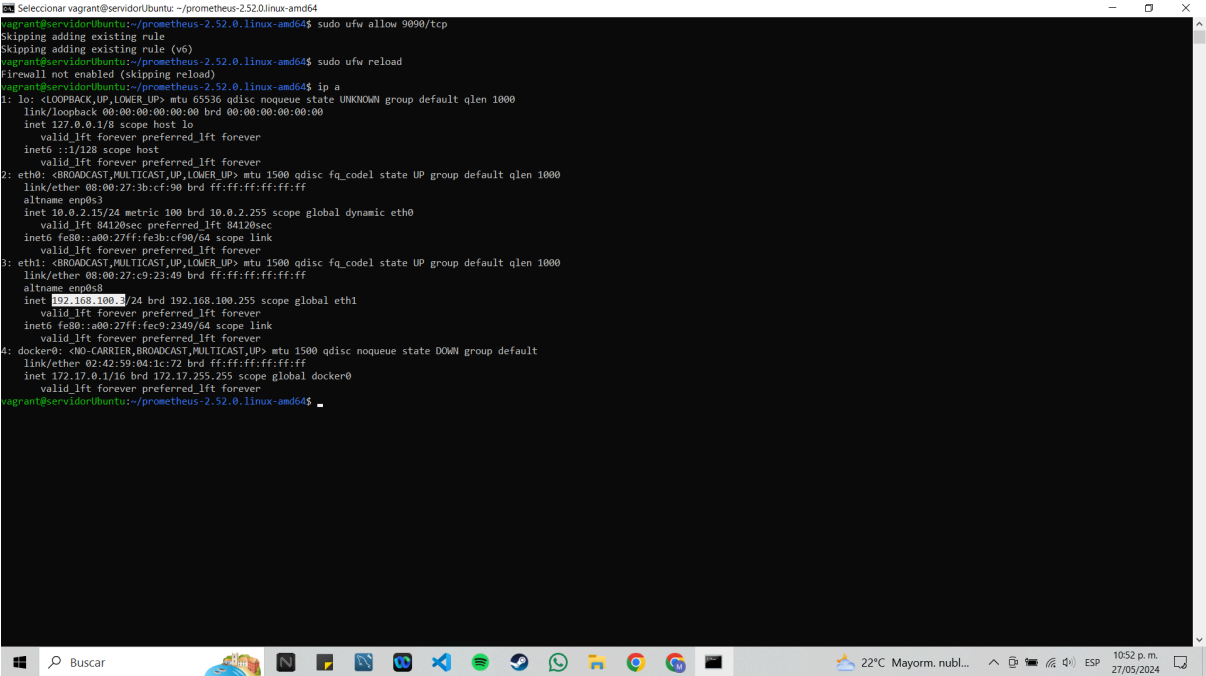


# EVIDENCIA PARCIAL FINAL SERVICIOS TELEMÁTICOS

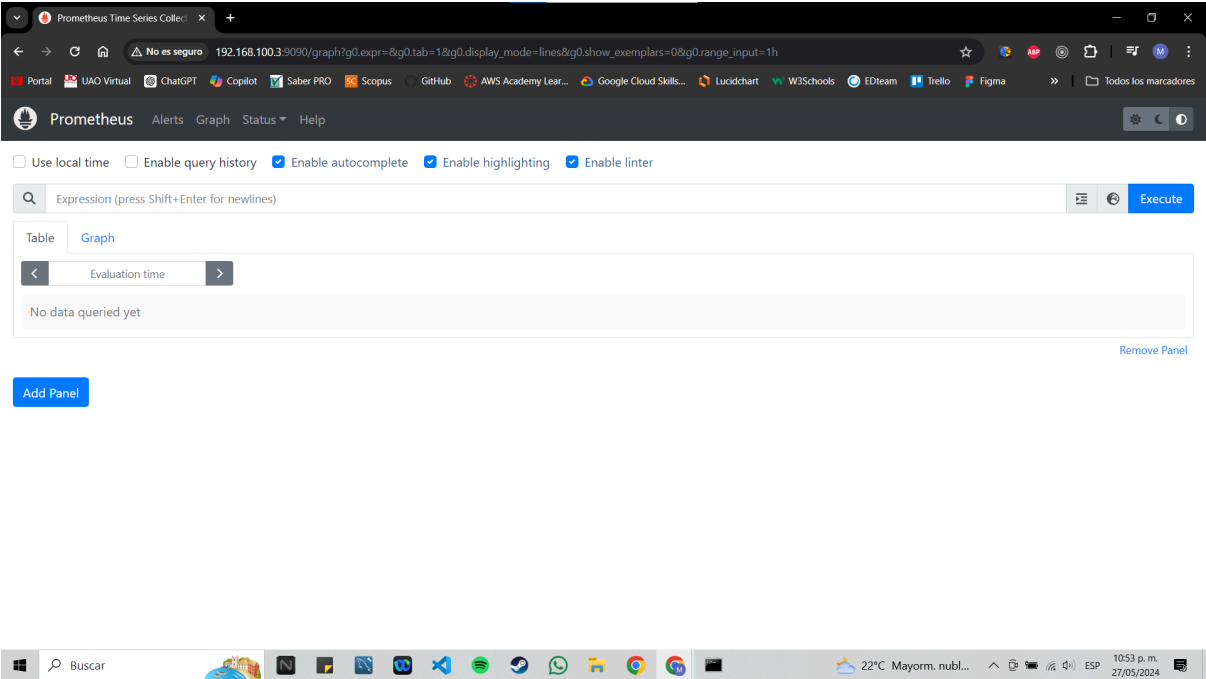
## Mauricio Córtes Vásquez 2171439

Configuración de Prometheus en Vagrant con servidorUbuntu.

(img1) Evidencia de Consola



(img2) Evidencia de Front-end



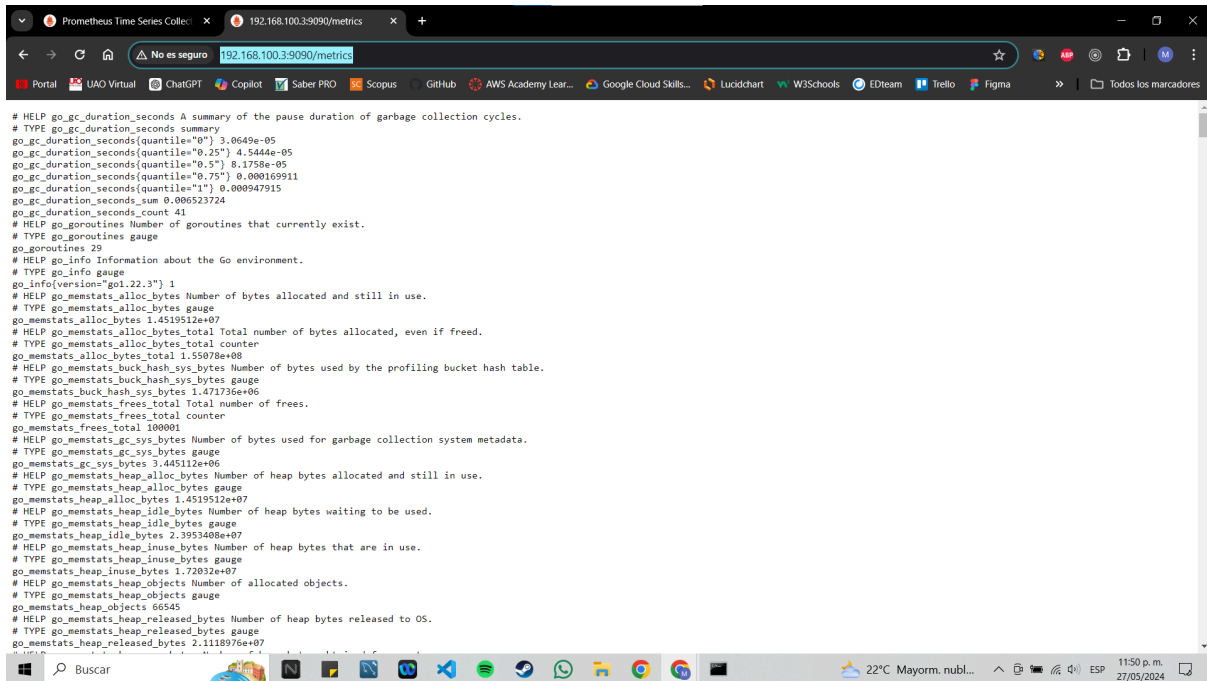
# EVIDENCIA PARCIAL FINAL SERVICIOS TELEMÁTICOS

**Mauricio Córtes Vásquez 2171439**

Para evidenciar las Metricas que se pueden desde el sistema debemos ingresar a <http://192.168.100.3:9090/metrics> donde se evidenciaran las metricas `node_cpu_seconds_total` y `node_exporter_build_info`.

(img3) Evidencia de Metricas buscandolas directamente con la URL

<http://192.168.100.3:9090/metrics>



```
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 3.0649e-05
go_gc_duration_seconds{quantile="0.25"} 4.5444e-05
go_gc_duration_seconds{quantile="0.5"} 0.1758e-05
go_gc_duration_seconds{quantile="0.75"} 0.000169911
go_gc_duration_seconds{quantile="1"} 0.000947915
go_gc_duration_seconds_sum 0.006523724
go_gc_duration_seconds_count 41
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 29
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.22.3"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.4519512e+07
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 1.55078e+08
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.471736e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 1000001
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 3.445112e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 1.4519512e+07
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 2.3953408e+07
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 1.72032e+07
# HELP go_memstats_heap_objects Number of allocated objects.
# TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 66543
# HELP go_memstats_heap_released_bytes Number of heap bytes released to OS.
# TYPE go_memstats_heap_released_bytes gauge
go_memstats_heap_released_bytes 2.1118976e+07
```

# EVIDENCIA PARCIAL FINAL SERVICIOS TELEMÁTICOS

## Mauricio Córtes Vásquez 2171439

### Docker y Docker Compose

Una vez clonada la app ya que no se tenía instalada en el dispositivo y realizada su configuración procedemos a realizar la construcción de la imagen Docker.

(img4) Con el comando **docker build -t miniwebapp** una vez ya creado el Dockerfile.

```
vagrant@servidorUbuntu:~$ docker build -t miniwebapp .
[+] Building 31.7s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/node:14-alpine
=> transferring dockerfile: 357B
=> [internal] load metadata for docker.io/library/node:14-alpine
=> transferring context: 2B
=> [1/4] FROM docker.io/library/node:14-alpine@sha256:434215b487a329c9e867202ff89e704d3a75e554822e07f3e0c0f9e066121b33
=> resolve docker.io/library/node:14-alpine@sha256:434215b487a329c9e867202ff89e704d3a75e554822e07f3e0c0f9e066121b33
=> sha256:e5f6ac395a62ec277102af9e5283f6edb3b3e4f20f798e3ce7e425be226ba0 2.37MB / 2.37MB
=> sha256:434215b487a329c9e867202ff89e704d3a75e554822e07f3e0c0f9e066121b33 1.43kB / 1.43kB
=> sha256:4e84c956cd72fa9ed14a8b2939a734364c2b0042485e98e1b97175c734f454b 1.1kB / 1.1kB
=> sha256:0dac3dc27b1ad570e6c3a77cd29e88e7138ff0cad31b2ec5a0f222f6e971bd0 6.44kB / 6.44kB
=> sha256:f56be85fc22e46face30e2c3de3f7e7c15f8fd7c4e5add29d7f64b87abdaa09 3.37MB / 3.37MB
=> sha256:8f665685b215c7da9f164545f1bbdd74d800af7700267db31fe9345c0c8f6b08 37.17MB / 37.17MB
=> extracting sha256:f56be85fc22e46face30e2c3de3f7e7c15f8fd7c4e5add29d7f64b87abdaa09
=> sha256:561cb9653d56a9725be56e02128e4e96fb434a8b4bddecf2bdeb479a225feaf 448B / 448B
=> extracting sha256:8f665685b215c7da9f164545f1bbdd74d800af7700267db31fe9345c0c8f6b08
=> extracting sha256:e5f6ac395a62ec277102af9e5283f6edb3b3e4f20f798e3ce7e425be226ba0
=> [internal] load build context
=> transferring context: 178.70MB
=> [2/4] WORKDIR /app
=> [3/4] COPY . .
=> [4/4] RUN npm install
=> exporting to image
=> exporting layers
=> writing image sha256:7174fd2c0ab04fab811cf19a15b2d5864d5826d828d5688b1f61d19ebd3a0c
=> naming to docker.io/library/miniwebapp
vagrant@servidorUbuntu:~$
```

Por consiguiente, procedemos a ejecutar el Docker Compose una vez dentro del directorio de MiniWebApp.

(img5)

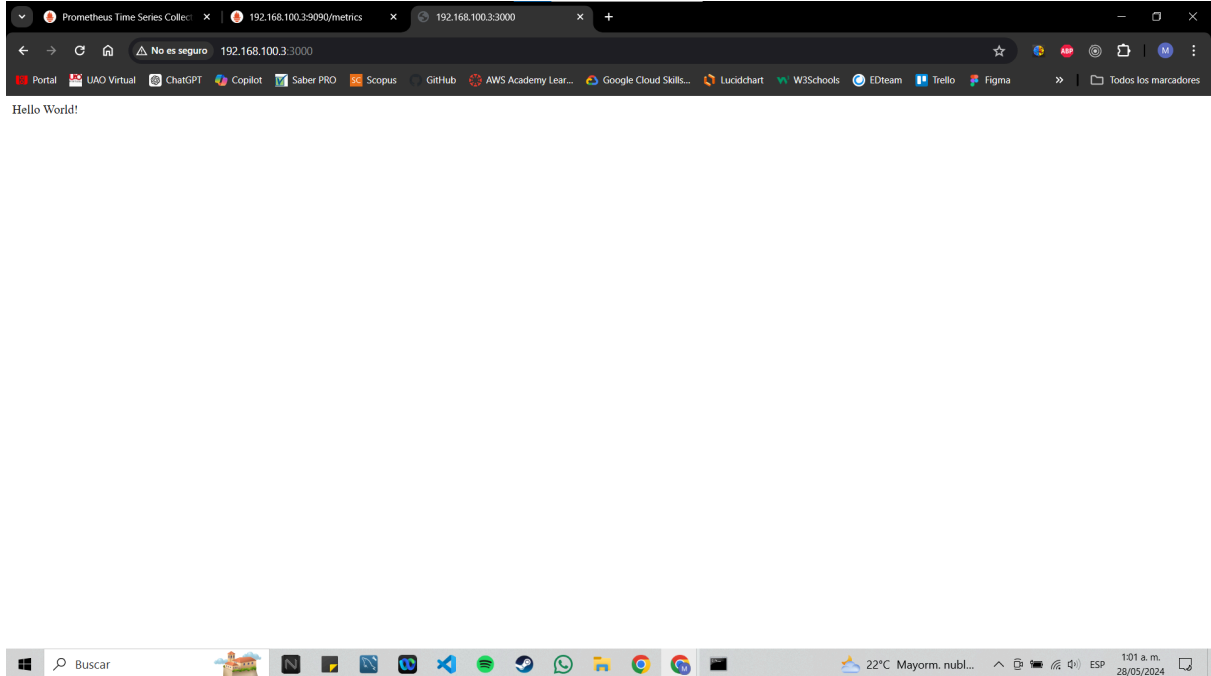
```
vagrant@servidorUbuntu:~$ docker-compose up
[+] Building 26.3s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load metadata for docker.io/library/node:14-alpine
=> transferring dockerfile: 357B
=> [internal] load metadata for docker.io/library/node:14-alpine
=> transferring context: 2B
=> [1/4] FROM docker.io/library/node:14-alpine@sha256:434215b487a329c9e867202ff89e704d3a75e554822e07f3e0c0f9e066121b33
=> [internal] load build context
=> transferring context: 178.70MB
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . .
=> [4/4] RUN npm install
=> exporting to image
=> exporting layers
=> writing image sha256:3c356328175db542c49da048519df50c3ace5bfff91a865b05994c7c19faa64b
=> naming to docker.io/library/vagrant_web
[+] Running 2/1
  Network vagrant_default Created
  Container vagrant-web-1 Created
Attaching to vagrant-web-1
vagrant-web-1 | npm ERR! code ENOENT
vagrant-web-1 | npm ERR! syscall open
vagrant-web-1 | npm ERR! path /app/package.json
vagrant-web-1 | npm ERR! errno -2
vagrant-web-1 | npm ERR! enoent: no such file or directory, open '/app/package.json'
vagrant-web-1 | npm ERR! This is related to npm not being able to find a file.
vagrant-web-1 | npm ERR! enoent
vagrant-web-1 | npm ERR! A complete log of this run can be found in:
vagrant-web-1 | npm ERR! /root/.npm/_logs/2024-05-28T05_25_05_346Z-debug.log
vagrant-web-1 exited with code 254
vagrant@servidorUbuntu:~$ docker-compose up
```

# EVIDENCIA PARCIAL FINAL SERVICIOS TELEMÁTICOS

## Mauricio Córtes Vásquez 2171439

Aquí podemos corroborar que la aplicación esta funcionando correctamente por el puerto 3000 desde Docker.

(img6)



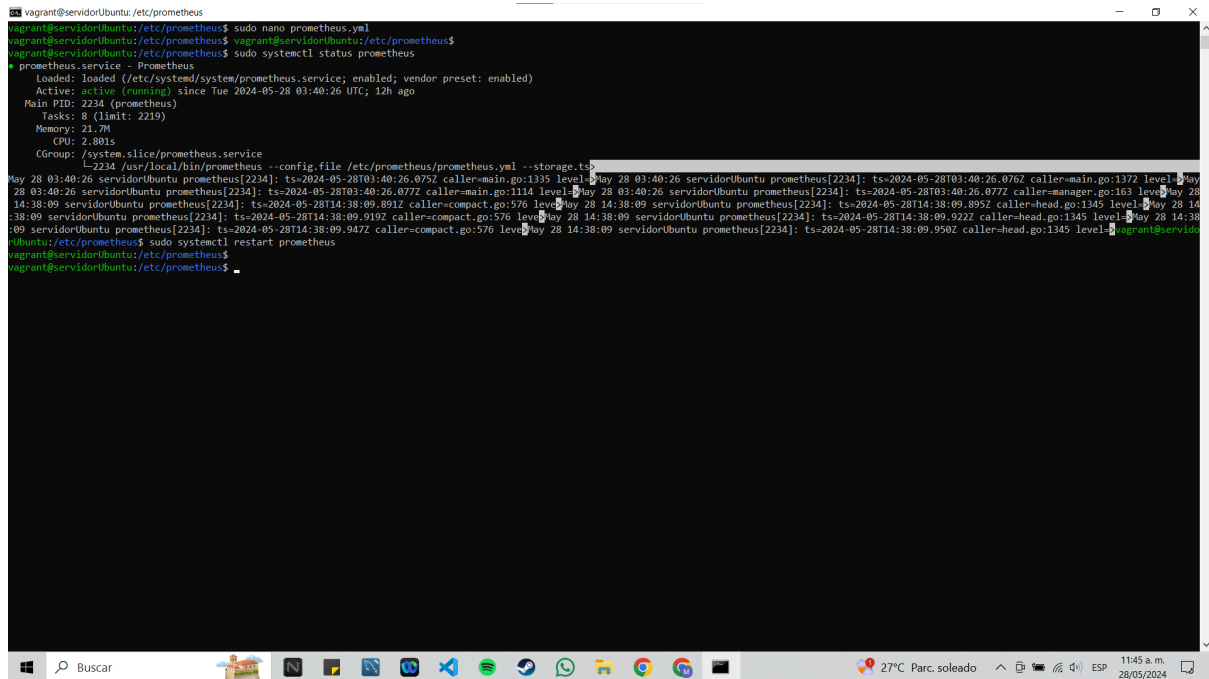
# EVIDENCIA PARCIAL FINAL SERVICIOS TELEMÁTICOS

**Mauricio Córtes Vásquez 2171439**

## Punto Adicional...

Una vez configurado el prometheus con Garfana could pasamos a divisar dicha configuración.

(img7)



```
vagrant@servidorUbuntu:/etc/prometheus$ sudo nano prometheus.yml
vagrant@servidorUbuntu:/etc/prometheus$ vagrant@servidorUbuntu:/etc/prometheus$
vagrant@servidorUbuntu:/etc/prometheus$ sudo systemctl status prometheus
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-05-28 03:40:26 UTC; 12h ago
     Main PID: 2234 (prometheus)
       Tasks: 8 (limit: 2219)
      Memory: 21.7M
         CPU: 2.801s
    CGroup: /system.slice/prometheus.service
            └─2234 /usr/local/bin/prometheus --config.file /etc/prometheus/prometheus.yml --storage.tsdb
May 28 03:40:26 servidorUbuntu prometheus[2234]: ts=2024-05-28T03:40:26.075Z caller=main.go:1335 level=info May 28 03:40:26 servidorUbuntu prometheus[2234]: ts=2024-05-28T03:40:26.076Z caller=main.go:1372 level=info May 28 03:40:26 servidorUbuntu prometheus[2234]: ts=2024-05-28T03:40:26.077Z caller=main.go:1114 level=info May 28 03:40:26 servidorUbuntu prometheus[2234]: ts=2024-05-28T03:40:26.077Z caller=manager.go:163 level=info May 28 14:38:09 servidorUbuntu prometheus[2234]: ts=2024-05-28T14:38:09.8912Z caller=compact.go:576 level=info May 28 14:38:09 servidorUbuntu prometheus[2234]: ts=2024-05-28T14:38:09.8952Z caller=head.go:1345 level=info May 28 14:38:09 servidorUbuntu prometheus[2234]: ts=2024-05-28T14:38:09.9197Z caller=compact.go:576 level=info May 28 14:38:09 servidorUbuntu prometheus[2234]: ts=2024-05-28T14:38:09.9222Z caller=head.go:1345 level=info May 28 14:38:09 servidorUbuntu prometheus[2234]: ts=2024-05-28T14:38:09.9472Z caller=compact.go:576 level=info May 28 14:38:09 servidorUbuntu prometheus[2234]: ts=2024-05-28T14:38:09.9502Z caller=head.go:1345 level=info
vagrant@servidorUbuntu:/etc/prometheus$ sudo systemctl restart prometheus
vagrant@servidorUbuntu:/etc/prometheus$
vagrant@servidorUbuntu:/etc/prometheus$
```

En pantalla aparecerá algo como...

(img8)

