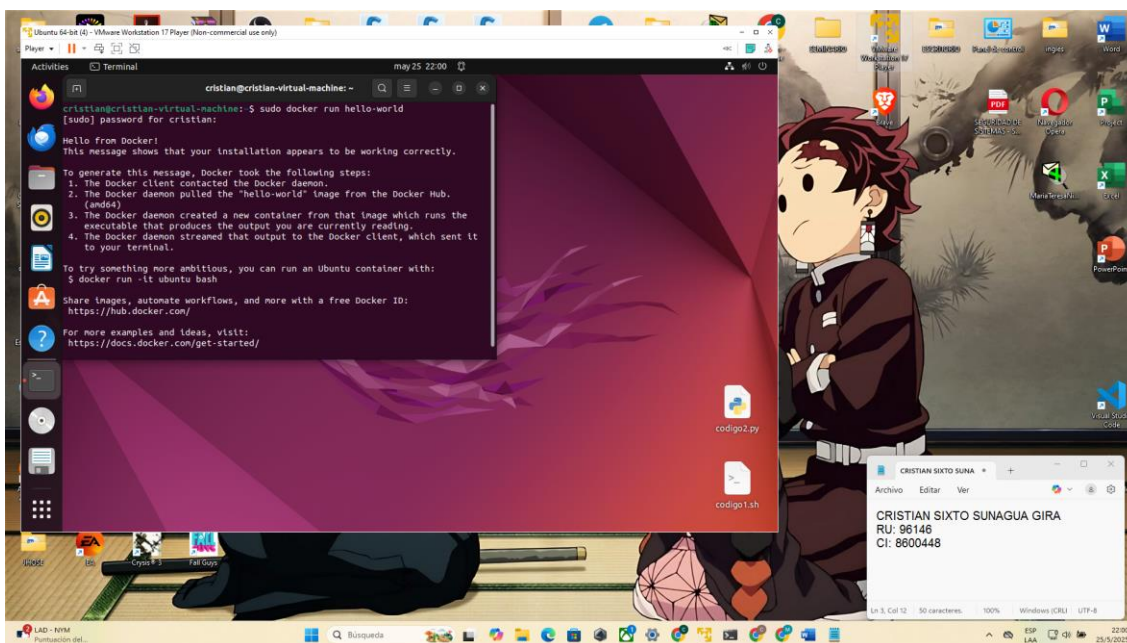
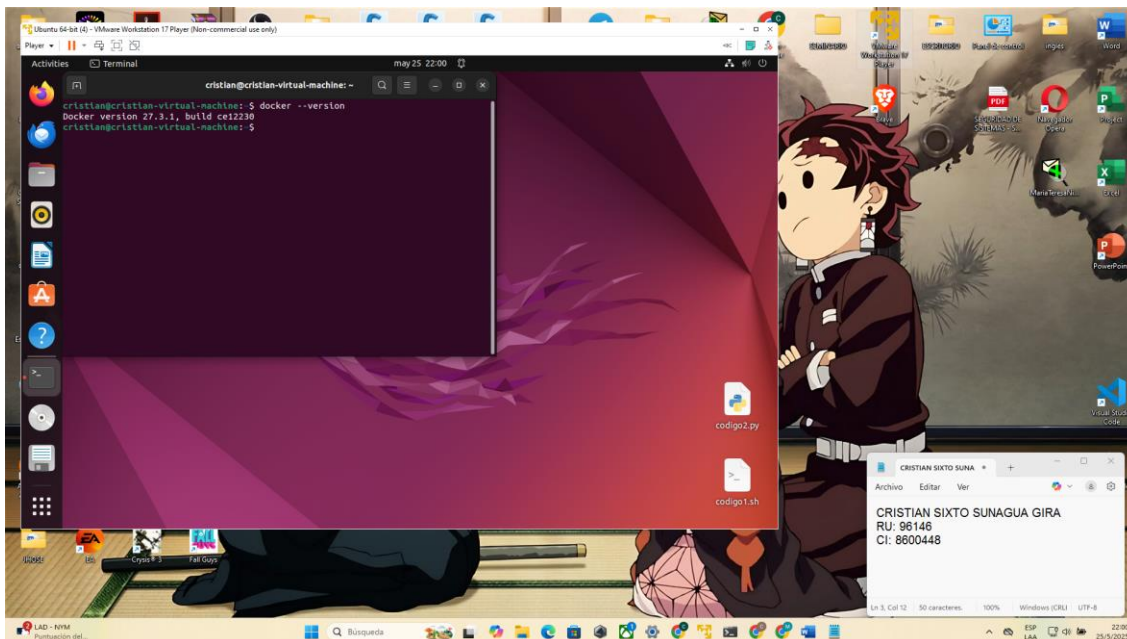


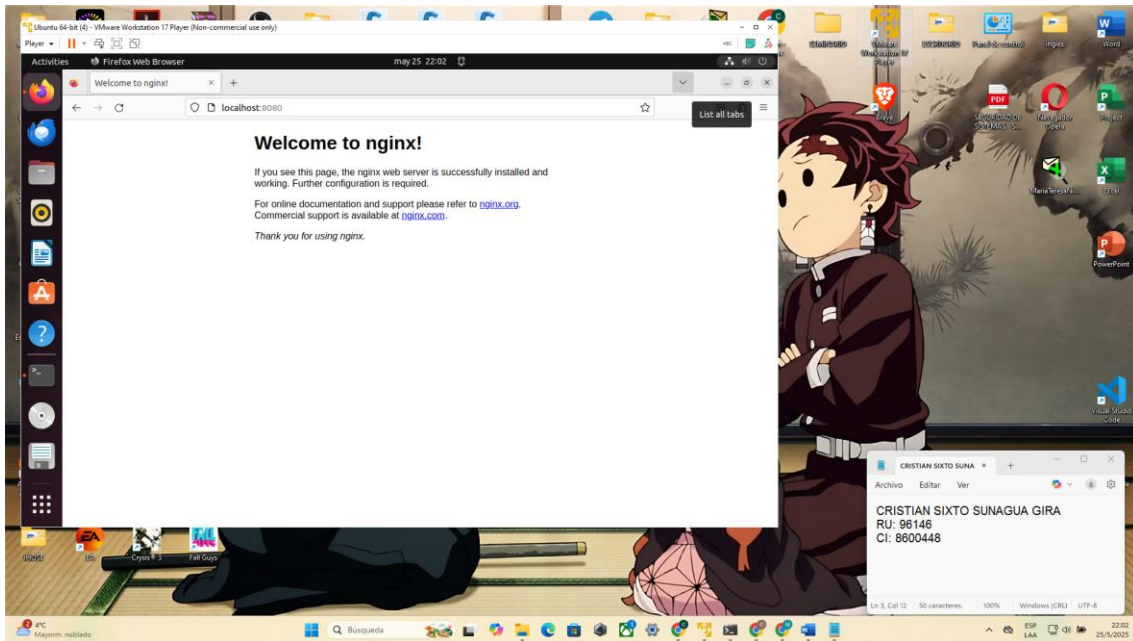
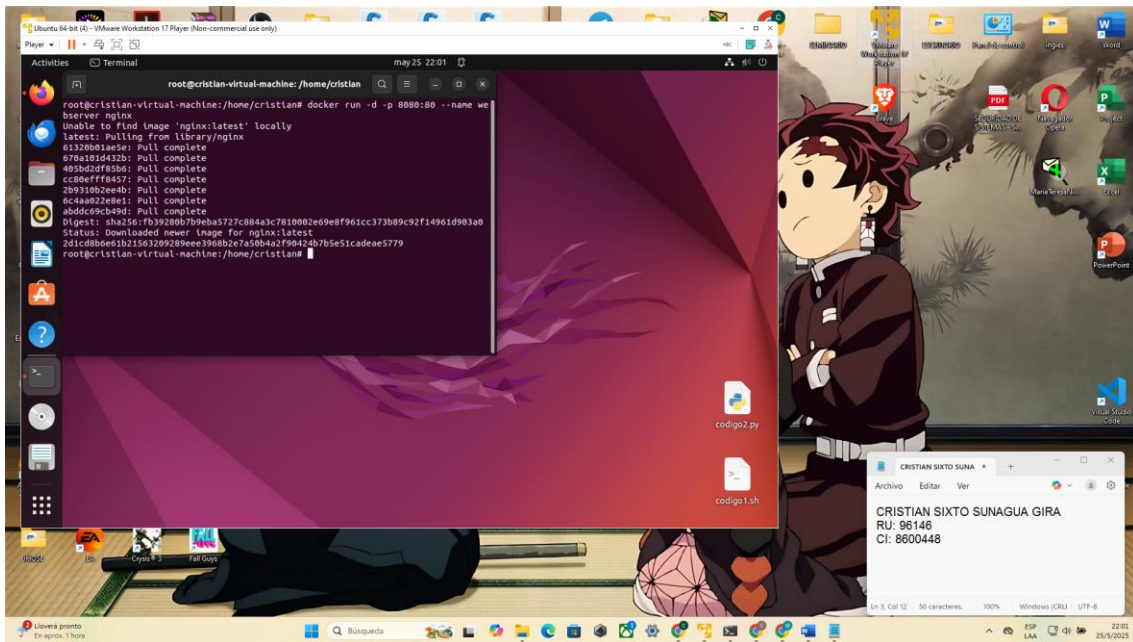


# UNIVERSIDAD AUTÓNOMA “TOMÁS FRÍAS”

## INGENIERÍA DE SISTEMAS

<b>Estudiante:</b>	Univ.Cristian Sixto Sunagua Gira			
<b>Docente:</b>	Ing. Ditmar David Castro Angulo	<b>Materia:</b>	SIS-737	<b>GRUPO 1</b>
<b>Auxiliar:</b>	Univ. Rodrigo Slucka Zárate	<b>Fecha:</b>	25/05/2025	<b>PRACTICA 2</b>
<b>Enlace:</b>	<a href="https://github.com/CSunaguaSeminario/Practica2-aux">https://github.com/CSunaguaSeminario/Practica2-aux</a>		<b>C.I:</b>	8600448







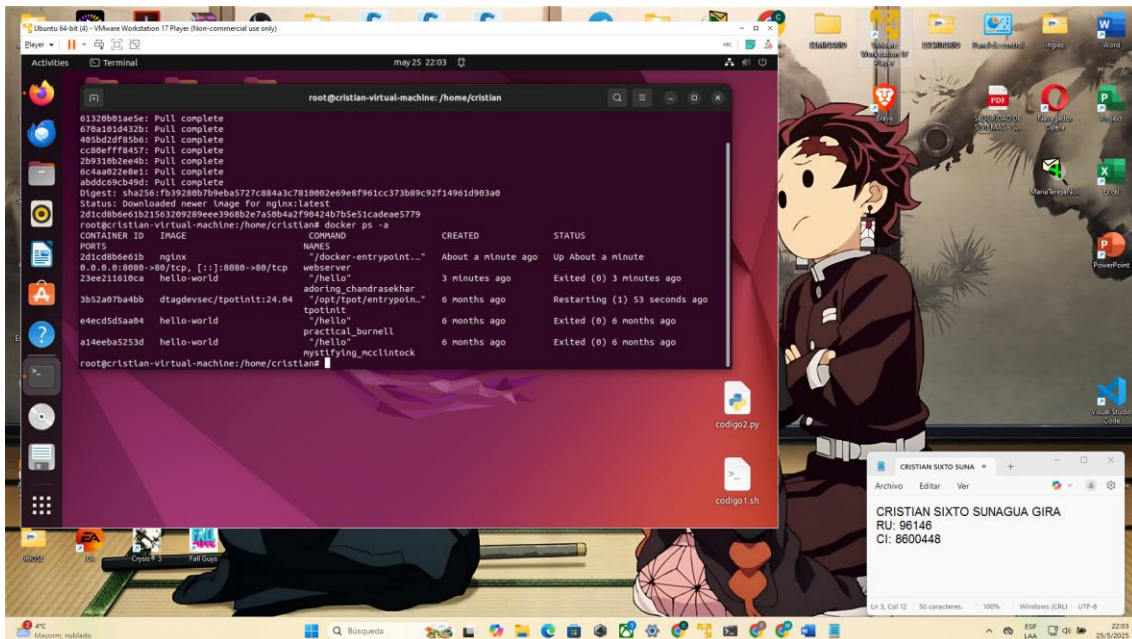
¿Qué pasa si el contenedor se detiene? ¿Cómo lo reiniciarías?

Con un docker ps -a

Esto mostrará todos los contenedores, incluidos los detenidos

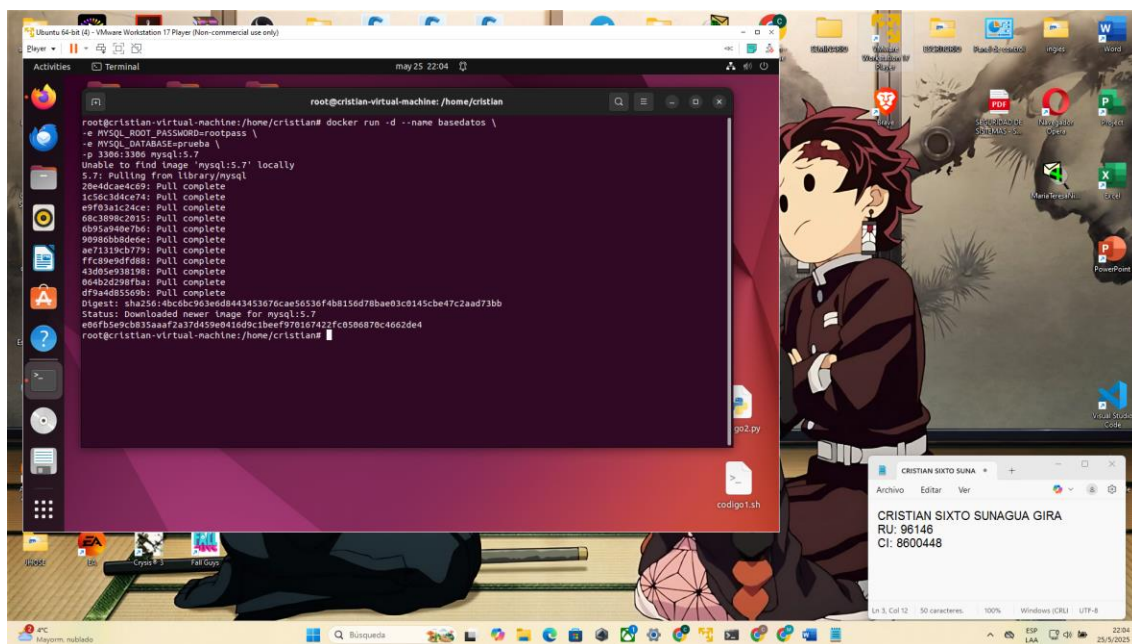
Y con un docker start webserver

“En webserver es el nombre del contenedor detenido”



```
root@cristian-virtual-machine:/home/cristian# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
61320b01a5e5	nginx	nginx	About a minute ago	Up About a minute
678a10d4132b	webserver	webserver	3 minutes ago	Exited (0) 3 minutes ago
4858d2f8f8b5	hello-world	hello-world	6 months ago	Restarting (1) 53 seconds ago
cc88efff8457	adoring_chandrasekhar	adoring_chandrasekhar	6 months ago	Exited (0) 6 months ago
2b911b2ee4b0	tpotinit	tpotinit	6 months ago	Exited (0) 6 months ago
6c4aa22ee1	practical_burnell	practical_burnell	6 months ago	Exited (0) 6 months ago
abddc9cb49d	mystifying_mcclintock	mystifying_mcclintock	6 months ago	Exited (0) 6 months ago



```
root@cristian-virtual-machine:/home/cristian# docker run -d --name basedatos \
```

```
-e MYSQL_ROOT_PASSWORD=rootpass \
```

```
-e MYSQL_DATABASE=basedatos \
```

```
-p 3306:3306 mysql:5.7
```

```
Unable to find image 'mysql:5.7' locally
```

```
5.7: Pulling from library/mysql
```

```
20e4dca4c09: Pull complete
```

```
1c36c3d4c74: Pull complete
```

```
e9f031c4c0e: Pull complete
```

```
68c3898c2015: Pull complete
```

```
6b95a948e7b6: Pull complete
```

```
90880b0bde4e: Pull complete
```

```
ae71319cb779: Pull complete
```

```
ffc89e9dfdb8: Pull complete
```

```
430b5ae938198: Pull complete
```

```
064b2d298fba: Pull complete
```

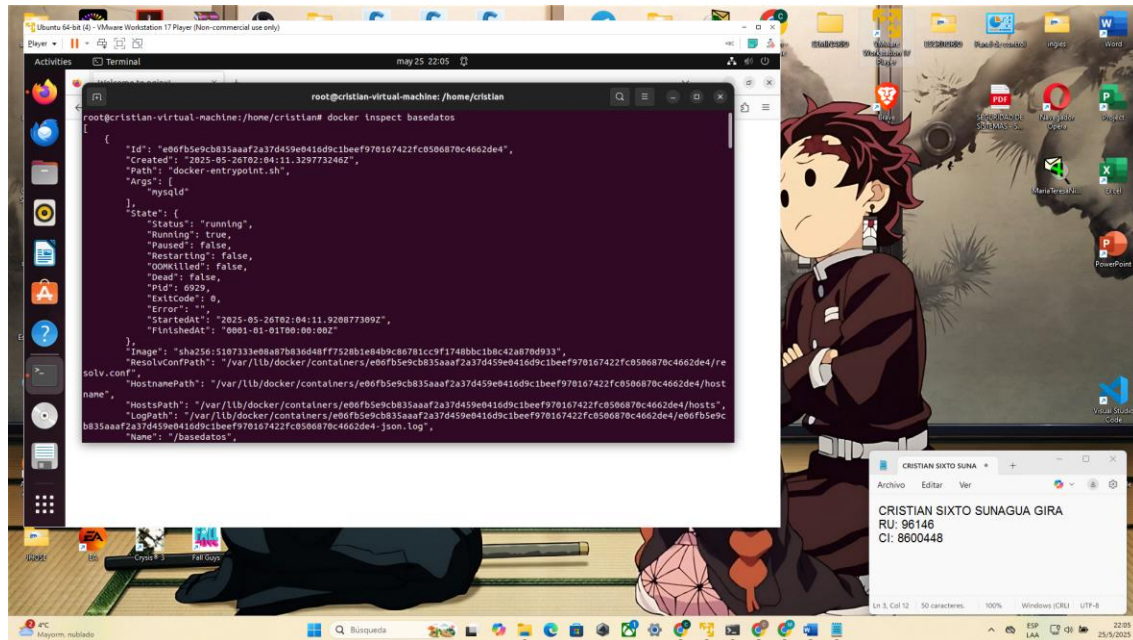
```
df9a4d85569b: Pull complete
```

```
Digest: sha256:4bcdb8c9bde88443453676cae5636f4b156d7bbae03c9145be47c2aad73bb
```

```
Status: Downloaded newer image for mysql:5.7
```

```
e0bf9e9c8b35aaaf2a37d459e041d9c1bee9f70167422fc0506070c4662de4
```

-Verificación de variables de entorno

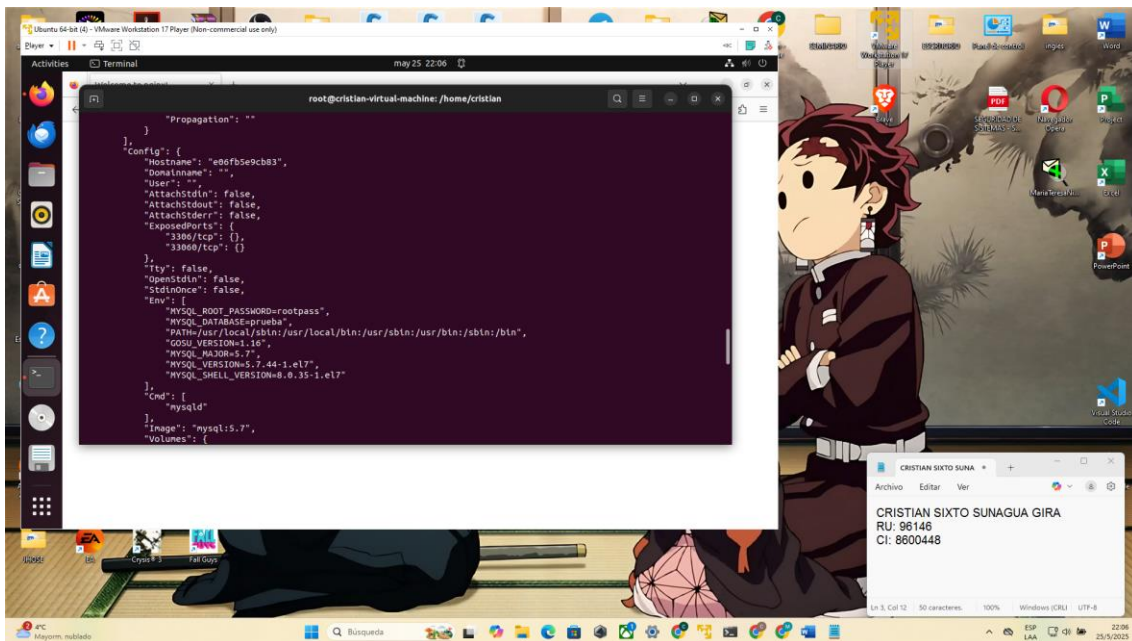


The screenshot shows a terminal window in a VM with the command `docker inspect basedatos` executed. The output is a JSON object detailing the container's configuration and state. A Notepad window in the background displays the user's name and ID.

```
root@cristian-virtual-machine: /home/cristian# docker inspect basedatos
[
  {
    "Id": "e06fb5e9cb835aaaf2a37d459eb416d9c1beef970167422fc050687bc4662de4",
    "Created": "2025-05-26T02:04:11.329773246Z",
    "Path": "docker-entrypoint.sh",
    "Args": [
      "mysqld"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 6929,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2025-05-26T02:04:11.920877309Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:5107333e0a87b836d48ff7528b1e84b9c86701cc9f1748bbc1bc42a870d933",
    "ResolveConfPath": "/var/lib/docker/containers/e06fb5e9cb835aaaf2a37d459eb416d9c1beef970167422fc050687bc4662de4/resolv.conf",
    "HostnamePath": "/var/lib/docker/containers/e06fb5e9cb835aaaf2a37d459eb416d9c1beef970167422fc050687bc4662de4/hostname",
    "HostsPath": "/var/lib/docker/containers/e06fb5e9cb835aaaf2a37d459eb416d9c1beef970167422fc050687bc4662de4/hosts",
    "LogPath": "/var/lib/docker/containers/e06fb5e9cb835aaaf2a37d459eb416d9c1beef970167422fc050687bc4662de4/e06fb5e9cb835aaaf2a37d459eb416d9c1beef970167422fc050687bc4662de4-json.log",
    "Name": "/basedatos"
  }
]
```

CRISTIAN SIXTO SUNAGUA GIRA  
RU: 96146  
CI: 8600448

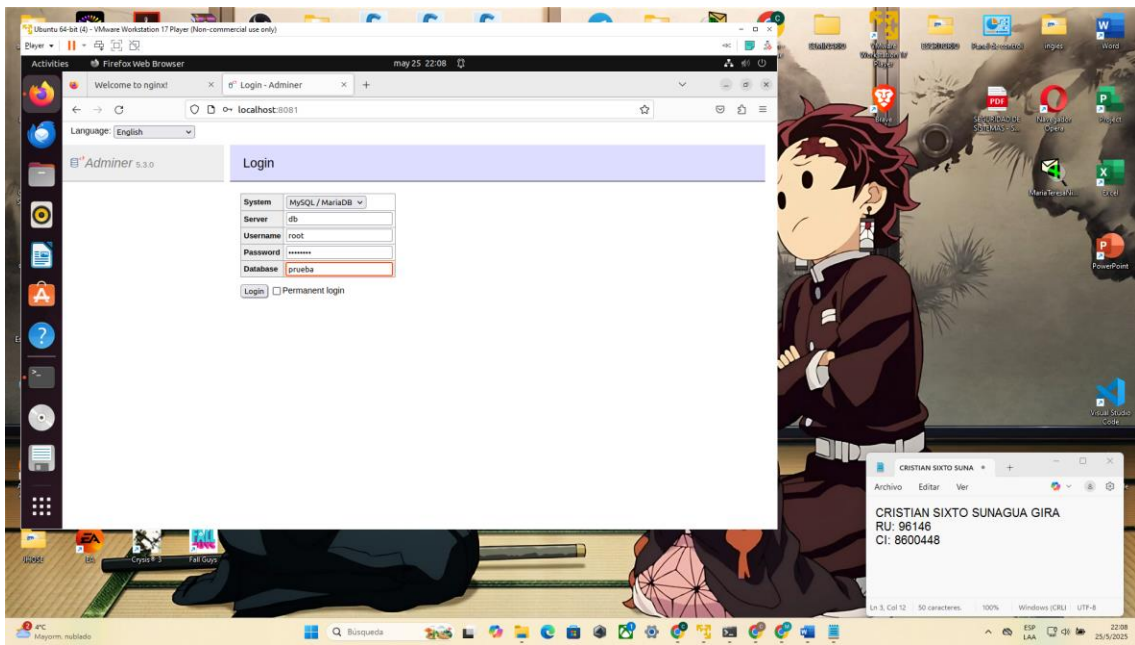
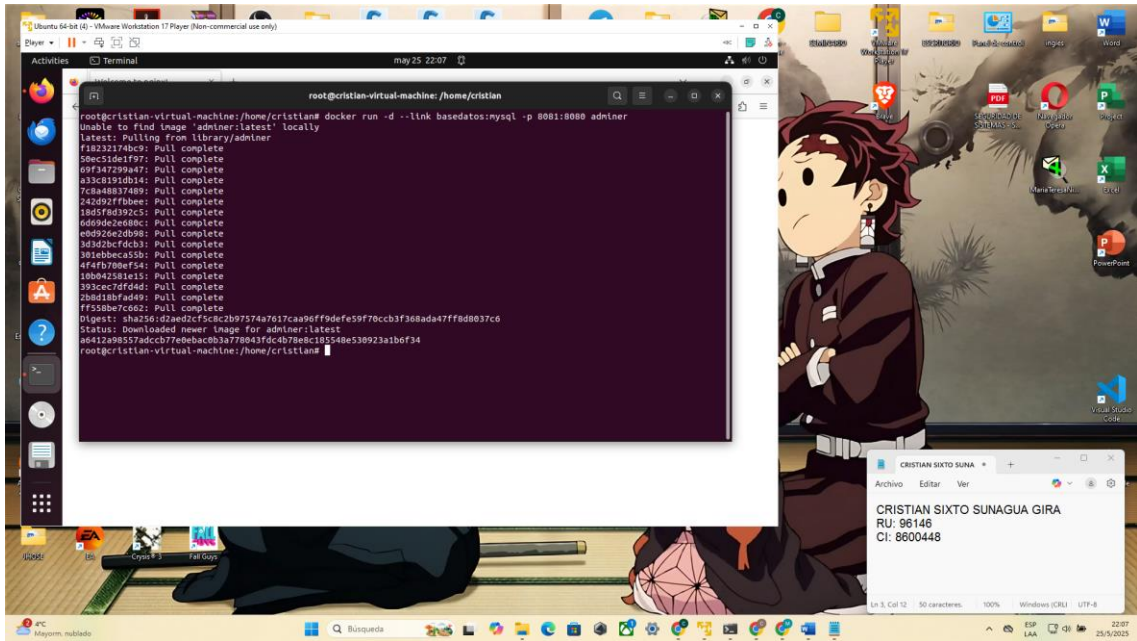
-Vamos a Config y en Env: y nos mostrara las variables



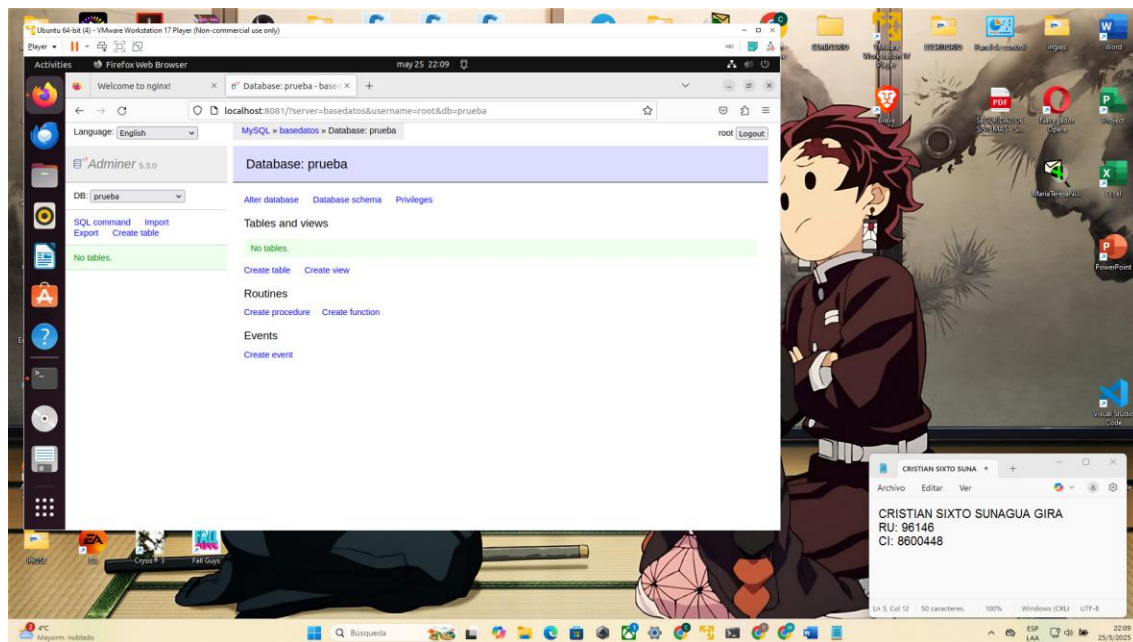
The screenshot shows a terminal window in a VM with the command `docker inspect basedatos` executed. The output is a JSON object detailing the container's configuration and state. A Notepad window in the background displays the user's name and ID.

```
root@cristian-virtual-machine: /home/cristian# docker inspect basedatos
[
  {
    "Id": "e06fb5e9cb835aaaf2a37d459eb416d9c1beef970167422fc050687bc4662de4",
    "Created": "2025-05-26T02:04:11.329773246Z",
    "Path": "docker-entrypoint.sh",
    "Args": [
      "mysqld"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 6929,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2025-05-26T02:04:11.920877309Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:5107333e0a87b836d48ff7528b1e84b9c86701cc9f1748bbc1bc42a870d933",
    "ResolveConfPath": "/var/lib/docker/containers/e06fb5e9cb835aaaf2a37d459eb416d9c1beef970167422fc050687bc4662de4/resolv.conf",
    "HostnamePath": "/var/lib/docker/containers/e06fb5e9cb835aaaf2a37d459eb416d9c1beef970167422fc050687bc4662de4/hostname",
    "HostsPath": "/var/lib/docker/containers/e06fb5e9cb835aaaf2a37d459eb416d9c1beef970167422fc050687bc4662de4/hosts",
    "LogPath": "/var/lib/docker/containers/e06fb5e9cb835aaaf2a37d459eb416d9c1beef970167422fc050687bc4662de4/e06fb5e9cb835aaaf2a37d459eb416d9c1beef970167422fc050687bc4662de4-json.log",
    "Name": "/basedatos"
  }
]
```

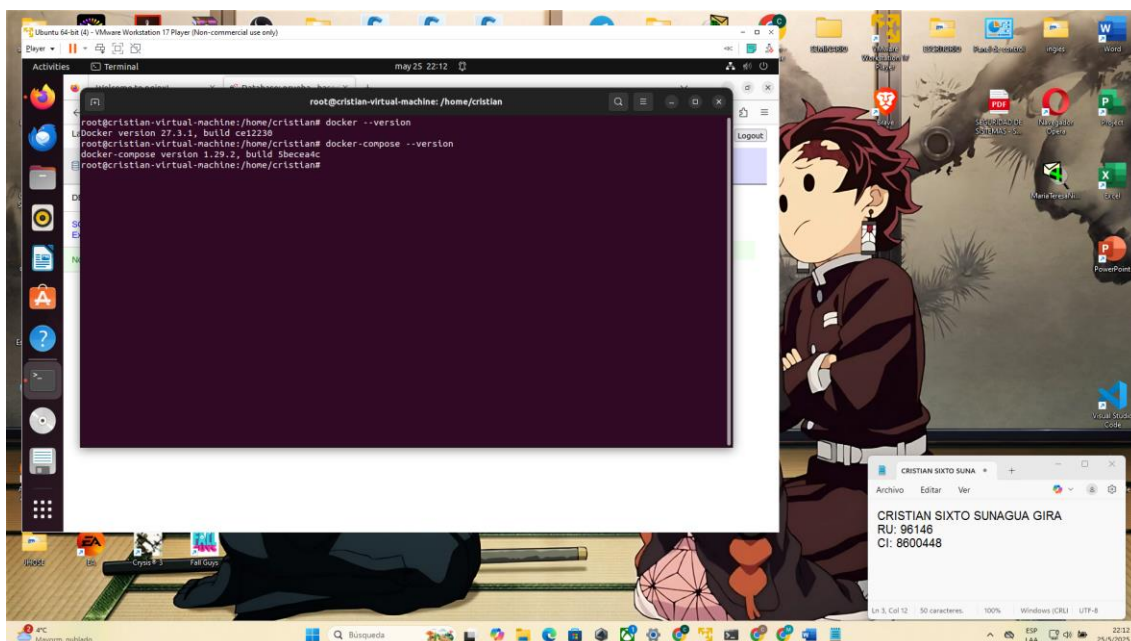
CRISTIAN SIXTO SUNAGUA GIRA  
RU: 96146  
CI: 8600448

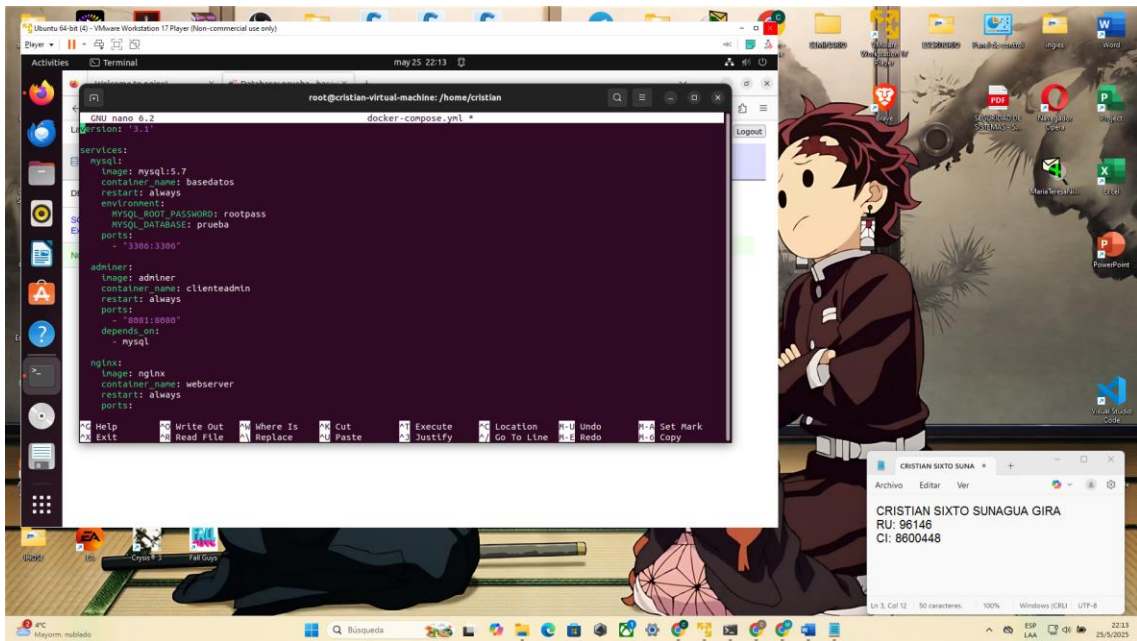
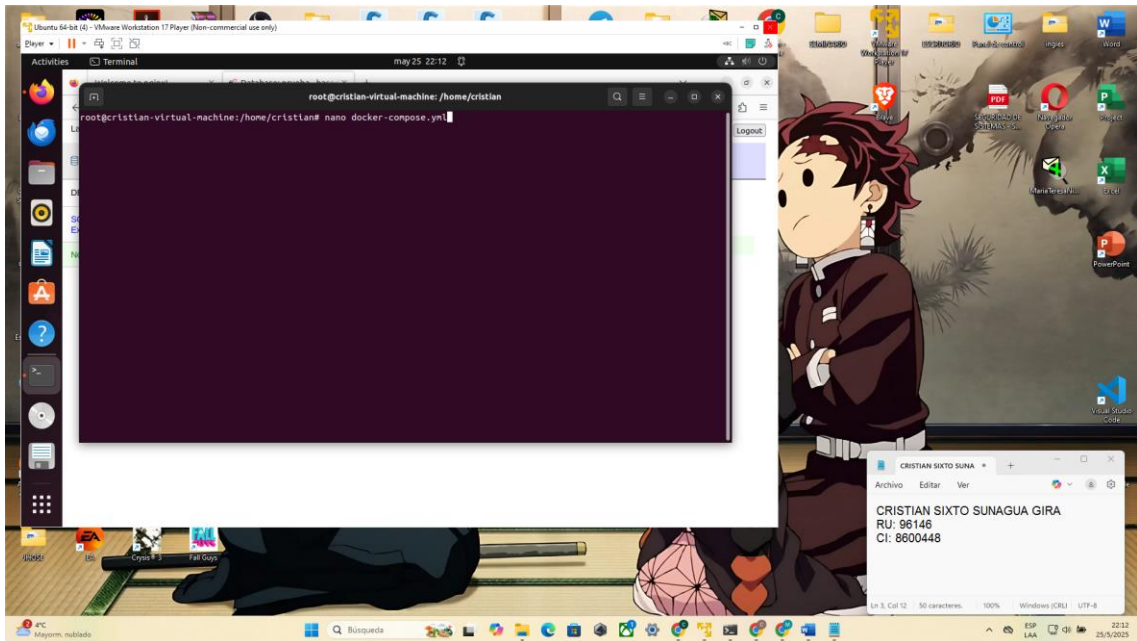


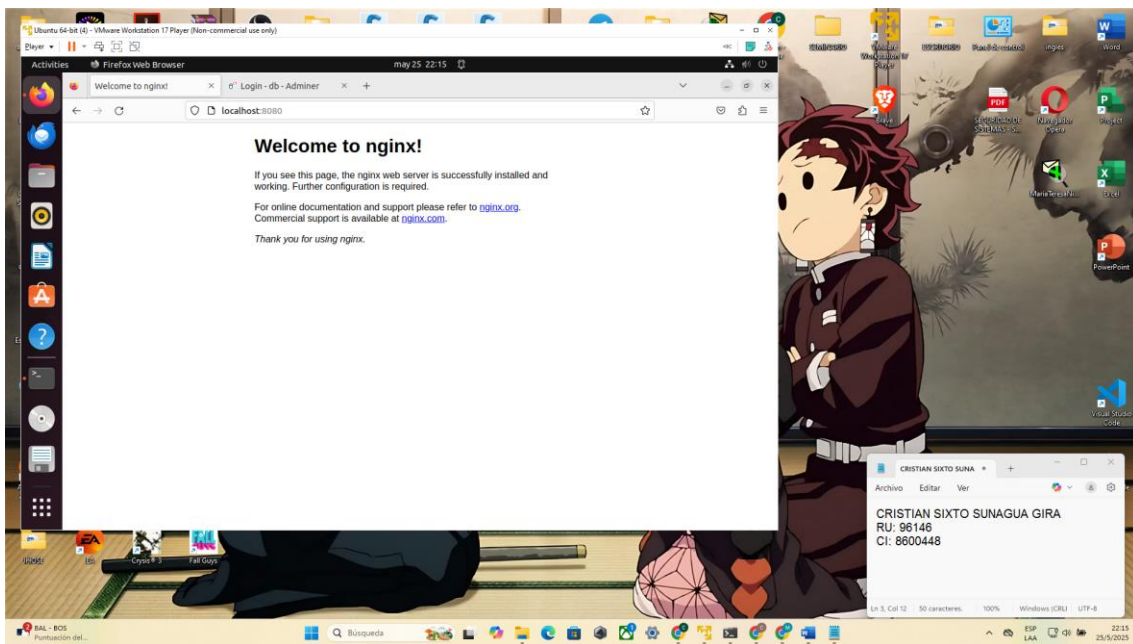
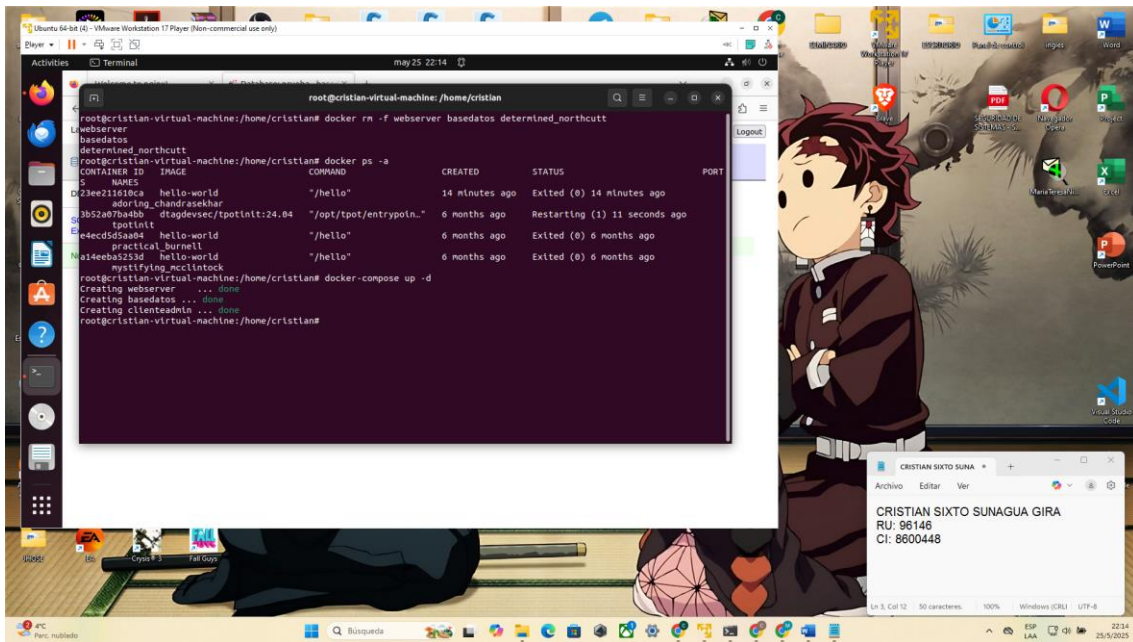




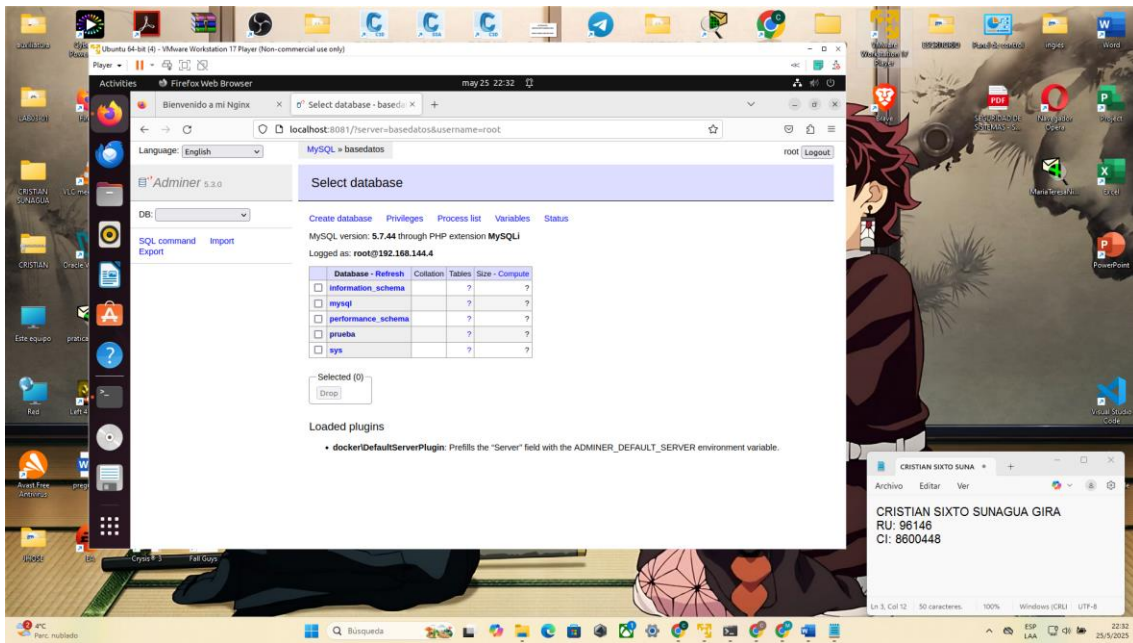
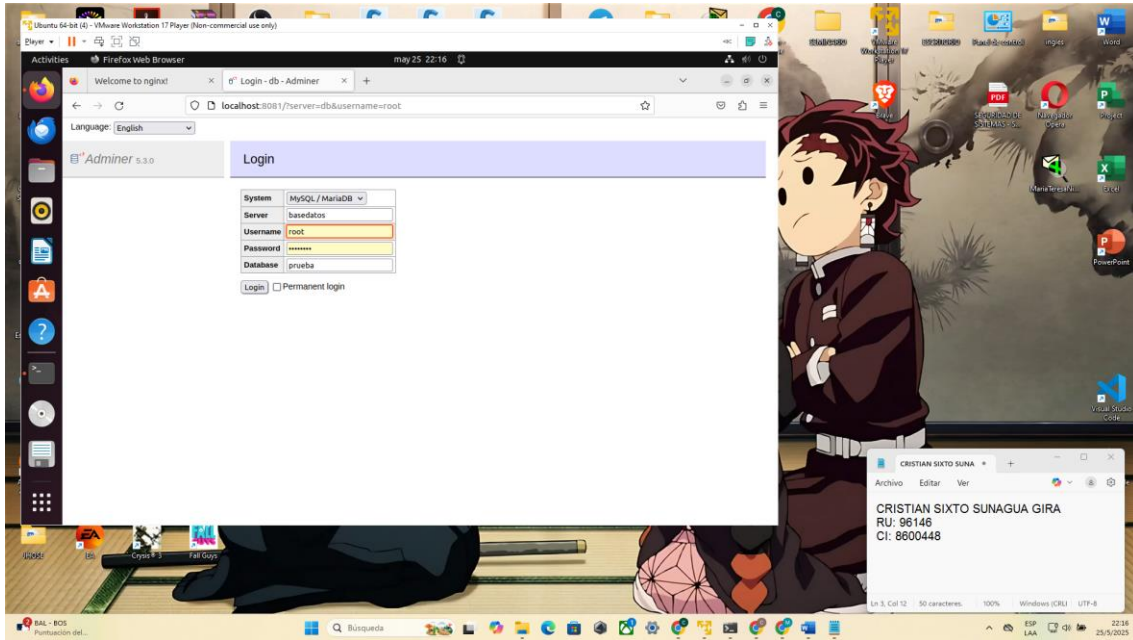
## RETO







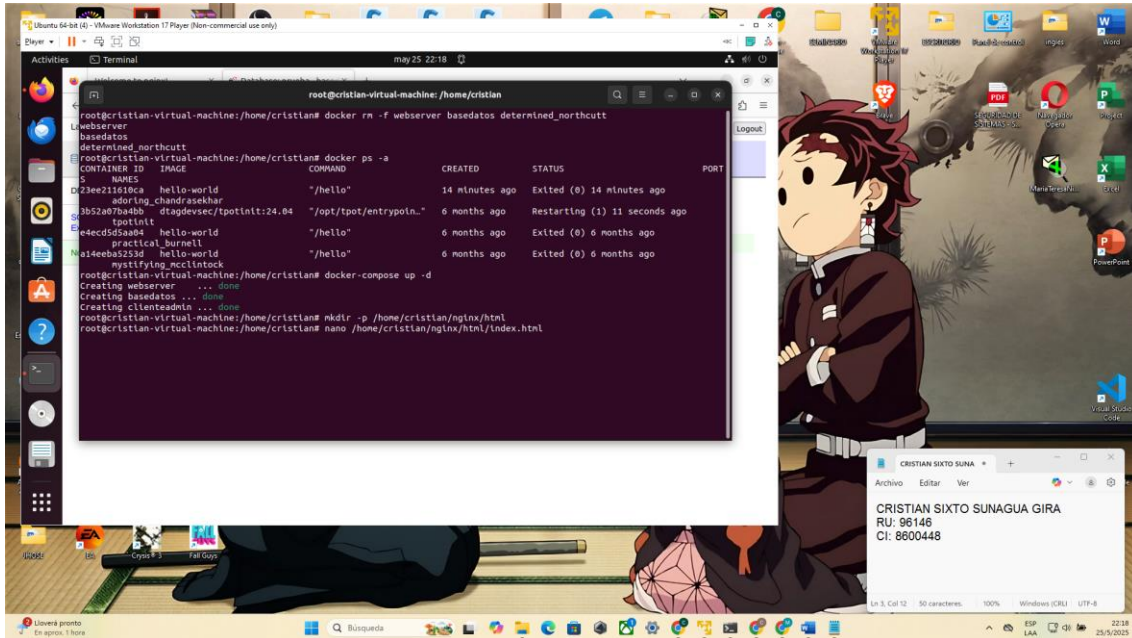




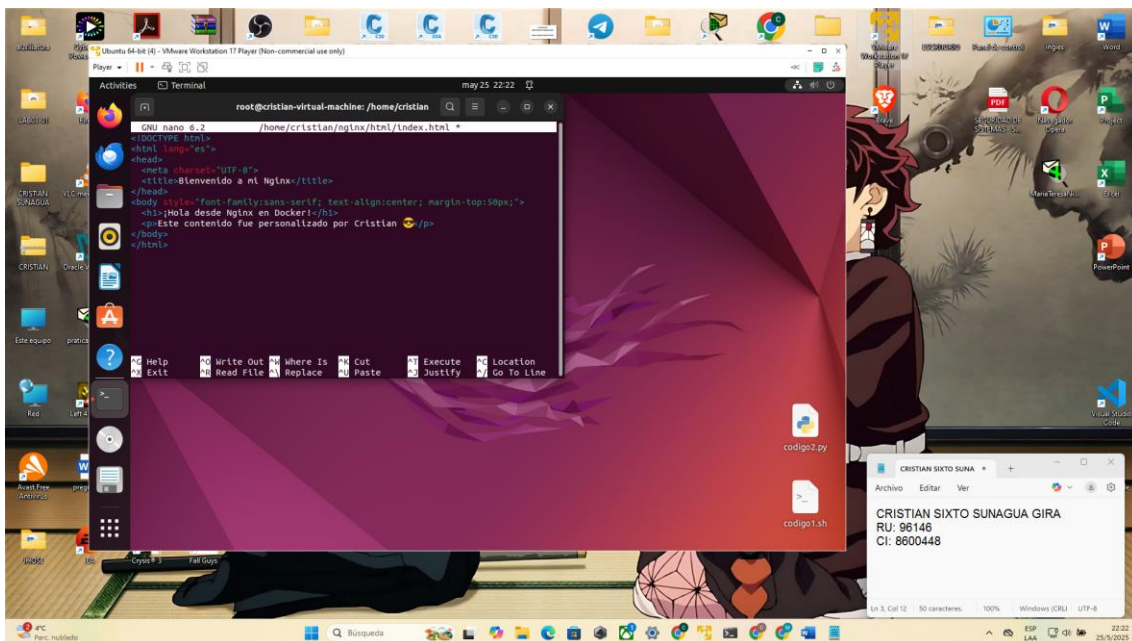
## EXTRA

Personalizar el contenido que se sirve en Nginx

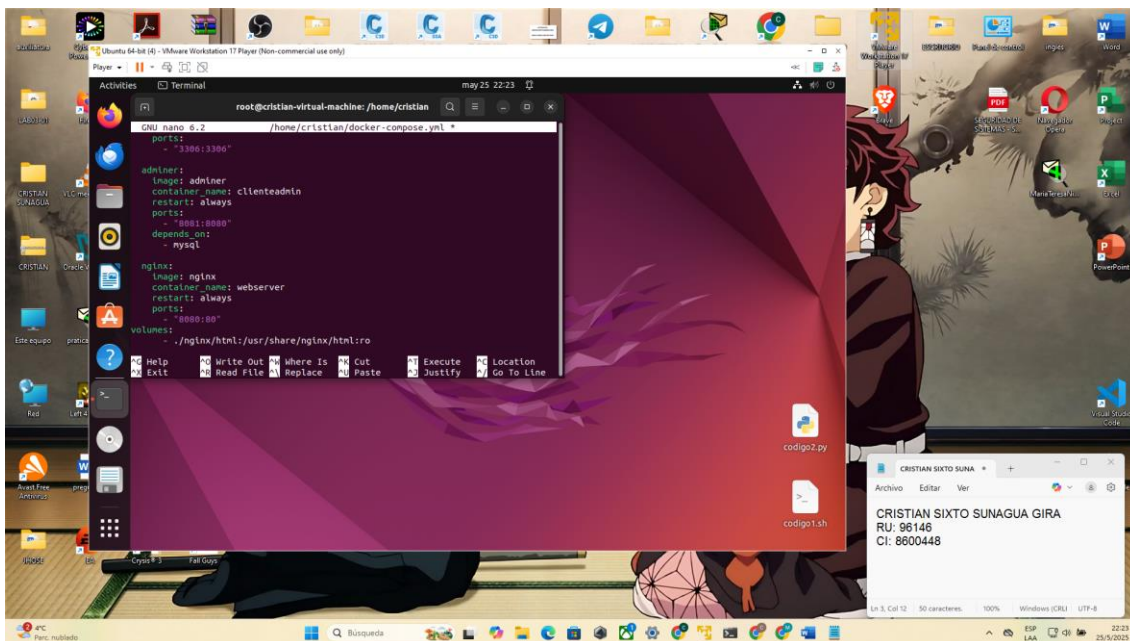
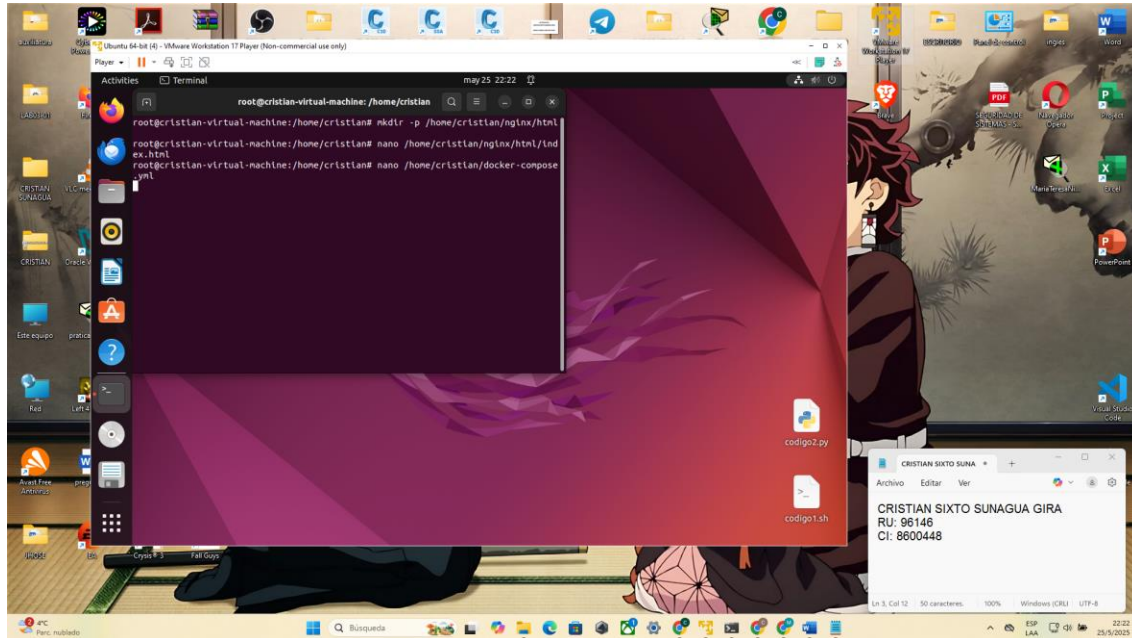
-Creamos una carpeta donde es html y un nano index.html



-En el nano estará el siguiente código que se vera en Ngnix

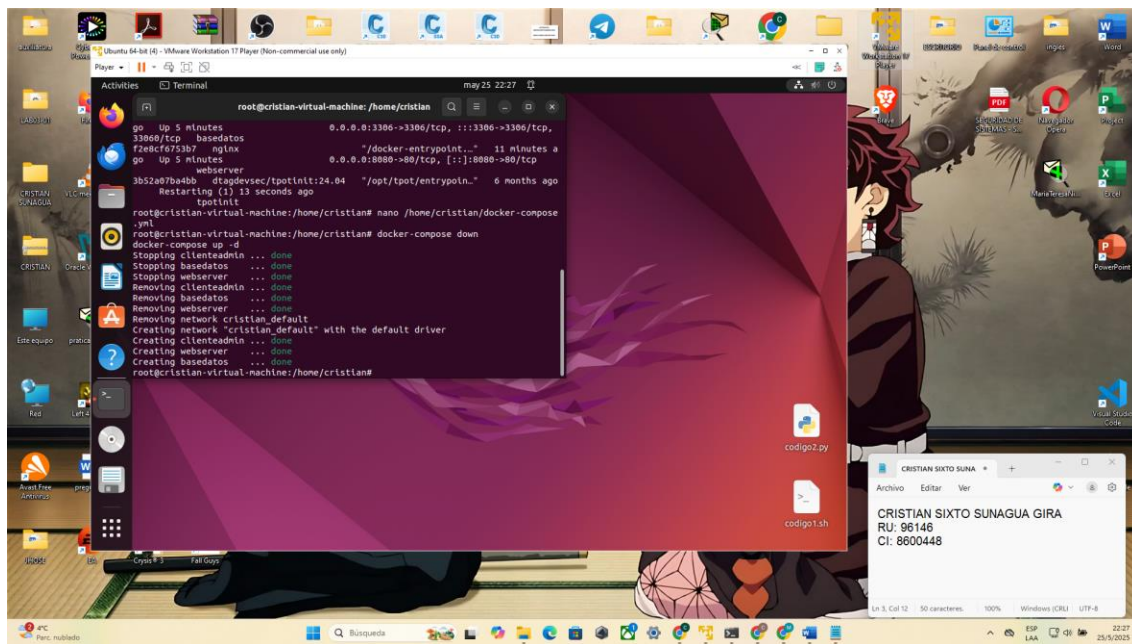


-En el nano docker -compose.yml agregaremos volumes con la dirección del html.





-Por ultimo lo volvemos a levantar



-Ahora entrando al <http://localhost:8080> nos mostrara el html que estamos llegando

