



UNIVERSIDAD TECNOLÓGICA NACIONAL - FACULTAD REGIONAL TUCUMAN

VIRTUALIZACIÓN

CONSOLIDACIÓN DE SERVIDORES

— TRABAJO PRACTICO FINAL: CASO DE ESTUDIO

NEME, MARIA BELEN

LEGAJO: **42215**



Virtualización: Consolidación de Servidores

Trabajo Práctico Final: Caso de Estudio

Elaborar una solución de un Blog Personal que deberá contemplar los siguientes requerimientos:

1. Listar las categorías de los temas
2. Detallar los datos personales del Autor e incorporar la foto del mismo
3. El primer y único artículo deberá ser la presentación del documento correspondiente al resultado final de este trabajo práctico final, adjuntando el documento (PDF) del mismo
4. Se deberá utilizar contenedores. Deberán ser configurados al menos dos. Uno con todo lo concerniente a la interfaz y otro con la base de datos
5. Se deberá elegir las herramientas para trabajar el lo propuesto y se deberá detallar cada una de ellas en el informe final
6. Se deberá realizar un detalle de los recursos utilizados por cada Contenedor (con sus respectivas categorías) e incluirlo en el informe final

Tecnologías utilizadas

Frontend

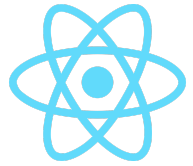
JavaScript

Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.



React

React es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página



HTML5

HTML es un lenguaje de marcado que nos permite indicar la estructura de nuestro documento mediante etiquetas. Este lenguaje nos ofrece una gran adaptabilidad, una estructuración lógica y es fácil de interpretar tanto por humanos como por máquinas.



CSS

en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado



Bootstrap

Bootstrap es una biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web.



Backend

NodeJS

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

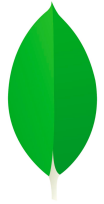


Express

Express.js o simplemente Express, es un entorno de trabajo para aplicaciones web para el programador Node.js, de código abierto y con licencia MIT. Se utiliza para desarrollar aplicaciones web y APIs

MongoDB

Es un sistema de base de datos NoSQL, orientado a documentos y de código abierto. MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.



Pasos de la implementación

Creación de contenedores

Se realizó la creación de dos contenedores utilizando la plataforma de Proxmox. Uno de ellos para ser utilizado para el frontend y otro para dar soporte al backend. En ambos contenedores se instaló Ubuntu 20.4 como sistema operativo.

Características de los contenedores

Detalles del Contenedor BE

Nombre: 39698637db

CPU: 1 núcleo

Memoria: 512 Mb

Almacenamiento: 2 Gb

Detalles del Contenedor FE

Nombre: 39698637i

CPU: 1 núcleo

Memoria: 512 Mb

Almacenamiento: 2 Gb

Instalaciones de programas en contenedor para Backend

1. Se realizó la instalación del comando curl
`apt install curl`
2. Se realizó la instalación de node con las siguientes instrucciones
`curl -sL https://deb.nodesource.com/setup_14.x -o nodesource_setup.sh`
`nano nodesource_setup.sh`
`bash nodesource_setup.sh`

apt install nodejs

3. Se realizó la instalación de mongoDB para el manejo de la base de datos
curl -fsSL https://www.mongodb.org/static/pgp/server-6.0.asc | sudo
apt-key add -
echo "deb [arch=amd64,arm64] https://repo.mongodb.org/apt/ubuntu
focal/mongodb-org/6.0 multiverse" | sudo tee
/etc/apt/sources.list.d/mongodb-org-6.0.list
apt update
apt install mongodb-org
4. Se realizó la instalación de git para poder clonar el proyecto donde tenemos
alojada nuestra API que permitirá la conexión a la DB
apt install git

```
root@39698637db:~# sudo apt-get install -y nodejs
sudo: setrlimit(RLIMIT_CORE): Operation not permitted
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  nodejs
0 upgraded, 1 newly installed, 0 to remove and 173 not upgraded.
Need to get 25.4 MB of archives.
After this operation, 123 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_14.x focal/main amd64 nodejs amd64 14.20.0-deb-1nodesource1 [25.4
MB]
Fetched 25.4 MB in 3s (9454 kB/s)
Selecting previously unselected package nodejs.
(Reading database ... 17703 files and directories currently installed.)
Preparing to unpack .../nodejs_14.20.0-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (14.20.0-deb-1nodesource1) ...
```

```
Hit:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu focal-security InRelease
Get:5 https://deb.nodesource.com/node_14.x focal/main amd64 Packages [774 B]
Fetched 5357 B in 1s (4389 B/s)
Reading package lists... Done

## Run `sudo apt-get install -y nodejs` to install Node.js 14.x and npm
## You may also need development tools to build native addons:
  sudo apt-get install gcc g++ make
## To install the Yarn package manager, run:
  curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | gpg --dearmor | sudo tee /usr/share/keyrings/y
arnkey.gpg >/dev/null
  echo "deb [signed-by=/usr/share/keyrings/yarnkey.gpg] https://dl.yarnpkg.com/debian stable main" |
  sudo tee /etc/apt/sources.list.d/yarn.list
  sudo apt-get update && sudo apt-get install yarn

root@39698637db:~# sudo apt-get install -y nodejs
```

Instalaciones de programas en contenedor para Frontend

1. Se realizó la instalación del comando curl
apt install curl
2. Se realizó la instalación de node con las siguientes instrucciones
curl -sL https://deb.nodesource.com/setup_14.x -o nodesource_setup.sh
nano nodesource_setup.sh
bash nodesource_setup.sh
apt install nodejs

3. Se realizó la instalación de git para poder clonar el proyecto donde tenemos alojada nuestra API que permitirá la conexión a la DB
apt install git

Prueba y Ejecución

Desde nuestro contenedor de la Base de Datos, realizamos las siguientes instrucciones:

1. Clonamos el proyecto donde tenemos alojada nuestra API con el comando "git clone"
2. Una vez clonado el proyecto, accedemos al mismo y ejecutamos "npm install" para descargar todas las dependencias necesarias
3. Para levantar la API, ejecutamos "npm start"

Desde nuestro contenedor dedicado a la interfaz, realizamos las siguientes instrucciones:

1. Clonamos el proyecto donde tenemos alojada nuestro frontend con el comando "git clone"
2. Una vez clonado el proyecto, accedemos al mismo y ejecutamos "npm install" para descargar todas las dependencias necesarias
3. Para levantar la API, ejecutamos "npm start"
4. Accedemos a la ip pública del mismo, y debemos comprobar que el blog se encuentra en servicio