

# Virtualización y Consolidación de Servidores

## Trabajo Final: Informe



Integrante:

- Bejas Víctor Adrian [46255]

## **Caso de estudio**

Para el siguiente trabajo final integrador de la cátedra de Virtualización y Consolidación de Servidores se propuso la elaboración e implementación de un Blog personal que debe contemplar las siguientes funcionalidades:

- Listar las categorías de los temas.
- Detallar los datos personales del Autor e incorporar la foto del mismo.
- 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.

Para realizar este trabajo se hizo uso de la plataforma Proxmox que es una plataforma de virtualización Open Source, en este caso se utilizó el nodo [bejuca.hopto.org:18006](https://bejuca.hopto.org:18006) proporcionado por la cátedra.

Para el mismo se optó por una Arquitectura cliente-servidor que estarán alojadas en contenedores distintos. A continuación se muestran los mismo con más detalles:

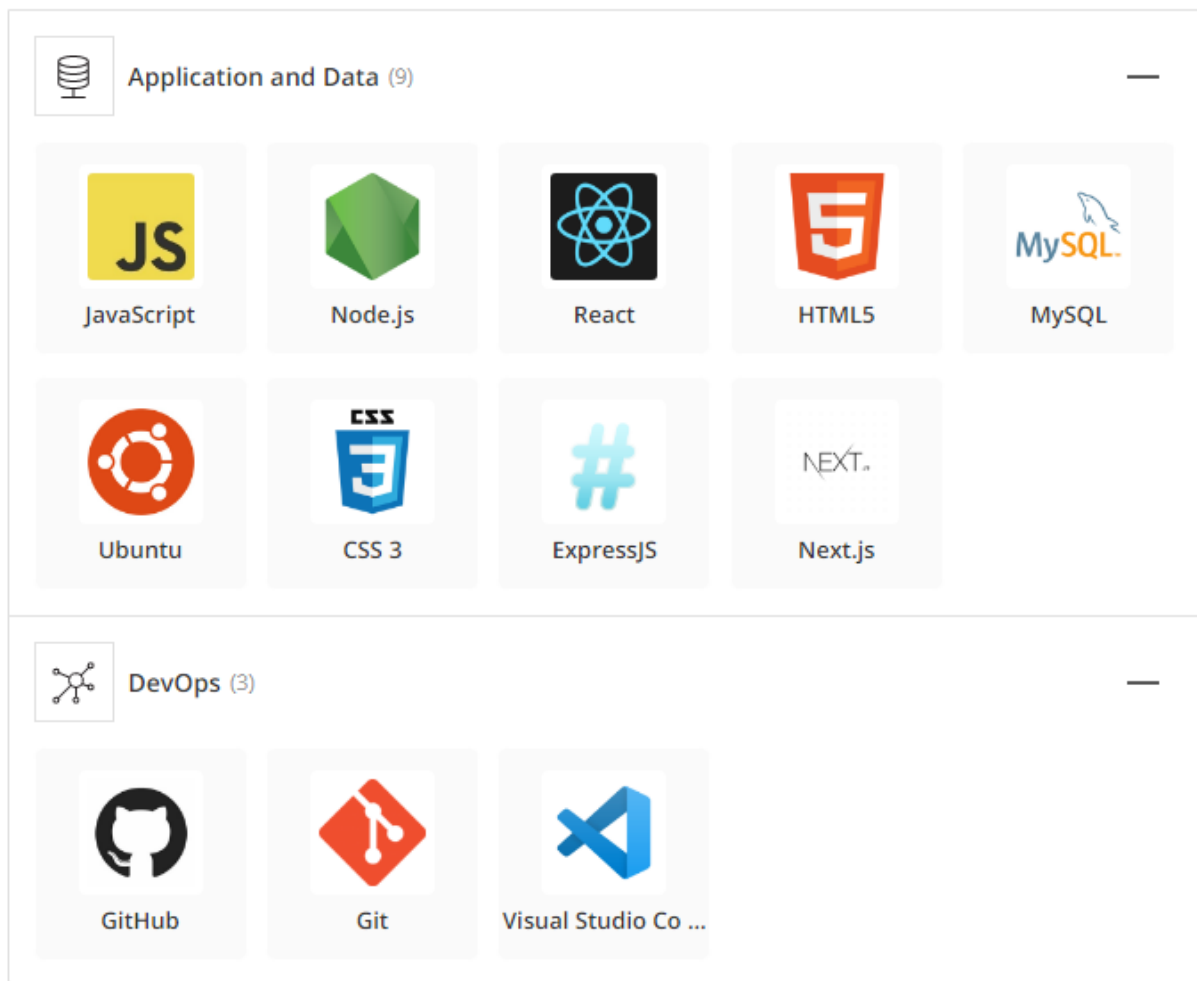
Nombre Contenedor	Servicio	Especificaciones
40697516i	Cliente	Procesador: 1 CPU(1 core) Memoria: 512 MiB Almacenamiento: 10 GiB
40697516db	Servidor	Procesador: 1 CPU(1 core) Memoria: 512 MiB Almacenamiento: 10 GiB

En cuanto a la categoría de red, ambos contenedores están obteniendo una dirección ip por dhcp. Para poder llevar a cabo este proyecto, se necesita tener abierto un puerto, para poder acceder red vía pública. En la siguiente tabla se detallan más aspectos sobre la categoría red a tener en cuenta:

Contenedores	Puertos solicitados	Dirección IP	Puertos asignados
40697516i	3000	192.168.88.128:3000	bejuca.hopto.org:11015

## Detalles sobre el stack tecnológico utilizado para realizar la implementación

- ❖ Para desarrollar el BackEnd, las herramientas utilizadas fueron:
  - JavaScript: Lenguaje de programación orientado a objetos, débilmente tipado y dinámico.
  - Node.js: Es un entorno que trabaja en tiempo de ejecución, de código abierto, multiplataforma, que permite crear toda clase de herramientas de lado servidor y aplicaciones en JavaScript.
  - Express.js: Es el Framework más popular de Node.js, se encarga del manejo de las peticiones con diferentes verbos HTTP.
  - Mysql: Es un sistema de gestión de bases de datos relacional que permite almacenar y acceder a los datos a través de múltiples motores de almacenamiento.
- ❖ Para el desarrollo del blog, las herramientas utilizadas fueron:
  - HTML y CSS: Lenguaje de marcado para la construcción de la estructura de la web (html) y lenguaje de estilos para diseñar sobre lenguaje de marcado.
  - JavaScript: Lenguaje de programación orientado a objetos, débilmente tipado y dinámico.
  - React: Biblioteca de JavaScript para construir interfaces de usuario.
  - Next.js: Framework de react para la construcción de aplicaciones web, dando prioridad a una mejor experiencia de usuario basada en rendimiento.
- ❖ Las herramientas utilizadas para el desarrollo y el versionado del código:
  - Git: Es un sistema distribuido de control de versiones, gratuito y de código abierto basado en repositorios para almacenar las diferentes versiones del código fuente.
  - GitHub: Es un servicio de alojamiento en la nube de código fuente basado en el sistema de control de versiones que Git ofrece para manejar repositorios.
  - Visual Studio Code: Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web.



## IMPLEMENTACIÓN

Para la implementación en primer lugar se crearon los contenedores en la plataforma Proxmox con los recursos anteriormente mencionados, donde se instaló el sistema operativo Ubuntu 20.04 LTS.

Contenedor para el cliente

```
Ubuntu 20.04 LTS 40697516i tty1
40697516i login: root
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.106-1-pve x86_64)

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

Last login: Fri Jun 17 14:06:18 UTC 2022 on tty1
root@40697516i:~#
```

Contenedor para el servidor

```

Ubuntu 20.04 LTS 40697516db tty1

40697516db login: root
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.106-1-pve x86_64)

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

Last login: Fri Jun 17 14:00:15 UTC 2022 on tty1
root@40697516db:~#

```

Para el contenedor donde estará alojada la API y la base de datos, tenemos que instalar los servicios de Mysql y de Node.js. Al momento de crear la API con Node.js, la misma necesita de otros servicios como ser Express para montar el servidor HTTP que se encargue de la gestión de las peticiones que llegan al mismo.

Los comandos utilizados para la instalación de nodejs en la versión 14.x fueron:

- `sudo apt update`
- `apt install curl`
- `curl -sL https://deb.nodesource.com/setup_14.x -o nodesource_setup.sh`
- `nano nodesource_setup.sh`
- `sudo bash nodesource_setup.sh`
- `sudo apt-get install -y nodejs`

```

GNU nano 4.8                                nodesource_setup.sh
#!/bin/bash

# Discussion, issues and change requests at:
#   https://github.com/nodesource/distributions
#
# Script to install the NodeSource Node.js 14.x repo onto a
# Debian or Ubuntu system.
#
# Run as root or insert `sudo -E` before `bash`:
#
# curl -sL https://deb.nodesource.com/setup_14.x | bash -
# or
# wget -qO- https://deb.nodesource.com/setup_14.x | bash -
#
# CONTRIBUTIONS TO THIS SCRIPT
#
[ Read 361 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^R 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

```

```

root@40697516db:~# 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 14.19.3-deb-1nodesource1 [25.4 MB]
Fetched 25.4 MB in 4s (6854 kB/s)
Selecting previously unselected package nodejs.
(Reading database ... 17703 files and directories currently installed.)
Preparing to unpack .../nodejs_14.19.3-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (14.19.3-deb-1nodesource1) ...
Setting up nodejs (14.19.3-deb-1nodesource1) ...
Processing triggers for man-db (2.9.1-1) ...
root@40697516db:~# node -v
v14.19.3
root@40697516db:~#

```

### *Instalación node en el contenedor 40697516db*

Los comandos utilizados para la instalación de mysql fueron:

- `sudo apt install mysql-server`

Para establecer contraseña al usuario root de mysql:

- `ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'vab199712';`
- `flush privileges;`

```

e.
Setting up libhttp-date-perl (6.05-1) ...
Setting up libhtml-tagset-perl (3.20-4) ...
Setting up liblwp-mediatypes-perl (6.04-1) ...
Setting up libencode-locale-perl (1.05-1) ...
Setting up libhtml-parser-perl (3.72-5) ...
Setting up libhttp-message-perl (6.22-1) ...
Setting up mysql-server (8.0.29-0ubuntu0.20.04.3) ...
Setting up libcgi-pm-perl (4.46-1) ...
Setting up libhtml-template-perl (2.97-1) ...
Setting up libcgi-fast-perl (1:2.15-1) ...
Processing triggers for systemd (245.4-4ubuntu3) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
root@40697516db:~# mysql -v
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.29-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Reading history-file /root/.mysql_history
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

```

40697516db login: root
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.106-1-pve x86_64)

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

* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.

  https://ubuntu.com/blog/microk8s-memory-optimisation
Last login: Sat Jun 18 19:33:39 UTC 2022 on tty1
root@40697516db:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.29-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 

```

### *Instalación mysql en el contenedor 40697516db*

Para el contenedor donde estará alojado el FrontEnd, es requerido tener instalado Node.js. Cabe recalcar que el proyecto está construido con el framework Next.js el cual al momento de instalarlo, requiere de dependencias de librerías para su funcionamiento, como pueden ser React.js.

Los comandos utilizados para la instalación de nodejs en la versión 14.x fueron:

- sudo apt update
- apt install curl
- curl -sL https://deb.nodesource.com/setup\_14.x -o nodesource\_setup.sh
- nano nodesource\_setup.sh
- sudo bash nodesource\_setup.sh
- sudo apt-get install -y nodejs

```

GNU nano 4.8                                nodedsource_setup.sh
#!/bin/bash

# Discussion, issues and change requests at:
#   https://github.com/nodedsource/distributions
#
# Script to install the NodeSource Node.js 14.x repo onto a
# Debian or Ubuntu system.
#
# Run as root or insert `sudo -E` before `bash`:
#
# curl -sL https://deb.nodesource.com/setup_14.x | bash -
# or
# wget -qO- https://deb.nodesource.com/setup_14.x | bash -
#
# CONTRIBUTIONS TO THIS SCRIPT
#
# This script is built from a template in
# https://github.com/nodedsource/distributions/tree/master/deb/src
# please don't submit pull requests against the built scripts.
#

export DEBIAN_FRONTEND=noninteractive
SCRSUFFIX="_14.x"
NODENAME="Node.js 14.x"

root@40697516i:~#

```

```

root@40697516i:~# 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 14.19.3-deb-1nodesource1 [25.4 MB]
Fetched 25.4 MB in 4s (6499 kB/s)
Selecting previously unselected package nodejs.
(Reading database ... 17703 files and directories currently installed.)
Preparing to unpack .../nodejs 14.19.3-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (14.19.3-deb-1nodesource1) ...
Setting up nodejs (14.19.3-deb-1nodesource1) ...
Processing triggers for man-db (2.9.1-1) ...
root@40697516i:~# node -v
v14.19.3
root@40697516i:~#

```

### *Instalación node en el contenedor 40697516i*

En ambos contenedores se realizó la instalación del servicio de git, para desde ahí obtener el código de la API(BackEnd) y el FrontEnd. Se hizo uso de la plataforma de versionado de código GitHub para alojar el ambos proyectos.



```

root@40697516db:~# apt install git
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man libcurl3-gnutls liberror-perl patch
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki
  git-svn diffutils-doc
The following NEW packages will be installed:
  git git-man libcurl3-gnutls liberror-perl patch
0 upgraded, 5 newly installed, 0 to remove and 172 not upgraded.
Need to get 5809 kB of archives.
After this operation, 39.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 libcurl3-gnutls amd64 7.68.0-1ubuntu2.11 [233
kB]
Get:2 http://archive.ubuntu.com/ubuntu focal/main amd64 liberror-perl all 0.17029-1 [26.5 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 git-man all 1:2.25.1-1ubuntu3.4 [885 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 git amd64 1:2.25.1-1ubuntu3.4 [4560 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal/main amd64 patch amd64 2.7.6-6 [105 kB]
Fetched 5809 kB in 2s (2388 kB/s)
Selecting previously unselected package libcurl3-gnutls:amd64.
(Reading database ... 25204 files and directories currently installed.)
Preparing to unpack .../libcurl3-gnutls_7.68.0-1ubuntu2.11_amd64.deb ...
Unpacking libcurl3-gnutls:amd64 (7.68.0-1ubuntu2.11) ...
Selecting previously unselected package liberror-perl.
Preparing to unpack .../liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...

```

```

Selecting previously unselected package patch.
Preparing to unpack .../patch_2.7.6-6_amd64.deb ...
Unpacking patch (2.7.6-6) ...
Setting up libcurl3-gnutls:amd64 (7.68.0-1ubuntu2.11) ...
Setting up liberror-perl (0.17029-1) ...
Setting up patch (2.7.6-6) ...
Setting up git-man (1:2.25.1-1ubuntu3.4) ...
Setting up git (1:2.25.1-1ubuntu3.4) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
root@40697516db:~# git --version
git version 2.25.1
root@40697516db:~#

```

*Instalación git contenedor 40697516db*

```

Setting up liberror-perl (0.17029-1) ...
Setting up git (1:2.25.1-1ubuntu3.4) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9) ...
root@40697516i:~# git --version
git version 2.25.1
root@40697516i:~#

```

*Instalación git contenedor 40697516i*

El próximo paso es clonar en el contenedor 40697516db la API que se encuentra alojada en la plataforma GitHub, para lo cual se utilizaron el siguiente comando:

- `git clone https://github.com/vab1997/API-VIRT.git`

```

root@40697516db:~# git clone https://github.com/vab1997/API-VIRT.git
Cloning into 'API-VIRT'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 9 (delta 0), reused 9 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), 20.27 KiB | 399.00 KiB/s, done.

```

Para poder crear la base de datos con las tablas correspondientes, tenemos que hacer uso de los siguientes comandos:

- CREATE DATABASE database\_tfi;
- USE database\_tfi;
- INSERT INTO datosalumno (id, nombre, apellido, legajo, asignatura, comision) VALUES (1, 'Victor', 'Bejas', '46255', 'Virtualizacion', '5K3');
- INSERT INTO categoria (id, nombre, servicio, procesador, memoria, almacenamiento) VALUES (1, '40697516db', 'Servidor', '1 CPU (1 core)', '512 MiB', '10 GiB');
- INSERT INTO categoria (id, nombre, servicio, procesador, memoria, almacenamiento) VALUES (2, '40697516i', 'Cliente', '1 CPU (1 core)', '512 MiB', '10 GiB');

```

mysql> USE database_tfi;
Database changed
mysql> CREATE TABLE categoria (
  id INT(11) NOT NULL,
  nombre VARCHAR(255) NOT NULL,
  servicio VARCHAR(255) NOT NULL,
  procesador VARCHAR(255) NOT NULL,
  memoria VARCHAR(255) NOT NULL,
  almacenamiento VARCHAR(255) NOT NULL
);

ALTER TABLE categoria
  ADD PRIMARY KEY (id);

ALTER TABLE categoria
  MODIFY id INT(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT = 1;Query OK, 0 rows affected, 1 warning (0.06 sec)

mysql>
mysql> ALTER TABLE categoria
  ADD PRIMARY KEY (id);
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
mysql> ALTER TABLE categoria
  MODIFY id INT(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT = 1;
Query OK, 0 rows affected, 1 warning (0.08 sec)

```

```
mysql> CREATE TABLE datosalumno (
  d    -> id INT(11) NOT NULL,
  nomb -> r nombre VARCHAR(150) NOT NULL,
  apellid -> o apellido VARCHAR(255) NOT NULL,
  le    -> g legajo INT(11) NOT NULL,
  asignatura -> asignatura VARCHAR(255) NOT NULL,
  c    -> comision VARCHAR(255) NOT NULL
);

-> A);
TER TABLE datosalumno
  ADD PRIMARY KEY (id);

ALTER TABLE datosalumno
  MODIFY id INT(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT = 1;Query OK, 0 rows affected, 2 warnings (0.05 s
ec)

mysql>
mysql> ALTER TABLE datosalumno
  -> ADD PRIMARY KEY (id);
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
mysql> ALTER TABLE datosalumno
  -> MODIFY id INT(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT = 1;
```

```
mysql> show tables;
+-----+
| Tables_in_database_tfi |
+-----+
| categoria               |
| datosalumno             |
+-----+
2 rows in set (0.00 sec)
```

```
mysql> INSERT INTO datosalumno (id, nombre, apellido, legajo, asignatura, comision) VALUES (1, 'Victor Adrian
', 'Bejas', '46255', 'Virtualizacion y Consolidacion de Servidores', '5K3');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO categoria (id, nombre, servicio, procesador, memoria, almacenamiento) VALUES (1, '40697516
db', 'Servidor', '1 CPU (1 core)', '512 MiB', '10 GiB');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO categoria (id, nombre, servicio, procesador, memoria, almacenamiento) VALUES (2, '40697516
i', 'Cliente', '1 CPU (1 core)', '512 MiB', '10 GiB');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from categoria;
+-----+-----+-----+-----+-----+-----+
| id | nombre      | servicio | procesador | memoria | almacenamiento |
+-----+-----+-----+-----+-----+-----+
| 1 | 40697516db | Servidor | 1 CPU (1 core) | 512 MiB | 10 GiB |
| 2 | 40697516i | Cliente | 1 CPU (1 core) | 512 MiB | 10 GiB |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from datosalumno;
+-----+-----+-----+-----+-----+-----+
| id | nombre      | apellido | legajo | asignatura | comision |
+-----+-----+-----+-----+-----+-----+
| 1 | Victor Adrian | Bejas | 46255 | Virtualizacion y Consolidacion de Servidores | 5K3 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Luego nos movemos a la carpeta donde se clonó el proyecto y ejecutamos el comando “npm install” para instalar las dependencias necesarias para que la API pueda funcionar correctamente. Y por último para poder ejecutar la API, con el comando “npm run start” queda operativa.

```

root@40697516db:~/API-VIRT# npm run dev

> api-tfi@1.0.0 dev /root/API-VIRT
> nodemon src

[nodemon] 2.0.16
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src`
Server on port 5000
DB is connected

```

De la misma forma en el contenedor 40697516i, para clonar el blog personal con el comando “git clone https://github.com/vab1997/blog-post-virt.git”, y movernos a la carpeta del proyecto y ejecutar el comando “npm install” para instalar todas las dependencias necesarias. Para levantar el proyecto, ejecutamos el comando “npm run start” para poder visualizar el blog personal.

```

root@40697516i:~# git clone https://github.com/vab1997/blog-post-virt.git
Cloning into 'blog-post-virt'...
remote: Enumerating objects: 63, done.
remote: Counting objects: 100% (63/63), done.
remote: Compressing objects: 100% (58/58), done.
remote: Total 63 (delta 4), reused 61 (delta 2), pack-reused 0
Unpacking objects: 100% (63/63), 765.66 KiB | 1.13 MiB/s, done.

```

Para ponerle operativo al proyecto del blog personal, con el comando “npm run start”:

```

root@40697516i:~/blog-post-virt# npm run dev

> client-blog-personal@0.1.0 dev /root/blog-post-virt
> next dev

ready - started server on 0.0.0.0:3000, url: http://localhost:3000
wait - compiling...
event - compiled client and server successfully in 859 ms (125 modules)
wait - compiling / (client and server)...
wait - compiling...
event - compiled client and server successfully in 3.3s (214 modules)

```

## Conclusiones

Todas las herramientas que fuimos aprendiendo a lo largo de la cursada, fueron de vital importancia para poder llevar a cabo este Trabajo Final. El mismo requirió de más investigación para poder llevarlo a cabo y obtener los resultados solicitados y esperados por la cátedra de Virtualización y Consolidación de servidores.

Este trabajo fue muy interesante ya que me permitió aprender más en profundidad como es la administración de recursos, la creación de contenedores, instalación de

servicios y la implementación de un Stack Tecnológico que dé soporte a la solución del problema planteado.

En cuanto a la plataforma Proxmox, al ser de Open Source y que esté disponible para cualquier persona que requiera de los servicios que brinda la misma, resulta la más adoptada tanto para entornos de producción como entornos de desarrollo.