

Instalación y prueba de los servidores comunes (una pila LAMP) y Node-RED

Introducción

Esta guía busca lograr que el estudiante aprenda a instalar y probar los servidores más comunes que se usan para crear páginas web, pero también servicios asociados con bases de datos y cierta inteligencia basada en PHP. Para el caso dado, el estudiante ya cuenta con una máquina virtual para este fin en Azure, pero la verdad este trabajo es independiente de Azure, puede hacerse en cualquier máquina que se destine como servidor. El producto de esta guía es un servidor web típico, conocido como una pila LAMP (Linux, Apache, MySQL, PHP).

Objetivos de este taller

Que el estudiante:

- Conquista el lenguaje (terminología) apropiado para este aprovechar este tipo de servicios.
- Implemente un servidor que combine: MySQL, Apache 2, PHP y phpmyadmin.

Sobre el informe a entregar

Solo se entregarán las tablas que aparecen en la tarea 4

Conocimientos previos

Apache2: Es un servidor web HTTP de código abierto, es ampliamente utilizado en Internet debido a su fiabilidad, estabilidad y escalabilidad. Además de permitir alojar sitios web también brinda la posibilidad de que otras cosas puedan ser accedidos por Internet como: archivos, bases de datos.

PHP: Es un lenguaje de programación a nivel servidor y puede ser instalado y utilizado en una máquina Ubuntu independientemente de Apache o cualquier otro servidor web. Pero su alianza con un servidor web permite que los productos hechos con PHP se puedan publicar como página web. En este sentido, tiene similitudes con JavaScript tal como se usa en Google Apps Script, pero también con Django y con Node.js.

Django: Es un framework de desarrollo web de alto nivel y escrito en Python que fomenta el desarrollo rápido y limpio. Proporciona una arquitectura basada en MVC (Modelo-Vista-Controlador) y un conjunto rico de herramientas y bibliotecas para construir aplicaciones web complejas y escalables.

Node.js: Es un entorno de ejecución de JavaScript del lado del servidor que permite ejecutar JavaScript en el servidor, lo que proporciona un entorno de programación consistente tanto en el frontend como en el backend. Node.js está diseñado para aplicaciones en tiempo real y de alta concurrencia, y es especialmente adecuado para el desarrollo de aplicaciones web escalables y orientadas a eventos.

Comparación entre PHP, Node.js y Django:

- PHP y Node.js son lenguajes de programación específicos del lado del servidor, mientras que Django es un framework de desarrollo web basado en Python.
- PHP, Django y Node.js tienen sus propias sintaxis y conjuntos de herramientas, lo que puede influir en la elección de una tecnología sobre otra según las preferencias y el contexto del proyecto.
- PHP, Django y Node.js permiten la creación de webservices, webhooks y APIs, así como la manipulación de documentos y archivos.
- Django y Node.js pueden proporcionar facilidades específicas para el desarrollo web moderno, como el enrutamiento basado en URL, el manejo de formularios y la gestión de sesiones, que pueden diferir de las herramientas disponibles en PHP.
- Tanto python como PHP y como javascript pueden ser usados en páginas html para llamar funciones que están en un servidor (backend), pero javascript, a diferencia de python y PHP, también puede aparecer en una página html para realizar operaciones a nivel de frontend

Comparación entre Node.js y Google Apps Script

- Tanto Node.js como Google Apps Script (GAS) son entornos de desarrollo a nivel de backend. Permiten crear y usar web services, API. Pero Node.js puede ser instalado y administrado en un servidor propio. GAS ya está en Google, los programadores solo lo usan y ya, con sus capacidades y limitaciones. En otras palabras, GAS es una solución serverless
- Node.js permite manipular archivos y directorios locales, GAS permite hacer lo mismo pero con archivos y directorios de Google Drive
- Ambos permiten manipular archivos remotos en otras plataformas, pero acudiendo al uso de API de esas plataformas.
- Node.js es más apropiado para grandes proyectos empresariales. Google lo es para los proyectos donde Google Drive y los recursos de Google en general son claves. Dicho en otras palabras Node.js es independiente de operadores específicos, GAS es dependiente de Google.

MySQL: Es la herramienta que permite crear bases de datos. Es comparable con Excel pero con estilos diferentes, comandos diferentes y mayores compatibilidades para trabajar de la mano con servidores web, en ese sentido es quizá más cercano a Google Sheet, pero igual tiene mayores aplicaciones si se cuenta con servidor propio. En el fondo, SQL es un

conjunto de reglas, pero cada empresa puede crear su propia versión de usarlas por eso se habla de: MySQL, Microsoft SQL Server

phpmyadmin: es una alternativa para poder operar MySQL como si fuese una especie de Excel, es decir, sin necesidad de usar una terminal de comandos que sería lo tradicional. Es operado por un Servidor Web y es ahí donde vemos el sentido de Apache

Tarea 1. Instalación de: Apache 2, MySQL, PHP y phpmyadmin

El fin de esta instalación es poder brindarle a las soluciones IoT la posibilidad de usar bases de datos.

Paso 0. Realizar un backup representado en un snapshot de la máquina que ya se tiene, con el fin de poder recuperarla en caso de cometer algún error o cometer fallas.


Seguir el [capítulo 6 de manual azure](#)

Paso 1. Las instalaciones

Preparativos	
Comando	Explicación
sudo su	<p>Abrir una sesión de superusuario o "root".</p> <p>sudo: Significa "superuser do" y se usa para ejecutar comandos con privilegios de superusuario o administrador. Requiere que ingreses tu contraseña de usuario actual para confirmar tu identidad antes de ejecutar el comando.</p> <p>su: Significa "switch user" (cambiar de usuario) y se utiliza para cambiar al usuario especificado (por defecto, el usuario root) o a otro usuario. Por sí solo, ejecutado como su, te pedirá la contraseña del usuario al que deseas cambiar.</p>
sudo apt update	Actualiza la lista de paquetes disponibles en los repositorios configurados en tu sistema. Asegura que tienes la información más actualizada sobre los paquetes disponibles.
sudo apt upgrade	Actualiza los paquetes instalados en tu sistema a las versiones más recientes disponibles. Te pedirá confirmación antes de realizar la actualización. Puedes ejecutarlo después de apt update para

	aplicar las actualizaciones.
Apache2	
sudo apt-get install apache2	<p>Instalación de Apache2.</p> <p>Apache HTTP Server, comúnmente conocido como Apache, es un servidor web de código abierto y uno de los servidores web más populares del mundo. Fue creado y es mantenido por la Apache Software Foundation. Apache se encuentra disponible para varios sistemas operativos, incluyendo Linux, Unix, Windows, y más. Permite crear páginas Web. Es software abierto, con alta seguridad, soporta protocolos como http, https, ftp.</p>
service apache2 status	Para comprobar que el servidor apache está activo. Hemos comprobado que el servidor arranca cada vez que la VM se reinicia
MySQL	
sudo apt-get install mysql-server mysql-client libmysqlclient-dev	<p>Instalación de MySQL.</p> <p>Es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto. Utiliza SQL (Structured Query Language), el lenguaje estándar para interactuar con bases de datos relacionales. MySQL es desarrollado, distribuido y soportado por Oracle Corporation y es ampliamente utilizado debido a su robustez, flexibilidad y facilidad de uso.</p>
service mysql status	Comprobar el estado de MySQL
sudo mysql -u root -p	<p>Inicia el cliente de línea de comandos, indicando que soy el usuario root. "-p" significa que quiero definir la contraseña del usuario.</p> <p>Para no correr riesgo de olvidar la clave, le asignaremos la misma de la VM. Con los estudiantes E3T, solemos usar la clave: e3te3te3t</p> <p>Como resultado estarás en la consola de mysql y verás el prompt:</p>

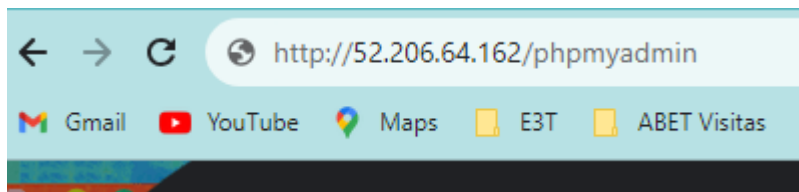
	<code>mysql></code>
<code>mysql> ALTER USER root@localhost IDENTIFIED WITH mysql_native_password BY 'e3te3te3t';</code>	<p>Dentro MySQL se usan comandos que terminan en “;”. Debes enviar de a un comando seguido de enter. OJO: en lugar de password debes poner el password asignado a mysql en el paso anterior.</p> <ul style="list-style-type: none">• el 1er comando realiza cambios al usuario root, por un lado para la autenticación se realiza mediante el plugin “mysql_native_password” y en segundo lugar para definir una nueva contraseña al root.• el 2do recarga las tablas de privilegios en el servidor MySQL, aplicando cualquier cambio realizado a los usuarios y sus permisos sin necesidad de reiniciar el servidor.• el 3ro es para salir de la terminal MySQL
<code>mysql> FLUSH PRIVILEGES;</code>	
<code>mysql> EXIT;</code>	
<pre>mysql> ALTER USER root@localhost IDENTIFIED WITH mysql_native_password BY 'e3te3te3t'; Query OK, 0 rows affected (0.01 sec) mysql> FLUSH PRIVILEGES; Query OK, 0 rows affected (0.01 sec) mysql> EXIT; Bye root@ip-172-31-43-14:/home/ubuntu#</pre>	
<h3>Instalación de PHP</h3>	
<code>sudo apt-get install php libapache2-mod-php php-mysql php-cgi php-curl php-json</code>	<p>Instalación de PHP</p> <p>Pero también varios módulos adicionales necesarios para ampliar las funcionalidades de PHP como:</p> <p>libapache2-mod-php: módulo PHP para Apache2, que permite a Apache2 procesar archivos PHP. Esto es esencial para ejecutar scripts PHP en un servidor web Apache</p> <p>php-mysql: módulo MySQL para PHP, que permite a PHP interactuar con bases de datos MySQL. Esto es necesario para aplicaciones web que requieren acceso a una base de datos MySQL.</p> <p>php-cgi: interfaz de puerta de enlace común (CGI) para PHP. CGI es una tecnología que permite al servidor web ejecutar programas externos para generar</p>

	<p>contenido dinámico. php-cgi es una alternativa al módulo Apache para ejecutar PHP scripts.</p> <p>php-curl: módulo cURL para PHP, que permite a PHP hacer solicitudes HTTP a otros servidores. Esto es útil para interactuar con APIs y otros servicios web.</p> <p>php-json: el soporte para JSON en PHP, lo que permite a PHP manipular datos JSON. JSON (JavaScript Object Notation) es un formato de datos ampliamente utilizado para intercambio de datos entre el servidor y las aplicaciones web.</p>
<pre>sudo nano /etc/apt/sources.list</pre> <p>Comentar (anteponer #) la línea:</p> <pre>deb http://us-east-1.ec2.archive.ubuntu.com/ubuntu/ jammy-updates main restricted</pre> <p>La línea comentada quedará así:</p> <pre>#deb http://us-east-1.ec2.archive.ubuntu.com/ubuntu/ jammy-updates main restricted</pre>	<p>Este paso solo se realiza en caso en que el anterior comando haya botado error.</p> <p>Nota: nosotros no hemos realizado este paso porque el comando anterior no arrojó error.</p> <p>En caso de error el proceso que recomienda el video es configurar el archivo sources.list así: se abre con sudo nano > se comentar la línea > se guardan los cambios (Ctrl+O > Ctrl+X)> se repite la instalación de PHP (el paso anterior)</p>
Instalación de phpmyadmin	
<pre>sudo apt-get install phpmyadmin</pre>	<p>Instalación de php myadmin. Este proceso saldrán varios cuadros de dialogo:</p> <ul style="list-style-type: none"> • 1er cuadro: Please choose web server> con la tecla espacio selecciona apache2 (OJO Superclave que aparezca el asterisco * en apache 2, como en la Fig siguiente) > con la tecla "tab" 'pasa al "ok" >enter  <ul style="list-style-type: none"> • 2do cuadro: The phpmyadmin package must have a database >

	secciona [yes] <ul style="list-style-type: none"> • 3er cuadro: password > el mismo que usado arriba para MySQL
sudo chmod 777 -R /var/www	Para que cualquier usuario, grupo y otros en el sistema pueda acceder y modificar la carpeta /var/www <ul style="list-style-type: none"> • chmod: comando usado para cambiar los permisos de archivos y directorios • 777: se refiere a permisos de "lectura, escritura y ejecución" • -R aplica el cambio de permisos de manera recursiva a todos los subdirectorios y archivos
sudo service apache2 restart	Reinicia apache

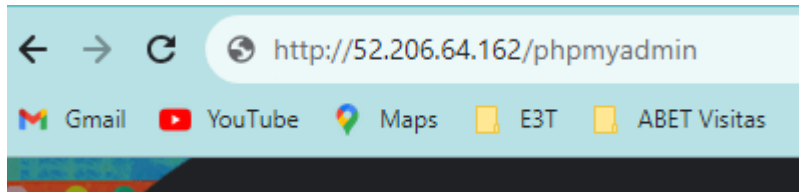
Paso 2. Comprobar que phpmyadmin está bien instalado. Abrir phpmyadmin que no es otra cosa que un editor/visualizador de MySQL

En un explorador de Internet llame phpmyadmin así: http://ip o dns/phpmyadmin
Por ejemplo:

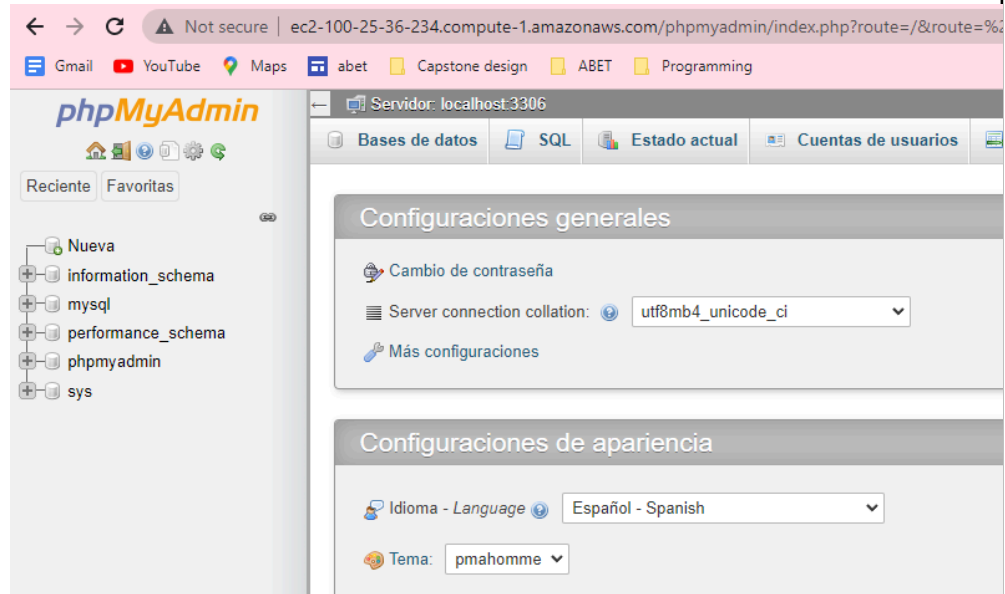


Abra sesión para el usuario root > contraseña definida para mysql > Continuar
Por ejemplo:

En un explorador de Internet llame phpmyadmin así: http://ip o dns/phpmyadmin
Por ejemplo:



El resultado es:



Paso 3. Pruebas a phpmyadmin/MySQL

Realiza tu primer reconocimiento de phpmyadmin/MySQL, siguiendo este [video tutorial desde el minuto 7:07](#)

Tarea 2. Instalación de Node.js

Node.js representa una revolución para desarrollar aplicaciones web y servicios del lado del servidor, destacándose por su enfoque asíncrono basado en JavaScript. Su eficiencia en la gestión de múltiples solicitudes concurrentes lo posiciona como la elección ideal para aplicaciones en tiempo real y servicios que requieren actualizaciones continuas, como chats y juegos en línea.

Para aquellos familiarizados con Google Apps Script, Node.js comparte similitudes valiosas. Ambas plataformas utilizan JavaScript, permiten el desarrollo de servicios web, APIs, y webhooks, y facilitan el cómputo en la nube, incluyendo el desarrollo tanto del backend como del frontend. Sin embargo, la distinción principal radica en que Node.js se instala en un servidor propio, lo cual significa independencia.

Cuando exploramos alternativas, Python, con su capacidad para crear servicios RESTful, se presenta como un competidor significativo para lograr algo similar. Mientras Node.js se destaca por su eficiencia y capacidad para manejar eventos en tiempo real, Python brilla en el desarrollo RESTful, ofreciendo un marco robusto y ampliamente utilizado llamado Flask. Este marco proporciona herramientas poderosas para la construcción de APIs RESTful y servicios web eficientes.

En la elección entre Node.js y Python, se trata de sopesar las fortalezas específicas de cada tecnología en función de los requisitos particulares del proyecto. Ambas ofrecen soluciones sólidas para el desarrollo web y del lado del servidor, permitiendo a los desarrolladores elegir la herramienta que mejor se adapte a sus necesidades y preferencias.

Las siguientes son las instrucciones tomadas del archivo Readme que aparece en el [repositorio de github](#) en el año 2023, semestre 2 (2023-2) para la instalación de Node. Este proceso también instala npm que es un gestor de paquetes para node. Es decir que para realizar instalaciones adicionales a Node se usará npm. Hacemos esta aclaración porque las instrucciones pueden llegar a cambiar con el tiempo y muchas veces es importante ir a la fuente. En el 2023-2, todo indica que estamos usando la versión 20.09 de Node que es la más madura y más apropiada para nuestra versión de Ubuntu, aunque ya existe la versión 21x.

Paso 1. Bajar el código fuente y la clave de seguridad

OJO: Entrega todos los siguientes comandos de un solo golpe, osea como un solo comando en la terminal

```
sudo apt-get update && sudo apt-get install -y ca-certificates curl gnupg &&
sudo mkdir -p /etc/apt/keyrings && curl -fsSL
https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | sudo gpg --dearmor
-o /etc/apt/keyrings/nodesource.gpg
```

Paso 2. Crear el repositorio deb

Nota: con "NODE_MAJOR=20" estarás indicando que deseas instalar la versión 20.x. Otras opciones son

NODE_MAJOR=16

NODE_MAJOR=18

NODE_MAJOR=20

NODE_MAJOR=21

OJO: Entrega todos los siguientes comandos de un solo golpe

```
NODE_MAJOR=20 && echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg]
https://deb.nodesource.com/node_${NODE_MAJOR}.x nodistro main" | sudo tee
/etc/apt/sources.list.d/nodesource.list
```

Paso 3. Correr las actualizaciones e instalar Node.js

```
sudo apt-get update && sudo apt-get install nodejs -y
```

Tarea 3. Instalación de Node Red

Node Red es una herramienta que se puede instalar en cualquier computadora, pero el verdadero sentido está en instalarlo en un servidor como es el caso de la nuestra VM en AWS. La página de Node-Red se refiere a ella como “Low-code programming for event-driven applications. Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways. It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click” De manera similar a cuando Moodle se instala en un servidor, es posible invocar Node Red desde cualquier computadora como una página web. Desde otro ángulo, Node-RED también es comparable, en la forma en que se crean soluciones, con gnuradio en el sentido que ambos usan la programación visual, mediante flujogramas. Pero en Node-RED es más común el término nodo para referirse no sólo a los elementos que se usan sino también a un flujograma, que surge como producto de unir varios nodos para una solución. En este sentido hay una similitud con la programación orientada a objetos, donde varios objetos se pueden unir para componer un objeto mayor. Pero la concepción detrás de Node-RED va más allá de todo lo existente porque está pensada más bien para soluciones en la nube a nivel de servidor.



Haz clic en la imagen para ver un vídeo introductorio sobre Node-RED

Más allá de todo eso, con Node-RED es posible crear, en línea, interfaces para visualización gráfica en línea. En este sentido es una competencia para otras herramientas que se usan en la UIS como llamada Ignition con MQTT Engine, pero esta última tiene un alto costo para su uso, mientras que Node es de uso libre.

[Node-RED](#). Podemos decir que Node-RED busca llegar a ser el equivalente gráfico de Node.js, lo cual es mucho decir si recordamos que Node.js está entre las herramientas más competitivas para los desarrollos en la nube.

Recomendaciones iniciales:

- Visita la página de Node-RED para conocer novedades: <https://nodered.org/>
- Vemos que Node-RED tiene muchas opciones de uso, desde la instalación local, en PC, en una raspberry, en android y en la nube. Nos interesa esta última y que sea en Azure. Por ahora, hemos visto en la página <https://nodered.org/> un enlace sobre el uso en AWS que puede servir para el caso de Azure: [Amazon Web Services](#)
- El enlace anterior es el recomendado para conocer las últimas novedades en la instalación. Los comandos de la siguiente tabla es una experiencia de instalación vivida en febrero 2023, siguiendo este [video](#)

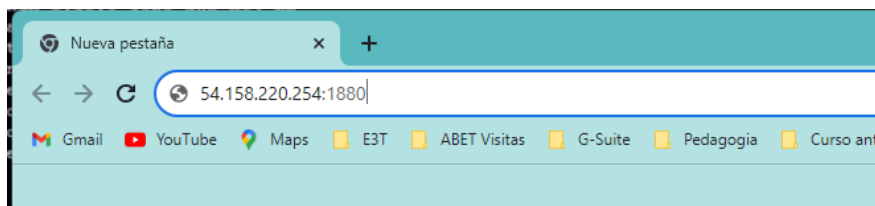
- La verdad, la manera menos riesgosa para una actualizada y buena instalación de Node-RED es siguiente [la documentación que se proporciona en la página](#) de node red

sudo npm install -g --unsafe-perm node-red	Usando nmp hacemos la instalación de Node-RED
sudo systemctl daemon-reload	Recarga la configuración del demonio del sistema para asegurarse de que los cambios realizados en la configuración o unidad systemd, hechos por la instalación de Node-RED, se apliquen correctamente.
sudo apt-get upgrade	
node-red -v	<p>Lanzamiento de Node Red. Es como cuando uno abre el editor de python o gnuadio. Entonces la consola se queda en espera. No se debe cerrar la consola porque se cerrará node red.</p> <p>Con Ctrl+C se cierra la consola, pero también la instancia de node-red. Pero no hagas esto porque la siguiente tarea requiere que Node-RED esté corriendo.</p>

Tarea 4. Solución a problemas de acceso remoto a Mosquitto y Node-RED (es lo único que se presentará en el informe a entregar)

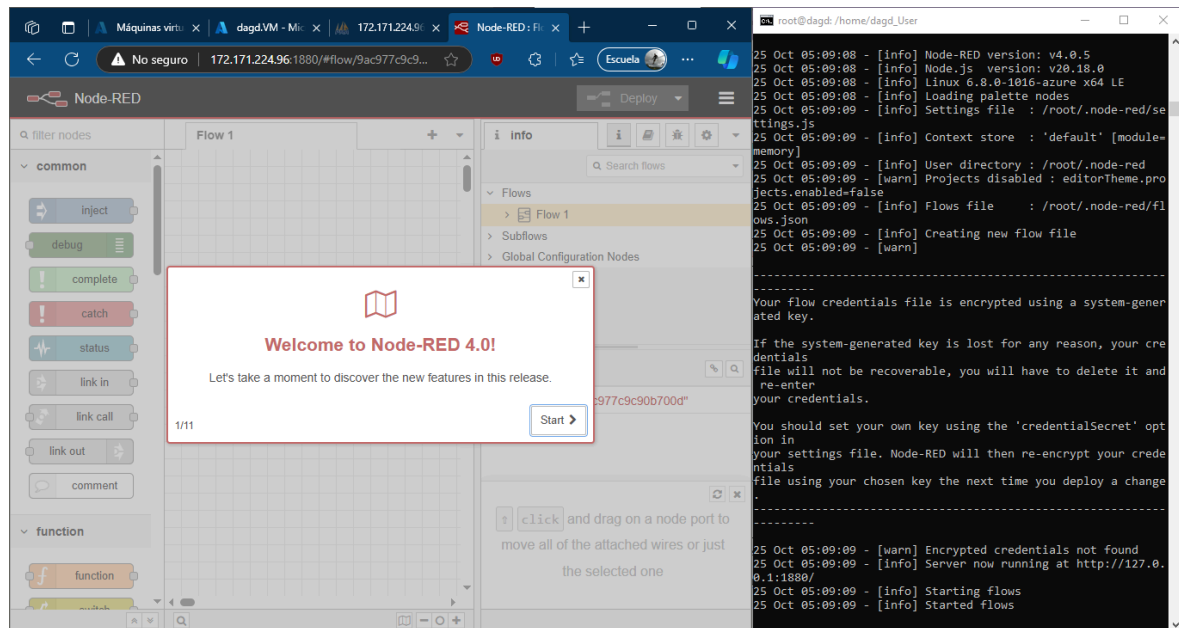
Paso 1. Prueba para llamar node-RED desde un lugar remoto

Prueba llamar node-RED desde un lugar remoto. Para ello recuerda que el puerto de Node-RED es de 1880. De manera que el proceso es, ir a un navegador y escribir la ip seguida de “:” y el puerto de este servicio que es 1880. En la siguiente figura hay un ejemplo



Toma un pantallazo del resultado obtenido en el navegador. **Nota: Lo más probable es**

que hayas fracasado en la conexión, pero no importa, toma el pantallazo.



Inicialmente se agregó una “Regla de seguridad de entrada” para hacer uso del **puerto 1880**:

Prioridad ↑↓	Nombre ↑↓	Puerto ↑↓	Protocolo ↑↓	Origen ↑↓	Destino ↑↓	Acción ↑↓
<input type="checkbox"/> 300	SSH	22	TCP	Cualquiera	Cualquiera	✓ Allow
<input type="checkbox"/> 320	HTTP	80	TCP	Cualquiera	Cualquiera	✓ Allow
<input type="checkbox"/> 340	HTTPS	443	TCP	Cualquiera	Cualquiera	✓ Allow
<input type="checkbox"/> 350	Permitir-Node-RED	1880	TCP	Cualquiera	Cualquiera	✓ Allow

Explica en la siguiente fila cuál es la razón de este resultado. Para ello, estudia lo siguiente:

1. ¿Cuál es el rol de los puertos, en el manual de Azure, capítulo “[Términos relevantes](#)”?
2. ¿Qué son los NSG en Azure y cómo ayudan a controlar el acceso a los puertos, capítulo “[Términos relevantes](#)”?
3. ¿Cuál es el rol de los protocolos IP, en el manual Azure, capítulo “[El rol de protocolos y puertos en configuraciones de seguridad](#)”?

1. **Rol de los puertos en el capítulo “Términos relevantes”:** Los puertos en Azure permiten que un servidor ofrezca múltiples servicios de manera simultánea, cada uno identificado por un número de puerto específico. Al igual que una tarjeta de acceso que permite la entrada a ciertas puertas en un edificio, los puertos ayudan a controlar quién tiene acceso a qué servicios en el servidor. Esto mejora la seguridad y facilita la administración del tráfico de red, ya que permite definir reglas para limitar o permitir el acceso a ciertos servicios según sea necesario.
2. **NSG (Network Security Group) y su papel en el control de acceso a puertos:** Un NSG es un conjunto de reglas de seguridad que controla el tráfico de red hacia y desde los recursos en una red virtual (VNet) de Azure. Similar a un firewall, un NSG puede definir permisos de acceso basados en puertos, IPs de origen y destino, y protocolos. Esto permite, por ejemplo, restringir el tráfico de entrada y salida de una máquina virtual o una subred en función del origen, destino, y tipo de tráfico, proporcionando un control detallado de acceso en los puertos utilizados.
3. **Rol de los protocolos IP en el capítulo “El rol de protocolos y puertos en**

configuraciones de seguridad”: Los protocolos IP son fundamentales en las configuraciones de seguridad, ya que cada protocolo define un tipo de comunicación específico (por ejemplo, TCP para conexiones fiables y UDP para transmisiones más rápidas pero no garantizadas). En la configuración de NSG, se puede especificar qué protocolos están permitidos en ciertos puertos. Esto ayuda a definir y restringir el tipo de tráfico que puede acceder a los recursos de la red, como en el caso de habilitar solo HTTP o HTTPS para la web o MQTT para IoT, aumentando la seguridad al limitar la superficie de ataque.

Investiga cual es el NSG asignado a tu VM - sigue el [Manual Azure, capítulo Caso en que quieres conocer la NSG de tu VM](#) . Escribe abajo de nombre del NSG

Nombre del NSG: dagd.VM-nsg


Prioridad	Nombre	Puerto	Protocolo	Origen	Destino	Acción
Reglas de seguridad de entrada						
300	SSH	22	TCP	Cualquiera	Cualquiera	Allow
320	HTTP	80	TCP	Cualquiera	Cualquiera	Allow
340	HTTPS	443	TCP	Cualquiera	Cualquiera	Allow
350	Permitir-Node-RED	1880	TCP	Cualquiera	Cualquiera	Allow
65000	AllowVnetInBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Cualquiera	Cualquiera	AzureLoadBalancer	Cualquiera	Allow
65500	DenyAllInBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Deny
Reglas de seguridad de salida						
65000	AllowVnetOutBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Cualquiera	Cualquiera	Cualquiera	Internet	Allow
65500	DenyAllOutBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Deny

Agrega al NSG una “regla de puerto de entrada” que permita el tráfico a Node-RED, sabiendo que a Node-RED usa el puerto 1880. Sigue el [Manual Azure, capítulo “Caso en que quieres editar el NSG de tu VM directamente en la VM”](#)

Origen: any
Intervalos de puertos de origen: *
Destino: Any
Servicio: custom
Intervalos de puertos de destino: 1880
Prioridad: 100
Nombre: EntradaNodeRed
Acción: Allow

Después de unos 40 segundos, prueba nuevamente acceder desde un navegador a Node-RED. Pega un pantallazo del resultado en la siguiente fila

Adición de la “Regla de puerto de entrada”:

 **Perimitir-Node-RED**
dagd.VM-nsg

Origen ⓘ

Any

Intervalos de puertos de origen * ⓘ

*

Destino ⓘ

Any

Servicio ⓘ

Custom

Intervalos de puertos de destino * ⓘ

1880

Protocolo

☐ Any


☒ TCP

☐ UDP

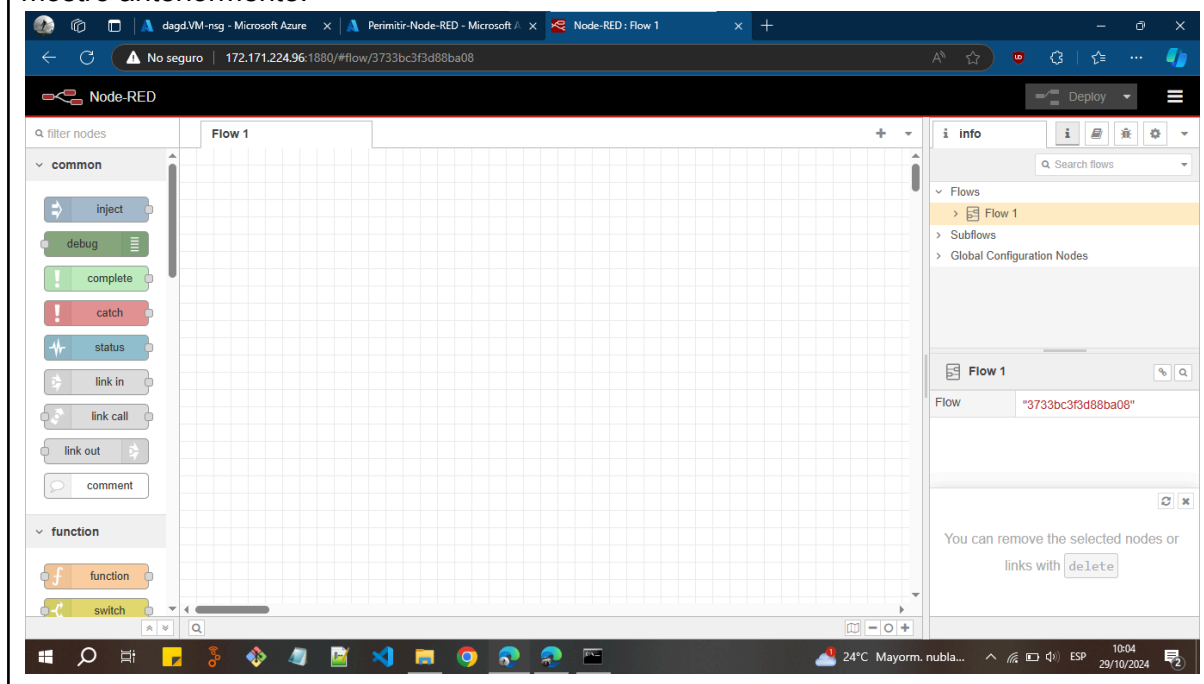
☐ ICMPv4

Guardar

Cancelar

 Enviar comentarios

Y finalmente se comprueba el acceso a Node-RED desde el navegador como ya se mostró anteriormente:



Paso 2. Prueba para invocar a Mosquitto desde un lugar remoto.

Mosquitto usa el protocolo MQTT que no es apto para navegadores. Entonces, para la prueba recurriremos al uso Python. Pasa a Google Colab, saca una copia del cuaderno de [este enlace](#) y usalo en la prueba

Escribe abajo el resultado obtenido. **Nota: Lo más probable es que hayas fracasado en la conexión, pero no importa, toma el pantallazo**




The screenshot shows a Jupyter Notebook titled "Fase II - Taller 2: mosquitto.pruebaConexion.ipynb". The interface includes a menu bar with options like "Archivo", "Editar", "Ver", "Insertar", "Entorno de ejecución", "Herramientas", "Ayuda", and a status bar indicating "Se guardaron todos los cambios". The notebook contains several code cells:

- Cell [2]: `# Importación de las librerías requeridas`
`import telnetlib`
- Cell [9]: `# Definición de la función necesaria para la pruebas`
`def check_mqtt_port(ip, port):`
 `try:`
 `# Intentar conectarse al puerto especificado`
 `telnetlib.Telnet(ip, port, timeout=10)`
 `print(f"Conexión exitosa a {ip} en el puerto {port}")`
 `except Exception as e:`
 `print(f"Horrorrrr. Error al conectar a {ip} en el puerto {port}: {e}")`
- Cell [8]: `# Los datos de la máquina que deseo probar`
`# Dirección IP y puerto de su VM de Azure`
`#ip = "40.71.16.83"`
`ip = "172.171.224.96"`
`port = 1883`
- Cell [10]: `# La prueba en sí de la conexión al puerto 1883 o puerto MQTT`
`check_mqtt_port(ip, port)`

The output of the last cell shows the error message: "Horrorrrr. Error al conectar a 172.171.224.96 en el puerto 1883: timed out". The status bar at the bottom indicates "✓ 10 s se ejecutó 8:59 p.m."

Usa lo aprendido sobre NSG para lograr corregir el fracaso en la conexión. Pregunta a GPT qué puertos deberían ser habilitados para un amplio uso de Mosquitto y usa esa información en la configuración de la regla de seguridad. Pega un pantallazo de la nueva configuración del NSG

Revisando el Manual de Azure, se encontró que el puerto a usar para Mosquitto es el 1883 y se agrega una nueva "Regla de puerto de entrada" para que se haga efectiva la comunicación:

 **Agregar regla de seguridad de entrada** ×

dagd.VM-nsg

Servicio ⓘ

Custom ▼

Intervalos de puertos de destino * ⓘ

1883 ✓

Protocolo

☐ Any

☒ TCP

☐ UDP

☐ ICMPv4

Acción

☒ Permitir

☐ Denegar

Prioridad * ⓘ


360

Nombre *

Permitir-Mosquitto ✓

Agregar

Cancelar

 [Enviar comentarios](#)

Además de agregar el puerto 1883 al firewall con los siguientes comandos:

```
sudo ufw allow 1883
```

```
sudo ufw status (Para verificar que quedó añadido el puerto 1883)
```

Prueba nuevamente acceder desde Python a Mosquitto. Pega un pantallazo del resultado en la siguiente fila



Fase II - Taller 2: mosquitto.pruebaConexion.ipynb



Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se guardaron todos los cambios](#)



+ Código + Texto



✓ [21] # Importación de las librerías requeridas

0 s import telnetlib

✓ [22] # Definición de la función necesaria para la pruebas

0 s def check_mqtt_port(ip, port):
 try:
 # Intentar conectarse al puerto especificado
 telnetlib.Telnet(ip, port, timeout=10)
 print(f"Conexión exitosa a {ip} en el puerto {port}")
 except Exception as e:
 print(f"Horrorrr. Error al conectar a {ip} en el puerto {port}: {e}")

✓ [23] # Los datos de la máquina que deseo probar

0 s # Dirección IP y puerto de su VM de Azure

#ip = "40.71.16.83"

ip = "172.171.224.96"

port = 1883

✓ # La prueba en sí de la conexión al puerto 1883 o puerto MQTT

0 s check_mqtt_port(ip, port)

→ Conexión exitosa a 172.171.224.96 en el puerto 1883

✓ 0 s se ejecutó 10:01 p.m.