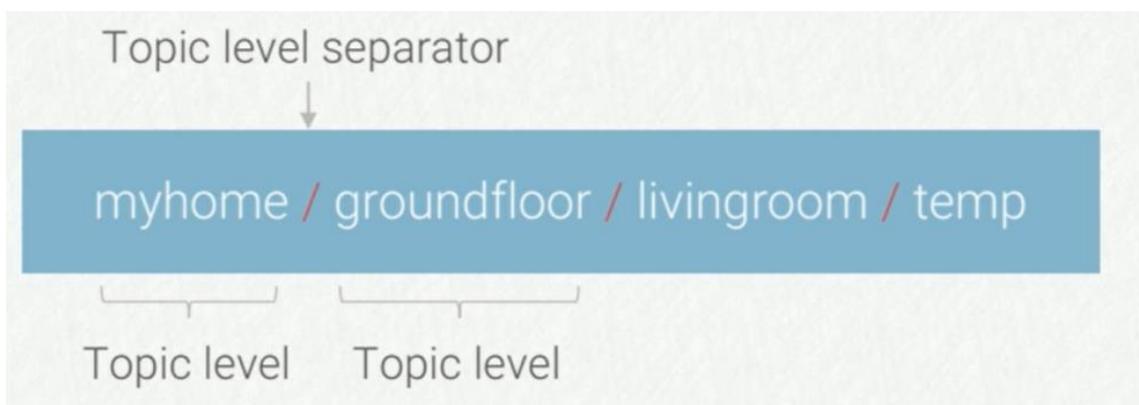
El cliente se suscribe a los tópicos del Bróker para recibir información por ello cada suscripción consta de pares de Topic-QoS y en caso de tener tópicos duplicados los QoS altos son los principales y por ultimo envía un mensaje SUBACK para reportar la confirmación de suscripción

Return Code	Return Code Response
0	Success – Maximum QoS 0
1	Success – Maximum QoS 1
2	Success – Maximum QoS 2
128	Failure



### 3.3 TOPICOS y COMODINES

Un tópico es una cadena de texto y sirve para que los clientes puedan suscribirse y publicar información usando tópicos .Los tópicos se pueden planificar en varios niveles y requieren al menos un carácter y las mayúsculas y minúsculas no son lo mismo por ejemplo Mytopic que mytopic





Los comodines son caracteres universales que nos permiten usar patrones para leer tópicos por ello con los comodines podemos suscribirnos a muchos tópicos a la vez pero solo se pueden utilizar en suscripciones no en publicaciones y son (+) de nivel único y (#) de multinivel el cual solo va al final del tópico



### 3.3.1 Como crear un TOPICO

1. No comience con (/) por ejemplo /mytopic ya que se convertirá en //mytopic
2. No utilice espacios en blanco porque UTF-8 ya contiene muchos
3. Utiliza solo caracteres ASCII
4. Mantenlo corto y entendible
5. Utilice tópicos de ejemplo (id/idcliente) para identificar aún más el origen del mensaje
6. Utilice temas específicos como temperatura, humedad, luminosidad por ejemplo en lugar de **home/room/valores** sería mejor poner **home/room/temperatura**
7. No se suscriba a # ya que el cliente tendrá problemas con el rendimiento

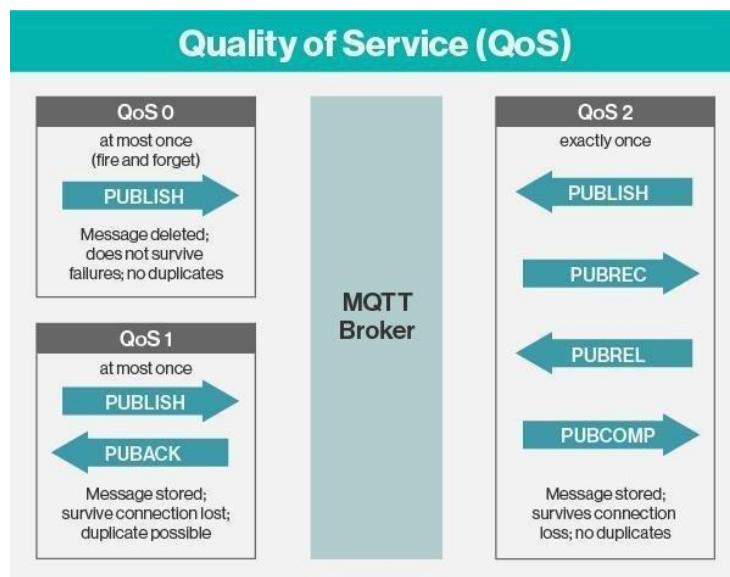


### 3.4 CALIDAD DE SERVICIOS

**Como máximo una vez QoS=0:** Es la modalidad de transferencia más rápida. Se denomina a veces transmitir y olvidar ya que el mensaje no se almacena. El mensaje puede perderse si se desconecta el cliente o si falla el servidor pero MQTT no requiere que los servidores reenvíen publicaciones en QoS=0 a un cliente.

**Al menos una vez QoS=1:** Es el valor predeterminado la modalidad de transferencia Si el emisor no recibe un acuse de recibo, el mensaje se envía de nuevo con el distintivo DUP establecido hasta que se reciba un acuse de recibo. Como resultado, un mismo mensaje se puede enviar varias veces al receptor y ser procesado varias veces además si el receptor es un intermediario, el mensaje se publica a todos sus suscriptores o si el receptor es un cliente se publicara a todos sus clientes

**Exactamente una vez QoS=2:** El mensaje se entrega siempre exactamente una vez además es la modalidad de transferencia más segura, pero la más lenta. Deben realizarse como mínimo dos pares de transmisiones entre el emisor y el receptor antes de que el mensaje pueda suprimirse de la parte del emisor además el receptor puede procesar el mensaje proporcionado en la primera o segunda fase, no tiene que volver a procesarlo y si el receptor es un intermediario, publica el mensaje a los suscriptores, si el receptor es un cliente, el mensaje se entrega a la aplicación de suscriptor y el receptor devuelve un mensaje de finalización al emisor para comunicarle que ha terminado de procesar el mensaje.



Entre mayor sea el nivel de QoS menor será el rendimiento



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

### 3.4.1 QUE SERVICIO UTILIZAR

Digamos que necesitamos mandar N número de mensajes y podemos decir que el tiempo necesario para mandar un mensaje publish en QoS0 es PT por otro lado en QoS1 la respuesta del servidor al cliente es de 2 bytes por lo que la respuesta en publish se llama MT o sea su tiempo de respuesta es PT+MT y por último el QoS2 al tener fluyendo PUBREC, PUBREL y PUBCOM este tomara más tiempo por lo que se determinara por PT+3MT

Por ejemplo Si necesitamos mandar 10 mensajes para ser transferido del cliente al servidor entonces sabemos que

$$N=10$$

$$PT=1 \text{ segundo}$$

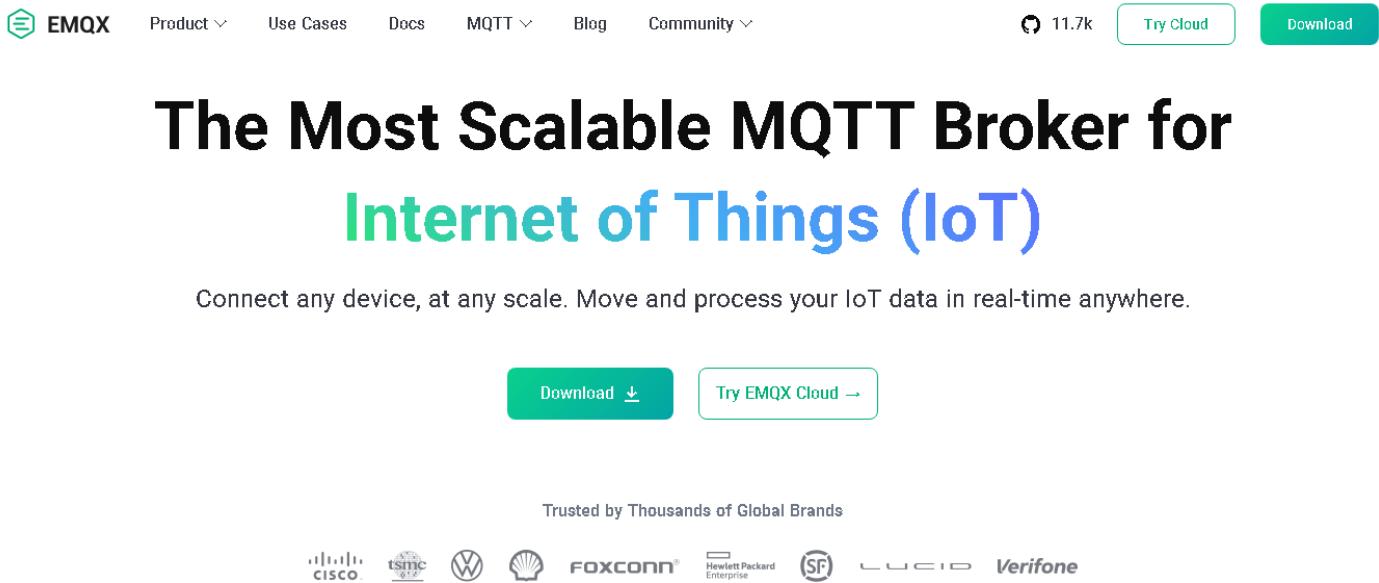
$$MT = 0.4 \text{ segundos}$$

QoS 0	QoS 1	QoS 2
$N \cdot PT$ $10 \cdot 1$	$N(PT+MT)$ $10(1+0.4)$	$N(PT+3MT)$ $10(1+1.2)$
Tarda 10s	Tarda 14s	Tarda 22s

## 4 Sección 3: EMQX V4 TODO SOBRE EL BROKER [2]

### 4.1 INSTALACION DEL BROKER EMQX

Lo primero que vamos a hacer es ir a la página de EMQX.io y una vez ahí buscaremos la plataforma que deseemos utilizar en este caso será Windows 10 y descargaremos el software 0



The screenshot shows the official website for EMQX. At the top, there's a navigation bar with links for Product, Use Cases, Docs, MQTT, Blog, and Community. To the right of the navigation is a user icon showing 11.7k and two buttons: 'Try Cloud' and 'Download'. The main heading is 'The Most Scalable MQTT Broker for Internet of Things (IoT)' in large, bold text. Below it, a sub-headline reads 'Connect any device, at any scale. Move and process your IoT data in real-time anywhere.' Two prominent buttons are centered below the headline: 'Download' with a downward arrow and 'Try EMQX Cloud' with a right-pointing arrow. At the bottom, it says 'Trusted by Thousands of Global Brands' followed by logos for Cisco, TSMC, Volkswagen, Shell, Foxconn, Hewlett Packard Enterprise, Lucid, and Verifone.

**Nota:** En este momento empezaremos con la versión 4 por lo que es importante descargar esta y no la 5

## Download EMQX

[Docker](#)[Ubuntu](#)[Debian](#)[CentOS/RHEL](#)[Amazon Linux 2](#)[macOS](#)[Windows](#)[Kubernetes](#)**Version**

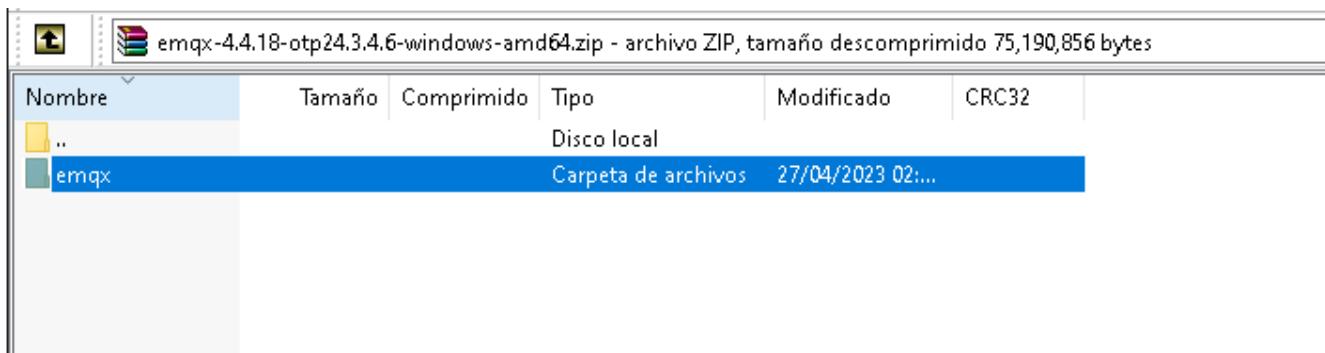
v5.0.26

[Release Notes](#) | [All Versions](#)1 Download [emqx-5.0.26-windows-amd64.zip](#), then decompress it

2 Access the decompression path from the command line, start EMQX

`./emqx/bin/emqx start`[Get Started](#) | [Server Estimate](#) NEW

Una vez descargado nos arrojara un Zip el cual adentro tiene una capeta llamada EMQX la cual debemos descomprimir y copiar al disco C o directorio raíz





Disco local (C:)

Nombre	Fecha de modifica...	Tipo	Tamaño
Archivos de programa	22/05/2023 04:57 ...	Carpeta de archivos	
Archivos de programa (x86)	23/03/2023 06:43 ...	Carpeta de archivos	
<b>data</b>	21/02/2023 12:45 a...	Carpeta de archivos	
Desarrollo	25/02/2022 01:59 a...	Carpeta de archivos	
Descargas	24/11/2022 05:18 ...	Carpeta de archivos	
emqx	27/04/2023 02:36 a...	Carpeta de archivos	
Intel	08/11/2018 04:10 ...	Carpeta de archivos	
laragon	20/06/2022 03:18 ...	Carpeta de archivos	
maquina virtual	14/06/2021 07:53 ...	Carpeta de archivos	
Mono	24/06/2020 12:59 a...	Carpeta de archivos	
My Web Sites	31/03/2020 04:06 ...	Carpeta de archivos	
PerfLogs	07/12/2019 03:14 a...	Carpeta de archivos	
Program Files (x86)	01/06/2020 05:39 ...	Carpeta de archivos	
Riot Games	26/08/2020 10:59 ...	Carpeta de archivos	
System	24/06/2020 12:59 a...	Carpeta de archivos	
System64	24/06/2020 01:01 a...	Carpeta de archivos	
tmp	20/06/2022 04:45 ...	Carpeta de archivos	
Usuarios	29/03/2021 05:10 ...	Carpeta de archivos	
Windows	22/05/2023 04:20 ...	Carpeta de archivos	
Windows10Upgrade	25/04/2019 03:26 ...	Carpeta de archivos	
xampp	25/02/2022 02:21 a...	Carpeta de archivos	
XboxGames	13/05/2022 03:30 ...	Carpeta de archivos	
HALionOne.dll	24/08/2007 02:24 ...	Extensión de la ap...	15,760 KB
mono	24/06/2020 12:59 a...	Paquete de Windo...	2,064 KB

Disco local (C:)

Nombre	Fecha de modifica...	Tipo	Tamaño
Archivos de programa	22/05/2023 04:57 ...	Carpeta de archivos	
Archivos de programa (x86)	23/03/2023 06:43 ...	Carpeta de archivos	
<b>data</b>	21/02/2023 12:45 a...	Carpeta de archivos	
Desarrollo	25/02/2022 01:59 a...	Carpeta de archivos	
Descargas	24/11/2022 05:18 ...	Carpeta de archivos	
<b>emqx</b>	27/04/2023 02:36 a...	Carpeta de archivos	
Intel	28/11/2018 04:10 ...	Carpeta de archivos	



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Después de tener la carpeta en el directorio correcto procederemos a acceder a ella por lo que tendremos que abrir la consola de comandos CMD

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.2965]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\hp>
```

Una vez abierta accederemos a la ruta de la carpeta EMQX y una vez ahí accederemos al directorio bin

```
Administrator: Símbolo del sistema

Microsoft Windows [Versión 10.0.19045.2965]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>cd..

C:\Windows>cd..

C:\>cd EMQX

C:\emqx>cd bin

C:\emqx\bin>
```



Una vez ahí ingresaremos el comando **emqx console** que es un comando que nos mostrara un log de todo lo que está bien y lo que está fallando con el protocolo así como iniciararlo

```
Administrator: Símbolo del sistema - emqx console
(c) Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>cd..

C:\Windows>cd..

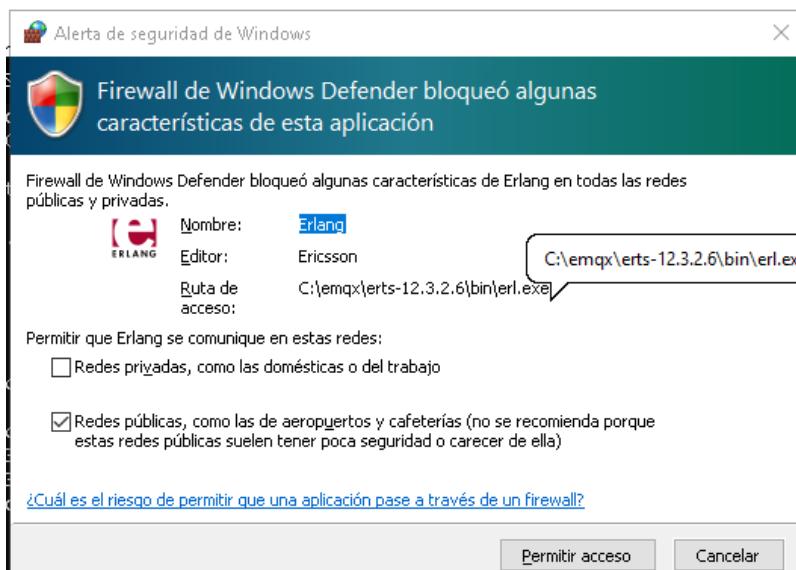
C:\>cd EMQX

C:\emqx>cd bin

C:\emqx\bin>emqx console

C:\emqx>"C:\emqx\erts-12.3.2.6\bin\erl.exe" -boot "C:\emqx\releases\4.4.18\start" -config c:/emqx/data/configs/app.2023.06.05.11.07.30.config -args_file c:/emqx/data/configs/vm.2023.06.05.11.07.30.args -vm_args c:/emqx/data/configs/vm.2023.06.05.11.07.30.args -mnesia dir 'c:/emqx/data/mnesia/emqx@127.0.0.1'
Starting emqx on node emqx@127.0.0.1
Start mqtt:tcp:internal listener on 127.0.0.1:11883 successfully.
Start mqtt:tcp:external listener on 0.0.0.0:1883 successfully.
Start mqtt:ws:external listener on 0.0.0.0:8083 successfully.
Start mqtt:ssl:external listener on 0.0.0.0:8883 successfully.
Start mqtt:wss:external listener on 0.0.0.0:8084 successfully.
Start http:management listener on 8081 successfully.
Start http:dashboard listener on 18083 successfully.
EMQX 4.4.18 is running now!
```

**Nota** es importante darle permisos a la aplicación cuando salte el aviso del firewall





GOBIERNO DE  
MÉXICO



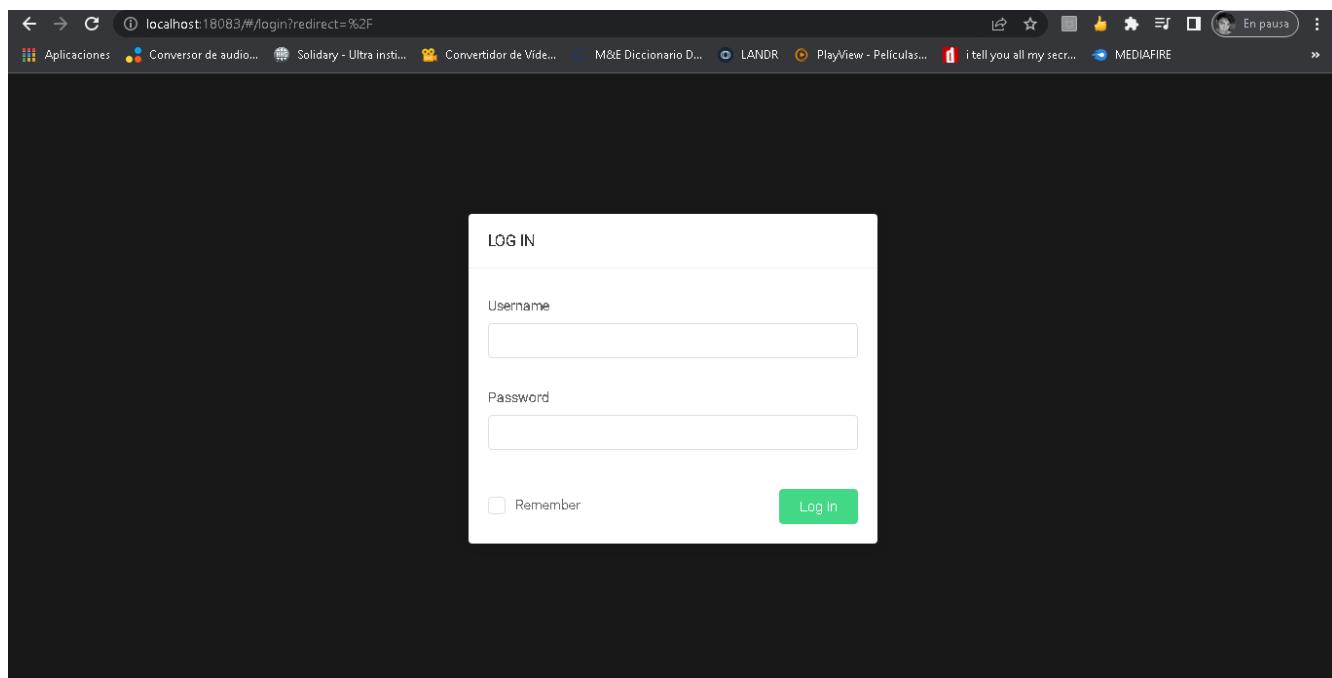
CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez arrancado el protocolo buscaremos en nuestro CMD la línea que diga dashboard y ahí veremos en qué puerto se inició este, en mi caso en el 18083

```
Start http:management listener on 8081 successfully.  
Start http:dashboard listener on 18083 successfully.  
EMQX 4.4.18 is running now!
```

Una vez que ya sabemos en qué puerto se inició el dasboard iremos a nuestro navegador y ahí colocaremos a dirección localhost:18083 y nos saldrá el dashboard principal



**Nota:** En mi caso es 18083 por que ahí se inició mi dashboard en tu caso pondrás el puerto que te haya salido a ti en la consola de comandos



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Iniciaremos sesión con las credenciales que viene por defecto que es:

Usuario **admin**

Contraseña **public**

LOG IN

Username

admin

Password

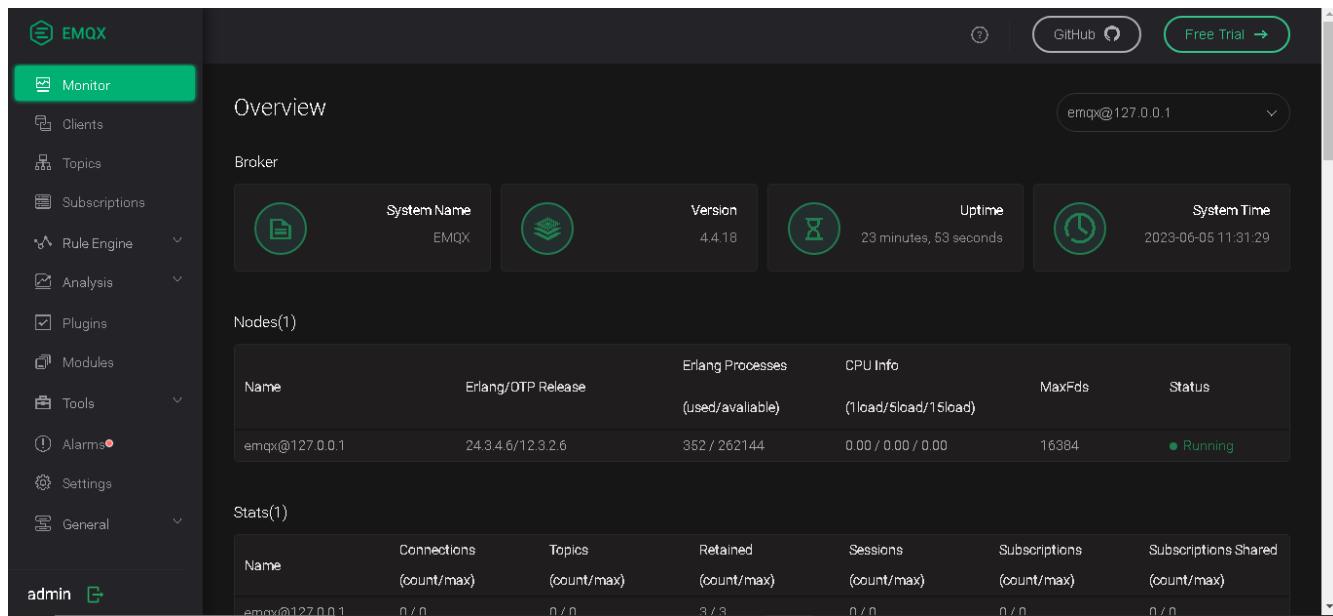
\*\*\*\*\*

Remember

Log In

**NOTA:** Una vez inicies sesión el programa te indicara que iniciaste sesión con las credenciales de default por lo que te pedirá que las cambies por las credenciales que tú deseas y volverás a iniciar sesión con tus nuevas credenciales

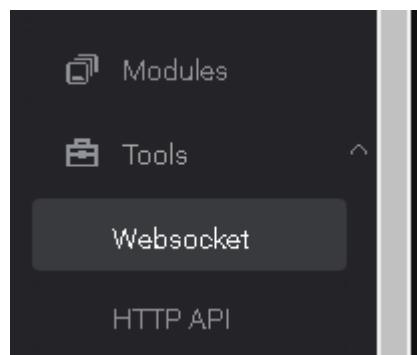
Una vez hayas iniciado sesión te saldrá la siguiente interfaz que significara que hicimos la instalación correctamente de nuestro bróker por lo que ya podremos navegar por la interfaz sin problema y podremos los módulos y apartados que tiene como lo es las alarmas o el apartado de añadir usuarios o quitar usuarios así como los dispositivos conectados y el módulo de análisis y suscripciones para más información a detalle de cada apartado ingrese aquí [4]



The screenshot shows the EMQX Dashboard interface. On the left, there's a sidebar with navigation links: Monitor (selected), Clients, Topics, Subscriptions, Rule Engine, Analysis, Plugins, Modules, Tools, Alarms, Settings, and General. A user 'admin' is logged in. The main area has tabs for Overview, Broker, Nodes(1), and Stats(1). Under Overview, it shows the System Name as EMQX, Version 4.4.18, Uptime 23 minutes, 53 seconds, and System Time 2023-06-05 11:31:29. The Nodes(1) table lists 'emqx@127.0.0.1' with Erlang/OTP Release 24.3.4.6/12.3.2.6, Erlang Processes 352 / 262144, CPU Info 0.00 / 0.00 / 0.00, MaxFds 16384, and Status Running. The Stats(1) table lists 'emqx@127.0.0.1' with Connections 0 / 0, Topics 0 / 0, Retained 3 / 3, Sessions 0 / 0, Subscriptions 0 / 0, and Subscriptions Shared 0 / 0.

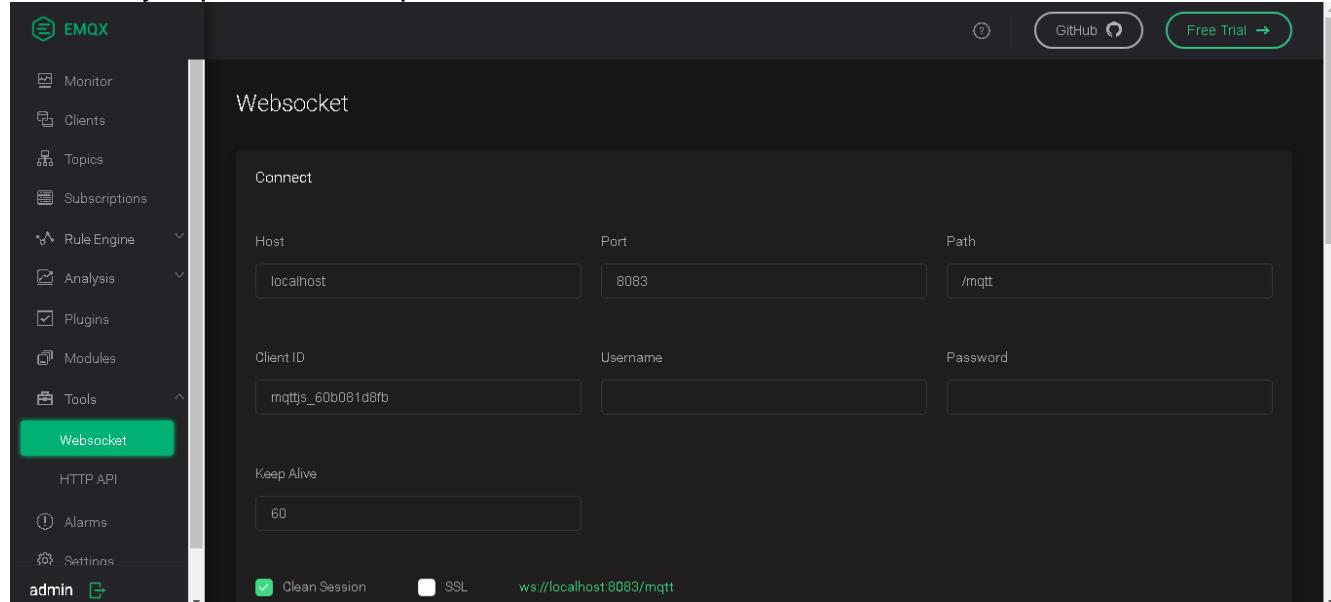
**Nota** la siguiente vez que quiere arrancar el bróker con consola y no quiera ver toda la información de este como con el comando “emqx console” puede usar el comando **emqx start** dentro del directorio bin y se iniciara el bróker directamente

## 4.2 CONEXIÓN POR WEBSOCKETS



Ahora vamos a probar si nuestro bróker funciona bien y nos permite realizar la conexión por websocket para ello nos iremos al apartado de tools y seleccionaremos donde dice websockets

Una vez ahí nos saldrá una ventana como esta y ahí en el apartado de connect estarán los datos de nuestra conexión en mi caso como es prueba los dejare por default y solo le daré en conectar y esperaremos a que el bróker inicie la conexión



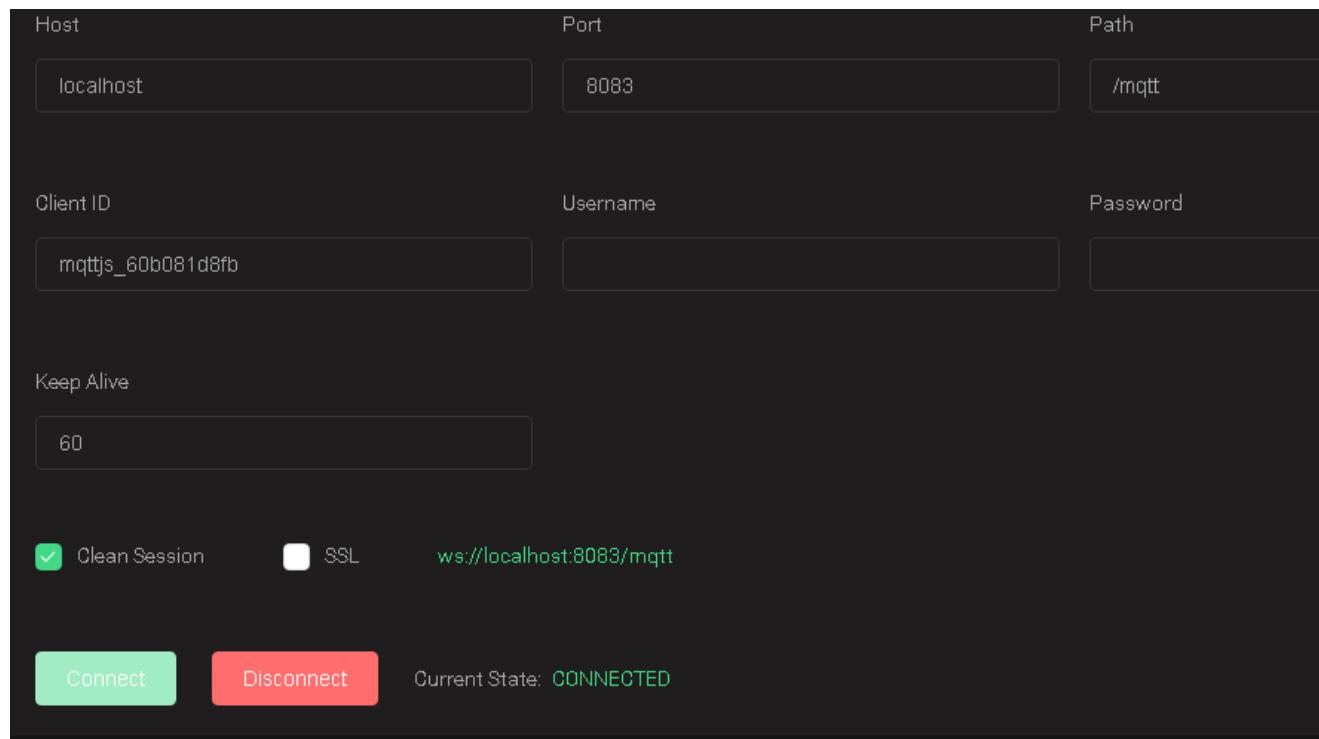
EMQX

- Monitor
- Clients
- Topics
- Subscriptions
- Rule Engine
- Analysis
- Plugins
- Modules
- Tools
- WebSocket**
- HTTP API
- Alarms
- Settings

admin

websocket

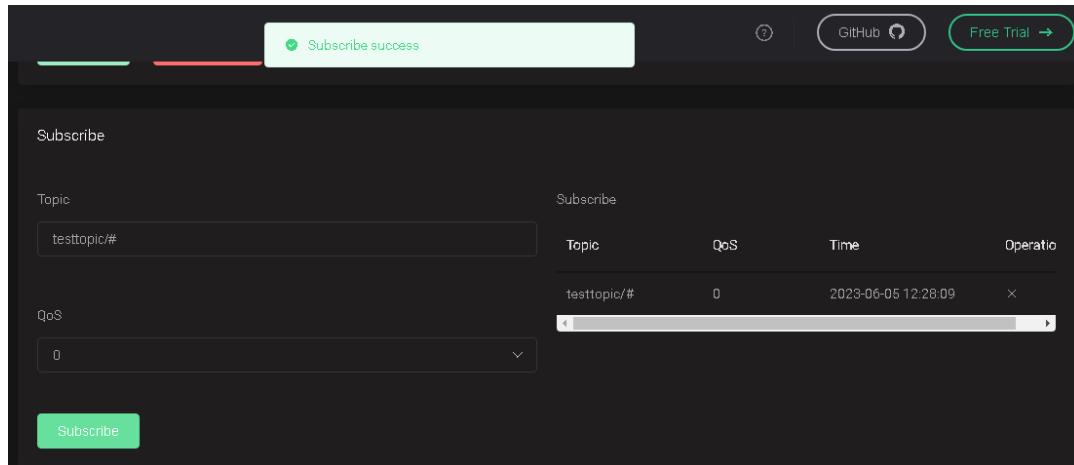
Host: localhost, Port: 8083, Path: /mqtt  
 Client ID: mqttjs\_60b081d8fb, Username: [empty], Password: [empty]  
 Keep Alive: 60  
 Clean Session,  SSL, ws://localhost:8083/mqtt



Host: localhost, Port: 8083, Path: /mqtt  
 Client ID: mqttjs\_60b081d8fb, Username: [empty], Password: [empty]  
 Keep Alive: 60  
 Clean Session,  SSL, ws://localhost:8083/mqtt

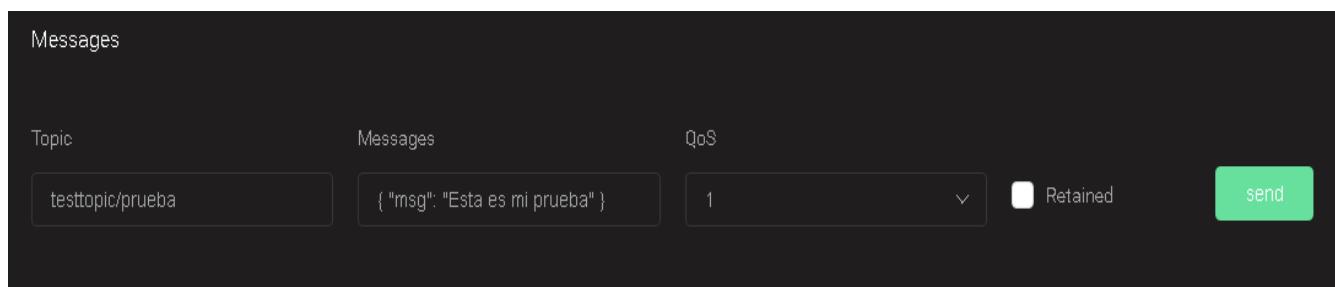
Connect, Disconnect, Current State: CONNECTED

Una vez conectado bajaremos al apartado de suscripciones y ahí un tópico y lo suscribiremos en mi caso será el tópico **testtopic/prueba** y seleccionaremos la calidad del servicio y le daremos en suscribir



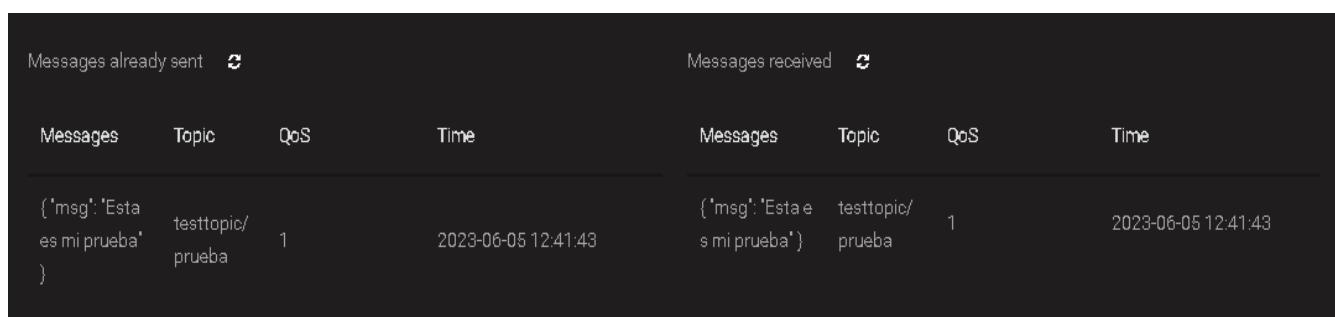
The screenshot shows a dark-themed MQTT client interface. At the top, there's a green progress bar followed by a success message box containing a green checkmark and the text "Subscribe success". To the right of the message are three buttons: a question mark icon, a GitHub icon, and a "Free Trial" button with a right-pointing arrow. Below the message box, the word "Subscribe" is displayed. On the left, there's a "Topic" input field containing "testtopic/#". On the right, there's a table titled "Subscribe" with columns: Topic, QoS, Time, and Operation. A single row is shown with "testtopic/#" in the Topic column, "0" in the QoS column, "2023-06-05 12:28:09" in the Time column, and a delete icon in the Operation column. Underneath the table, there's a "QoS" dropdown menu set to "0". At the bottom left is a green "Subscribe" button.

Una vez tengamos echo la suscripción procederemos a mandar un mensaje a este tópico para ver si está funcionando la conexión por websocket por ello escribiremos el nombre del tópico, el mensaje que queremos mandar y la QoS en que queremos mandar el mensaje



The screenshot shows a dark-themed MQTT client interface. At the top, the word "Messages" is displayed. Below it, there are three input fields: "Topic" containing "testtopic/prueba", "Messages" containing a JSON object {"msg": "Esta es mi prueba"}, and "QoS" set to "1". To the right of these fields are two buttons: a "Retained" checkbox and a green "send" button.

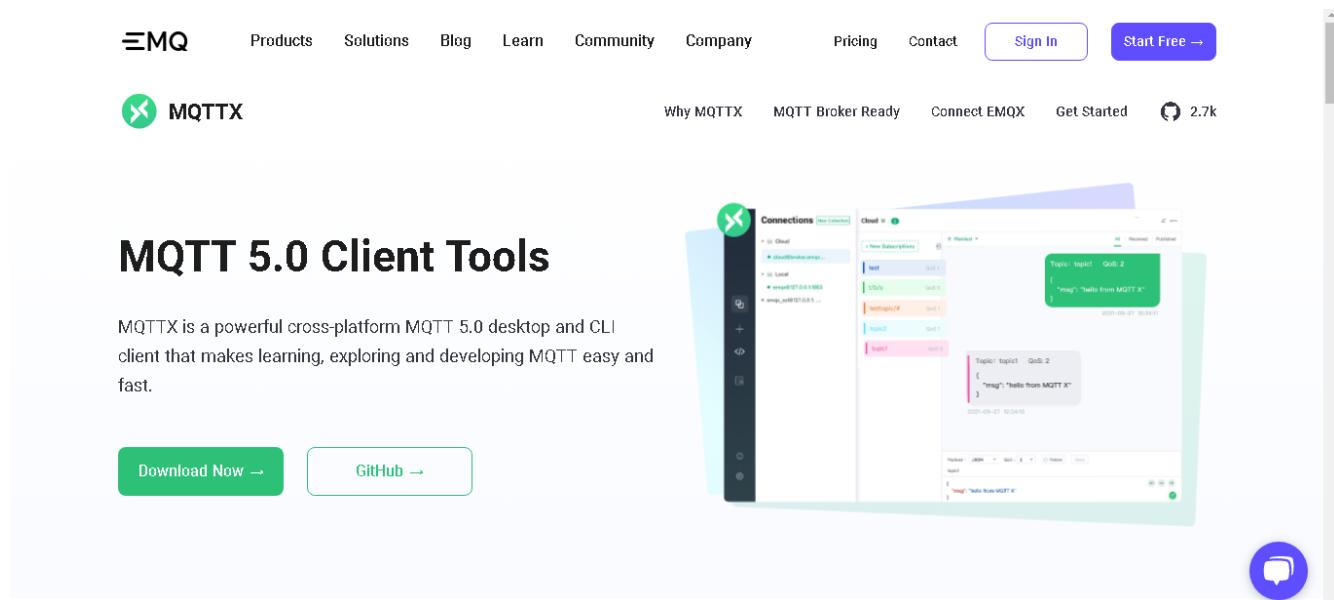
Una vez tengamos listo todos los datos le daremos en send y mandaremos el mensaje a nuestro tópico y se podrá visualizar en una tabla lo que se mandó y si es que llegó de mensajes



The screenshot shows a dark-themed MQTT client interface with two tables. The left table is titled "Messages already sent" and the right table is titled "Messages received". Both tables have columns: Messages, Topic, QoS, and Time. The "Messages already sent" table has one entry: {"msg": "Esta es mi prueba"} in the Topic column, "testtopic/prueba" in the Topic column, "1" in the QoS column, and "2023-06-05 12:41:43" in the Time column. The "Messages received" table also has one entry: {"msg": "Esta es mi prueba"} in the Topic column, "testtopic/prueba" in the Topic column, "1" in the QoS column, and "2023-06-05 12:41:43" in the Time column.

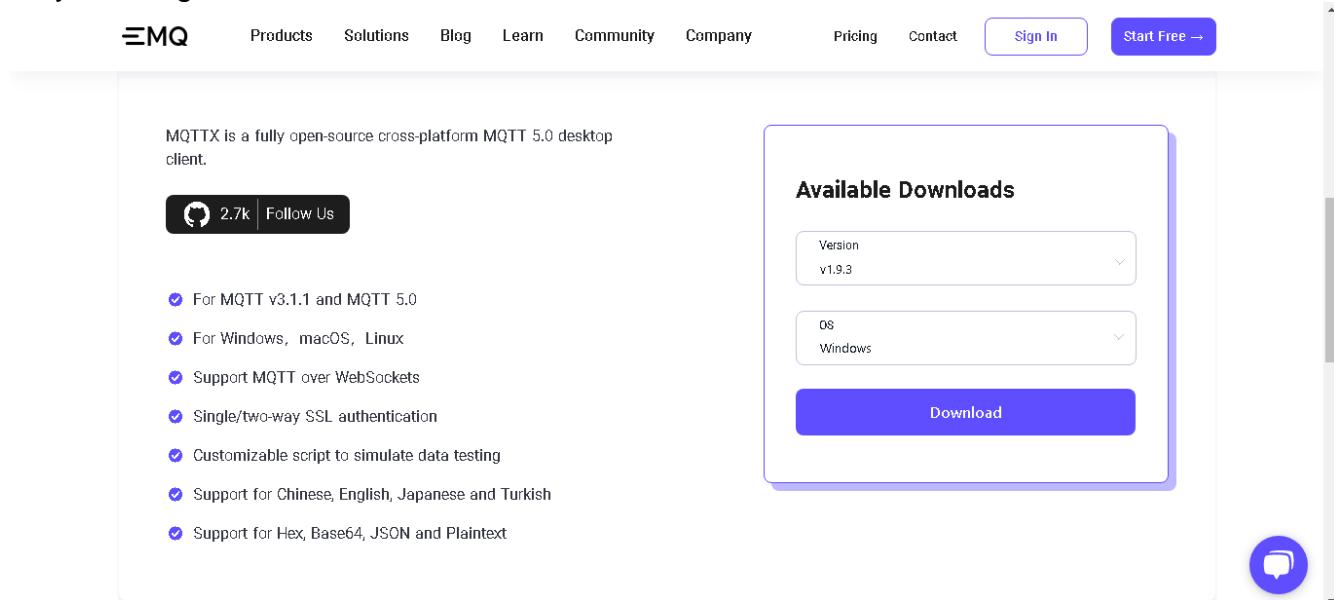
## 4.3 CLIENTE EMQTTX

Una vez tengamos nuestro bróker funcionando correctamente procederemos a descargar nuestro cliente con el que podremos trabajar por lo que iremos a la página de MQTT Client [5]



The screenshot shows the EMQ MQTT Client Tools website. At the top, there's a navigation bar with links for Products, Solutions, Blog, Learn, Community, Company, Pricing, Contact, Sign In, and Start Free →. Below the navigation, there's a search bar with the placeholder "MQTTX". The main content area features a heading "MQTT 5.0 Client Tools" and a brief description: "MQTTX is a powerful cross-platform MQTT 5.0 desktop and CLI client that makes learning, exploring and developing MQTT easy and fast." Below the description are two buttons: "Download Now →" and "GitHub →". To the right of the text, there's a screenshot of the MQTTX application interface, which includes a sidebar with "Connections" and "Cloud" sections, and a central pane showing "Subscriptions" and "Topics". A message in the topics pane reads: "Topic: topic1 QoS: 2 { "msg": "Hello from MQTT X" }". At the bottom right of the page is a blue speech bubble icon.

Una vez ahí seleccionaremos nuestra plataforma donde trabajaremos en este caso Windows 10 y descargaremos la última versión



The screenshot shows the same website as the previous one, but now focusing on the download options. On the left, there's a list of features: "For MQTT v3.1.1 and MQTT 5.0", "For Windows, macOS, Linux", "Support MQTT over WebSockets", "Single/two-way SSL authentication", "Customizable script to simulate data testing", "Support for Chinese, English, Japanese and Turkish", and "Support for Hex, Base64, JSON and Plaintext". On the right, there's a large callout box titled "Available Downloads" with dropdown menus for "Version" (set to v1.9.3) and "OS" (set to Windows). A prominent blue "Download" button is at the bottom of this box. A blue speech bubble icon is located at the bottom right.



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez descargado el cliente procederemos a iniciar los pasos de instalación de programa

Instalación de MQTTX

— □ ×

### Elegir opciones de instalación

¿Para quién se instalará esta aplicación?



Elige si deseas que este software esté disponible para todos los usuarios o solo para ti.

- Cualquiera que utilice este ordenador (todos los usuarios)
- Solo para mí. (hp)

Instalación nueva solo para el usuario actual.

MQTTX 1.9.3

Siguiente >

Cancelar

Instalación de MQTTX

— □ ×

### Instalando

Por favor espere mientras MQTTX se instala.

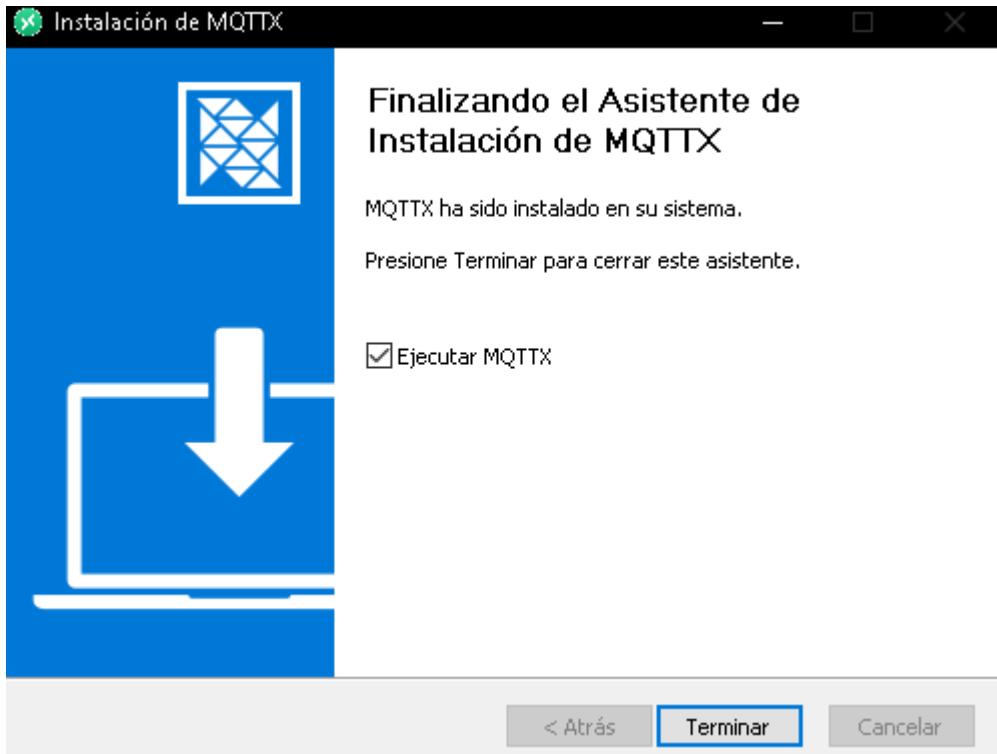


MQTTX 1.9.3

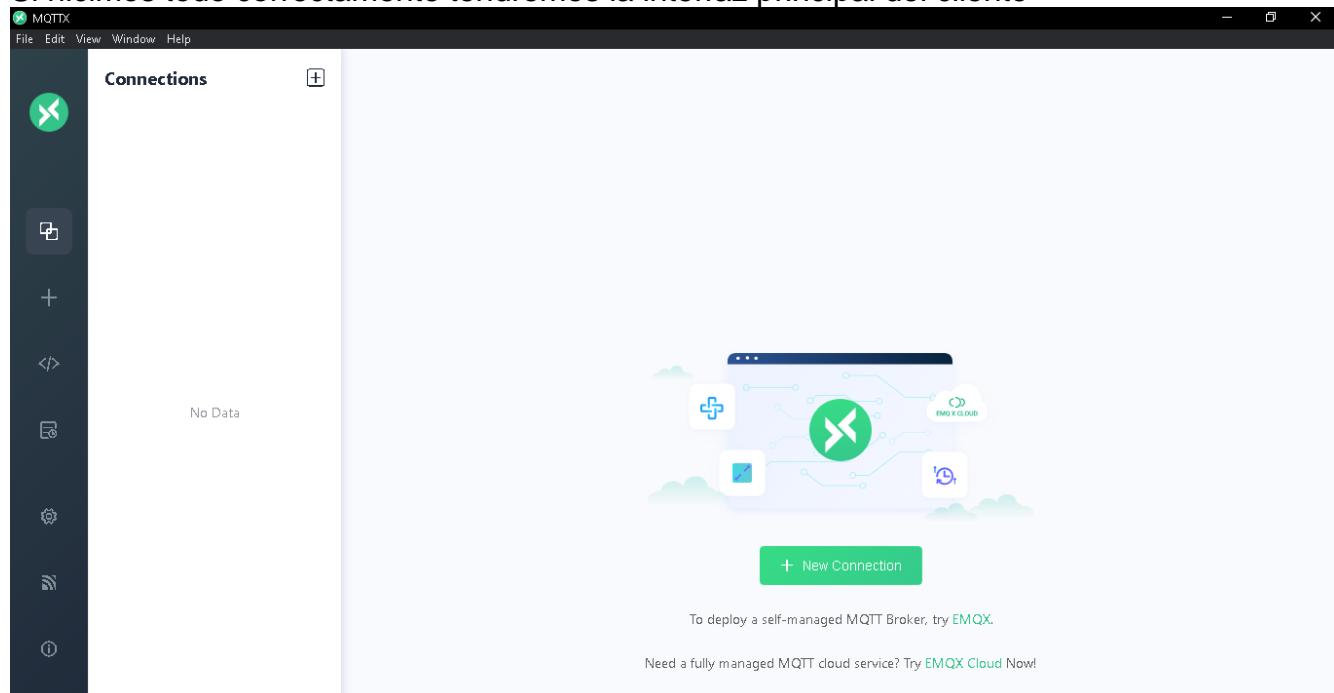
< Atrás

Siguiente >

Cancelar



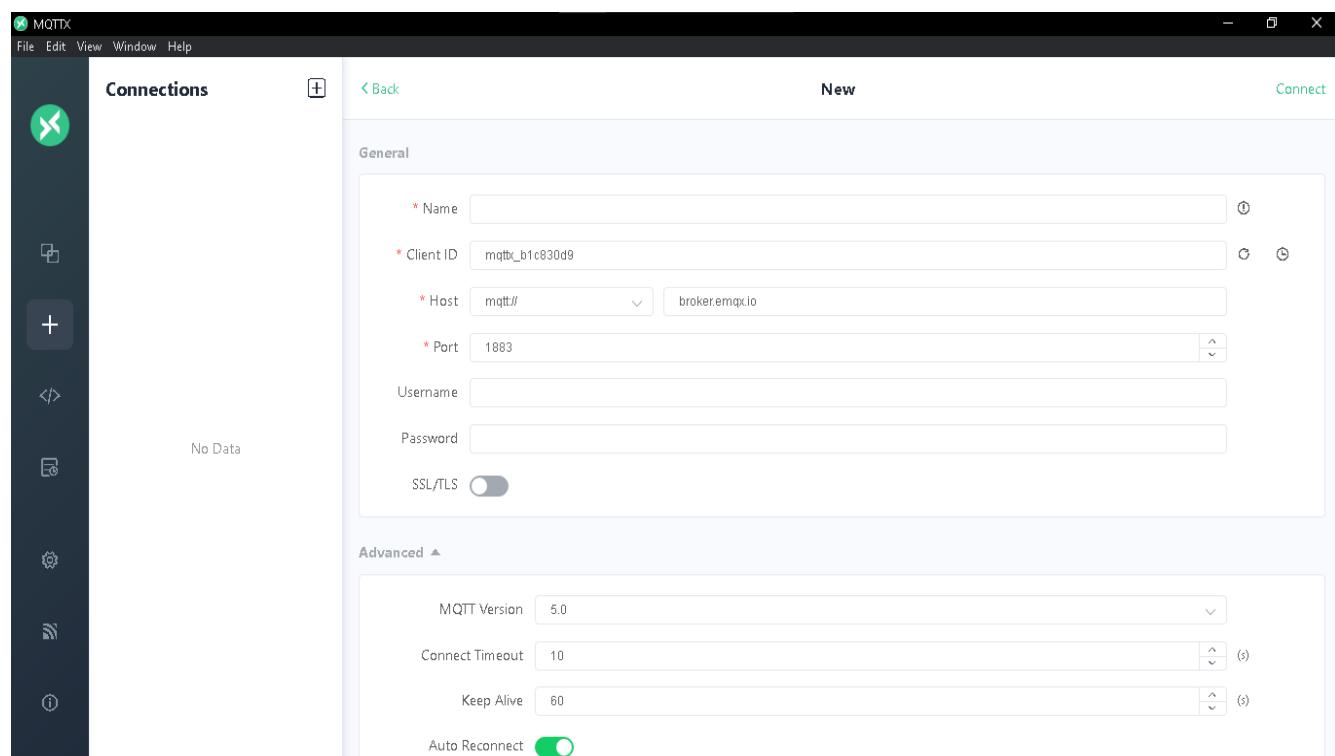
Si hicimos todo correctamente tendremos la interfaz principal del cliente





## 4.4 CONEXIÓN CON CLIENTE MQTTX

Una vez tengamos el cliente instalado en nuestro equipo lo abriremos y le daremos en nueva conexión y nos saldrá lo siguiente





En el apartado de general podremos ingresar los datos de la conexión que deseemos crear como:

General

* Name	TestCliente	Nombre de la conexión
* Client ID	mqtb_b1c830d9	Este se genera
* Host	mqtt://localhost	Como trabajamos de manera local en el host pondremos localhost
* Port	1883	Puerto al que se conectó el
Username		Credenciales de acceso
Password		
SSL/TLS	<input checked="" type="checkbox"/>	

En el apartado de Advanced tendremos lo siguiente

Advanced ▲

MQTT Version	3.1.1	Versión del MQTT 5 o 3
Connect Timeout	10	Tiempo muerto del servidor
Keep Alive	60	Tiempo que mantiene vivo el
Auto Reconnect	<input checked="" type="checkbox"/>	
Reconnect Period	4000	Si hay una desconexión intenta conectarse solo cada tiempo
Clean Session	<input checked="" type="checkbox"/>	



Y por último tenemos el apartado de Testament donde podemos poner un tópico y luego le daremos en conectar

Last Will and Testament ▾

Last-Will Topic: cliente01/status Nombre del tópico

Last-Will QoS:  0  1  2 El QoS que queremos (Recomendado el 2)

Last-Will Retain:  Si queremos retener los mensajes

Last-Will Payload: `{"status": false}` Un JSON que haga la función que queremos que haga el tópico en este caso mi JSON me permite ver el estado de conexión de mi cliente

Una vez llenado todo tendremos nuestra conexión lista

The screenshot shows the MQTTX application interface. On the left, there's a sidebar with icons for creating connections, subscriptions, and other MQTT-related functions. The main area is titled 'Connections' and shows a single connection named 'TestCliente@localhost...'. Below this, there's a 'TestCliente' tab with a 'Connected' status indicator. In the bottom right corner of the main window, there's a small preview pane showing a JSON payload: `{"msg": "hello"}`.

## Clients

All Nodes ▼

Client ID	Username	IP Address	Keepalive(s)	Expiry Interval(s)	Subscriptions Count	Connect Status	Cor	Operation
mqttx_b1c830d9	undefined	127.0.0.1:61799	60	0	0	<span style="color: green;">●</span> CONNECTED	2023-06-20 10:45:22	<span style="border: 1px solid #ccc; padding: 2px;">Kick Out</span>
mqttjs_60b081d8fb		127.0.0.1:61816	60	0	0	<span style="color: green;">●</span> CONNECTED	2023-06-20 10:45:22	<span style="border: 1px solid #ccc; padding: 2px;">Kick Out</span>

## 4.5 INSTALAR OPENSSL

Para poder conectarnos de manera segura necesitamos hacerlo por medio del protocolo SSL ya que este nos permite tener la información y mensajes cifrados del cliente al servidor y evitar alguna fuga o alguna vulnerabilidad de seguridad y por ello lo primero que necesitamos hacer es descargar openSSL en nuestro equipo [6]

**Shining Light Productions**  
"Meeting the Needs of Fellow Programmers"

[Home](#) [Products](#) [Support](#) [About](#)

© 2000-2003 Shining Light Productions 

**Win32/Win64 OpenSSL**

The Win32/Win64 OpenSSL Installation Project is dedicated to providing a simple installation of OpenSSL for Microsoft Windows. It is easy to set up and easy to use through the simple, effective installer. No need to compile anything or jump through any hoops, just click a few times and it is installed, leaving you to doing real work. Download it today! Note that these are default builds of OpenSSL and subject to local and state laws. More information can be found in the legal agreement of the installation.

LEGAL NOTICE: This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org>)

**System Requirements**

Minimum system requirements:

Windows XP or later  
32MB RAM  
200MHz CPU  
30MB hard drive space

Recommended system requirements:

Windows XP or later  
128MB RAM  
500MHz CPU  
300MB hard drive space

March 14, 2023 - OpenSSL 3.1 is available. Also, Happy Pi Day. Users should currently install a 1.1.1 series + a 3.x series for maximum application compatibility. All application developers should begin migrating their applications to use OpenSSL 3.x series. Note that 3.0.x is a LTS release series while 3.1.x is not.

September 9, 2021 - OpenSSL 3.0 is available. Users should currently install a 1.1.1 series + a 3.0 series for maximum application compatibility. All application developers should begin migrating their applications to use OpenSSL 3.0 series. Experimental, tested ARM64 builds are now available. Most users only need to install OpenSSL Light editions. The full downloads for 3.0, which include the precompiled developer libraries and a bloated test suite, have nearly doubled in size from 1.1.1 series.

April 21, 2020 - All users and applications should be using the OpenSSL 1.1.1 (LTS) series at this point.

Latest installer cryptographic hashes - MD5, SHA-1, SHA-256, and SHA-512 [available in JSON format](#). For those who are exceptionally needy. Now stop bothering me.

The following things in red are the result of my in-box being inundated with requests that resulted in many facepalm moments. The intentionally satirical responses are placed here for your enjoyment and education.

**Sponsors & Donators**

**Businesses**



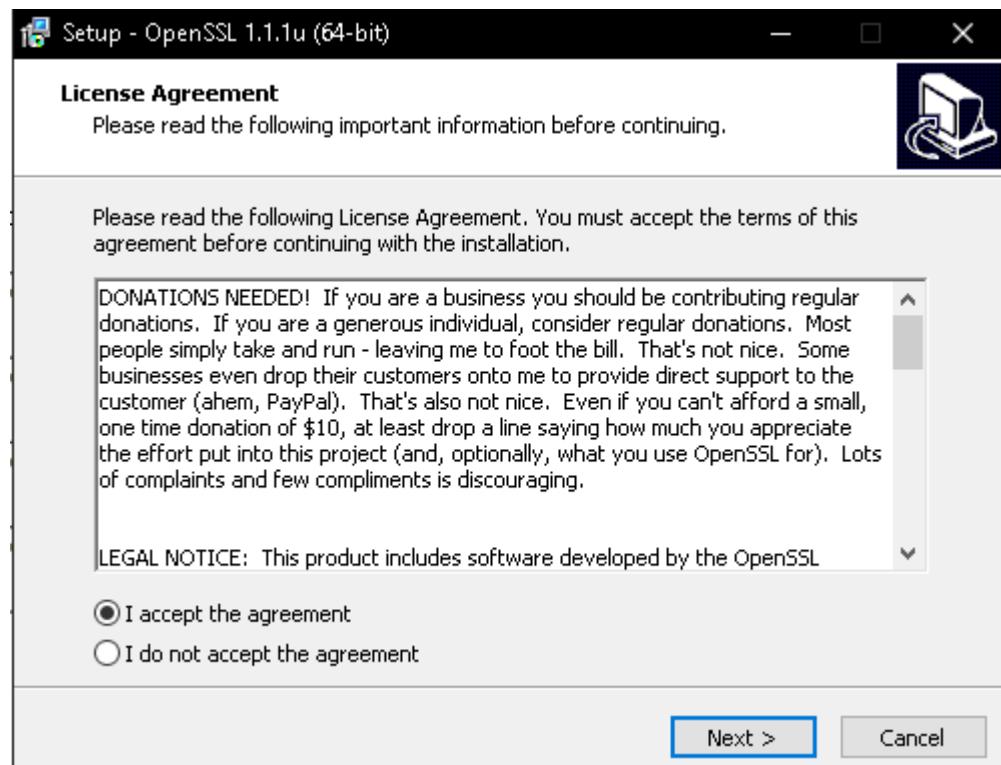
**TOTAL UPTIME**  
The Cloud Availability Platform  
[Secure Cloud Load Balancing](#)

  
**PATCH MY PC**  
Patch Management Made Easy.

Una vez en la página bajaremos y buscaremos la versión que se aadecue a nuestro equipo siempre de preferencia la versión 1.1 y que no se la light

Win64 OpenSSL v1.1.1u Light <a href="#">EXE</a>   <a href="#">MSI</a>	3MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v1.1.1u (Recommended for users by the creators of <a href="#">OpenSSL</a> ). Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v1.1.1u <a href="#">EXE</a>   <a href="#">MSI</a>	63MB Installer	Installs Win64 OpenSSL v1.1.1u (Recommended for software developers by the creators of <a href="#">OpenSSL</a> ). Only installs on 64-bit versions of Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v1.1.1u Light <a href="#">EXE</a>   <a href="#">MSI</a>	3MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v1.1.1u (Only install this if you need 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation).
Win32 OpenSSL v1.1.1u <a href="#">EXE</a>   <a href="#">MSI</a>	54MB Installer	Installs Win32 OpenSSL v1.1.1u (Only install this if you need 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation).

Una vez descargado el archivo procederemos a realizar los pasos de instalación



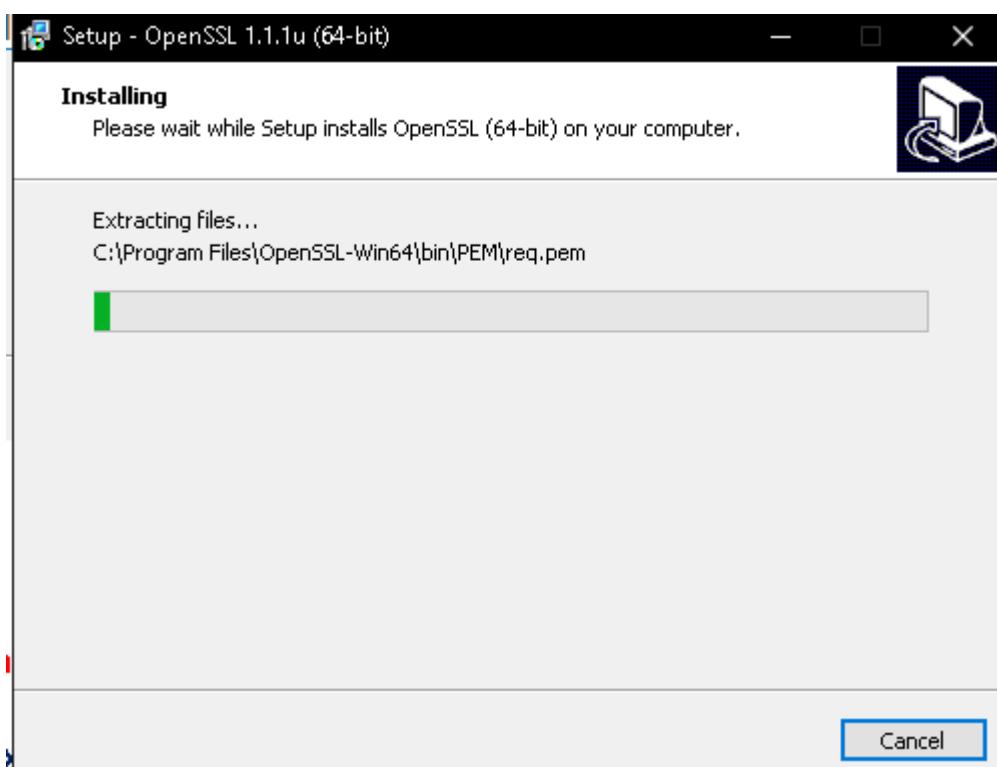
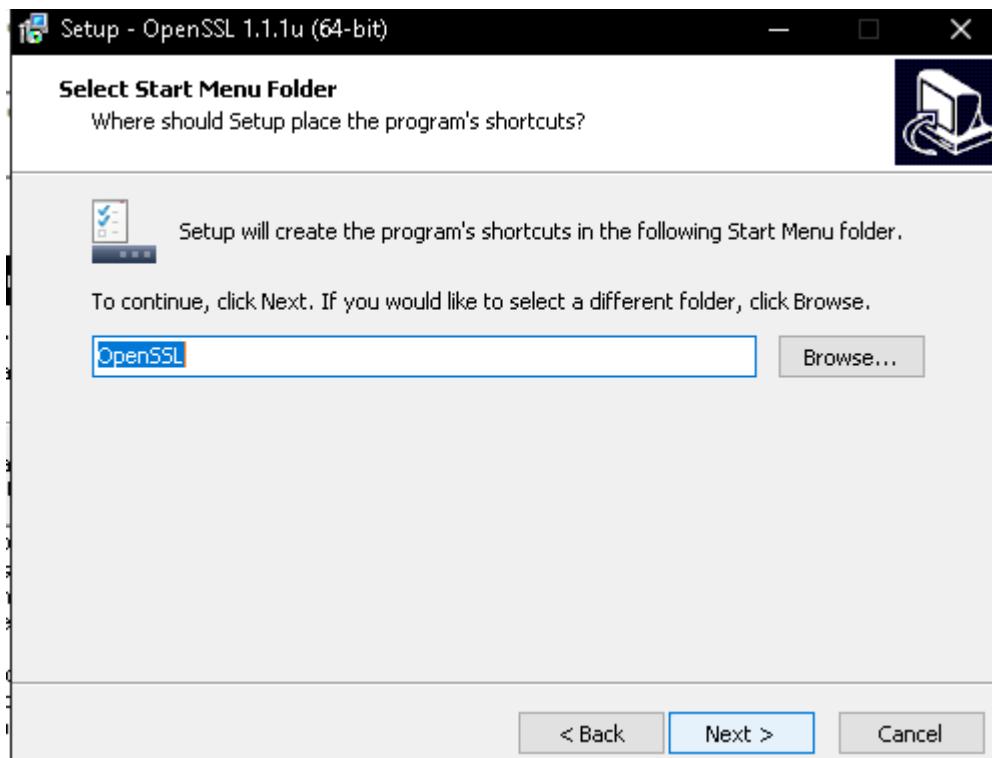


GOBIERNO DE  
MÉXICO



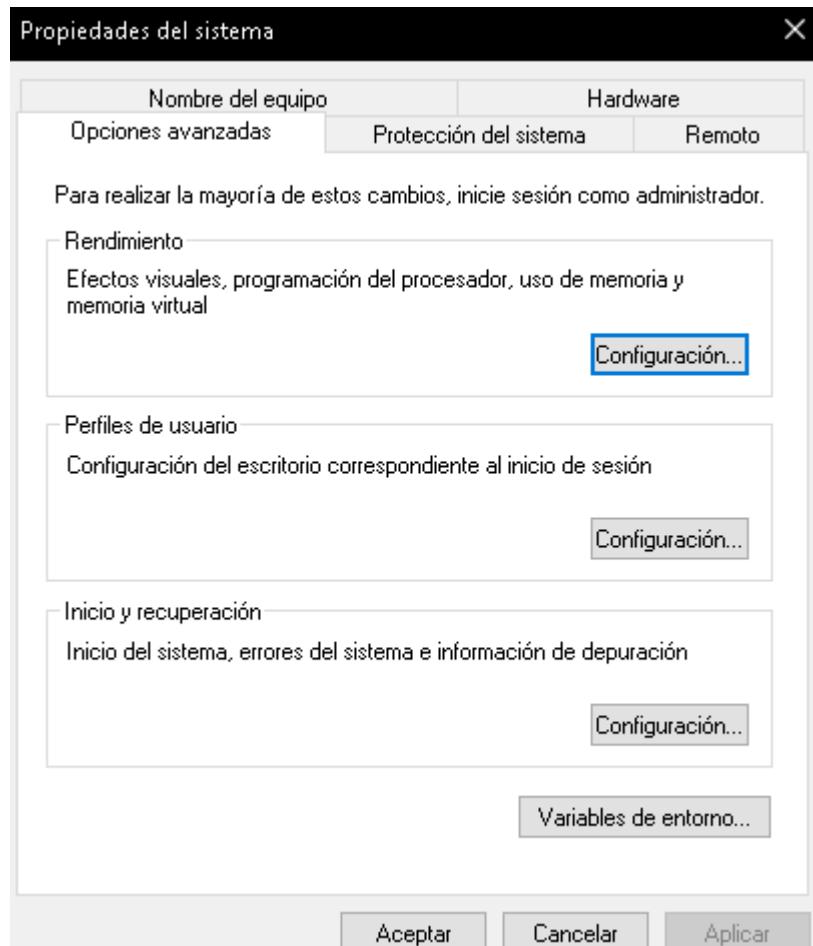
CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®





Una vez instalado buscaremos el editor de variables de entorno en Windows y lo abriremos y una vez ahí le daremos en variables de entorno y buscaremos path en las variables del sistema le daremos editar





Variables del sistema

Variable	Valor
JAVA_HOME	C:\Program Files\Java\jdk1.8.0_111
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Program... .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 42 Stepping 7. GenuineIntel

Nueva... Editar... Eliminar

Una vez ahí para agregar la nueva variable le daremos en nuevo y en examinar y nos iremos manualmente a la ruta **C:\Program Files\OpenSSL-Win64\bin** aceptamos y cerramos todas las ventanas del editor de variables

```
C:\Program Files\nodejs\  
C:\Program Files\OpenSSL-Win64\bin
```

Una vez hecho todo lo anterior abriremos un nuevo CMD y pondremos el siguiente comando **openssl**

```
C:\ Administrador: Símbolo del sistema - openssl  
Microsoft Windows [Versión 10.0.19045.2965]  
(c) Microsoft Corporation. Todos los derechos reservados.  
C:\WINDOWS\system32>openssl  
OpenSSL>
```

**Nota:** En caso de que les salga que no reconoce le comando es porque no instalaron bien la variable del sistema por lo que revisen de nuevo el editor de variables y verifiquen



## 4.6 GENERAR CERTIFICADOS AUTOFIRMADOS

Una vez hayamos instalado SLL en nuestro equipo procederemos a crear nuestros certificados y continuar con el protocolo SLL por ello abriremos una terminal y nos ubicaremos en la carpeta emqx

```
Administrator: Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.2965]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>cd..

C:\Windows>cd..

C:\>cd emqx

C:\emqx>
```

Una vez ahí crearemos una capeta para nuestros certificados llamada certs y una vez dentro agregaremos el siguiente comando **openssl genrsa -out ca.key 2048**

```
C:\emqx>cd certs

C:\emqx\certs>openssl genrsa -out ca.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
..+++++
.....+++++
e is 65537 (0x010001)
```

Disco local (C:) > emqx > certs			
Nombre	Fecha de modificación	Tipo	Tamaño
ca.key	05/06/2023 04:51 p. m.	Archivo KEY	2 KB

Después de verificar que se haya creado el archivo.key procederemos a ingresar el siguiente comando para ahora crear un certificado raíz



**openssl req -x509 -new -nodes -key ca.key -sha256 -days 3650 -out ca.pem**

```
C:\emqx\certs>openssl req -x509 -new -nodes -key ca.key -sha256 -days 3650 -out ca.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
```

**Nota:** El numero 3650 es el tiempo en que este certificado raíz va a durar ósea 3650 equivale a 10 años por lo que si queremos que este tiempo sea más o menos podemos editar ese número en el comando

Una vez ejecutado el comando nos pedirá cierta información primero el país donde se está trabajando, el nombre de la empresa y quien lo trabaja

```
C:\emqx\certs>openssl req -x509 -new -nodes -key ca.key -sha256 -days 3650 -out ca.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:QRO
Locality Name (eg, city) []:El Marquez CIDESE
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CIDESE
Organizational Unit Name (eg, section) []:CIDESE
Common Name (e.g. server FQDN or YOUR name) []:Aldair Isaac
Email Address []:aldairisaacmg10@gmail.com
C:\emqx\certs>
```

Una vez que creamos el certificado raíz procederemos a crear el certificado del servidor con el nombre emqx en mi caso en tu caso le puedes poner como quieras, por ello pondremos el siguiente comando en el CMD **openssl genrsa -out emqx.key 2048**



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

```
C:\emqx\certs>openssl genrsa -out emqx.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
```

Una vez hecho esto nos iremos a la carpeta de certs y crearemos un archivo llamado openssl.cnf y con un editor de texto le pondremos el siguiente código [7]

ca.key	05/06/2023 04:51 p. m.	Archivo KEY	2 KB
ca.pem	05/06/2023 05:05 p. m.	Archivo PEM	2 KB
emqx.key	05/06/2023 05:10 p. m.	Archivo KEY	2 KB
openssl.cnf	05/06/2023 05:23 p. m.	Archivo CNF	0 KB

```
[req]
default_bits = 2048
distinguished_name = req_distinguished_name
req_extensions = req_ext
x509_extensions = v3_req
prompt = no
[req_distinguished_name]
countryName = CN
stateOrProvinceName = Zhejiang
localityName = Hangzhou
organizationName = EMQX
commonName = CA
[req_ext]
subjectAltName = @alt_names
[v3_req]
subjectAltName = @alt_names
[alt_names]
IP.1 = BROKER_ADDRESS
DNS.1 = BROKER_ADDRESS
```

Una vez puesto el código en el editor de texto tendremos que editarlo para nosotros



```
[req]
default_bits = 2048
distinguished_name = req_distinguished_name
req_extensions = req_ext
x509_extensions = v3_req
prompt = no
[req_distinguished_name]
countryName = MX INICIALES DE NUESTRO PAÍS
stateOrProvinceName = El Marquez PROVINCIA
localityName = El Marquez LOCALIDAD
organizationName = CIDESI INSTITUCION DE TRABAJO
commonName = CIDESI ABREVIATURA DEL LA
[req_ext]
subjectAltName = @alt_names
[v3_req]
subjectAltName = @alt_names
[alt_names]
IP.1 = 127.0.0.1 DIRECCION DONDE ESTARA CORRIENDO EMQX (EN ESTE CASO COMO TRABAJAREMOS DE MANERA LOCAL ENTONCES LA DIRECCION SERA 127.0.0.1)
DNS.1 = 127.0.0.1
```

Después de esto tendremos que generar una solicitud de certificado por ello en nuestra consola de comandos y apuntando a la carpeta certs ejecutaremos el siguiente código **openssl req -new -key ./emqx.key -config openssl.cnf -out emqx.csr**

```
C:\emqx>cd certs
C:\emqx\certs>openssl req -new -key ./emqx.key -config openssl.cnf -out emqx.csr
```

	emqx.csr	07/06/2023 12:22 p. m.	Archivo CSR	2 KB
--	----------	------------------------	-------------	------



Ahora generaremos un certificado .pem con el certificado generado anteriormente por ello en la consola de comandos ingresaremos el siguiente código **openssl x509 -req -in ./emqx.csr -CA ca.pem -CAkey ca.key -CAcreateserial -out emqx.pem -days 3650 -sha256 -extensions v3\_req -extfile openssl.cnf**

```
C:\emqx\certs>openssl x509 -req -in ./emqx.csr -CA ca.pem -CAkey ca.key -CAcreateserial -out emqx.pem
-days 3650 -sha256 -extensions v3_req -extfile openssl.cnf
Signature ok
subject=C = MX, ST = El Marquez, L = El Marquez, O = CIDESI, CN = CIDESI
Getting CA Private Key
```

Nota: Si quieres comprobar los certificados creados puedes usar el comando **openssl x509 -in emqx.pem -noout -text**

Después de hacer todo esto ahora tendremos que realizar la configuración EMQX por ello en nuestra carpeta certs copiaremos los siguientes archivos

	ca.key	05/06/2023 04:51 p. m.	Archivo KEY	2 KB
	ca.pem	05/06/2023 05:05 p. m.	Archivo PEM	2 KB
	emqx.key	05/06/2023 05:10 p. m.	Archivo KEY	2 KB
	openssl.cnf	07/06/2023 12:18 p. m.	Archivo CNF	1 KB
	emqx.csr	07/06/2023 12:22 p. m.	Archivo CSR	2 KB
	ca.srl	07/06/2023 12:27 p. m.	Archivo SRL	1 KB
	emqx.pem	07/06/2023 12:27 p. m.	Archivo PEM	2 KB

Y nos iremos al siguiente directorio en nuestra carpeta emqx **emqx\etc\certs**, una vez ahí pegaremos los archivos anteriormente mencionados



Disco local (C:) &gt; emqx &gt; etc &gt; certs

Nombre	Fecha de modificación	Tipo	Tamaño
ca.key	05/06/2023 04:51 p. m.	Archivo KEY	2 KB
ca.pem	05/06/2023 05:05 p. m.	Archivo PEM	2 KB
cacert.pem	27/04/2023 02:29 a. m.	Archivo PEM	2 KB
cert.pem	27/04/2023 02:29 a. m.	Archivo PEM	2 KB
client-cert.pem	27/04/2023 02:29 a. m.	Archivo PEM	2 KB
client-key.pem	27/04/2023 02:29 a. m.	Archivo PEM	2 KB
emqx.csr	07/06/2023 12:22 p. m.	Archivo CSR	2 KB
emqx.key	05/06/2023 05:10 p. m.	Archivo KEY	2 KB
emqx.pem	07/06/2023 12:27 p. m.	Archivo PEM	2 KB
key.pem	27/04/2023 02:29 a. m.	Archivo PEM	2 KB
README	27/04/2023 02:29 a. m.	Archivo	1 KB

Una vez pegados los archivos ahora nos iremos al directorio **emqx\etc** y una vez ahí veremos un archivo llamado **emqx.conf** el cual abriremos con nuestro editor de texto favorito

Disco local (C:) &gt; emqx &gt; etc &gt;

Nombre	Fecha de modificación	Tipo	Tamaño
certs	07/06/2023 12:47 p. m.	Carpeta de archivos	
lwm2m_xml	27/04/2023 02:36 a. m.	Carpeta de archivos	
plugins	27/04/2023 02:36 a. m.	Carpeta de archivos	
acl.conf	27/04/2023 02:29 a. m.	Archivo CONF	1 KB
emqx.conf	27/04/2023 02:29 a. m.	Archivo CONF	77 KB
psk.txt	27/04/2023 02:36 a. m.	Documento de te...	1 KB
ssl_dist.conf	27/04/2023 02:29 a. m.	Archivo CONF	1 KB
vm.args	27/04/2023 02:29 a. m.	Archivo ARGS	4 KB



```
emqx.conf x
1 ## EMQX Configuration
2
3 ## NOTE: Do not change format of CONFIG_SECTION_{BGN,END} comments!
4
5 ## CONFIG_SECTION_BGN=cluster =====
6
7 ## Cluster name.
8 ##
9 ## Value: String
10 cluster.name = emqxcl
11
12 ## Specify the erlang distributed protocol.
13 ##
14 ## Value: Enum
15 ## - inet_tcp: the default; handles TCP streams with IPv4 addressing.
16 ## - inet6_tcp: handles TCP with IPv6 addressing.
17 ## - inet_tls: using TLS for Erlang Distribution.
18 ##
19 ## vm.args: -proto_dist inet_tcp
20 cluster.proto_dist = inet_tcp
21
22 ## Cluster auto-discovery strategy.
23 ##
24 ## Value: Enum
25 ## - manual: Manual join command
26 ## - static: Static node list
27 ## - mcast: IP Multicast
28 ## - dns: DNS A Record
29 ## - etcd: etcd
30 ## - k8s: Kubernetes
31 ##
32 ## Default: manual
33 cluster.discovery = manual
34
35 ## Enable cluster autoheal from network partition.
```

Como es un archivo grande usaremos los buscadores de nuestro editor para encontrar la líneas que queremos editar en (este caso línea que necesito editar está en la 1509) una vez encontrada la línea cambiaremos el key.pem por **emqx.key**

```
1507 ##
1508 ## Value: File
1509 listener.ssl.external.keyfile = etc/certs/key.pem
1510
1511 ## Path to a file containing the user certificate.
1512 ##
1513 ## See: http://erlang.org/doc/man/ssl.html
```

```
1508 ## Value: File
1509 listener.ssl.external.keyfile = etc/certs/emqx.key
1510
1511 ## Path to a file containing the user certificate.
1512 ##
1513 ## See: http://erlang.org/doc/man/ssl.html
1514 ##
```



Luego cambiaremos el cert.pem por **emqx.pem** y el cacert.pem por **ca.pem** y guardaremos todo

```
1515 ## Value: File
1516 | listener.ssl.external.certfile = etc/certs/cert.pem
1517
1518 ## Path to the file containing PEM-encoded CA certificates. The CA certificates
1519 ## file must contain the certificate chain from the root CA to the certificate used by the listener.
```

```
1515 ## Value: File
1516 | listener.ssl.external.certfile = etc/certs/emqx.pem
1517
1518 ## Path to the file containing PEM-encoded CA certificates. The CA certificates
1519 ## file must contain the certificate chain from the root CA to the certificate used by the listener.

1520 ##
1521 ## Value: File
1522 | listener.ssl.external.cacertfile = etc/certs/cacert.pem
1523
```

```
1520 ##
1521 ## Value: File
1522 | listener.ssl.external.cacertfile = etc/certs/ca.pem
1523
```

Una vez hecho tendremos que reiniciar el bróker por CMD con **emqx restart** y abrir el cliente MQTTX y conectarnos pero ahora habilitando la conexión por SLL/TCL y por el puerto 8883

General

* Name	TestCliente
* Client ID	mqtb_b1c830d9
* Host	mqtts://localhost
* Port	8883
Username	
Password	
SSL/TLS	<input checked="" type="checkbox"/>
SSL Secure	<input type="checkbox"/>
Certificate	<input checked="" type="radio"/> CA signed server <input type="radio"/> Self signed



**Nota:** Con la opción CA signed server activada el cliente encontrara el certificado automático de lo contrario con la otra opción nosotros tendremos que seleccionar el certificado CA que deseemos utilizar de manera manual

Es importante recordar que el puerto 8883 es para conectarnos por SSL MQTT pero si deseamos conectarnos por websocket seguro entonces el puerto será el 8084 e igualmente antes tendríamos que buscar el puerto 8084 en el archivo emqx.conf y editar los certificados como hicimos anteriormente

```
2033 ## Examples: 8084, 127.0.0.1:8084, ::1:8084
2034 listener.wss.external = 8084
2035
```

De key.pem a emqx.key

```
## Value: File
listener.wss.external.keyfile = etc/certs/key.pem
```

```
## Value: File
listener.wss.external.keyfile = etc/certs/emqx.key
```

De cert.pem a emqx.pem

```
##
## Value: File
listener.wss.external.certfile = etc/certs/cert.pem
```

```
##
## Value: File
listener.wss.external.certfile = etc/certs/emqx.pem
```

El ultimo no es necesario habilitarlo y cambiar los certificados pero loaremos aun así para tener todo en un orden

```
##
## Value: File
## listener.wss.external.cacertfile = etc/certs/cacert.pem
```

```
## Value: File
listener.wss.external.cacertfile = etc/certs/ca.pem
```



## 4.7 CONEXIÓN SEGURA HTTPS EN EL DASHBOARD

Una vez ya tengamos configurado todos los certificados ahora tocara conectarnos por una conexión segura para ello abriremos la carpeta **emqx/etc/plugins** y con nuestro editor de texto abriremos el archivo **emqx\_dashboard.conf**

```
emqx_dashboard.conf x
1 ##-----
2 ## EMQX Dashboard
3 ##-----
4
5 ## Default user's login name.
6 ## For safety, it should be changed as soon as possible.
7 ## Please use the './bin/emqx_ctl admins' CLI to change it.
8 ## Then comment `dashboard.default_user.login/password` from here
9 ## Value: String
10 dashboard.default_user.login = admin
11
12 ## Default user's password.
13 ## The initial default password for 'dashboard.default_user.login'"
14 ## For safety, it should be changed as soon as possible.
15 ## Please use the './bin/emqx_ctl admins' CLI to change it.
16 ## Then comment `dashboard.default_user.login/password` from here
17 ## Value: String
18 dashboard.default_user.password = public
19
20 ##-----
21 ## HTTP Listener
22
23 ## The port that the Dashboard HTTP listener will bind.
24 ##
25 ## Value: Port
26 ##
27 ## Examples: 18083
28 dashboard.listener.http = 18083
29
30 ## The acceptor pool for external Dashboard HTTP listener.
31 ##
32 ## Value: Number
33 dashboard.listener.httpn_acceptors = 4
```



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez ahí con el buscador de nuestro editor localizaremos el puerto 18084

```
##  
## Examples: 18084  
## dashboard.listener.https = 18084
```

Una vez localizado empezaremos a habilitar algunas líneas de código (borrando los ##) y cambiar los certificados por los que generemos antes

```
57 ## Examples: 18084  
58 dashboard.listener.https = 18084  
59  
60 ## The acceptor pool for external Dashboard HTTPS listener.  
61 ##  
62 ## Value: Number  
63 dashboard.listener.https.acceptors = 2  
64  
65 ## Maximum number of concurrent Dashboard HTTPS connections.  
66 ##  
67 ## Value: Number  
68 dashboard.listener.https.max_clients = 512  
69  
70 ## Set up the socket for IPv6.  
71 ##  
72 ## Value: false | true  
73 dashboard.listener.https.inet6 = false  
74  
75 ## Listen on IPv4 and IPv6 (false) or only on IPv6 (true). Use with inet6.  
76 ##  
77 ## Value: false | true  
78 | dashboard.listener.https.ipv6_v6only = false  
79
```

De key.pem a emqx.key

```
81 ##  
82 ## Value: File  
83 dashboard.listener.https.keyfile = etc/certs/emqx.key  
84
```

De cert.pem a emqx.pem

```
92 ##  
93 ## Value: File  
94 dashboard.listener.https.certfile = etc/certs/emqx.pem  
95
```



De cacert.pem a ca.emqx.

```
97 ##  
98 ## Value: File  
99 dashboard.listener.https.cacertfile = etc/certs/ca.pem  
100
```

Una vez echo y guardado todo nos iremos ahí mismo en plugins al archivo llamado **emqx\_management.conf**

```
emqx_management.conf x  
1 ##-----  
2 ## EMQX Management Plugin  
3 ##-----  
4  
5 ## Max Row Limit  
6 management.max_row_limit = 10000  
7  
8 ## Application default secret  
9 ##  
10 ## Value: String  
11 ## management.application.default_secret = public  
12  
13 ## Default Application ID  
14 ##  
15 ## Value: String  
16 management.default_application.id = admin  
17  
18 ## Default Application Secret  
19 ##  
20 ## Value: String  
21 management.default_application.secret = public  
22  
23 ## Initialize apps file  
24 ## Is used to add administrative app/secrets when EMQX is launched for the first time.  
25 ## This config will not take any effect once EMQX database has one or more apps.  
26 ## The file content format is as below:  
27 ## ``  
28 ##819e5db182cf:19C5suZClF3FvdzWqmINrVU61WNfIjcglxw9CVM7y1VI  
29 ##bb5a6cf1c06a:WuNRgCRTGiNcuyrE49Bpwz4PGPrRnP4hUMi647kNSbN  
30 ## ``  
31 # management.bootstrap_apps_file = etc/bootstrap_apps.txt  
32  
33 ##-----
```



Igualmente en este archivo tendremos que habilitar algunas líneas y editar cosas

```
45  ##-----
46  ## HTTPS Listener
47
48  ## management.listener.https = 8081
49  ## management.listener.https.acceptors = 2
50  ## management.listener.https.max_clients = 512
51  ## management.listener.https.backlog = 512
52  ## management.listener.https.send_timeout = 15s
53  ## management.listener.https.send_timeout_close = on
54  ## management.listener.https.certfile = etc/certs/cert.pem
55  ## management.listener.https.keyfile = etc/certs/key.pem
56  ## management.listener.https.key_password = yourpass
57  ## management.listener.https.cacertfile = etc/certs/cacert.pem
58  ## management.listener.https.verify = verify_peer
59  ## NOTE: Do not use tlsv1.3 if emqx is running on OTP-22 or earlier
60  ## management.listener.https.tls_versions = tlsv1.3,tlsv1.2,tlsv1.1,tlsv1
```

```
45  ##-----
46  ## HTTPS Listener
47
48  management.listener.https = 8082
49  management.listener.https.acceptors = 2
50  management.listener.https.max_clients = 512
51  management.listener.https.backlog = 512
52  management.listener.https.send_timeout = 15s
53  management.listener.https.send_timeout_close = on
54  management.listener.https.certfile = etc/certs/emqx.pem
55  management.listener.https.keyfile = etc/certs/emqx.key
56  ## management.listener.https.key_password = yourpass
57  management.listener.https.cacertfile = etc/certs/ca.pem
58  ## management.listener.https.verify = verify_peer
59  ## NOTE: Do not use tlsv1.3 if emqx is running on OTP-22 or earlier
60  ## management.listener.https.tls_versions = tlsv1.3,tlsv1.2,tlsv1.1,tlsv1
```



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

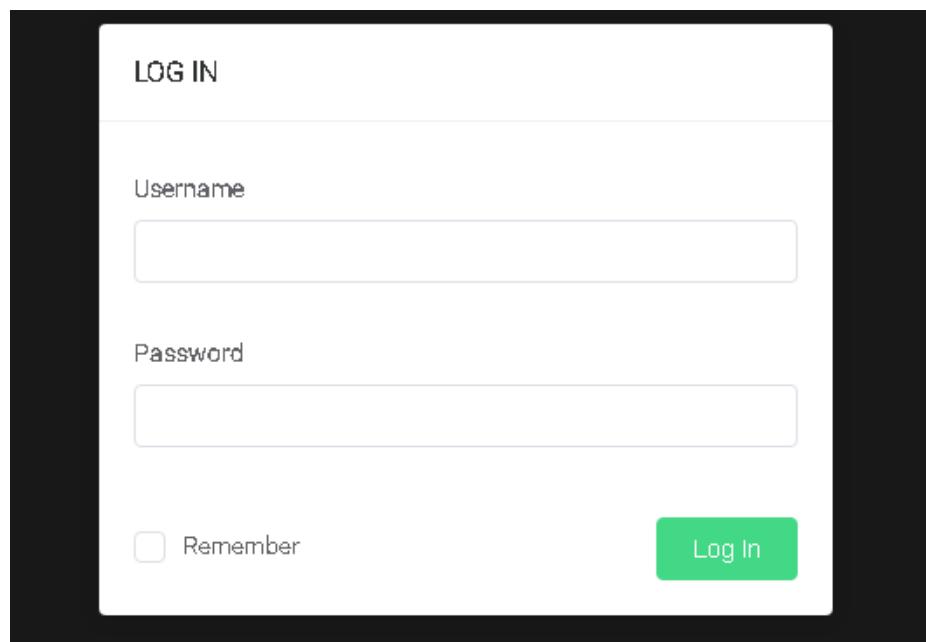
Una vez hecho esto abriremos el CMD y en la ruta bin pararemos el bróker y lo volveremos a ejecutar con el comando console para que se inicie de nuevo el bróker y ver que no exista ningún problema (como veremos en consola ya está habilitado la conexión https)

```
C:\emqx\bin>emqx stop
ok

C:\emqx\bin>emqx console

C:\emqx>"C:\emqx\erts-12.3.2.6\bin\erl.exe" -boot "C:\emqx\releases\4.4.18\start" -config c:/emqx/data/configs/app.2023.06.07.15.02.28.config -args_file c:/emqx/data/configs/vm.2023.06.07.15.02.28.args -vm_args c:/emqx/data/configs/vm.2023.06.07.15.02.28.args -mnesia dir 'c:/emqx/data/mnesia/emqx@127.0.0.1'
Starting emqx on node emqx@127.0.0.1
Start mqtt:tcp:internal listener on 127.0.0.1:11883 successfully.
Start mqtt:tcp:external listener on 0.0.0.0:1883 successfully.
Start mqtt:ws:external listener on 0.0.0.0:8083 successfully.
Start mqtt:ssl:external listener on 0.0.0.0:8883 successfully.
Start mqtt:wss:external listener on 0.0.0.0:8084 successfully.
Start https:management listener on 8082 successfully.
Start http:management listener on 8081 successfully.
Start http:dashboard listener on 18083 successfully.
Start https:dashboard listener on 18084 successfully.
EMQX 4.4.18 is running now!
Eshell V12.3.2.6 (abort with ^G)
(emqx@127.0.0.1)1>
```

Una vez todo este correcto pondremos en el buscador <https://localhost:18084> y ya nos saldrá nuestro dashboard por conexión https





GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Si le damos en inspeccionar los certificados de la página nos mostrara los certificados que creamos anteriormente

Visor de certificados: CIDESI X

**General** Detalles

**Enviado a**

Nombre común (CN)	CIDESI
Organización (O)	CIDESI
Unidad organizativa (OU)	<No incluido en el certificado>

**Emitido por**

Nombre común (CN)	Aldair Isaac
Organización (O)	CIDESI
Unidad organizativa (OU)	CIDESI

**Período de validez**

Emitido el	miércoles, 7 de junio de 2023, 12:27:43
Vencimiento el	sábado, 4 de junio de 2033, 12:27:43

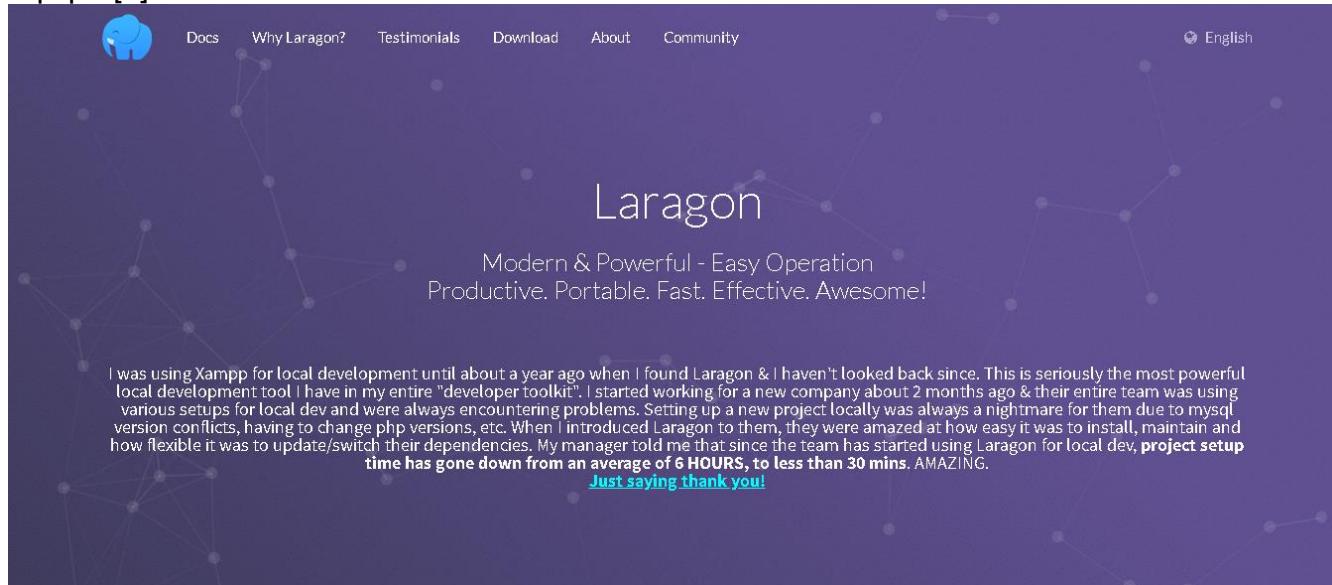
**Huellas digitales**

Huella digital SHA-256	D8 72 16 CF 31 C5 5B CF CE 3D 01 89 D8 BA B3 BE F2 E1 34 A7 95 84 EF 6E ED 9C A2 BA A3 74 2B B4
Huella digital SHA-1	BA 9C F8 3C 80 B3 69 DC 6A 4D A0 6E 93 38 C9 7A 8A 5D A9 A0



## 4.8 AUTENTIFICACION MySQL con Laragon

Ahora que tenemos nuestro bróker levantado y conectado con el cliente es importante implementar una autenticación para evitar conexiones de dispositivos desconocidos y que cualquiera pueda acceder a la información que guardamos por ello lo primero será crear una base datos con la ayuda de Laragon para ello tendremos que instalar Laragon en nuestro equipo [8]



Buscaremos y descargaremos la adecuada para nuestro equipo y lo instalamos

### Other versions

Laragon 5.0 (32-bit)

### Migrate to Laragon

Migrate from WAMP  
Migrate from XAMPP  
Migrate from others

### Tutorials

How to install Rails  
Build a Blog with Rails  
How to install Django  
How to install Flask

### Misc.

## Download

Laragon is a universal development environment. It has many features to make you more productive:

### Benefits of Laragon

After downloading, You can add **git, phpmyadmin, Node.js/MongoDB, Python/Django/Flask/Postgres, Ruby, Java, Go** using "Tools > Quick add"

Note: [You can also download from GitHub](#)



### Contents

Edition  
Why Laragon?  
Tutorials  
[Back to Top](#)

### Edition

#### [Download Laragon - Full \(173 MB\)](#)

- **Laragon Full (64-bit):** Apache 2.4, Nginx, MySQL 8, PHP 8, Redis, Memcached, Node.js 18, npm, git

#### [Download Laragon - Portable \(38 MB\)](#)

- **Laragon Portable:** PHP 5.4, MySQL 5.1, bitman - Good for getting started with PHP, then you can add newer versions of PHP/MySQL easily later using "Tools > Quick add"



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®



Instalar - Laragon 6.0.0



#### Auto start, Auto detect and create virtual hosts

You can change later in Laragon by going to Menu > Preferences



Laragon runs extremely fast and has very low memory footprint (< 4 MB)

Run Laragon when Windows starts

Your app will get pretty url ---> <http://app.test>

Auto virtual hosts

Quick ways to open Text Editor & Command Prompt

Add Notepad++ & Terminal to the Right-Click Menu

Atrás

Siguiente

Cancelar



Instalar - Laragon 6.0.0



#### Instalando

Por favor, espere mientras se instala Laragon en su sistema.



Extrayendo archivos...

C:\laragon\bin\git\mingw64\share\doc\git-doc\git-count-objects.html



Cancelar



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez instalado Laragon tendremos que iniciar lo

Laragon Full 6.0 220916 php-7.4.19-Win32-vc15-x64 [TS] 192.168.56.1

X



Menú

h ?



© Leo K

Apache httpd-2.4.47-win64-VS16 started

80

Recargar

MySQL mysql-5.7.33-winx64 started

3306

Detener     Web     Base de Datos     Terminal     Root

Es importante en las configuraciones de Laragon habilitar el puerto SSL

Preferencias

General  Servicios & Puertos  Mail Catcher  Mail Sender

Apache:  SSL:   Habilitado

MySQL:

[Avanzado](#)



Una tengamos esto le daremos clik en base de datos y crearemos un administrador en este caso yo la llamaré iot emqx

Administrador de sesiones

Filter ...

Nombre de la sesión	Host
practica2	127.0...
iot emqx	127.0...

Ajustes Avanzado Estadísticas

Tipo de red: MariaDB or MySQL (TCP/IP)

Library: libmariadb.dll

Nombre del host / IP: 127.0.0.1

Pedir credenciales

Usar autenticación de Windows

Usuario: root

Contraseña:

Puerto: 3306

Protocolo cliente/servidor comprimido

Bases de datos: Separadas por punto y coma ()

Comentario:



Una vez creado el administrador crearemos una base de datos en este lo llamare mqtt

The screenshot shows the MySQL Workbench interface. On the left, a tree view lists databases: iot, emqx, information\_schema, crudnodejsmysql, mascotas, mysql, performance\_schema, provedor, sys, and videojuegos. A central window titled 'Base de datos (8)' displays a dialog box for creating a new database. The 'Nombre:' field contains 'mqtt', and the 'Collation:' dropdown is set to 'utf8\_spanish\_ci'. Below the dialog, a preview of the 'CREATE DATABASE' SQL command is shown: `CREATE DATABASE `mqtt` /*I40100 COLLATE 'u`. At the bottom right of the dialog are 'Aceptar' (Accept) and 'Cancelar' (Cancel) buttons.

En la base de datos crearemos una nueva consulta y ahí crearemos la siguiente tabla [10]

The screenshot shows the MySQL Workbench interface. The top bar indicates 'Host: 127.0.0.1' and 'Base de datos: mqtt'. The main area contains a SQL editor with the following code:

```
1 CREATE TABLE `mqtt_user` (
2     `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
3     `username` varchar(100) DEFAULT NULL,
4     `password` varchar(100) DEFAULT NULL,
5     `salt` varchar(35) DEFAULT NULL,
6     `is_superuser` tinyint(1) DEFAULT 0,
7     `created` datetime DEFAULT NULL,
8     PRIMARY KEY (`id`),
9     UNIQUE KEY `mqtt_username` (`username`)
10 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Below the SQL editor, the database tree shows the 'mqtt' database selected, containing the 'mqtt\_user' table. The table has a size of 32.0 KiB. The bottom of the interface shows a decorative footer with the Mexican coat of arms and the text '2023 Francisco Villa'.



Una vez generada la base de datos y la tabla crearemos una nueva consulta y en ella pondremos el siguiente código para crear el usuario user: emqx password:public

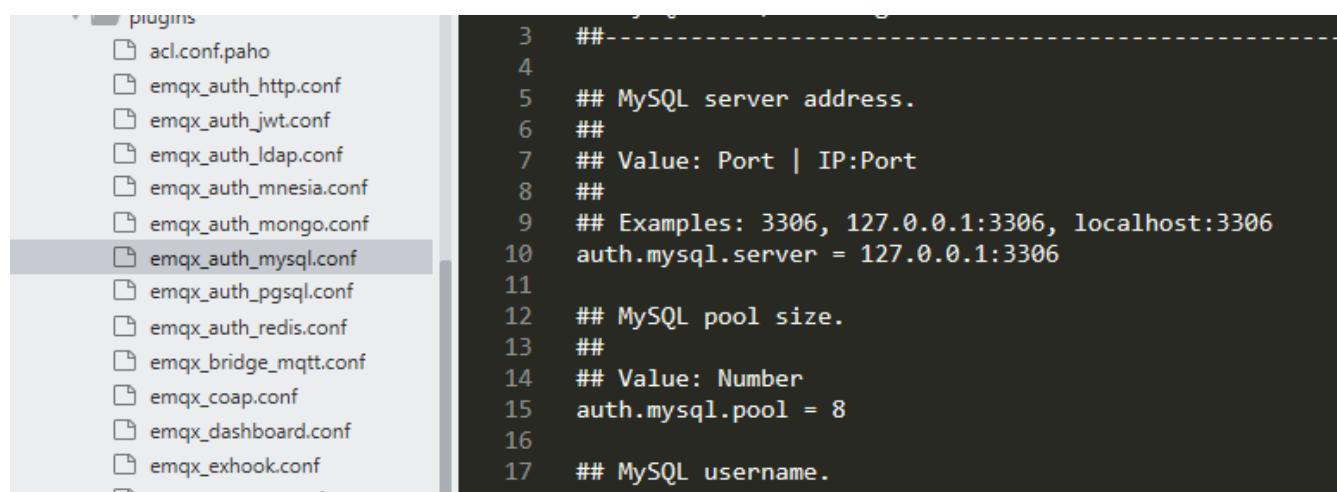


```
Host: 127.0.0.1 Base de datos: mqtt Tabla: mqtt_user Datos Consulta* Consulta #2* X
```

```
1 INSERT INTO `mqtt_user` (`username`, `password`, `salt`)
2 VALUES
3     ('emqx', 'efa1f375d76194fa51a3556a97e641e61685f914d446979da50a551a4333ffd7', NULL);
4
```

id	username	password	salt	is_superuser	created
1	emqx	efa1f375d76194fa51a3556a97e641e61685f914d446979da50a551a4333ffd7	(NULL)	0	(NULL)

Una vez hecho esto nos iremos a la carpeta **emqx/etc/plugins** y con nuestro editor de texto abriremos el archivo llamado **emqx\_auth\_mysql.conf**



```
plugins
  acl.conf.paho
  emqx_auth_http.conf
  emqx_auth_jwt.conf
  emqx_auth_ldap.conf
  emqx_auth_mnesia.conf
  emqx_auth_mongo.conf
  emqx_auth_mysql.conf
  emqx_auth_pgsql.conf
  emqx_auth_redis.conf
  emqx_bridge_mqtt.conf
  emqx_coap.conf
  emqx_dashboard.conf
  emqx_exhook.conf
```

```
3 ##
4
5 ## MySQL server address.
6 ##
7 ## Value: Port | IP:Port
8 ##
9 ## Examples: 3306, 127.0.0.1:3306, localhost:3306
10 auth.mysql.server = 127.0.0.1:3306
11
12 ## MySQL pool size.
13 ##
14 ## Value: Number
15 auth.mysql.pool = 8
16
17 ## MySQL username.
```



Una vez abierto el archivo tendremos que editarle cosas

```
1 ##-----  
2 ## MySQL Auth/ACL Plugin  
3 ##-----  
4  
5 ## MySQL server address.  
6 ##  
7 ## Value: Port | IP:Port  
8 ##  
9 ## Examples: 3306, 127.0.0.1:3306, localhost:3306  
10 auth.mysql.server = 127.0.0.1:3306  
11  
12 ## MySQL pool size.  
13 ##  
14 ## Value: Number  
15 auth.mysql.pool = 8  
16  
17 ## MySQL username.  
18 ##  
19 ## Value: String  
20 #auth.mysql.username =  
21  
22 ## MySQL password.  
23 ##  
24 ## Value: String  
25 #auth.mysql.password =  
26  
27 ## MySQL database.  
28 ##  
29 ## Value: String  
30 auth.mysql.database = mqtt  
31  
32 ## MySQL query timeout  
33 ##
```



```
5 ## MySQL server address.  
6 ##  
7 ## Value: Port | IP:Port  
8 ##  
9 ## Examples: 3306, 127.0.0.1:3306, localhost:3306  
10 auth.mysql.server = 127.0.0.1:3306 ← Colocamos la dirección del servidor en mi caso como estoy trabajando de manera local se quedara así  
11  
12 ## MySQL pool size.  
13 ##  
14 ## Value: Number  
15 auth.mysql.pool = 8  
16  
17 ## MySQL username.  
18 ##  
19 ## Value: String  
20 auth.mysql.username = emqx ← Coloco el usuario que creamos en la base de datos  
21  
22 ## MySQL password.  
23 ##  
24 ## Value: String  
25 auth.mysql.password = public ← Coloco la contraseña que creamos en la base de datos  
26  
27 ## MySQL database.  
28 ##  
29 ## Value: String  
30 auth.mysql.database = mqtt ← El nombre de nuestra base de datos  
31  
32 ## MySQL query timeout  
33 ##  
34 ## Value: Duration  
35 auth.mysql.query_timeout = 5s ← Habilito esta línea
```



Después de crear los cambios en el archivo tendremos que guardar y reiniciar el bróker como lo hemos visto anteriormente.

Y una vez hecho esto en nuestra base de datos agregaremos un nuevo administrador con las credenciales que creamos antes que es user emqx y contraseña public con todos los permisos de administración para ello nos iremos a este apartado



HS Administrador de usuarios - Sin nombre

Seleccione cuenta de usuario:

Nombre d...	Host
mysql.ses...	localhost
mysql.sys	localhost
root	localhost
Sin nombre	localhost

Credenciales Limitaciones Opciones SSL

Nombre de usuario: emqx  
Desde el host: localhost  
Contraseña:   
Repetir contraseña:

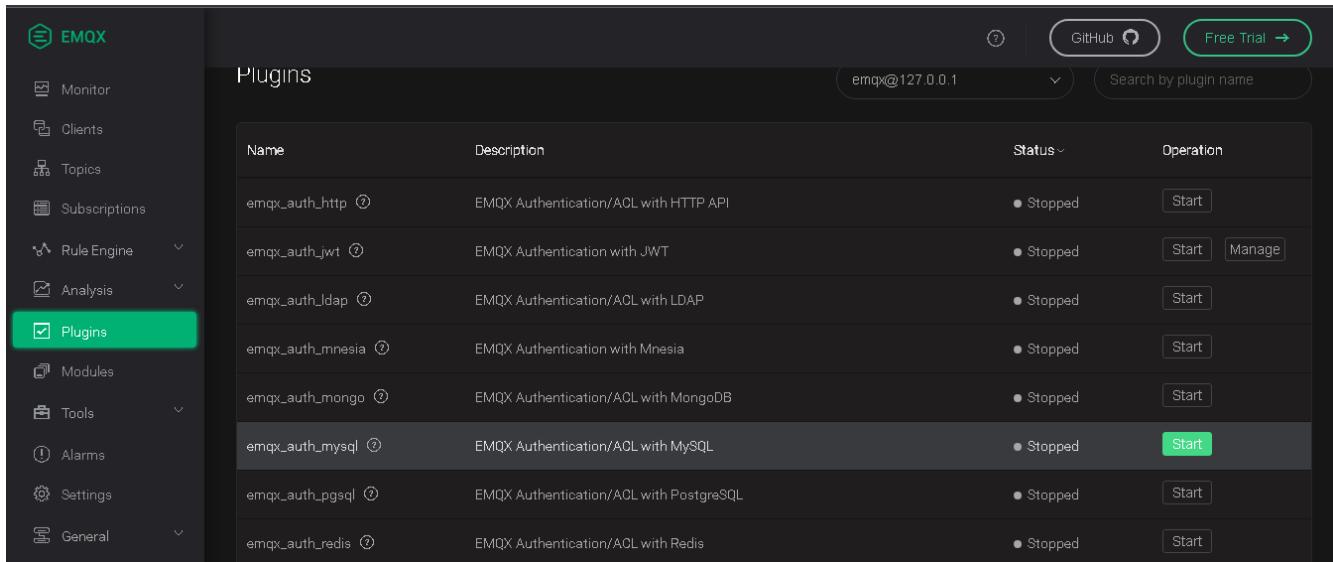
Permitir acceso a: Agregar objeto

Privilegios globales

- EXECUTE
- PROCESS
- SELECT
- SHOW DATABASES
- SHOW VIEW
- ALTER

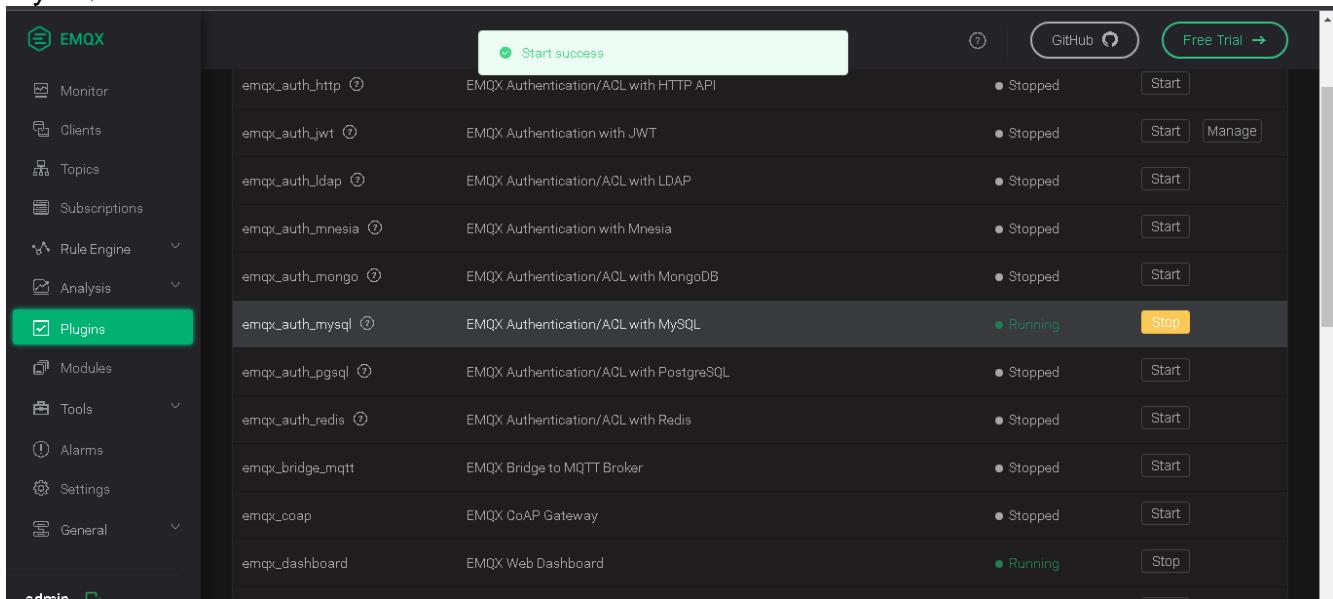
Guardar Descartar Cerrar

Luego nos iremos a nuestro dashboard y una vez nos vamos al apartado de plugins y una vez ahí le daremos start al que dice with MySQL



Name	Description	Status	Operation
emqx_auth_http ⓘ	EMQX Authentication/ACL with HTTP API	● Stopped	<button>Start</button>
emqx_auth_jwt ⓘ	EMQX Authentication with JWT	● Stopped	<button>Start</button> <button>Manage</button>
emqx_auth_ldap ⓘ	EMQX Authentication/ACL with LDAP	● Stopped	<button>Start</button>
emqx_auth_mnesia ⓘ	EMQX Authentication with Mnesia	● Stopped	<button>Start</button>
emqx_auth_mongo ⓘ	EMQX Authentication/ACL with MongoDB	● Stopped	<button>Start</button>
emqx_auth_mysql ⓘ	EMQX Authentication/ACL with MySQL	● Stopped	<button>Start</button>
emqx_auth_pgsql ⓘ	EMQX Authentication/ACL with PostgreSQL	● Stopped	<button>Start</button>
emqx_auth_redis ⓘ	EMQX Authentication/ACL with Redis	● Stopped	<button>Start</button>

Y si hicimos todo correctamente el plugin funcionara y nos creara la autentificación por MySQL



Name	Description	Status	Operation
emqx_auth_http ⓘ	EMQX Authentication/ACL with HTTP API	● Stopped	<button>Start</button>
emqx_auth_jwt ⓘ	EMQX Authentication with JWT	● Stopped	<button>Start</button> <button>Manage</button>
emqx_auth_ldap ⓘ	EMQX Authentication/ACL with LDAP	● Stopped	<button>Start</button>
emqx_auth_mnesia ⓘ	EMQX Authentication with Mnesia	● Stopped	<button>Start</button>
emqx_auth_mongo ⓘ	EMQX Authentication/ACL with MongoDB	● Stopped	<button>Start</button>
emqx_auth_mysql ⓘ	EMQX Authentication/ACL with MySQL	● Running	<button>Stop</button>
emqx_auth_pgsql ⓘ	EMQX Authentication/ACL with PostgreSQL	● Stopped	<button>Start</button>
emqx_auth_redis ⓘ	EMQX Authentication/ACL with Redis	● Stopped	<button>Start</button>
emqx_bridge_mqtt	EMQX Bridge to MQTT Broker	● Stopped	<button>Start</button>
emqx_coap	EMQX CoAP Gateway	● Stopped	<button>Start</button>
emqx_dashboard	EMQX Web Dashboard	● Running	<button>Stop</button>



Ya que tenemos la autenticación por MySQL lista procederemos a configurar el bróker para que evite conexión anónima por ello iremos al directorio **emqx/etc** y ahí con nuestro editor de texto abriremos el archivo **emqx.conf** y buscaremos la línea que diga **allow\_anonymous**

```
657  ## (without username). So the auth backend check
658  ##
659  ## Value: true | false | false_quick_deny
660  allow_anonymous = true
661
662  ## Allow or deny if no ACL rules matched.
663  ##
664  ## Value: allow | deny
665  acl_nomatch = allow
666
```

Cambiaremos el valor de la línea de verdadero a falso y reiniciamos nuestro bróker

```
658  ##
659  ## Value: true | false | false_quick_deny
660  allow_anonymous = false
661
```

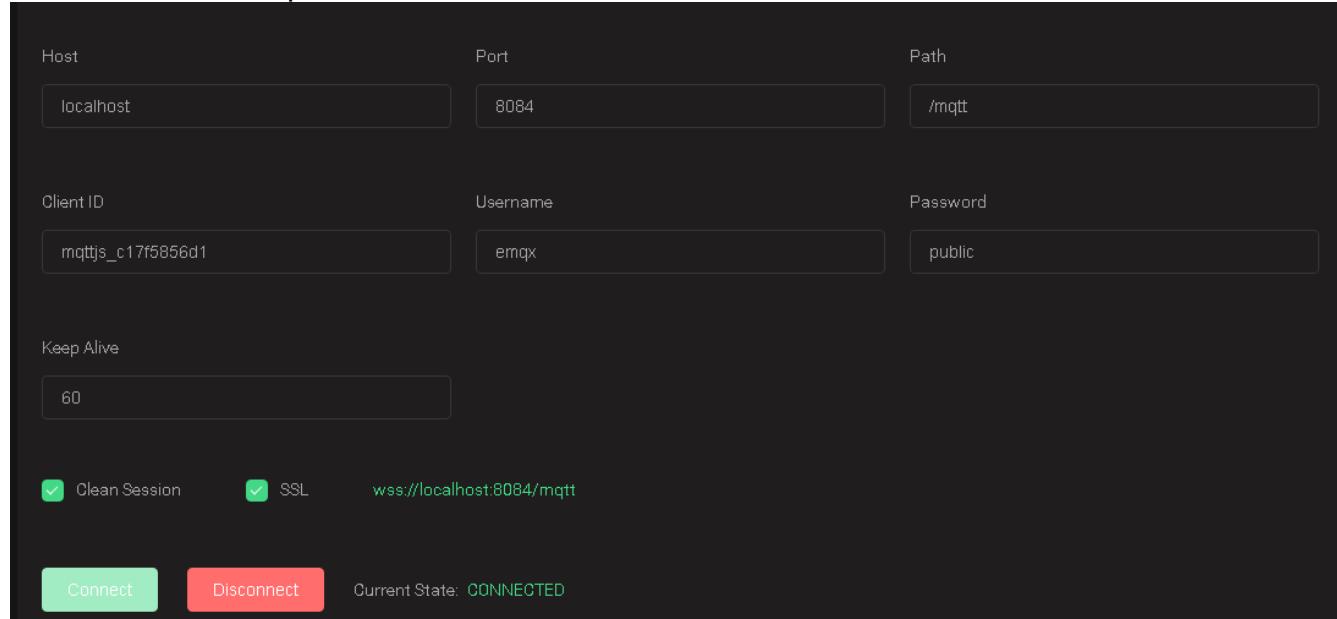
```
C:\emqx\bin>emqx start
```

```
C:\emqx>cd bin
```

```
C:\emqx\bin>emqx restart
ok
```

```
C:\emqx>
```

Una vez hayamos hecho esto nuestro bróker ya no se podrá conectar a menos que coloquemos las credenciales que creamos anteriormente en la base de datos : user:**emqx** password **public** y con ello ya no se podrán conectar al bróker a menos que coloquen las credenciales correspondientes



**Nota** nosotros creamos las credenciales de acceso en una tabla llamada **mqtt\_user** por lo que si decidimos cambiar el nombre de esta tabla es importante acceder al archivo **emqx\_auth\_musql.conf** en el directorio **plugins** y cambiar el nombre de la tabla por el que tienes en tu base de datos en caso de que la hayas cambiado

```

50 ## - %C: common name of client TLS cert
51 ## - %d: subject of client TLS cert
52 ##
53 |auth.mysql.auth_query = select password from mqtt_user where username = '%u' limit 1
54 ## auth.mysql.auth_query = select password_hash as password from mqtt_user where username = '%u' limit 1
55

```



```
80  ## - %c: clientid
81  ## - %C: common name of client TLS cert
82  ## - %d: subject of client TLS cert
83  ##
84  auth.mysql.super_query = select is_superuser from mqtt_user where username = '%u' limit 1
85
```

## 4.9 LISTAS DE CONTROL DE ACCESO (ACL)

Ahora que tenemos nuestra autentificación entre nuestro bróker y una base datos ahora toca crear la lista de control de acceso por lo que en nuestra base de datos crearemos la siguiente tabla llamada mqtt\_acl [11]

```
1  CREATE TABLE `mqtt_acl` (
2    `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
3    `allow` int(1) DEFAULT 1 COMMENT '0: deny, 1: allow',
4    `ipaddr` varchar(60) DEFAULT NULL COMMENT 'IpAddress',
5    `username` varchar(100) DEFAULT NULL COMMENT 'Username',
6    `clientid` varchar(100) DEFAULT NULL COMMENT 'ClientId',
7    `access` int(2) NOT NULL COMMENT '1: subscribe, 2: publish, 3: pubsub',
8    `topic` varchar(100) NOT NULL DEFAULT '' COMMENT 'Topic Filter',
9    PRIMARY KEY (`id`),
10   INDEX (ipaddr),
11   INDEX (username),
12   INDEX (clientid)
13 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
14 |
```

Aquí te explico brevemente que hace cada parte de la consulta

Descripción del campo de la tabla de reglas:

- Permitir: Denegar/deny (0), Permitir/allow (1)
- ipaddr: Establecer dirección IP
- username: Nombre de usuario para conectarse al cliente. Si el valor se establece en \$ all, la regla se aplica a todos los usuarios.
- clientid: ID de cliente del cliente conectado
- access: Operaciones permitidas: suscribirse (1), publicar (2), tanto suscribir como publicar (3)
- Topic: los temas que se controlarán, que pueden usar comodines, y los marcadores de posición se pueden agregar al tema para que coincida con la información del cliente. Por ejemplo, el topic se reemplazará con el ID de cliente del cliente actual al hacer coincidir/%c
- %u: nombre de usuario
- %c : ID de cliente



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Ahora que ya tenemos creada nuestra tabla vamos a crear unas consultas como ejemplo de cómo funciona ACL por ello primero vamos a crear una inserción que no deje a los usuarios suscribirse a los tópicos del sistema

```
1 INSERT INTO mqtt_acl (allow, ipaddr, username, clientid, access, topic) VALUES (0, NULL, '$all', NULL, 1, '$SYS/#');
```

Una vez hecho esta inserción si creamos y nos tratamos de conectar al tópico \$SYS/# nos saldrá un error porque le estamos diciendo a la base de datos que no deje a ningún usuario



The screenshot shows the MQTT.fx interface. At the top, there is an error message: "Subscribe Failed Rejected the \$SYS topic Unexpected QoS, MQTT Broker declined the subscription. Please check ACL configuration." Below this, a "New Subscription" dialog box is open. It contains the following fields:

- Topic:** \$SYS/#
- QoS:** 0 (At most once)
- Color:** #8641D5
- Alias:** (empty)

At the bottom of the dialog are "Cancel" and "Confirm" buttons.

También les dejo el ejemplo de una consulta que permita que todos los clientes con la dirección ip 10.59.1.100 se puedan suscribir a los tópicos del sistema

**INSERT INTO mqtt\_acl (allow, ipaddr, username, clientid, access, topic) VALUES (1, '10.59.1.100', NULL, NULL, 1, '\$SYS/#');**

Otro ejemplo es una consulta que permita que todos los usuarios que tenga el clienteid **mqtx\_b1c830d9** se puedan conectar al tópico **/smarthome/%c/temperatura** por ello primero pondremos la consulta que ponga que ningún usuario se puede suscribirse

```
1 INSERT INTO mqtt_acl (allow, ipaddr, username, clientid, access, topic) VALUES (0, NULL, '$all', NULL, 1, '/smarthome/+/temperature');
2
```

Luego la consulta que nos dice que si se pueden suscribir pero solo los que tenga el cliente id que queremos



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

```
1 INSERT INTO mqtt_acl (allow, ipaddr, username, clientid, access, topic) VALUES (1, NULL, '$all', NULL, 1, '/smarthome/%c/temperature');
```

Creamos y nos suscribimos al tópico y como tenemos ese clienteid nos dejara conectarnos al tópico de manera correcta

New Subscription

\* Topic

/smarthome/mqttx\_b1c830d9/temperature

\* QoS

0 At most once Color #E98A2F

Alias

Cancel Confirm

/smarthome/mqtt... QoS 0

**Nota** recuerda que %c es donde pondrás el número de cliente que tienes para más ejemplos revisar ACL ejemplos [12]

#### 4.10 PHP MyAdmin EN LARAGON

En caso de que no te sientas cómodo trabajando con HeidiSQL puedes instalar php MyAdmin en Laragon por ello nos iremos a <https://www.phpmyadmin.net/> descargaremos el zip y lo descomprimiremos en la siguiente carpeta **laragon\etc\apps** y listo ya tenemos php MyAdmin el Laragon

Since July 2015 all phpMyAdmin releases are cryptographically signed by the releasing developer. You should verify that the signature matches the archive you have downloaded. Verification instructions are placed in our documentation in the [Verifying phpMyAdmin releases](#) chapter.

## phpMyAdmin 5.2.1

Released 2023-02-08, see [release notes](#) for details.

Current version compatible with PHP 7.2 and newer and MySQL/MariaDB 5.5 and newer.

File	Size	Verification
<a href="#">phpMyAdmin-5.2.1-all-languages.zip</a>	14.4 MB	[PGP] [SHA256]
<a href="#">phpMyAdmin-5.2.1-all-languages.tar.gz</a>	12.4 MB	[PGP] [SHA256]
<a href="#">phpMyAdmin-5.2.1-all-languages.tar.xz</a>	7.1 MB	[PGP] [SHA256]
<a href="#">phpMyAdmin-5.2.1-english.tar.gz</a>	8.0 MB	[PGP] [SHA256]
<a href="#">phpMyAdmin-5.2.1-english.tar.xz</a>	5.2 MB	[PGP] [SHA256]
<a href="#">phpMyAdmin-5.2.1-english.zip</a>	10.0 MB	[PGP] [SHA256]
<a href="#">phpMyAdmin-5.2.1-source.tar.xz</a>	18.3 MB	[PGP] [SHA256]

Disco local (C:) > laragon > etc > apps >			
Nombre	Fecha de modificación	Tipo	Tamaño
adminer	20/06/2022 03:21 p. m.	Carpeta de archivos	
laragon	20/06/2022 03:21 p. m.	Carpeta de archivos	
memcached	20/06/2022 03:21 p. m.	Carpeta de archivos	
phpMyAdmin	07/02/2023 03:26 p. m.	Carpeta de archivos	
phpRedisAdmin	20/06/2022 03:21 p. m.	Carpeta de archivos	

Solo queda iniciar sesión y ya tendremos acceso a nuestra base de datos en php MyAdmin



← Servidor: localhost → Base de datos: mqtt

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Rutinas Más

Filtros

Que contengan la palabra:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> mqtt_acl	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8mb4_general_ci	64.0 KB	-
<input type="checkbox"/> mqtt_user	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	1	InnoDB	utf8mb4_general_ci	32.0 KB	-
2 tablas	Número de filas	4	InnoDB	utf8_spanish_ci	96.0 KB	0 B

↑  Seleccionar todo Para los elementos que están marcados: ▾

## 4.11 INSTALACION EN LINUX

De pues de haber levantado todo el bróker con MQTT en Windows ahora tocara hacerlo en Linux para ello lo primero quearemos es descargar putty e instalarlo en nuestro equipo [13]



INDICATE VERSIÓN DE PUTTY QUE TIENES ACTUALMENTE PARA VERIFICAR SI PUEDES USARLO. SI NO PUEDES USARLO, PUEDES DESCARGAR LA ÚLTIMA VERSIÓN DE PUTTY. SI NO PUEDES DESCARGARLA, PUEDES INTENTAR LOS BLOJOS DE PRE-LIBERAJE DE 0.79 O LOS SNAPS DE DESARROLLO.

**Package files**

You probably want one of these. They include versions of all the PuTTY utilities (except the new and slightly experimental Windows pterm).

**Bug:** this installer was built differently to other versions, in a way that causes trouble for upgrades (among other problems) – see the [bug record](#) for details. To avoid upgrade trouble, when moving between 0.78 and other versions, we recommend completely uninstalling the existing version first. You can avoid the need for this (and other problems with this installer) by installing 0.78 with a special command-line invocation like:

```
msiexec.exe /i path\to\putty-64bit-0.78-installer.msi ALLUSERS=1
```

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

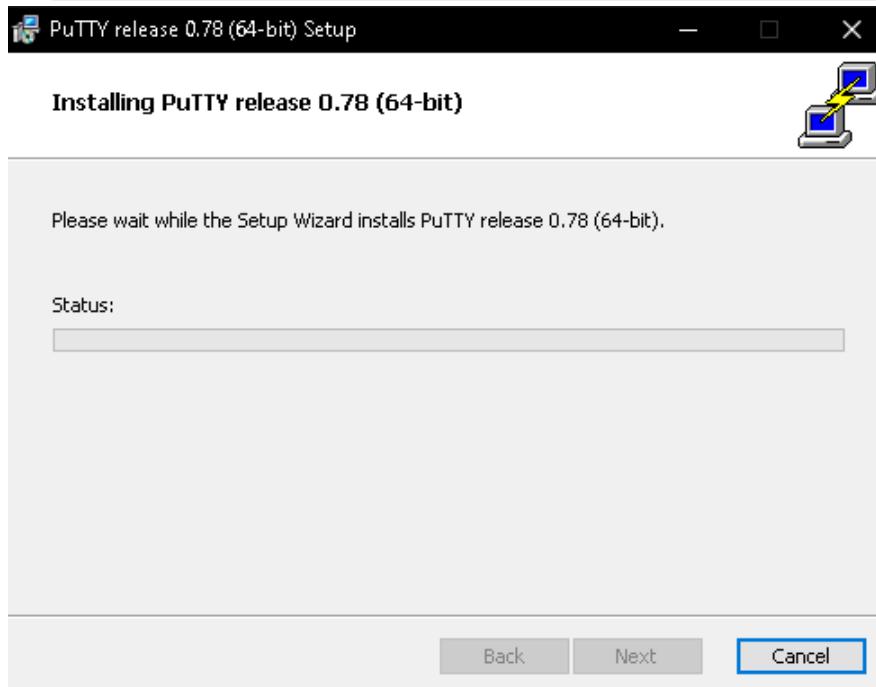
We also publish the latest PuTTY installers for all Windows architectures as a free-of-charge download at the [Microsoft Store](#); they usually take a few days to appear there after we release them.

**MSI ('Windows Installer')**

64-bit x86:	<a href="#">putty-64bit-0.78-installer.msi</a>	(signature)
64-bit Arm:	<a href="#">putty-arm64-0.78-installer.msi</a>	(signature)
32-bit x86:	<a href="#">putty-0.78-installer.msi</a>	(signature)

**Unix source archive**

.tar.gz:	<a href="#">putty-0.78.tar.gz</a>	(signature)
----------	-----------------------------------	-------------



Después de haber instalado putty nos iremos a la página de MQTT.io y ahí seleccionaremos la versión 4 del bróker EMQX pero la instalación que dice package y no la que dice apt



**Version**

v5.0.26 ▼

[Release Notes](#) | [All Versions](#)

**Package Type**

Ubuntu22.04 amd64 / deb ▼

**1 Download emqx-5.0.26-ubuntu22.04-amd64.deb** SHA256

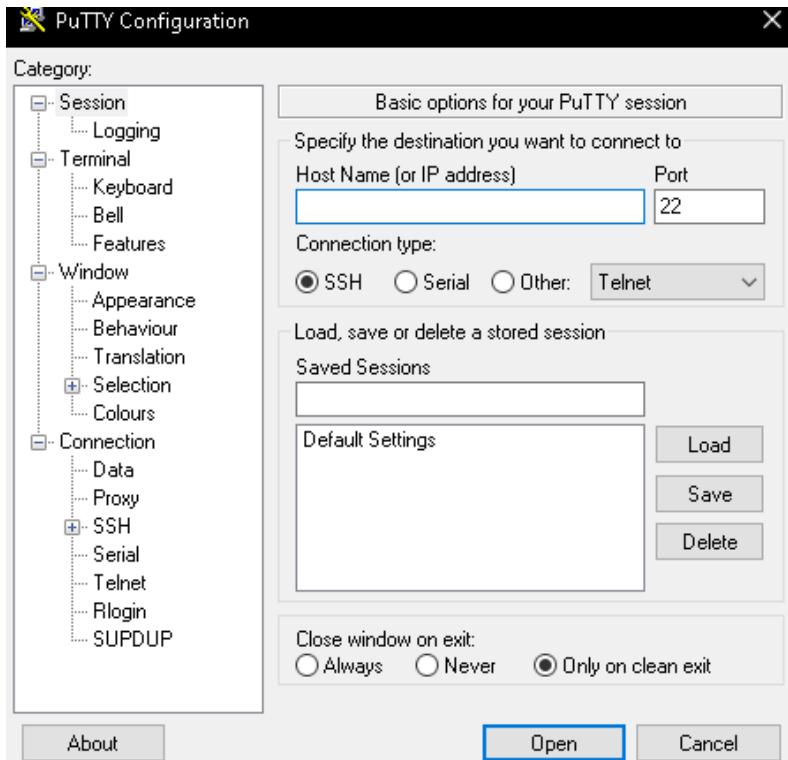
```
wget https://www.emqx.com/en/downloads/broker/5.0.26/emqx-5.0.26-ubuntu22.04-amd64.deb
```

**2 Install EMQX**

```
sudo apt install ./emqx-5.0.26-ubuntu22.04-amd64.deb
```

**3 Run EMQX**

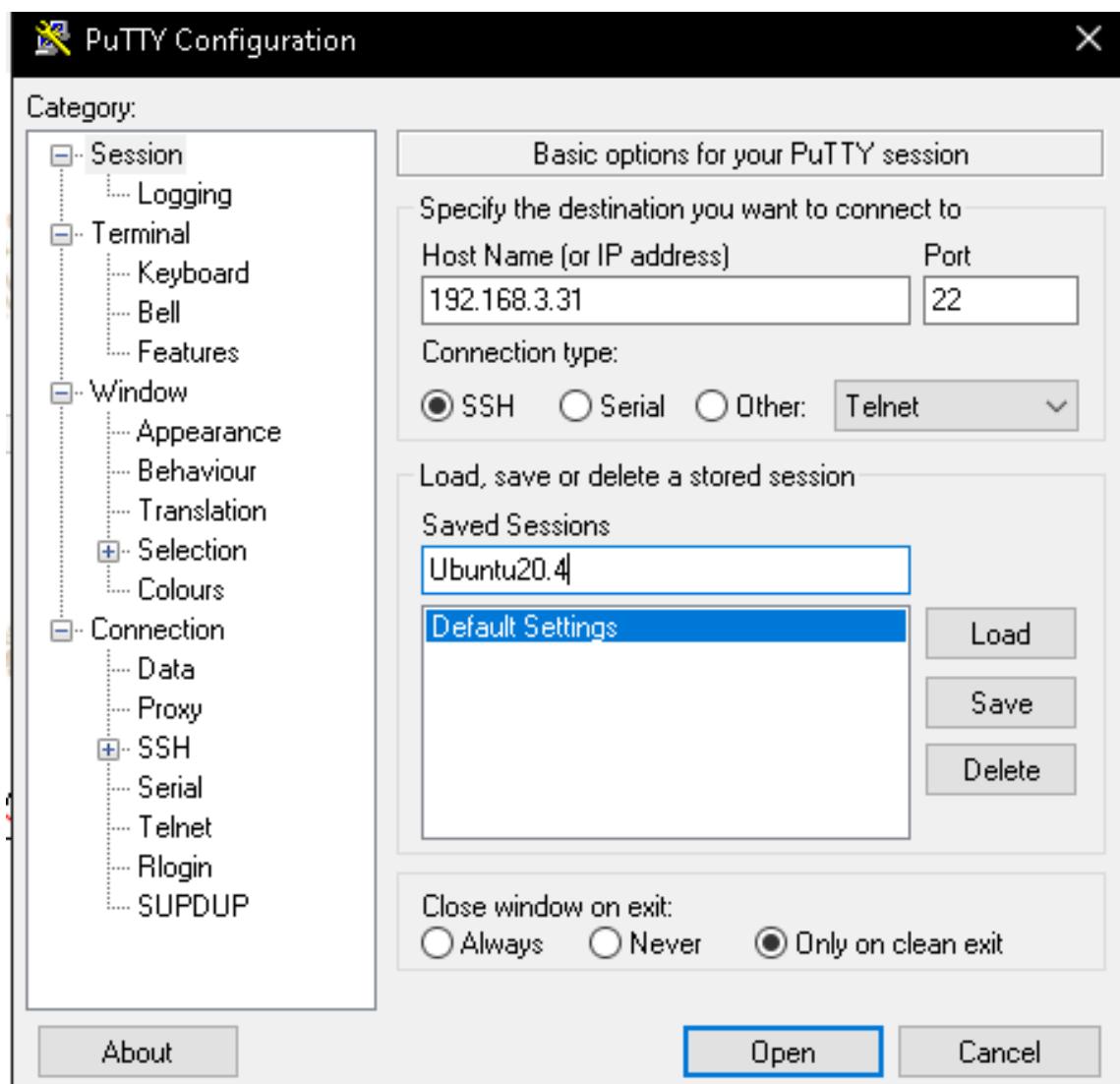
Una vez tenido seleccionado la versión que deseamos abriremos nuestro putty



En nuestro putty introduciremos los datos necesarios para conectarnos al sistema Linux donde deseamos levantar el bróker EMQX como lo es el nombre de la conexión y la dirección ip del



servidor Linux y el puerto a conectarse en este caso usaremos el que viene por defecto que es el puerto 22





Una vez ya se establezca la conectividad procederemos a ingresar las credenciales de entrada y nos conectaremos son súper usuario y una vez hecho esto ya estaremos dentro de nuestro sistema Linux

```
i0t0st@iotmqtt: ~

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of dom 21 ago 2022 00:33:06 UTC

System load: 0.0          Processes: 225
Usage of /: 45.0% of 9.75GB Users logged in: 1
Memory usage: 11%          IPv4 address for ens33: 192.168.52.131
Swap usage: 0%

* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroKOs to make it the smallest full KOs around.

https://ubuntu.com/blog/microk0s-memory-optimisation

0 updates can be applied immediately.

*** System restart required ***

System information as of dom 21 ago 2022 00:33:06 UTC

System load: 0.0          Processes: 225
Usage of /: 45.0% of 9.75GB Users logged in: 1
Memory usage: 11%          IPv4 address for ens33: 192.168.52.131
Swap usage: 0%

* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroKOs to make it the smallest full KOs around.

https://ubuntu.com/blog/microk0s-memory-optimisation

0 updates can be applied immediately.

*** System restart required ***
Last login: Sun Aug 21 00:17:20 2022
i0t0st@iotmqtt:~$ sudo su

(sudo) password for i0t0st:
Sorry, try again.
(sudo) password for i0t0st:
root@iotmqtt:/home/i0t0st# cd
root@iotmqtt:~#
```

Una vez situados en la carpeta raíz de nuestro sistema regresaremos la página de emqx.io y

coparemos el comando que necesitamos para instalar el broker en emqx y lo pegaremos en la terminal de Putty y se empezará a instalar

v4.4.18 Release Notes | All Versions

**Package Type**

Ubuntu22.04 amd64 / deb

**1 Download emqx-4.4.18-otp24.3.4.2-1-ubuntu22.04-amd64.deb** SHA256

```
wget https://bilbo.emqx.com/en/downloads/broker/4.4.18/emqx-4.4.18-otp24.3.4.2-1-ubuntu22.04-amd64.deb
```

**2 Install EMQX**

```
sudo apt install ./emqx-4.4.18-otp24.3.4.2-1-ubuntu22.04-amd64.deb
```

**3 Run EMQX**

```
sudo systemctl start emqx
```

**Terminal Screenshot:**

```
root@iotmqtt:/# ls
bin dev home lib32 libx32 media opt root sbin srv sys usr
boot etc lib lib64 lost+found mnt proc run snap swap.img tmp var
root@iotmqtt:/# wget https://www.emqx.com/en/downloads/broker/4.4.7/emqx-4.4.7-otp24.1.5-3-ubuntu20.04-amd64.zip
--2022-08-21 00:35:19-- https://www.emqx.com/en/downloads/broker/4.4.7/emqx-4.4.7-otp24.1.5-3-ubuntu20.04-amd64.zip
Resolving www.emqx.com (www.emqx.com)... 52.53.184.178
Connecting to www.emqx.com (www.emqx.com)|52.53.184.178|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://packages.emqx.io/emqx-ce/v4.4.7/emqx-4.4.7-otp24.1.5-3-ubuntu20.04-amd64.zip [following]
--2022-08-21 00:35:20-- https://packages.emqx.io/emqx-ce/v4.4.7/emqx-4.4.7-otp24.1.5-3-ubuntu20.04-amd64.zip
Resolving packages.emqx.io (packages.emqx.io)... 13.227.97.121, 13.227.97.126, 13.227.97.10, ...
Connecting to packages.emqx.io (packages.emqx.io)|13.227.97.121|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 48318800 (46M) [application/zip]
Saving to: 'emqx-4.4.7-otp24.1.5-3-ubuntu20.04-amd64.zip'

emqx-4.4.7-otp24.1.5-3 8%[==>] 4,12M 2,62MB/s
```

Una vez ya se haya descargado en nuestro Linux procederemos a descomprimirlo por ello utilizaremos el siguiente comando:

**unzip emqx-4.4.18-otp24.3.4.2-1-ubuntu22.04-amd64.zip**



```
root@iotmqt: /  
inflating: emqx/etc/plugins/emqx_lwm2m.conf  
inflating: emqx/etc/plugins/emqx_auth_mongo.conf  
inflating: emqx/etc/plugins/acl.conf.paho  
inflating: emqx/etc/plugins/emqx_retailer.conf  
inflating: emqx/etc/plugins/emqx_auth_redis.conf  
inflating: emqx/etc/plugins/emqx_auth_mnesia.conf  
inflating: emqx/etc/plugins/emqx_auth_http.conf  
inflating: emqx/etc/plugins/emqx_auth_jwt.conf  
inflating: emqx/etc/plugins/emqx_management.conf  
inflating: emqx/etc/plugins/emqx_stomp.conf  
inflating: emqx/etc/plugins/emqx_coap.conf  
inflating: emqx/etc/plugins/emqx_telemetry.conf  
inflating: emqx/etc/plugins/emqx_dashboard.conf  
inflating: emqx/etc/plugins/emqx_sn.conf  
inflating: emqx/etc/plugins/emqx_bridge_mqtt.conf  
inflating: emqx/etc/plugins/emqx_psk_file.conf  
inflating: emqx/etc/plugins/emqx_recon.conf  
inflating: emqx/etc/plugins/emqx_auth_ldap.conf  
inflating: emqx/etc/plugins/emqx_prometheus.conf  
inflating: emqx/etc/plugins/emqx_rule_engine.conf  
inflating: emqx/etc/plugins/emqx_exhook.conf  
inflating: emqx/etc/plugins/emqx_web_hook.conf  
inflating: emqx/etc/plugins/emqx_auth_pgsql.conf  
creating: emqx/releases/  
creating: emqx/releases/4.4.7/  
inflating: emqx/releases/4.4.7/start_clean.boot  
inflating: emqx/releases/4.4.7/start.boot  
inflating: emqx/releases/4.4.7/relup  
inflating: emqx/releases/4.4.7/no_dot_erlang.boot  
inflating: emqx/releases/4.4.7/emqx.rel  
extracting: emqx/releases/4.4.7/BUILT_ON  
inflating: emqx/releases/4.4.7/start.script  
inflating: emqx/releases/4.4.7/emqx.schema  
inflating: emqx/releases/emqx.rel  
extracting: emqx/releases/start_erl.data  
inflating: emqx/releases/RELEASES  
inflating: emqx/releases/emqx_vars  
root@iotmqt: #
```

**Nota:** Si te sale un error es porque no tienes el programa para descomprimir este archivo por lo que para instalarlo pondrás el comando **apt install unzip** y se descargara el programa para descomprimir



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez ya esté descomprimido el archivo procederemos a ejecutar el broker por ello ingresamos el siguiente comando para ponerlo en marcha **./emqx/bin/emqx start**

```
root@iotmqtt: /  
inflating: emqx/etc/plugins/emqx_web_hook.conf  
inflating: emqx/etc/plugins/emqx_auth_pgsql.conf  
creating: emqx/releases/  
creating: emqx/releases/4.4.7/  
inflating: emqx/releases/4.4.7/start_clean.boot  
inflating: emqx/releases/4.4.7/start.boot  
inflating: emqx/releases/4.4.7/relup  
inflating: emqx/releases/4.4.7/no_dot_erlang.boot  
inflating: emqx/releases/4.4.7/emqx.rel  
extracting: emqx/releases/4.4.7/BUILT_ON  
inflating: emqx/releases/4.4.7/start.script  
inflating: emqx/releases/4.4.7/emqx.schema  
inflating: emqx/releases/emqx.rel  
extracting: emqx/releases/start_erl.data  
inflating: emqx/releases/RELEASES  
inflating: emqx/releases/emqx_vars  
root@iotmqtt:/# ls  
bin  emqx          home   lib64      media    proc    sbin  swap.img  usr  
boot emqx-4.4.7-otp24.1.5-3-ubuntu20.04-amd64.zip lib     libx32    mnt    root    snap  sys   var  
dev  etc           lib32  lost+found opt     run    srv   tmp  
root@iotmqtt:/# ./emqx/bin/emqx console  
Erlang/OTP 24 [erts-12.1.5] [emqx] [64-bit] [smp:2:2] [ds:2:2:8] [async-threads:4] [jit]  
Starting emqx on node emqx@127.0.0.1  
Start mqtt:tcp:internal listener on 127.0.0.1:11883 successfully.  
Start mqtt:tcp:external listener on 0.0.0.0:1883 successfully.  
Start mqtt:ws:external listener on 0.0.0.0:8083 successfully.  
Start mqtt:ssl:external listener on 0.0.0.0:8883 successfully.  
Start mqtt:wss:external listener on 0.0.0.0:8084 successfully.  
Start http:management listener on 8081 successfully.  
2022-08-21T00:37:53.445525+00:00 [warning] [Dashboard] Using default password for dashboard 'admin' user. Please use './bin/emqx_ctl admins' command to change it. NOTE: the default password in config file is only used to initialise the database record, changing the config file after database is initialised has no effect.  
Start http:dashboard listener on 18083 successfully.  
EMQ X Broker 4.4.7 is running now!  
Eshell V12.1.5 (abort with ^G)  
/emqx@127.0.0.1>
```

Ahora si nuestra en otra máquina colocamos la siguiente dirección **192.168.52.131:18083** y nos llevará al dashboard ya funcionando



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

⚠ No seguro | 192.168.52.131:18083/#/login?redirect=%2F

The screenshot shows a login form with the title 'LOG IN' at the top. It has two input fields: 'Username' and 'Password', both with placeholder text 'I'. Below the password field is a checkbox labeled 'Remember'. To the right of the 'Remember' checkbox is a green 'Log In' button.

**Nota:** Recuerda que en este caso 192.168.52.131 es la dirección de mi dispositivo Linux por lo que ahí tú tendrás que poner la dirección que te salga en tu servidor

Una vez hecho esto tocaría generar los certificados de seguridad. CLA, etc. como lo vimos en Windows pero como los que generamos en Windows nos funciona ahora solo tocara pasarlo a nuestro equipo Linux por ello descargamos Cyberduck en <https://cyberduck.io/>



Una vez descargado abrimos nuestro cliente MQTTX y probaremos la conexión a nuestro dashboard configurando el host a nuestro servidor Linux



Protocol	Listen On	Max Connections	Current Connections
mqtt:ssl	8883	102400	0
<b>mqtt:tcp</b>	<b>0.0.0.1:1883</b>	<b>102400</b>	<b>1</b>
mqtt:tcp	127.0.0.1:1883	102400	0
http:dashboard	18083	512	1
http:management	8081	512	0
mqtt:ws:8083	8083	102400	0
mqtt:wss:8084	8084	102400	0

Una vez veamos que si se está haciendo la conexión correctamente procederemos a abrir el Cyberduck y configurarlo para que pueda copiar los certificados SSL a nuestro Linux

Conexión nueva

Servidor: 192.168.52.131 Puerto: 22

Nombre de usuario: iosthost

Contraseña: \*\*\*\*\*

Clave Privada SSH: Ninguno

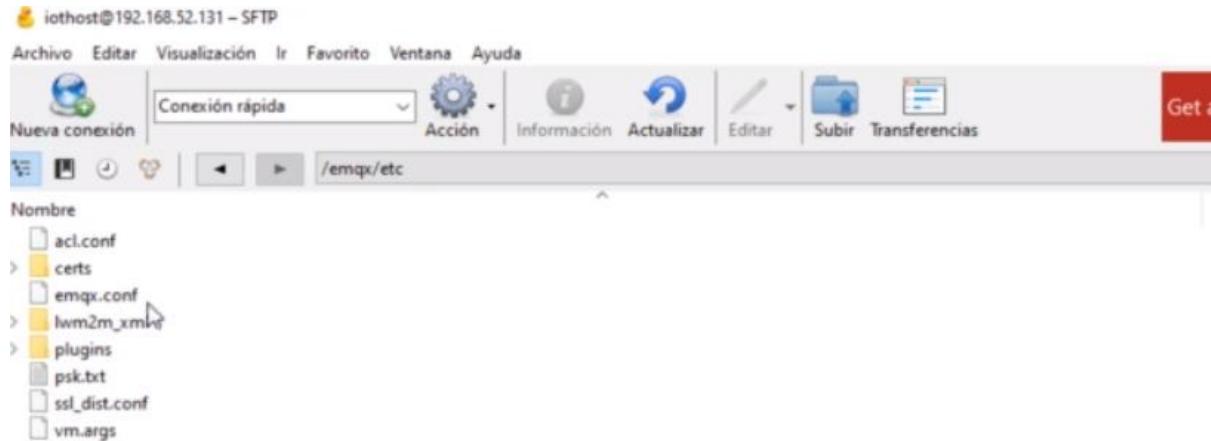
Guardar contraseña

Conectar Cancelar

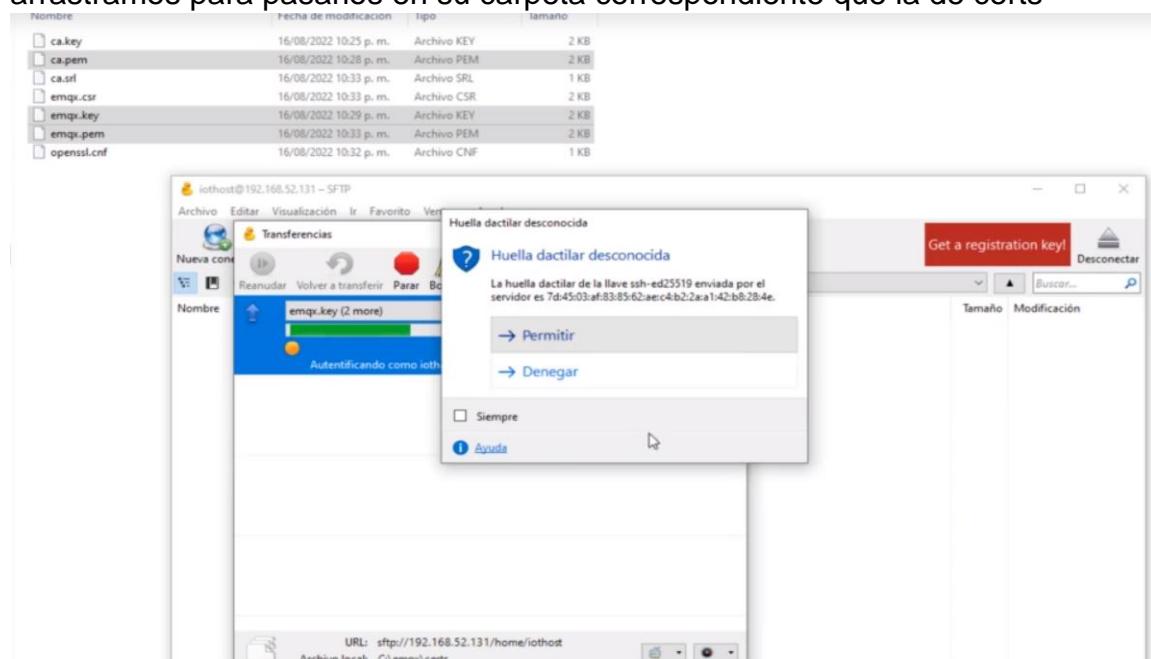
## 1. Seleccionamos la conexión SFTP

2. Después en servidor agregaremos la dirección de nuestro equipo Linux
3. Agregaremos las credenciales de nuestro equipo Linux Usuario y Contraseña
4. Seleccionamos el puerto y le damos en conectar

Una vez se haya conectado correctamente ya podremos navegar y transferir archivos de nuestro Windows a nuestro Linux



Ahora solo buscaremos los certificados que deseamos pasar de Windows a Linux y los arrastramos para pasarlo en su carpeta correspondiente que es de certs





```
root@iotmqtt: ~ cd /home/iothost
root@iotmqtt:~/home/iothost# ls
ca.pem emqx.key emqx.pem
root@iotmqtt:/home/iothost# cp ca.pem /emqx/etc/certs
cp: cannot create regular file '/emqx/etc/certs': No such file or directory
root@iotmqtt:/home/iothost# cp ca.pem /emqx/etc/certs
root@iotmqtt:/home/iothost# cp ca.pem /emqx/etc/certs
```

**Nota:** Es importante haberse logueado con credenciales de súper usuario en el Cyberduck porque de lo contrario no nos dejará pasar los archivos a Linux

Finalmente solo queda pasar los archivos de certificados y realizar las mismas configuraciones que hicimos en Windows (Apartados de configuración de certificados autoformados página 32) pero ahora en Linux y ya tendremos el bróker EMQX funcionando en nuestro servidor Linux con la base de datos

**Nota:** Podemos montar el bróker EMQX V5 en un servidor Linux en caso de necesitarlo

## 5 Sección 6: EMQX MODO DE PRODUCCION (NUBE DE ORACLE)

### 5.1 ¿Qué es VPS?

Es un servidor virtual privado, término que proviene del inglés virtual private server, es una modalidad de alojamiento web donde un único servidor físico es particionado virtualmente para brindar múltiples servidores virtuales

### 5.2 Producción VPS de Oracle

Lo primero que tenemos que hacer es ir a la siguiente página <https://www.oracle.com/mx/cloud/> y una vez ahí le daremos en iniciar prueba gratuita y veremos los distintos planes y modalidades de uso pero en nuestro caso seguiremos en el apartado de empezar de manera gratuita



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

OCI

Acerca de Servicios Soluciones Precios Partners Recursos



Iniciar sesión en Oracle Cloud

## Oracle Cloud Infrastructure (OCI)

La nube de última generación diseñada para ejecutar cualquier aplicación de forma más rápida y segura, por un menor costo.

Presentamos la región de nube más reciente de Oracle en Chicago, Illinois. La cuarta región de nube pública de Oracle en EE. UU. lleva las últimas innovaciones en la nube a empresas de toda Norteamérica.

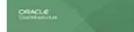
Prueba gratuita de OCI

Lee el comunicado de prensa



### 8 preguntas para plantearle a tu proveedor de nube

Habilite su empresa para la transformación digital.



### Multinube con OCI y Azure

Oracle Database Service for Microsoft Azure brinda a los clientes de Azure el



### OCI, reconocida como solución

Habla con un especialista

Contactar o llamar



## Oracle Cloud (modo gratuito)

Crea, comprueba e implementa aplicaciones en Oracle Cloud de forma gratuita.

Se han agregado nuevos servicios de uso libre, incluidos los recursos informáticos de Arm Ampere A1. Para proyectos de desarrollo de Arm a gran escala, puedes solicitar [OCI Arm Accelerator](#).

Comienza gratis

Inicia sesión en Oracle Cloud



Servicios de uso libre de Oracle Cloud Infrastructure (0:38)



Una vez ahí se nos abrirá una ventana en la que tendremos que llenar un formulario para poder crear nuestra cuenta y poder usar los servicios de Oracle

Cuenta gratuita de Oracle Cloud

Empiece a disfrutar de:

Acceso gratuito permanente a servicios esenciales, incluyendo:

- Base de datos autónoma
- Almacenamiento de objetos

Además, 300 \$ en créditos que podrá usar durante 30 días en servicios adicionales:

- Container Engine for Kubernetes
- Analytics Cloud
- Integración de datos

Información de la cuenta

País/Territorio

Nombre

Apellidos

Correo electrónico

Soy humano  hCaptcha Privacidad | Condiciones

Verificar mi correo electrónico

Siguiente

Una vez creada la cuenta en Oracle tendremos que iniciar sesión y acceder al panel principal

OCI

Services Solutions Why OCI Customers Pricing Learn Developers Support Marketplace

Sign in to Oracle Cloud

ORACLE Cloud

Cloud Account Name

infohost54

Next

Forgot your cloud account name? Get help

Do you have a Traditional Cloud Account? Sign in

Not an Oracle Cloud customer yet?

Sign Up

Launch a pre-built automated CI/CD pipeline for Kubernetes

Simplify and automate your software development cycle with the fully-managed OCI DevOps service.

Deploy now View architecture

Red Bull Racing Workshop

Learn how to make predictions with analytics and machine learning models using real race data.

Start workshop



Your Free Trial ended. You can continue to use Always Free resources. Unless you [upgrade](#), other resources are removed, which can result in data loss.

ORACLE Cloud Search resources, services, documentation, and Marketplace Brazil East (Sao Paulo) 150% Restablecer X

DESARROLLO DE APLICACIONES

Desplegar una zona de llegada base

7-9 minutos

BASE DE DATOS

Visualizar y analizar datos de Strava en Autonomous Database

2-4 minutos Sempre gratis elegible

OCI Search Service with OpenSearch 15 ago 2022

New Languages Supported for Speech 12 ago 2022

New metrics added for GoldenGate service 12 ago 2022

Java Management 5.0 ya está disponible 11 ago 2022

Integration 3 11 ago 2022

Ver notas de versión...

▼ Launch resources

RECURSOS INFORMÁTICOS

Crear una instancia de VM

2-6 minutos Sempre gratis elegible

PROCESAMIENTO DE TRANSACCIONES AUTÓNOMO

Crear una base de datos ATP

3-5 minutos Sempre gratis elegible

ALMACÉN DE DATOS AUTÓNOMO

Crear una base de datos ADW

RED

Configurar una red con un asistente

View all resources

Ahora crearemos una instancia de máquina virtual por ello accederemos al apartado de Crear una instancia VM

▼ Launch resources

RECURSOS INFORMÁTICOS

Crear una instancia de VM

2-6 minutos Sempre gratis elegible



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

ORACLE Cloud

iothost84

Conexión a la cuenta de Oracle Cloud

Dominio ⓘ  
Default

Nombre de usuario  
iothost84@gmail.com

Contraseña:  
\*\*\*\*\*

Forgot Password?

Conectar

¿Necesita ayuda para conectarse?

Una vez ahí pondremos el nombre de la instancia que estamos creando y dejamos lo demás como esta exceptuando **Imagen y unidad** en la cual le daremos en editar y seleccionar la imagen Linux que deseemos en mi caso será la de Ubuntu versión 22.04

## Crear instancia informática

Cree una instancia para desplegar y ejecutar aplicaciones, o bien guarde como una pila de Terraform reutilizable para crear una instancia con el gestor de recursos.

Nombre

Crear en compartimento

### Ubicación

[Editar](#)

Dominio de disponibilidad: AD-1 Siempre gratis elegible

Tipo de capacidad: Capacidad bajo demanda

Dominio de errores: Permitir a Oracle elegir el mejor dominio de errores

### Imagen y unidad

[Editar](#)

Imagen: Oracle Linux 8

Unidad: VM.Standard.E2.1.Micro Siempre gratis elegible

Compilación de imagen: 2022.08.08-0

Recuento de OCPU: 1

Memoria (GB): 1

Ancho de banda de red (Gbps): 0.48

ORACLE Cloud
Search resources, services, documentation, and Marketplace
Brazil East (Sao Paulo)

### Crear instancia informática

**Imagen y unidad**

Una **unidad** es una plantilla que determina el número de CPU, la cantidad de memoria y el ancho de banda de red que se ejecuta sobre la unidad.

Imagen

	Oracle Linux 8	Compilación de imagen: 2022.08.08-0
--	----------------	-------------------------------------

Shape

	VM.Standard.E2.1.Micro <span>Siempre gratis elegible</span>	Máquina virtual, 1 core OCPU, 1 GB memory, 0.48 Gbps
--	---	--

[Mostrar opciones avanzadas](#)

**Examinar todas las imágenes**

Compartimento

Las [imágenes de plataforma](#) son sistemas operativos preincorporados para Oracle Cloud Infrastructure.

Nombre de Imagen	Versión del Sistema Operativo	Compilación de imagen	
Canonical Ubuntu <span>Siempre gratis elegible</span>	22.04	2022.08.10-0	<a href="#">Opciones Avanzadas</a>
CentOS <span>Siempre gratis elegible</span>	8 Stream	2022.08.18-0	<a href="#">Opciones Avanzadas</a>
Oracle Autonomous Linux <span>Siempre gratis elegible</span>	7.9		<a href="#">Opciones Avanzadas</a>
Oracle Linux <span>Siempre gratis elegible</span>	9	2022.08.17-0	<a href="#">Opciones Avanzadas</a>
Oracle Linux Cloud Developer <span>Siempre gratis elegible</span>	8	2022.05.22-0	<a href="#">Opciones Avanzadas</a>
Windows	Server 2019 Standard		<a href="#">Opciones Avanzadas</a>

1 seleccionado(s)

Mostrando 6 elementos

En el apartado de red podemos editar la red y sub red virtual en la nube para ello solo tendremos que darle en editar y poner el nombre de la red que queremos

## Red

[Reducir](#)

La [red](#) permite conectar su instancia a Internet y a otros recursos en la consola. Para asegurarse de que puede [conectar su instancia](#), asignele una dirección IP pública.

### Red principal

- Seleccionar red virtual existente en la nube
- Crear nueva red virtual en la nube
- Introducir OCID de subred

### Red virtual en la nube en [localhost84 \(raíz\)](#) ([Cambiar compartimiento](#))

vcn-iot

### Subred

- Seleccionar subred existente
- Crear nueva subred pública

### Subred en [localhost84 \(raíz\)](#) [\(Cambiar compartimiento\)](#)

subnet-iot (regional)

### Dirección IP pública

- Asignar una dirección IPv4 pública
- No asignar una dirección IPv4 pública



Al asignar una dirección IP pública, se puede acceder a esta instancia desde Internet. Si no está seguro sobre si necesita una dirección IP pública, siempre puede asignar una más tarde.

En el apartado de claves SSH tendremos que generar claves para poder conectarnos a la instancias por ello le daremos en **Generar una par de claves para mí** y luego le daremos clic en guardar clave privada y al resto de apartados los dejamos como están y le damos en crear

## Agregar claves SSH

Genere un [par de claves SSH](#) para conectarse a la instancia mediante la conexión de shell seguro (SSH), o bien cargue una clave pública con la que ya cuente.

- Generar un par de claves para mí
- Cargar archivos de clave pública (.pub)
- Pegar claves públicas
- No se han encontrado claves SSH

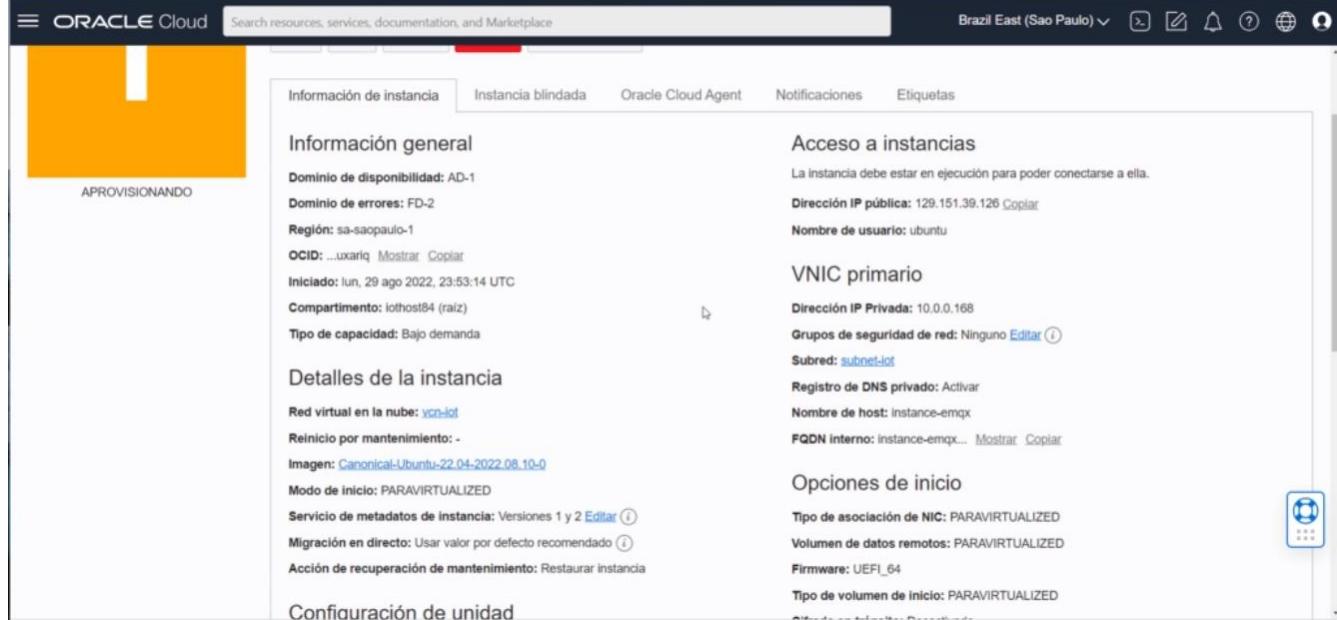


Descargue la clave privada para poder conectarse a la instancia mediante SSH. No se mostrará de nuevo.

[Guardar clave privada](#)

[Guardar clave pública](#)

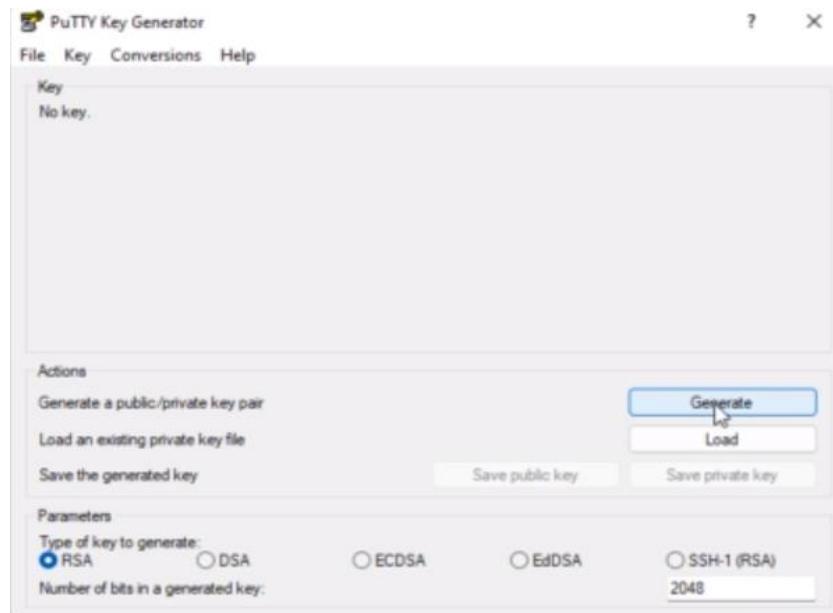
Si hicimos todo correctamente nos saldrá nuestra instancia creada y nos arrojara nuestra dirección IP pública



The screenshot shows the Oracle Cloud interface for an instance named 'instancia-emqx'. The status is 'APROVISIONANDO' (Provisioning). The 'Información general' section includes details like Region: sa-saopaulo-1, OCID: ..., and Launch Time: lun, 29 ago 2022, 23:53:14 UTC. The 'Acceso a instancias' section shows the Public IP: 129.151.39.126 and the Primary VNIC information. The 'Detalles de la instancia' section lists network, boot, and maintenance details. The 'Configuración de unidad' section is partially visible at the bottom.

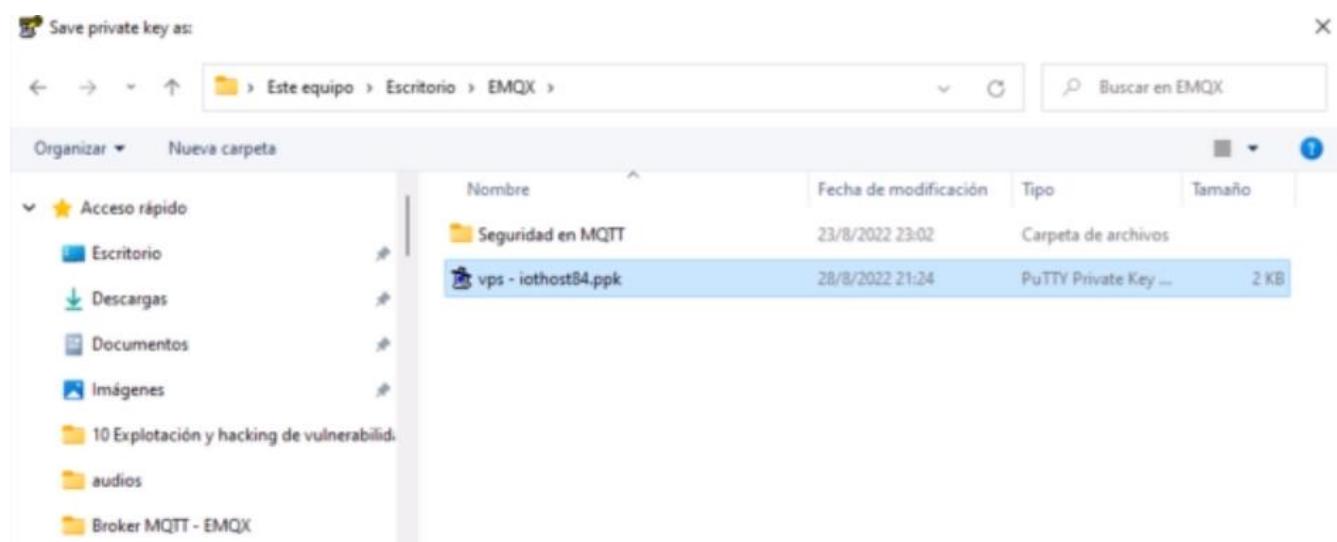
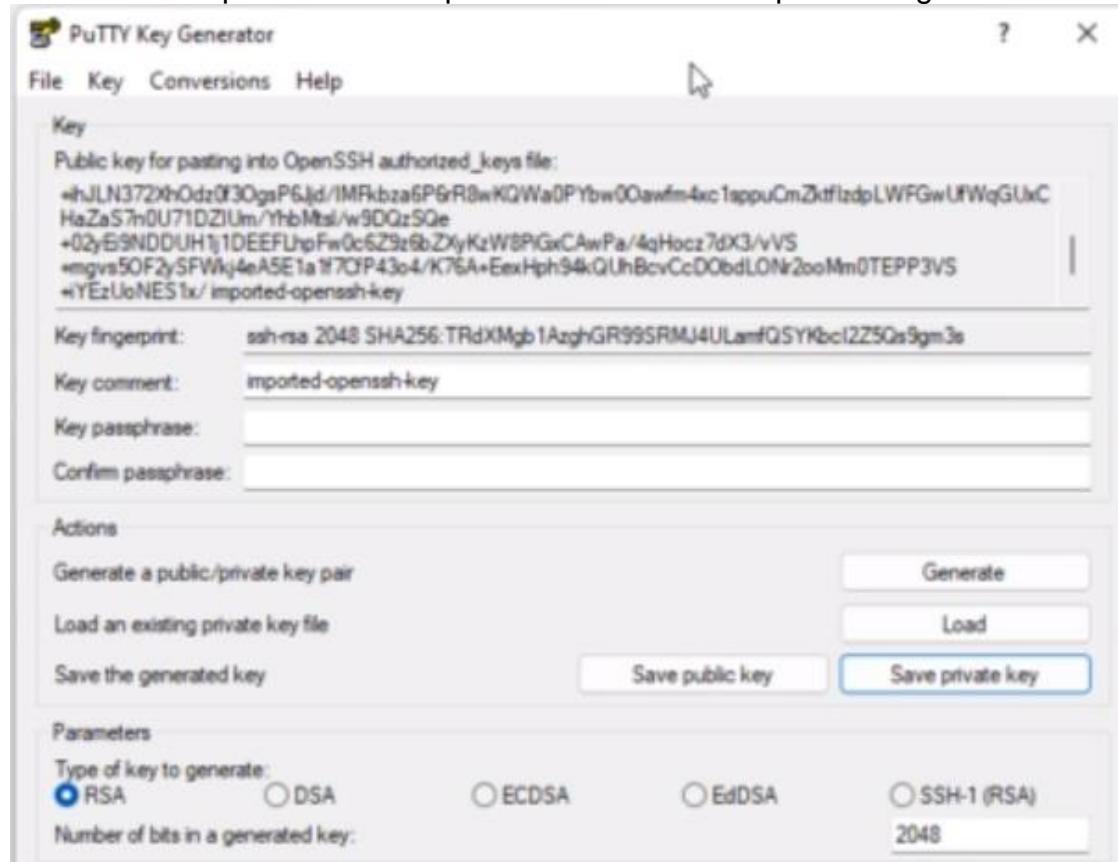
### 5.3 Iniciar sesión con Putty en la instancia

Una vez creada nuestra instancia en Oracle abriremos **Puttygen** la cual nos abrirá una ventana como esta



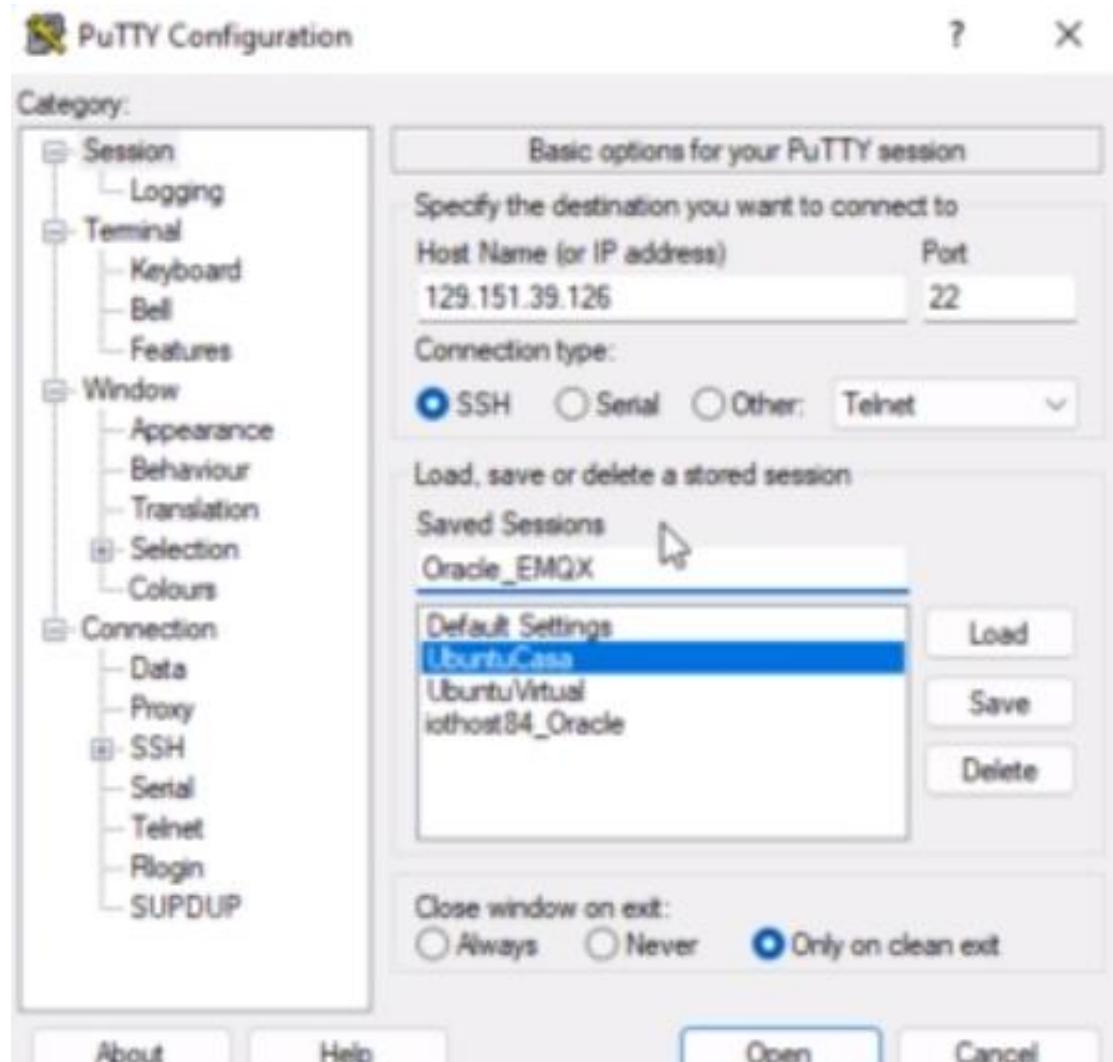


Una vez ahí le daremos clic en el apartado de load y buscaremos el archivo SSH que se nos descargó anteriormente y una vez cargado le daremos en Save private Key y nos saldrá una ventana en la que tendremos que seleccionar donde queremos guardar este key



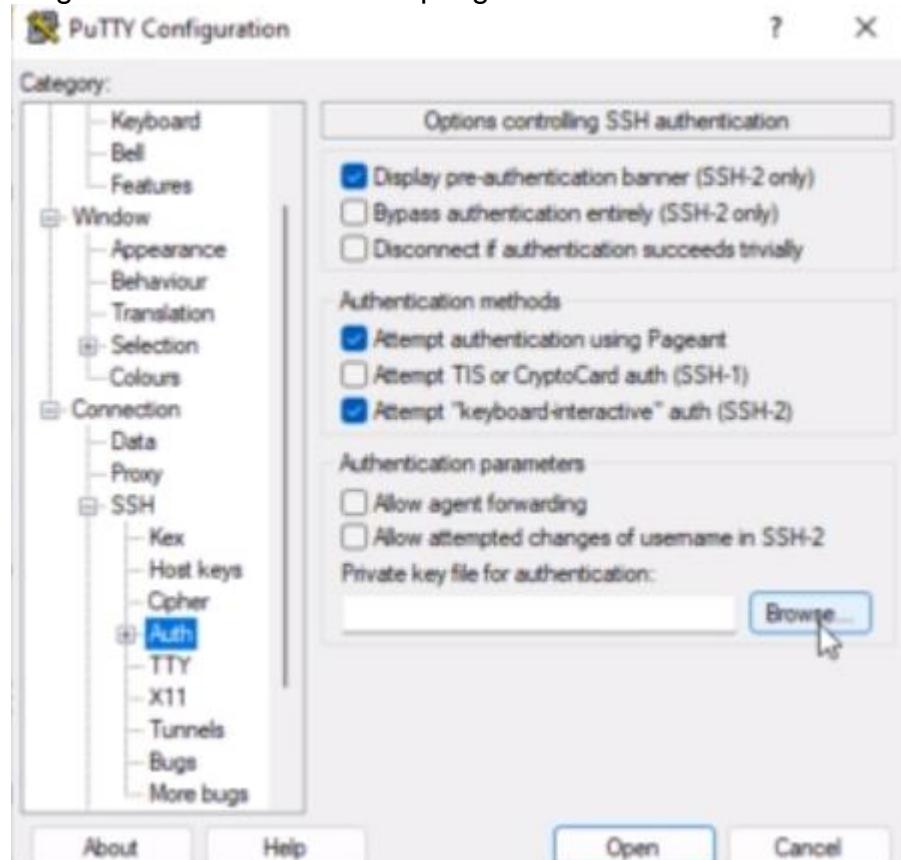


Ahora abriremos **PuTTY** y ahí colocaremos los datos correspondientes para conectarnos a nuestra instancia como lo es la dirección pública que nos generó Oracle y el nombre con el cual queremos guardar la conexión en mi caso será Oracle\_EMQX y seleccionaremos el tipo de conexión que es SSH



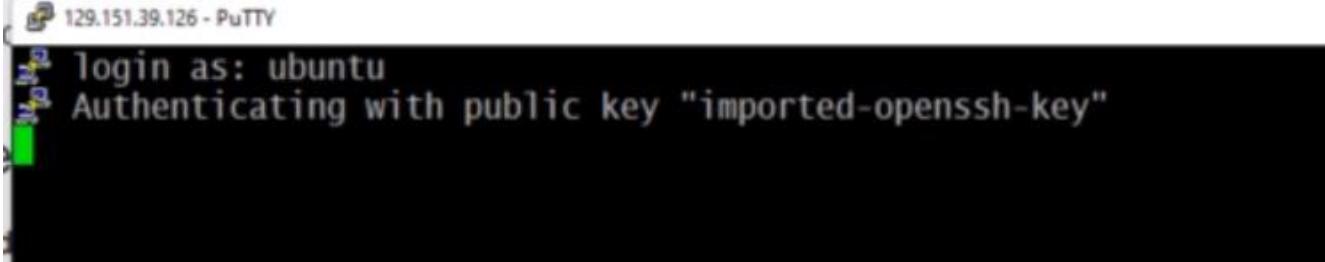


Y una vez hecho esto iremos a la izquierda donde tenemos todas las categorías y ahí buscaremos la que dice SSH y ahí en la subcategoría de Auth le daremos en browse y cargaremos el archivo SSH que guardamos con anterioridad

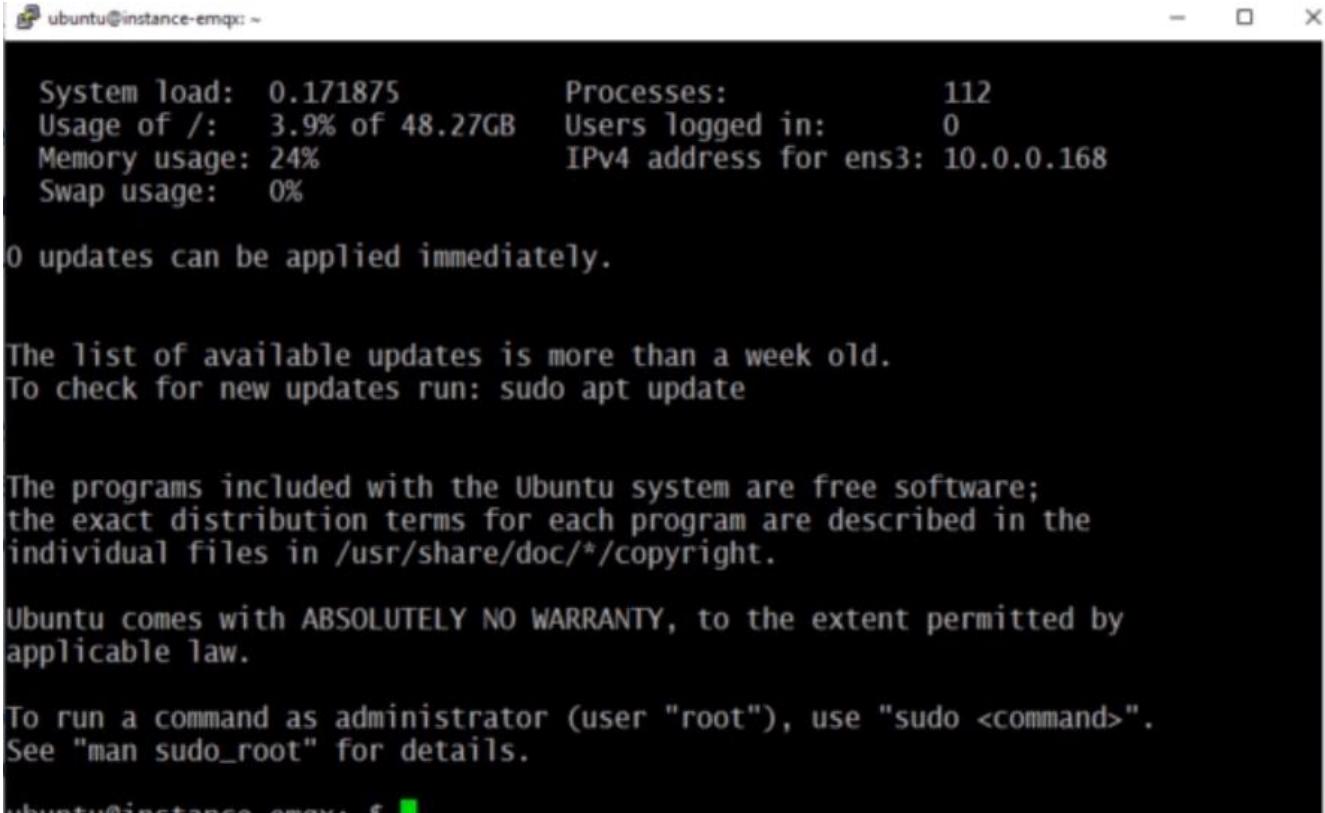


Nombre	Fecha de modificación	Tipo	Tamaño
Seguridad en MQTT	23/8/2022 23:02	Carpeta de archivos	
vps - iothost84.ppk	29/8/2022 20:00	Putty Private Key ...	2 KB

Una vez cargado el archivo SSH volveremos a nuestra sesión guardaremos la configuración y le daremos en open y se nos abrirá una especie de consola la cual indicara que nos hemos conectado a nuestra instancia de manera correcta ahora solo nos logaremos con el usuario de nuestra instancia el cual Oracle me generó y nombro Ubuntu, una vez iniciado con este usuario cargara la llave key y nos dejara conectar al servidor



129.151.39.126 - PuTTY  
login as: ubuntu  
Authenticating with public key "imported-openssh-key"



```
ubuntu@instance-emqx: ~  
  
System load: 0.171875      Processes:          112  
Usage of /: 3.9% of 48.27GB  Users logged in:      0  
Memory usage: 24%          IPv4 address for ens3: 10.0.0.168  
Swap usage: 0%  
  
0 updates can be applied immediately.  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@instance-emqx:~$
```



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez dentro del servidor pondremos el comando **sudo su** para pasar al modo de súper usuario y ahí pondremos el comando **apt update** para actualizar todas las dependencias y cuando termine pondremos ahora el comando **apt upgrade**

```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.
```

```
ubuntu@instance-emqx:~$ sudo su  
root@instance-emqx:/home/ubuntu# cd ..  
root@instance-emqx:/home# cd ..  
root@instance-emqx:/# cd ..  
root@instance-emqx:/# apt update
```

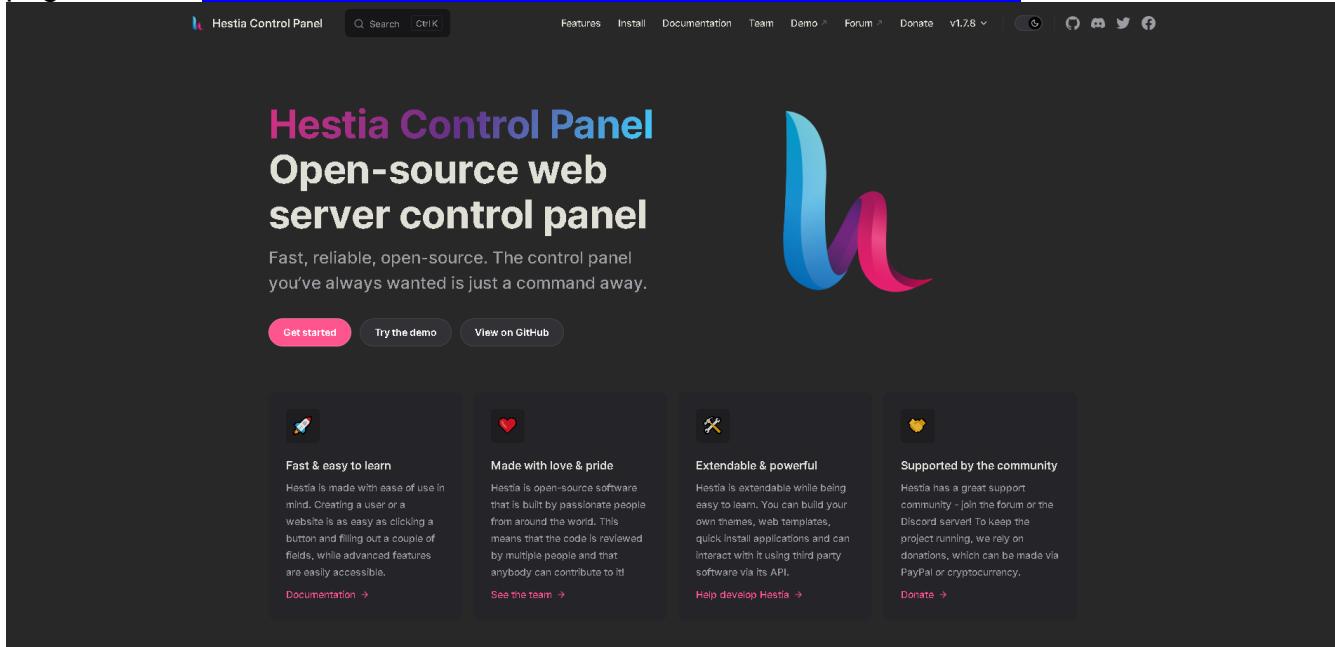
```
Building dependency tree... Done  
Reading state information... Done  
18 packages can be upgraded. Run 'apt list --upgradable' to see them.  
root@instance-emqx:/# apt upgrade
```

Después de esto nos saldrá una ventana color rosa en la cual solo le daremos enter y listo ya tendríamos el sistema instalado correctamente



## 5.4 Instalar Hestia Panel [14]

Lo que vamos a hacer ahora es instalar un panel administrativo para Linux gratis en el cual nos permitirá gestionar bases de datos, usuarios, cortafuegos etc. Para instalarlo iremos a su página oficial <https://hestiacp.com/docs/introduction/getting-started.html>



Una vez ahí le daremos en get started y de esta página coparemos el siguiente comando en nuestro Putty para que se descargue Hestia  
**wget https://raw.githubusercontent.com/hestiacp/hestiacp/release/install/hst-install.sh**

```
root@instance-emqx:/# wget https://raw.githubusercontent.com/hestiacp/hestiacp/release/install/hst-install.sh
--2022-08-30 00:15:17-- https://raw.githubusercontent.com/hestiacp/hestiacp/release/install/hst-install.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.109.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3240 (3.2K) [text/plain]
Saving to: 'hst-install.sh'

hst-install.sh          100%[=====]  3.16K --.-KB/s   in 0s

2022-08-30 00:15:17 (53.8 MB/s) - 'hst-install.sh' saved [3240/3240]

root@instance-emqx:/#
```



Una vez descargado procederemos a instalarlo con el siguiente comando **bash hst-install.sh -f**

```
root@instance-emqx:/# ls
bin dev home lib lib64 lost+found mnt proc run snap sys usr
boot etc hst-install.sh lib32 libx32 media opt root sbin srv tmp var
root@instance-emqx:/# bash hst-install.sh -f
```

The terminal window shows the following sequence:

- Initial directory listing: `ls` command shows standard Linux directories like bin, dev, home, lib, lib64, lost+found, mnt, proc, run, snap, sys, and usr.
- Running the installer: `bash hst-install.sh -f` command is entered.
- Logo: A large, stylized logo consisting of a grid of squares and lines forming a complex geometric pattern.
- Panel title: "Hestia Control Panel" followed by the version "1.6.7" and the website "www.hestiacp.com".
- Installation message: "Thank you for downloading Hestia Control Panel! In a few moments, we will begin installing the following components on your server:"
- Component list: A bulleted list of components to be installed:
  - NGINX Web / Proxy Server
  - Apache Web Server (as backend)
  - PHP-FPM Application Server
  - Bind DNS Server
  - Exim Mail Server
  - Dovecot POP3/IMAP Server
  - MariaDB Database Server
  - Vsftpd FTP Server
  - Firewall (iptables) + Fail2Ban Access Monitor
- Final question: "Would you like to continue with the installation? [Y/N]:"



Nos saldrá una ventana que nos pedirá agregar un correo electrónico, un dominio (Si no tienes un dominio puedes ir a este apartado donde explico cómo crear uno pag. 98 ) y nuestro hostname y listo empezara a instalarse

```
Would you like to continue with the installation? [Y/N]: Y
Please use a valid email address (ex. info@domain.tld).
Please enter admin email address: info@emqxiot.tk
Please enter FQDN hostname [instance-emqx]: emqxiot.tk

Please use a valid hostname according to RFC1178 (ex. hostname.domain.tld).
Please enter FQDN hostname [instance-emqx]: www.emqxiot.tk
Installation backup directory: /root/hst_install_backups/300820220015
Installation log file: /root/hst_install_backups/hst_install-300820220015.log

Setting up swapspace version 1, size = 1024 MiB (1073737728 bytes)
no label, UUID=3e9bd352-371b-439c-a329-cb04ceb1ac1a
Adding required repositories to proceed with installation:

[ * ] NGINX
[ * ] PHP
```

**Nota:** Cuando se termine de instalar nos arrojara una información como esta en la cual es muy importante guardar ya que será nuestra Admin URL y nuestro usuario y contraseña

Congratulations!

You have successfully installed Hestia Control Panel on your server.

Ready to get started? Log in using the following credentials:

Admin URL:	<a href="https://129.151.39.126:8083">https://129.151.39.126:8083</a>
Username:	admin
Password:	CPRDT8oNWQqCfOLO

Thank you for choosing Hestia Control Panel to power your full stack web server, we hope that you enjoy using it as much as we do!

Please feel free to contact us at any time if you have any questions, or if you encounter any bugs or problems:

Documentation:	<a href="https://docs.hestiacp.com/">https://docs.hestiacp.com/</a>
Forum:	<a href="https://forum.hestiacp.com/">https://forum.hestiacp.com/</a>
Discord:	<a href="https://discord.gg/nXRUZch">https://discord.gg/nXRUZch</a>
GitHub:	<a href="https://www.github.com/hestiacp/hestiacp">https://www.github.com/hestiacp/hestiacp</a>



Una vez instalado Hestia podremos tomar el Admin URL y ponerlo en nuestro navegador para acceder a él pero si lo hacemos así no nos cargara la interfaz ya que no tenemos el puerto 8083 habilitado por ello iremos a nuestro panel de la instancia y abriremos el apartado de Red Virtual en la Nube

The screenshot shows the Oracle Cloud interface for an instance named 'sa-saopaulo-1'. Under the 'EN EJECUCIÓN' tab, it displays the following details:

- Dominio de errores:** FD-2
- Región:** sa-saopaulo-1
- OCID:** ...uxariq [Mostrar](#) [Copiar](#)
- Iniciado:** lun, 29 ago 2022, 23:53:14 UTC
- Compartimento:** iothost84 (raíz)
- Tipo de capacidad:** Bajo demanda

**Detalles de la instancia**

- Red virtual en la nube:** vcn-iot [Mostrar](#) [Copiar](#)
- Reinicio por mantenimiento:** -
- Imagen:** Canonical-Ubuntu-22.04-2022.08.10-10.0
- Modo de inicio:** PARAVIRTUALIZED
- Servicio de metadatos de instancia:** Versiones 1 y 2 [Editar](#) [i](#)
- Migración en directo:** Usar valor por defecto recomendado [i](#)
- Acción de recuperación de mantenimiento:** Restaurar instancia

**Configuración de unidad**

- Unidad:** VM.Standard.E2.1.Micro [Mostrar](#) [Copiar](#)
- Recuento de OCPU:** 1

**Redes:** (SSH) Necesitará la clave privada del par de claves SSH que se utilizó para crear la instancia.

**Dirección IP pública:** 129.151.39.126 [Copiar](#)

**Nombre de usuario:** ubuntu

**VNIC primario**

- Dirección IP Privada:** 10.0.0.168
- Grupos de seguridad de red:** Ninguno [Editar](#) [i](#)
- Subred:** subnet-iot
- Registro de DNS privado:** Activar
- Nombre de host:** instance-emqx
- FQDN interno:** instance-emqx... [Mostrar](#) [Copiar](#)

**Opciones de inicio**

- Tipo de asociación de NIC:** PARAVIRTUALIZED
- Volumen de datos remotos:** PARAVIRTUALIZED
- Firmware:** UEFI\_64
- Tipo de volumen de inicio:** PARAVIRTUALIZED
- Cifrado en tránsito:** Desactivado

Se no abrirá un panel donde muestra nuestras redes y subredes y ahí en el apartado de subredes le daremos clic a la subred que ya tenemos creada en mi caso mi subred se llama subnet-iot

The screenshot shows the Oracle Cloud interface for a VCN named 'vcn'. Under the 'DISPONIBLE' tab, it displays the following details:

- Compartimento:** iothost84 (raíz)
- Creación:** lun, 29 ago 2022, 1:19:28 UTC
- Bloque de CIDR IPv4:** 10.0.0.0/16
- IPv6 prefix:** Ningún valor

**Subredes:**

- OCID:** ...hrifnq [Mostrar](#) [Copiar](#)
- Solucionador DNS:** vcn-iot
- Tabla de rutas por defecto:** Default Route Table for vcn-iot
- Nombre de dominio DNS:** vcn08282119.oraclevcn.com

**cursos**

**Subredes en iothost84 (raíz) Compartimento**

Crear Subred						
Nombre	Estado	Bloque de CIDR IPv4	IPv6 Prefixes	Acceso a la subred	Creación	
subnet-iot	● Disponible	10.0.0.0/24	-	Público (Regional)	lun, 29 ago 2022, 1:19:31 UTC	<a href="#">Mostrar 1 elemento</a> < 1 de 1 > <a href="#">Copiar</a>



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez ahí se nos abrirá otro panel en el cual nos dirigiremos al apartado de lista de seguridad y lo abriremos

The screenshot shows the Cloud Infrastructure Management Platform interface. On the left, there's a sidebar with a large green hexagonal icon containing a white letter 'S' and the word 'DISPONIBLE' below it. The main content area has two tabs: 'Información de subred' (selected) and 'Etiquetas'. Under 'Información de subred', there are several details: OCID: ...zjkzya, Bloque de CIDR IPv4: 10.0.0.0/24, IPv6 Prefix: Ningún valor, Dirección MAC del enrutador virtual: 00:00:17:DF:ED:FB, and Tipo de subred: Regional. To the right, under 'Compartimiento', it shows 'iohost84 (raiz)'. Other options like 'Nombre de dominio DNS', 'Acceso a la subred', 'Opciones de DHCP', and 'Tabla de rutas' are listed. Below this, the 'Listas de seguridad' section is shown, featuring a table with one row: 'Default Security List for vcn-iot' (Estado: Disponible, Compartimiento: iohost84 (raiz), Creación: lun, 29 ago 2022, 1:19:28 UTC). A message at the bottom says 'Mostrando 1 elemento < 1 de 1'.

Igualmente se nos abrirá un panel llamado Reglas de entrada en el cual le daremos clic al apartado azul de Agregar reglas de entrada

#### Reglas de entrada

The screenshot shows the 'Reglas de entrada' (Network Rules) section. At the top, there are buttons for 'Agregar reglas de entrada' (blue), 'Editar' (grey), and 'Eliminar' (grey). The table below has columns: Estado (checkbox), Sin estado (dropdown), Origen, Protocolo IP, Rango de puertos de origen, Rango de puertos de destino, Tipo y código, Permite, and Descripción. There are three entries:

Estado	Sin estado	Origen	Protocolo IP	Rango de puertos de origen	Rango de puertos de destino	Tipo y código	Permite	Descripción
<input type="checkbox"/>	No	0.0.0.0/0	TCP	Todo	22		Permitir	Tráfico TCP para puerto s: 22 SSH Remote Login Protocol
<input type="checkbox"/>	No	0.0.0.0/0	ICMP		3, 4		Permitir	Tráfico ICMP para: 3, 4 Destino no accesible: Se necesita fragmentación y se había definido no fragmentar
<input type="checkbox"/>	No	10.0.0.0/16	ICMP		3		Permitir	Tráfico ICMP para: 3 Destino no accesible



Una vez ahí solo tendremos que llenar los campos según corresponda para poder habilitar nuestro puerto y le daremos en agregar y listo ya no aparecerá nuestro puerto habilitado

### Agregar reglas de entrada

Regla de entrada 1

Permite tráfico TCP 8083

Sin estado *(i)*

Tipo de origen CIDR

CIDR de origen 0.0.0.0/0

Direcciones IP especificadas: 0.0.0.0-255.255.255.255 (direcciones IP 4.294.967.296)

Protocolo IP *(i)* TCP

Rango de puertos de origen *Opcional* (i) Todo

Ejemplos: 80, 20-22

Rango de puertos de destino *Opcional* (i) 8083

Ejemplos: 80, 20-22

Descripción *Opcional* VESTA

255 caracteres máximo

+ Otra regla de entrada

En tipo de origen podemos seleccionar entre CIDR y Servicios

El puerto que vamos a habilitar

**Agregar reglas de entrada** [Cancelar](#)

Agregar reglas de entrada								
<input type="checkbox"/>	<a href="#">Sin estado ▾</a>	Origen	Protocolo IP	Rango de puertos de origen	Rango de puertos de destino	Tipo y código	Permite	Descripción
<input type="checkbox"/>	No	0.0.0.0/0	TCP	Todo	22		Tráfico TCP para puerto s: 22 SSH Remote Login Protocol	
<input type="checkbox"/>	No	0.0.0.0/0	ICMP		3, 4		Tráfico ICMP para: 3, 4 Destino no accesible: Se necesita fragmentación y se había definido no fragmentar	
<input type="checkbox"/>	No	10.0.0.0/16	ICMP		3		Tráfico ICMP para: 3 Destino no accesible	
<input type="checkbox"/>	No	0.0.0.0/0	TCP	Todo	8083		Tráfico TCP para puerto s: 8083 VESTA	



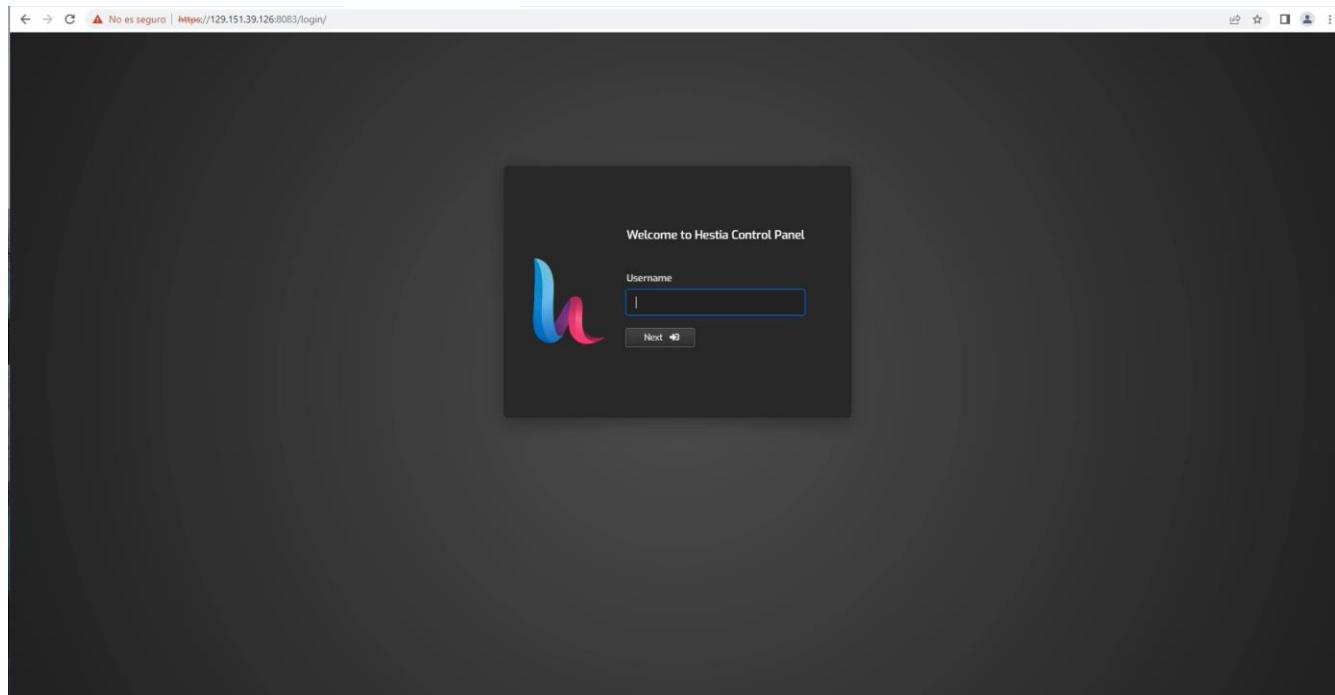
GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez tengamos habilitado el puerto ya podemos poner el Admin Url (En mi caso 129.151.39.126:8083) en nuestro navegador y ahora si nos permitirá conectarnos a Hestia y nos pedirá nuestras credenciales de acceso las cuales son las que nos generó por consola cuando habíamos instalado Hestia





Una vez dentro de Hestia podremos ver todos los apartados que tiene y uno de ellos es FireWall aquí es donde nos mostrara los puertos que utilizan cada uno de los servicios como lo es Web (Para alojar una página web con puertos 80 y 443) por ello si deseas utilizar alguno de estos servicios tendrás que habilitar el puerto esto como lo vimos anteriormente pag. 92

The screenshot shows the Hestia firewall configuration page. At the top, there are tabs for USERS, WEB, DNS, MAIL, CRON, and BACKUPS. The WEB tab is selected, showing a list of allowed ports for various services. The table has columns for Action, Service, Protocol, Port, and IP Address. The allowed ports listed are:

Action	Service	Protocol	Port	IP Address
Allow	PING	ICMP	0	0.0.0.0/0
Allow	HESTIA	TCP	8083	0.0.0.0/0
Allow	IMAP	TCP	143,993	0.0.0.0/0
Allow	POP3	TCP	110,995	0.0.0.0/0
Allow	SMTP	TCP	25,465,587	0.0.0.0/0
Allow	DNS	TCP	53	0.0.0.0/0
Allow	DNS	UDP	53	0.0.0.0/0
Allow	FTP	TCP	21,12000-12100	0.0.0.0/0
Allow	WEB	TCP	80,443	0.0.0.0/0
Allow	SSH	TCP	22	0.0.0.0/0



ORACLE Cloud Search resources, services, documentation, and Marketplace Brazil East (Sao Paulo) ▾

Creation, Jun, 29 ago 2022, 1:1

## Reglas de entrada

### Agregar reglas de entrada

Regla de entrada 1

Permite tráfico TCP 80

Sin estado ⓘ

Tipo de origen CIDR CIDR de origen 0.0.0.0/0 Protocolo IP ⓘ TCP

Direcciones IP especificadas: 0.0.0-255.255.255.255 (direcciones IP 4.294.967.296)

Rango de puertos de origen Opcional ⓘ Todo Ejemplos: 80, 20-22

Rango de puertos de destino Opcional ⓘ 80 Ejemplos: 80, 20-22

Descripción Opcional HTTPS I 255 caracteres máximo

+ Otra regla de entrada

## Agregar reglas de entrada

### Regla de entrada 1

Permite tráfico TCP 443 HTTPS

 Sin estado ⓘ

Tipo de origen

CIDR

CIDR de origen

0.0.0.0/0

Protocolo IP ⓘ

TCP

Direcciones IP especificadas: 0.0.0-255.255.255.255 (direcciones IP 4.294.967.296)

Rango de puertos de origen Opcional ⓘ

Todo

Rango de puertos de destino Opcional ⓘ

443

Ejemplos: 80, 20-22

Ejemplos: 80, 20-22

Descripción Opcional

HTTPS

255 caracteres máximo

+ Otra regla de entrada





## 5.5 Registro de un dominio

Una vez tengamos todo lo visto anteriormente ahora nos dirigiremos a la siguiente página de Freenom y en esta crearemos una cuenta con el mismo correo que pusimos en la de Oracle para trabajar [15]

The screenshot shows the Freenom website's homepage. The background is a photograph of a busy street in a city. A large blue circular callout bubble is positioned in the upper right area, pointing towards a white button labeled "Comprobar disponibilidad". Above this button is a smaller text box containing the text "Encontrar un nuevo dominio GRATIS". Below the main image, there's a promotional section for "freenomworld". It features a "NEW" badge, the "freenomworld" logo, and the text "Freenom World is a fast and anonymous Public DNS resolver". To the right of this text are two input fields under the heading "CHANGE YOUR DNS TO": one containing "80.80.80.80" and another containing "80.80.81.81". To the right of these fields is a video player showing a YouTube video titled "Windows 10 HowTo Español". At the bottom of the page, there are links for "WHY FREENOM WORLD? WATCH VIDEO" and "VISIT FREENOM WORLD".

Una vez creada su cuenta nos posicionaremos en el apartado de comprobar disponibilidad y ahí introduciremos el nombre de nuestro dominio que inventamos con anterioridad para ver si está disponible para utilizarlo GRATIS por 12 meses.

**Nota:** Si no sabes que dominio pusiste en Hestia puedes ir al panel de Hestia y entrar al apartado de Web ahí te saldrá el dominio que ingresaste anteriormente esto en caso de que hayas instalado Hestia antes de crear tu dominio

Hestia

- Records
- Files
- Statistics

USERS

WEB

DNS

MAIL

CRON

BACKUPS

users: 1  
suspended: 0
domains: 1 / 1 (0)  
aliases: 1 / 1
zones: 0 / 1 (0)  
records: 0 / ↴
domains: 0 / 1 (0)  
accounts: 0 / 1
jobs: 11 / ↴ (0)
backups: 0 / 1

[Add Web Domain](#)

SORT BY: NAME ↑
Apply to selected
→
🔍

Name	IP Address	Disk	Bandwidth	SSL	Statistics
emqxiot.tk	129.151.39.126	0 mb	0 mb	✗	✗

1 web domain

Una vez ingreses tú dominio te saldrán varias opciones de dominios disponibles parecidos al tuyo por si este no estuviera disponible

Comprobar  
Disponibilidad

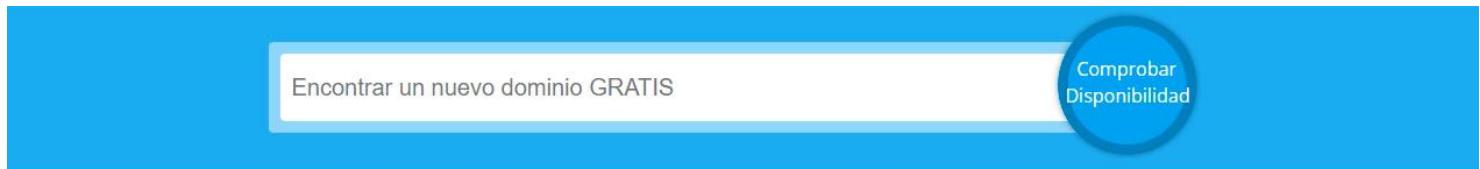
1 dominio en el carro
[Finalizar la compra](#)

Consiga uno de estos dominios. Son gratis!

emqxiot <b>.tk</b>	<b>• GRATIS</b>	USD 0.00	Selected
emqxiot <b>.ml</b>	<b>• GRATIS</b>	USD 0.00	<a href="#" style="color: #0070C0;">Get it now!</a>
emqxiot <b>.ga</b>	<b>• GRATIS</b>	USD 0.00	<a href="#" style="color: #0070C0;">Get it now!</a>
emqxiot <b>.cf</b>	<b>• GRATIS</b>	USD 0.00	<a href="#" style="color: #0070C0;">Get it now!</a>
emqxiot <b>.gq</b>	<b>• GRATIS</b>	USD 0.00	<a href="#" style="color: #0070C0;">Get it now!</a>



Una vez seleccionado nuestro dominio correspondiente nos saldrá una ventana en donde deberemos de configurar algunos parámetros para crear nuestro dominio, una vez hecho esto le daremos en continuar y listo ya tendremos nuestro dominio creado



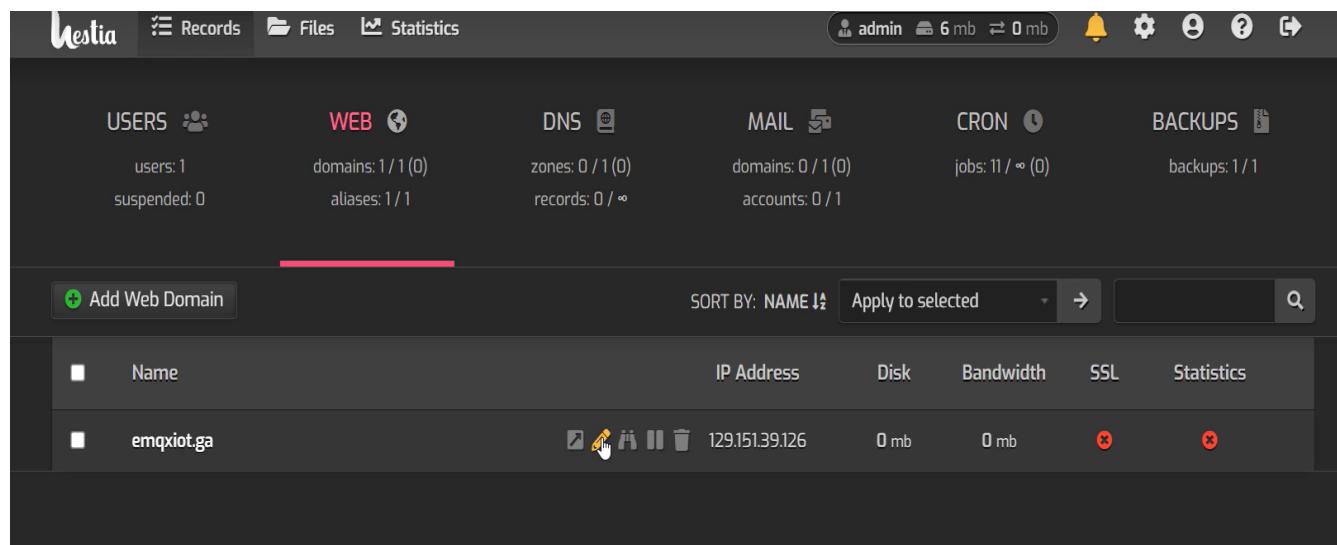
Dominio	IDSHIELD	?	Usar su nuevo dominio	Período								
emqxiot.tk			<input type="checkbox"/> Redirigir este dominio or <input checked="" type="checkbox"/> Usar DNS	12 Meses @ FREE								
<p>Introduzca su registro A aquí</p> <table><tr><td>Nombre de host</td><td>emqxiot.tk</td><td>Dirección IP</td><td>129.151.39.126</td></tr><tr><td>Nombre de host</td><td>www.emqxiot.tk</td><td>Dirección IP</td><td>129.151.39.126</td></tr></table>					Nombre de host	emqxiot.tk	Dirección IP	129.151.39.126	Nombre de host	www.emqxiot.tk	Dirección IP	129.151.39.126
Nombre de host	emqxiot.tk	Dirección IP	129.151.39.126									
Nombre de host	www.emqxiot.tk	Dirección IP	129.151.39.126									

Ingresamos la dirección IP a la cual queremos que nos dirija cuando alguien ponga la dirección de nuestro host

El número de meses que queremos que esté disponible ese host

## 5.6 Certificados SSL al Dominio

Una vez tengamos creado nuestro dominio procederemos a colocarles nuestro certificados para ello en nuestro panel de Hestia nos dirigiremos al apartado de Web y ahí nos aparecerá nuestro dominio y si no posamos sobre él nos saldrá un lápiz de color amarillo y ahí le daremos clic



Name	IP Address	Disk	Bandwidth	SSL	Statistics
emqxiot.ga	129.151.39.126	0 mb	0 mb		

Una vez ahí nos saldrá un aparto como este en el cual podremos editar el re direccionamiento del host o en nuestro caso habilitar el apartado de estadísticas el cual nos permitirá colocar un usuario y contraseña a nuestro host así como habilitar el SSL y otros apartados



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

**Domain**

**Aliases**

**IP Address**

**Web Statistics**

**Enable domain redirection**

**Enable SSL for this domain**

129.151.39.126

**Web Statistics**

**Statistics Authorization**

**Username**

**Password / Generate**

**Enable domain redirection**

**Enable SSL for this domain**

**Use Lets Encrypt to obtain SSL certificate**

**Enable automatic HTTPS redirection**

**Enable HTTP Strict Transport Security (HSTS) ?**

Una vez  
habilitado



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

todo lo que necesitamos le daremos a guardar (esto tardara un poco) y una vez se hayan realizado los cambios nos arrojara los certificados

### Enable HTTP Strict Transport Security (HSTS) ?

#### SSL Certificate

```
-----BEGIN CERTIFICATE-----  
MIIGKjCCBRKgAwIBAgISBI2Rj8IQarNNdzD+BmzDBK9YMA0GCSqGSIb  
MDIxCzAJBgNVBAYTA1VTMRYwFAYDVQQKEw1MZXQncyBFbmNyeXB0MQs  
EwJSMzAeFw0yMjA4MzEwMDM2NDNaFw0yMjExMjkwMDM2NDJaMBUxEzA  
CmVtcXhpB3QuZ2EwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIKAoI  
PQ1...Et...3V1...4YQfD...S...Q...C...P...M...9...K...1...E...t...G...t...I...F...D...u...n...P...O...h...k...l...t...  
-----END CERTIFICATE-----
```

#### SSL Key

```
-----BEGIN PRIVATE KEY-----  
MIIJQQIBADANBgkqhkiG9w0BAQEFAASCCSswggknAgEAAoICAQC2dz1  
V+5+4YQhPzf...u6iDME9iK815ajytG+ci5UDrmwBcObxhtovVZtMnmd  
MM3gwRaw5kAhJOE3ULUJG+yx/rQZ/pTf/wCBQoADM9hjQzcFQhH5RJV  
nfmu...0ORGaaBTRqSqpFEeb4s/YvTM53j5pR2A5noqcqpcAvgxishYT...  
TCE...wV...---X7...GIMT...791K60...75H1...1W...NN...S...L...DW...MP...1...J...M...P...  
-----END PRIVATE KEY-----
```

#### SSL Certificate Authority / Intermediate (Optional)

```
-----BEGIN CERTIFICATE-----  
MIIFFFjCCAv6gAwIBAgIRAJERCeRPDBinU/bWLiWnX1owDQYJKoZIhv  
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVyb...V0IFN1Y3VyaXR  
cmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwHhcNMjAwOT...  
WhcNMjUwOTE1MTYwMDAwWjAyMQswCQYDVQQGEwJVUzEWMBQGA1UECh...  
RWE...4...1...D...E...L...M...G...A...1...H...E...A...M...C...H...M...E...I...M...A...G...C...C...G...S...T...L...3...D...C...F...A...Q...U...M...T...  
-----END CERTIFICATE-----
```



Y con esto ya podremos trabajar de con una conexión manera segura en nuestro host sin embargo también es importante habilitar los certificados para nuestro panel de control de Hestia por ello en nuestra consola de Linux o consola de Putty pondremos el comando **hostname** el cual nos arrojara nuestra dirección de host de Hestia la cual debemos cambiar si no es la misma dirección que registramos anteriormente en le dominio

```
Last login: Tue Aug 30 00:02:27 2022 from 131.0.197.37
ubuntu@www:~$ sudo su
root@www:/home/ubuntu# cd ..
root@www:/home# cd
root@www:~# cd ..
root@www:# hostname
www.emqxiot.tk
root@www:/# █
```

Para cambiar esa dirección lo que vamos a hacer es poner el siguiente comando **v-change-sys-hostname el nombre del host que registramos con anterioridad** [16]

En caso de que nos salga un error es porque en las últimas versiones de Hestia no instala las cosas correctamente en el path por ello en la consola podremos la siguiente ruta **/usr/local/Hestia/bin/v-change-sys-hostname el nombre del host que registramos con anteriorida** y listo con esto quedaría cambiado

```
Last login: Tue Aug 30 00:02:27 2022 from 131.0.197.37
ubuntu@www:~$ sudo su
root@www:/home/ubuntu# cd ..
root@www:/home# cd
root@www:~# cd ..
root@www:# hostname
www.emqxiot.tk
root@www:# v-change-sys-hostname emqxiot.ga
v-change-sys-hostname: command not found
root@www:# ^C
root@www:# /usr/local/hestia/bin/v-change-sys-hostname emqxiot.ga
root@www:# hostname
emqxiot.ga
```



Ahora lo que vamos a hacer ahí mismo en la consola es poner el siguiente comando **/usr/local/Hestia/bin/v-add-letsencrypt-host** ya que lo que hace esta función es verificar y validar el certificado backend y generar un nuevo certificado y listo ya tendremos el certificado en nuestro panel de Hestia

```
Last login: Tue Aug 30 00:02:27 2022 from 131.0.197.37
ubuntu@www:~$ sudo su
root@www:/home/ubuntu# cd ..
root@www:/home# cd
root@www:~# cd ..
root@www:# hostname
www.emqxiot.tk
root@www:# v-change-sys-hostname emqxiot.ga
v-change-sys-hostname: command not found
root@www:# ^C
root@www:# /usr/local/hestia/bin/v-change-sys-hostname emqxiot.ga
root@www:# hostname
emqxiot.ga
root@www:# /usr/local/hestia/bin/v-add-letsencrypt-host
root@www:#
```

## 5.7 Instalar EMQX en el VPS

Una vez ya tengamos instalado todo el panel de Hestia en nuestro VPS (**Si hasta este punto has tenido problemas puede que sea un bug de Ubuntu por lo cual recomiendo probar con la versión 20.04**) nos iremos al apartado de USERS y ahí le daremos clic al apartado packages

Name	Package	IPs	0 mb	0 mb	1	0	0	0	11	1
admin (System Administrator) Email: info@emqxiot.ga	system	1	0 mb	0 mb	1	0	0	0	11	1
demo (demo demo) Email: demo@coso.com	default	0	0 mb	0 mb	0	0	0	0	0	0



Una vez ahí nos mostrara una tabla en la cual le daremos clic en Add Package y configuraremos los siguientes parámetros que solo sería agregar el nombre del packges

The screenshot shows the CDESÍ web interface with a dark theme. At the top, there's a navigation bar with links for 'Records', 'Files', and 'Statistics'. Below the navigation, there are six main categories: 'USERS', 'WEB', 'DNS', 'MAIL', 'CRON', and 'BACKUPS'. Under 'WEB', there's a sub-section for 'Add Package' where three packages are listed: 'Package', 'default', and 'system'. Each package has a row of icons for managing its settings.

This screenshot shows the 'Add Package' form. It includes fields for 'Package Name' (set to 'emqx'), 'Quota (in MB)' (set to '1000'), 'Bandwidth (in MB)' (set to '1000'), and 'Backups' (set to '1'). Below these fields, there are sections for 'Web' and 'DNS' services, each with a '+' button to add more configurations.



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Después de agregar el nombre abriremos ahí mismo el apartado de web y lo configuraremos de la siguiente manera y para el resto de apartados los podemos dejar como están o editarlos para lo que necesitemos como en el apartado de databases que si solo necesitamos sola una base de datos lo podemos dejar así como está pero de lo contrario tendremos que ir a su apartado y editarlo como deseemos

The screenshot shows a dark-themed web interface for managing web configurations. At the top, there are tabs for USERS, WEB, DNS, and MAIL. Below the tabs, there is a 'Back' button with a left arrow icon. The main area is titled 'Web Domains' and shows a range from 1 to infinity. Under 'Web Aliases (Per Domain)', it also shows a range from 1 to infinity, which is highlighted with a blue border. Below these are dropdown menus for 'Web Template' set to 'APACHE2', 'Backend Template' set to 'PHP-FPM', and 'Proxy Template' set to 'NGINX', all currently set to 'default'. A small cursor arrow is visible on the left side of the interface.



Finalmente le daremos en guardar y esperaremos que se realicen los cambios correspondientes, ahora una vez hecho esto y estando como súper usuarios en nuestra consola lo que tenemos que hacer es poner el siguiente comando **/usr/local/Hestia/bin/v-change-sys-php 7083** que es para cambiar el puerto en el que está funcionando Hestia debido a que Hestia funcionan en el 8083 y el bróker EMQX también por ello cambiaremos el puerto de Hestia en mi caso por el 7083

```
root@emqxiot:/home/ubuntu# /usr/local/hestia/bin/v-change-sys-port 7083
root@emqxiot:/home/ubuntu#
```

Si intentamos entrar al panel de Hestia por nuestro nuevo puerto nos dará un error ya que recordemos que en el panel de Oracle habilitamos el puerto 8083 para Hestia por lo que el 7083 no está habilitado, por ello lo único que tenemos que hacer es ir al mismo apartado donde activamos el puerto 8083 y cambiarlo por el 7083

Protocolo	Origen	Destino	Acción	Opciones
ICMP	10.0.0.0/16		No	Tráfico ICMP para: 3 Destino no accesible
TCP	0.0.0.0/0	8083	No	Tráfico TCP para puertos: 8083 HES
TCP	0.0.0.0/0	80	No	Tráfico TCP para puertos: 80 HTTP
TCP	0.0.0.0/0	443	No	Tráfico TCP para puertos: 443 HTTPS

≡ ORACLE Cloud Buscar recursos, servicios, documentación y Marketplace Brazil East (Sao Paulo) ▾

### Editar regla de entrada

Regla de entrada 1

Permite tráfico TCP 7083

Sin estado ⓘ

Tipo de origen	CIDR	CIDR de origen	Protocolo IP ⓘ
	▼	0.0.0.0/0	TCP

Direcciones IP especificadas: 0.0.0.0-255.255.255.255 (direcciones IP 4.294.967.296)

Rango de puertos de origen Opcional ⓘ

Todo Ejemplos: 80, 20-22

Rango de puertos de destino Opcional ⓘ

7083 Ejemplos: 80, 20-22

Descripción Opcional

Guardar cambios Cancelar

Una vez que ya podamos entrar a nuestro panel de Hestia lo que tenemos que hacer es ir a la página EMQX.io que ya hemos visto con anterioridad y ahí nos dirigiremos al apartado de descargas para Ubuntu y seleccionaremos la versión 4.8 de Packges

EMQX Enterprise Docs MQTT Developers ⓘ 10.3k EMQX Cloud Download

### Download EMQX

Docker Ubuntu Debian CentOS/RHEL macOS Windows Kubernetes Terraform

Apt Package

Version v5.0.7 Release Notes All Versions

v5.0.7 v4.4.8 v4.3.19 v4.2.14 v4.1.5 v4.0.13 v3.2.8

d64.deb SHA256 loads/broker/5.0.7/emqx-5.0.7-ubuntu20.04-amd64.deb

sudo apt install ./emqx-5.0.7-ubuntu20.04-amd64.deb

Run EMQX sudo emqx start

Contact Us Get Started → Server Estimate



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez seleccionado la versión de Ubuntu en la que vamos a trabajar y el tipo de paquete deb, solo coparemos los comandos en nuestra consola para descargar e instalar nuestro EMQX (**Recuerda que debes de estar en el directorio raíz antes de ingresar los comandos para instalar EMQX**)

```
root@emqxiot:/# wget https://www.emqx.com/en/downloads/broker/4.4.8/emqx-4.4.8-otp24.1.5-3-ubuntu20.04-amd64.deb
--2022-09-02 00:09:49-- https://www.emqx.com/en/downloads/broker/4.4.8/emqx-4.4.8-otp24.1.5-3-ubuntu20.04-amd64.deb
Resolving www.emqx.com (www.emqx.com)... 52.53.184.178
Connecting to www.emqx.com (www.emqx.com)|52.53.184.178|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://packages.emqx.io/emqx-ce/v4.4.8/emqx-4.4.8-otp24.1.5-3-ubuntu20.04-amd64.deb [following]
--2022-09-02 00:09:50-- https://packages.emqx.io/emqx-ce/v4.4.8/emqx-4.4.8-otp24.1.5-3-ubuntu20.04-amd64.deb
Resolving packages.emqx.io (packages.emqx.io)... 108.158.122.58, 108.158.122.69, 108.158.122.121, ...
Connecting to packages.emqx.io (packages.emqx.io)|108.158.122.58|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26160888 (25M) [application/x-debian-package]
Saving to: 'emqx-4.4.8-otp24.1.5-3-ubuntu20.04-amd64.deb'

emqx-4.4.8-otp24.1.5-3-ubuntu20.04 100%[=====] 24.95M 64.8MB/s in 0.4s
2022-09-02 00:09:52 (64.8 MB/s) - 'emqx-4.4.8-otp24.1.5-3-ubuntu20.04-amd64.deb' saved [26160888/26160888]

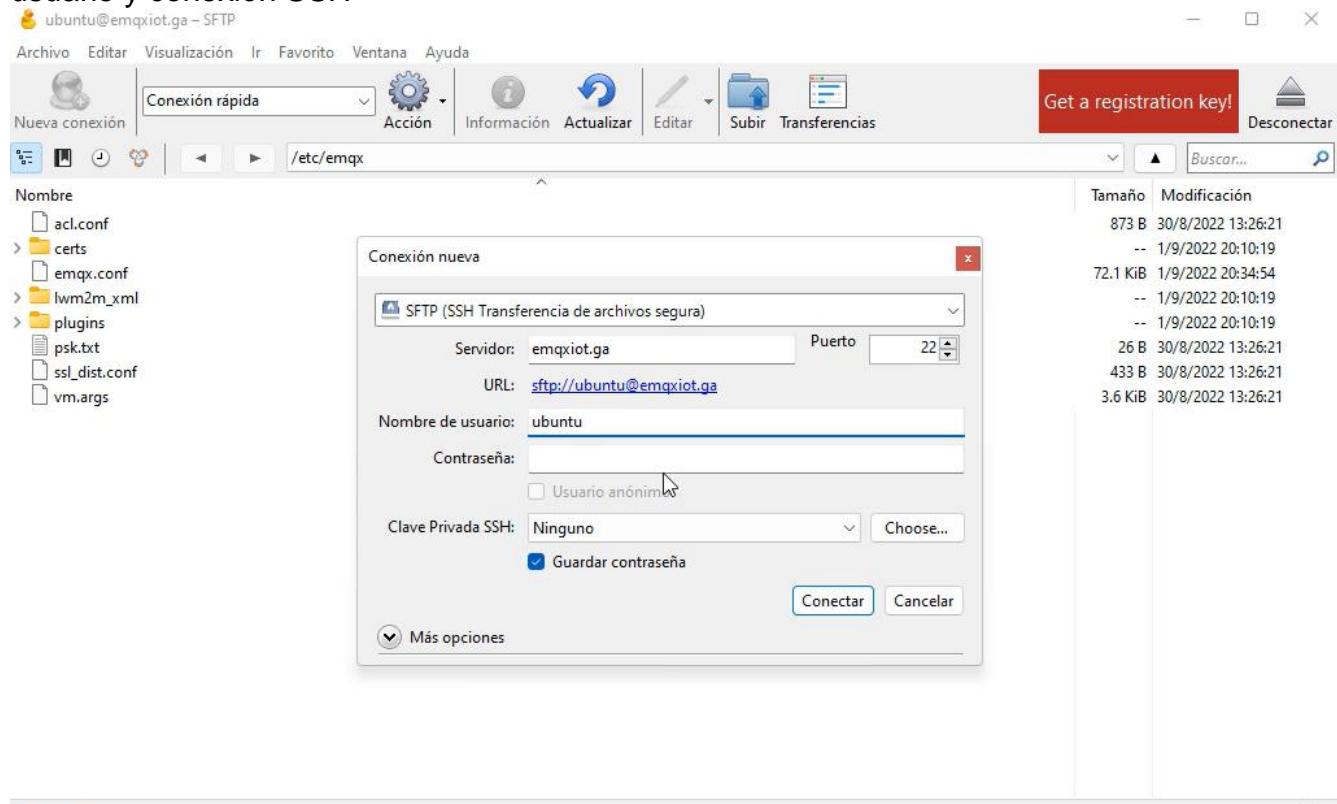
root@emqxiot:/#
```

```
root@emqxiot:/# sudo apt install ./emqx-4.4.8-otp24.1.5-3-ubuntu20.04-amd64.deb
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'emqx' instead of './emqx-4.4.8-otp24.1.5-3-ubuntu20.04-amd64.deb'
The following NEW packages will be installed:
  emqx
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/26.2 MB of archives.
After this operation, 44.8 MB of additional disk space will be used.
Get:1 /emqx-4.4.8-otp24.1.5-3-ubuntu20.04-amd64.deb emqx amd64 4.4.8 [26.2 MB]
Selecting previously unselected package emqx.
(Reading database ... 84738 files and directories currently installed.)
Preparing to unpack .../emqx-4.4.8-otp24.1.5-3-ubuntu20.04-amd64.deb ...
Unpacking emqx (4.4.8) ...

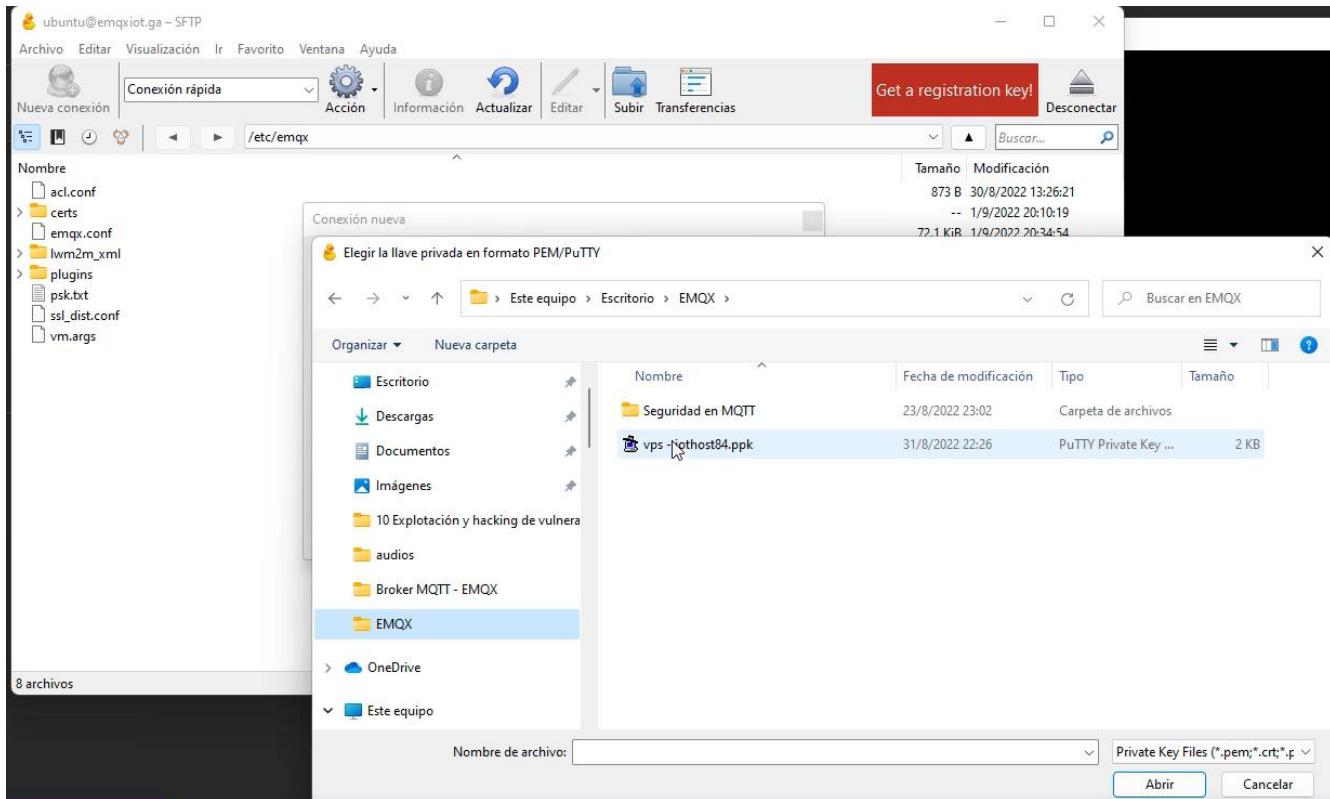
Progress: [ 20%] [#####.....]
```



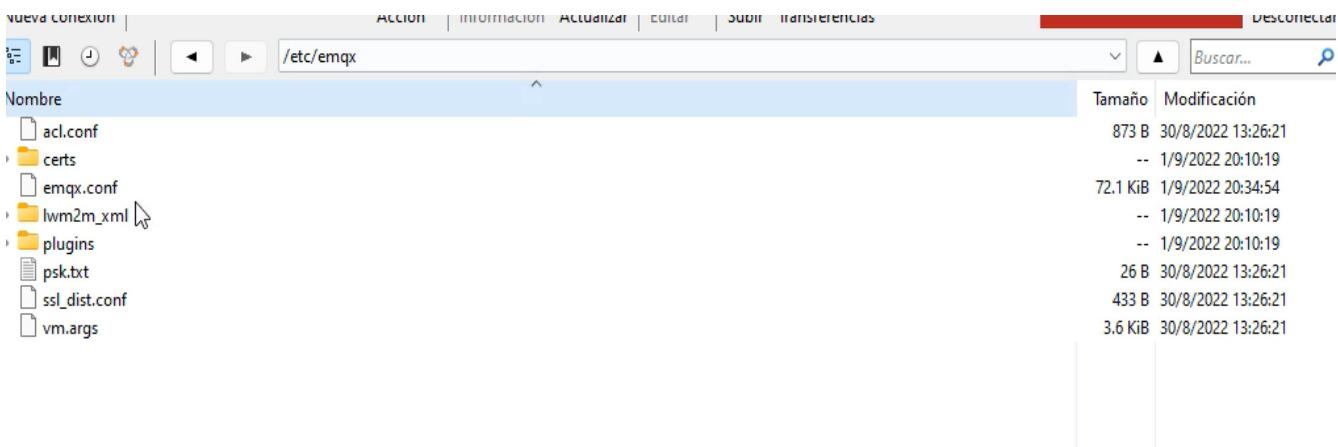
Una vez instalado nuestro EMQX en nuestro VPS solo le tendremos que dar el comando **emqx console** para iniciar el bróker y ver si tiene algún error. En caso de que les salga un error relacionado con los puertos lo que tenemos que hacer es editarlos y para hacer esto abriremos el programa que usamos con anterioridad que es CyberDuck e ingresaremos los datos correspondientes para conectarnos a nuestro VPS como lo es el servidor y el nombre de usuario y conexión SSH



En el apartado de Clave Privada SSH lo que tenemos que hacer es buscar la key que aviamos guardado antes y que es la misma que creamos para conectarnos con Putty



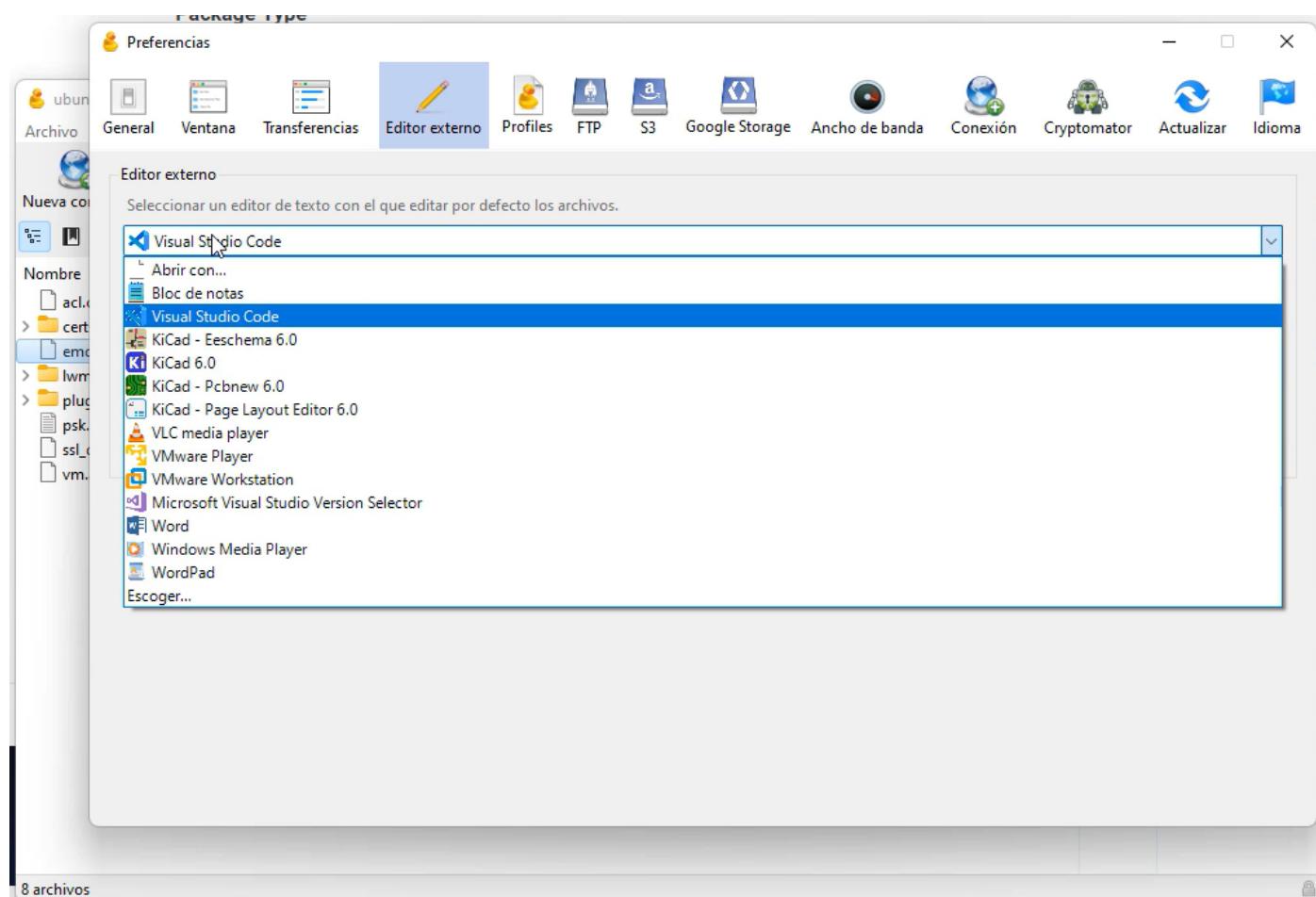
SI hicimos todo correctamente nos conectaremos a nuestro VPS y ahora solo tendremos que buscar la carpeta en la cual se instaló el bróker en este caso sería en **/etc/emqx**



Una vez hayamos encontrado la carpeta emqx lo que debemos hacer es editar algunos archivos por ello en el apartado de **editar/preferencias/opciones de editor de texto** en



CyberDuck podremos asignar nuestro editor de texto de preferencia en mi caso sería Visual Studio Code



Una vez hecho esto ya podremos editar nuestros archivos pero antes de hacerlo es importante pedirle permiso a Ubuntu ya que si no lo hacemos cuando estemos guardado los cambios en



los archivos nos saldrá un error por ello en nuestra consola lo que vamos a hacer es entrar como usuario root y una vez ahí poner el siguiente comando **chown -R nombre\_de\_usuario /ruta de los archivos a editar**

```
root@emqxiot:/ # chown -R ubuntu /etc/emqx
root@emqxiot:/ #
```

Ahora si abriremos el archivo llamado emqx.conf y buscaremos el puerto que deseamos configurar según el error que tuvimos en mi caso será el 8084 por ello lo buscare en el archivo y lo cambiaremos por un puerto que este disponible en mi caso 8074

```
1925
1926 ##-
1927 ## External WebSocket/SSL listener for MQTT Protocol
1928
1929 ## listener.wss.$name is the IP address and port that the MQTT/WebSocket/SSL
1930 ## listener will bind.
1931 ##
1932 ## Value: IP:Port | Port
1933 ##
1934 ## Examples: 8084, 127.0.0.1:8084, ::1:8084 -> change 8074
1935 listener.wss.external = 8074
1936
1937 ## The path of WebSocket MQTT endpoint
1938 ##
1939 ## Value: URL Path
1940 listener.wss.external.mqtt_path = /mqtt
1941
1942 ## The acceptor pool for external MQTT/WebSocket/SSL listener.
1943 ##
1944 ## Value: Number
1945 listener.wss.external.acceptors = 4
1946
1947 ## Maximum number of concurrent MQTT/Websocket/SSL connections.
1948 ##
1949 ## Value: Number
1950 listener.wss.external.max_connections = 102400
```



Guardamos el archivo que acabamos de editar y volvemos a iniciar el bróker para ver si no hay más errores en mi caso me da un error en el puerto de management que por defecto es el 8081 por ello buscare su archivo editable en este caso se encuentra en la carpeta plugins

```
root@emqxiot:/# chown -R ubuntu /etc/emqx
root@emqxiot:/# emqx console
Erlang/OTP 24 [erts-12.1.5] [emqx] [64-bit] [smp:2:2] [ds:2:2:8] [async-threads:4] [jit]

Starting emqx on node emqx@127.0.0.1
Start mqtt:tcp:internal listener on 127.0.0.1:11883 successfully.
Start mqtt:tcp:external listener on 0.0.0.0:1883 successfully.
Start mqtt:ws:external listener on 0.0.0.0:8083 successfully.
Start mqtt:ssl:external listener on 0.0.0.0:8883 successfully.
Start mqtt:wss:external listener on 0.0.0.0:8074 successfully.
2022-09-02T00:43:33.444683+00:00 [error] Failed to start Ranch listener 'http:management' in ranch_tcp: listen([{{certs,'...'}, {key,'...'}, {cert,'...'}}, {port,8081}, {nodelay,true}, {send_timeout_close,true}, {send_timeout,15000}, {backlog,512}]) for reason eaddrinuse (address already in use)
2022-09-02T00:43:33.444891+00:00 [error] Supervisor: {<0.1832.0>,ranch_listener_sup}. Context: start_error. Reason: {listen_error,['http:management'],eaddrinuse}. Offender: id=ranch_acceptors_sup,pid=undefined.
2022-09-02T00:43:33.445073+00:00 [error] crasher: initial call: supervisor:ranch_acceptors_sup/1, pid: <0.1834.0>, registered_name: [], exit: {listen_error,['http:management'],[ranch_acceptors_sup]}, ancestors: []
2022-09-02T00:43:33.445073+00:00 [error] crasher: initial call: application_master:init/4, pid: <0.1829.0>, registered_name: [], exit: {bad_return,{[{emqx_mgmt_app,start,[normal,[]]},{'EXIT',[{minirest,start_http,3,[{file,"minirest.erl"},{line,57}],{lists,foreach,2,[{file,"lists.erl"},{line,1342}],{emqx_mgmt_app,start,2,[{file,"emqx_mgmt_app.erl"},{line,30}]},{application_master,start_it_old,4,[{file,"application_master.erl"},{line,293}]}]}]},[[{application_master,start_it_old,4,[{file,"application_master.erl"},{line,142}]}],{proc_lib,init_p_do_apply,3,[{file,"proc_lib.erl"},{line,226}]}]}}, ancestors: []
2022-09-02T00:43:33.445073+00:00 [error] Minirest(Handler): Start http:management listener on 8081 unsuccessfully: the port is occupied
2022-09-02T00:43:33.490091+00:00 [error] crasher: initial call: application_master:init/4, pid: <0.1829.0>, registered_name: [], exit: {bad_return,{[{emqx_mgmt_app,start,[normal,[]]},{'EXIT',[{minirest,start_http,3,[{file,"minirest.erl"},{line,57}],{lists,foreach,2,[{file,"lists.erl"},{line,1342}],{emqx_mgmt_app,start,2,[{file,"emqx_mgmt_app.erl"},{line,30}]},{application_master,start_it_old,4,[{file,"application_master.erl"},{line,293}]}]}]},[[{application_master,start_it_old,4,[{file,"application_master.erl"},{line,142}]}],{proc_lib,init_p_do_apply,3,[{file,"proc_lib.erl"},{line,226}]}]}}, ancestors: []
```

The screenshot shows a file manager window with the following details:

- Barra superior:** Archivo, Editar, Visualización, Ir, Favorito, Ventana, Ayuda.
- Camino:** /etc/emqx
- Opciones:** Conexión rápida, Acción, Información, Actualizar, Editar, Subir, Transferencias.
- Botones:** Get a registration key!, Desconecta.
- Árbol de directorios:** emqx\_auth\_mnesia.conf, emqx\_auth\_mongo.conf, emqx\_auth\_mysql.conf, emqx\_auth\_pgsql.conf, emqx\_auth\_redis.conf, emqx\_bridge\_mqtt.conf, emqx\_coap.conf, emqx\_dashboard.conf, emqx\_exhook.conf, emqx\_exproto.conf, emqx\_lua\_hook.conf, emqx\_lwm2m.conf, emqx\_management.conf (selecciónada), emqx\_prometheus.conf.
- Tabla de archivos:**

Nombre	Tamaño	Modificación
emqx_auth_mnesia.conf	993 B	30/8/2022 13:26:21
emqx_auth_mongo.conf	4.7 KiB	30/8/2022 13:26:21
emqx_auth_mysql.conf	3.2 KiB	30/8/2022 13:26:21
emqx_auth_pgsql.conf	3.0 KiB	30/8/2022 13:26:21
emqx_auth_redis.conf	3.0 KiB	30/8/2022 13:26:21
emqx_bridge_mqtt.conf	5.4 KiB	30/8/2022 13:26:21
emqx_coap.conf	3.3 KiB	30/8/2022 13:26:21
emqx_dashboard.conf	4.8 KiB	30/8/2022 13:26:21
emqx_exhook.conf	2.4 KiB	30/8/2022 13:26:21
emqx_exproto.conf	9.6 KiB	30/8/2022 13:26:21
emqx_lua_hook.conf	160 B	30/8/2022 13:26:21
emqx_lwm2m.conf	5.6 KiB	30/8/2022 13:26:21
emqx_management.conf	2.8 KiB	30/8/2022 13:26:21
emqx_prometheus.conf	423 B	30/8/2022 13:26:21



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez abierto el archivo buscaremos el puerto correspondiente y lo cambiaremos por uno disponible en mi caso lo cambiare por el 8071 y lo guardare

```
21 management.default_application.secret = public
22
23 #-----
24 ## HTTP Listener
25
26 management.listener.http = 8071
27 management.listener.http.acceptors = 2
28 management.listener.http.max_clients = 512
29 management.listener.http.backlog = 512
30 management.listener.http.send_timeout = 15s
31 management.listener.http.send_timeout_close = on
32 management.listener.http.inet6 = false
33 management.listener.http.ipv6_v6only = false
34
35 #-----
```

Una vez hecho esto volveremos a arrancar el bróker si ya no salen error entonces ya podremos inicializarlo en nuestro VPS de lo contrario tendremos que seguir editando y arreglando los errores que salgan en mi caso ya funciona entonces lo inicializare y para ello lo que tengo que hacer es habilitar los puertos en Hestia por eso me iré a configuración

The screenshot shows the Hestia Control Panel dashboard. At the top, there are navigation links for 'Records', 'Files', and 'Statistics'. On the right side, there's a user status bar showing 'admin' with 0 mb used and 0 mb free, along with icons for notifications, help, and settings. Below the header, there are seven main performance metrics displayed in a grid:

USERS	WEB	DNS	MAIL	DB	CRON	BACKUPS
users: 2	domains: 1 / ∞ (0)	zones: 0 / ∞ (0)	domains: 0 / ∞ (0)	databases: 0 / ∞ (0)	jobs: 11 / ∞ (0)	backups: 1 / 1
suspended: 0	aliases: 1 / ∞	records: 0 / ∞	accounts: 0 / ∞			



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez ahí me iré al apartado de firewall y veremos las reglas que tiene por defecto Hestia por ello ahora tenemos que crear las nuestras para el bróker por ello le daremos en Add Rule y configuraremos las reglas que necesitemos

Hestia    Records    Files    Statistics    admin    0 mb    0 mb    🔔    ⚙    🌐    🛡    🌐    🌐

USERS	WEB	DNS	MAIL	DB	CRON	BACKUPS
users: 2 suspended: 0	domains: 1 / ∞ (0) aliases: 1 / ∞	zones: 0 / ∞ (0) records: 0 / ∞	domains: 0 / ∞ (0) accounts: 0 / ∞	databases: 0 / ∞ (0)	jobs: 11 / ∞ (0)	backups: 1 / 1

Back    Add Rule    Manage Banned IPs    Manage IP lists    SORT BY: ACTION ↑    Apply to selected

Action	Service	Protocol	Port	IP Address
<input checked="" type="checkbox"/> Allow	PING	ICMP	0	0.0.0.0/0
<input checked="" type="checkbox"/> Allow	HESTIA	TCP	7083	0.0.0.0/0
<input checked="" type="checkbox"/> Allow	IMAP	TCP	143,993	0.0.0.0/0
<input checked="" type="checkbox"/> Allow	POP3	TCP	110,995	0.0.0.0/0
<input checked="" type="checkbox"/> Allow	SMTP	TCP	25,465,587	0.0.0.0/0
<input checked="" type="checkbox"/> Allow	DNS	TCP	53	0.0.0.0/0
<input checked="" type="checkbox"/> Allow	DNS	UDP	53	0.0.0.0/0
<input checked="" type="checkbox"/> Allow	FTP	TCP	21,12000-12100	0.0.0.0/0



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Lo que vamos a editar en las reglas es primero es permitir luego que se comunique por protocolo TCP y agregar el puerto que queremos que se abra en este caso el 1883 para visualizar el bróker y con IP list 0.0.0.0/0 para que cualquiera pueda acceder a él y por ultimo alguna nota de que hace esa regla

## Add Firewall Rule

**Action**

Allow

**Protocol**

TCP

**Port (Ranges and Lists are acceptable)**

1883

**IP address / IP list (CIDR format is supported)**

0.0.0.0/0

**Service (Optional)**

EMQX



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Ahora habilitaremos también el puerto 8883 para el tráfico seguro y el puerto 18083, 18084, 8083, 8073, 8074 y loaremos igual como vimos antes llenando cada apartado según lo que se necesite

## Add Firewall Rule

Rule has been created successfully.

### Action

Allow

### Protocol

TCP

### Port (Ranges and Lists are acceptable)

8883

### IP address / IP list (CIDR format is supported)

0.0.0.0/0



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

# Add Firewall Rule

✓ Rule has been created successfully.

## Action

Allow

## Protocol

TCP

## Port (Ranges and Lists are acceptable)

18083

## IP address / IP list (CIDR format is supported)

0.0.0.0/0

## Service (Optional)

EMQX



# Add Firewall Rule

✓ Rule has been created successfully.

## Action

Allow

## Protocol

TCP

## Port (Ranges and Lists are acceptable)

18084

## IP address / IP list (CIDR format is supported)

0.0.0.0/0



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

# Add Firewall Rule

Rule has been created successfully.

## Action

Allow

## Protocol

TCP

## Port (Ranges and Lists are acceptable)

8073-8074

## IP address / IP list (CIDR format is supported)

0.0.0.0/0

## Service (Optional)



Una vez hecho esto nos iremos a nuestro panel de control de Oracle y en el apartado de nuestras redes y subredes ahí en reglas de entrada como vimos anteriormente habilitaremos los mismos puertos que habilitamos en Hestia

ORACLE Cloud Buscar recursos, servicios, documentación y Marketplace Brazil East (Sao Paulo) Perfil

Recursos Reglas de entrada Agregar reglas de entrada

Regla de entrada 1

Permite tráfico TCP 1883,8883

Sin estado ⓘ

Tipo de origen CIDR CIDR de origen 0.0.0.0/0 Direcciones IP especificadas: 0.0.0.0-255.255.255.255 (direcciones IP 4.294.967.296)

Rango de puertos de origen Opcional ⓘ Todo Ejemplos: 80, 20-22

Rango de puertos de destino Opcional ⓘ 1883,8883 Ejemplos: 80, 20-22

Descripción Opcional EMQX 255 caracteres máximo

+ Otra regla de entrada

Agregar reglas de entrada Cancelar

ORACLE Cloud Buscar recursos, servicios, documentación y Marketplace Brazil East (Sao Paulo) Perfil

Recursos Reglas de entrada Agregar reglas de entrada

Regla de entrada 1

Permite tráfico TCP 1883,18084,8073,8074

Sin estado ⓘ

Tipo de origen CIDR CIDR de origen 0.0.0.0/0 Direcciones IP especificadas: 0.0.0.0-255.255.255.255 (direcciones IP 4.294.967.296)

Rango de puertos de origen Opcional ⓘ Todo Ejemplos: 80, 20-22

Rango de puertos de destino Opcional ⓘ 18083,18084,8073,8074 Ejemplos: 80, 20-22

Descripción Opcional EMQX 255 caracteres máximo

+ Otra regla de entrada

Agregar reglas de entrada Cancelar



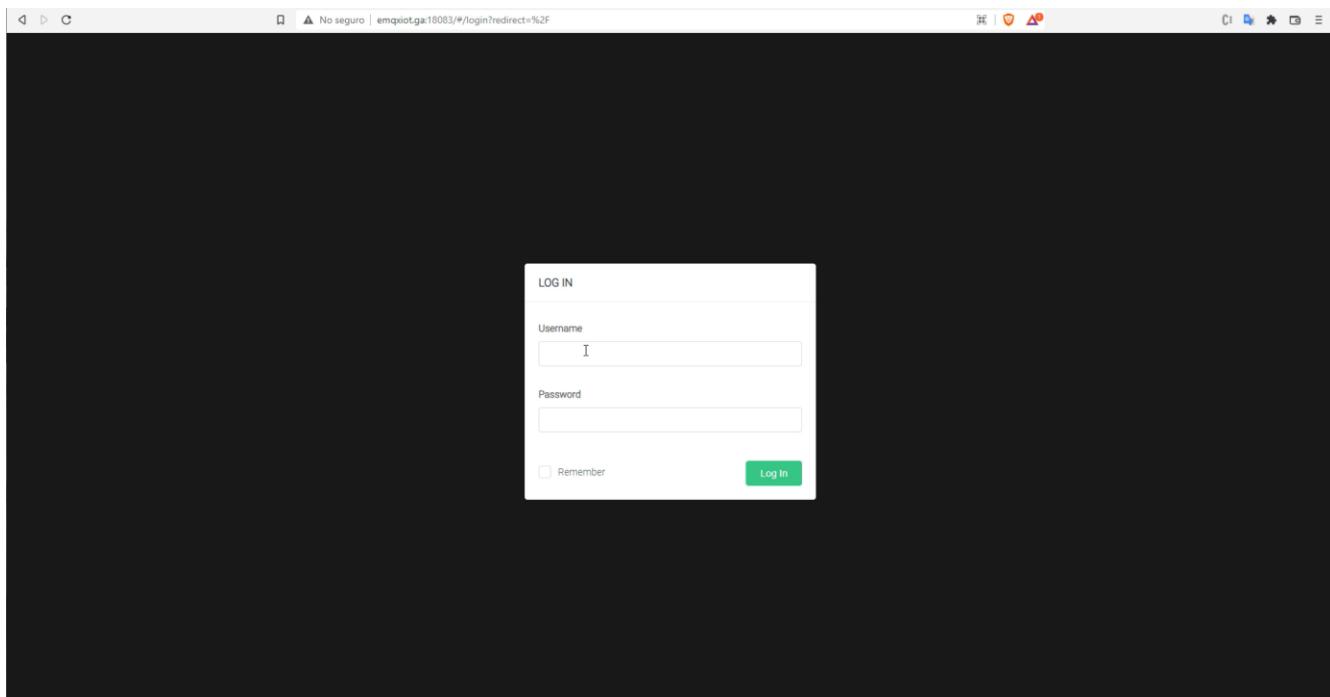
GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez hecho esto si ponemos nuestro dominio ya podremos acceder al bróker atreves de uno de los puertos que habilitamos, en mi caso quedaría así [www.emqxiot.ga:18083](http://www.emqxiot.ga:18083) y ya solo pondríamos las credenciales de default y listo ya tendríamos el bróker





Una vez ya tengamos el bróker en nuestro VPS podemos probar realizar conexiones con nuestro cliente para ver si se está conectando correctamente por ello abriremos nuestro cliente MQTTX y crearemos una nueva conexión con los datos correspondientes para conectarnos a nuestro servidor y que ya vimos antes como se llena

The screenshot shows the MQTTX application interface. On the left is a sidebar with icons for disconnecting, adding a new connection, opening files, and help. The main area has tabs for 'Connections' (selected) and 'New Collection'. A list shows a single connection named 'mqtt\_emqx@emqxi...'. The right side is the 'Edit' view for this connection. It has two sections: 'General' and 'Advanced'. In the 'General' section, fields include: Name (mqtt\_emqx), Client ID (mqtx\_a1499b4b), Host (mqtt://emqxiot.ga), Port (1883), Username (yamir84), and Password (empty). The 'SSL/TLS' toggle is off. In the 'Advanced' section, fields include: Connect Timeout (10), Keep Alive (60), Clean Session (true selected), and Auto Reconnect (false selected). A 'Connect' button is visible at the top right of the edit view.



Una vez el cliente se haya conectado a nuestro bróker lo que tendremos que hacer es ir al Websocket y ahí crear una conexión con los parámetros que deseamos y debemos poner y una vez se conecten ya podremos mandar mensajes entre cliente y servidor

## 5.8 Agregar certificados SSL al bróker EMQX

Lo primero que vamos a hacer es abrir nuestra terminal y una vez ahí nos loguearemos a nuestro VPS y una vez hecho esto entraremos a la carpeta donde se generaron nuestros certificados en el directorio **/home/admin/config/web** y ahí nos saldrá el nombre de nuestro dominio en (mi caso emqxiot.ga) y una vez ahí abriremos este archivo y entraremos a la carpeta ssl y ahí estarán los certificados que vamos a clonar

```
root@emqxiot: /home/admin/conf/web/emqxiot.ga/ssl
root@emqxiot:/# cd /home/admin/conf/web
root@emqxiot:/home/admin/conf/web# ls
EMQXIOT.GA
root@emqxiot:/home/admin/conf/web# cd emqxiot.ga
root@emqxiot:/home/admin/conf/web/emqxiot.ga# ls
apache2.conf      awstats.conf    nginx.conf_letsencrypt  nginx.hsts.conf  nginx.ssl.conf_letsencrypt
apache2.ssl.conf  nginx.conf     nginx.foressl.conf   nginx.ssl.conf  ssl
root@emqxiot:/home/admin/conf/web/emqxiot.ga# cd ssl
bash: cd: ssl: No such file or directory
root@emqxiot:/home/admin/conf/web/emqxiot.ga# cd ssl
root@emqxiot:/home/admin/conf/web/emqxiot.ga/ssl# ls
emqxiot.ga.ca  emqxiot.ga.crt  emqxiot.ga.key  emqxiot.ga.pem
root@emqxiot:/home/admin/conf/web/emqxiot.ga/ssl#
```

Una vez visualizados estos certificados nos iremos a nuestra carpeta **etc/emqx/certs** y una vez ahí tendremos que cambiarle el nombre a nuestro certificado cert.pem y key.pem por el nombre que queremos en caso de que no quieras borrarlos

```
root@emqxiot:/home/admin/conf/web/emqxiot.ga/ssl# cd /etc/emqx
root@emqxiot:/etc/emqx# ls
acl.conf  certs  emqx.conf  lwm2m_xml  plugins  psk.txt  ssl_dist.conf  vm.args
root@emqxiot:/etc/emqx# cd certs
root@emqxiot:/etc/emqx/certs# ls
README  cacert.pem  cert.pem  client-cert.pem  client-key.pem  key.pem
root@emqxiot:/etc/emqx/certs# mv cert.pem old_cert.pem
root@emqxiot:/etc/emqx/certs# mv key.pem old_key.pem
root@emqxiot:/etc/emqx/certs# ls
README  cacert.pem  client-cert.pem  client-key.pem  old_cert.pem  old_key.pem
root@emqxiot:/etc/emqx/certs#
```



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez hecho esto ahora tocara crear un link referencial así los certificados que necesitamos por ello usaremos el siguiente comando

**In –s /home/admin/conf/web/tudominio\_com/ssl/tudominio.crt/etc/emqx/certs/cert.pem**  
Recordemos que en **tudominio\_com** tendrás que cambiarlo por el nombre de la carpeta donde estamos referenciando los archivos y en tu **dominio.crt** lo cambiaremos por el nombre del archivo del dominio.crt que tienes tú y **cert.pem** deberás cambiarlo por el nombre de los certificados que necesitamos en mi caso quedaría así

**In –s /home/admin/conf/web/emqxiot.ga/ssl/emqxiot.ga.crt/etc/emqx/certs/cert.pem**

```
In: failed to create symbolic link '/emqx/etc/certs/cert.pem': No such file or directory
root@emqxiot:/etc/emqx/certs# ln -s /home/admin/conf/web/emqxiot.ga/ssl/emqxiot.ga.crt /etc/emqx/certs/cert.pem
root@emqxiot:/etc/emqx/certs# ls
README cacert.pem cert.pem client-cert.pem client-key.pem old_cert.pem old_key.pem
```

Ahora volveremos a hacer lo mismo pero para el certificado .key

```
root@emqxiot:/etc/emqx/certs# ln -s /home/admin/conf/web/emqxiot.ga/ssl/emqxiot.ga.key /etc/emqx/certs/key.pem
root@emqxiot:/etc/emqx/certs# ls
README cacert.pem cert.pem client-cert.pem client-key.pem key.pem old_cert.pem old_key.pem
root@emqxiot:/etc/emqx/certs#
```

Una vez hecho esto solo queda reiniciar el bróker con emqx restart y listo los certificados estarán funcionando ya podremos conectarnos mqqtts con el puerto 8074 (en mi caso)



Si tienes errores al quererte conectar puede ser debido a la versión del bróker que estés utilizando y para solucionarlo lo que tenemos que hacer es abrir nuestra terminal y ahí iremos la carpeta **/etc/emqx/certs** y pondremos el comando **ls -l**, nos mostrara a quien pertenece los archivos que copeamos con anterioridad y cómo podemos ver la ruta para linkear los certificados que necesitamos pertenece al usuario root por lo que si nosotros estamos logueados como otro usuario diferente no nos dejara utilizarlos en mi caso yo estaba logueado como Ubuntu y como emqx por lo que tengo que agregar estos perfiles a los permisos de la ruta de certificados

```
root@www:/etc/emqx/certs# ls -l
total 24
-rw-r--r-- 1 ubuntu emqx 38 Aug 30 17:26 README
-rw-r--r-- 1 ubuntu emqx 1212 Aug 30 17:26 cacert.pem
lrwxrwxrwx 1 root  root  50 Sep  3 02:33 cert.pem -> /home/admin/conf/web/emqxiot.ga/ssl/emqxiot.ga.crt
-rw-r--r-- 1 ubuntu emqx 1127 Aug 30 17:26 client-cert.pem
-rw-r--r-- 1 ubuntu emqx 1679 Aug 30 17:26 client-key.pem
lrwxrwxrwx 1 root  root  53 Sep  3 02:33 key.pem -> /home/admin/conf/web/tudominio_com/ssl/emqxiot.ga.key
-rw-r--r-- 1 ubuntu emqx 1127 Aug 30 17:26 old_cert.pem
-rw-r--r-- 1 ubuntu emqx 1675 Aug 30 17:26 old_key.pem
root@www:/etc/emqx/certs#
```

Para hacer esto lo que tenemos que hacer es elegir qué perfil quiero que tenga acceso a estos archivos en mi caso será el perfil emqx por que pondremos el siguiente comando  
**chown perfil /ruta del archivo.crt** y quedaría así  
**chown emqx /home/admin/conf/web/emqxiot.ga/ssl/emqxiot.ga.crt**

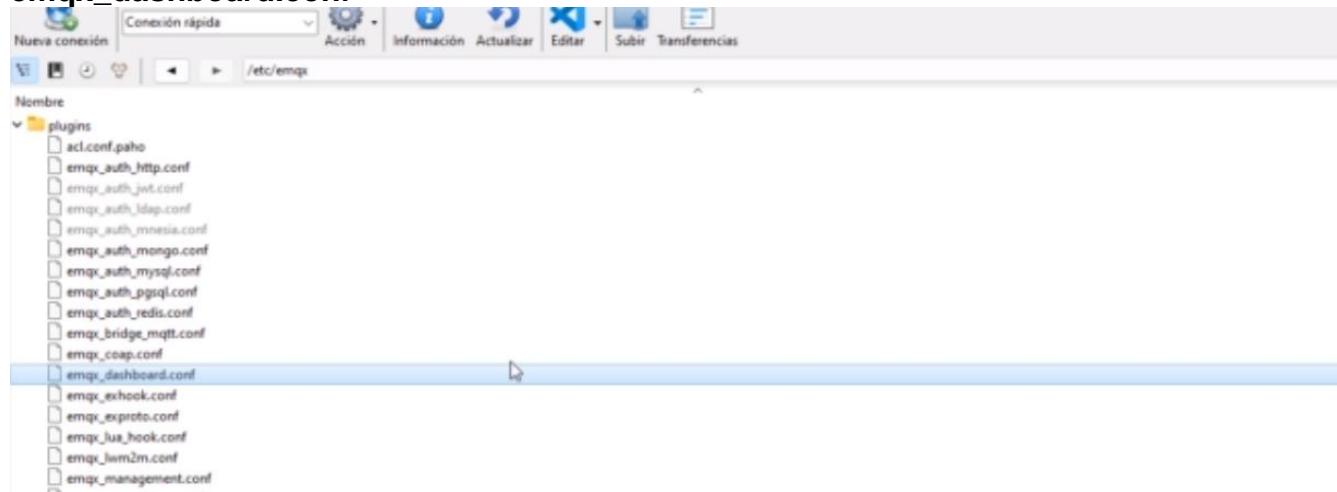
```
-rw-r--r-- 1 ubuntu emqx 38 Aug 30 17:26 README
-rw-r--r-- 1 ubuntu emqx 1212 Aug 30 17:26 cacert.pem
lrwxrwxrwx 1 root  root  50 Sep  3 02:33 cert.pem -> /home/admin/conf/web/emqxiot.ga/ssl/emqxiot.ga.crt
-rw-r--r-- 1 ubuntu emqx 1127 Aug 30 17:26 client-cert.pem
-rw-r--r-- 1 ubuntu emqx 1679 Aug 30 17:26 client-key.pem
lrwxrwxrwx 1 root  root  53 Sep  3 02:33 key.pem -> /home/admin/conf/web/tudominio_com/ssl/emqxiot.ga.key
-rw-r--r-- 1 ubuntu emqx 1127 Aug 30 17:26 old_cert.pem
-rw-r--r-- 1 ubuntu emqx 1675 Aug 30 17:26 old_key.pem
root@www:/etc/emqx/certs#
root@www:/etc/emqx/certs#
root@www:/etc/emqx/certs# chown emqx /home/admin/conf/web/emqxiot.ga/ssl/emqxiot.ga.crt
```

Esto lo deberemos hacer por cada certificado que necitemos como lo es el cert.pem y key.pem también podrás ver sus rutas en caso de que hayas escrito mal el comando anterior para agregar los certificados y así poder editarlos



## 5.9 Certificados SSL al Dashboard EMQX

Lo que vamos a hacer ahora es agregar nuestros certificados al dashboard por ello vamos a abrir nuestro CyberDuck y ahí buscaremos en el directorio de emqx el archivo `emqx_dashboard.conf`



Con nuestro editor de texto favorito editaremos este archivo para habilitar el número del puerto, las conexiones como lo hemos hecho antes

```
56  ##
57  ## Examples: 18084
58  dashboard.listener.https = 18084
59
60  ## The acceptor pool for external Dashboard HTTPS listener.
61  ##
62  ## Value: Number
63  dashboard.listener.https.acceptors = 2
64
65  ## Maximum number of concurrent Dashboard HTTPS connections.
66  ##
67  ## Value: Number
68  dashboard.listener.https.max_clients = 512
69
```



```
72 ## Value: false | true
73 dashboard.listener.https.inet6 = false
74
75 ## Listen on IPv4 and IPv6 (false) or only on IPv6 (true). Use with inet6.
76 ##
77 ## Value: false | true
78 dashboard.listener.https.ipv6only = false
79
80 ## Path to the file containing the user's private PEM-encoded key.
81 ##
82 ## Value: File
83 dashboard.listener.https.keyfile = etc/certs/key.pem
84
```

```
82 ## Value: File
83 dashboard.listener.https.keyfile = /etc/emqx/certs/key.pem
84
```

Una vez habilitado los que necesitamos tocara también editar el emqx\_management.conf para la conexión segura por ello lo abrimos y habilitamos y configuramos todos lo que necesitamos

```
37
38 ## management.listener.https = 8081
39 ## management.listener.https.acceptors = 2
40 ## management.listener.https.max_clients = 512
41 ## management.listener.https.backlog = 512
42 ## management.listener.https.send_timeout = 15s
43 ## management.listener.https.send_timeout_close = on
44 ## management.listener.https.certfile = etc/certs/cert.pem
45 ## management.listener.https.keyfile = etc/certs/key.pem
46 ## management.listener.https.key_password = yourpass
47 ## management.listener.https.cacertfile = etc/certs/cacert.pem
48 ## management.listener.https.verify = verify_peer
49 ## NOTE: Do not use tlsv1.3 if emqx is running on OTP-22 or earlier
50 ## management.listener.https.tls_versions = tlsv1.3,tlsv1.2,tlsv1.1,tlsv1
51 ## management.listener.https.ciphers = TLS_AES_256_GCM_SHA384,TLS_AES_128_GCM_SHA256,TLS_CHACHA20_POLY1305_SHA256,TLS_AES_128_GCM_SHA256
52 ## management.listener.https.fail_if_no_peer_cert = true
53 ## management.listener.https.inet6 = false
54 ## management.listener.https.ipv6only = false
55
```



```
35 #####-----#
36 ## HTTPS Listener
37
38 management.listener.https = 8072
39 management.listener.https.acceptors = 2
40 management.listener.https.max_clients = 512
41 management.listener.https.backlog = 512
42 management.listener.https.send_timeout = 15s
43 management.listener.https.send_timeout_close = on
44 management.listener.https.certfile = /etc/emqx/certs/cert.pem
45 management.listener.https.keyfile = /etc/emqx/certs/key.pem
46 ## management.listener.https.key_password = yourpass
47 ## management.listener.https.cacertfile = etc/certs/cacert.pem
48 ## management.listener.https.verify = verify_peer
49 ## NOTE: Do not use tlsv1.3 if emqx is running on OTP-22 or earlier
50 ## management.listener.https.tls_versions = tlsv1.3,tlsv1.2,tlsv1.1,tlsv1
51 ## management.listener.https.ciphers = TLS_AES_256_GCM_SHA384,TLS_AES_128_GCM_SHA256,TLS_CHACHA20_POLY1305_SHA256,TLS_AES_128_GCM_SHA256
52 ## management.listener.https.fail_if_no_peer_cert = true
53 ## management.listener.https.inet6 = false
54 ## management.listener.https.ipv6_only = false
55
```

Una vez hecho todo esto solo quedara reiniciar la consola y conectarnos por el 18084 en nuestro navegador y listo ya funciona el bróker

The screenshot shows the EMQ X Broker interface. On the left is a navigation sidebar with the following items: Monitor (selected), Clients, Topics, Subscriptions, Rule Engine, Analysis, Plugins (with a checked checkbox), Modules, Tools, Alarms (with a red dot), Settings, and General. The main area is titled "Overview". It has two sections: "Broker" and "Nodes(1)". The "Broker" section displays two cards: one for the System Name "EMQ X Broker" and another for Uptime "34 seconds". The "Nodes(1)" section shows a table with one row for "emqx@127.0.0.1". The columns are: Name, Erlang/OTP Release, Erlang Processes (used/available), CPU Info (load/memory), Memory, and MaxFds.

Name	Erlang/OTP Release	Erlang Processes (used/available)	CPU Info (1load/5load/15load)	Memory	MaxFds
emqx@127.0.0.1	24.1.5/12.1.5	359 / 2097152	0.19 / 0.11 / 0.10	558.53M / 964.17M	1024



## WebSocket

Connect

Host	Port	Path
emqxiot.ga	8074	/mqtt
Client ID	Username	Password
mqqtjs_9b4160c6b3		
Keep Alive	60	

Clean Session     SSL    <wss://emqxiot.ga:8074/mqtt>

[Connect](#)    [Disconnected](#)    Current State: CONNECTED

Una vez ya tengamos esto funcionando correctamente lo que tenemos que hacer ahora es agregar las autenticaciones para ello vamos abrir el archivo de emqx.conf y una vez ahí vamos a negar las conexiones anónimas por ello buscaremos el apartado de Access control y negamos el anonymous

```
649  ##-----  
650  ## Authentication/Access Control  
651  ##-----  
652  
653  ## Allow anonymous authentication by default if no auth plugins loaded.  
654  ## Notice: Disable the option in production deployment!  
655  ##  
656  ## Configured to 'false_quick_deny' means quick deny of anonymous clients  
657  ## (without username). So the auth backend checks can be skipped.  
658  ##  
659  ## Value: true | false | false_quick_deny  
660  allow_anonymous = false  
661
```



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez configurado esto ahora tendremos que ir a nuestro panel de herramientas de Hestia y una vez ahí pasaremos el cursor sobre el admin y nos saldrá un símbolo de lápiz y le daremos clic

The screenshot shows the Hestia Control Panel interface. At the top, there are links for Records, Files, and Statistics. On the right side of the header, there are icons for admin, memory (6 mb), disk space (0 mb), notifications, settings, help, and a refresh button. Below the header, there are six main sections: USERS, WEB, DNS, MAIL, CRON, and BACKUPS. The USERS section is currently selected. It shows 1 user (admin) and 1 suspended user. Under the WEB section, it shows 1 domain (emqxiot.ga) and 1 alias (emqxiot.ga). Under DNS, it shows 0 zones and 0 records. Under MAIL, it shows 0 domains and 0 accounts. Under CRON, it shows 11 jobs. Under BACKUPS, it shows 0 backups. At the bottom of the screen, there is a table listing users. The first row has columns for Name, Package, IPs, and various status indicators. The second row, which is highlighted, shows the user 'admin (System Administrator)' with the package 'system'. The table also includes columns for Email and a series of numerical values (1, 6 mb, 0 mb, 1, 0, 0, 0, 11, 0).

Una vez le hallamos dado clic nos saldrá un apartado de cosas a editar y nos iremos hasta abajo al apartado de Package y lo editaremos de system a default para poder habilitar el apartado de base de datos

The screenshot shows the Appearance settings in the Hestia Control Panel. At the top, there is a dropdown menu for 'English (English)'. Below it, there is a 'dark' theme selection. Under 'Default list sort order', there is a dropdown menu set to 'Name'. In the 'Package' section, there is a dropdown menu set to 'default'. The background of the page is dark.

Una vez hecho esto nos iremos al apartado de DB y le daremos en crear una base de datos y llenaremos los campos necesarios

## Add Database

Prefix `admin_` will be automatically added to database name and database user

**Database**

emqx	admin_emqx
------	------------

**Type**

mysql
-------

**Username** (maximum 32 characters in length (including prefix))

emqx	admin_emqx
------	------------

**Password** 

Emqx12345	
-----------	---

Your password must have at least:

- 8 characters long
- 1 uppercase & 1 lowercase character
- 1 number

Email login credentials to:

[Redacted]
------------



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Una vez creada la base de datos nos iremos al apartado de phpMyAdmin e iniciaremos sesión

Bienvenido a phpMyAdmin

Idioma (Language)

Español - Spanish

Iniciar sesión

Usuario:

Contraseña:

root

cursomania\_cursos

admin\_ihost

phpmyadmin

admin

demo\_test

Una vez hecho esto con nuestro programa de CyberDuck en el directorio de emqx abriremos la carpeta plugins y editaremos el archivo emqx\_auth\_mysql.conf con las credenciales de acceso de nuestra base de datos

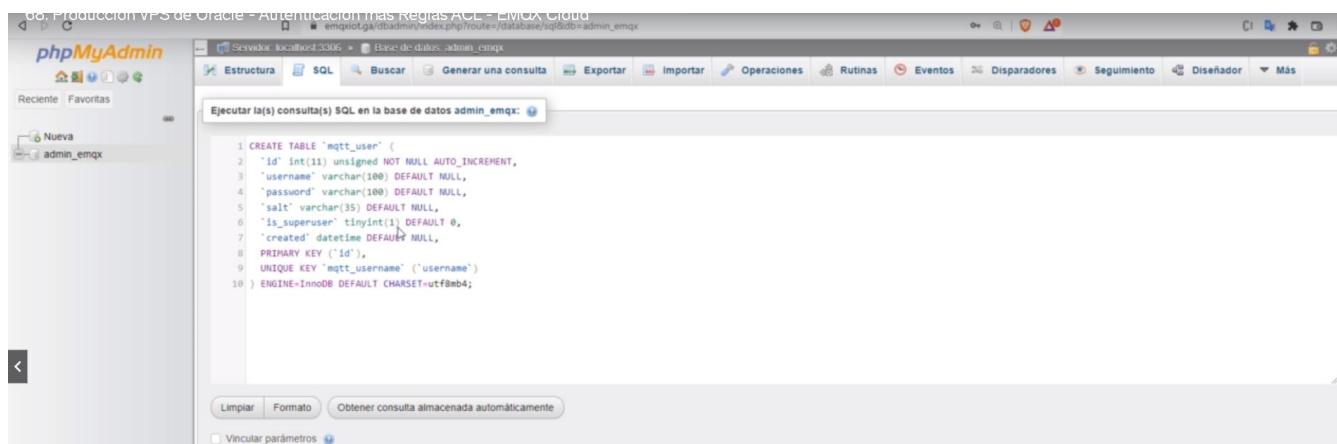
```
14 ## Value: Number
15 auth.mysql.pool = 8
16
17 ## MySQL username.
18 ##
19 ## Value: String
20 auth.mysql.username = admin_emqx
21
22 ## MySQL password.
23 ##
24 ## Value: String
25 auth.mysql.password = Emqx12345
26
27 ## MySQL database.
28 ##
29 ## Value: String
30 auth.mysql.database = admin_emqx
31
32 ## MySQL query timeout
33 ##
34 ## Value: Duration
35 auth.mysql.query_timeout = 5s
36
```

Una vez configurado esto nos iremos a nuestro phpMyAdmin y crearemos una tabla con estos mismos datos

```

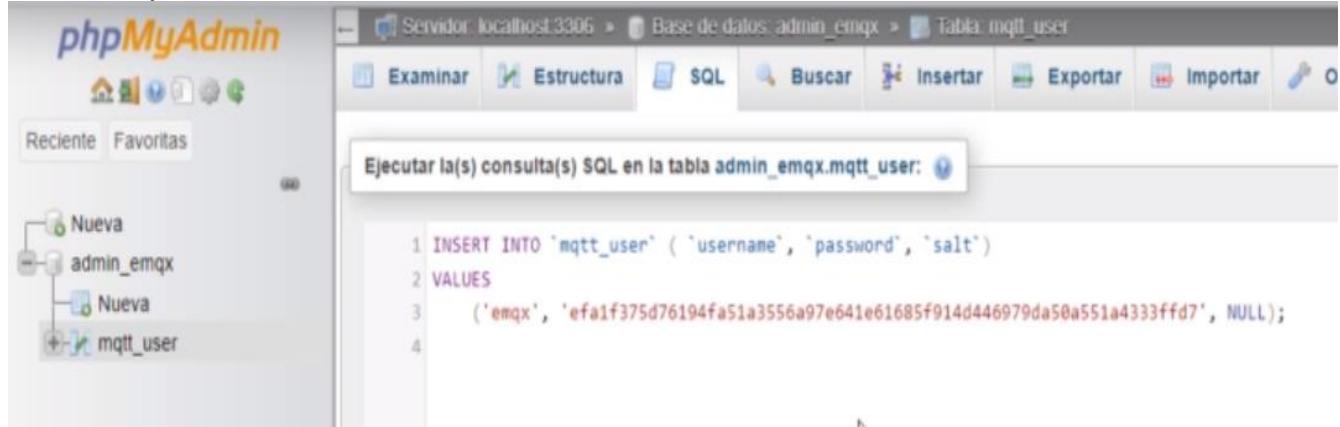
CREATE TABLE `mqtt_user` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `username` varchar(100) DEFAULT NULL,
  `password` varchar(100) DEFAULT NULL,
  `salt` varchar(35) DEFAULT NULL,
  `is_superuser` tinyint(1) DEFAULT 0,
  `created` datetime DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `mqtt_username` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```



The screenshot shows the phpMyAdmin interface. The left sidebar shows a database named 'admin\_emqx' with a single table 'Nueva'. The main area has a large text input field containing the SQL code for creating the 'mqtt\_user' table. Below the input field are several buttons: 'Limpiar', 'Formato', 'Obtener consulta almacenada automáticamente', and a checked checkbox 'Vincular parámetros'.

Una vez creada esta tabla crearemos el siguiente usuario y después ya podemos crear otros usuarios que necesitemos



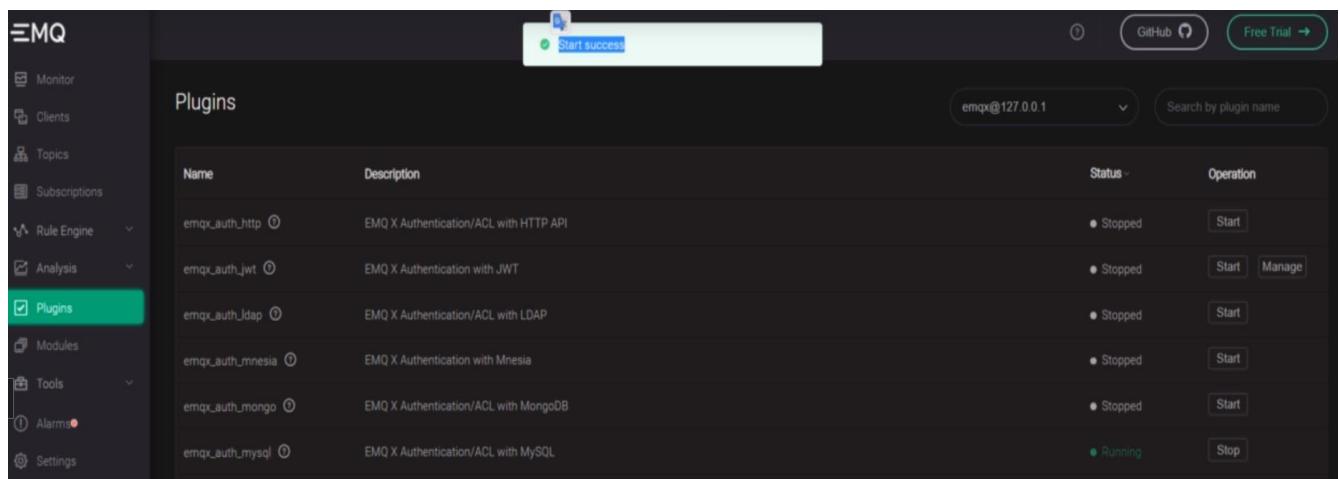
The screenshot shows the phpMyAdmin interface for a MySQL database named 'admin\_emqx'. A new table 'mqtt\_user' has been created. In the SQL tab, the following SQL code is being run:

```

1 INSERT INTO `mqtt_user` ( `username`, `password`, `salt` )
2 VALUES
3     ('emqx', 'efa1f375d76194fa51a3556a97e641e61685f914d446979da50a551a4333ffd7', NULL);
4

```

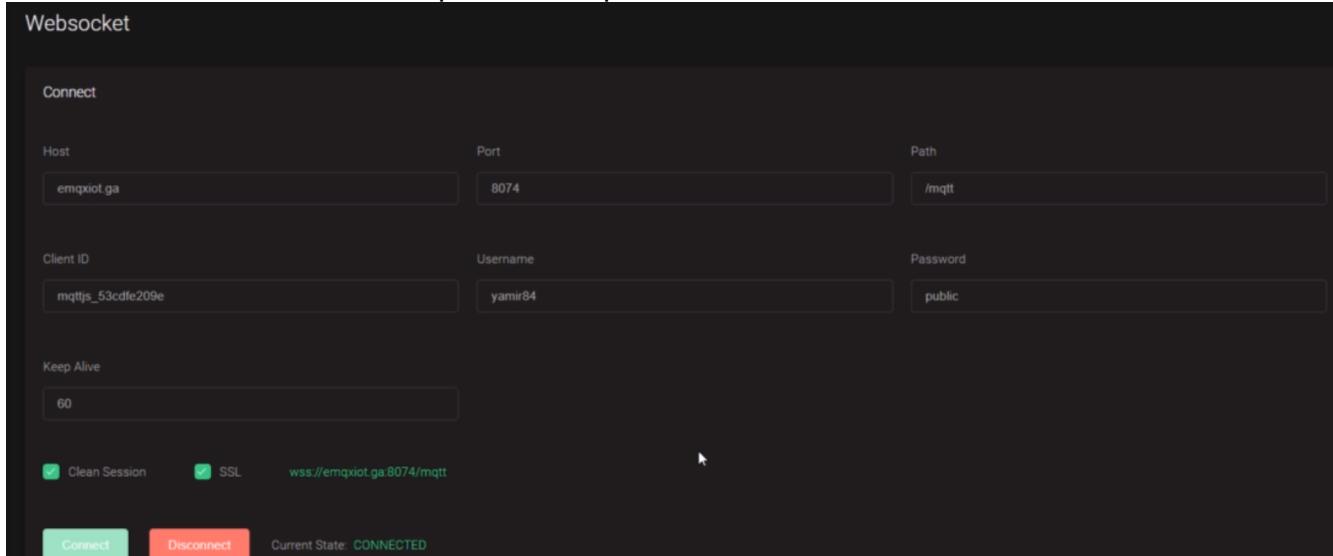
Una vez hecho esto tendremos que reiniciar el bróker y una vez reiniciado en el apartado de plugins buscaremos el de MySQL y lo iniciamos y si hicimos todo correctamente nos saldrá un aviso de que se inició sin problemas



The screenshot shows the EMQ X Broker interface. The left sidebar is open, showing the 'Plugins' section. The 'emqx\_auth\_mysql' plugin is listed as 'Running'. A green success message at the top right says 'Start success'.

Name	Description	Status	Operation
emqx_auth_http	EMQ X Authentication/ACL with HTTP API	Stopped	Start
emqx_auth_jwt	EMQ X Authentication with JWT	Stopped	Start   Manage
emqx_auth_ldap	EMQ X Authentication/ACL with LDAP	Stopped	Start
emqx_auth_mnesia	EMQ X Authentication with Mnesia	Stopped	Start
emqx_auth_mongo	EMQ X Authentication/ACL with MongoDB	Stopped	Start
emqx_auth_mysql	EMQ X Authentication/ACL with MySQL	Running	Stop

Finalmente si hicimos todo correctamente podremos conectarnos por WSS siempre agregando un usuario y contraseña y así llevaremos un control para evitar que cualquier persona sin unas credenciales de acceso correspondientes pueda conectarse a nuestro bróker



Para tener aun mayor control de quien puede suscribirse a nuestros tópicos podemos agregar reglas ACL para ello en nuestra base de datos crearemos la siguiente tabla

```

1 CREATE TABLE `mqtt_acl` (
2   `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
3   `allow` int(1) DEFAULT 1 COMMENT '0: deny, 1: allow',
4   `ipaddr` varchar(60) DEFAULT NULL COMMENT 'IpAddress',
5   `username` varchar(100) DEFAULT NULL COMMENT 'Username',
6   `clientid` varchar(100) DEFAULT NULL COMMENT 'ClientId',
7   `access` int(2) NOT NULL COMMENT '1: subscribe, 2: publish, 3: pubsub',
8   `topic` varchar(100) NOT NULL DEFAULT '' COMMENT 'Topic Filter',
9   PRIMARY KEY (`id`),
10  INDEX (ipaddr),
11  INDEX (username),
12  INDEX (clientid)
13 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```



Ejecutar la(s) consulta(s) SQL en la tabla admin\_emqx.mqtt\_user:

```
1 CREATE TABLE `mqtt_acl` (
2     `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
3     `allow` int(1) DEFAULT 1 COMMENT '0: deny, 1: allow',
4     `ipaddr` varchar(60) DEFAULT NULL COMMENT 'IpAddress',
5     `username` varchar(100) DEFAULT NULL COMMENT 'Usern' Cargando...
6     `clientid` varchar(100) DEFAULT NULL COMMENT 'ClientId',
7     `access` int(2) NOT NULL COMMENT '1: subscribe, 2: publish, 3: pubsub',
8     `topic` varchar(100) NOT NULL DEFAULT '' COMMENT 'Topic Filter',
9     PRIMARY KEY (`id`),
10    INDEX (ipaddr),
11    INDEX (username),
12    INDEX (clientid)
13 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Una vez creado la tabla agregaremos las siguientes consultas

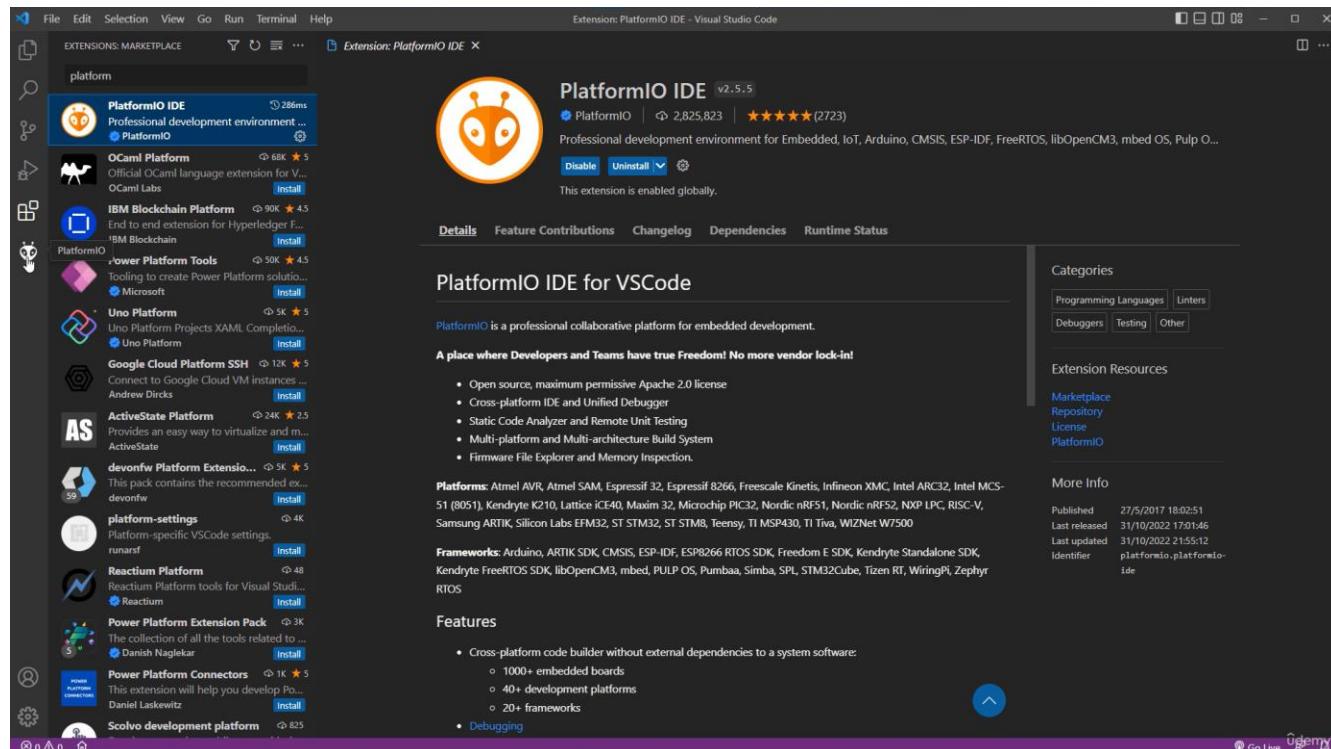
```
1 -- All users cannot subscribe to system topics
2 INSERT INTO mqtt_acl (allow, ipaddr, username, clientid, access, topic) VALUES (0, NULL, '$all', NULL, 1, '$SYS/#');
3
4 -- Allow clients on 10.59.1.100 to subscribe to system topics
5 INSERT INTO mqtt_acl (allow, ipaddr, username, clientid, access, topic) VALUES (1, '10.59.1.100', NULL, NULL, 1, '$SYS/#');
6
7 -- Deny client to subscribe to the topic of /smarthome/+temperature
8 INSERT INTO mqtt_acl (allow, ipaddr, username, clientid, access, topic) VALUES (0, NULL, '$all', NULL, 1, '/smarthome/+temperature');
9
10 -- Allow clients to subscribe to the topic of /smarthome/${clientid}/temperature with their own Client ID
11 INSERT INTO mqtt_acl (allow, ipaddr, username, clientid, access, topic) VALUES (1, NULL, '$all', NULL, 1, '/smarthome/%c/temperature');
```

Con estas reglas podemos establecer que usuario con que IP o con que id puede conectarse a ciertos tópicos y así tener una restricción y una capa más de seguridad (Cada una de estas reglas ya las explique antes con ejemplos en la [página 59](#) por lo que si tienes duda de que hace cada regla puedes consultarla ahí)



## 6 SECCION 7: CLIENTE ESP32

En este punto empezaremos a crear una plataforma con nuestro bróker recibiendo datos gracias a una ESP32 pero para ello antes de empezar es importante contar con el programa de Visual Estudio Code en nuestro equipo e instalarle una extensión llamada PlatformIO IDE [17]



Una vez tengamos instalado todo lo anterior mencionado procederemos a iniciar la extensión dando clic sobre ella y una vez abierta le daremos en crear un nuevo proyecto y llenaremos lo



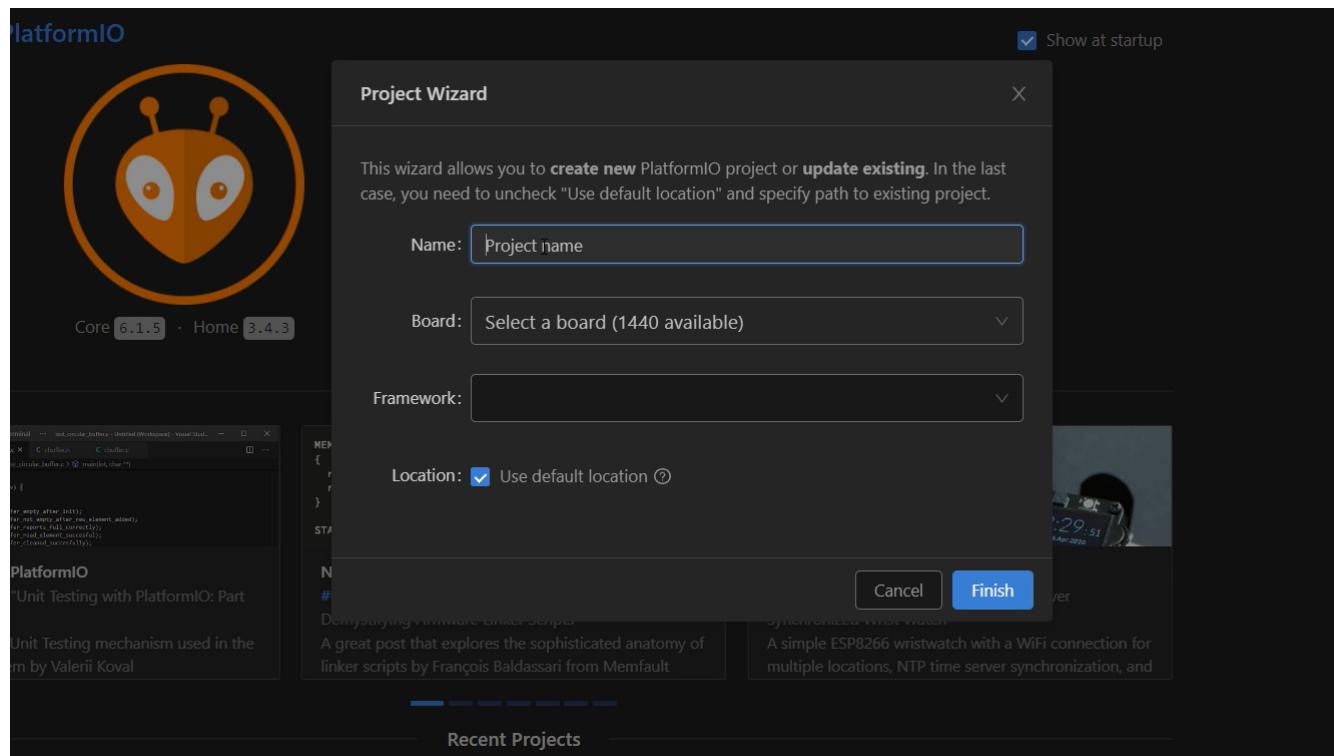
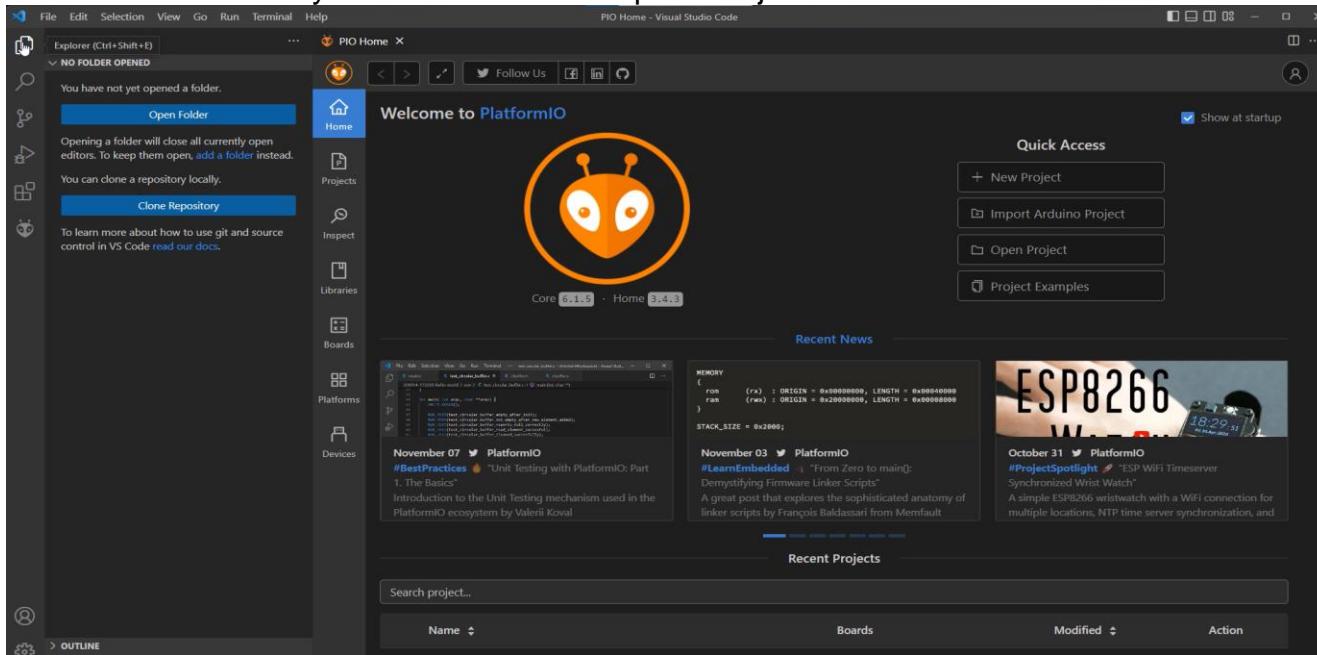
GOBIERNO DE  
MÉXICO



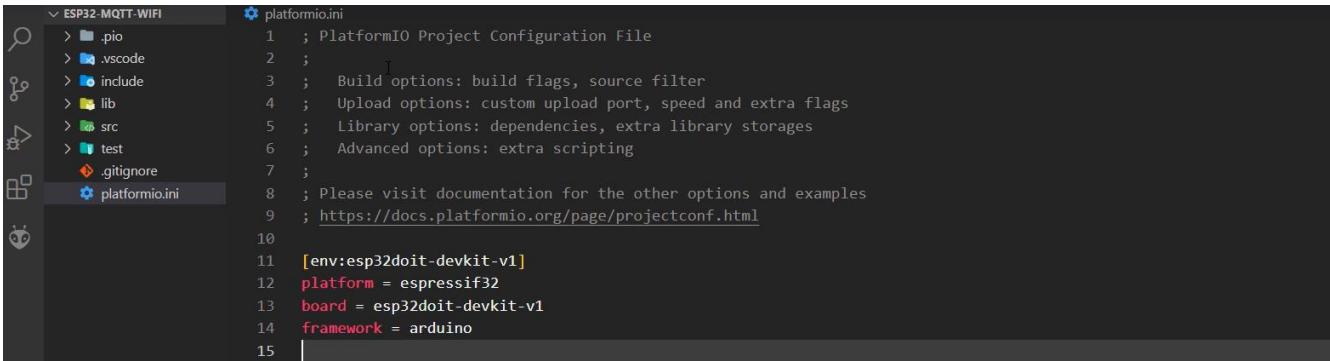
CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

datos como lo es el nombre de nuestro proyecto, buscaremos y seleccionaremos nuestra placa en mi caso la ESP32 y el framework con el que trabajaremos en este caso Arduino



Finalmente seleccionaremos el lugar donde queremos que se guarde nuestro proyecto y listo empezara a crear todo lo necesario para trabajar



```

ESP32-MQTT-WIFI
  - .pio
  - .vscode
  - include
  - lib
  - src
  - test
  - .gitignore
  - platformio.ini

platformio.ini
1 ; PlatformIO Project Configuration File
2 ;
3 ; Build options: build flags, source filter
4 ; Upload options: custom upload port, speed and extra flags
5 ; Library options: dependencies, extra library storages
6 ; Advanced options: extra scripting
7 ;
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:esp32doit-devkit-v1]
12 platform = espressif32
13 board = esp32doit-devkit-v1
14 framework = arduino
15

```

Una vez hecho esto en el archivo platformio.ini pondremos los siguientes comandos

```

> test
  - .gitignore
  - platformio.ini

platformio.ini
7 ;
8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:esp32doit-devkit-v1]
12 platform = espressif32
13 board = esp32doit-devkit-v1
14 framework = arduino
15
16 ; flags
17 build_flags =
18
19 ; Library options
20 lib_deps =
21
22 ; Serial Monitor options
23 monitor_speed = 115200
24 monitor_port = COM10
25
26 ; Upload options
27 upload_speed = 921600
28 upload_port = COM10

```

Agregamos las variables de entorno que vamos a utilizar

Agregamos el nombre de las librerías que vamos a utilizar

Agregamos el comando de monitor serial con estos valores

Agregamos el puerto de subida con estos valores



Después nos iremos a nuestro archivo main y ahí ya podremos empezar a desarrollar nuestro proyecto por ello mandaremos llamar la librería que nos permite conectarnos por WiFi y agregamos un código en el cual declaramos el nombre de nuestra red y su contraseña al la cual se conectara la placa [18]

```
C++ main.cpp > ...
1 #include <Arduino.h>
2 #include <WiFi.h>
3
4 // WIFI
5 const char *ssid = "iotmaster";           // Enter your WiFi name
6 const char *password = "iotmaster84";     // Enter WiFi password
7
8
```

Ahora declararemos unas variables necesarias para el bróker que el nombre del host, un tópico, el usuario y contraseña del bróker y por último el puerto por el que nos vamos a conectar a este bróker

```
7
8 // MQTT Broker
9 const char *mqtt_broker = "iothost.org";
10 const char *topic = "esp32/test";
11 const char *mqtt_username = "emqx";
12 const char *mqtt_password = "public";
13 const int mqtt_port = 1883;
14
15
```



Ahora crearemos un código para declarar a nuestro cliente WiFi y lista ya tendremos nuestra conexión pero también es importante que como lo mencione anteriormente poner las librerías que vamos a utilizar en el archivo **platformio.ini** de lo contrario nos darán errores

```
File Edit Selection View Go Run Terminal Help
main.cpp - ESP32-MQTT-WIFI - Visual Studio Code

EXPLORER      ... C++ main.cpp X platformio.ini
ESP32-MQTT-WIFI
> .pio
> .vscode
> include
> lib
> src
  > main.cpp
> test
  .gitignore
  platformio.ini

src > C++ main.cpp > ...
1 #include <Arduino.h>
2 #include <WiFi.h>
3 #include <PubSubClient.h>
4
5 // WIFI
6 const char *ssid = "iotmaster";           // Enter your WiFi name
7 const char *password = "iotmaster84";     // Enter WiFi password
8
9 // MQTT Broker
10 const char *mqtt_broker = "iothost.org";
11 const char *topic = "esp32/test";
12 const char *mqtt_username = "emqx";
13 const char *mqtt_password = "public";
14 const int mqtt_port = 1883;
15
16 WiFiClient mqttEmqx;
17 PubSubClient clinet(mqttEmqx);
18
19
20 void setup() {
21   // put your setup code here, to run once:
22 }
23
24 void loop() {
25   // put your main code here, to run repeatedly:
26 }
```

```
18
19 ; Library options
20 lib_deps =
21   knolleary/PubSubClient @ ^2.8
22
```



Una vez tengamos hecho sin ningún error vamos a crear una función que procese los mensajes que llegan por MQTT

```
15
16 WiFiClient mqttEmqx;
17 PubSubClient clinet(mqttEmqx);
18
19 // Declaración de Funciones
20 void callback(char *topic, byte *payload, unsigned int length);
21
```

Ahora en el apartado de serial declararemos la velocidad de la tarjeta y crearemos la conexión por WiFi y las conexiones para el bróker

```
15
16 WiFiClient mqttEmqx;
17 PubSubClient client(mqttEmqx);
18
19 // Declaración de Funciones
20 void callback(char *topic, byte *payload, unsigned int length);
21
22 void setup() {
23     // put your setup code here, to run once:
24     // Set software serial baud to 115200;
25     Serial.begin(115200);
26     // connecting to a WiFi network
27     WiFi.begin(ssid, password);
28     while (WiFi.status() != WL_CONNECTED) {
29         delay(500);
30         Serial.println("Connecting to WiFi..");
31     }
32     Serial.println("Connected to the WiFi network");
33     //connecting to a mqtt broker
34     client.setServer(mqtt_broker, mqtt_port);
35     client.setCallback(callback);|
36
37
```



Una vez creada la conexión ahora solo queda conectarnos y para ello utilizaremos el siguiente código

```
src > C++ main.cpp > ⌂ setup()
22 void setup() {
23     // put your setup code here, to run once:
24     // Set software serial baud to 115200;
25     Serial.begin(115200);
26     // connecting to a WiFi network
27     WiFi.begin(ssid, password);
28     while (WiFi.status() != WL_CONNECTED) {
29         delay(500);
30         Serial.println("Connecting to WiFi..");
31     }
32     Serial.println("Connected to the WiFi network");
33     //connecting to a mqtt broker
34     client.setServer(mqtt_broker, mqtt_port);
35     client.setCallback(callback);
36     while (!client.connected()) {
37         String client_id = "esp32-client-";
38         client_id += String(WiFi.macAddress());
39         Serial.printf("The client %s connects to the public mqtt broker\n", client_id.c_str());
40         if (client.connect(client_id.c_str(), mqtt_username, mqtt_password)) {
41             Serial.println("Public emqx mqtt broker connected");
42         } else {
43             Serial.print("failed with state ");
44             Serial.print(client.state());
45             delay(2000);
46         }
47     }
48 }
49
50
51 }
52
53 void loop() {
54     // put your main code here, to run repeatedly:
55 }
```



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CDESÍ®

Ahora vamos a publicar y suscribirnos al tópico que declaramos al principio

```
50 // publish and subscribe
51 client.publish(topic, "Hi EMQ X I'm ESP32 ^^");
52 client.subscribe(topic);
53
54
```

Una vez hecho esto crearemos la función callback que procesara los mensajes que nos llegaran y de ahí nos remos al aparado de loop donde pondremos un comando para garantizar que la conexión MQTT siempre este escuchando y llamando a la función callback

```
49 }
50 // publish and subscribe
51 client.publish(topic, "Hi EMQ X I'm ESP32 ^^");
52 client.subscribe(topic);
53 }
54
55 void callback(char *topic, byte *payload, unsigned int length) {
56     Serial.print("Message arrived in topic: ");
57     Serial.println(topic);
58     Serial.print("Message:");
59     for (int i = 0; i < length; i++) {
60         Serial.print((char) payload[i]);
61     }
62     Serial.println();
63     Serial.println("-----");
64 }
```



```
65
66 void loop() {
67     // put your main code here, to run repeatedly:
68     client.loop();
69 }
```

Por ultimo solo quedara subir el código a la ESP32 para ello en la parte de abajo le daremos clic a símbolo de check para ver si no hay un error y también le daremos clic al símbolo de una flecha para subir el código a la placa

The screenshot shows the PlatformIO IDE interface with the title "PlatformIO: Upload". The central area displays the build log for a project named "Default (ESP32-MQTT-WIFI)". The log output is as follows:

```
57 Serial.println(topic);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
○ - tool-mkfatfs @ 2.0.1
- tool-mklittlefs @ 1.203.210628 (2.3)
- tool-mkspiffs @ 2.230.0 (2.30)
- toolchain-xtensa-esp32 @ 8.4.0+2021r2-patch3
LDF: Library Dependency Finder -> https://bit.ly/configure-pio-ldf
LDF Modes: Finder ~ chain, Compatibility ~ soft
Found 33 compatible libraries
Scanning dependencies...
Dependency Graph
|-- PubSubClient @ 2.8.0
|--- WiFi @ 2.0.0
Building in release mode
Compiling .pio\build\esp32doit-devkit-v1\src\main.cpp.o
Generating partitions .pio\build\esp32doit-devkit-v1\partitions.bin
Compiling .pio\build\esp32doit-devkit-v1\lib69c\PubSubClient\PubSubClient.cpp.o
Compiling .pio\build\esp32doit-devkit-v1\lib568\WiFi\WiFi.cpp.o
Compiling .pio\build\esp32doit-devkit-v1\lib568\WiFi\WiFiAP.cpp.o
Compiling .pio\build\esp32doit-devkit-v1\lib568\WiFi\WiFiClient.cpp.o
Compiling .pio\build\esp32doit-devkit-v1\lib568\WiFi\WiFiGeneric.cpp.o
Compiling .pio\build\esp32doit-devkit-v1\lib568\WiFi\WiFiMulti.cpp.o
Compiling .pio\build\esp32doit-devkit-v1\lib568\WiFi\WiFiSTA.cpp.o
Compiling .pio\build\esp32doit-devkit-v1\lib568\WiFi\WiFiScan.cpp.o
Compiling .pio\build\esp32doit-devkit-v1\lib568\WiFi\WiFiServer.cpp.o
Compiling .pio\build\esp32doit-devkit-v1\lib568\WiFi\WiFiUpd.cpp.o
Compiling .pio\build\esp32doit-devkit-v1\FrameworkArduino\Esp.cpp.o
```

OUTLINE TIMELINE PlatformIO: Upload Default (ESP32-MQTT-WIFI) Ln 6

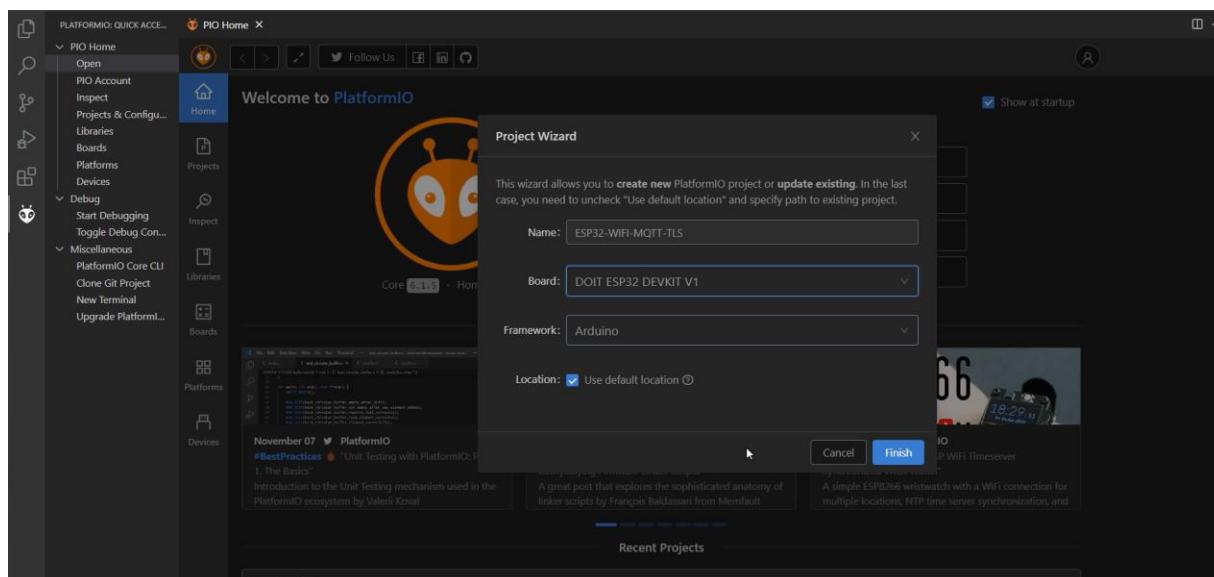
Si la conexión funciona nos saldrá un mensaje como este en consola y ya podremos mandarnos mensaje entre la placa y el bróker

```

entry 0x400000e4
Connecting to WiFi..
Connected to the WiFi network
The client esp32-client-0C:B8:15:DC:3E:70 connects to the public mqtt broker
Public emqx mqtt broker connected
    
```

## 6.1 MQTTY ESP32 por conexión segura

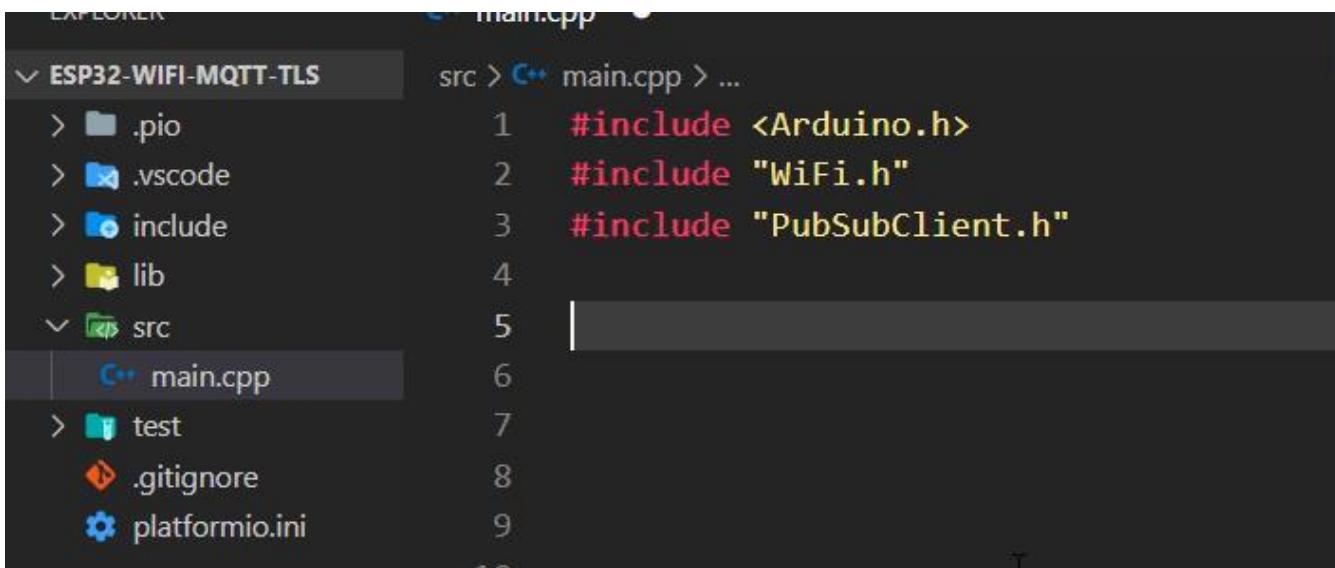
Creamos un nuevo archivo con la extensión platformio con los elementos que necesitamos y en el archivo platformio.ini ingresas los comandos que necesitamos como las librerías, etc. como ya lo vimos anteriormente



```

> .vscode           9 ; https://docs.platformio.org/page/projectconf.html
> include          10
> lib              11 [platformio]
> src              12 default_envs = esp32doit-devkit-v1
> test             13
> .gitignore       14 [env:esp32doit-devkit-v1]
> platformio.ini   15 platform = espressif32
                   16 board = esp32doit-devkit-v1
                   17 framework = arduino
                   18
                   19 ; flags
                   20 build_flags =
                   21     '-D WIFI_SSID="iotmaster"'
                   22     '-D WIFI_PASSWORD="iotmaster84"'
                   23
                   24 ; Library options
                   25 lib_deps =
                   26     knolleary/PubSubClient @ ^2.8
                   27
                   28 ; Serial Monitor options
                   29 monitor_speed = 115200    I
                   30 monitor_port = COM10
                   31
                   32 ; Upload options
                   33 upload_speed = 921600
                   34 upload_port = COM10
                   35
  
```

En nuestro archivo main lo que haremos será empezar a crear nuestro código por ello lo primero que haremos será importar las librerías de WiFi y las se PubSubClient



The screenshot shows the VS Code interface with the Explorer sidebar open. The project structure is displayed under 'ESP32-WIFI-MQTT-TLS'. The 'src' folder contains 'main.cpp'. The code editor shows the beginning of the 'main.cpp' file:

```

src > main.cpp > ...
1 #include <Arduino.h>
2 #include "WiFi.h"
3 #include "PubSubClient.h"
4
5
6
7
8
9
  
```

De igual manera volveremos a definir el bróker con los parámetros que ya vimos anteriormente pero con el único detalle que en el puerto de conexión al bróker lo cambiaremos por el 8883 ya que este es el puerto de conexión segura



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

```
1 #include <Arduino.h>
2 #include "WiFi.h"
3 #include "PubSubClient.h"
4
5 // MQTT Broker
6 const char *mqtt_broker = "iothost.org";
7 const char *topic = "esp32/test";
8 const char *mqtt_username = "emqx";
9 const char *mqtt_password = "public";
10 const int mqtt_port = 8883;
11
12
```



Ahora vamos a definir un Led para ver cómo funciona el recibir órdenes desde el bróker así como también definimos que queremos realizar una conexión por TLS así como agregar la librería que nos permite las conexión de manera segura

```
5 // MQTT Broker
6 const char *mqtt_broker = "iothost.org";
7 const char *topic = "esp32/test";
8 const char *mqtt_username = "emqx";
9 const char *mqtt_password = "public";
10 const int mqtt_port = 8883;
11
12 #define MQTT_MAX_MESSAGE_SIZE 1024
13
14 #define LED 26
15
16 #define TLS
17
18 #ifdef TLS
19     #include <WiFiClientSecure.h>
20 #endif
21
```



Después vamos a definir la instancia de conexión segura y definir el certificado pero para ello tendremos que tener el certificado a la mano

```
17
18 #ifdef TLS
19     #include <WiFiClientSecure.h>
20 #endif
21
22 #ifdef TLS
23     WiFiClientSecure esp_client;
24 #endif
25
26 PubSubClient client(esp_client);
27
28
```



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Si no tienes el certificado ve a tu panel de Hestia y en el apartado de Web entraremos a nuestro dominio y ahí en el apartado de mostrar certificados nos saldrán todos los que tenemos pero el que necesitamos es el último el SSL Certificate Authority pero solo la primera parte

### SSL Certificate Authority / Intermediate (Optional)

```
-----BEGIN CERTIFICATE-----  
MIIFjCCA...  
-----END CERTIFICATE-----
```



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

## SSL Certificate Authority / Intermediate (Optional)

-----BEGIN CERTIFICATE-----

```
MIIFFjCCAv6gAwIBAgIRAJErCErPDBinU/bWLiWnX1owDQYJKoZIhvcNAQELBQAw
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFNlY3VyaXR5IFJlc2Vh
cmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwHhcNMjAwOTA0MDAwMDAw
WhcNMjUwOTE1MTYwMDAwWjAyMQswCQYDVQQGEwJVUzEWMBQGA1UEChMNTGV0J3Mg
RW5jcnlwdDELMAkGA1UEAxAMCUjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQCTAhUozPaglNMPEuyNVZLD+ILxmaZ6QoinXSaqtSu5xUyxr45r+XXIo9cP
R5QUVTVXjJ6oojkZ9YI8Qql0bvU7wy7bjcCwXPN20ftz2nwWgsbvsCUJCWH+jdx
sxPnHKzhm+/b5DtFUkWWqcFTzjTIUu61ru2P3mBw4qVUq7ZtDpelQDRrK908Zutm
NHz6a4uPVymZ+DAXXbpyb/uBxa3Shlg9F8fnCbvxK/eG3MHacV3URuPMrSXBiLxg
Z3Vms/EY96Jc51P/Ooi2R6X/ExjqmA13P51T+c8B5fWmcBcUr20k/5mzk53cU6cG
/kfHaFpriV1uxPMUgP17VGhi9sVAgMBAAGjggEIMIIBBDAOBgNVHQ8BAf8EBAMC
AYYwHQYDVR01BBYwFAYIKwYBBQUHawIGCCsGAQUFBwMBMBIGA1UdEwEB/wQIMAYB
Af8CAQAwHQYDVR00BBYEFBQusxe3WFbLrlAJQOYfr52LFMLGMB8GA1UdIwQYMBaA
FHm0WeZ7tuXkAXOACIjIGlj26ZtuMDIGCCsGAQUFBwEBBCYwJDAiBggrBgfEFBQcw
AoYWaHR0cDovL3gxLmkubGVuY3Iub3JnLzAnBgNVHR8EIDAeMBygGqAYhhZodHRw
Oi8veDEuYy5sZW5jci5vcmcvMCIGA1UdIAQbMBkwCAYGZ4EMAQIBMA0GCysGAQQB
gt8TAQEBMA0GCSqGSIb3DQEBCwUA4ICAQCFyk5HPqP3hUSFvNVneLKYY611TR6W
PTNlclQtgaDqw+34IL9fzLdwALduO/ZeI1N7kJI+m74uyA+eitRY8kc607TkC53w1
ikfmZW4/RvTZ8M6UK+5UzhK8jCdLuMGYL6KvzXGRSGi3yLgjewQtCPkIVz6D2QQz
CkcheAmCJ8MqyJu5zlzyZMjAvnnAT45tRAxelsru94sQ4egdRCnbWSDtY7kh+BI
1JNXoB11BMEKIq4QDUOXoRgffuDghje1WrG9ML+Hbisq/yFOGwXD9RiX8F6sw6W4
avAuvDsue5L3sz85K+EC4Y/wFVDNvZo4TYXao6Z0f+lQKc0t8DQYzk1OXVu8rp2
yJMC6allbBfODALZvYH7n7do1AZls4I9d1P4jnkDrQoxB3UqQ9hVl3LEKQ73xF10
yK5GhDDX8oVfGKF5u+decIsH4YaTw7mP3GFxJSqv3+01UFJoi5Lc5da149p90Ids
hCExroL1+7mryIkXPeFM5Tg09r0rvZaBF0vV2z0gp35Z0+L4WP1buEjN/lxPFin+
HlUjr8gRsI3qfJOQFy/9rKIJR0Y/80mwt/8oTWgy1mdeHmmjk7j1nYsvC9JSQ6Zv
Mld1TTKB3zhThV1+XWYp6rjd5JW1zbVWEkLNxE7GJThEUG3szgBVGP7pSWTUTsqX
nLrbwHOoq7hHwg==
```

-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----

```
MIIFYDCBElgAwIBAgIQQQAF3ITfU6UK47naqPGQKtzANBgkqhkiG9w0BAQsFADA/
MSQwTzYDVQK5Eyt5oWdndCEsTENa725hdUVv7SPUenVz-dCRDby4x5-AVR-nVRAmT
```



Una vez que tengas el certificado solo lo pegamos en nuestro archivo

```
26 PubSubClient client(esp_client);
27
28 #ifdef TLS
29 const char *ca_cert = // \
30     "-----BEGIN CERTIFICATE-----\n"
31     "MIIFFjCCAv6gAwIBAgIRAJErCErPDBinU/bWLiWnX1owDQYJKoZIhvcNAQELBQAw\n"
32     "TzELMAkGA1UEBhMCVVmxkTAnBgNVBAoTIEludGVybmv0IFN1Y3VyaXR5IFJ1c2Vh\n"
33     "cmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwHhcNMjAwOTA0MDAwMDAw\n"
34     "WhcNMjUwOTE1MTYwMDAwWjAyMQswCQYDVQQGEwJVUzEWMBQGA1UEChMNTGV0J3Mg\n"
35     "Rw5jcmlwdDELMAkGA1UEAxMCUjMwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK\n"
36     "AoIBAQCTAhUozPag1NMPeuyNVZLD+ILxmaZ6QoinXsaqtSu5xUyxr45r+XXIo9cP\n"
37     "R5QUVTVXjJ6ojkZ9YI8Qql0bvU7wy7bjcCwXPNZOOftz2nwWgsbvsCUJCWH+jdx\n"
38     "sxPnHKzhm+/b5DtFUkWqcfTzjTIUu61ru2P3mBw4qVUq7ZtDpe1QRrK908Zutm\n"
39     "NHZ6a4uPVymZ+DAXXbpyb/uBxa3Shlg9F8fnCbvxK/eG3MHacV3URuPMrSXBilxg\n"
40     "Z3Vms/EY96Jc51P/Ooi2R6X/ExjqmA13P51T+c8B5fWmcBcUr20k/5mzk53cU6cG\n"
41     "/kiFHaFpriV1uxPMUgP17VGhi9sVAgMBAAGjggEIMIIBBDA0BgvNHQ8BAf8EBAMC\n"
42     "AYYwHQYDVR01BBYwFAYIKwYBBQUHawIGCCsGAQUFBwMBMBIGA1UdEwEB/wQIMAYB\n"
43     "Af8CAQAwHQYDVR0OBBYEFBQusxe3WFbLr1AJQOYfr52LFMLGMB8GA1UdIwQYMBaA\n"
44     "FHm0WeZ7tuXkAXOACIJIG1j26ZtuMDIGCCsGAQUFBwEBBCYwJDAiBgrBgfEFBQcw\n"
45     "AoYWaHR0cDovL3gxLmkubGVuY3Iub3JnLzAnBgNVHR8EIDAeMBygGqAYhhZodHRw\n"
46     "Oi8veDEUy5szW5jci5vcmcvMCIGA1UdIAQbMBkwCAYGZ4EMAQIBMA0GCysGAQQB\n"
47     "gt8TAQEBMA0GCSqGSIb3DQEBCwUA4ICAQCFyk5HPqP3hUSFvNVneLKYY611TR6W\n"
48     "PTNlc1QtgaDqw+34IL9fzLdwALdu0/Ze1N7kIJ+m74uyA+eitRY8kc607TkC53w1\n"
49     "ikfmZW4/RvTZ8M6UK+5UzhK8jCdLuMGYL6KvzXGRSGi3yLgjewQtCPkIVz6D2Qz\n"
50     "CkcheAmCJ8MqyJu5z1zyZMjAvnnAT45tRAxelsru94sQ4egdRCnbWSDtY7kh+BIm\n"
51     "1JNxoB1lBMEKIq4QDUOXoRgffuDghje1WrG9ML+Hbisq/yFOGwXD9RiX8F6sw6W4\n"
52     "avAuvDszue5L3sz85K+EC4Y/wFVDNvZo4TYXao6Z0f+1QKc0t8DQYzk10XVu8rp2\n"
53     "yJMC6a1LbBfODALZvYH7n7do1AZls4I9d1P4jnkDrQoxB3UqQ9hV13LEKQ73xF10\n"
54     "yK5GhDDX8oVfGKF5u+decIsH4YaTw7mP3GFxJSqv3+01UFJoi5Lc5da149p90Ids\n"
55     "hCExroL1+7mryIkXPefM5Tg09r0rvZaBFovV2z0gp35Z0+L4WP1buEjN/lxPFin+\n"
56     "H1Ujr8gRsI3qfJOQFy/9rKIJR0Y/80mwtt/8oTwgy1ndeHmmjk7j1nYsvC9JSQ6Zv\n"
57     "Mld1TTKB3zhThV1+XWYp6rjd5JW1zbVWEkLNxE7GJThEUG3szgBVGP7pSWTUTsqX\n"
```



Después lo que haremos será agregar la función callback que vimos anteriormente y un código para que pueda realizarse las reconexiones de ser necesario

```
60  #endif
61
62  uint8_t connection_intention_init = 0;
63  uint8_t cont_intet_conect;
64
65  void callback(char *topic, byte *payload, unsigned int length)
66  {
67      String incoming = "";
68      Serial.print("Mensaje recibido desde =>");
69      Serial.print(topic);
70      Serial.println("");
71      for (int i = 0; i < length; i++)
72      {
73          incoming += (char)payload[i];
74      }
75      incoming.trim();
76      Serial.println("Mensaje=> " + incoming);
77
78      if (incoming == "on")
79      {
80          digitalWrite(LED, HIGH);
81      }
82      else
83      {
84          digitalWrite(LED, LOW);
85      }
86  }
```



```
86    }
87
88 void reconnect()
89 {
90     while (!client.connected())
91     {
92         Serial.println("Intentando conexion Mqtt...");
93         String clienteId = "esp32_";
94         clienteId += String(WiFi.macAddress());
95         if (client.connect(clienteId.c_str(), mqtt_username, mqtt_password))
96         {
97             Serial.print("Conectado --> Code: ");
98             Serial.println(client.state());
99             if (client.subscribe(topic))
100             {
101                 Serial.printf("subscrito a topico %s\r\n", topic);
102             }
103         }
104     else
105     {
106         Serial.print("Falló con error => Code: ");
107         Serial.println(client.state());
108         Serial.println("Intentamos de nuevo en 5 segundos");
109         delay(5000);
110     }
111 }
112 }
113 }
```



Ahora pondremos el siguiente código en el setup configurado que la ESP32 se conecte de manera segura al bróker y prenda un LED como prueba de su conexión por puerto seguro

```
114 void setup() {
115     // Seteo de frecuencia esp32 240 MHZ
116     setCpuFrequencyMhz(240);
117     // Inicializador del puerto serial
118     Serial.begin(115200);
119     // PIN de Salida
120     pinMode(LED, OUTPUT);
121     digitalWrite(LED, LOW);
122     // WIFI
123     uint8_t cont_intent = 0;
124     WiFi.mode(WIFI_AP_STA);
125     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
126     while (WiFi.status() != WL_CONNECTED)
127     {
128         connection_intention_init++;
129         delay(500);
130         Serial.println("Connecting to WiFi..");
131         if (connection_intention_init == 40)
132         {
133             Serial.println("No se pudo conectar a la red espere por favor ");
134             connection_intention_init = 0;
135             cont_intent++;
136             if (cont_intent == 5)
137             {
138                 Serial.println("No se pudo conectar a la red reiniciando el dispositivo");
139                 delay(1000);
140                 ESP.restart();
141             }
142         }
```



```
116     setcpufrequencymhz(240);
117     // Inicializador del puerto serial
118     Serial.begin(115200);
119     // PIN de Salida
120     pinMode(LED, OUTPUT);
121     digitalWrite(LED, LOW);
122     // WIFI|
123     uint8_t cont_intent = 0;
124     WiFi.mode(WIFI_AP_STA);
125     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
126     while (WiFi.status() != WL_CONNECTED)
127     {
128         connection_intention_init++;
129         delay(500);
130         Serial.println("Connecting to WiFi..");
131         if (connection_intention_init == 40)
132         {
133             Serial.println("No se pudo conectar a la red espere por favor ");
134             connection_intention_init = 0;
135             cont_intent++;
136             if (cont_intent == 5)
137             {
138                 Serial.println("No se pudo conectar a la red reiniciando el dispositivo");
139                 delay(1000);
140                 ESP.restart();
141             }
142         }
143     }
144     Serial.print("El dispositivo se conecto exitosamente su direccion IP es -> ");
145     Serial.println(WiFi.localIP());
146     delay(200);
147     Serial.printf("Conectado a la red -> %s\r\n", WiFi.SSID().c_str());
148     delay(200);
149
150
```

Ahora definiremos la conexión al bróker pero como esta vez lo queremos hacer por TLS haremos el código así

```

150
151     #ifdef TLS
152         esp_client.setCACert(ca_cert);
153         client.setServer(mqtt_broker, mqtt_port);
154     #endif
155     client.setCallback(callback);
156
157 }
158
159
160     client.setCallback(callback);
161
162     while (!client.connected())
163     {
164         Serial.println("Intentando conexion Mqtt...");
165         String clienteId = "esp32_";
166         clienteId += String(WiFi.macAddress());
167         if (client.connect(clienteId.c_str(), mqtt_username, mqtt_password))
168         {
169             Serial.print("Conectado --> Code: ");
170             Serial.println(client.state());
171             if (client.subscribe(topic))
172             {
173                 Serial.printf("subscrito a topico %s\r\n", topic);
174             }
175         }
176         else
177         {
178             cont_intet_conect++;
179             Serial.print("Fallo con error => Code: ");
180             Serial.println(client.state());
181             Serial.println("Intentamos de nuevo en 5 segundos");
182             delay(5000);
183             if (cont_intet_conect == 10)
184             {
185                 Serial.println("No se pudo conectar verifique la conexion");
186             }
187         }
188     }
189 }
```



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Ya solo en el loop podremos el siguiente código para que si existen problemas de conexión este se reconecte automáticamente y repita todo el código en bucle

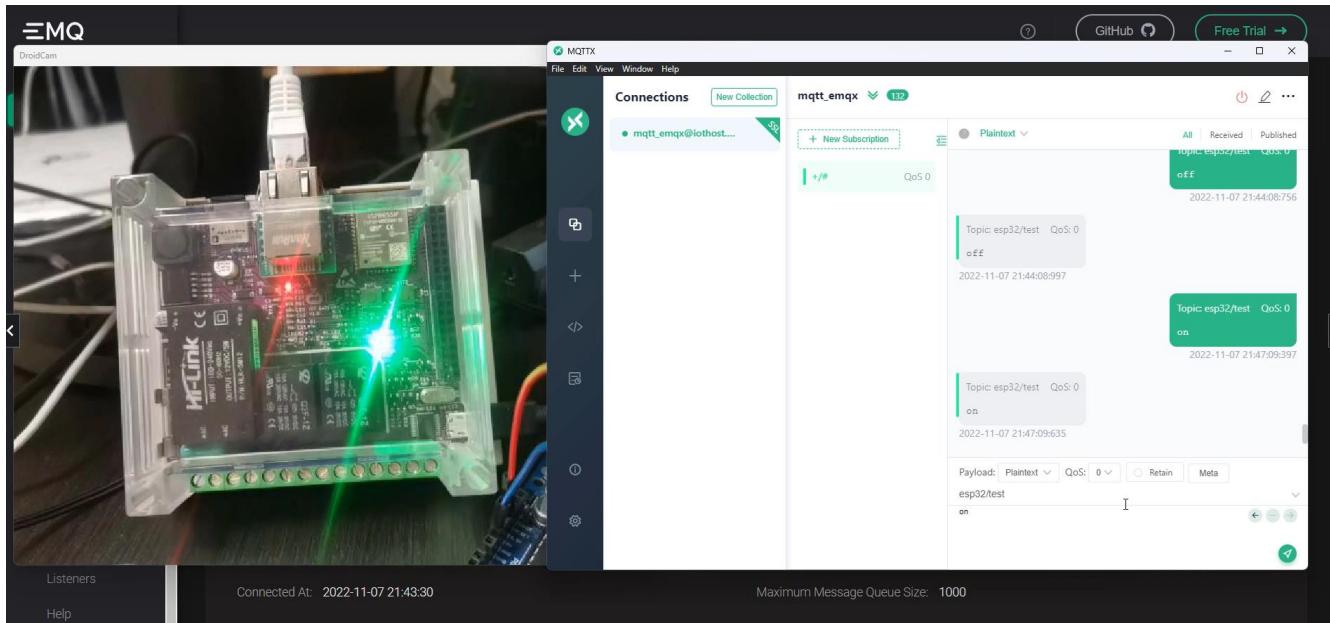
```
186 void loop() {
187     if (!client.connected())
188     {
189         reconnect();
190     }
191     client.loop();
192 }
```

```
main.cpp ->
src > C++ main.cpp > loop()
1 #include <Arduino.h>
2 #include "WiFi.h"
3 #include "PubSubClient.h"
4
5 // MQTT Broker
6 const char *mqtt_broker = "iothost.org";
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
* Executing task: C:\Users\yamir-hidalgo\.platformio\penv\Scripts\platformio.exe device monitor
--- Terminal on COM10 | 115200 8-N-1
--- Available filters and text transformations: colorize, debug, default, direct, esp32_exception_decoder, hexlify, log2file, nocontrol, printable, send_on_enter, time
--- More details at https://bit.ly/pio-monitor-filters
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H
ets Jun 20 2023 15:57:07

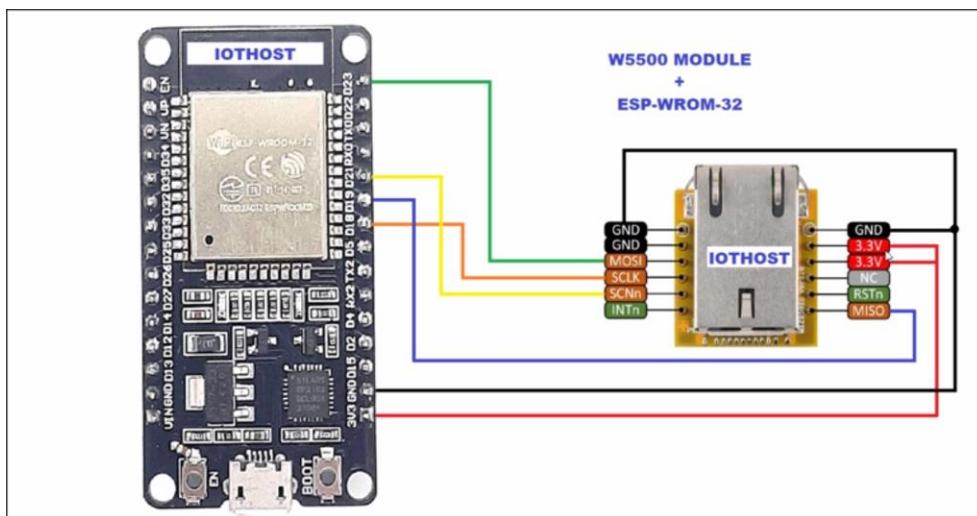
rst:0x1 (POWERON_RESET),boot:0x1,B0%_NO_FLASH_BOOT)
config:ip: 0, SPIWP:0xee
clk@0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff003bb@1184
load:0x40078000,len:12784
load:0x40080400@0000032
entry 0x400805e4
Connecting to WiFi..
El dispositivo se conectó exitosamente su dirección IP es -> 192.168.20.108
Conectado a la red -> iotmaster
Intentando conexión Mqtt...
Conectado --> Code: 0
suscripto a topico esp32/test
```



Como podemos ver podemos mandar mensajes entre el bróker y la tarjeta de manera asegura incluso podemos atreves del cliente decirle a un LED que se apague o se prenda



Este código sirve para conectarse al bróker EMQX por el módulo WiFi que tiene la ESP32 por lo que si por alguna necesidad del proyecto necesitamos tener una conexión Ethernet podemos seguir la misma lógica del código anterior cambiando solo que la conexión sea por un puerto Ethernet conectado a la placa en lugar de usar el módulo WiFi (Dejo el código con este cambio y el diagrama de conexiones del puerto Ethernet a la placa)



## CODIGO PARA CONEXIÓN WIFI SEGURA TLS [19]

```
#include <Arduino.h>
#include "WiFi.h"
#include "PubSubClient.h"

// MQTT Broker
const char *mqtt_broker = "broker.org";
const char *topic = "esp32/test";
const char *mqtt_username = "user";
const char *mqtt_password = "password";
const int mqtt_port = 8883;

#define MQTT_MAX_MESSAGE_SIZE 1024

#define LED 26

#define TLS

#ifdef TLS
    #include <WiFiClientSecure.h>
#endif

#ifdef TLS
    WiFiClientSecure esp_client;
#endif

PubSubClient client(esp_client);

#ifdef TLS
const char *ca_cert = // \n
"AQUI PONES TUS CERTIFICADO";

#endif

uint8_t connection_intention_init = 0;
uint8_t cont_intet_conect;

void callback(char *topic, byte *payload, unsigned int length)
{
    String incoming = "";
    Serial.print("Mensaje recibido desde =>");


```



```
Serial.print(topic);
Serial.println("");
for (int i = 0; i < length; i++)
{
    incoming += (char)payload[i];
}
incoming.trim();
Serial.println("Mensaje=> " + incoming);

if (incoming == "on")
{
    digitalWrite(LED, HIGH);
}
else
{
    digitalWrite(LED, LOW);
}
}

void reconnect()
{
    while (!client.connected())
    {
        Serial.println("Intentando conexion Mqtt... ");
        String clientId = "esp32_";
        clientId += String(WiFi.macAddress());
        if (client.connect(clientId.c_str(), mqtt_username, mqtt_password))
        {
            Serial.print("Conectado --> Code: ");
            Serial.println(client.state());
            if (client.subscribe(topic))
            {
                Serial.printf("subscrito a topico %s\r\n", topic);
            }
        }
        else
        {
            Serial.print("Falló con error => Code: ");
            Serial.println(client.state());
            Serial.println("Intentamos de nuevo en 5 segundos");
            delay(5000);
        }
    }
}
```



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

}

```
void setup() {
    // Seteo de frecuencia esp32 240 MHZ
    setCpuFrequencyMhz(240);
    // Inicializador del puerto serial
    Serial.begin(115200);
    // PIN de Salida
    pinMode(LED, OUTPUT);
    digitalWrite(LED, LOW);
    // WIFI
    uint8_t cont_intent = 0;
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

    while (WiFi.status() != WL_CONNECTED)
    {
        connection_intention_init++;
        delay(500);
        Serial.println("Connecting to WiFi..");
        if (connection_intention_init == 40)
        {
            Serial.println("No se pudo conectar a la red espere por favor ");
            connection_intention_init = 0;
            cont_intent++;
            if (cont_intent == 5)
            {
                Serial.println("No se pudo conectar a la red reiniciando el dispositivo");
                delay(1000);
                ESP.restart();
            }
        }
    }
    Serial.print("El dispositivo se conecto exitosamente su dirección IP es -> ");
    Serial.println(WiFi.localIP());
    delay(200);
    Serial.printf("Conectado a la red -> %s\r\n", WiFi.SSID().c_str());
    delay(200);

#ifndef TLS
    esp_client.setCACert(ca_cert);
    client.setServer(mqtt_broker, mqtt_port);
#endif
}
```



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

```
client.setCallback(callback);

while (!client.connected())
{
    Serial.println("Intentando conexion Mqtt... ");
    String clientId = "esp32_";
    clientId += String(WiFi.macAddress());
    if (client.connect(clientId.c_str(), mqtt_username, mqtt_password))
    {
        Serial.print("Conectado --> Code: ");
        Serial.println(client.state());
        if (client.subscribe(topic))
        {
            Serial.printf("subscrito a topico %s\r\n", topic);
        }
    }
    else
    {
        cont_intet_conect++;
        Serial.print("Fallo con error => Code: ");
        Serial.println(client.state());
        Serial.println("Intentamos de nuevo en 5 segundos");
        delay(5000);
        if (cont_intet_conect == 10)
        {
            Serial.println("No se pudo conectar verifique la conexion");
        }
    }
}
void loop() {
    if (!client.connected())
    {
        reconnect();
    }
    client.loop();
}
```

## CODIGO PARA CONEXIÓN POR PUERTO ETHERNET [19]

```
#include <Arduino.h>
#include <Ethernet.h>
#include "PubSubClient.h"

// MQTT Broker
const char *mqtt_broker = "broker.org";
const char *topic = "esp32/test";
const char *mqtt_username = "user";
const char *mqtt_password = "password";
const int mqtt_port = 1883;

#define LED 32

byte mac[] = { 0x70, 0x69, 0x69, 0x2D, 0x30, 0x31 };

void callback(char *topic, byte *payload, unsigned int length)
{
    String incoming = "";
    Serial.print("Mensaje recibido desde =>");
    Serial.print(topic);
    Serial.println("");
    for (int i = 0; i < length; i++)
    {
        incoming += (char)payload[i];
    }
    incoming.trim();
    Serial.println("Mensaje=> " + incoming);

    if (incoming == "on")
    {
        digitalWrite(LED, HIGH);
    }
    else
    {
        digitalWrite(LED, LOW);
    }
}

EthernetClient ethClient;
PubSubClient client(mqtt_broker, mqtt_port, callback, ethClient);
```

```

void reconnect()
{
    while (!client.connected())
    {
        Serial.println("Intentando conexion Mqtt... ");
        String clientId = "esp32_ethernet";
        if (client.connect(clientId.c_str()), mqtt_username, mqtt_password))
        {
            Serial.print("Conectado --> Code: ");
            Serial.println(client.state());
            if (client.subscribe(topic))
            {
                Serial.printf("subscrito a topico %s\r\n", topic);
            }
        }
        else
        {
            Serial.print("Falló con error => Code: ");
            Serial.println(client.state());
            Serial.println("Intentamos de nuevo en 5 segundos");
            delay(5000);
        }
    }
}

void setup() {
    // Seteo de frecuencia esp32 240 MHZ
    setCpuFrequencyMhz(240);
    // Inicializador del puerto serial
    Serial.begin(115200);
    // PIN de Salida
    pinMode(LED, OUTPUT);
    digitalWrite(LED, LOW);
    // Ethernet
    Ethernet.init(21); // CS pin 21
    Ethernet.begin(mac); // use DHCP

    // Check for Ethernet hardware present
    if (Ethernet.hardwareStatus() == EthernetNoHardware) {
        Serial.println("Ethernet shield was not found. Sorry, can't run without hardware. :(");
        while (true) {
            delay(1); // do nothing, no point running without Ethernet hardware
    }
}

```



```
    }
}

if (Ethernet.linkStatus() == LinkOFF) {
    Serial.println("Ethernet cable is not connected.");
}

Serial.print("IP: ");
Serial.println(Ethernet.localIP());

delay(1500);
}

void loop() {
    if (!client.connected())
    {
        reconnect();
    }
    client.loop();
}
```

## 7 SECCION 9: CLIENTE PHP

Lo primero que vamos a hacer es abrir nuestro Laragon y una vez de ahí nos iremos a la carpeta donde se guardan los proyectos y nuestro documento y ahí crearemos un nuevo proyecto

B (.) > laragon > www >			
Nombre	Fecha de modificación	Tipo	Tamaño
api-iohost	30/7/2022 18:36	Carpeta de archivos	
mqtt-php	10/11/2022 20:03	Carpeta de archivos	
php-mqtt	6/11/2022 21:13	Carpeta de archivos	



Abriremos la carpeta en nuestro editor de texto favorito y luego realizaremos la instalación de una librería para php y MQTT para ello primero verificaremos que tengamos Composer en nuestro equipo de lo contrario tendremos que instalarlo (ya sea para Windows o Linux)

[Home](#) | [Getting Started](#) | [Download](#) | [Documentation](#) | [Browse Packages](#)

## Download Composer Latest: v2.5.8

### Windows Installer

The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

### Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === 'e21205b207c3ff031906575712edab6f13eb0b361f2085f1f
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

This installer script will simply check some `php.ini` settings, warn you if they are set incorrectly, and then download the latest `composer.phar` in the current directory. The 4 lines above will, in order:

Una vez sepamos que tenemos Composer pondremos el siguiente comando en la consola para descargar la librería de php para MQTT **composer require bluerhinos/phpmqtt=@dev**

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS J:\laragon\www\mqtt-php> composer --version
Composer version 2.4.4 2022-10-27 14:39:29
PS J:\laragon\www\mqtt-php>
PS J:\laragon\www\mqtt-php>
PS J:\laragon\www\mqtt-php>
PS J:\laragon\www\mqtt-php> composer require bluerhinos/phpmqtt=@dev
Info from https://repo.packagist.org: #StandWithUkraine
./composer.json has been created
Running composer update bluerhinos/phpmqtt
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
- Locking bluerhinos/phpmqtt (dev-master fe4b6b2)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Installing bluerhinos/phpmqtt (dev-master fe4b6b2): Extracting archive
Generating autoload files
No security vulnerability advisories found
PS J:\laragon\www\mqtt-php>
```



Después de instalar la librería se creara una carpeta llamada vendor y en esa carpeta crearemos un archivo llamado mqtt.php y en el pondremos los siguientes códigos que son para mandar llamar la librería y la creación de la conexión con MQTT a nuestro servidor

```
git mqtt.php > ...
1  <?php
2
3  require('vendor/autoload.php');
4
5  use \Bluerhinos\phpMQTT;
6
7  // Conexion MQTT
8  $server = 'iothost.org';
9  $port = 1883;
10 $username = 'emqx';
11 $password = 'public';
12 $clientId = 'phpMQTT-' . rand(5,15);
13
14 $c = new phpMQTT($server, $port, $clientId, null);
15
16
17
18
```

```
1  use \Bluerhinos\phpMQTT;
2
3
4
5
6
7  // Conexion MQTT
8  $server = 'iothost.org';
9  $port = 1883;
10 $username = 'emqx';
11 $password = 'public';
12 $clientId = 'phpMQTT-' . rand(5,15);
13 $topicPublish = 'phpClient/test';
14 $topicSubscribe = 'phpClient/subscribe';
15
16 $c = new phpMQTT($server, $port, $clientId, null);
17
18 $conn_result = $c -> connect(false, null, $username, $password);
19
20 if ($conn_result) {
21     $c->publish($topicPublish, 'Hello World from PHP Client', 0, false);
22     $topics[$topicSubscribe] = array("qos"=>0, "function"=>"procmsg");
23     $c ->subscribe($topics);
24 }
25
```

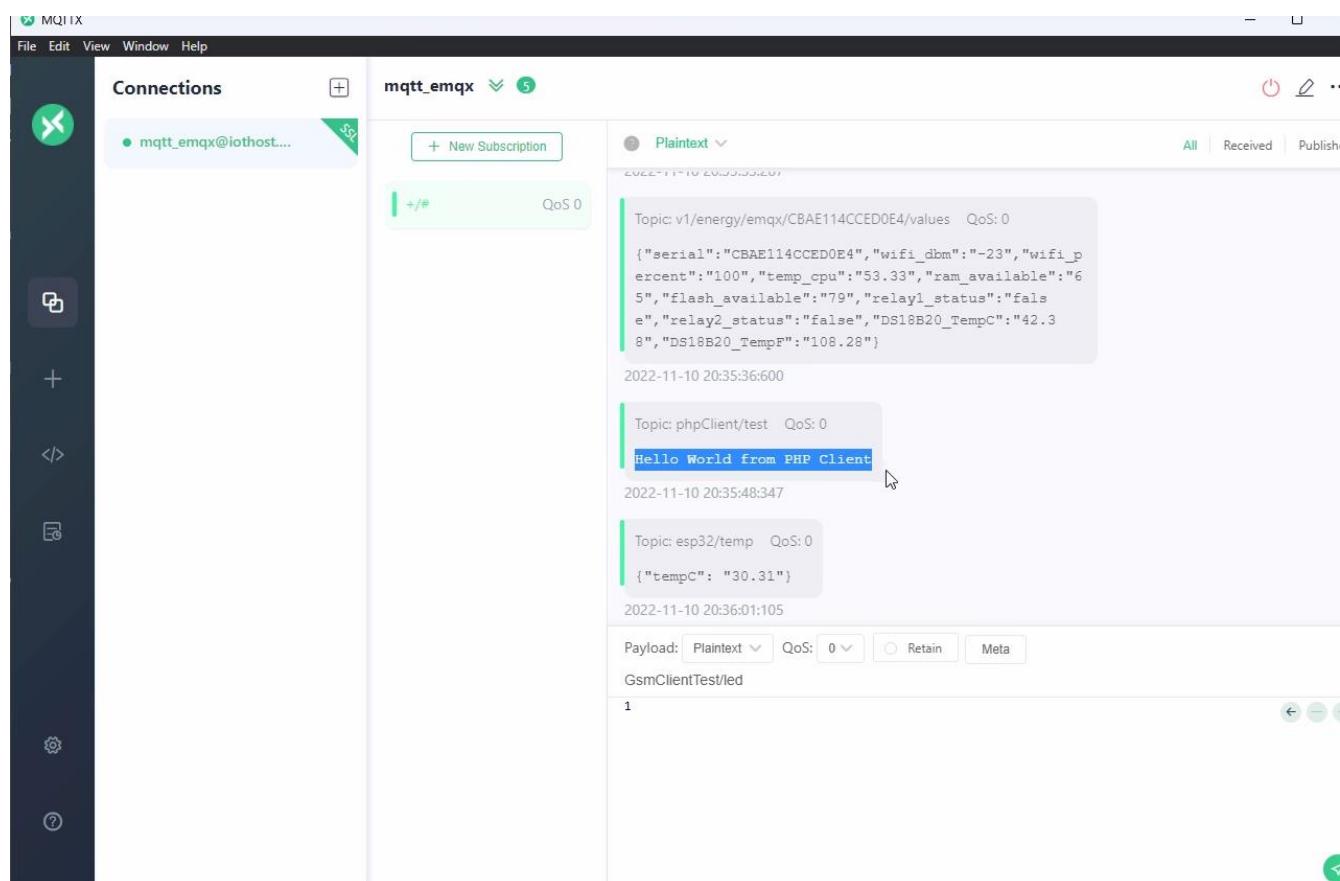
Después crearemos una función que mande a llamar el topic y así poder mostrar un mensaje así como dejaremos un comando para estar escuchando constantemente los mensajes de los tópicos

```

25
26   function procmsg($topic, $msg){
27     echo 'topic: '.$topic.' msg: '.$msg;
28   }
29
30   while($c->proc()){
31   }

```

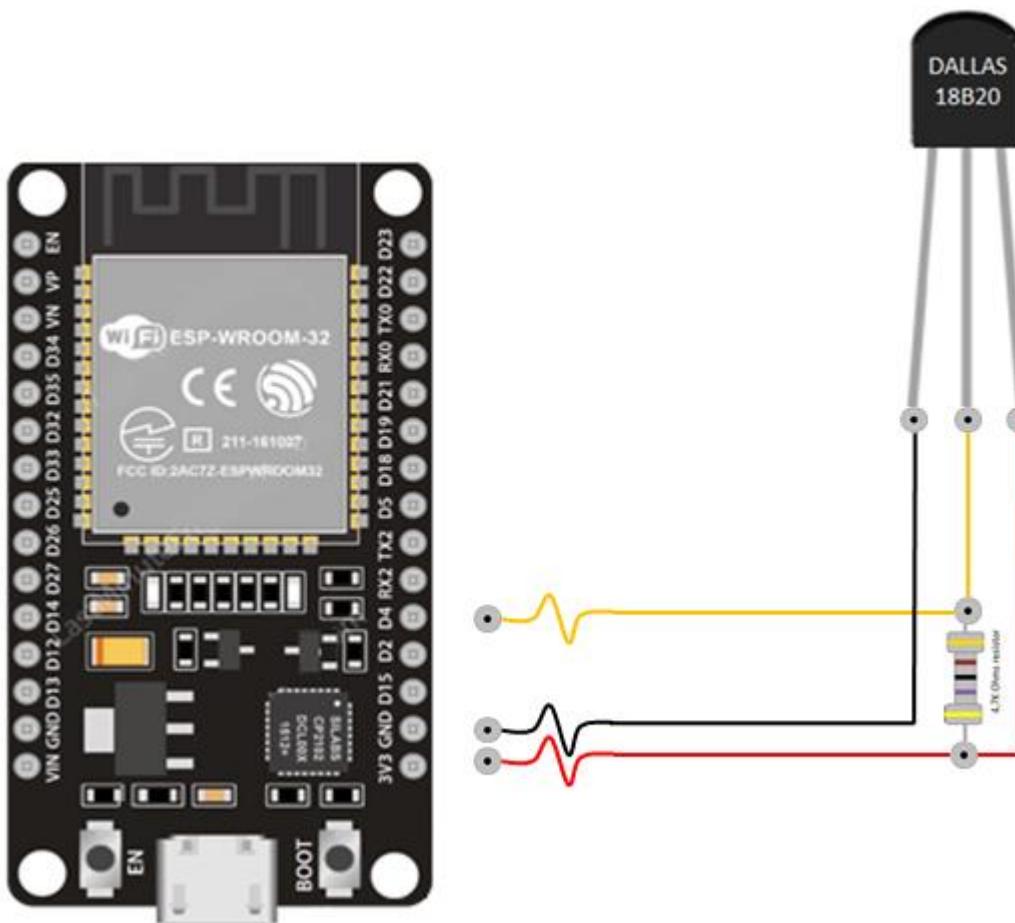
Después abriremos nuestro cliente MQTTX y una vez abierto entraremos en nuestra terminal de Visual Code y ahí pondremos el siguiente comando **php mqtt.php** y si nos conectamos con nuestro cliente nos saldrá los mensajes entre php y nuestro bróker



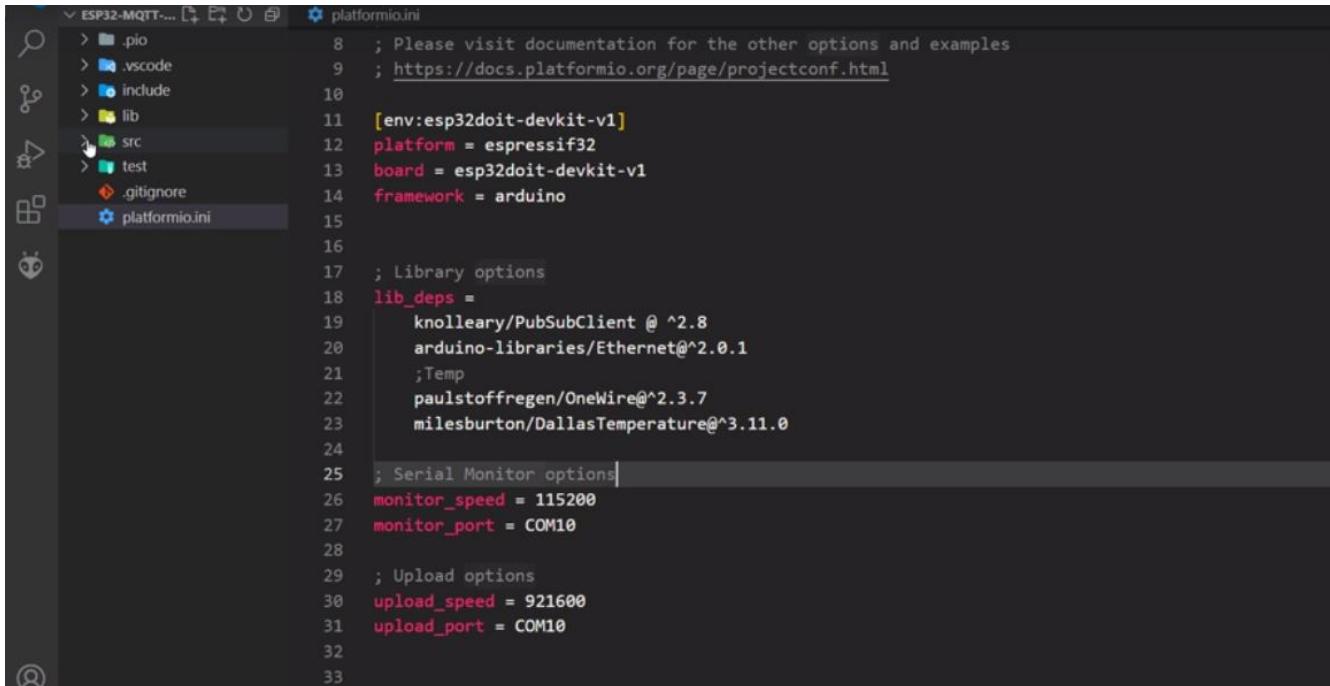


## 7.1 Cliente PHP con ESP32 con envío de temperatura por MQTT

Una vez ya tengamos funcionando nuestro bróker con PHP ahora haremos un pequeño proyecto para ver cómo podemos comunicar datos por MQTT por ello vamos a mandar datos de temperatura a nuestro bróker y los almacenaremos en una base de datos por ello tendremos que agregar el sensor de temperatura a la tarjeta en este caso yo utilizare el **ds18b20**



Una vez ensamblada la tarjeta con el sensor de temperatura, crearemos un nuevo proyecto con la extensión platformio como hemos visto anteriormente y una vez creado nos iremos al platformio.ini y agregaremos las librerías y comandos que necesitemos



```

ESP32-MQTT-...
platformio.ini

8 ; Please visit documentation for the other options and examples
9 ; https://docs.platformio.org/page/projectconf.html
10
11 [env:esp32doit-devkit-v1]
12 platform = espressif32
13 board = esp32doit-devkit-v1
14 framework = arduino
15
16
17 ; Library options
18 lib_deps =
19   knolleary/PubSubClient @ ^2.8
20   arduino-libraries/Ethernet@^2.0.1
21   ;Temp
22   paulstoffregen/OneWire@^2.3.7
23   milesburton/DallasTemperature@^3.11.0
24
25 ; Serial Monitor options
26 monitor_speed = 115200
27 monitor_port = COM10
28
29 ; Upload options
30 upload_speed = 921600
31 upload_port = COM10
32
33

```

Después de eso nos iremos a nuestro archivo main y ahí colocaremos todo el código correspondiente para que la placa pueda leer la información del sensor así como crear la conexión con el bróker y así poder mandar los datos de la temperatura que esta capturado el sensor por ello yo dejo este código que lo que hace es capturar la temperatura y mostrarla en el cliente (En caso de que requiera otra cosa más que un sensor de temperatura solo sería cuestión de cambiar el código que permite capturar la temperatura y poner el código de lo que necesitamos hacer así como el tópico que estamos usando y ya el resto del código se queda igual ya que es la conexión y el envío de los datos capturados )



Cuando tengamos el código listo solo lo guardaremos en la tarjeta y lo ejecutaremos y como podremos ver se realiza la conexión con el bróker y cada minuto nos estará saliendo el valor de la temperatura en la consola y en el cliente

The screenshot shows a development environment with three main panes:

- Code Editor:** Displays the C++ code for an ESP32 temperature sensor. The code includes a loop that requests temperature from a sensor, checks if the reading is successful, prints the temperature to the Serial port, packages the data into a JSON string, and publishes it to the topic `esp32/temp`. The code also handles disconnection from the broker.
- Terminal:** Shows the output of the code execution. It starts with boot logs, followed by the message "Intentando conexión Mqtt...", "Conectado --> Code: 0", "subscripto a topico esp32/test", "Tomando valores de Temperatura...", and finally "Temperatura del sensor 1 (index 0) es: 30.".
- MQTT Client:** Shows the connection to the `mqtt_emqx` broker. A single message is listed under the topic `esp32/temp`: `{"tempC": "30.25"}`, timestamped at 2022-11-10 20:52:28:596. The message payload is shown as "Hola EMQX".



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

## CODIGO PARA LEER VALORES DE TEMPERATURA Y MANDARLOS AL BROKER EMQX

```
#include <Arduino.h>
#include <Ethernet.h>
#include "PubSubClient.h"
// Temp
#include <OneWire.h>
#include <DallasTemperature.h>

// MQTT Broker
const char *mqtt_broker = "tuserver.org";
const char *topic = "esp32/test";
const char *mqtt_username = "user";
const char *mqtt_password = "password";
const int mqtt_port = 1883;

// Publicar Temp
const char* topic_temp = "esp32/temp";
long lastMsgMQTT = 0;

#define LED 32

// Temp
#define ONE_WIRE_BUS 13

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

byte mac[] = { 0x70, 0x69, 0x69, 0x2D, 0x30, 0x31 };

void callback(char *topic, byte *payload, unsigned int length)
{
    String incoming = "";
    Serial.print("Mensaje recibido desde =>");
    Serial.print(topic);
    Serial.println("");
    for (int i = 0; i < length; i++)
    {
        incoming += (char)payload[i];
    }
    incoming.trim();
    Serial.println("Mensaje=> " + incoming);
}
```



```
if (incoming == "on")
{
    digitalWrite(LED, HIGH);
}
else
{
    digitalWrite(LED, LOW);
}

EthernetClient ethClient;
PubSubClient client(mqtt_broker, mqtt_port, callback, ethClient);

void reconnect()
{
    while (!client.connected())
    {
        Serial.println("Intentando conexion Mqtt...");
        String clientId = "esp32_ethernet";
        if (client.connect(clientId.c_str(), mqtt_username, mqtt_password))
        {
            Serial.print("Conectado --> Code: ");
            Serial.println(client.state());
            if (client.subscribe(topic))
            {
                Serial.printf("subscrito a topico %s\r\n", topic);
            }
        }
        else
        {
            Serial.print("Falló con error => Code: ");
            Serial.println(client.state());
            Serial.println("Intentamos de nuevo en 5 segundos");
            delay(5000);
        }
    }
}

// Envio de datos de temp por MQTT
void mqtt_publish() {
    // leer valores del sensor
    Serial.println("Tomando valores de Temperatura...");
```



```
sensors.requestTemperatures();
float tempC = sensors.getTempCByIndex(0);

// Check if reading was successful
if (tempC != DEVICE_DISCONNECTED_C)
{
    Serial.print("Temperatura del sensor 1 (index 0) es: ");
    Serial.println(tempC);
    // empaquetar datos y enviar por MQTT json
    // {"tempC" : "24.5"}
    String mqtt_data = "{\"tempC\": \""+String(tempC)+"\"}";
    client.publish(topic_temp, mqtt_data.c_str());
}
else
{
    Serial.println("Error: no se puede leer la Temperatura");
}

void setup() {
    // Seteo de frecuencia esp32 240 MHZ
    setCpuFrequencyMhz(240);
    // Inicializador del puerto serial
    Serial.begin(115200);
    // PIN de Salida
    pinMode(LED, OUTPUT);
    digitalWrite(LED, LOW);
    // Ethernet
    Ethernet.init(21); // CS pin 21
    Ethernet.begin(mac); // use DHCP

    // Check for Ethernet hardware present
    if (Ethernet.hardwareStatus() == EthernetNoHardware) {
        Serial.println("Ethernet shield was not found. Sorry, can't run without hardware. :(");
        while (true) {
            delay(1); // do nothing, no point running without Ethernet hardware
        }
    }
    if (Ethernet.linkStatus() == LinkOFF) {
        Serial.println("Ethernet cable is not connected.");
    }
}
```



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

```
Serial.print("IP: ");
Serial.println(Ethernet.localIP());

delay(1500);

// iniciar el sensor de temp
sensors.begin();
}

void loop() {
if (!client.connected())
{
    reconnect();
}
client.loop();

// Publicar cada 60 segundos 1 min
if (client.connected()){
    if (millis() - lastMsgMQTT > 20000){
        lastMsgMQTT = millis();
        mqtt_publish();
    }
}
}
```



## 7.2 Almacenar datos de Temperatura en BD desde MQTT

Lo primero que vamos a hacer es cambiar o agregar el tópico al cual nos debemos suscribir para recibir la temperatura en mi caso mi tópico se llama **esp32/temp**

```
8 $server           = 'iothost.org';
9 $port             = 1883;
10 $username        = 'emqx';
11 $password         = 'public';
12 $clientId         = 'phpMQTT-' . rand(5, 15);
13 $topicPublish     = 'phpClient/test';
14 $topicSubscribe   = 'esp32/temp';
15
```

Ahora vamos a modificar la función msg para poder procesar los mensajes y ver que nos está llegando

```
25 // Función para manejar mensajes entrantes
26 function procmsg($topic, $msg){
27     //echo 'topic: '.$topic.' msg: '.$msg;
28
29     //$json = json_decode($msg, true);
30     $json = json_decode($msg);
31     // {"tempC": "30.31"}
32     //$send = $json["tempC"];
33     $send = $json->tempC;
34
35     echo 'msg: ',$send;
36
```

Ahora solo abriremos una terminal y ejecutaremos el comando **php mqtt.php** y veremos como empieza a mandar mensajes a nuestro cliente

Topic: esp32/temp QoS: 0

{"tempC": "30.31"}

2022-11-10 21:04:28:622

Ahora que nos está llegando las lecturas de la temperatura al bróker lo que tenemos que hacer es crear la conexión a la base de datos por ello vamos a definir el host, el nombre de la bd, usuario y password

```

34
35     echo 'msg: '.$sendDB;
36
37     $host      = 'localhost';
38     $dbname    = 'iothost';
39     $username  = 'root';
40     $password  = '';
41

```

Ahora para establecer la conexión vamos a usar un try y crear todo lo necesario para poder conectarnos a la base de datos

```

try {

    $conn = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);

    echo "Connected to $dbname at $host successfully";

    $cmd = $conn->prepare('INSERT INTO daros VALUES(NULL,?,?)');

    $cmd->execute(array($topic, $sendDB));

    $conn = null;

}

} catch (PDOException $error) {
    die("Could not connect to the DB: ". $error->getMessage());
}

```



```
51
52     $conn = null;
53
54 } catch (PDOException $error) {
55     die("Could not connect to the DB: ". $error->getMessage());
56 }
57
58 }
```

Ahora vamos a abrir nuestra base de datos y vamos a crear una tabla llamada datos y que tenga un campo id, topic, msg

The screenshot shows the 'Básico' tab of MySQL Workbench's table creation interface. The table name is 'datos'. The columns are defined as follows:

#	Nombre	Tipo de datos	Longitud/Co...	Sin signo	Permitir NULL	Rellenar con ceros	Predeterminado	Comentario	Collation
1	id	INT	11	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	AUTO_INCREMENT	Sin valor predeter...	utf8_spanish_ci
2	topic	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Sin valor predeter...	
3	msg	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		Sin valor predeter...	utf8_spanish_ci



Una vez hecho esto solo queda iniciar php con el comando que ya hemos visto y después dde algunos minutos podemos ver como siguen llegando los datos al bróker y que igualmente estos se guardan en la base de datos

The screenshot shows a code editor with PHP code for an MQTT client, a terminal window running PHP, and a MySQL HeidiSQL interface.

**Code Editor:**

```
mqtt.php > ...
45
46     echo "Connected to $dbname";
47
48     $cmd = $conn->prepare('INSERT INTO `datos`(`topic`, `msg`) VALUES (?, ?)');
49     $cmd->execute(array($topic, $msg));
50
51     $conn = null;
52
53 } catch (PDOException $error) {
54     die("Could not connect to database");
55 }
56
57
58 }
59 // Mantener la conexión
60 while($c->proc()){
61 }
```

**Terminal:**

```
PS J:\laragon\www\mqtt-php> ^
PS J:\laragon\www\mqtt-php>
PS J:\laragon\www\mqtt-php>
PS J:\laragon\www\mqtt-php> php mqtt.php
msg: 30.31Connected to iothost at localhost
```

**MySQL HeidiSQL Interface:**

localhost.datos: 2 filas en total (aproximadamente)

id	topic	msg
1	esp32/temp	30.31
2	esp32/temp	30.31

SQL Query Log (71-77):

```
71 SELECT `id`, LEFT(`topic`, 256), LEFT(`msg`, 256) FROM `iothost`.`datos` LIMIT 1000
72 SELECT `id`, LEFT(`topic`, 256), LEFT(`msg`, 256) FROM `iothost`.`datos` LIMIT 1000
73 SELECT `id`, LEFT(`topic`, 256), LEFT(`msg`, 256) FROM `iothost`.`datos` LIMIT 1000
74 SELECT `id`, LEFT(`topic`, 256), LEFT(`msg`, 256) FROM `iothost`.`datos` LIMIT 1000
75 SELECT `id`, LEFT(`topic`, 256), LEFT(`msg`, 256) FROM `iothost`.`datos` LIMIT 1000
76 SELECT `id`, LEFT(`topic`, 256), LEFT(`msg`, 256) FROM `iothost`.`datos` LIMIT 1000
77 SELECT `id`. LEFT(`topic`. 256). LEFT(`mse`. 256) FROM `iothost`.`datos` LIMIT 1000
```

Topic: esp32/temp QoS: 0

```
{"tempC": "30.31"}
```

2022-11-10 21:22:37:265

Topic: esp32/temp QoS: 0

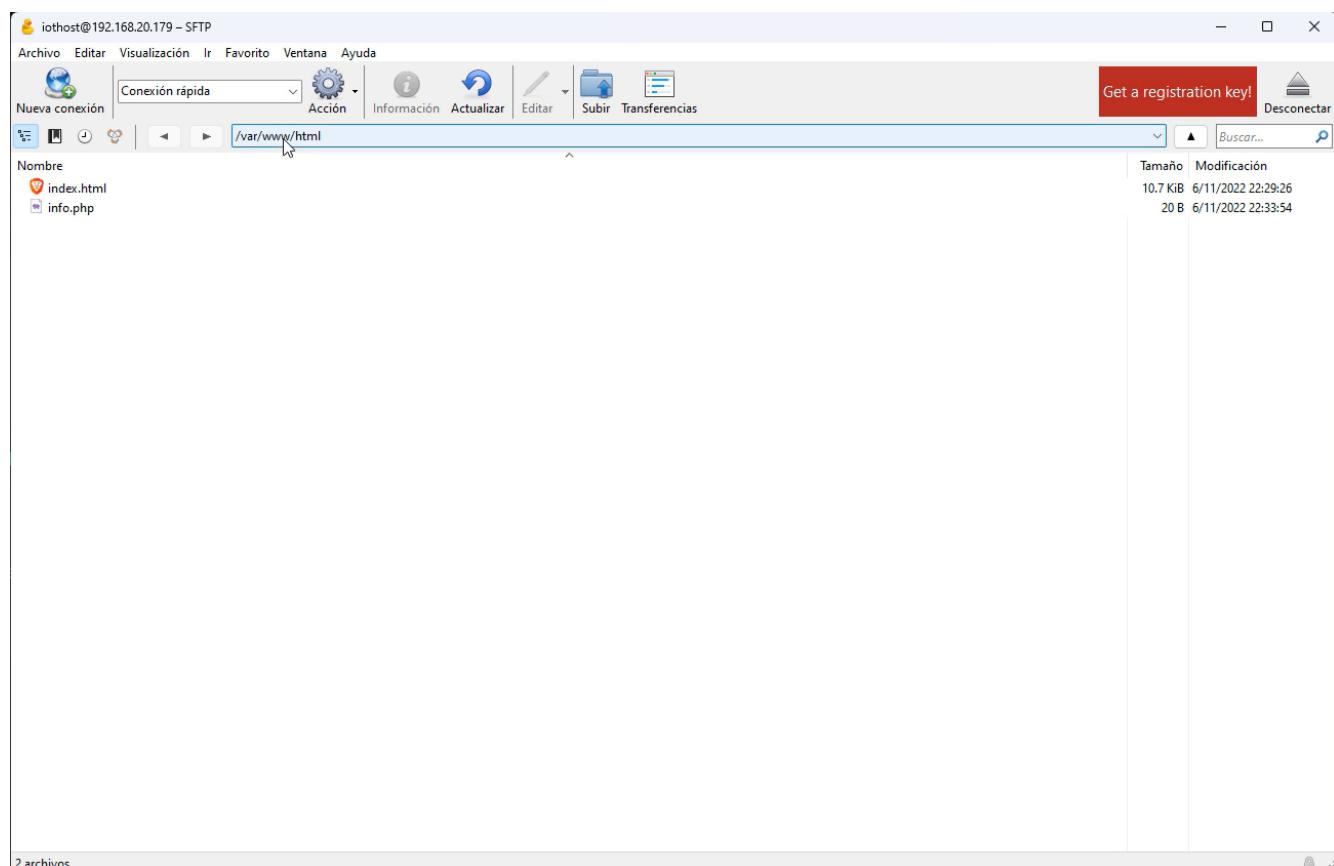
```
{"tempC": "30.31"}
```

2022-11-10 21:22:57:661



## 7.3 Cliente PHP del lado del servidor para guardar en BD

Ahora vamos a usar nuestro servidor para crear un servicio que almacene los datos en una base de datos cada vez que deseemos o programemos ya que como lo hicimos el paso anterior este solo guardara 5 veces los datos y de ahí en fuera no lo hará mas por lo que necesitaremos hacerlo en nuestro servidor por ello abriremos nuestro CyberDuck y nos conectaremos a nuestro servidor y nos situaremos en /var/www/html (carpeta publica de html)

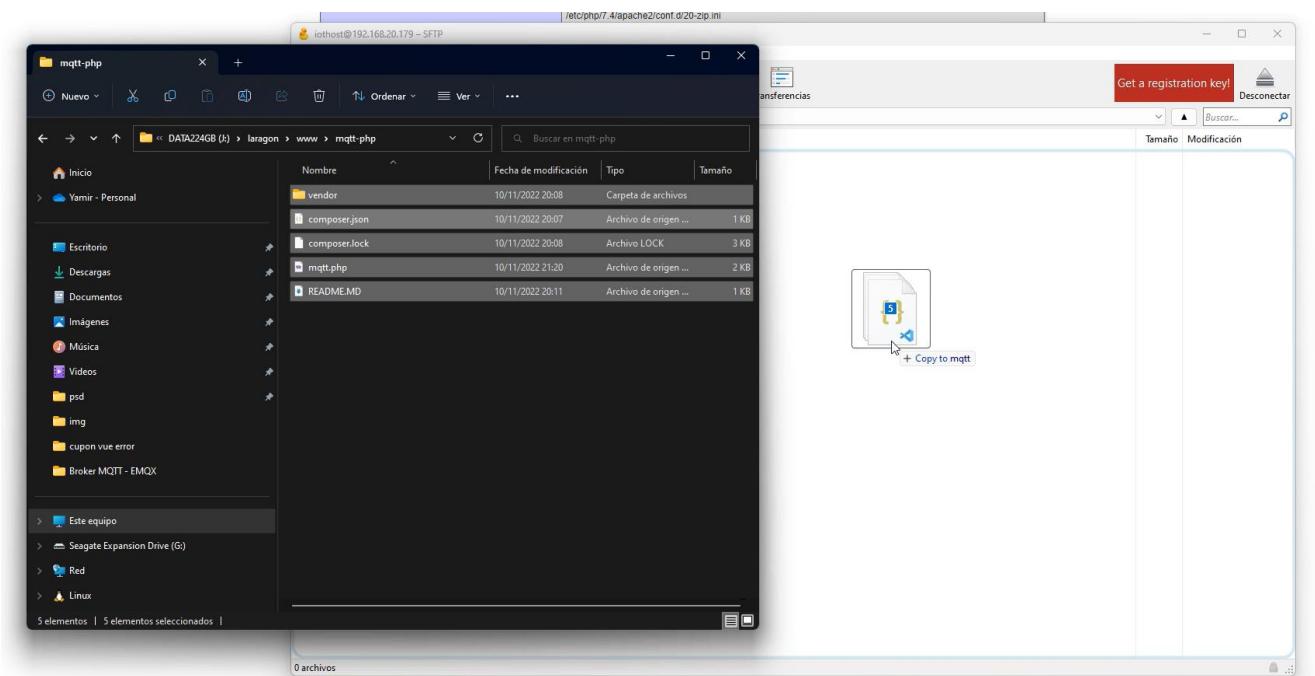


Dentro de esta dirección vamos a crear una carpeta que se llame mqtt





Una vez creada la carpeta tendremos que pasarle a esta nuestro proyecto que hemos realizado en Laragon por ello arrastremos todo el proyecto a la carpeta mqtt



Ahora abriremos nuestro Putty y nos conectaremos al servidor y una vez conectados nos iremos a esta dirección /var/www/html/mqtt y con el comando nano mqtt.php editaremos las variables de conexión como ponerle contraseña al usuario

```
root@iothost:/var/www/html/mqtt
GNU nano 4.8                         mqtt.php
?php

require('vendor/autoload.php');

use \Bluerhinos\phpMQTT;

// Conexion MQTT
$server      = 'iothost.org';
$port        = 1883;
$username   = 'emqx';
$password   = 'public';
$clientID   = 'phpMQTT-' . rand(5,15);
$topicPublish = 'phpClient/test';
$topicSubscribe = 'esp32/temp';

$c = new phpMQTT($server, $port, $clientID, null);
```



```
echo 'msg: '.$sendDB;

$host      = 'localhost';
$dbname   = 'iothost';
$username = 'root';
$password = '12345678';

try {
```

Una vez hecho esto solo iniciaremos sesión en nuestro phpMyAdmin y ahí ya tendremos toda la base de datos migrada por lo que solo sería en nuestro PUTTY poner el comando **php mqtt.php** y listo el programa ya estaría funcionando en nuestro servidor y guardaría los datos en la nube

The screenshot shows the phpMyAdmin interface. On the left, the database structure is visible with the 'datos' table under the 'iothost' schema. In the main area, the 'datos' table is selected, showing one row with the following data:

	id	topic	msg
<input type="checkbox"/>	19	esp32/temp	30.19

Below the table, there are buttons for Editar, Copiar, and Borrar.



Ahora para dejarlo funcionando toda la vida este servicio tendremos que usar los siguientes comandos que son para abrir y editar crontab usaremos el comando crontab –e

```
root@iohost:/var/www/html/mqqt          -  X
GNU nano 4.8                               /tmp/crontab.oel33Y/crontab           Modified
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m. every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
1 * * * * [REDACTED]
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^\_ Go To Line M-E Redo

Una vez ahí usaremos una página que ajustando los parámetros de días, horas, meses etc. Nos dará el código para modificar el tiempo de ejecución de un comando [20]

crontab guru

The quick and simple editor for cron schedule expressions by [Cronitor](#)

1 \* \* \* \*

minute	hour	day	month	day
		(month)		(week)
*				any value
,				value list separator
-				range of values
/				step values
1-12				allowed values
JAN-DEC				alternative single values



Por ello vamos a poner el código que nos puso la página para el tiempo de ejecución del comando luego una / y quien ejecutara este comando en mi caso php por lo que tendremos que poner donde se encuentra php y por ultimo de donde vamos a llamar al archivo en este caso de la carpeta mqtt y con esto el comando quedaría así

1\*\*\* **/usr/bin/php /var/www/html/mqtt/mqtt.php**

```
root@iohost:/var/www/html/mqtt
GNU nano 4.8                               /tmp/crontab.oel33Y/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
1 * * * * /usr/bin/php /var/www/html/mqtt/mqtt.php
```

Y finalmente veremos cómo dependiendo el tiempo que pusimos el comando se ejecutara automáticamente y capturara los datos en la base de datos

+ Opciones					
	← →	▼	id	topic	msg
<input type="checkbox"/>			19	esp32/temp	30.19
<input type="checkbox"/>			20	esp32/temp	30.31
<input type="checkbox"/>			21	esp32/temp	30.31
<input type="checkbox"/>			22	esp32/temp	30.31
<input type="checkbox"/>			23	esp32/temp	30.25
<input type="checkbox"/>			24	esp32/temp	30.25



## 8 SECCION 11: CLIENTE MQTT CON NODEJS

Lo primero que vamos a hacer es instalar Nodejs para nuestro equipo desde su página oficial <https://nodejs.org/es>

The screenshot shows the Node.js download page. At the top, there's a navigation bar with links: INICIO, ACERCA, DESCARGAS, DOCUMENTACIÓN, PARTICIPE, SEGURIDAD, NOTICIAS, and CERTIFICATION. Below the navigation bar, the word "node" is displayed with its logo. A large green button labeled "LTS Recomendado para la mayoría" contains links for "Instalador Windows" (node-v18.12.1-x64.msi) and "Instalador macOS" (node-v18.12.1.pkg). To the right, a grey box labeled "Actual Últimas características" contains a link for "Código Fuente" (node-v18.12.1.tar.gz). Below these are two tables:

LTS	Actual	
Instalador Windows (.msi)	32-bit	64-bit
Binario Windows (.zip)	32-bit	64-bit
Instalador macOS (.pkg)		64-bit / ARM64
Binario macOS (.tar.gz)	64-bit	ARM64
Binario Linux (x64)		64-bit

### 8.1 Registro de un dominio

### 8.2 Registro de un dominio



## 9 Referencias

- [1]. Udemy.com. Recuperado el 5 de junio de 2023, de <https://www.udemy.com/course/emqx-broker-servidor-iot/learn/lecture/33417262#overview>
- [2]. Udemy.com. Recuperado el 5 de junio de 2023, de <https://www.udemy.com/course/emqx-broker-servidor-iot/learn/lecture/33544932#overview>
- [3]. EMQ Technologies Inc. (s/f). EMQX: The world's #1 open source distributed MQTT Broker. Emqx.io. Recuperado el 5 de junio de 2023, de <https://www.emqx.io/>
- [4]. Udemy.com. Recuperado el 5 de junio de 2023, de <https://www.udemy.com/course/emqx-broker-servidor-iot/learn/lecture/33544940#overview>
- [5]. EMQ Technologies Inc. (s/f). Test MQTT services quickly with powerful MQTT client. Wwww.emqx.com. Recuperado el 5 de junio de 2023, de <https://www.emqx.com/en/products/mqtx>
- [6]. Win32/Win64 OpenSSL installer for windows - shining light productions. (s/f). Slproweb.com. Recuperado el 5 de junio de 2023, de <https://slproweb.com/products/Win32OpenSSL.html>
- [7]. Yu, S. (s/f). Enable two-way SSL/TLS for EMQX. Wwww.emqx.com. Recuperado el 7 de junio de 2023, de <https://www.emqx.com/en/blog/enable-two-way-ssl-for-emqx>
- [8]. Laragon - portable, isolated, fast & powerful universal development environment for PHP, Node.js, Python, Java, Go, Ruby. (2019, abril 25). Laragon - Portable, Isolated, Fast & Powerful
- [9]. Universal Development Environment for PHP, Node.Js, Python, Java, Go, Ruby. <https://laragon.org/>
- [10]. MySQL. (2020, febrero 7). EMQX 4.4 Documentation. <https://www.emqx.io/docs/en/v4.4/advanced/auth-mysql.html>
- [11]. MySQL ACL. (2020, febrero 7). EMQX 4.4 Documentation. <https://www.emqx.io/docs/en/v4.4/advanced/acl-mysql.html>

- [12]. (S/f). Udemy.com. Recuperado el 8 de junio de 2023, de <https://www.udemy.com/course/emqx-broker-servidor-iot/learn/lecture/33576118#questions>
- [13]. Download PuTTY: latest release (0.78). (s/f). Org.uk. Recuperado el 9 de junio de 2023, de <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- [14]. Getting Started | Hestia Control Panel. (n.d.). Hestiacp.com. Retrieved July 5, 2023, from <https://hestiacp.com/docs/introduction/getting-started.html>
- [15]. Freenom - Un nombre para todo el mundo. (n.d.). Freenom.com. Retrieved July 6, 2023, from <http://freenom.com/es/index.html>
- [16]. CLI Reference | Hestia Control Panel. (n.d.). Hestiacp.com. Retrieved July 7, 2023, from <https://hestiacp.com/docs/reference/cli.html#v-change-sys-hostname>
- [17]. Microsoft. (2016, April 14). Visual Studio Code. Visualstudio.com; Microsoft. <https://code.visualstudio.com/>
- [18]. Arduino Client for MQTT. (n.d.). Pubsubclient.knolleary.net. <https://pubsubclient.knolleary.net/api>
- [19]. (S/f). Udemy.com. Recuperado el 14 de julio de 2023, de <https://www.udemy.com/course/emqx-broker-servidor-iot/learn/lecture/34778434#overview>
- [20]. Crontab.guru - The cron schedule expression editor. (n.d.). Crontab.guru. <https://crontab.guru/>
- [21].
- [22].
- [23].
- [24].
- [25].
- [26].
- [27].
- [28].
- [29].
- [30].
- [31].
- [32].
- [33].
- [34].



GOBIERNO DE  
MÉXICO



CONACYT  
Consejo Nacional de Ciencia y Tecnología

CIDESI®

Av. Playa Pie de la Cuesta No. 702, Desarrollo San Pablo, CP. 76125, Santiago de Querétaro, Qro., México.  
Tel: 442 211 9800 y 800 552 2040 [www.cidesi.com](http://www.cidesi.com)

