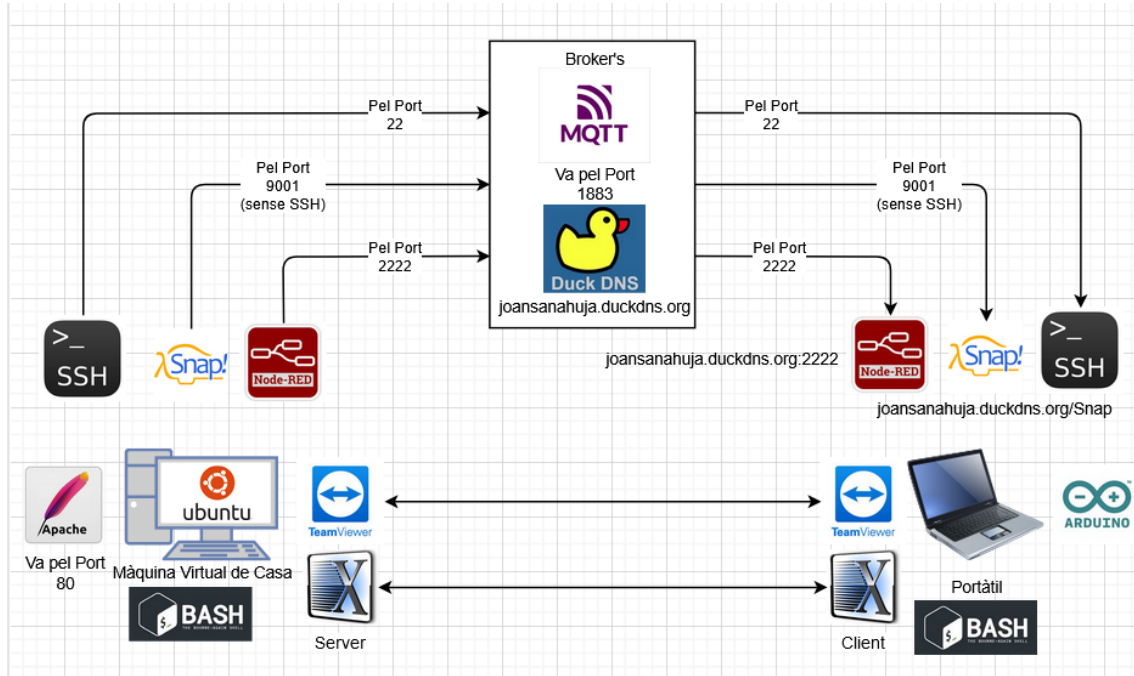


Projecte 18b: Aplicacions Node-Red i MQTTS: Monitorització de sistemes



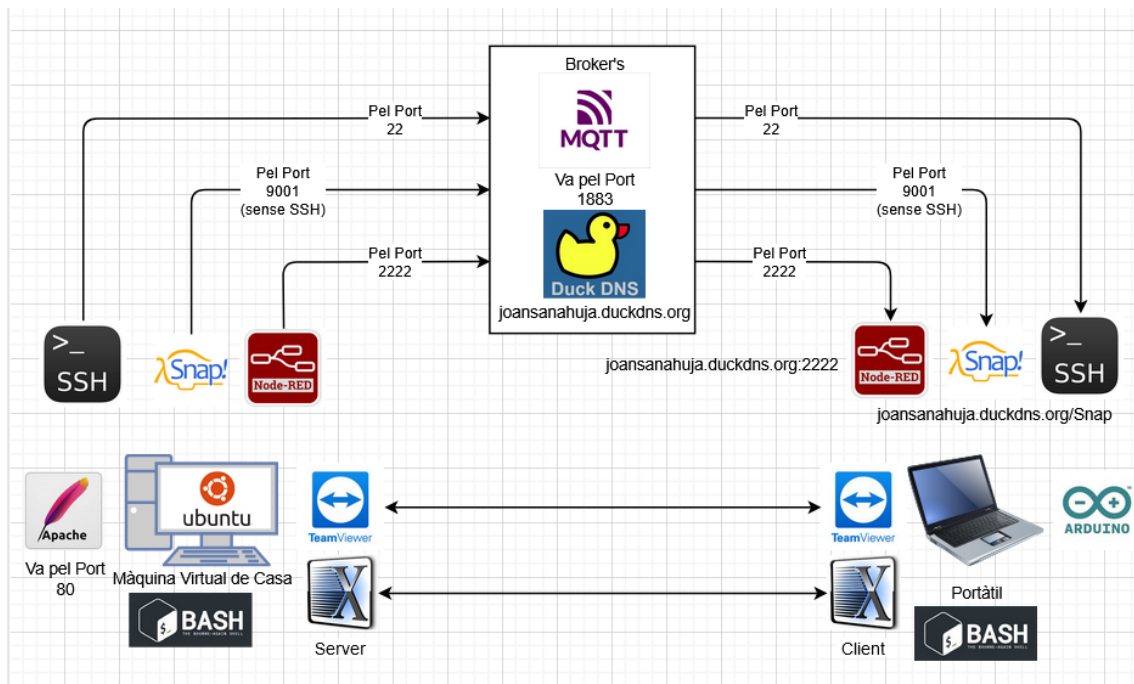
Índex:

1. Resum inicial (en un sol Full).
2. Desenvolupament i execució del projecte.
 - a. Teoria dels nous Programes/Serveis.
 - b. Documentació dels passos que he fet.
 - i. PRIMERA SETMANA.
 - ii. SEGONA SETMANA.
 - iii. TERCERA SETMANA.
 - iv. QUARTA SETMANA.
3. Pressupost.
4. Manual.
5. Annex A: Bibliografia.

Resum Inicial (en un sol Full):

L'objectiu final d'aquest projecte es poder controlar una placa desde qualsevol lloc amb wifi, sense que la placa es tingui que moure del seu lloc.

Per poder-ho fer tindrem que poder-nos connectar entre dos ordinadors diferents en xarxes wifi diferents, per fer això necessitem tindre un dels dos ordinadors quiet sempre en la mateixa xarxa wifi i que no s'apagui/suspengui de manera automàtica, en aquest ordinador necessitem que tingui instal·lat i/o iniciat sessió en els següents programes i/o serveis, en el meu cas ho tindre tot dintre d'una màquina virtual Linux amb un Ubuntu, tindrem que iniciar sessió i/o instal·lar els següents programes: DuckDns, TeamViewer, NodeRed, Snap, X2Go Server, SSH i MQTT i mitjançant la pàgina de configuració del router, habilitarem cinc ports concrets, el 80, el 22, el 2222, el 1883, el 9001. En l'altre ordinador tindrem que iniciar sessió i/o instal·lar els programes següents: DuckDns, TeamViewer, NodeRed, Snap, X2Go Client, SSH, MQTT i Arduino.



Aquí es pot veure un resum de les connexions que farem durant el projecte.

Desenvolupament i execució del projecte:

Teoria dels nous Programes/Serveis:

A continuació es diran els programes/serveis que es faran servir per aquest projecte i s'explicaran els nous i/o els que em utilitzat menys.

Els programes que es faran servir són: Apache, Linux (Ubuntu), DuckDns, TeamViewer, Node-Red, Snap, X2Go, SSH, MQTT, Arduino i el Bash Script. S'explicaran breument tots els programes/serveis excepte el Apache i el Linux (Ubuntu), perquè ja els hem fet servir anteriorment i els coneixem. En canvi els altres no els hem fet servir encara o els hem fet servir molt poc. Explicaré breument els següents programes/serveis:

- **DuckDns:** És un Dynamic DNS (DDNS) gratuït que transforma la IP pública per un nom de domini a escollir.
- **TeamViewer:** És un programari que serveix per connectar-se de forma remota a un altre equip (que també tingui TeamViewer).
- **Node-Red:** És una eina de programari que permet connectar dispositius físics en el "internet de les coses".
- **Snap:** És un llenguatge de programació educatiu lliure. Permet crear animacions i jocs entre d'altres coses.
- **X2Go:** És un software de codi obert que serveix per poder-se connectar de manera remota a un escritori, la diferencia que té en referencia al TeamViewer, és que el X2Go quan es connecta tots els processos s'estan fent en el dispositiu amb el qual t'estàs connectant, mentre que amb el TeamViewer els processos s'estan fent en el dispositiu en el qual t'estàs connectant.
- **SSH:** És un protocol i el programa informàtic que l'acompanya, serveix per connectar-se de manera remota a un altre dispositiu, però en el seu cas mitjançant la xarxa.
- **MQTT:** És un protocol de missatgeria basat en el protocol TCP/IP, els broker's s'ubicarien dintre d'aquest protocol.
- **Arduino:** És una placa de circuit imprès simple de codi obert, és pot fer servir per desenvolupar objectes interactius autònoms o pot ser connectat a programari de l'ordinador.
- **Bash Script:** És un intèrpret d'ordres, serveix igual que un terminal.

Documentació dels passos que he fet:

PRIMERA SETMANA:

Crear Màquina Virtual:

Característiques/Requeriments:

El primer que tenim que fer es crear una nova màquina virtual amb les següents característiques, 4 GB de RAM (segons la RAM màxima de cada ordinador de sobretaula o portàtil), 40 - 50 GB d'emmagatzematge (per no tindre problemes d'espai més endavant), l'adaptador de xarxa en adaptador pont i amb la ISO de Ubuntu més recent.

Programes a instal·lar:

Net-tools (Màquina virtual):

Instal·larem el grup de comandaments “net-tools”, en la màquina virtual, amb el comandament “sudo apt install net-tools”, però abans d'instal·lar res farem servir el comandament “sudo apt update” per actualitzar els paquets i perquè tot s'instali en la seva última versió i no falti cap paquet per instal·lar, després d'haver fet servir el comandament “update” farem servir el d'instal·lar les “net-tools”.

Aptitude (Màquina Virtual):

Instal·larem el comandament “aptitude”, en la màquina virtual, com ja hem fet abans el comandament “update” ara no fa falta que el tornem a fer, instal·larem el comandament “aptitude” amb el comandament “sudo apt install aptitude”.

Openssh-server (Màquina Virtual):

Instal·larem el programa “openssh-server”, en la màquina virtual, amb el comandament “sudo apt install openssh-server”, que es la versió “server” del programa “openssh” que serveix per poder-se connectar entre dispositius de manera segura.

Netstat (Màquina Virtual):

Això no es una instal·lació, però el que hem instal·lat abans, el “net-tools, aptitude i openssh-server” servia per poder fer servir aquest comandament. Serveix per consultar els ports oberts a GNU/Linux, es pot fer servir amb el comandament “sudo netstat -atpun | grep 22”.

X2go-Server (Màquina Virtual):

Instal·larem el programa “x2go” la versió “server”, en la màquina virtual, amb el comandament “sudo apt install x2goserver”, que serveix per poder-nos connectar amb un altre dispositiu desde qualsevol lloc, per poder-se comunicar necessita que l'altre dispositiu tingui instal·lat el “x2go” versió client per poder connectar-se amb el server.

X2go (Client):

Instal·larem el programa x2go la versió client en el dispositiu que volem que es connecti a la màquina virtual, per poder instal·lar-lo accedirem a la pàgina web [“https://wiki.x2go.org/doku.php/download:start”](https://wiki.x2go.org/doku.php/download:start) i ens descarregarem la opció que posa “Install X2Go Client” que es troba en l'apartat “X2Go Client” de la pàgina web.

Screen (Màquina Virtual):

Instal·larem el programa “screen”, en la màquina virtual, amb el comandament “sudo apt install screen” que serveix per poder tindre tasques executant-se en segon pla i que no molestin mentre que és fan altres coses. Quan s'accedeix al programa “screen” pots posar a executar qualsevol tasca, per sortir de l'screen es deben apretar les tecles “Ctrl A + D” per sortir, però deixant la tasca executant-se de fons, per tornar a l'screen i veure la tasca executant-se es faria servir el comandament “screen -r, pero es posible que hi hagi més d'un screen al mateix moment, per

poder-te connectar-te a un en concret tindrem que posar el nom que apareixeria si hagués més d'un screen funcionant al mateix moment. Per sortir de l'screen farem servir el comandament "exit" i si volem veure tots els processos que estan en marxa en aquell moment es faria servir el comandament "top".

Escriptoris lxde i xfce4 (Màquina Virtual):

Instal·larem els escriptoris "lxde i xfce4", en la màquina virtual, amb el comandament "sudo apt install lxde xfce4", són uns escriptoris per Linux que necessiten menys recursos per funcionar que la resta d'escriptoris disponibles de Linux.

Node-Red (Màquina Virtual):

Instal·larem el programa "Node-Red", en la màquina virtual, el primer que tenim que fer es accedir com "root", després fer servir el comandament "curl -sL https://deb.nodesource.com/setup_14.x | bash -" per descarregar el fitxer d'instal·lació del node-red. Ara amb el comandament "apt install -y nodejs" instal·larem el programa node-red, després sortirem de "root" amb el comandament "exit". Després de sortir de "root" comprovarem en quina versió s'ha instal·lat el node-red amb el comandament "node --version" i a sortit que hi ha instal·lada la versió "v14.15.4" i amb el comandament "npm -- version" ho comprovarem d'aquest programa i ens a sortit la versió "6.14.12".

Ara instal·larem el Node-Red autònom, en la màquina virtual, el primer que farem sera fer servir el comandament "mkdir bin" per crear la carpeta bin en el directori actual a la qual accedirem més tard, després d'accedir-hi farem servir el comandament "sudo apt install unzip" per instal·lar el programa "unzip", ara crearem una nova carpeta amb el nom de "nodered-ui" i que després accedirem, dintre de la carpeta farem servir el comandament "wget https://binefa.cat/dam2021/20200928/node_red_ui.zip", quan s'hagi acabat de descarregar farem servir el comandament "unzip node_red_ui.zip" per descomprimir l'arxiu que acabem de descarregar, després farem servir el comandament "node node_modules/node-red/red.js -- userDir node_modules/node-red/data --settings node_modules/node-red/settings.js -v --port 2222 --title elmeuTitol --safe flows_meuTitol.json"

Comandaments Extres:

Per connectar-se a un altre dispositiu:

Amb el comandament "ssh 172.20.19.23 -l usuari" on "172.20.19.23" poses la direcció IP del dispositiu al qual et vols connectar i en "usuari" poses a quin usuari et vols connectar del dispositiu al qual et vols connectar.

Memòria lliure (Màquina Virtual):

Amb el comandament "free -h" podem comprovar l'espai de memòria lliure que hi ha en el dispositiu.

Espai de disc lliure (Màquina Virtual):

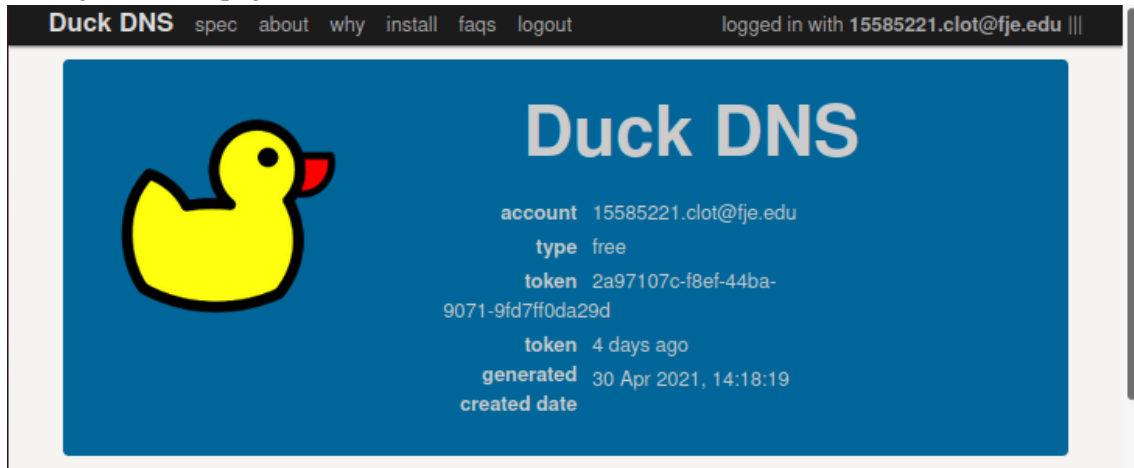
Amb el comandament "df -Th" podem comprovar l'espai de disc lliure que hi ha en el dispositiu.

SEGONA SETMANA:

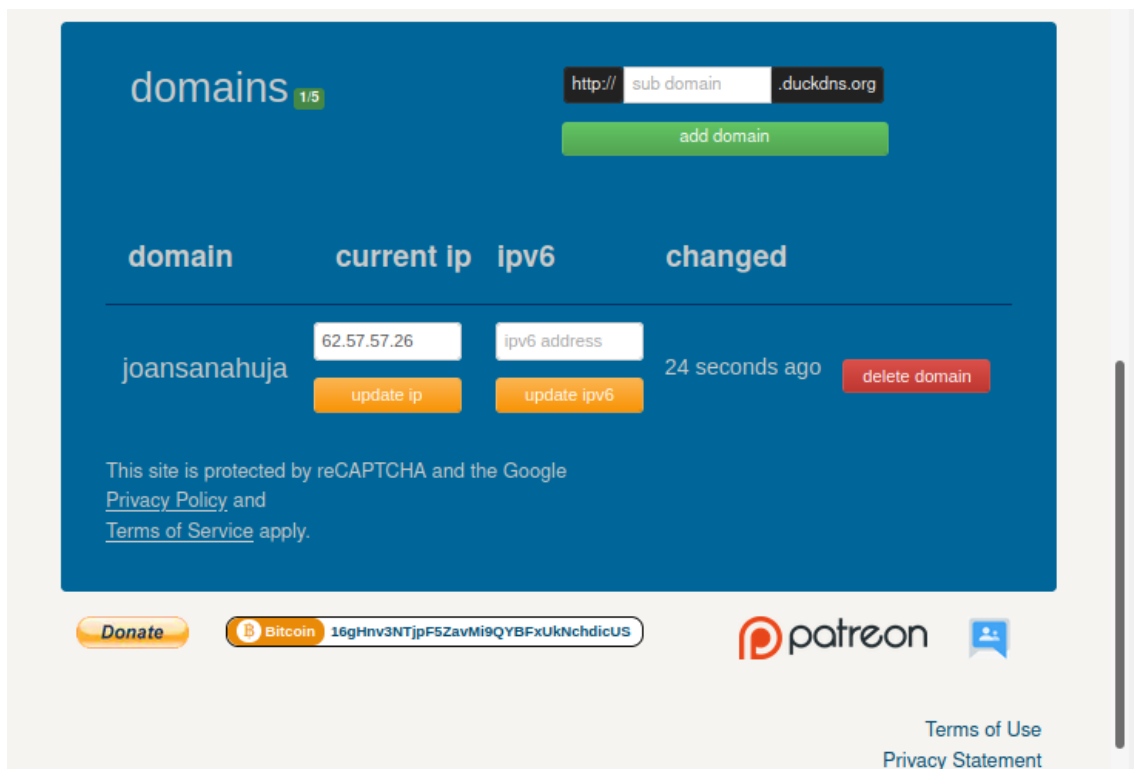
Pàgines web donades d'alta:

DuckDns:

Me registrat en la pàgina web de DuckDns.



Aquí es pot veure el resultat de registrar-se en la pàgina web.



I aquí es pot veure el domini que hem creat assignant-lo al la IP pública de la xarxa on està la màquina virtual de casa, que és la 62.57.57.26, al domini li he posat el nom de "joansanahuja.duckdns.org".

GitHub:

Me registrat en la pàgina web de GitHub per poder guardar informació, documents, etc, dintre seu i que altres, si jo ho permet-ho, puguin accedir i descarregar-se el seu contingut.

Avenços fets:

Instal·lar el TeamViewer en la Màquina Virtual de casa i en el portàtil:

He instal·lat el programa TeamViewer en la màquina virtual de casa i en el portàtil per poder connectar-me a la màquina de casa desde qualsevol xarxa de wifi diferent a la que està connectada la màquina de casa.

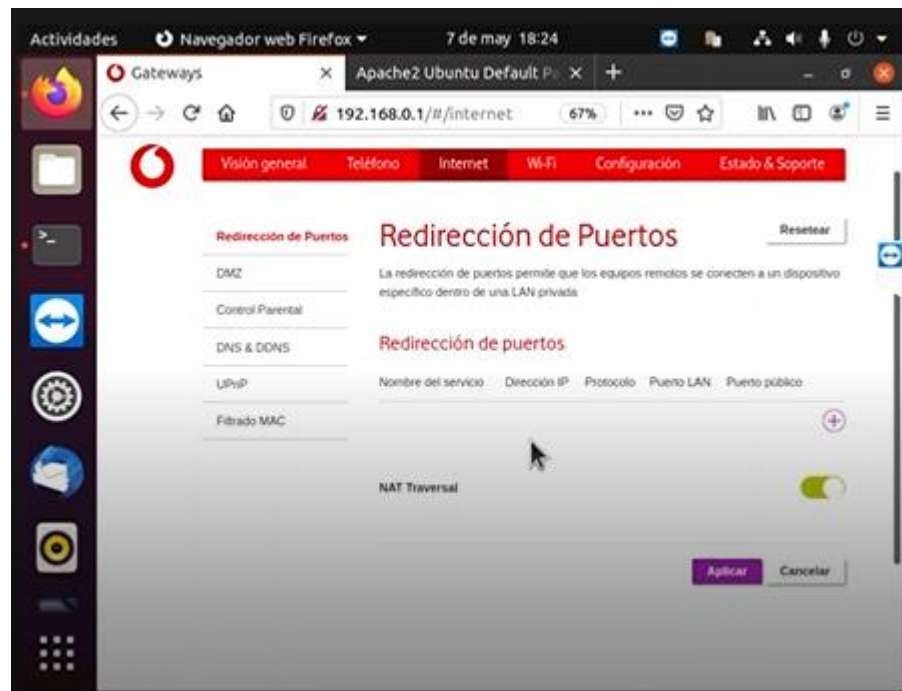
Redirecció de ports del router (desde el cole amb la Màquina Virtual de casa amb el TeamViewer):

He instal·lat el TeamViewer en la màquina virtual de casa, per poder-me connectar desde qualsevol xarxa i poder configurar el router de casa per poder redireccionar els ports i assignar-los a la direcció IP de la màquina virtual.

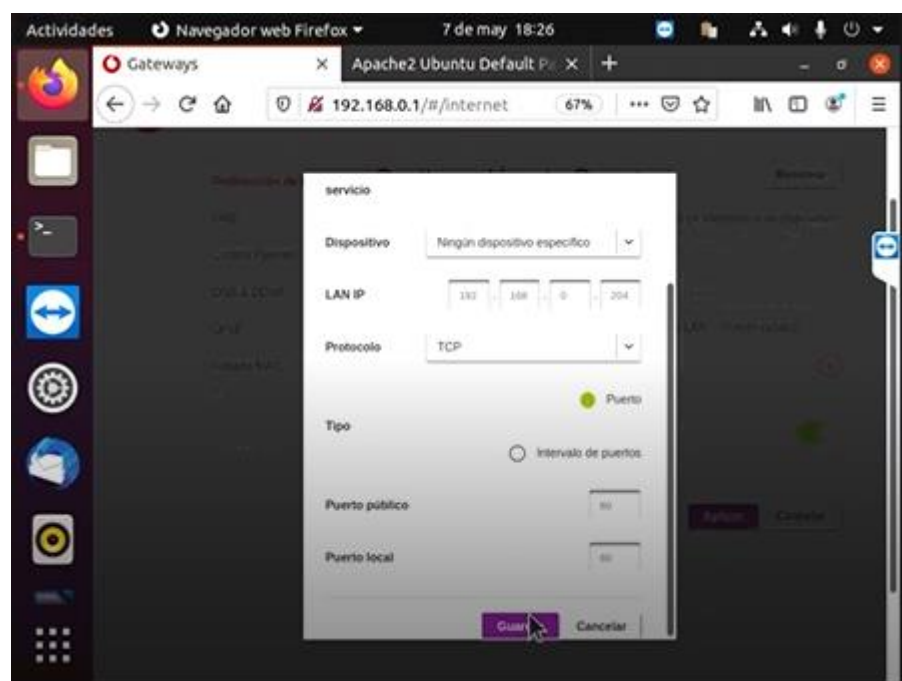
Quan ja haguem instal·lat el TeamViewer, ens connectarem a la màquina de casa amb ell, quan ja estiguem connectats, accedir-em al lloc web del teleoperador que tinguem, Vodafone en el meu cas, i introduïrem la informació necessària per poder iniciar sessió.



Aquí es pot veure com, desde el col·legi estem controlant la màquina de casa amb el TeamViewer i accedint al lloc web del meu teleoperador. Quan ja haguem accedit al lloc web, en el cas de tenir Vodafone com a teleoperador, anirem al apartat de "Internet" en la barra superior, un cop dintre entrarem a l'opció de "Redirección de Puertos" i després apretarem el signe "+" que hi ha per afegir una nova redirecció de ports.



Aquesta seria la part del lloc web on redireccionar-em els ports necessaris, el primer port que redireccionar-em serà el “80” el de l’Apache, per fer-ho apretarem el signe “+” que hi ha.



Una cop apareixi una finestra emergent amb la configuració per poder redireccionar un nou port, li posarem un nom, si volem, seleccionarem un dispositiu al qual volem que s’apliqui, després posarem la direcció IP de la màquina de casa, escollirem l’opció de “Puerto” i a sota seu posarem el port “80” en els dos apartats. Després guardarem el que acabem de fer i després aplicarem els canvis amb un botó que hi ha justament a sota del signe “+”, repetirem els passos quatre vegades més, però amb els ports corresponents en cada cas, el segon port que redireccionar-em serà el “22” que és el del “SSH”, el tercer serà el port “2222” que es el del NodeRed, el quart que he afegit es el port “1883” que és el del MQTT i el cinquè i últim port

que he afegit es el “9001” que es el del Snap sense SSH, cada vegada que afegim un port tindrem que aplicar els canvis.

Tot això s’ha fet per poder accedir a la màquina virtual de casa sense tenir cap problema per estar en una xarxa diferent.

Redirección de puertos					
Nombre del servicio	Dirección IP	Protocolo	Puerto LAN	Puerto público	
BigUp/21					
libtrt/1.2.11.0	192.168.0.146	UDP	6881	6881-6881	  
Downloader/16240					
NodeRed	192.168.0.204	TCP	2222	2222	  
MQTT	192.168.0.204	TCP	1883	1883	  
Snap	192.168.0.204	TCP	9001	9001	  
SSH	192.168.0.204	TCP	22	22	  
Apache	192.168.0.204	TCP	80	80	  
					

Aquest seria el resultat final d’habilitar els ports corresponents, el primer port que apareix no fa falta posar-lo, és posa de manera automàtica, per molt que el treguis, després torna a aparèixer automàticament.

Programes instal·lats:

També he instal·lat el Curl i el Git, amb el comandament “sudo apt-get install curl git”.

Comandaments Extres:

També he fet servir el comandament “sudo chmod -R 777 html/” està en la carpeta “/var/www” en la màquina virtual de casa, això serveix per donar accés al fitxer “html/”, per poder penjar arxius dintre seu desde el Windows a l’Ubuntu de casa, en aquest cas era per fer-ho amb el Filezilla.

TERCERA SETMANA:

Avenços fets:

Node Red a Windows:

Igual que quan vam instal·lar el Node Red en la màquina Linux, ho fem també en el dispositiu Windows.

Instal·lacions Extres:

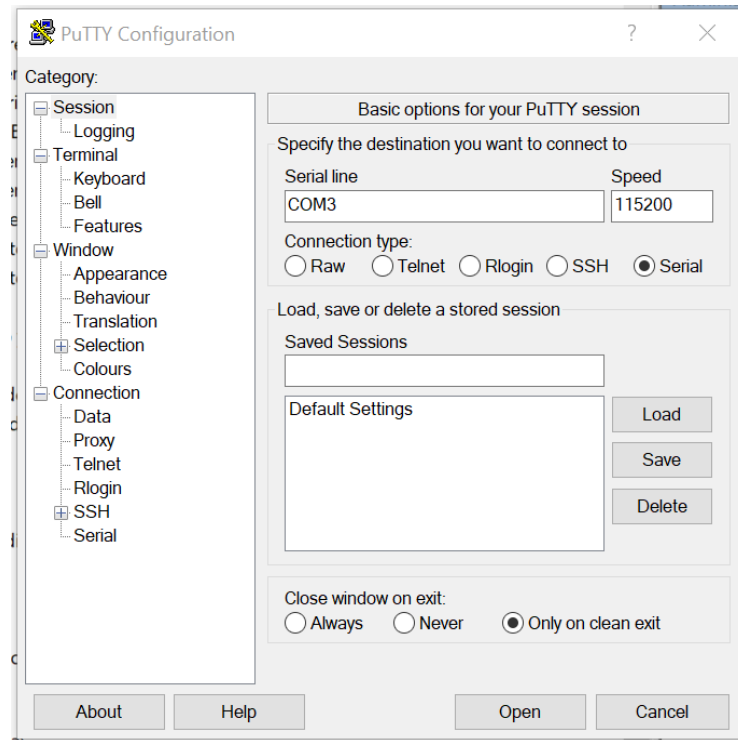
Hem instal·lat el Curl i el Git per Windows.

També hem instal·lat el broker Mosquitto.

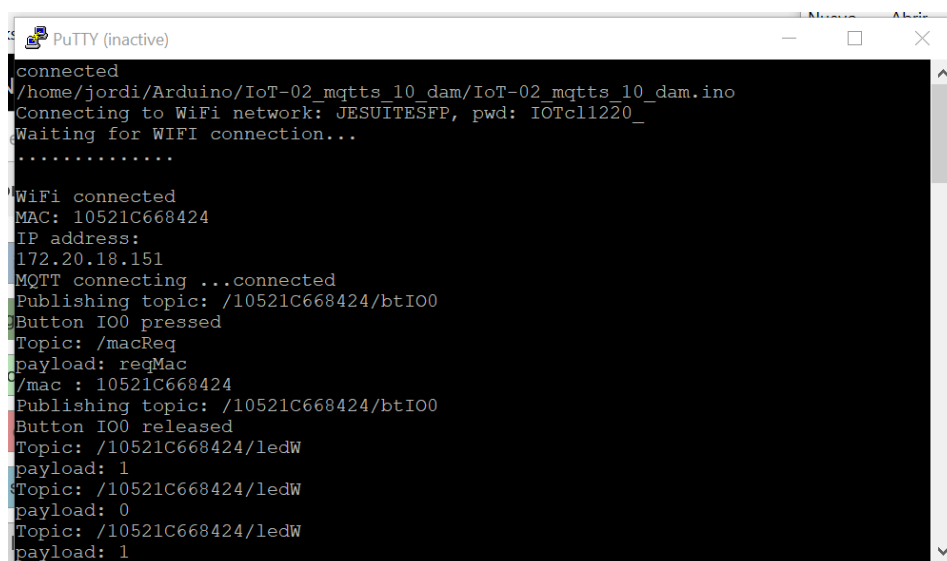
Avenços amb la placa:

Em connectat la placa al portàtil i mitjançant el administrador de dispositius i uns drivers, hem pogut instal·lar el controlador de la placa per poder connectar-nos a ella.

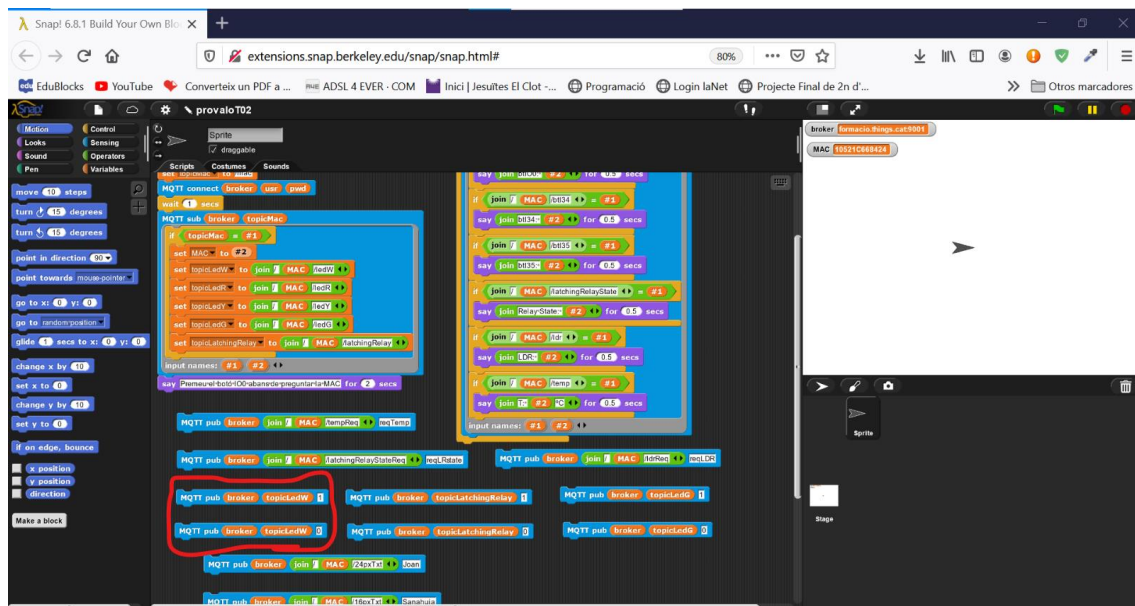
Mitjançant el Putty, ens hem connectat a la placa.



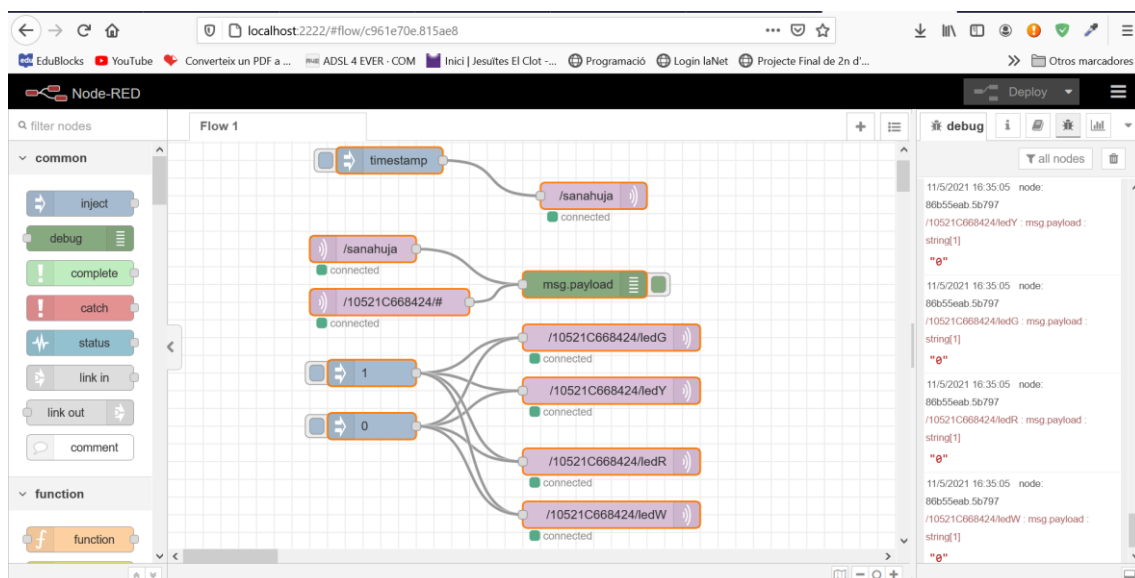
Ens hem connectat seleccionant l'opció de "Serial" del Putty i introduint en "Serial Line" "COM3" i en "Speed" 115200, després li hem donat a "Open".



Quan ens hem connectat, buscara una direcció wifi entre les que té disponibles, entre les que hi ha per la zona, quan ja s'ha connectat a una xarxa wifi, ens donarà una MAC i una direcció IP, la qual ens guardarem per fer-la servir posteriorment.



Ara obrirem el snap (sense ssh), obrirem un “.xml” que ens hem descarregat anteriorment, i apretarem la bandera verda, després agafarem la placa i apretarem un botó en concret i després seleccionarem el grup de blocs que apareix a la esquerra, això en deuria donar la MAC de la placa, la qual comparar-em amb la MAC que ens ha donat el Putty, que deuriem ser la mateixa. Per provar que està ben connectada apretarem el bloc que apareix marcat en la part inferior de la fotografia, primer premerem el primer bloc, que acaba en 1, i es deuria encendre una llum en la placa, després apretarem el següent bloc, el de sota del que acabem de prémer, i es deuria d’apagar la llum que s’acabava d’encendre. Ara farem lo mateix, pero desde el Node Red.



Després obrirem el Node Red i afegirem tres “inject”, cinc “mqtt out” i dos de “mqtt in” i un “debug”, en el primer “inject” no farem cap canvi i el connectar-em a un “mqtt out” que li posarem de “topic” “/sanahuja” juntament a un “mqtt in”, pero no els connectar-em entre ells, posarem un broker que tinguem disponible, després agafarem l’altre “mqtt in” i li canviarem el “topic” per “/la MAC que ens hagi donat el Putty/#” el “#” és perquè s’apliqui a tot, ara el text que afegim en “inject” apareixerà en la pantalla de la placa, si tot està ben fet. Després agafarem els dos “inject” que queden i li posarem 1 i 0, després agafarem el quatre “mqtt out” que queden

i de “topic” els hi posarem “/MAC/ledG”, “/MAC/ledY”, “/MAC/ledR” i “/MAC/ledW” i connectar-em els “inject” 1 i 0 als quatre “mqtt out”, si tot funciona correctament al prémer el botó blau al costat de cada “inject”, apretant el 1 es deurien encendre totes les llums i prement el 0 es deurien apagar totes les llums de la placa.

QUARTA SETMANA:

Avenços fets:

Placa:

Quan tinguem la placa connectada al portatil, mitjançant el “administrador de dispositivos” i alguns arxius del GitHub del professor assignat, farem que el portatil pugui detectar la placa i assignar-li un port “COM”.

Arduino:

El primer pas de tots que he fet és mitjançant la pàgina web del projecte, he accedit a la pàgina oficial d’Arduino i he descarregat el seu instal·lador més actual.

Després, quan ja he instal·lat el Arduino, tornarem a la pàgina del projecte i ens descarregarem alguns arxius, com el de les biblioteques ja modificades pel professor que ens ha sigut assignat, també el arxiu “IoT-02_mqtts_10_dam” que serveix per poder-nos connectar amb la placa, quan tinguem tot descarregat, ho descomprimir-em en el mateix lloc on estigui descarregat.

Arduino IDE amb l'ESP32

Descarregueu-vos el darrer IDE d'Arduino 1.x (al moment de fer aquesta wiki era 1.8.13)

[Añadir ESP32 al IDE de Arduino](#)

[Algunes biblioteques d'Arduino](#)

[Alguns codis d'Arduino](#)

[Placa IoT-02 al GitHub](#)

IoT-02

[provaloT02.xml](#) Baixeu-vos-ho amb el botó dret del ratolí (*Desa l'enllaç com a...*). Cal obrir-ho des d' Snap! **sense** SSL

[Microprogramari IoT-02_mqtts_10_dam](#) per a provar la connexió MQTT amb la placa IoT-02 (un exemple de microprogramari fent servir MQTTS en comptes d'MQTT)

Flux de NodeRED per a provar la connexió MQTT amb la placa IoT-02 (Caldrà afegir l'usuari **ecat** i la contrasenya **clotClot** al *broker*. A l'exportar el flux no es copien per seguretat)

Quan ja tinguem tinguem instal·lat el Arduino i les llibreries del projecte, anirem a la carpeta “Arduino” que trobarem a “Documents” i substituïrem la carpeta “libraries” que hi hagi en el moment i posarem la nostra.

Després accedir-em al link amb el nom de “Añadir ESP32 al IDE de Arduino” i farem tots els passos que ens digui.

Arduino IDE amb l'ESP32

Descarregueu-vos el darrer IDE d'Arduino 1.x (al moment de fer aquesta wiki era 1.8.13)

Añadir ESP32 al IDE de Arduino

Algunes biblioteques d'Arduino

Alguns codis d'Arduino

Placa IoT-02 al GitHub

Añadir ESP32 al IDE de Arduino

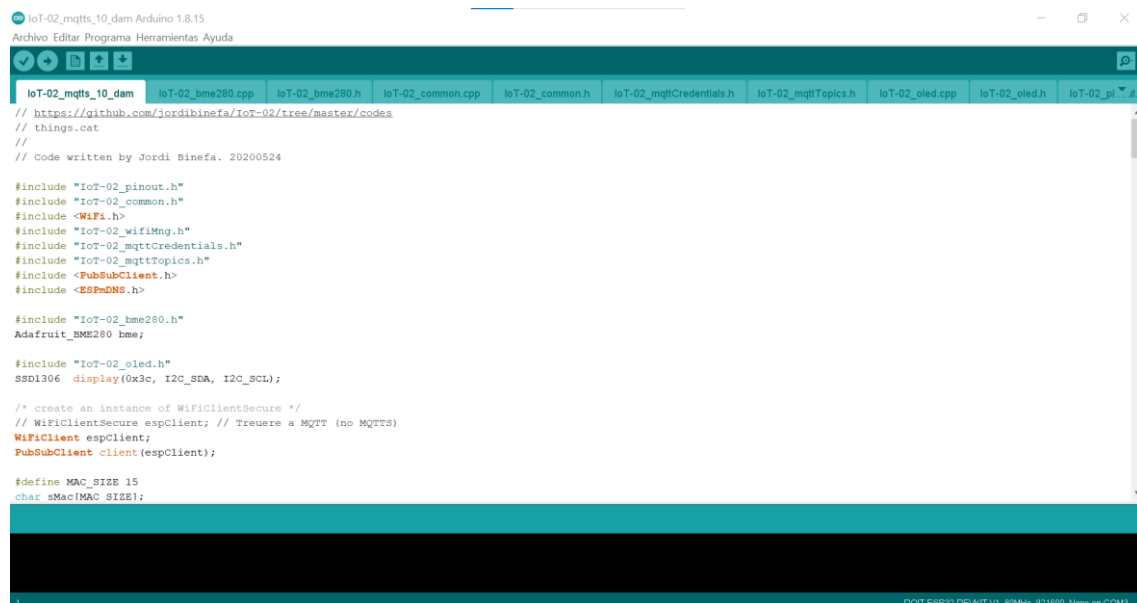
En el IDE de Arduino ir al menú y seleccionar *Fichero / Preferencias*. Copiar en la casilla indicada como *URLs adicionales del Gestor de placas*:

```
https://dl.espressif.com/dl/package_esp32_index.json, http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

En el menú *Herramientas / Placa: "... / Gestor de placas* escribir **ESP32**, seleccionar **esp32 by Espressif Systems** y clicar el botón *Instalar*. Aproximadamente se bajan automáticamente unos 300MB.

Una vez instalado, cerrar la ventana emergente de instalación de la placa y seleccionar *Herramientas / Placa: "... / DOIT ESP32 DEVKIT V1*. Después seleccionar el puerto adecuado en *Herramientas / Puerto / COMx*.
Video e información más detallada: *Installing the ESP32 Board in Arduino IDE (Windows, Mac OS X, Linux)*

Després de fer el que diu el el link obrirem un nou Arduino i obrirem el que em descarregat anteriorment, el “IoT-02_mqtts_10_dam”, i si tot a funcionat correctament ens deurien aparèixer les llibreries que hem instal·lat anteriorment.

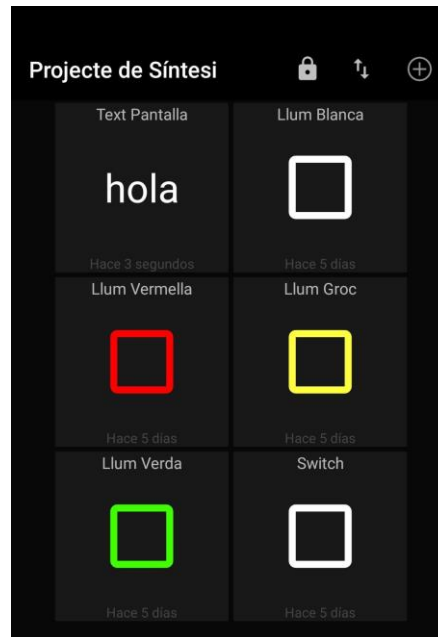


Així és tindria que veure l'Arduino si tot ho hem fet correctament.

Ara per configurar la placa de manera que faigi el que nosaltres volgum, tindrem que accedir a la pestanya de “IoT-02_mqttCredentials.h” i on posa “mqtt_server” posarem el nostre broker i la contrasenya i usuari que li vam posar quan vam instal·lar el broker Mosquitto, també després anirem a la pestanya “IoT-02_wifiCredentials.h” i seleccionarem una xarxa wifi i la canviarem per la nostre de casa i/o la del cole.

Aplicació MQTT Dash:

Ens instal·larem l'aplicació MQTT Dash en el mòbil, per poder fer coses en la placa desde el movil, nomes necessitariem la MAC de la placa, saber el nostre broker i saber quins son els “topics” per fer cada cosa.



Aquest seria el resultat final de configurar l'aplicació MQTT Dash per poder controlar la placa desde el mòbil.

Pressupost:

El pressupost que hi ha sería, dos ordinadors i una placa amb una pantalla, llums de diferents colors i un interruptor, tot totalment funcional.

Manual:

Per fer que la placa funcioni i la poguem controlar tenim que fer els següents passos:

- L'ordinador de casa amb la Màquina Virtual Ubuntu funcionant.
- Màquina virtual estigui en adaptador pont.
- Tots el programes i/o serveis instal·lats o iniciat sessió en la màquina virtual i l'altre ordinador.
- Habilitar els ports corresponents en la configuració del router de casa.
- Tots els programes i/o serveis que es tenen que comunicar entre els dos ordinadors es puguin comunicar.
- Tots els programes i/o serveis funcionant i/o activats correctament.
- Configuració de la placa té que ser la corresponent.
- Node-Red i Snap ven configurats.
- Tenir l'aplicació MQTT Dash instal·lada en el mòbil.
- Configurar l'aplicació MQTT Dash per poder controlar la placa des del mòbil.
- Poder controlar la placa desde l'aplicació del mòbil, desde l'Snap de la màquina virtual (obert en l'altre ordinador) i desde el Node-Red de la màquina virtual (obert en l'altre ordinador).

Annex A: Bibliografia:

Enllaços de les pàgines web més utilitzades/importants:

- [http://binefa.com/index.php?title=Projecte_Final_de_2n_d%27SMX - Curs 2020-2021](http://binefa.com/index.php?title=Projecte_Final_de_2n_d%27SMX_-_Curs_2020-2021) (Pàgina del Projecte de Síntesi).
- <https://github.com/jordibinefa> (GitHub del professor assignat).
- <https://es.wikipedia.org/wiki/Wikipedia:Portada> (On he tret la informació de la teoria).
- <http://joansanahuja.duckdns.org:2222/#flow/ef04e289.ad0708> (Node-Red de la màquina virtual de casa).
- <http://joansanahuja.duckdns.org/snap/snap.html#> (Snap de la màquina virtual de casa).
- <https://github.com/JoanSanahuja> (La meua pàgina del GitHub).